25784/H/06

MILIK PERPUSTAKAAN ITS

PEMANFAATAN SMS UNTUK MENYAMPAIKAN INFORMASI PEMBATALAN DAN KETERLAMBATAN JADWAL PESAWAT TERBANG DAN KERETA API MELALUI SALURAN TELEVISI

TUGAS AKHIR

RSTf
005.1
ATA
P-1
2006

Institut Teknologi Sepuluh Nopember

Disusun Oleh :

FEYZUL ATHIR
5101 109 027

PERPUSTAKAAN ITS

Tgl. Terima	18-5-06
Terima Dari	H
No. Agenda Prp.	225116

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2006

PEMANFAATAN SMS UNTUK MENYAMPAIKAN INFORMASI PEMBATALAN DAN KETERLAMBATAN JADWAL PESAWAT TERBANG DAN KERETA API MELALUI SALURAN TELEVISI

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui/Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Siti Rochimah, M.T **NIP. 132 103 631**	**Ahmad Hoirul Basori, S.Kom**

SURABAYA
03 MEI 2006

ABSTRAK

Kebutuhan masyarakat akan informasi pesawat terbang dan kereta api seiring dengan perkembangan masyarakat dan globalisasi menuntut informasi yang cepat dan akurat. Untuk memenuhi kebutuhan masyarakat umum akan hal tersebut diperlukan suatu media informasi yang umum dan dapat menginformasikan pembatalan dan keterlambatan jadwal tersebut. Televisi merupakan media informasi audio dan visual yang sangat umum di masyarakat. Televisi dapat berintegrasi dengan teknologi informasi sehingga televisi merupakan media yang efektif untuk menampilkan hasil olahan dari teknologi informasi

Tugas Akhir ini akan merancang dan membangun suatu perangkat lunak yang menyediakan informasi-informasi terkini mengenai perubahan penjadwalan sistem transportasi yaitu Pesawat Terbang dan Kereta Api khususnya tentang keterlambatan, dan pembatalan. Disamping itu untuk mendukung PT. JTV (Jawa Timur Televisi) sebagai stasiun lokal masyarakat Jawa Timur yang selain memberikan hiburan, juga dapat memberikan kemudahan informasi bagi pemirsanya.

Uji coba dilakukan dengan meginputkan data pada software pengirim berita baik pesawat maupun kereta api ke software penerima di stasiun televisi JTV untuk ditayangkan sebagai running text. Hasil uji coba perangkat lunak menunjukkan kemampuan sistem untuk mendeskripsikan, bagaimana keseluruhan aplikasi dapat berlangsung.

***Kata Kunci** :SMS, running text*

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah SWT yang telah memberi petunjuk dan rahmat sehingga penulis dapat menyelesaikan tugas akhir yang berjudul:

" **PEMANFAATAN SMS UNTUK MENYAMPAIKAN INFORMASI PEMBATALAN DAN KETERLAMBATAN JADWAL PESAWAT TERBANG DAN KERETA API MELALUI SALURAN TELEVISI** ".

Pada tugas akhir ini dibuat perangkat lunak yang dapat memberikan kemudahan bagi masyarakat dalam memberikan informasi-informasi terkini mengenai perubahan penjadwalan sistem transportasi yaitu pesawat terbang dan kereta api khususnya tentang keterlambatan dan pembatalan, yang kemudian jadwal tersebut akan ditayangkan melalui saluran televisi. Perangkat lunak ini didukung dengan teknologi SMS gateway untuk memudahkan pengiriman informasi berita dari pihak penyedia transportasi kepada pihak stasiun televisi, mengenai jadwal transportasi yang mengalami pembatalan dan keterlambatan.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan dan kelemahan. Oleh karena itu penulis mengaharapkan saran dan kritik membangun untuk pengembangan kedepannya.

Akhir kata, penulis mengucapkan terimakasih kepada semua pihak yang telah membantu menyelesaikan tugas akhir ini.

Surabaya, April 2006

Penulis

UCAPAN TERIMAKASIH

Alhamdulillah, segala puji bagi Allah SWT. Secara khusus, dalam kesempatan ini pula, penulis ingin mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada semua pihak yang telah menyertai dan membantu penulis selama ini, antara lain kepada:

1. Bapak Prof. Dr. Ir. Arif Djunaidy, M.Sc, selaku Dekan Fakultas Teknologi Informasi.
2. Bapak Yudhi Purwananto, M.Kom, selaku Ketua Jurusan Teknik Informatika.
3. Ibu Ir. Siti Rochimah, M.T selaku Dosen pembimbing pertama penulis. Terima kasih banyak atas segala bantuan dan kesabaran Ibu.
4. Bapak Ahmad Hoirul Basori, S.Kom selaku Dosen pembimbing kedua penulis. Terima kasih banyak atas segala bantuan.
5. Bapak Ir Suhadi Lili, Bapak Daniel Oranova S, dan Ibu Umi Laili Yuhana, S.Kom sebagai dosen penguji.
6. Segenap dosen pengajar dan karyawan Fakultas Teknologi Informasi, Bapak Yudi TU atas berita-beritanya yang uptodate.
7. Papa dan Mama tercinta. Terimakasih atas segala kasih sayang yang tak terbatas yang engkau curahkan dan do'a yang engkau iringkan. Semoga aku masih memiliki kesempatan untuk membalas kasih sayangmu.
8. Bapak Ali, terima kasih atas doa dan amalannya. It works...
9. Mas Bagus Andris Lesmana, good bro and thanx for da support, Mas Awie, Mbak Win, Mbak Ria, adik-adik ponakanku yang lucu-lucu 3F (Farah,Feisya,Fadia) I Loveee My Families.

10. Mas Choiru Zain dan Didik Stikom, my best friend dan pahlawan tanpa tanda jasa.
11. Popon, Tatik, Nona, da good assistants.
12. Adrian dan Dani, teman baikku, ayo smangat susul aku...!!!!.
13. Reka dan Yoppi, terima kasih buat hp plus kabel datanya.
14. Ade Soediro thanx 4 being patient with ur lovely son n thanx 4 many advice that u give to me.
15. All crew Istara fm Surabaya.
16. Warga DAN Band, let's rockin da world man !!!!....a miracle will come to us, in da second album... trust me..
17. Siti malaikat rational rose.
18. Adi02, buat pinjaman laptopnya.
19. Mas Triman, Mas Sony, JTV reeeek!!!.
20. Semua pihak yang tidak dapat disebutkan satu per satu.

DAFTAR ISI

ABSTRAK iii

KATA PENGANTAR iv

UCAPAN TERIMAKASIH v

DAFTAR GAMBAR x

DAFTAR TABEL xiv

BAB I PENDAHULUAN 1

1.1. Latar Belakang 1

1.2. Tujuan dan Manfaat 3

1.3. Perumusan Masalah 3

1.4. Batasan Masalah 3

1.5. Metodologi Pembuatan Tugas Akhir 4

1.6. Sistematika Pembahasan 5

BAB II DASAR TEORI 7

2.1 Short Message Service(SMS) 7

2.1.1 SMS Gateway 7

2.1.2 Cara Kerja SMS 8

2.2 Proses Broadcasting SMS pada PT. JTV (Jawa Timur Televisi) 15

2.2.1 Konsep Broadcasting SMS pada PT. JTV (Jawa Timur Televisi) 15

2.2.2 Penerapan Teknologi Broadcast pada PT. JTV (Jawa Timur Televisi) 18

2.3 Konsep Desain Antarmuka 21

2.3.1 Desain yang Bagus 21

2.3.2 Pedoman Desain Antarmuka 22

BAB III ANALISIS DAN PERANCANGAN SISTEM 24

3.1 Deskripsi Umum Sistem 24

3.1.1 Fungsi-fungsi pada perangkat lunak 26

3.1.2 Karakteristik Pengguna 28

3.2 Pemodelan Proses 30

3.2.1 Analisa Use Case View 30

3.2.2 Desain Logical View 35

3.3 Pemodelan Data 57

3.4 Pemodelan Antarmuka 69
3.4.1 Pemodelan Antarmuka perangkat lunak pada Pesawat terbang 71
3.4.2 Pemodelan Antarmuka perangkat lunak pada Kereta Api 73
3.4.3 Pemodelan Antarmuka perangkat lunak pada TV 75
BAB IV IMPLEMENTASI 78
4.1 Implementasi Data 78
4.2 Implementasi Proses 81
4.2.1 Proses Koneksi SMS Gateway 82
4.2.2 Proses Pengiriman Jadwal Kereta Api 84
4.2.3 Proses Koneksi SMS TV 92
4.2.4 Proses Manajemen SMS TV 97
4.3 Implementasi Antarmuka 105
4.3.1 Proses Koneksi SMS Gateway 105
Operator kereta sebelum melakukan proses pengiriman sms, terlebih dahulu melakukan setting koneksi untuk sms (SMS gateway), yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas 106
4.3.2 Proses Pengiriman Jadwal Kereta Api 106
4.3.3 Proses Koneksi SMS TV 107
4.3.4 Proses Manajemen SMS TV 108
BAB V UJI COBA DAN EVALUASI 109
5.1 Lingkungan Uji Coba 109
5.2 Pelaksanaan Uji Coba 110
5.2.1 Skenario Proses Manajemen Koneksi sms Kereta 110
5.2.2 Skenario Proses Manajemen Data Kereta Api 116
5.2.3 Skenario Proses Manajemen Jadwal Kereta Api 129
5.2.4 Skenario Proses Manajemen Koneksi sms TV 137
5.2.5 Skenario Proses Manajemen SMS TV 150
5.3 Kesimpulan Uji Coba 154
BAB VI PENUTUP 155
6.1 Kesimpulan 155
DAFTAR PUSTAKA 157

LAMPIRAN TABEL PROSES SKENARIO UJI COBA PERANGKAT LUNAK 1
Tabel 5.1 Skenario Proses Manajemen Koneksi sms Kereta 1
Tabel 5.2 Skenario Proses Manajemen Data Kereta Api 2
Tabel 5.3 Skenario Proses Manajemen Jadwal Kereta Api 4
Tabel 5.4 Skenario Proses Manajemen Koneksi sms TV 5
Tabel 5.5 Skenario Proses Manajemen SMS TV 7
LAMPIRAN ACTIVITY DIAGRAM USE CASE 8

DAFTAR GAMBAR

Gambar 2.1 Skema Format SMS Deliver PDU 10
Gambar 2.2 Skema Format SMS Submit PDU 13
Gambar 2.3 Skema proses pengiriman data SMS Jadwal Penerbangan 17
Gambar 2.4 Skema proses pengiriman data SMS Jadwal Kereta Api 17
Gambar 2.5 Piranti Perangkat Keras Matrox Digisuite 20
Gambar 2.6 Desain *layout* 21
Gambar 3.1. Skema Proses Pengiriman dan Penerimaan Data Jadwal Transportasi 25
Gambar 3.2 Aktor-aktor Perangkat Lunak PPK 31
Gambar 3.3 Use Case Diagram Perangkat Lunak PPK 34
Gambar 3.4 Sequence Diagram Manajemen Data Pesawat 37
Gambar 3.5 Sequence Diagram Manajemen Jadwal Pesawat 39
Gambar 3.6 Sequence Diagram Manajemen koneksi sms Pesawat 40
Gambar 3.7 Sequence Diagram Manajemen Data Kereta 42
Gambar 3.8 Sequence Diagram Manajemen Jadwal Kereta 44
Gambar 3.9 Sequence Diagram Manajemen koneksi sms Kereta 45
Gambar 3.10 Sequence Daigram Manajemen SMS TV 47
Gambar 3.11 Sequence Diagram Manajemen koneksi sms TV 49
Gambar 3.12 Class Diagram Manajemen data pesawat 50
Gambar 3.13 Class Diagram Manajemen jadwal pesawat 51
Gambar 3.14 Class Diagram Manajemen koneksi sms pesawat 52
Gambar 3.15 Class Diagram Manajemen data kereta 53
Gambar 3.16 Class Diagram Manajemen jadwal kereta 54
Gambar 3.17 Class Diagram Manajemen koneksi sms kereta 55
Gambar 3.18 Class Diagram Manajemen sms TV 56
Gambar 3.19 Class Diagram Manajemen koneksi sms TV 57
Gambar 3.20 Conceptual Data Model Diagram *TrainInformationSystem* 58
Gambar 3.21 Physical Data Model Diagram *TrainInformationSystem* 59
Gambar 3.22 Conceptual Data Model diagram TVOnair 65

Gambar 3.23 Physical Data Model diagram TVOnair **66**
Gambar 3.24 Desain Antarmuka **70**
Gambar 3.25 Desain Antarmuka Pengiriman Jadwal Pesawat Terbang (Manajemen Jadwal Pesawat) **71**
Gambar 3.26 Desain Antarmuka Database Pesawat Terbang (Manajemen Data Pesawat) **72**
Gambar 3.27 Desain Antarmuka Setting Gateway (Manajemen Koneksi sms Pesawat) **73**
Gambar 3.28 Desain Antarmuka Pengiriman Jadwal Kereta (Manajemen Jadwal Kereta) **73**
Gambar 3.29 Desain Antarmuka Database Kereta Api (Manajemen Data Kereta) **74**
Gambar 3.30 Desain Antarmuka Setting Koneksi Gateway (Manajemen Koneksi sms Kereta) **75**
Gambar 3.31 Desain Antarmuka Penerima sms pada TV (Manajemen SMS TV) **76**
Gambar 3.32 Desain Antarmuka koneksi sms pada TV (Manajemen Koneksi sms TV) **77**
Gambar 4.1 Skrip Database TrainInformationSystem **79**
Gambar 4.2 Skrip Database TVOnair **80**
Gambar 4.3 Hubungan Antar Tabel TrainInformationSystem **81**
Gambar 4.4 Hubungan Antar Tabel Database TVOnair **81**
Gambar 4.5 Penggalan Perintah untuk Mengkoneksikan sms gateway **83**
Gambar 4.6 Penggalan Perintah untuk method *conect( )* **83**
Gambar 4.7 Penggalan Perintah untuk menyimpan setting koneksi **84**
Gambar 4.8 Penggalan Perintah untuk memanggil form TrainTourInfo yang berfungsi untuk mengisikan jadwal kereta api beserta status dan informasi **85**
Gambar 4.9 Penggalan Perintah untuk pengambilan data tour atau jadwal kereta api yaitu *RefreshTours()* dan pengambilan data status yang terjadi pada kereta api yaitu *RefreshTourStatuses()* **86**

Gambar 4.10 Penggalan Perintah proses yang terjadi pada saat button *OK* yang terdapat pada form TrainTourInfo ditekan 87
Gambar 4.11 Penggalan Perintah pengambilan data *recipient* pada table *SIMs* 88
Gambar 4.12 Penggalan Perintah pengiriman sms jadwal kereta api 90
Gambar 4.13 Penggalan Perintah untuk menampilkan jadwal kereta api yang telah mengalami proses pengiriman sms dilengkapi dengan status report sms 91
Gambar 4.14 Perintah sortir gateway 92
Gambar 4.15 Penggalan Perintah klik button properties 93
Gambar 4.16 Penggalan Perintah untuk koneksi sms gateway 94
Gambar 4.17 Penggalan Perintah pada method *connect ( )* 95
Gambar 4.18 Penggalan Perintah pada saat button OK ditekan 97
Gambar 4.19 Penggalan Perintah *NewSMSReceived( )* 97
Gambar 4.20 Penggalan Perintah *SMSRunningTextFiltering()* 99
Gambar 4.21 Penggalan Perintah *AddMessage()* yang terdapat didalam *frmRunningText* 100
Gambar 4.22 Penggalan Perintah *RefreshRunningTextQueues()* 100
Gambar 4.23 Penggalan Perintah *btnRunPauseRunningText_Click()* 101
Gambar 4.24 Penggalan Perintah *RunningTextRun()* 101
Gambar 4.25 Penggalan Perintah *timRunningText_Tick()* 102
Gambar 4.26 Penggalan Perintah *btnStopRunningText_Click()* 103
Gambar 4.27 Penggalan Perintah *RunningTextRestart()* 103
Gambar 4.28 Penggalan Perintah *ArrangeMessages()* 104
Gambar 4.29 Penggalan Perintah btnShowHideRunningText_Click() 105
Gambar 4.30 Antarmuka Setting Koneksi Gateway 105
Gambar 4.31 Desain Antarmuka Pengiriman Jadwal Kereta 106
Gambar 4.32 Desain Antarmuka koneksi sms TV 107
Gambar 4.33 Desain Antarmuka Manajemen SMS TV 108
Gambar 5.1 Menu setting koneksi 111
Gambar 5.2 Antarmuka pemilihan menu mode 111

MILIK PERPUSTAKAAN
ITS

Gambar 5.3 Antarmuka pemilihan menu protocol 112
Gambar 5.4 Antar Muka pemilihan menu port..... 112
Gambar 5.5 Antar muka pemilihan menu Baud rate..... 113
Gambar 5.6 Antar muka pengisian nomor SIM 114
Gambar 5.7 Antar muka button connect ditekan 115
Gambar 5.8 Antar muka koneksi gateway terjadi..... 115
Gambar 5.9 Antar muka Proses penyimpanan berhasil dilakukan..... 116
Gambar 5.10 Melihat Data Kereta Api..... 117
Gambar 5.11 Antar muka Add dan Edit data Kereta Api..... 118
Gambar 5.12 Tampilan data kereta api setelah mengalami penambahan data dan pengubahan (edit) data 119
Gambar 5.13 Menghapus entry data kereta api dan tampilan setelah entry data kereta api terhapus..... 120
Gambar 5.14 Melihat Data stasiun Kereta Api..... 121
Gambar 5.15 Antar muka Add dan Edit data stasiun Kereta Api..... 122
Gambar 5.16 Tampilan data stasiun kereta api setelah mengalami penambahan data dan pengubahan (edit) data..... 123
Gambar 5.17 Menghapus entry data stasiun kereta api dan tampilan setelah entry data stasiun kereta api terhapus..... 124
Gambar 5.18 Melihat Data jurusan Kereta Api 125
Gambar 5.19 Antar muka Add dan Edit data jurusan Kereta Api 126
Gambar 5.20 Tampilan data jurusan kereta api setelah mengalami penambahan data dan pengubahan (edit) data..... 127
Gambar 5.21 Menghapus entry data jurusan kereta api dan tampilan setelah entry data jurusan kereta api terhapus 128
Gambar 5.22 Main Menu 129
Gambar 5.23 Antar muka Add dan Edit data jadwal Kereta Api..... 130
Gambar 5.24 Tampilan data jadwal kereta api setelah mengalami penambahan data dan pengubahan (edit) data..... 131
Gambar 5.25 Menghapus entry data jadwal kereta api dan tampilan setelah entry data jadwal kereta api terhapus 132

Gambar 5.26 Antar muka pemilihan nomor tujuan sms pada menu Recipient 133
Gambar 5.27 Antar muka pada saat tombol Send ditekan dan terjadi perubahan status report dari Sent menjadi Delivered 134
Gambar 5.28 Antar muka pengiriman dua jadwal kereta api sekaligus..... 135
Gambar 5.29 Antar muka pada saat Sent failed terjadi 136
Gambar 5.30 Tampilan menu Settings..... 138
Gambar 5.31 Antar muka setelah tombol add ditekan 139
Gambar 5.32 menu Settings setelah mengalami penambahan koneksi gateway..... 139
Gambar 5.33 Antar muka pemilihan menu Properties..... 140
Gambar 5.34 Antar muka tombol Connect ditekan 141
Gambar 5.35 Antar muka koneksi gateway yang telah diaktifkan..... 142
Gambar 5.36 Antar muka proses penghapusan koneksi..... 144
Gambar 5.37 Antar muka Menu Advance yang tidak menggunakan filter sms 145
Gambar 5.38 Antar muka Menu Advance yang menggunakan Sender Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan..... 147
Gambar 5.39 Antar muka Menu Advance yang menggunakan Message Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan 148
Gambar 5.40 Antar muka Menu Advance yang menggunakan Description Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan 150
Gambar 5.41 Antarmuka Menu Running Texts 151
Gambar 5.42 Antarmuka tampilan Running Text 153

DAFTAR TABEL

Tabel 2.1. Tabel Daftar Perintah AT Command 9

Tabel 2.2 Tabel Komponen Service Center Address 11
Tabel 2.3 Tabel Komponen Originator Address 11
Tabel 2.4 Tabel Daftar Data Coding Scheme (DCS) 11
Tabel 2.5 Tabel Validity Period 14
Tabel 3.1 Definisi dan Tugas Aktor 31
Tabel 3.2 Daftar Use Case 33
Tabel 3.3 Penjelasan Tabel Trains 60
Tabel 3.4 Penjelasan Tabel Stations 60
Tabel 3.5 Penjelasan Tabel Tours 61
Tabel 3.6 Penjelasan Tabel TourInfos 61
Tabel 3.7 Penjelasan Tabel TourStatuses 62
Tabel 3.8 Penjelasan Tabel SIMs 62
Tabel 3.9 Penjelasan Tabel SMSCenters 63
Tabel 3.10 Penjelasan Tabel SMSs 63
Tabel 3.11 Penjelasan Tabel MessageStatuses 64
Tabel 3.12 Penjelasan Tabel SMSTourInfo 64
Tabel 3.13 Penjelasan Tabel SMSs 67
Tabel 3.14 Penjelasan Tabel SMSCenters 67
Tabel 3.15 Penjelasan Tabel TSIMs 68
Tabel 3.16 Penjelasan Tabel MessageStatuses 68
Tabel 3.17 Penjelasan Tabel RunningTexts 69

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

Bab ini akan menjelaskan latar belakang, tujuan, permasalahan, dan batasan masalah yang dibahas dalam Tugas Akhir. Selain itu bab ini juga mendefinisikan metodologi pembuatan Tugas Akhir dan sistematika laporan Tugas Akhir.

1.1. Latar Belakang

Teknologi informasi yang berkembang dengan pesat sangat mempengaruhi semua sisi kehidupan manusia. Teknologi informasi menyediakan kemudahan informasi terutama bagi kegiatan bisnis. Dengan teknologi informasi semua data dari kegiatan bisnis dapat diinformasikan dengan mudah melalui media informasi dengan cepat dan luas. Imbas teknologi informasi ini tidak hanya mencakup semua alat yang berhubungan dengan computer saja melainkan juga alat-alat yang menunjang informasi seperti handphone, faximile, email, bahkan sampai media radio dan televisi.

Televisi yang merupakan media informasi audio dan visual yang sangat umum di masyarakat. Saat ini masyarakat dapat mendapatkan layanan informasi dari televisi dengan mudah dan efisien tanpa harus keluar rumah. Televisi saat ini sudah dapat berintegrasi dengan teknologi informasi yang bersifat memberikan informasi terkini seperti berita-berita hangat, talk show, siaran langsung, game online, bahkan dapat juga memberikan informasi penjadwalan online.

Kebutuhan masyarakat akan transportasi yang mendukung kegiatan masyarakat merupakan suatu hal yang mutlak. Masyarakat yang semakin

berkembang akan semakin menuntut profesionalisme dari layanan perusahaan penyedia layanan transportasi. Dalam hal ini sebagai penyedia layanan transportasi yang dominan untuk masyarakat yaitu Pesawat terbang dan Kereta Api yang sangat perlu untuk menyediakan informasi yang cepat dan akurat. Dalam hal ini televisi merupakan media yang efektif untuk menampilkan hasil teknologi informasi, sehingga perubahan dan informasi dapat langsung diketahui oleh masyarakat tanpa membutuhkan prosedur yang sangat sulit.

Pada tugas akhir ini akan dibuat sebuah perangkat lunak yang hasilnya akan dapat memberikan informasi yang sangat dibutuhkan oleh masyarakat umum, yaitu mengenai penjadwalan yang terjadi dalam sistem transportasi Pesawat Terbang dan Kereta Api. Dalam tugas akhir ini akan dijelaskan secara terperinci mengenai keberangkatan, kedatangan, keterlambatan, dan pembatalan yang terjadi dalam kedua sistem transportasi tersebut, Namun sebenarnya fungsi perangkat lunak ini lebih ditekankan pada keterlambatan dan pembatalan. Untuk menampilkan informasi yang terjadi pada kedua sistem transportasi tersebut diperlukan sebuah media visual sebagai penunjangnya. PT. JTV (Jawa Timur Televisi) sebagai stasiun televisi lokal masyarakat Jawa Timur yang akan memberikan kemudahan bagi seluruh masyarakat dengan informasi-informasi terkini mengenai penjadwalan kedua sistem transportasi tersebut.

Proses pengiriman berita mengenai jadwal transportasi yang mengalami pembatalan dan keterlambatan bersifat *up to date* dan mengikuti arus perubahan pada jadwal transportasi tersebut, sehingga apabila setiap saat terjadi perubahan terhadap jadwal transportasi, maka pihak operator dari transportasi berkewajiban

untuk mengirimkan perubahan jadwal transportasi tersebut secara langsung kepada pihak stasiun televisi dalam bentuk sms, untuk segera ditayangkan oleh stasiun televisi kepada pemirsanya, sehingga dapat memberikan informasi terbaru mengenai perubahan-perubahan terhadap jadwal transportasi yaitu berupa pembatalan dan keterlambatan yang kemungkinan dapat terjadi setiap saat.

1.2. Tujuan dan Manfaat

Tujuan dari Tugas Akhir ini adalah membuat suatu aplikasi yang memudahkan masyarakat dalam mendapatkan informasi-informasi terkini mengenai perubahan penjadwalan sistem transportasi yaitu Pesawat Terbang dan Kereta Api khususnya tentang keterlambatan, dan pembatalan. Disamping itu untuk mendukung PT. JTV (Jawa Timur Televisi) sebagai stasiun lokal masyarakat Jawa Timur yang selain memberikan hiburan, juga dapat memberikan kemudahan informasi bagi pemirsanya.

1.3. Perumusan Masalah

Permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut.

1. Bagaimana mendapatkan data-data yang fix mengenai penjadwalan yang berasal dari basis data masing-masing sistem tranportasi baik pesawat terbang maupun kereta api
2. Bagaimana menampilkan jadwal-jadwal kereta api dan pesawat terbang yang mengalami delay pemberangkatan pada layar stasiun televisi.

1.4. Batasan Masalah

Berdasarkan permasalahan-permasalahan diatas, maka batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Pembuatan tugas akhir ini adalah studi kasus pada PT. JTV (Jawa Timur Televisi).
2. Hanya informasi delay atau canceling sistem transportasi saja yang kemudian ditayangkan oleh stasiun televisi.

1.5. Metodologi Pembuatan Tugas Akhir

Pembuatan Tugas Akhir ini terbagi menjadi beberapa tahapan sebagai berikut :

1. Studi literatur

Tahap ini merupakan tahap pencarian informasi sehubungan dengan proses penjadwalan pesawat terbang dan kereta api dan juga proses penerimaan berikut penayangan sms jadwal transportasi pada layar televisi dengan dibantu oleh stasiun televisi, dimana dalam hal ini stasiun televisi yang bersangkutan adalah PT. JTV (Jawa Timur Televisi). Literatur yang digunakan meliputi buku referensi, dokumentasi internet, dan jurnal yang berhubungan dengan proses penjadwalan pesawat terbang dan kereta api dan juga proses penerimaan berikut penayangan sms jadwal transportasi pada layar televisi dengan bantuan PT. JTV. Selain itu juga diperlukan literatur yang berhubungan dengan penggunaan SMS Gateway.

2. Analisa dan perancangan perangkat lunak

Tahap analisa dan perancangan perangkat lunak merupakan tahapan untuk melakukan analisa data-data, konsep, dan proses bisnis yang telah terkumpul dan membuat desain perangkat lunak dengan menggunakan Rational Rose. Desain meliputi *use case diagram*, *sequence diagram*, serta *class diagram*.

3. Implementasi

Tahap ini merupakan tahap untuk mengimplementasikan hasil analisa, kemudian hasil analisa tersebut diimplementasikan dengan perangkat lunak yang berfungsi untuk menyampaikan informasi pembatalan jadwal pesawat terbang dan kereta api.

4. Uji coba dan evaluasi

Tahap uji coba dan evaluasi merupakan tahap untuk memeriksa apakah hasil implementasi yang dibuat sesuai dengan analisa dan rancangan dan dilakukan evaluasi untuk mengetahui kekurangan untuk disempurnakan kemudian.

5. Penyusunan buku tugas akhir

Pada tahap ini disusun buku sebagai dokumentasi dari pengerjaan tugas akhir. Dokumentasi ini dibuat agar memudahkan pengembangan perangkat lunak ini di kemudian hari..

1.6. Sistematika Pembahasan

Dalam penulisan Tugas Akhir ini, penulis membagi dalam beberapa bab seperti berikut:

1. BAB 1 PENDAHULUAN

   Bab ini berisi latar belakang, tujuan, permasalahan, batasan permasalahan, dan metodologi, serta sistematika pembuatan Tugas Akhir.

2. BAB 2 DASAR TEORI

   Bab ini membahas mengenai teknologi yang digunakan dalam implementasi penjadwalan sistem transportasi pesawat terbang dan kereta api.

3. BAB 3 ANALISA dan PERANCANGAN SISTEM PERANGKAT LUNAK

Bab ini akan membahas analisa dan berikut proses perancangan perangkat lunak aplikasi pemanfaatan sms teknologi wavecom untuk menyampaikan informasi pembatalan jadwal pesawat terbang dan Kereta Api

4. BAB 4 IMPLEMENTASI

   Bab ini berisi tentang pembuatan aplikasi dari perancangan sistem yang telah dibuat pada Bab III.

5. BAB 5 UJI COBA DAN EVALUASI

   Bab ini menjelaskan tentang proses pengujian terhadap sistem yang telah dibuat dan melakukan evaluasi terhadap hasil pengujian tersebut.

6. BAB 6 PENUTUP

   Bab ini menjelaskan tentang kesimpulan dari sistem yang dibuat serta saran untuk kepentingan pengembangan lebih lanjut.

BAB II

DASAR TEORI

BAB II

DASAR TEORI

Bab ini menjelaskan tentang sejumlah teori dasar yang mendukung secara teknis pengerjaan aplikasi yang dikerjakan, meliputi: Short Message Service (SMS), Proses broadcasting SMS pada PT. JTV (Jawa Timur Televisi), Konsep Desain Antarmuka

2.1 Short Message Service(SMS)

Teknologi ponsel telah dilengkapi dengan sejumlah aplikasi untuk pengiriman pesan. Aplikasi tersebut antara lain Short Message Service (SMS), dan Multimedia Message Service (MMS). Aplikasi untuk pengiriman pesan ini tampaknya akan terus mengalami perkembangan yang cukup pesat.

Diantara sejumlah aplikasi ponsel untuk pengiriman pesan yang paling umum digunakan adalah SMS. Selain karena aplikasinya tersedia di semua jenis ponsel, SMS juga sangat mudah digunakan dan relatif murah biayanya.

SMS merupakan aplikasi ponsel yang menyediakan layanan untuk mengirim dan menerima pesan pendek berupa angka, huruf, atau karakter lain. Satu pengiriman SMS pesan maksimal terdiri dari 160 karakter, jika pesan yang dikimkan lebih dari 160 karakter maka pesan tersebut akan dipecah menjadi beberapa kali pengiriman, setelah sampai di tujuan akan disatukan kembali.

2.1.1 SMS Gateway

SMS *gateway* berfungsi untuk menerima dan mengirim SMS dengan menggunakan terminal mobile (misalnya modem GSM dan ponsel) pada sebuah

server. SMS *gateway* harus terlebih dulu dikonfigurasi untuk merespon pesan-pesan SMS secara otomatis tanpa membutuhkan operator manusia.

Ada 2 elemen dasar untuk membangun SMS *gateway* yakni:

- Komputer

  Komputer yang digunakan berupa server yang menyediakan layanan menerima, mengirim, dan mengolah pesan SMS.

- Ponsel

  Ponsel merupakan peralatan *mobile* yang menyediakan interkoneksi ke jaringan GSM.

2.1.2 Cara Kerja SMS

2.1.2.1 AT Command

AT Commands digunakan untuk melakukan komunikasi dengan terminal (ponsel). AT Commands untuk ponsel ditetapkan dengan adanya GSM 07.07 dan GSM 07.05. Dengan menggunakan kabel data yang tersedia pada masing masing merk ponsel, dapat dilakukan komunikasi dengan ponsel melalui PC.

Daftar perintah modem GSM bisa didapat dari vendor masing masing ponsel. Perintah perintah dasar AT untuk GSM bersifat standar, namun beberapa vendor menambahkan perintah AT yang hanya bisa dijalankan pada produk mereka. (www.forumponsel.com)

Berikut ini beberapa contoh AT Commands untuk GSM yang umum digunakan.

Tabel 2.1. Tabel Daftar Perintah AT Command

PERINTAH	FUNGSI
AT+CGMI	Menampilkan kode manufaktur (pabrik pembuat)
AT+CGMM	Menampilkan model ponsel
AT+CGMR	Menampilkan versi software ponsel
AT+CGSN	Menampilkan nomor IMEI
AT+GSN	Menampilkan nomor IMEI
AT+CHUP	Memutus percakapan
AT+CEER	Menampilkan sebab terputusnya koneksi terakhir
AT+CREG	Registrasi jaringan
AT+COPS	Registrasi jaringan
AT+CPBS	Memilih buku telepon (SIM atau mobile)
AT+CPBR	Membaca isi buku telepon
AT+CPBW	Menambah isi buku telepon
AT+CSMS	Memilih
AT+CPMS	Memilih memori penyimpanan SMS
AT+CMGF	Memilih format SMS (teks atau PDU)
AT+CSCA	Menampilkan nomor message center yg digunakan
AT+CNMI	Menampilkan SMS baru
AT+CMGL	Menampilkan daftar SMS
AT+CMGR	Membaca SMS
AT+CMGS	Mengirim SMS
AT+CMSS	Mengirim SMS dari memori
AT+CMGW	Menulis SMS ke memori
AT+CMGD	Menghapus SMS di memori
AT+CMGC	Mengirimkan perintah SMS

2.1.2.2 Mengirim dan menerima SMS

Pada dasarnya terdapat dua mode untuk mengirim dan menerima SMS, yaitu mode teks dan mode PDU (*Protocol Data Unit*). Akan tetapi, sistem mode teks tidak didukung oleh semua operator GSM maupun terminal.

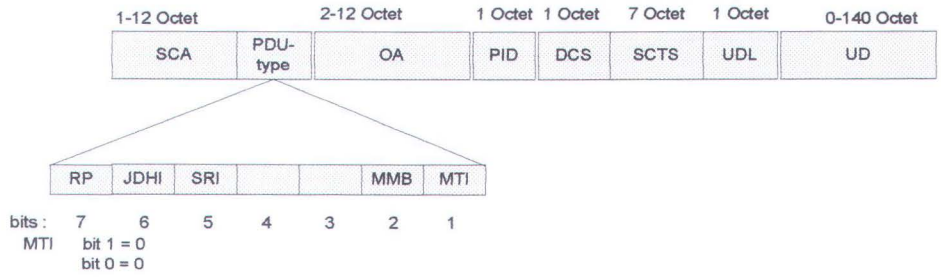
Mode teks merupakan cara termudah untuk mengirimkan pesan. Pada mode teks pesan yang kita kirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Sesungguhnya, mode teks adalah hasil *encode* yang direpresentasikan dalam format PDU. Kelemahannya kita tidak dapat menyisipkan gambar atau nada dering ke dalam pesan yang akan dikirim serta terbatasnya tipe *encoding*.

PDU Mode adalah format pesan dalam heksadesimal octet dan semi desimal octet dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8

bit) karakter . Kelebihan menggunakan mode PDU adalah kita dapat melakukan *encoding* sendiri yang tentunya harus pula didukung oleh hardware dan operator GSM, melakukan kompresi data, menambahkan nada dering dan gambar pada pesan yang akan dikirim. Beberapa tipe *encoding* yang umum digunakan adalah "PCCP437", "PCDN", "8859-1", "IRA", dan "GSM". Tipe *encoding* dapat dicek dengan perintah "AT+CSCS". Kita juga dapat menambahkan header ke dalam pesan yang akan dikirim seperti *timestamp*, nomor SMSC dan meta-informasi lainnya. Keterangan lengkap mengenai mode ini dapat dilihat dalam ETSI GSM 03.40 dan GSM 03.38. Aplikasi yang penulis kembangkan menggunakan mode PDU dengan *encoding* 7 bit default alphabet.

SMS Deliver PDU

SMS Deliver PDU adalah terminal menerima pesan yang masuk dari smsc dalam format PDU yang terdiri dari 8 header



Gambar 2.1 Skema Format SMS Deliver PDU

Message asli	"terima sms", "terima" + spasi + "sms"
PDU	059126181642240C91261885980482000030503161 4 034820AF4B23CDD0E83E6ED39

Penjelasannya sebagai berikut:

- Service Center Address (SCA)

  SCA memiliki tiga komponen utama, yaitu len, type of number dan BCD Digits

Tabel 2.2 Tabel Komponen Service Center Address

Octet	Keterangan	Nilai
Len	Panjang informasi SMSC dalam Octet	05
Type of number	Type of address dari SMSC 81h = lokal format 91h = internasional format	61
BCD Digits	Nomor SMSC. Jika panjangnya ganjil, pada akhir karakter tambahkan 0F hexa	26181642

- Protocol Data Unit (PDU) Type

  Nilai default dari PDU untuk SMS-Deliver adalah 04h. pada contoh di atas, PDU type adalah 24 yang memiliki arti :

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRI	<none>	<none>	MMS	MTI	MTI
Nilai	0	0	1	0	0	1	0	0

- Originator Address(OA)

Tabel 2.3 Tabel Komponen Originator Address

Octet	Keterangan	Nilai
Len	Panjang informasi SMSC dalam Octet	05
Type of number	Type of address dari SMSC 81h = lokal format 91h = internasional format	91
BCD Digits	Nomor SMSC. Jika panjangnya ganjil, pada akhir karakter tambahkan 0F hexa	26181642

- Protocol Identifier (PID)

  Nilai default dari PID adalah 0 = "Standart Text-SMS". Pada contoh diatas PID adalah 00.

- Data Coding Scheme (DCS)

  Pada contoh di atas DCS adalah 00, juga menentukan message class seperti pada tabel 2.4 di bawah ini.

Tabel 2.4 Tabel Daftar Data Coding Scheme (DCS)

Nilai (hexa)	Character Coding	Message Class
00	Default (7-bit)	No class
F0	Default (7-bit)	class 0 (immediately display)
F1	Default (7-bit)	Class 1 (Mobile Equipment-spucific)
F2	Default (7-bit)	Class 2 (SIM specific message)
F3	Default (7-bit)	class 3 (Terminate Equipment-specific)
F4	8-bit	class 0 (immediately display)
F5	8-bit	Class 1 (Mobile Equipment-spucific)
F6	8-bit	Class 2 (SIM specific message)
F7	8-bit	Class 3 (Terminate Equipment-specific)

Hal yang perlu diperhatikan disini, pada beberapa handphone message dengan class 0 dengan enkoding 7 bit berupa flash SMS. Sedangkan dengan enkoding 16 bit Unicode (ucs2), message yang didahului "0001" dengan class 0 berupa *blinking flash* SMS.

- Service Center Time Stamp (SCTS)

  Pada contoh di atas, SCTS adalah 30503161403482, yang berarti:

Tahun	30 menjadi 2003
Bulan	50 menjadi 05, Mei
Tanggal	31 menjadi 13
Jam	61 menjadi 16
Menit	40 menjadi 04
Detik	34 menjadi 43
Zona Waktu	82 menjadi 28, dimana 1 unit = 15 menit. (15x28)/60 = 7, menjadi GMT+07.00

  Penulisan di atas dibalik (*swapped nibble presentation*) 30 menjadi 03, dan seterusnya. Arti SCTS di atas sama dengan 13 Mei 2003 16:04:43 GMT+07.00

- User Data Length (UDL)

  Pada contoh di atas, UDL adalah 0A. UDL menunjukkan panjang User Data (UD).

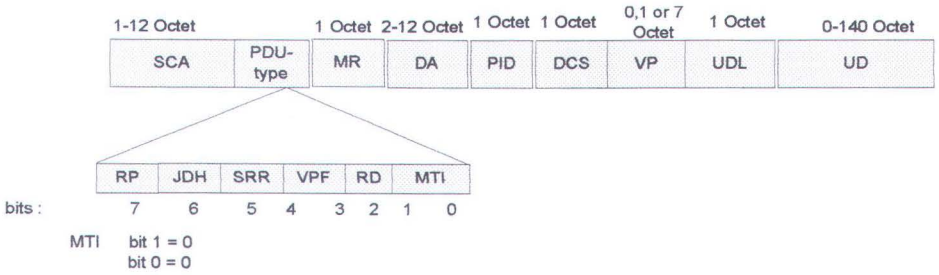
- User Data (UD)

  Pesan "terima sms" yang dikodekan menjadi 7 bit default alphabet (septet) harus di enkode menjadi 8 bit (octet) untuk SMS-Transfer. Pada Contoh di atas, UD adalah "F4B23CDD0E83E6ED39".

ITS

SMS Submit PDU

SMS Deliver PDU adalah terminal menerima pesan yang masuk dari smsc dalam format PDU



Gambar 2.2 Skema Format SMS Submit PDU

Message asli	"kirim sms", "kirim" + spasi + "sms"
PDU	0011000C91261885980482000OAA09EBB43CDD06 CDDB73 (23 octet)

Penjelasannya sebagai berikut:

- Service Center Address (SCA)

  SCA memiliki tiga komponen utama, yaitu len, type of number dan BCD Digits

Octet	Keterangan	Nilai
Len	Panjang informasi SMSC dalam Octet	05
Type of number	Type of address dari SMSC 81h = lokal format 91h = internasional format	61
BCD Digits	Nomor SMSC. Jika panjangnya ganjil, pada akhir karakter tambahkan 0F hexa	26181642

Oleh karena panjang dari SMSC adalah 0, field-field yang lain dapat diabaikan. SMSC yang digunakan adalah SMSC yang terdapat pada SIM card berdasarkan perintah "AT+CSCA"

- Protocol Data Unit (PDU) Type

  Nilai default dari PDU untuk SMS-Deliver adalah 04h. pada contoh di atas, PDU type adalah 24 yang memiliki arti :

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRR	VPF	VPF	RD	MTI	MTI
Nilai	0	0	0	1	0	0	0	1

- Message Reference (MR)

  Pada contoh diatas MR adalah 00.

- Destination Address

  Penulisan format Destination Address sama dengan Originator Address.

Octet	Keterangan	Nilai
Len	Panjang informasi SMSC dalam Octet	05
Type of number	Type of address dari SMSC 81h = lokal format 91h = internasional format	91
BCD Digits	Nomor SMSC. Jika panjangnya ganjil, pada akhir karakter tambahkan 0F hexa	26181642

- Protocol Identifier (PID)

  Pada contoh diatas PID adalah 00.

- Data Coding Scheme (DCS)

  Pada contoh di atas DCS adalah 00.

- Validity Period (VP)

  Pada contoh di atas, VP adalah AA, atau 170d, 170-166 = 4 hari

Tabel 2.5 Tabel Validity Period

Nilai VP	Nilai validitas periode
0 – 143	(VP + 1) * 5 menit (interval 5 menit hingga 12 jam)
144 – 167	12 jam + ((TP-VP-143)*30 menit)
168 – 196	(VP - 160) * 1 hari
197 – 255	(VP - 192) * 1 minggu

- User Data Length (UDL)

  Pada contoh di atas, UDL adalah 09. UDL menunjukkan panjang User Data (UD).

- User Data (UD)

  Pesan "kirim sms" yang dikodekan menjadi 7 bit default alphabet (septet) harus di enkode menjadi 8 bit (octet) untuk SMS-Transfer. Pada Contoh di atas, UD adalah "EBB43CDD06CDDB73". Pada tiap octet jika jumlah bit kurang dari 8, maka diambil bit paling kanan pada septet selanjutnya dan ditambahkan pada bagian kiri septet sebelumnya

2.2 Proses Broadcasting SMS pada PT. JTV (Jawa Timur Televisi)

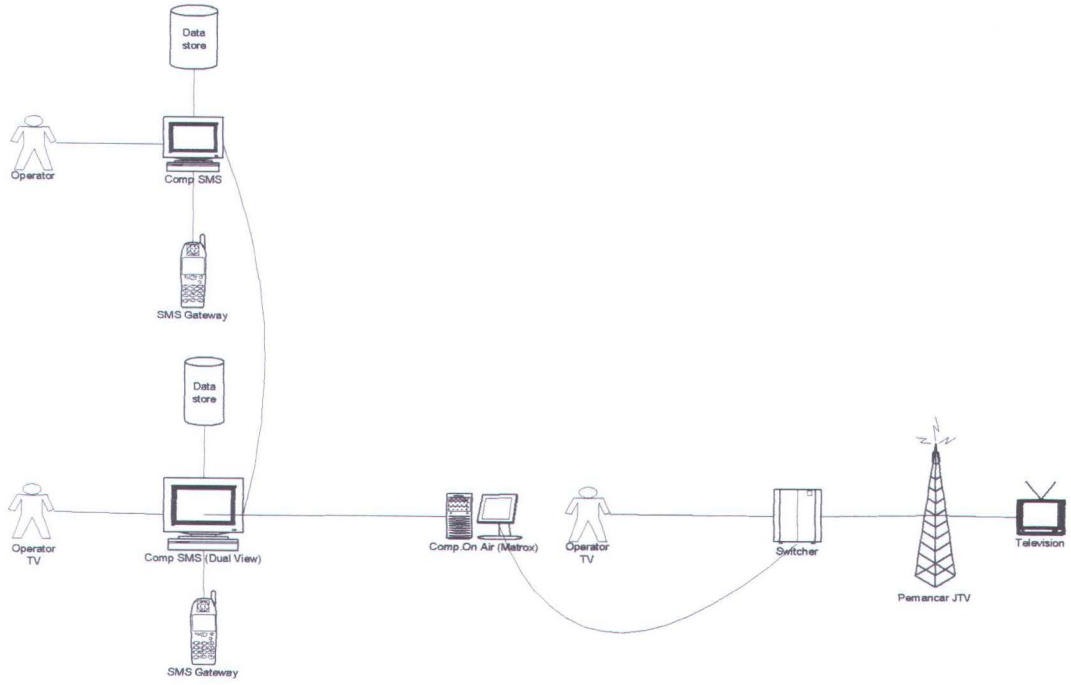
Sub bab ini menjelaskan tentang konsep broadcasting (proses penyiaran) sms, dimana PT. JTV (Jawa Timur Televisi) sebagai studi kasus yang digunakan dalam tugas akhir ini, dan penerapan teknologi broadcast pada PT. JTV (Jawa Timur Televisi).

2.2.1 Konsep Broadcasting SMS pada PT. JTV (Jawa Timur Televisi)

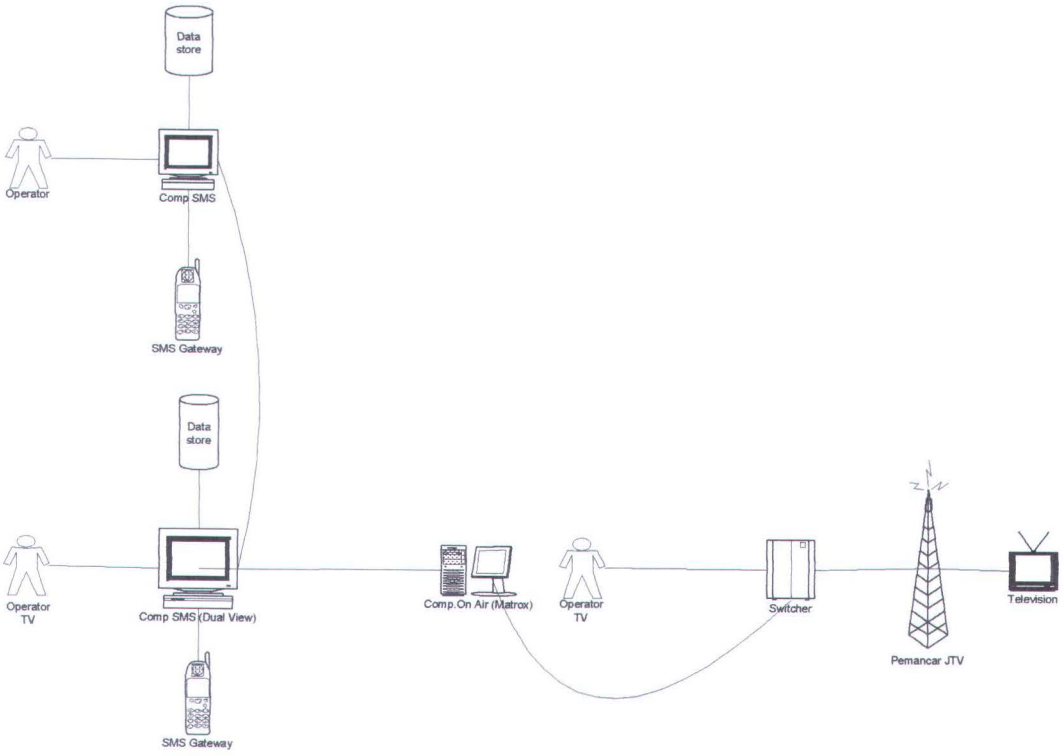
Sistem Broadcasting pada PT. JTV terdiri dari beberapa tools seperti alat pemancar yaitu pemancar televisi yang digunakan untuk memancarkan segala bentuk aktivitas penyiaran pada PT JTV untuk dapat diterima oleh televisi yang ada di setiap rumah. Alat penayang yaitu *Switcher* yang berfungsi untuk menayangkan gambar dan tulisan, dan juga memberikan efek pada gambar. Pada bagian penyiaran SMS terdapat alat pengolah data yang terdiri dari dua perangkat

komputer yaitu komputer SMS (Computer Dual View) dan komputer on air (Matrox).

Studi kasus yang dibuat pada tugas akhir ini adalah menampilkan data-data yang berupa SMS yang berasal dari operator perusahaan pesawat terbang dan juga operator perusahaan kereta api mengenai jadwal-jadwal dari kedua sistem transportasi yang terkait baik itu berupa keberangkatan, kedatangan, keterlambatan, maupun pembatalan yang ditayangkan melalui PT. JTV (Jawa Timur Televisi) dengan adanya operator yang mengendalikan sistem pada perangkat lunak yang berfungsi menerima data sms dan kemudian menayangkannya melalui *Switcher*. Proses kerja yang terjadi pada ketiga belah pihak yaitu operator perusahaan pesawat terbang, operator perusahaan kereta api, dan operator pada PT. JTV sebagai pihak penerima data dari kedua pihak yang bersangkutan dan menayangkannya, akan dijelaskan dalam bentuk gambar dibawah ini, dimana gambar dibawah ini juga merupakan deskripisi bentuk penyiaran (broadcasting) SMS yang terdapat pada PT. JTV (Jawa Timur Televisi)



Gambar 2.3 Skema proses pengiriman data SMS Jadwal Penerbangan



Gambar 2.4 Skema proses pengiriman data SMS Jadwal Kereta Api

Seperti dapat terlihat dengan jelas, bahwa operator dari pihak pesawat maupun dari pihak kereta api mengirimkan jadwal tranportasi yang berupa sms melalui komputer sms yang telah terhubung dengan SMS Gateway, kemudian komputer SMS (dual view) pada JTV yang juga telah terhubung dengan SMS Gateway akan menerima jadwal transportasi tersebut. Operator televisi berperan untuk memfilter (menyaring) jadwal transportasi SMS yang masuk. Apabila jadwal tersebut akan ditayangkan ke televisi, maka operator televisi harus mengirimkan data jadwal transportasi SMS melalui jaringan lokal ke komputer On Air (Matrox). Data jadwal transportasi SMS yang sudah masuk pada komputer On Air (Matrox) secara otomatis akan terhubung ke *Switcher,* sehingga hanya dengan digerakkan pada posisi tayang oleh operator televisi, maka pemancar televisi akan merespon dan kemudian menayangkannnya secara umum.

2.2.2 Penerapan Teknologi Broadcast pada PT. JTV (Jawa Timur Televisi)

Ada empat piranti broadcast yang menunjang terjadinya proses penayangan sms *running text* pada layar televisi, diantaranya yaitu komputer sms (dual view), Komputer on air (Matrox), *Switcher*, dan Pemancar televisi. Pada sub bab ini akan dijelaskan spesifikasi dari piranti broadcast yang digunakan pada PT. JTV (Jawa Timur Televisi).

2.2.2.1 Computer dual view

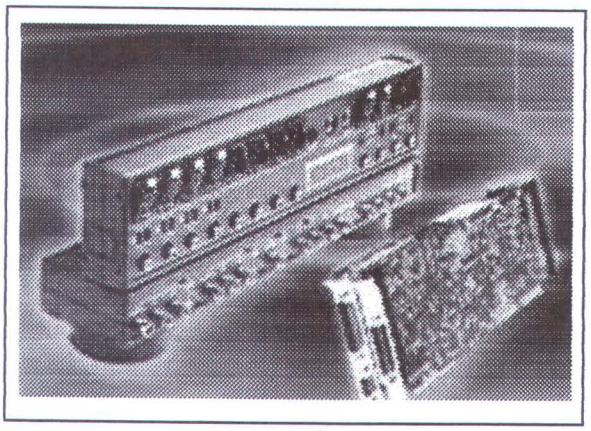
Komputer dual view adalah komputer pengolah pertama dari data yang diterima dari sms gateway. Pada dasarnya computer dual view adalah sebuah computer PC biasa, akan tetapi yang penting dalam computer dual view adalah

terdapatnya vgacard computer yang memiliki fasilitas TVOUT yang nantinya akan menjadi media digital yang diolah di hardware berikutnya yang bernama Matrox Digisuite. Untuk spesifikasi computer dual view yang berada pada JTV adalah PC Pentium IV 2.4GHz, Memori 512MB dengan operating system Microsoft Windows XP Professional Versi 2002 Service Pack 2, dan VGA Card NVIDIA GeForce4 MX 4000

2.2.2.2 Computer on air (Matrox)

Spesifikasi computer on air yang digunakan oleh PT. JTV adalah computer yang terdapat piranti *Matrox Digisuite* didalamnya. Matrox Digisuite adalah hardware video yang umum digunakan untuk kebutuhan broadcast. Dengan matrox digisuite media digital yang dihasilkan dari computer dual view dapat diedit seperti : *nonlinear editing, digital compositing, painting, titling, 3D animation recording,* dan *audio sweetening.* Saat ini hanya hardware inilah dalam dunia broadcasting untuk menambahkan, maupun mengubah media digital yang ada. Proses kerja yang dilakukan oleh Matrox Digisuite adalah menyatukan dua buah frame, yaitu frame gambar asli pada televisi, dengan frame yang hanya berisikan tulisan berjalan / *running text.* Agar dapat menyatu dengan frame gambar asli pada televisi maka frame untuk *running text* harus ditransparasi terlebih dahulu. Untuk membuat sebuah frame yang bersifat transparan, frame tersebut harus berwarna hitam, karena piranti pada matrox digisuite akan menganggap warna hitam sebagai warna transparan. Sehingga apabila ingin membuat tulisan berjalan / *running text*, tulisan tersebut harus berwarna selain hitam, putih misalnya, dan selain tulisan harus berwarna hitam. Matrox Digisuite

akan mengolah frame tersebut untuk disatukan dengan frame gambar asli pada televisi, sehingga yang tampak adalah satu kesatuan antara gambar asli dengan *running text*.



Gambar 2.5 Piranti Perangkat Keras Matrox Digisuite

2.2.2.3 Switcher

Swither adalah hardware broadcast yang terdiri dari 8 input port dan 2 output port. Hardware ini dapat dikonfigurasi sesuai dengan aturan yang telah ditentukan. Kombinasi yang dibutuhkan tiap aturan port dapat menghasilkan output yang berbeda. Dengan swither, operator dapat menentukan hasil olahan media digital yang akan ditayangkan di televisi. Sebagai misal, hasil olahan matrox digisuite yang ditambahkan perbaikan suara dan penambahan mode running text dari computer dual view, dengan switcher akan dapat dipilih hasil olahan media yang akan ditayangkan.

2.3 Konsep Desain Antarmuka

2.3.1 Desain yang Bagus

Ada enam aspek yang harus diperhatikan dalam pembuatan desain yang bagus, yaitu:

1. Clarity

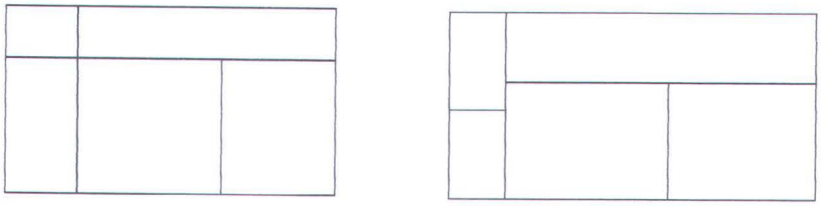
   Mengikuti arah pandangan mata, memberikan waktu untuk mata beristirahat, tidak terlalu banyak hal yang tidak berguna.

2. Consistency

   Harus kosistens dalam *layout*, warna, gambar, ikon, topografi, teks, dan juga corak dasar.

3. Aesthetic

   Menggunakan *layout* yang konsisten, masih terdapat ruang kosong, dan halaman antarmuka dirancang mudah dilihat dan diikuti. Layout yang dapat diikuti seperti terlihat pada gambar 2.6 di bawah. Menggunakan warna sesuai tujuan, tidak asal memberi warna, sebagai contoh warna kuning digunakan untuk peringatan. Warna *background* harus kontras akan tetapi tidak bertentangan(senada) dengan *foreground.*



Gambar 2.6 Desain *layout*

4. Functionality

   Menentukan fungsi-fungsi utama yang akan digunakan pada perangkat lunak sesuai dengan spesifikasi permintaan dan menghindari fungsi fungsi yang tidak terlalu dibutuhkan.

5. Speed dan Efficiency

   Meminimalisasi penekanan tombol mouse, urutan penekanan tombol (*tab order*), dan mempersingkat waktu eksekusi.

6. Usability

   Perangkat lunak harus memenuhi kebutuhan dan kemampuan pengguna dan diusahakan pengguna baru dapat dengan mudah melakukan interaksi dan mencapai hasil yang maksimal.

2.3.2 Pedoman Desain Antarmuka

Pedoman desain antarmuka terdiri dari delapan poin yang biasa disebut dengan Eight Golden Rule yang terdiri dari:

1. Berusaha untuk konsisten

   Konsisten dalam urutan aksi, posisi, dan corak dasar.

2. Menyediakan fasilitas *shortcut*

   Memberikan *shortcut* untuk mempercepat pengaksesan. Sebisa mungkin shortcut yang disediakan mengikuti aturan global semisal ctrl-s untuk menyimpan.

3. Memberikan informasi feedback

Adanya progres bar, sehingga pengguna mengetahui sejauh mana proses yang sedang dilakukan.

4. Menampilkan dialog hasil proses atau peringatan

   Adanya kotak dialog yang memberitahukan hasil proses yang dijalankan atau peringatan.

5. Menyediakan *error handling* yang mudah

   Desain antarmuka harus meminimalisasi kemungkinan terjadinya kesalahan(*error*) yang fatal.

6. Menyediakan fasilitas untuk kembali ke kondisi semula

   Untuk mengantisipasi pengguna melakukan hal yang diluar kontrol disediakan fasiltas untuk kembali ke kondisi semula, misal dengan adanya *undo.*

7. Mendukung *locus of control* internal

   Pengguna yang berpengalaman sangat ingin untuk menguasai sistem dan sistem dapat memberikan respon pada aksi mereka.

8. Mengurangi waktu pengguna berpikir

   Antarmuka harus sesederhana mungkin sehingga pengguna tidak memerlukan waktu yang banyak untuk memilih atau memahami menu.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

BAB III

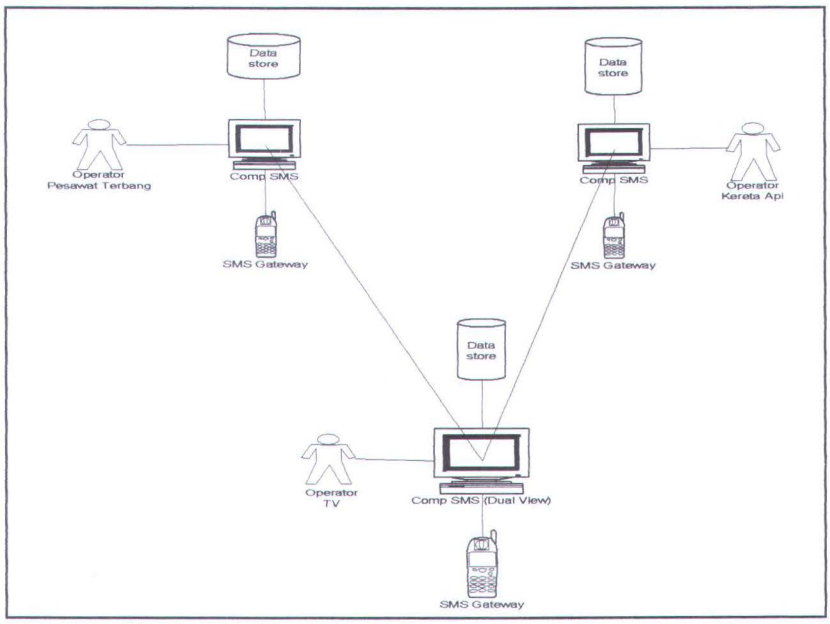
ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang Pemodelan perangkat lunak dari sistem yang akan dibuat, meliputi: deskripsi umum, Pemodelan proses, Pemodelan data dan Pemodelan antarmuka. Pemodelan dilakukan dengan menggunakan metode berorientasi objek.

3.1 Deskripsi Umum Sistem

Perangkat Lunak yang akan dibangun pada Tugas Akhir ini berupa Perangkat lunak Penjadwalan Pesawat Terbang dan Kereta Api yang mengalami keterlambatan dan juga pembatalan jalan, dimana data jadwal tersebut akan dikirimkan kepada stasiun televisi lokal yaitu PT. JTV (Jawa Timur Televisi) melalui SMS (Short Messaging Service) untuk ditayangkan dalam bentuk *running text* secara umum. Tujuan dari dibangunnya perangkat lunak ini adalah supaya masyarakat bisa mendapatkan informasi terbaru mengenai jadwal Pesawat Terbang dan Kereta Api yang mengalami keterlambatan atau pembatalan jalan, yang dapat dilihat di layar televisi karena ditayangkan melalui stasiun televisi lokal yaitu PT. JTV (Jawa Timur Televisi). Ini akan sangat memudahkan masyarakat dalam mendapatkan informasi jadwal transportasi terbaru tanpa harus keluar dari rumah. Ada tiga perangkat lunak yang dibangun untuk melengkapi tiga sektor yang mendukung penjadwalan transportasi. Dimana dari tiga sektor tersebut dibedakan menjadi 2 (dua) berdasarkan kinerjanya yaitu pengiriman dan penerimaan jadwal transportasi. Terdapat 2 (dua) sektor yang menunjang proses

pengiriman jadwal transportasi yaitu sektor perusahaan pesawat terbang dan sektor perusahaan kereta api. Dua perangkat lunak telah dibangun dan dipasang pada tiap sektor untuk menunjang sektor perusahaan pesawat terbang dan sektor perusahaan kereta api dalam proses pengiriman jadwal transportasi dalam bentuk SMS. Proses pengiriman jadwal dilakukan oleh operator dari pihak perusahaan penerbangan atau operator dari pihak kereta api yang memasukkan data jadwal melalui perangkat lunak, terhadap pesawat terbang atau kereta api yang mengalami keterlambatan atau pembatalan jalan. Satu sektor terakhir adalah sektor stasiun televisi lokal yaitu PT. JTV (Jawa Timur Televisi) yang menunjang proses penerimaan jadwal transportasi yang berupa SMS tersebut. Terdapat perangkat lunak yang telah dibangun dan dipasang pada sektor ini dan dioperasikan oleh satu operator yang berasal dari pihak stasiun televisi.



Gambar 3.1. Skema Proses Pengiriman dan Penerimaan Data Jadwal Transportasi

Gambar di atas merupakan skema proses pengiriman dan penerimaan data jadwal transportasi. Pada gambar di atas operator dari pihak pesawat terbang

maupun pihak kereta api memasukkan data jadwal transportasi dan secara otomatis jadwal transportasi tersebut akan tersimpan di dalam basis data yang ada pada masing-masing komputer sms lokal. Untuk mengirim data jadwal transportasi yang berupa sms tersebut maka setiap komputer sms harus terhubung dengan SMS gateway. Setelah sms jadwal transportasi dikirim maka komputer sms (komputer dual view) pada PT. JTV yang juga telah tersambung dengan SMS gateway akan menerima jadwal transportasi sms tersebut dan secara otomatis akan disimpan dalam basis data lokal yang ada pada komputer sms PT. JTV. Terdapat operator yang berperan dalam pengoperasian perangkat lunak dan manajemen sms yang masuk.

3.1.1 Fungsi-fungsi pada perangkat lunak

Adapun fungsi-fungsi dasar utama dari sistem ini antara lain :

- Penambahan, penghapusan, pengubahan data pesawat yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data airport yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data rute perjalanan / jurusan pesawat yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data jadwal penerbangan berdasarkan status yang telah diisi yang kemudian disimpan pada database lokal.
- Pemilihan nomor telepon tujuan sms data jadwal penerbangan.
- Pengiriman data jadwal penerbangan kepada nomor telepon tujuan yang telah dipilih dalam bentuk sms.

- Penampilan sms data jadwal penerbangan yang telah dikirim yang dilengkapi dengan status dan informasi status berupa *message sent failed, sent, delivery failed, delivered.*
- Penyambungan dan pemberhentian koneksi SMS Gateway pada komputer sms pesawat terbang.
- Penyimpanan setting koneksi yang telah dibuat pada database lokal.
- Penambahan, penghapusan, pengubahan data kereta yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data stasiun yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data rute perjalanan / jurusan kereta api yang kemudian disimpan pada database lokal.
- Penambahan, penghapusan, pengubahan data jadwal kereta api berdasarkan status yang telah diisi yang kemudian disimpan pada database lokal.
- Pemilihan nomor telepon tujuan sms data jadwal kereta api.
- Pengiriman data jadwal kereta kepada nomor telepon tujuan yang telah dipilih dalam bentuk sms.
- Penampilan sms data jadwal kereta yang telah dikirim yang dilengkapi dengan status dan informasi status berupa *message sent failed, sent, delivery failed, delivered.*
- Penyambungan dan pemberhentian koneksi SMS Gateway pada komputer sms kereta api.
- Penyimpanan setting koneksi yang telah dibuat pada database lokal.

- Penerimaan sms data jadwal transportasi yang telah dikirim oleh perusahaan penerbangan dan kereta api.
- Penayangan sms yang masuk pada layar televisi dalam bentuk *running text* dimana terdapat tiga proses di dalamnya yaitu jalankan text sms (*run*), berhentikan text sms (*stop*), sembunyi atau munculkan text sms (*hide/show*).
- Penyambungan dan pemberhentian koneksi SMS Gateway pada komputer sms JTV.
- Penambahan setting koneksi.
- Penyimpanan setting koneksi yang telah dibuat pada database lokal.
- Penyaringan sms yang masuk, menghindari masuknya penulisan sms yang tidak diinginkan.

3.1.2 Karakteristik Pengguna

Pada perangkat lunak ini, *user* dibagi menjadi tiga yaitu :

3.1.2.1 Operator pesawat

Operator pesawat adalah User yang berada di perusahaan penerbangan yang bertugas memberikan informasi data jadwal penerbangan. Spesifikasi tugas yang diberikan pada operator pesawat adalah sebagai berikut :

- Mengaktifkan koneksi SMS Gateway.
- Memasukkan data pesawat, merubah data pesawat, menghapus data pesawat.
- Memasukkan data airport, merubah data airport, menghapus data airport.
- Memasukkan data jurusan pesawat, merubah data jurusan pesawat, menghapus data jurusan pesawat.

- Memasukkan data jadwal penerbangan, merubah data jadwal penerbangan, menghapus data jadwal penerbangan.
- Mengirimkan data jadwal penerbangan dalam bentuk sms.

3.1.2.2 Operator kereta api

Operator kereta adalah User yang berada di perusahaan kereta api yang bertugas memberikan informasi data jadwal kereta api. Spesifikasi tugas yang diberikan pada operator kereta api adalah sebagai berikut :

- Mengaktifkan koneksi SMS Gateway.
- Memasukkan data kereta, merubah data kereta, menghapus data kereta.
- Memasukkan data stasiun, merubah data stasiun, menghapus data stasiun.
- Memasukkan data jurusan kereta, merubah data jurusan kereta, menghapus data jurusan kereta.
- Memasukkan data jadwal kereta, merubah data jadwal kereta, menghapus data jadwal kereta.
- Mengirimkan data jadwal kereta dalam bentuk sms.

3.1.2.3 Operator JTV

Operator JTV adalah User yang berada di perusahaan televisi dalam hal ini adalah PT. JTV (Jawa Timur Televisi) sebagai perusahaan televisi yang bersangkutan. Spesifikasi tugas yang diberikan pada operator televisi adalah sebagai berikut :

- Mengaktifkan koneksi SMS Gateway.
- Menambahkan koneksi SMS Gateway apabila diperlukan.

- Menghapus koneksi SMS gateway yang sudah tidak diperlukan lagi.
- Membuat batasan-batasan terhadap sms yang masuk supaya tidak terjadi hal-hal yang tidak diinginkan (*filterisasi sms* yang masuk).
- Menjalankan sms yang masuk melalui perangkat lunak yang sudah dibangun dan kemudian ditayangkan ke layar televisi.

3.2 Pemodelan Proses

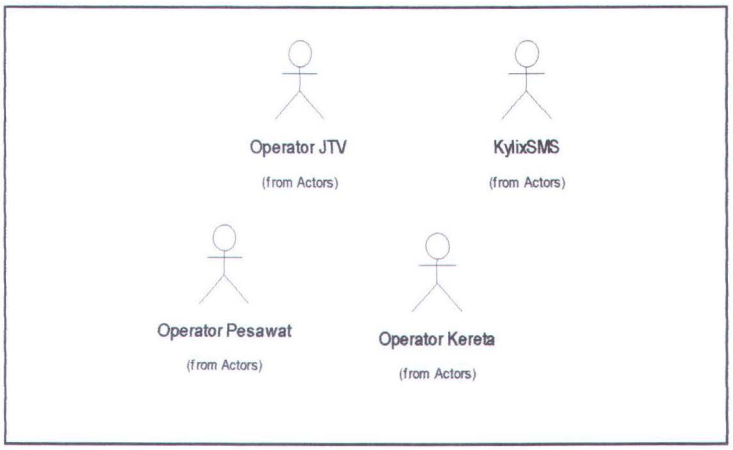
Pemodelan proses perangkat lunak ini dilakukan dengan menggunakan notasi UML (*Unified Modelling Language*) yang dimiliki oleh perangkat lunak Rational Rose Enterprise Edition 2002. Rancangan perangkat lunak tersebut berupa diagram *usecase*.

3.2.1 Analisa Use Case View

Tahap ini mempunyai sebuah tujuan, yaitu menggambarkan tingkah laku sistem dengan fungsi-fungsi yang akan disediakan olehnya. Langkah pertama adalah mengidentifikasi aktor yang akan terlibat, dan yang kedua adalah mendefinisikan *usecase* untuk tiap-tiap aktor.

3.2.1.1 Aktor

Aktor tidak termasuk bagian dari sistem. Aktor merepresentasikan orang atau sesuatu obyek yang berinteraksi dengan sistem. Aktor dapat hanya menerima informasi, memberi informasi atau menerima dan memberi informasi kepada sistem [QUA98].



Gambar 3.2 Aktor-aktor Perangkat Lunak PPK

Sebagaimana dapat dilihat pada gambar 3.2, ada 4 aktor yang terlibat dalam perangkat lunak pada tugas akhir ini, yaitu Operator kereta, Operator JTV, Operator penerbangan, Operator KylixSMS . Tabel 3.1 berikut akan menjelaskan definisi dari masing masing aktor. User dan Administrator pada dasarnya adalah pengguna, yang membedakan adalah hak untuk mengakses menu.

Tabel 3.1 Definisi dan Tugas Aktor

Nama Aktor	Definisi	Hak Akses
Operator Pesawat	Pengguna perangkat lunak yang merupakan pegawai perusahaan penerbangan yang mengentrikan data keadaan pesawat.	Semua jenis akses perangkat lunak
Operator Kereta	Pengguna perangkat lunak yang merupakan pegawai perusahaan kereta api yang mengentrikan data keadaan kereta.	Semua jenis akses perangkat lunak
Operator JTV	Operator yang mengoperasikan manajemen sms di TV.	Semua jenis akses perangkat lunak
KylixSMS	Komponen software yang berfungsi untuk mengirim dan menerima sms.	-

ITS

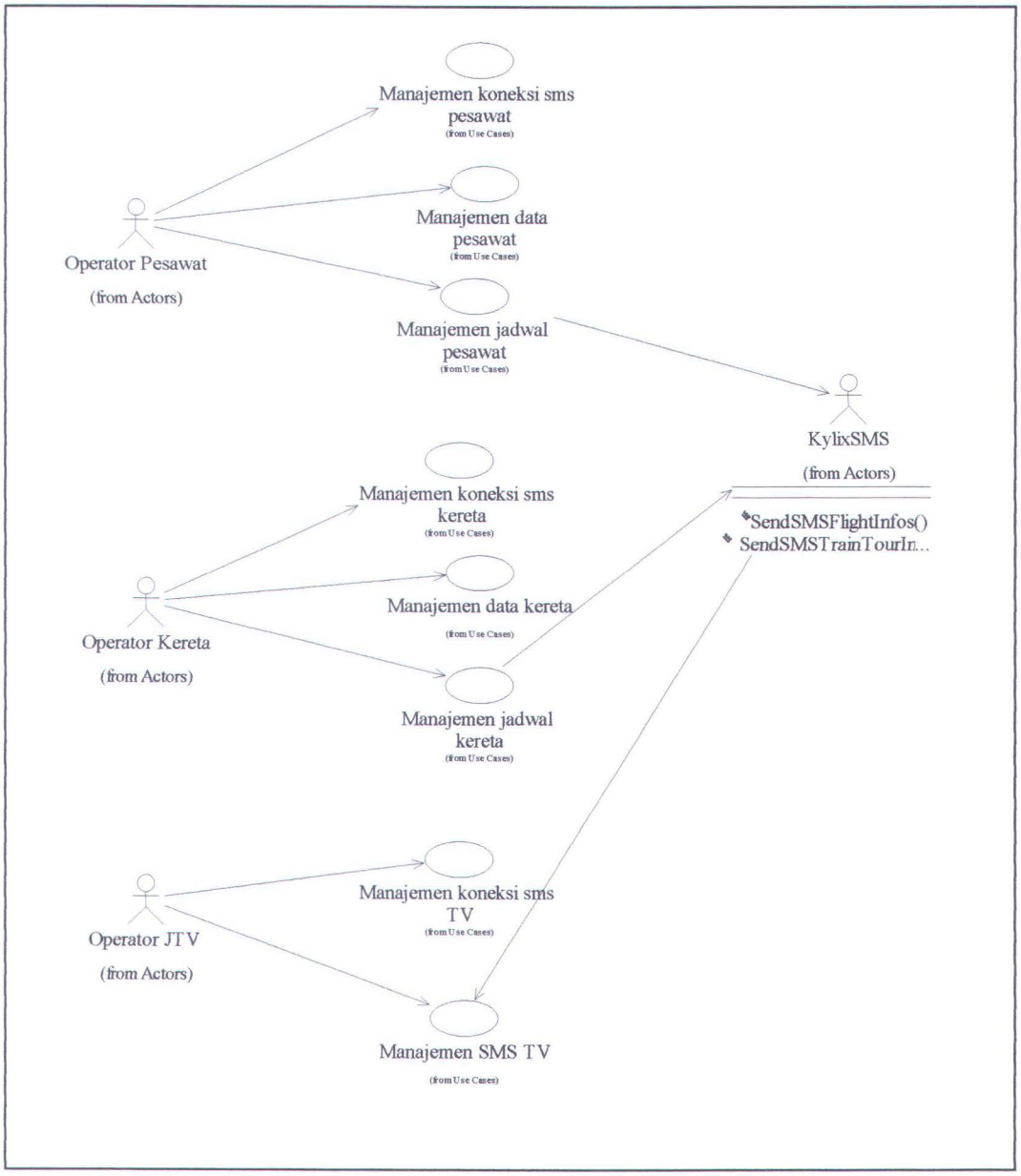
3.2.1.2 Use Case

Model *usecase* adalah dialog antara seorang aktor dengan sistem. Sebuah *usecase* adalah urutan transaksi-transaksi yang dilakukan oleh sebuah sistem yang menyediakan hasil terukur untuk sebagian aktor [QUA98]. Usecase yang terdapat dalam perangkat lunak Pemanfaatan sms untuk menyampaikan informasi pembatalan dan keterlambatan jadwal Pesawat Terbang dan Kereta Api melalui saluran televisi dijelaskan pada table 3.2 berikut.

Tabel 3.2 Daftar Use Case

No	Use Case	Penjelasan
1	Manajemen data pesawat	Menambah, mengubah, menghapus data pesawat, data bandara, dan data rute perjalanan pesawat.
2	Manajemen jadwal pesawat	Menambah jadwal, mengedit jadwal, menghapus jadwal, dan mengirim jadwal pesawat yang mengalami keterlambatan dan pembatalan jalan dalam bentuk sms kepada aplikasi penerima SMS yang terdapat di JTV (Running).
3	Manajemen koneksi sms pesawat	Menyambungkan koneksi, memberhentikan koneksi, dari komputer ke handphone sebagai jembatan untuk proses sms (sms gateway) dan sekaligus bisa menyimpan setting koneksi sms yang telah dibuat.
4	Manajemen data kereta	Menambah, mengubah, menghapus data kereta, data stasiun, dan data rute perjalanan kereta.
5	Manajemen jadwal kereta	Menambah jadwal, mengedit jadwal, menghapus jadwal, dan mengirim jadwal kereta yang mengalami keterlambatan dan pembatalan jalan dalam bentuk sms kepada aplikasi penerima SMS yang terdapat di JTV (Running).
6	Manajemen koneksi sms kereta	Menyambungkan koneksi, memberhentikan koneksi, dari komputer ke handphone sebagai jembatan untuk proses sms (sms gateway) dan sekaligus bisa menyimpan setting koneksi sms yang telah dibuat.
7	Manajemen sms TV	Menerima SMS, menjalankan, menghentikan, menyembunyikan, menampilkan, dan menghapus SMS pada aplikasi penerima sms yang terdapat di JTV (Running).
8	Manajemen koneksi sms TV	Menyambungkan koneksi, memberhentikan koneksi, menambah koneksi, menghapus koneksi yang terjadi dari komputer ke handphone sebagai jembatan untuk proses sms (sms gateway) dan sekaligus bisa menyimpan setting koneksi sms yang telah dibuat.

Usecases yang terlibat dalam sistem dapat digambarkan pada *Rational Rose* seperti pada gambar 3.3 berikut. Dan untuk penjelasan lebih terperinci dari setiap *usecase* dibahas pada subbab 3.2.2 pada subbab desain *logical view*.



Gambar 3.3 Use Case Diagram Perangkat Lunak PPK

3.2.2 Desain Logical View

Setelah membuat *usecase view*, langkah selanjutnya adalah membuat *logical view*. Pada tahap ini akan dibuat *activity diagram* untuk menggambarkan aktifitas-aktifitas yang ada dalam menyusun aplikasi.

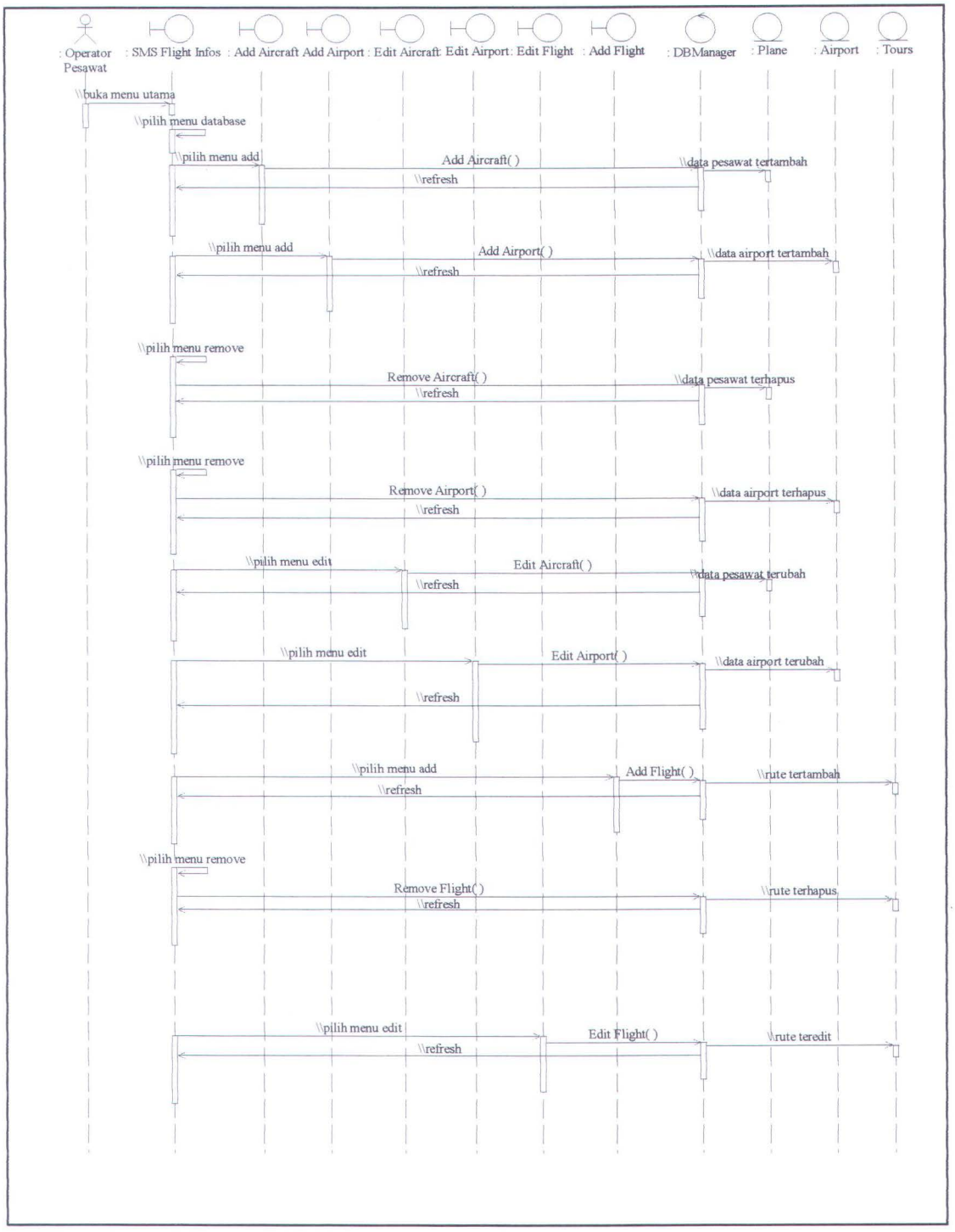
3.2.2.1 Sequence Diagram

Pada pembuatan *logical view*, setelah pembuatan *usecase*, selanjutnya membuat *sequence diagram*. *Sequence diagram* menggambarkan urutan proses antar *class* dalam melakukan suatu *usecase*. *Sequence diagram* melibatkan aktor dan *classes* yang terlibat pada *usecase*, seperti *class boundary*, *entity* dan *class control*. Satu *usecase* minimal memiliki satu sequence diagram. Penjelasan sequence diagram untuk masing-masing *usecase* sebagai berikut.

3.2.2.1.1 Sequence Diagram Manajemen Data Pesawat

Pada perangkat lunak ini Operator pesawat dapat mengisikan data dari pesawat yang berupa id dan nama pesawat. Id merupakan keterangan identitas penggolongan pesawat, misal pesawat Garuda Indonesia dengan idnya adalah GIA. Sedangkan Garuda Indonesia merupakan nama pesawat. Operator pesawat juga dapat mengisikan data airport yang berupa id airport, nama airport, kota. Id merupakan keterangan identitas dari airport yang bersangkutan, yang kemudian untuk memperjelas isi dari data airport tersebut maka akan ditunjang dengan keterangan nama airport dan kota tempat dimana airport tersebut berada. Setelah mengisikan data dari pesawat dan data airport, maka operator pesawat diharuskan untuk mengisikan data rute perjalanan atau jurusan yang ditempuh oleh pesawat

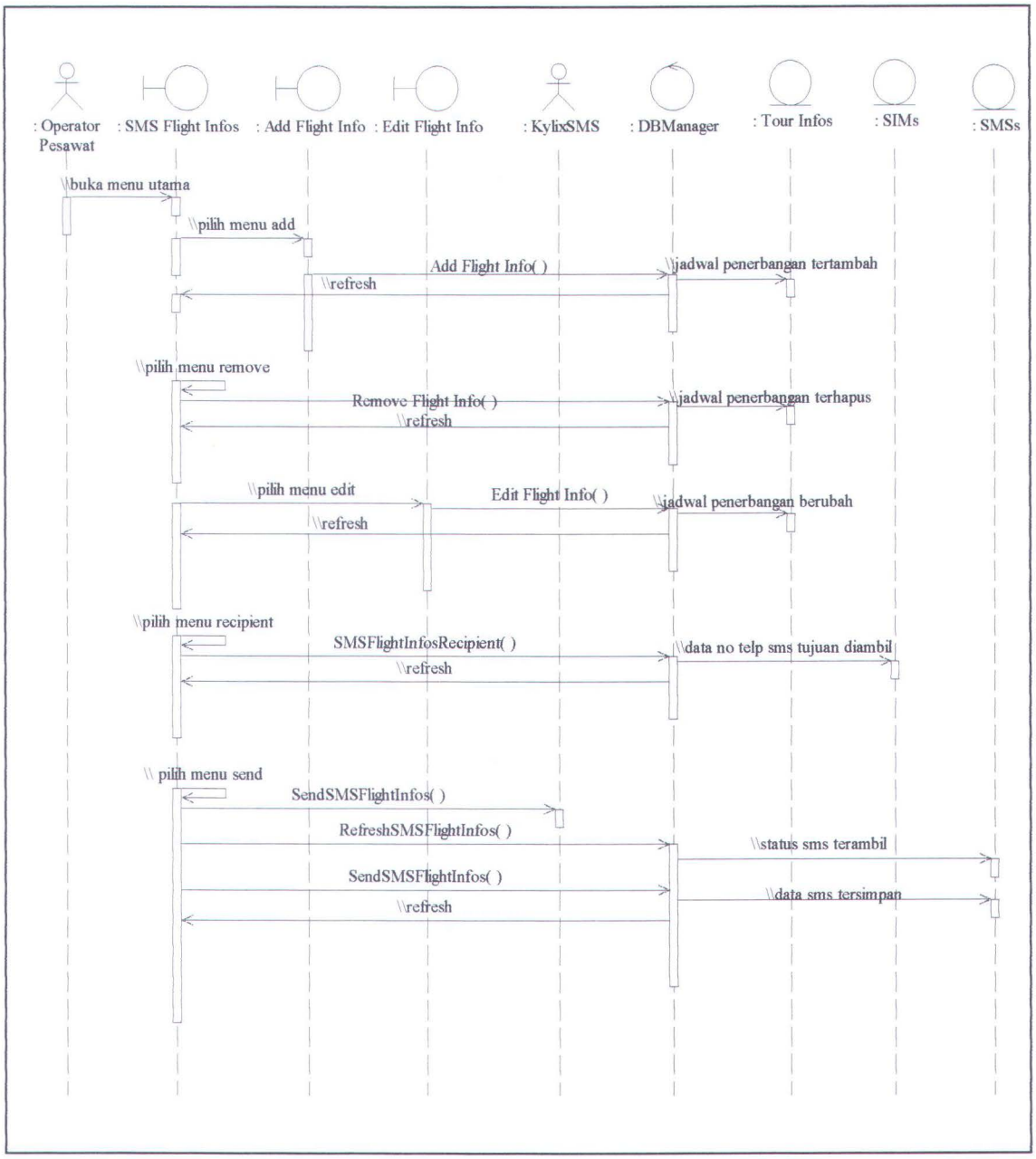
yang bersangkutan, data yang harus diisi oleh operator pesawat adalah Tour ID yang merupakan identitas dari rute perjalanan, kemudian operator dapat memilih data pesawat dan data airport yang telah dimasukkan terlebih dahulu dan telah disimpan pada database, sesuai dengan yang akan diinformasikan kepada masyarakat. Proses memasukkan/menambah, mengubah, dan menghapus data pesawat, data airport, dan data rute perjalanan atau jurusan yang akan ditempuh oleh pesawat, selengkapnya dapat dilihat pada gambar 3.4 berikut.



Gambar 3.4 Sequence Diagram Manajemen Data Pesawat

3.2.2.1.2 Sequence Diagram Manajemen Jadwal Pesawat

Data dari Operator Pesawat mengenai jadwal Penerbangan yang mengalami status yang telah diisi oleh operator pesawat yang berupa keterlambatan ataupun status-status yang lain, dan nantinya informasi dari jadwal penerbangan tersebut dapat dikirimkan kepada server stasiun televisi. Selain mengisikan status, operator pesawat juga dapat menambahkan informasi yang berhubungan dengan status tersebut. Proses yang terjadi dalam manajemen jadwal pesawat ini adalah proses penambahan, pengeditan, penghapusan data jadwal penerbangan, pemilihan nomor telp untuk sms tujuan, pengiriman data jadwal penerbangan dalam bentuk sms, penampilan data-data sms jadwal penerbangan yang telah terkirim dan dilengkapi dengan status dari sms tersebut yang berupa *sent failed, sent, delivery failed, delivered.* Untuk pengisian status yang terjadi dalam jadwal dan informasinya, dilakukan dalam proses penambahan. Proses menambah, mengedit dan menghapus Jadwal dapat dilihat dalam gambar 3.5 berikut.

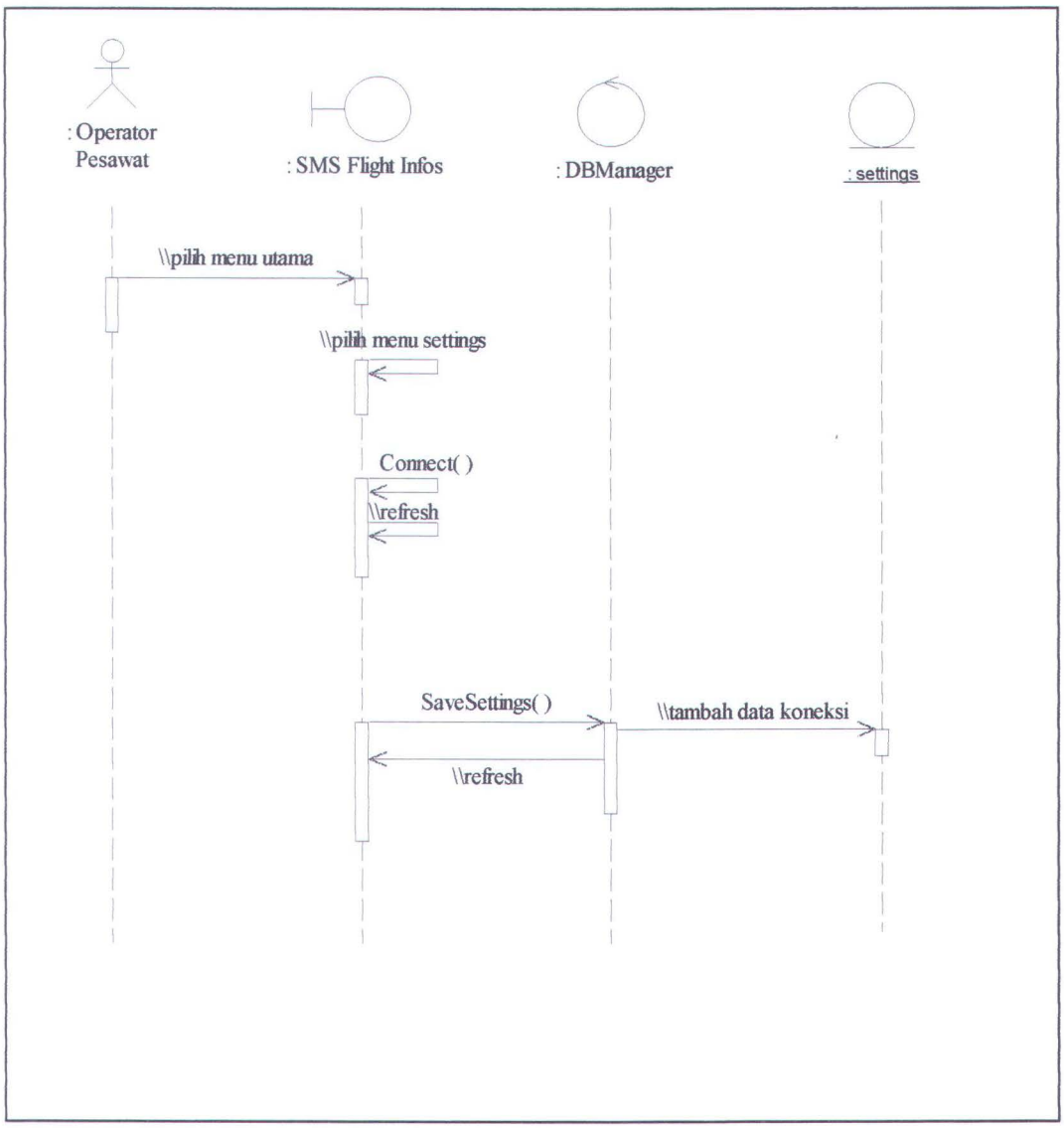


Gambar 3.5 Sequence Diagram Manajemen Jadwal Pesawat

3.2.2.1.3 Sequence Diagram Manajemen koneksi sms Pesawat

Proses yang terjadi pada perangkat lunak ini adalah operator pesawat yang menyambungkan koneksi yang terjadi antara komputer dan handphone agar dapat digunakan untuk proses sms. Proses koneksi tersebut lebih dikenal dengan sebutan sms gateway. Operator pesawat dapat mengkoneksikan dan juga memberhentikan

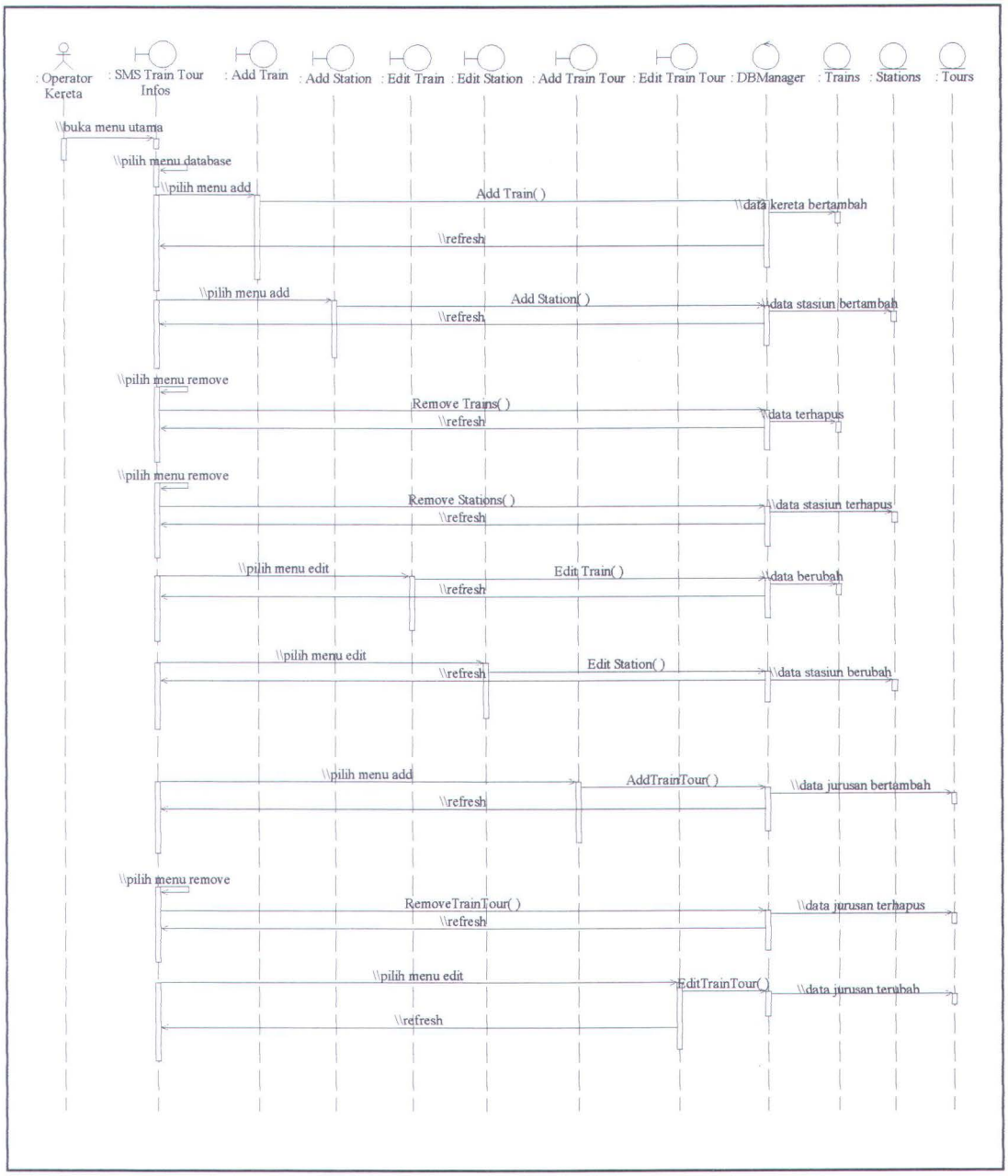
koneksi yang terjadi. Dan kemudian setting koneksi yang telah dibuat datanya dapat di simpan ke dalam sebuah file yang diberi nama *settings.ini* dan dilakukan operasi file di dalamnya. Proses selengkapnya dapat dilihat pada gambar 3.6 berikut.



Gambar 3.6 Sequence Diagram Manajemen koneksi sms Pesawat

3.2.2.1.4 Sequence Diagram Manajemen Data Kereta

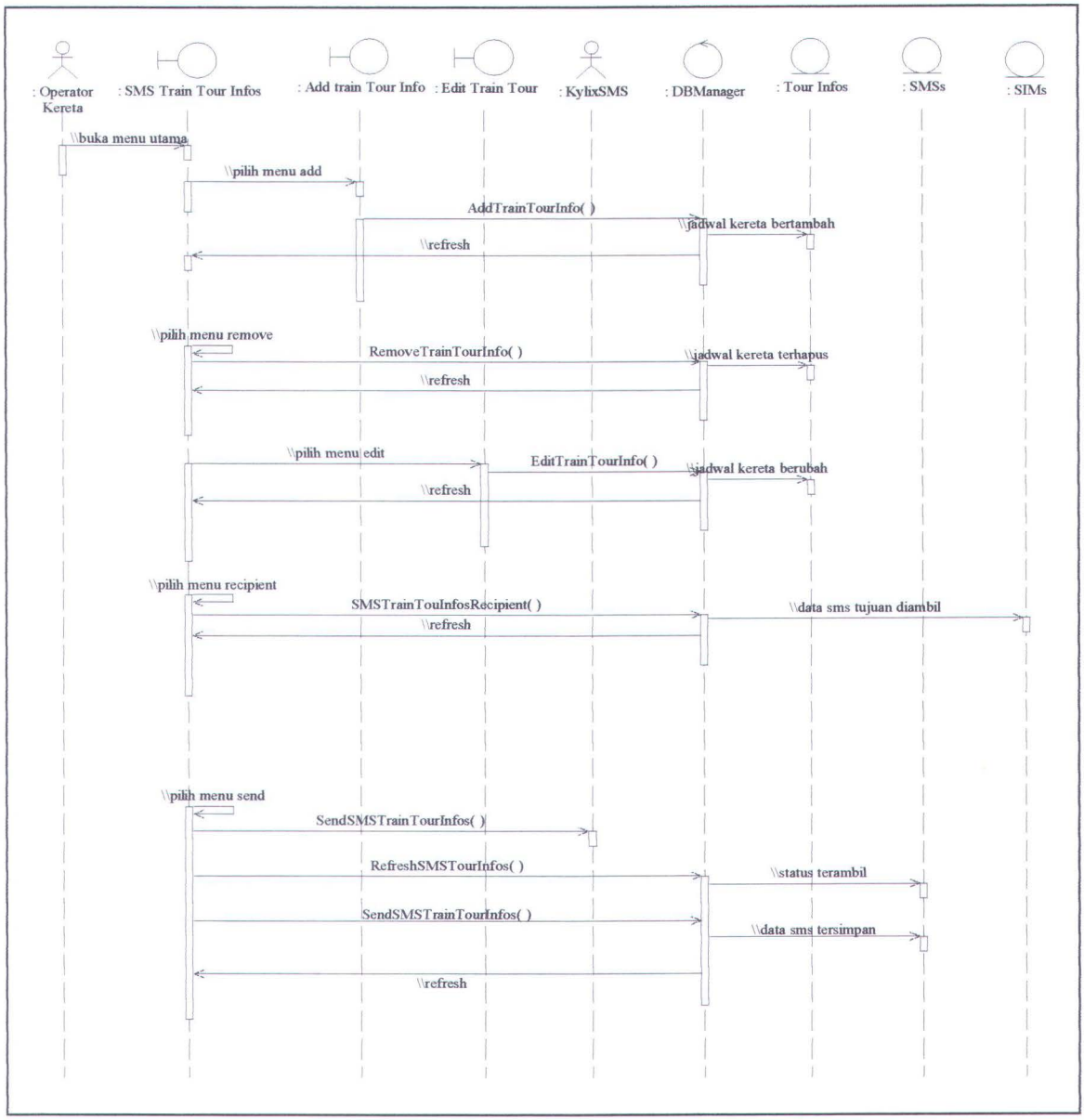
Pada perangkat lunak ini Operator kereta dapat mengisikan data dari kereta yang berupa id dan nama kereta. Id merupakan keterangan identitas penggolongan kereta, misal kereta Argo Bromo dengan idnya adalah AB. Sedangkan Argo Bromo merupakan nama kereta. Operator kereta juga dapat mengisikan data stasiun yang berupa id station, nama station, kota. Id merupakan keterangan identitas dari stasiun yang bersangkutan, yang kemudian untuk memperjelas isi dari data stasiun tersebut maka akan ditunjang dengan keterangan nama stasiun dan kota tempat dimana stasiun tersebut berada. Setelah mengisikan data dari kereta dan data stasiun, maka operator kereta diharuskan untuk mengisikan data rute perjalanan atau jurusan yang ditempuh oleh kereta yang bersangkutan, data yang harus diisi oleh operator kereta adalah Tour ID yang merupakan identitas dari rute perjalanan, kemudian operator dapat memilih data kereta dan data stasiun yang telah dimasukkan terlebih dahulu dan telah disimpan pada database, sesuai dengan yang akan diinformasikan kepada masyarakat. Proses memasukkan/menambah, mengubah, dan menghapus data kereta, data stasiun, dan data rute perjalanan atau jurusan yang akan ditempuh oleh kereta, selengkapnya dapat dilihat pada gambar 3.7 berikut.



Gambar 3.7 Sequence Diagram Manajemen Data Kereta

3.2.2.1.5 Sequence Diagram Manajemen Jadwal Kereta

Data dari Operator Kereta mengenai jadwal kereta yang mengalami status yang telah diisi oleh operator kereta yang berupa keterlambatan ataupun status-status yang lain, dan nantinya informasi dari jadwal kereta tersebut dapat dikirimkan kepada server stasiun televisi. Selain mengisikan status, operator kereta juga dapat menambahkan informasi yang berhubungan dengan status tersebut. Proses yang terjadi dalam manajemen jadwal kereta ini adalah proses penambahan, pengeditan, penghapusan data jadwal kereta, pemilihan nomor telp untuk sms tujuan, pengiriman data jadwal kereta dalam bentuk sms, penampilan data-data sms jadwal kereta yang telah terkirim dan dilengkapi dengan status dari sms tersebut yang berupa *sent failed, sent, delivery failed, delivered.* Untuk pengisian status yang terjadi dalam jadwal dan informasinya, dilakukan dalam proses penambahan. Proses menambah, mengedit dan menghapus Jadwal dapat dilihat dalam gambar 3.8 berikut.

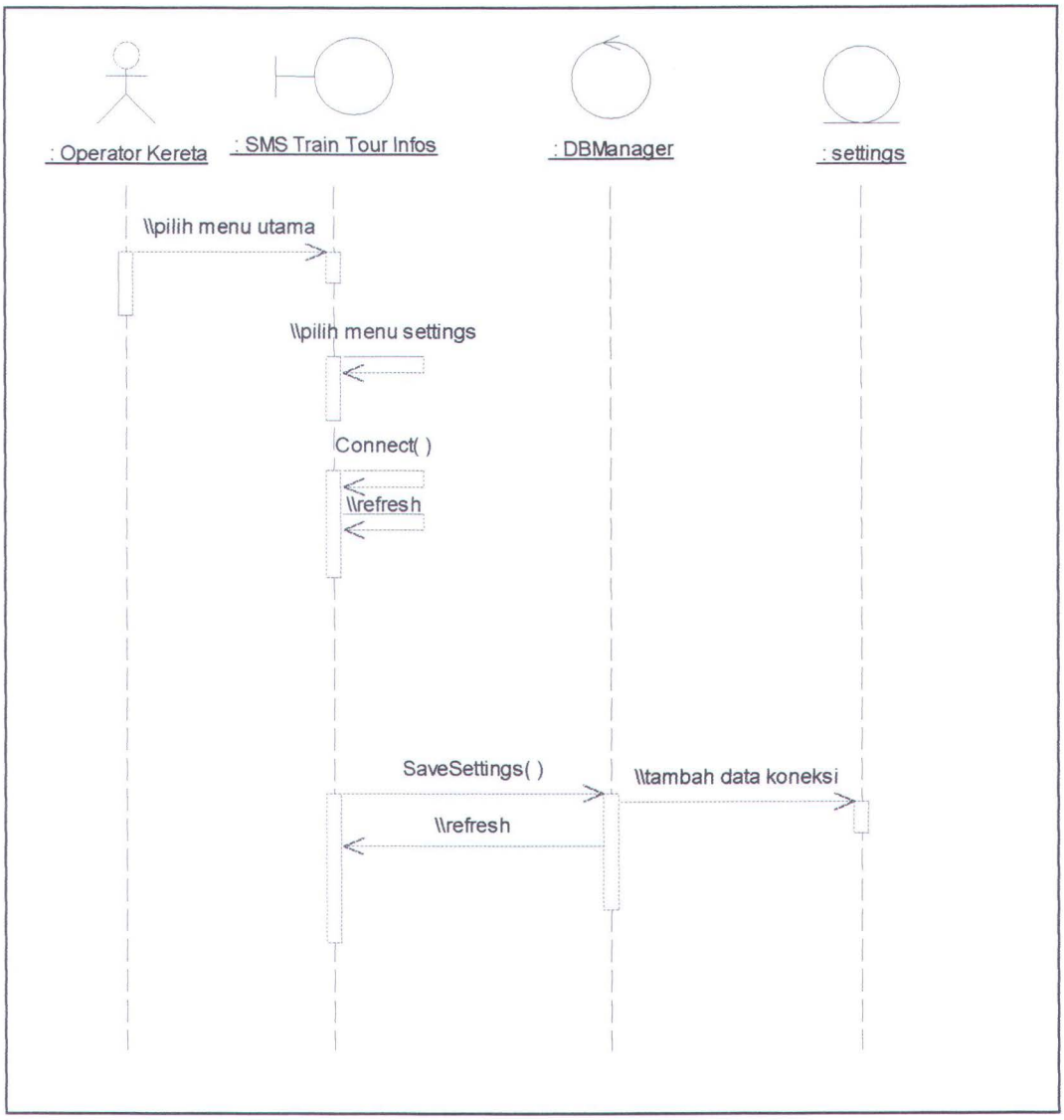


Gambar 3.8 Sequence Diagram Manajemen Jadwal Kereta

3.2.2.1.6 Sequence Diagram Manajemen koneksi sms Kereta

Proses yang terjadi pada perangkat lunak ini adalah operator kereta yang menyambungkan koneksi yang terjadi antara komputer dan handphone agar dapat digunakan untuk proses sms. Proses koneksi tersebut lebih dikenal dengan sebutan sms gateway. Operator kereta dapat mengkoneksikan dan juga memberhentikan

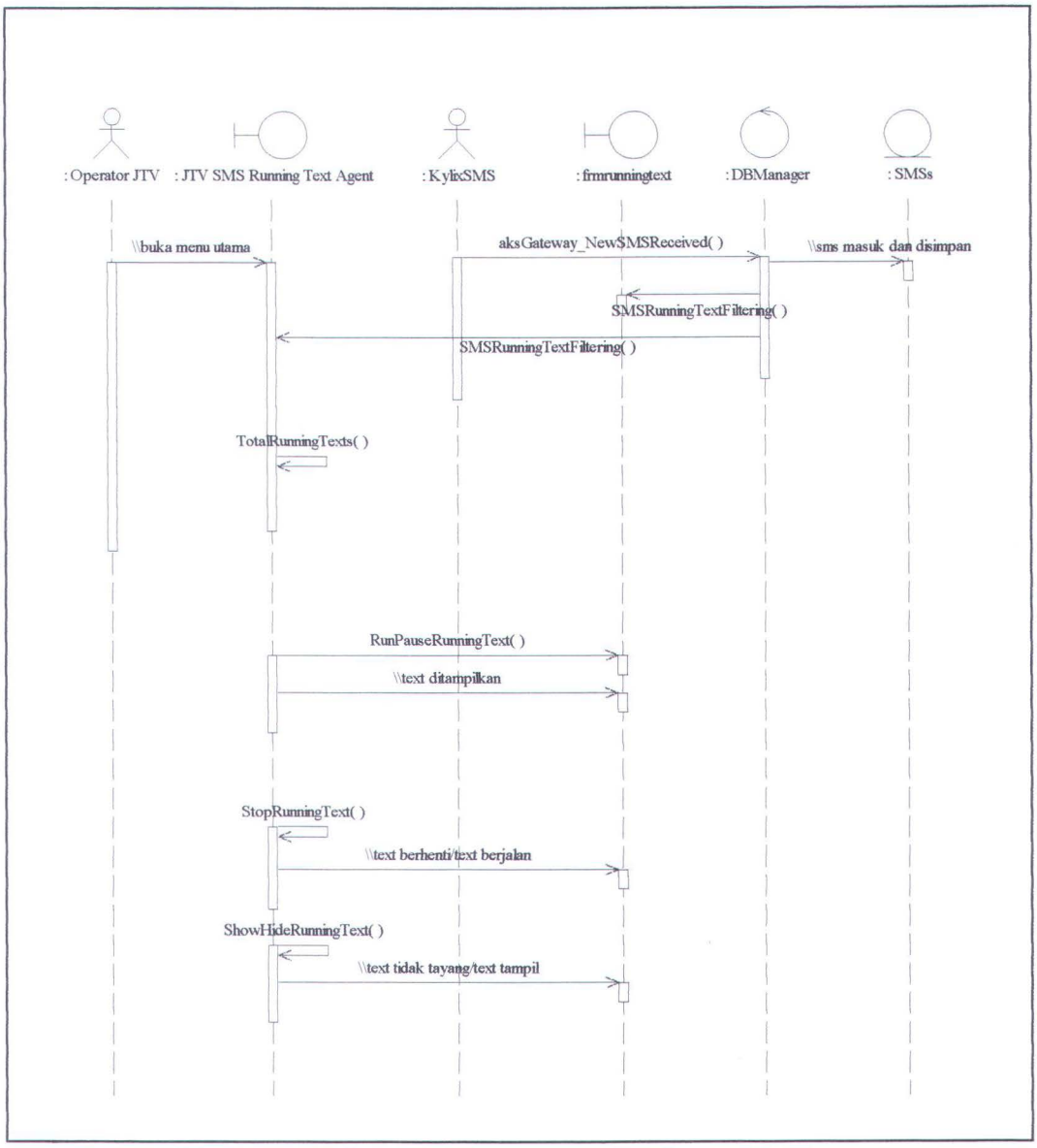
koneksi yang terjadi. Dan kemudian setting koneksi yang telah dibuat, datanya dapat di simpan ke dalam sebuah file yang bernama *settings.ini*, dan akan dilakukan operasi file di dalamnya. Proses selengkapnya dapat dilihat pada gambar 3.9 berikut.



Gambar 3.9 Sequence Diagram Manajemen koneksi sms Kereta

3.2.2.1.7 Sequence Diagram Manajemen SMS TV

Perangkat lunak ini adalah sebagai penerima sms yang berasal dari Pesawat dan Kereta, yang disebut sebagai server stasiun Televisi. Data yang diterima oleh perangkat lunak ini adalah berupa sms jadwal penerbangan dan kereta yang telah mengalami status perubahan dan ditujukan sebagai informasi bagi masyarakat. Data sms yang masuk oleh operator JTV akan ditayangkan di televisi melalui perangkat lunak ini dan data tersebut berupa *Running text*. Proses kerja yang dilakukan oleh perangkat lunak ini adalah menerima data sms, menampilkan data sms ke televisi (*Running Text*), menghentikan dan menjalankan kembali data sms pada televisi, menyembunyikan dan menampilkan kembali data sms di layar televisi. Selengkapnya dapat dilihat pada gambar 3.10.



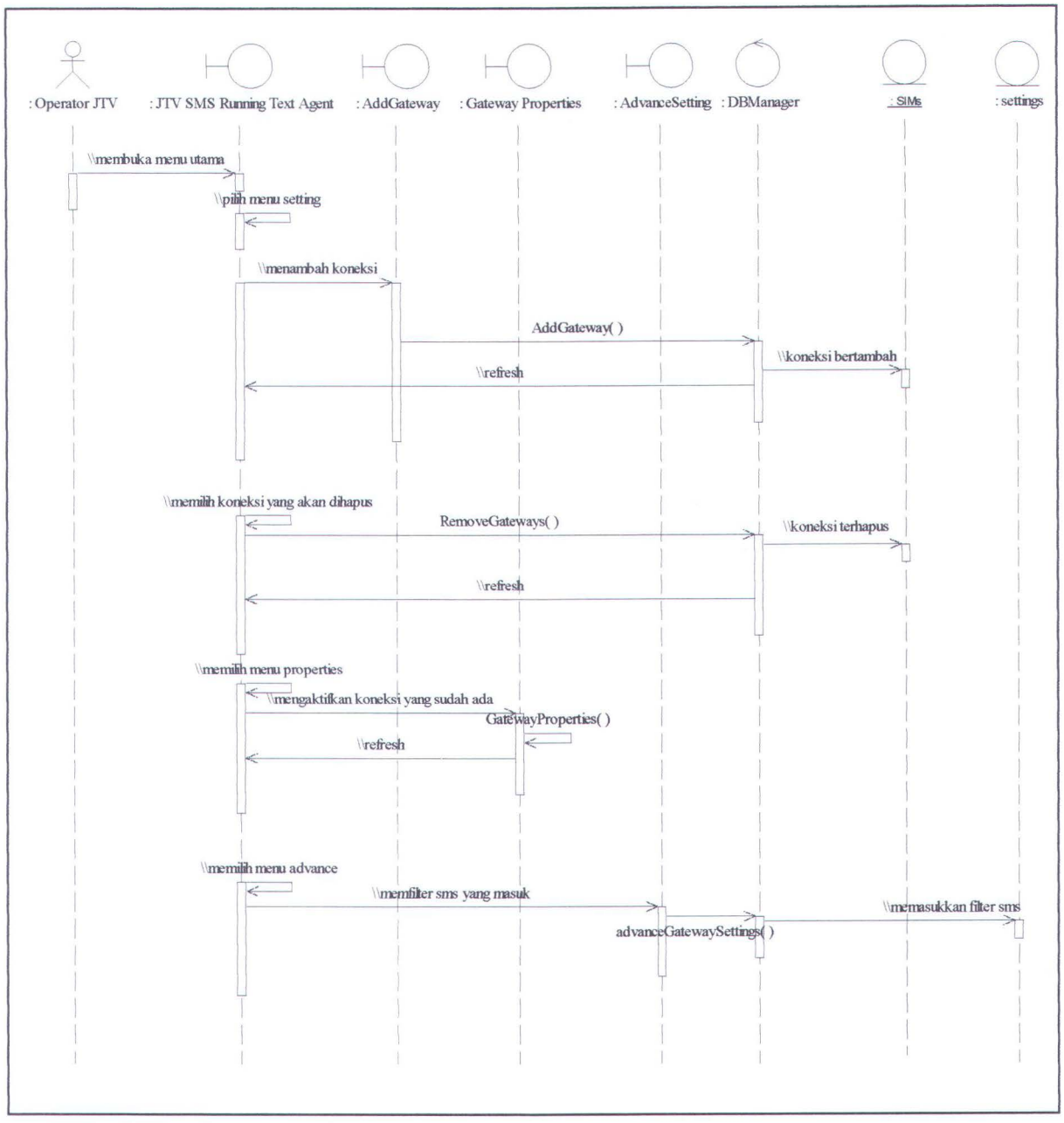
Gambar 3.10 Sequence Diagram Manajemen SMS TV

3.2.2.1.8 Sequence Diagram Manajemen koneksi sms TV

Proses yang terjadi pada perangkat lunak ini adalah operator televisi yang menyambungkan koneksi yang terjadi antara komputer dan handphone agar dapat digunakan untuk proses sms. Proses koneksi tersebut lebih dikenal dengan sebutan sms gateway. Operator televisi dapat mengkoneksikan dan juga memberhentikan koneksi yang terjadi, dimana proses ini terdapat dalam sebuah menu *properties*.

Dan kemudian setting koneksi yang telah dibuat, datanya dapat di simpan ke dalam sebuah table yang bernama *SIMSs* dan akan ditampilkan. Ada tiga bentuk penampilan data koneksi yaitu ditampilkan secara *active*, dimana ini menampilkan koneksi yang sedang terhubung, ditampilkan secara *inactive*, yaitu menampilkan koneksi yang tidak terhubung, dan yang terakhir ditampilkan secara keseluruhan *all*, dimana data koneksi baik *active* maupun *inactive* akan ditampilkan semuanya. Operator televisi dapat menambah setting koneksi dan juga menghapus setting tersebut apabila memang setting tersebut sudah tidak gunakan lagi. Terdapat menu *advance* yang berfungsi untuk menyaring dan membatasi data sms yang masuk, sehingga dapat menghindari kemungkinan terjadinya sms yang tidak layak untuk ditampilkan. Data penyaringan dan pembatasan sms yang masuk dapat disimpan ke dalam sebuah file yang bernama *settings.ini* . Penggolongan data sms yang akan disaring berdasarkan *sender prefix* yaitu nomor telp pengirim, dimana hanya dengan memasukkan beberapa digit awal atau nomor keseluruhan, maka sms yang bisa masuk hanyalah yang berasal dari nomor yang telah diinputkan pada *sender prefix*. Penggolongan data sms yang akan disaring berdasarkan *message prefix* yaitu berdasarkan pesan awal yang ditulis, maka sms yang bisa masuk hanyalah sms yang berasal dari pesan yang telah diinputkan pada *message prefix*. Penggolongan data sms yang akan disaring berdasarkan *description prefix* adalah penggolongan dari deskripsi data yang berasal dari inputan terhadap *sender prefix* dan *message prefix*, sehingga operator televisi bisa memilih satu diantara banyak data yang ada, dan data yang dipilih

adalah sebagai penanda sms yang bisa masuk. Proses selengkapnya dapat dilihat pada gambar 3.11 berikut.



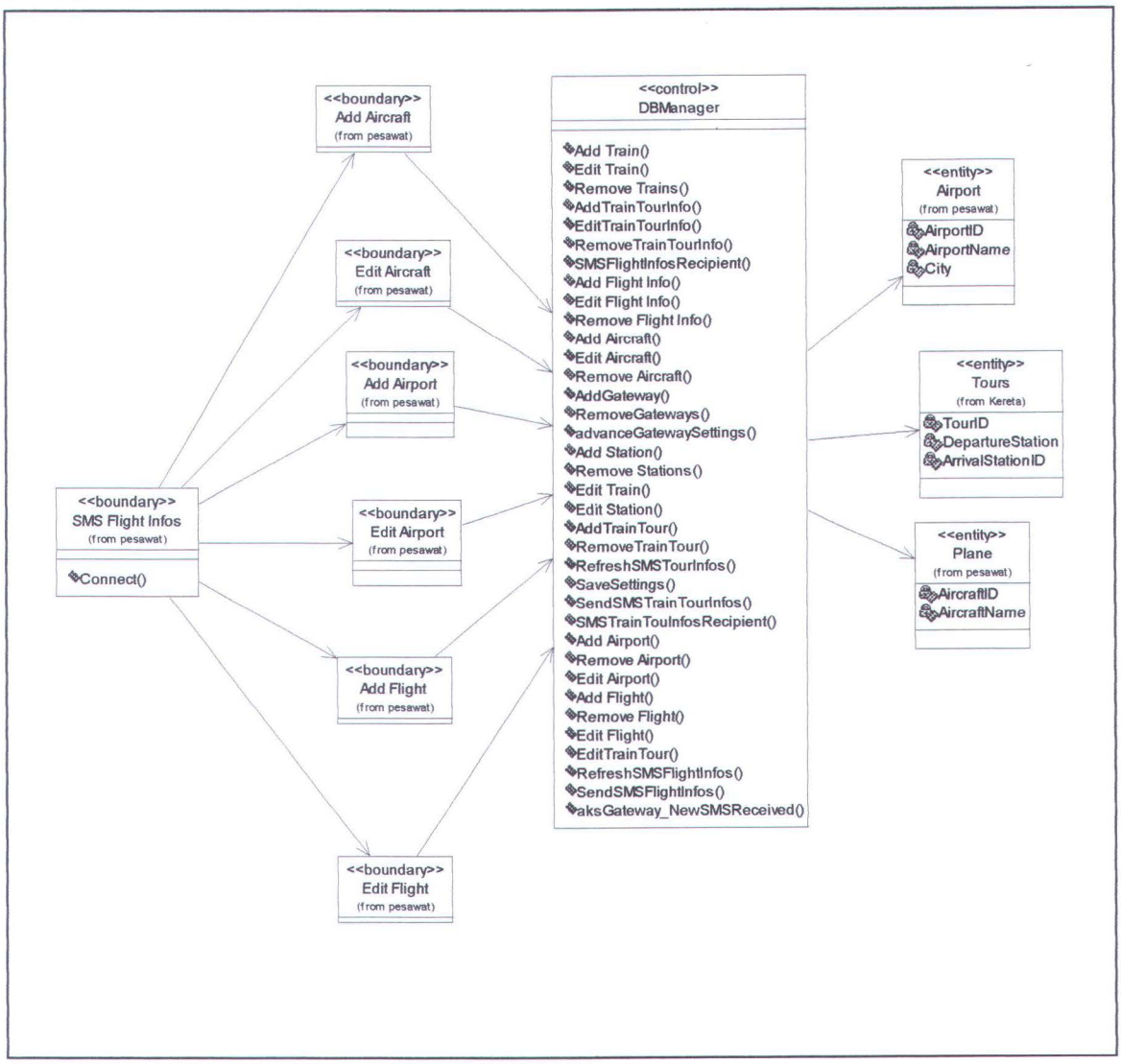
Gambar 3.11 Sequence Diagram Manajemen koneksi sms TV

3.2.2.2 Class Diagram

Setelah *sequence diagram* langkah selanjutnya adalah membuat *class diagram*. *Class diagram* ini menunjukkan relasi antar *class*.

3.2.2.2.1 Class Diagram Manajemen Data pesawat

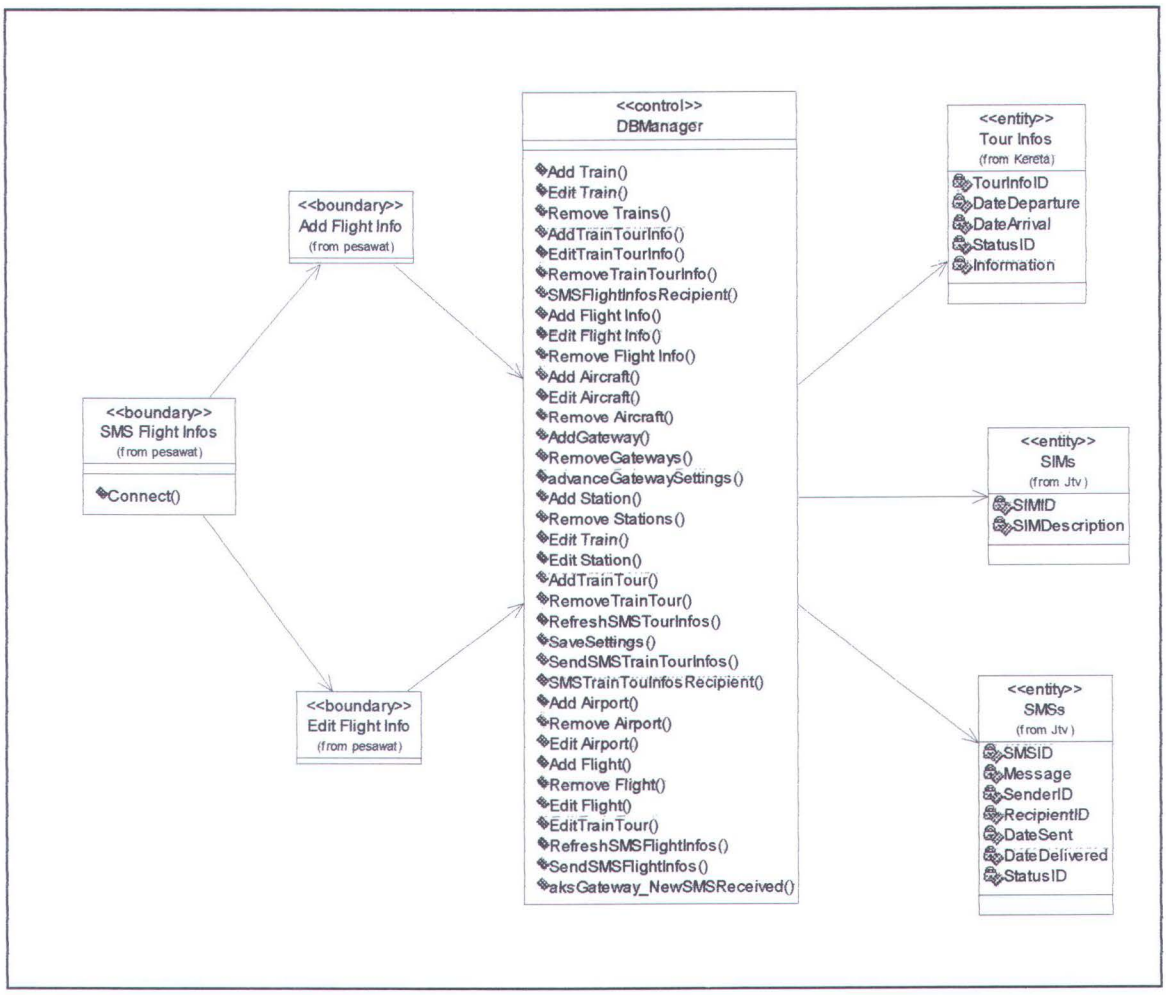
Hubungan antar *class* pada usecase manajemen data pesawat dapat dilihat pada gambar 3.12 berikut.



Gambar 3.12 Class Diagram Manajemen data pesawat

3.2.2.2.2 Class Diagram Manajemen Jadwal pesawat

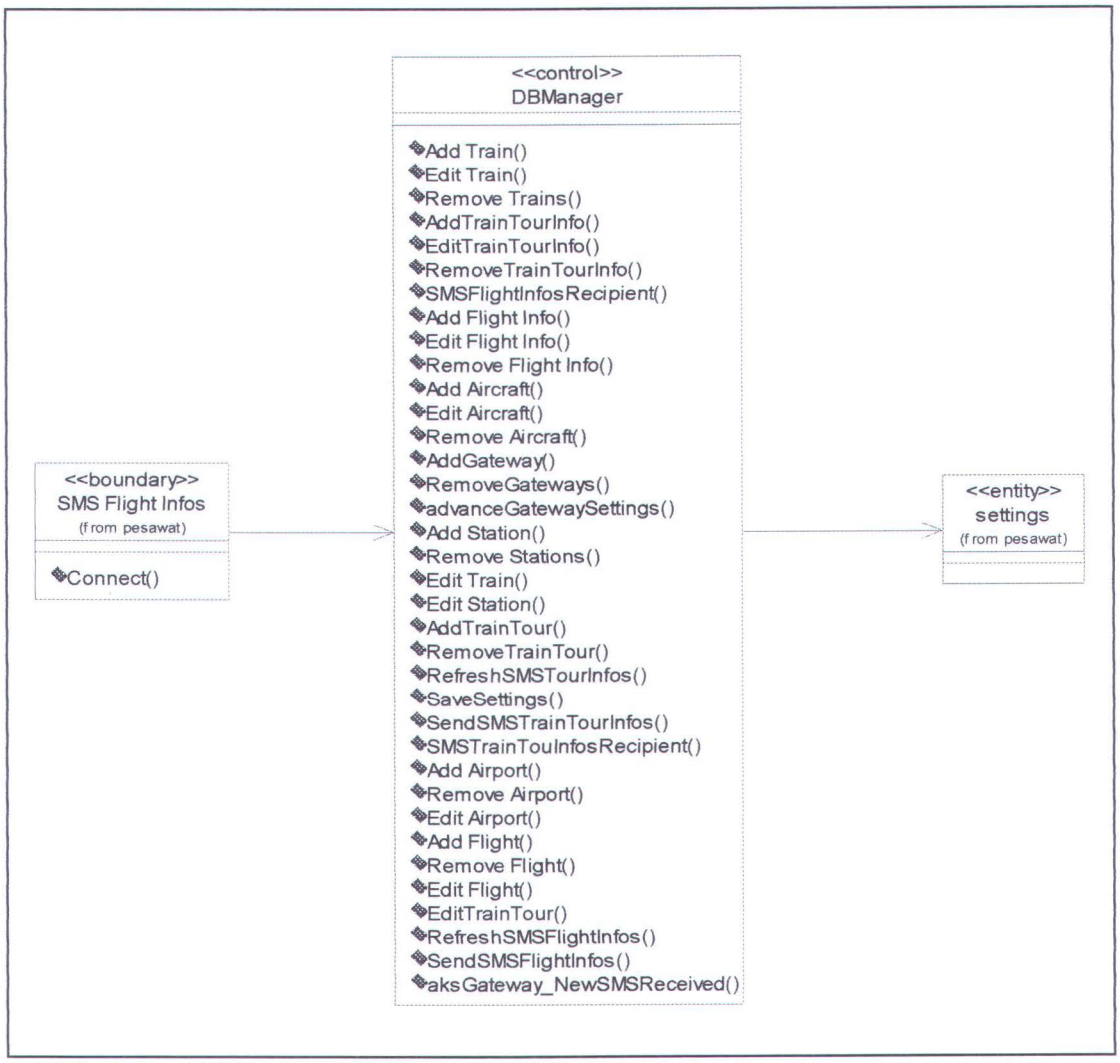
Hubungan antar *class* pada use case Manajemen jadwal pesawat dapat dilihat pada gambar 3.13 berikut.



Gambar 3.13 Class Diagram Manajemen jadwal pesawat

3.2.2.2.3 Class Diagram Manajemen Koneksi sms pesawat

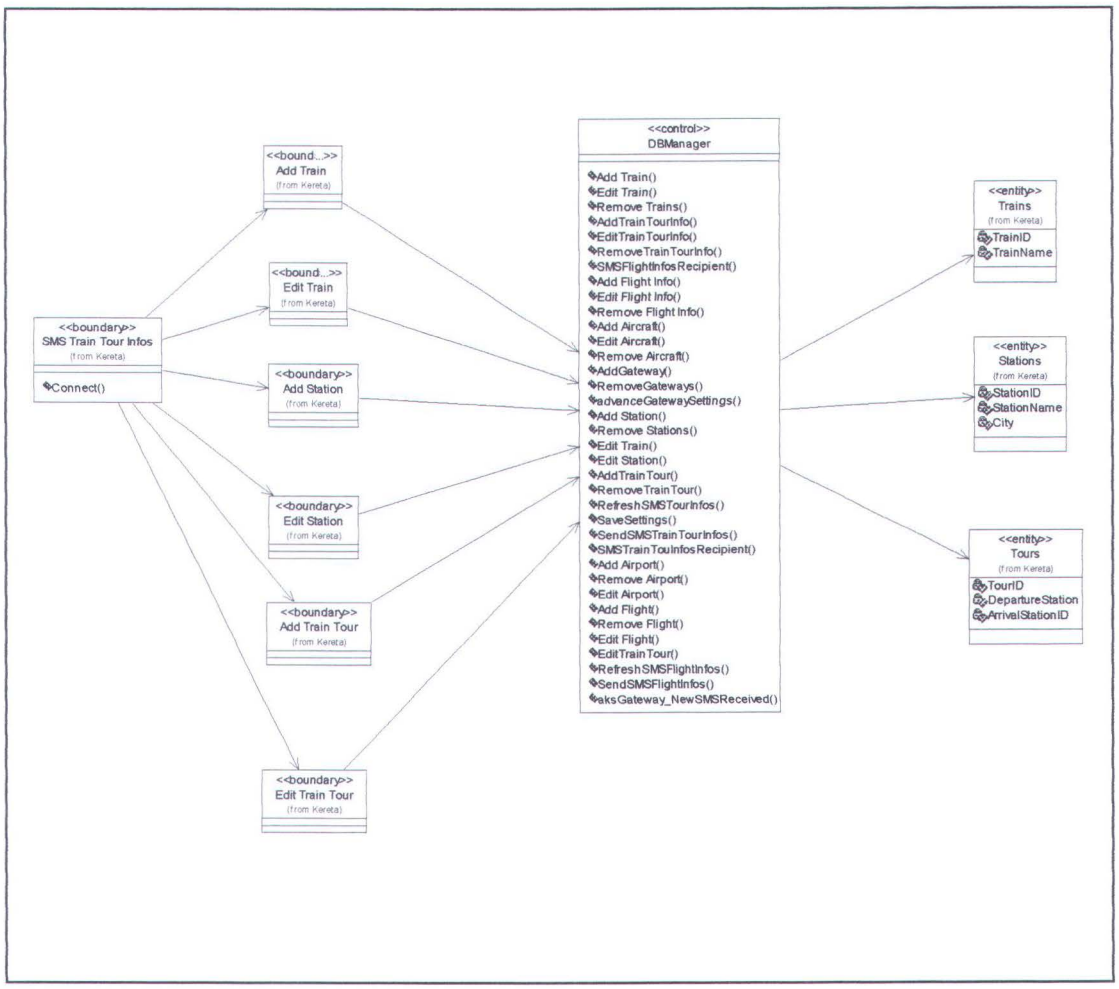
Hubungan antar *class* pada use case Manajemen koneksi sms pesawat dapat dilihat pada gambar 3.14 berikut.



Gambar 3.14 Class Diagram Manajemen koneksi sms pesawat

3.2.2.2.4 Class Diagram Manajemen Data kereta

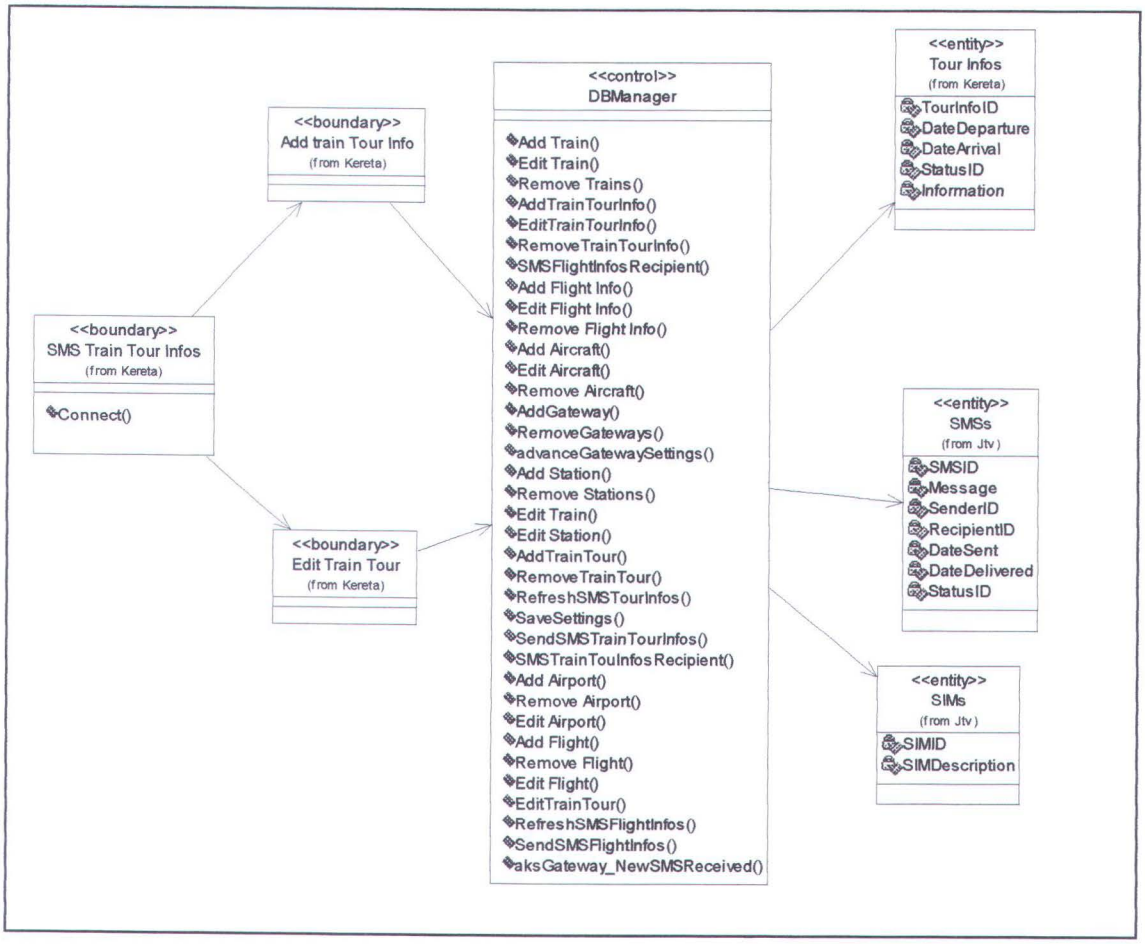
Hubungan antar *class* pada use case Manajemen data kereta dapat dilihat pada gambar 3.15 berikut.



Gambar 3.15 Class Diagram Manajemen data kereta

3.2.2.2.5 Class Diagram Manajemen Jadwal kereta

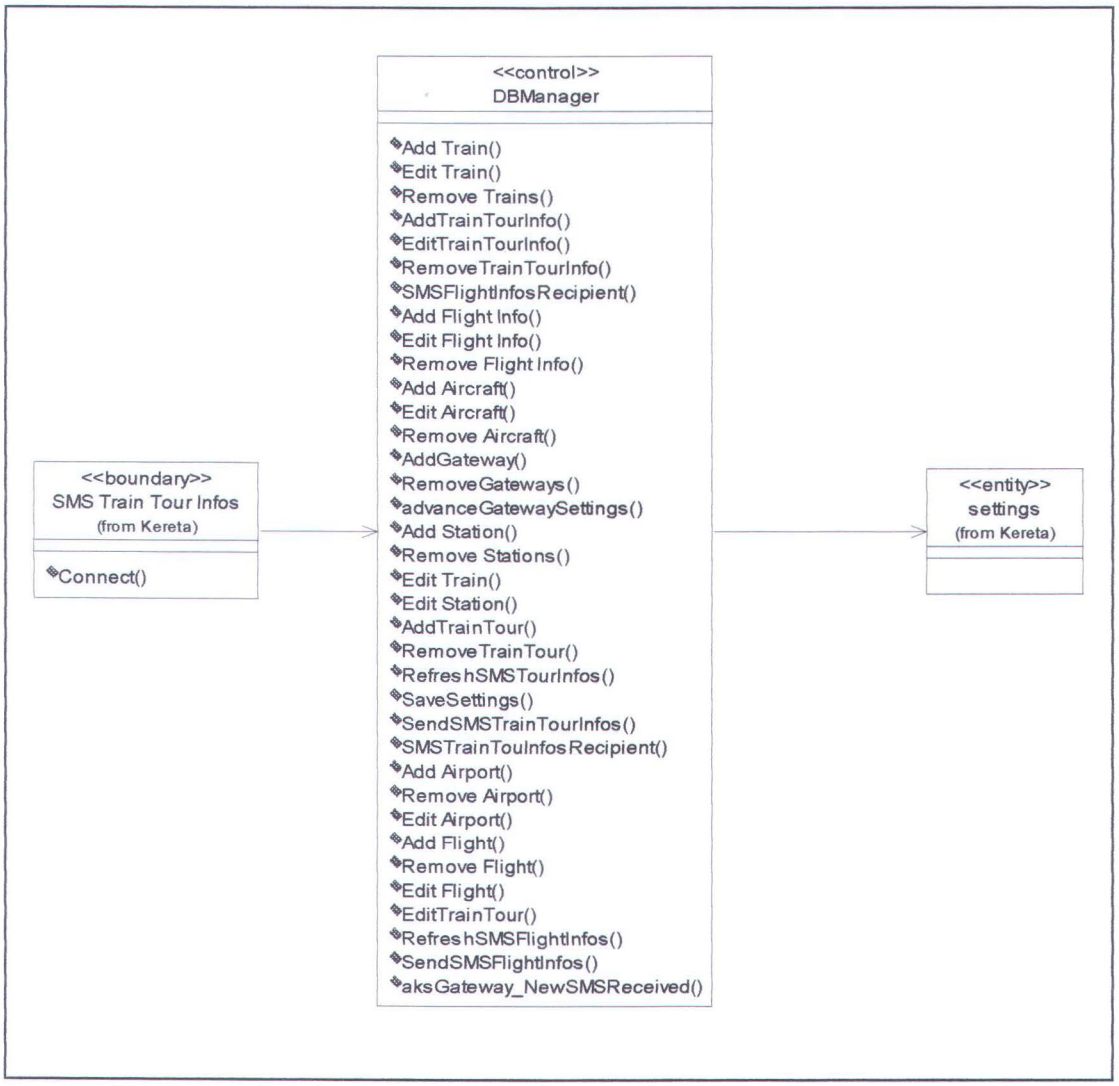
Hubungan antar *class* pada use case manajemen jadwal kereta dapat dilihat pada gambar 3.16 berikut.



Gambar 3.16 Class Diagram Manajemen jadwal kereta

3.2.2.2.6 Class Diagram Manajemen Koneksi sms kereta

Hubungan antar *class* pada use case manajemen koneksi sms kereta dapat dilihat pada gambar 3.17 berikut.

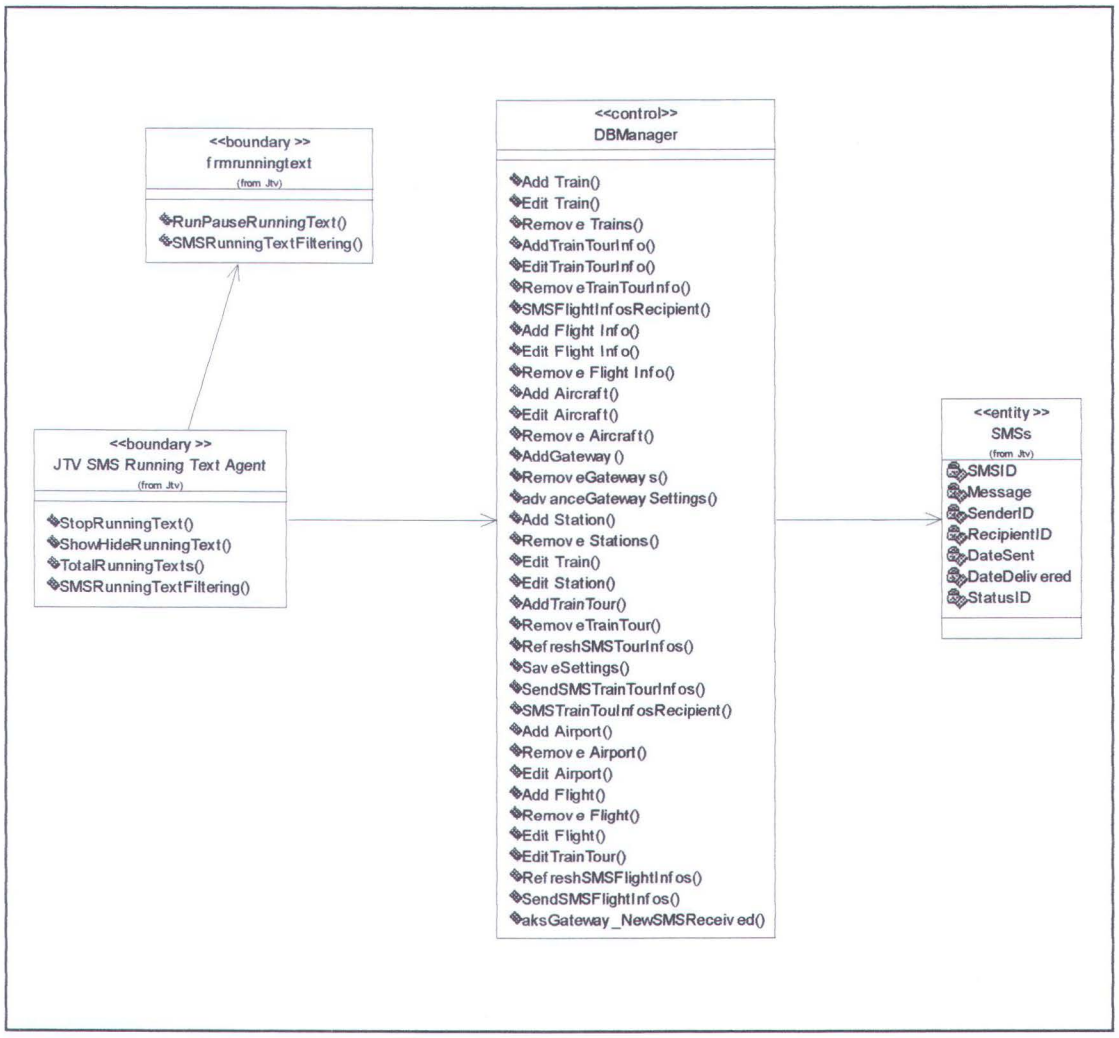


Gambar 3.17 Class Diagram Manajemen koneksi sms kereta

3.2.2.2.7 Class Diagram Manajemen sms TV

Hubungan antar *class* pada use case manajemen sms TV dapat dilihat pada gambar 3.18 berikut.

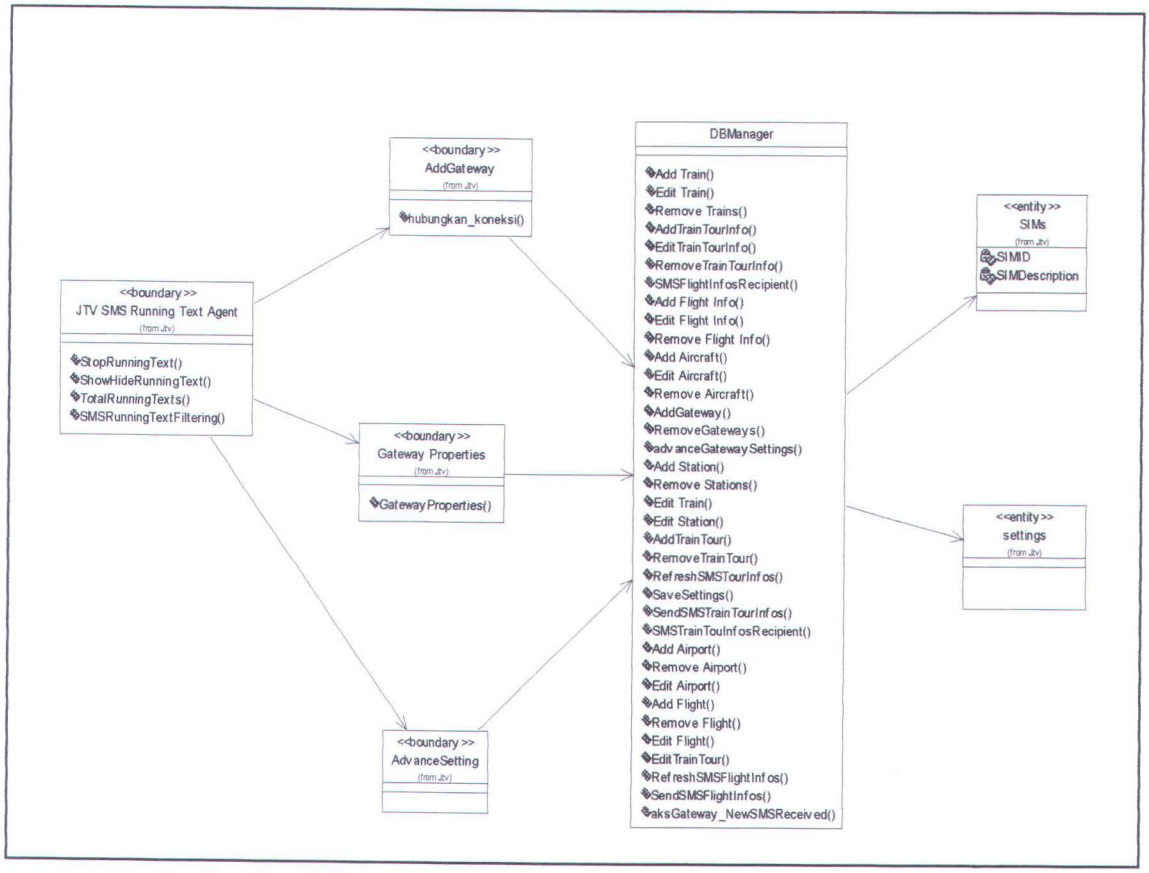
PERPUSTAKAAN ITS



Gambar 3.18 Class Diagram Manajemen sms TV

3.2.2.2.8 Class Diagram Manajemen Koneksi sms TV

Hubungan antar *class* pada use case manajemen koneksi sms TV dapat dilihat pada gambar 3.19 berikut.



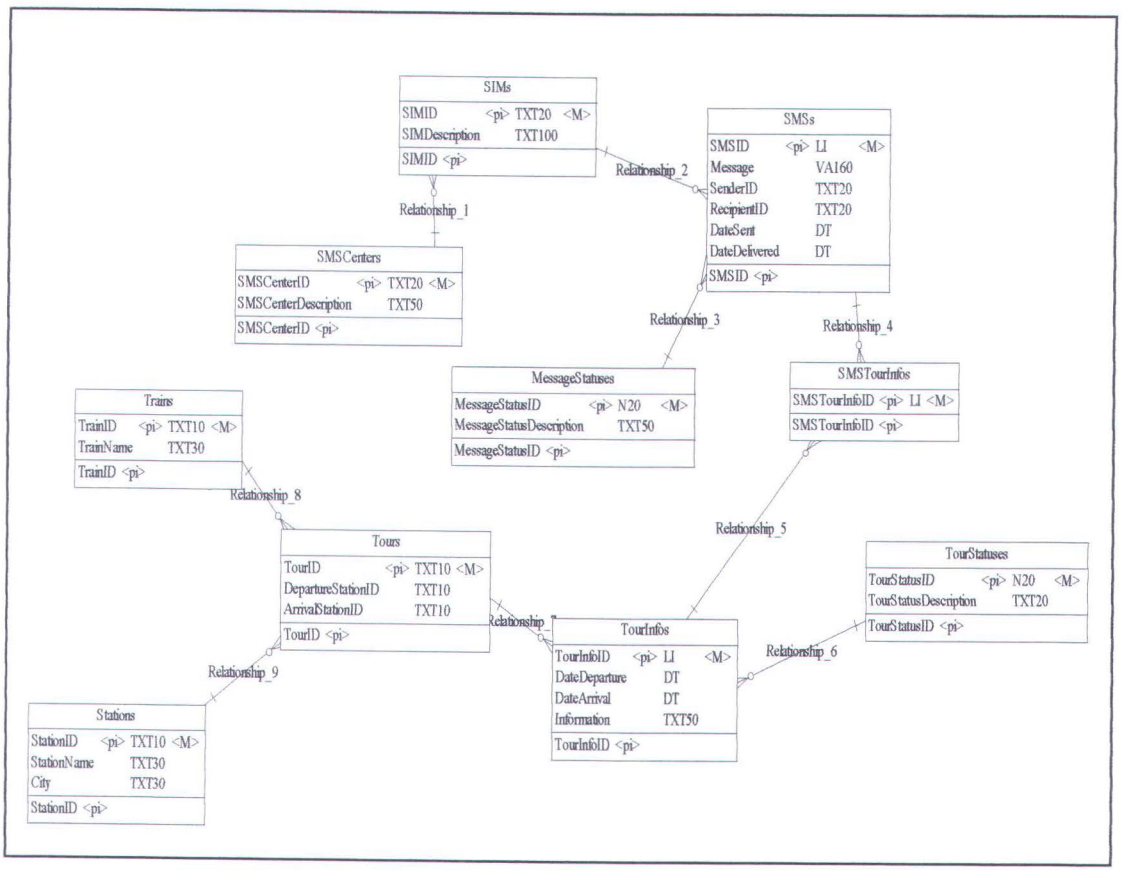
Gambar 3.19 Class Diagram Manajemen koneksi sms TV

3.3 Pemodelan Data

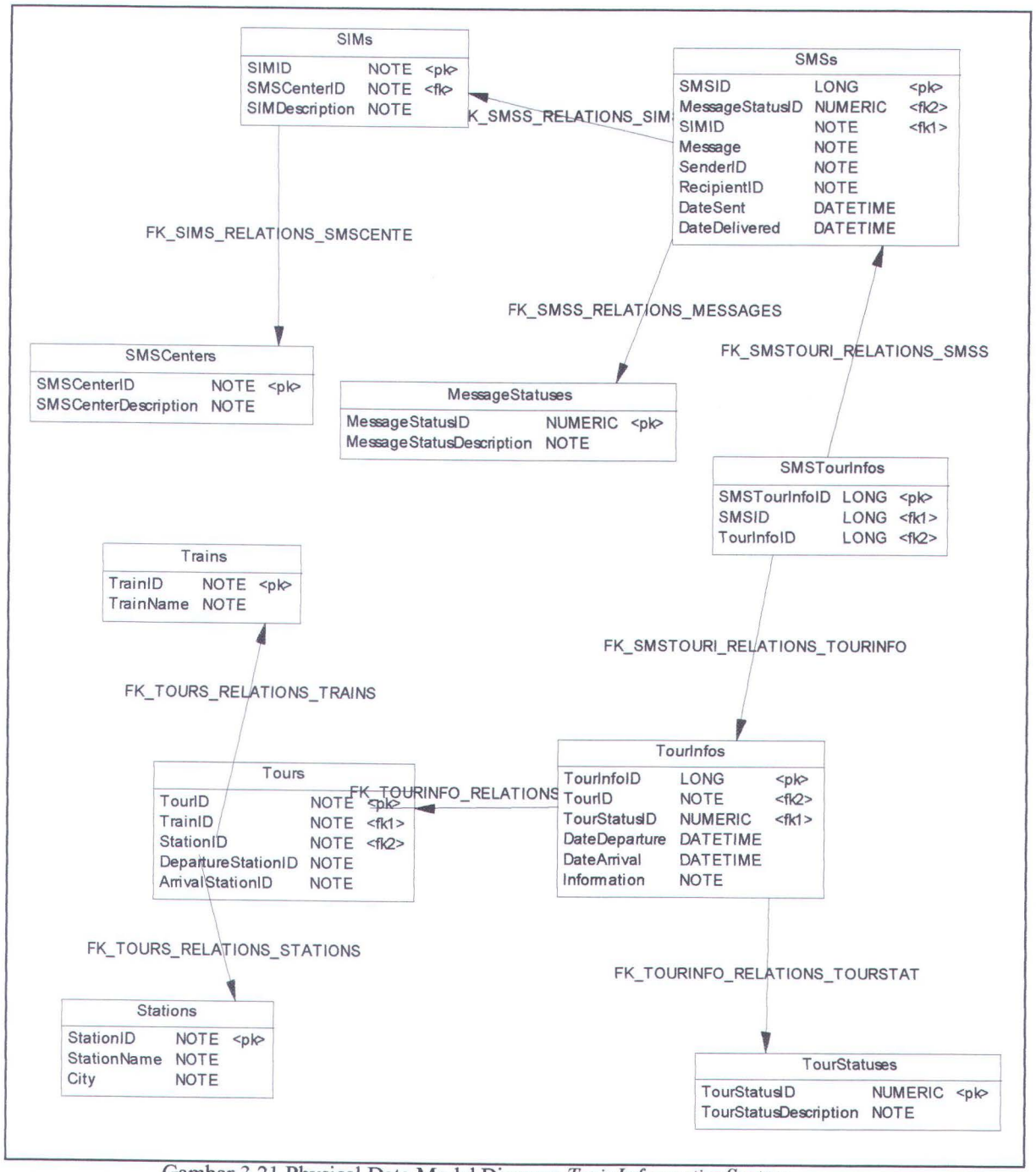
Tujuan dari Pemodelan data ini adalah untuk mempersiapkan data yang akan digunakan dalam perangkat lunak. Secara umum Pemodelan data dalam pembuatan perangkat lunak ini terdiri dari tiga basis data, basis data yang pertama adalah basis data yang berfungsi untuk menyimpan data-data yang berhubungan dengan perusahaan penerbangan dengan nama *FlightInformationSystem.* Basis data kedua adalah basis data yang berfungsi untuk menyimpan data-data yang berhubungan dengan perusahaan kereta api dengan nama *TrainInformationSystem.* Dan basis data terakhir adalah basis data yang berfungsi

untuk menyimpan data-data yang berhubungan dengan perusahaan televisi, dalam hal ini perusahaan televisi yang bersangkutan adalah PT. JTV (Jawa Timur Televisi) dengan nama *TVOnair*. Secara bentuk fisik, prinsip penyimpanan data, dan proses kerja pada basis data *FlightInformationSystem* dan *TrainInformationSystem* adalah sama, untuk itu pada sub bab ini hanya akan dijelaskan Pemodelan data pada *TrainInformationSystem* dan *TVOnair*.

Gambar 3.20 dan 3.21 berikut ini akan menggambarkan Pemodelan basis data yang pertama, disebut basis data *TrainInformationSystem*, dengan menggunakan *Conceptual Data Model diagram* (CDM) dan *Physical Data Model diagram* (PDM)



Gambar 3.20 Conceptual Data Model Diagram *TrainInformationSystem*



Gambar 3.21 Physical Data Model Diagram *TrainInformationSystem*

Pada *data model diagram* di atas memiliki Sepuluh buah tabel, penjelasan secara lebih detail dari tabel- tabel tersebut adalah sebagai berikut:

a. Tabel Trains

Tabel ini digunakan untuk menyimpan data kereta api. Field – field yang terdapat pada tabel Trains diuraikan pada tabel 3.3 berikut.

Tabel 3.3 Penjelasan Tabel Trains

Nama Field	Penjelasan
TrainID	Berisi data Id kereta api. Field ini berfungsi sebagai primary key dari tabel Trains. Tipe data varchar/text(10)
TrainName	Berisi data nama kereta api. Tipe data varchar/text(30).

b. Tabel Stations

Tabel ini digunakan untuk menyimpan data stasiun kereta api. Field-field yang terdapat pada tabel Stations diuraikan pada tabel 3.4 berikut.

Tabel 3.4 Penjelasan Tabel Stations

Nama Field	Penjelasan
StationID	Berisi data Id stasiun kereta api. Field ini berfungsi sebagai primary key dari tabel Stations. Tipe data varchar/text(10)
StationName	Berisi data nama stasiun kereta api. Tipe data varchar/text(30)
City	Berisi data nama kota dari stasiun berada. Tipe data varchar/text(30)

c. Tabel Tours

Tabel ini digunakan untuk menyimpan data rute perjalanan/jurusan yang ditempuh oleh kereta api. Kolom-kolom yang terdapat pada tabel Tours diuraikan pada tabel 3.5 berikut.

Tabel 3.5 Penjelasan Tabel Tours

Nama Field	Penjelasan
TourID	Berisi data Id tour kereta api. Field ini berfungsi sebagai primary key dari tabel Tours. Tipe data varchar/text(10)
TrainID	Berisi data Id kereta api. Field ini berfungsi sebagai primary- foreign key yang mengacu pada tabel Trains. Tipe data varchar/text(10)
DepartureStationID	Berisi data Id stasiun kereta api, sebagai stasiun keberangkatan kereta api. Tipe data varchar/text(10)
ArrivalStationID	Berisi data Id stasiun kereta api, sebagai stasiun kedatangan kereta api. Tipe data varchar/text(10)

d. Tabel TourInfos

Tabel ini digunakan untuk menyimpan data jadwal kereta api yang telah mengalami status, baik status keterlambatan dan pembatalan jalan. Kolom-kolom yang terdapat pada tabel TourInfos diuraikan pada tabel 3.6 berikut.

Tabel 3.6 Penjelasan Tabel TourInfos

Nama Field	Penjelasan
TourInfoID	Berisi Id dari tour info, informasi mengenai jadwal kereta api. Field ini berfungsi sebagai primary key dari tabel TourInfos. Tipe data Long Integer/autonumber.
TourID	Berisi data Id tour kereta api. Field ini berfungsi sebagai primary-foreign key yang mengacu pada tabel Tours. Tipe data varchar/text(10)
DateDeparture	Berisi data waktu keberangkatan kereta api. Tipe data date/time.
DateArrival	Berisi data waktu kedatangan kereta api. Tipe data date/time.
StatusID	Berisi data id status yang terjadi terhadap kereta api. Field ini merupakan foreign key yang mengacu pada tabel TourStatuses. Tipe data integer/number
Information	Berisi informasi akibat status yang terjadi pada kereta api. Tipe data varchar/text(50)

e. Tabel TourStatuses

Tabel ini digunakan untuk menyimpan data status yang terjadi pada kereta api. Kolom-kolom yang terdapat pada tabel TourStatuses diuraikan pada tabel 3.7 berikut.

Tabel 3.7 Penjelasan Tabel TourStatuses

Nama Field	Penjelasan
TourStatusID	Berisi data id status yang terjadi terhadap kereta api. Field ini merupakan primary key pada tabel TourStatuses. Tipe data integer/number
TourStatusDescripition	Berisi data deskripsi status yang terjadi terhadap kereta api. Tipe data varchar/text(20)

f. Tabel SIMs

Tabel ini digunakan untuk menyimpan data nomor telepon, baik nomor telepon yang digunakan untuk mengirim sms maupun nomor telepon sebagai tujuan sms. Field–field yang terdapat pada tabel SIMs diuraikan pada tabel 3.8 berikut.

Tabel 3.8 Penjelasan Tabel SIMs

Nama Field	Penjelasan
SIMID	Berisi data nomor telepon. Field ini merupakan primary key pada tabel SIMs.Tipe data *varchar/text* (20)
SIMDescription	Berisi data deskripsi dari sim yang akan dipakai sebagai *sender* atau *receiver* dalam proses sms. Tipe data varchar/text(100)
SMSCenterID	Berisi data nomor telepon SMS Center dari setiap provider yang dipakai dalam proses sms. Field ini merupakan primary-foreign key yang mengacu pada tabel SMSCenters. Tipe data varchar/text(20)

g. Tabel SMSCenters

Tabel ini digunakan untuk menyimpan data nomor telepon SMS Center yang berasal dari provider perusahaan komunikasi. Kolom-kolom yang terdapat pada tabel SMSCenters diuraikan pada tabel 3.9 berikut.

Tabel 3.9 Penjelasan Tabel SMSCenters

Nama Field	Penjelasan
SMSCenterID	Berisi data nomor telepon SMS Center dari setiap provider yang dipakai dalam proses sms. Field ini merupakan primary key pada tabel SMSCenters. Tipe data varchar/text(20)
SMSCenterDescription	Berisi data deskripsi dari SMS Center yaitu data berbagai provider perusahaan komunikasi. Tipe data varchar/text(50)

h. Tabel SMSs

Tabel ini digunakan untuk menyimpan data mengenai proses sms yang terjadi. Kolom-kolom yang terdapat pada tabel SMSs diuraikan pada tabel 3.10 berikut.

Tabel 3.10 Penjelasan Tabel SMSs

Nama Field	Penjelasan
SMSID	Berisi data Id terhadap proses sms. Field ini berfungsi sebagai primary key dari tabel SMSs, field ini auto increment dengan Tipe data long integer/autonumber.
Message	Berisi data pesan yang berupa jadwal kereta api yang mengalami status. Tipe data *varchar*/memo
SenderID	Berisi data nomor telepon yang digunakan untuk proses pengiriman sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel SIMs. Tipe data varchar/text (20)
RecipientID	Berisi data nomor telepon yang digunakan untuk proses penerimaan sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel SIMs. Tipe data varchar/text (20)
DateSent	Berisi data waktu pada saat sms dikirim ke nomor yang dituju. Tipe data date/time
DateDelivered	Berisi data waktu pada saat sms terkirim ke nomor yang dituju. Tipe data date/time.
StatusID	Berisi data id dari status yang terjadi pada sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel MessageStatuses. Tipe data integer/number

i. Tabel MessageStatuses

Tabel ini digunakan untuk menyimpan data status yang terjadi pada proses sms. Kolom-kolom yang terdapat pada tabel MessageStatuses diuraikan pada tabel 3.11 berikut.

Tabel 3.11 Penjelasan Tabel MessageStatuses

Nama Field	Penjelasan
MessageStatusID	Berisi data id dari status yang terjadi pada sms. Field ini berfungsi sebagai Primary key pada tabel MessageStatuses. Tipe data integer/number
MessageStatusDescription	Berisi data deskripsi status yang terjadi pada sms. Tipe data varchar/text(50)

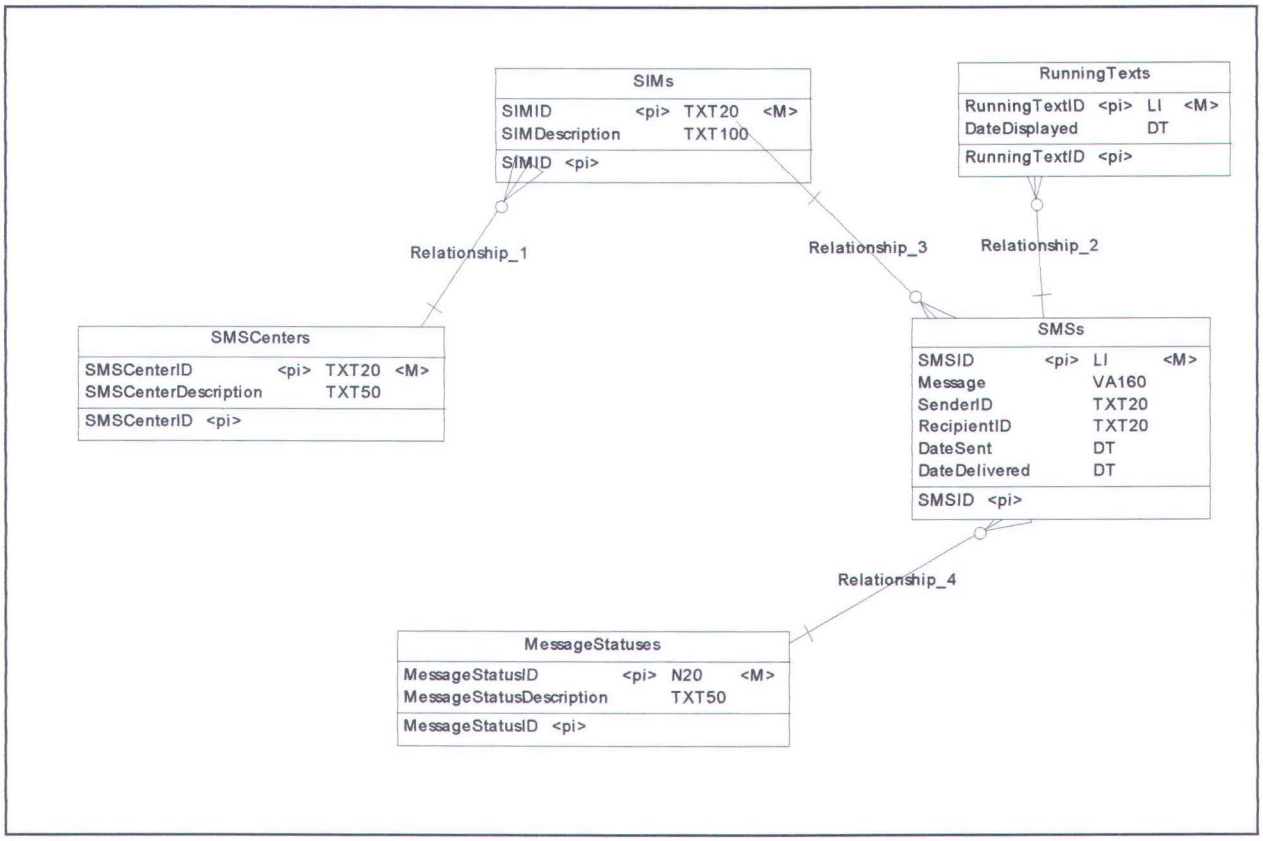
j. Tabel SMSTourInfos

Tabel ini digunakan untuk menyimpan data dari SMSID yang merupakan primary-foreign key yang berasal dari tabel SMSs dan TourInfoID yang juga merupakan primary-foreign key yang berasal dari tabel TourInfos. Dimana data dari masing-masing tabel tersebut akan dijadikan satu kemudian digunakan untuk proses pengiriman sms dan setelah itu isi data tersebut akan ditampilkan dalam menu utama. Kolom-kolom yang terdapat pada tabel SMSTourInfos diuraikan pada tabel 3.12 berikut.

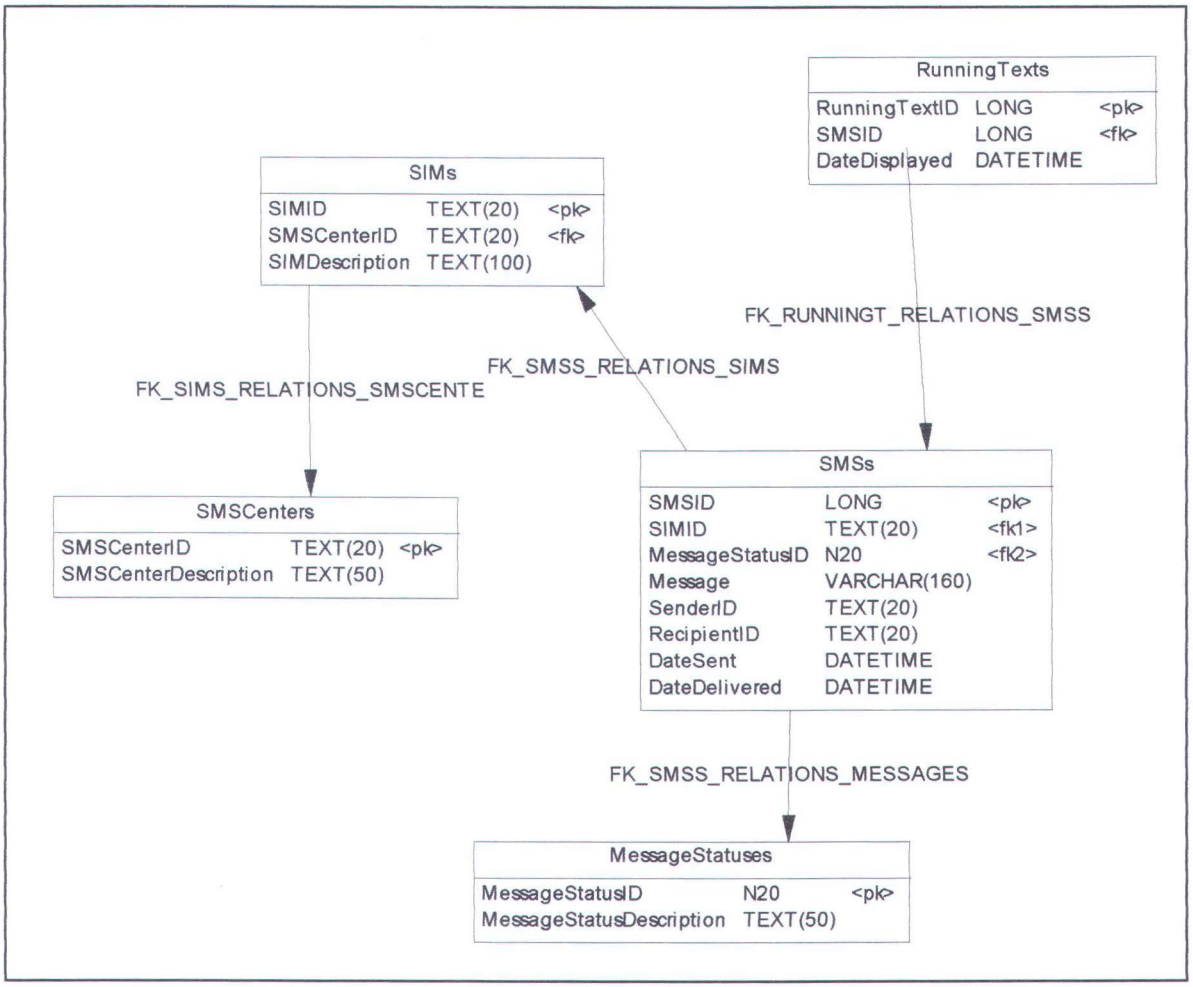
Tabel 3.12 Penjelasan Tabel SMSTourInfo

Nama Field	Penjelasan
SMSTourInfoID	Berisi data Id terhadap penggabungan dua entiti, yaitu proses sms dan informasi jadwal kereta api. Field ini berfungsi sebagai primary key pada tabel SMSTourInfo, field ini auto increment dengan Tipe data long integer/autonumber.
TourInfoID	Berisi Id dari tour info, informasi mengenai jadwal kereta api. Field ini berfungsi sebagai primary-foreign key dari tabel TTourInfos. Tipe data Long Integer/autonumber.
SMSID	Berisi data Id terhadap proses sms. Field ini berfungsi sebagai primary-foreign key yang mengacu pada tabel SMSs. Tipe data long integer/autonumber.

Gambar 3.22 dan 3.23 berikut ini akan menggambarkan Pemodelan basis data yang kedua, disebut basis data *TVOnair*, dengan menggunakan *Conceptual Data Model diagram* (CDM) dan *Physical Data Model diagram* (PDM)



Gambar 3.22 Conceptual Data Model diagram TVOnair



Gambar 3.23 Physical Data Model diagram TVOnair

Pada *data model diagram* di atas memiliki lima buah tabel, yang tidak jauh berbeda dengan tabel yang terdapat pada *TrainInformationSystem.* Penjelasan secara lebih detail dari tabel-tabel tersebut adalah sebagai berikut:

a. Tabel SMSs

Tabel ini digunakan untuk menyimpan data mengenai proses sms yang terjadi. Kolom-kolom yang terdapat pada tabel SMSs diuraikan pada tabel 3.13 berikut.

Tabel 3.13 Penjelasan Tabel SMSs

Nama Field	Penjelasan
SMSID	Berisi data Id terhadap proses sms. Field ini berfungsi sebagai primary key dari tabel SMSs, field ini auto increment dengan Tipe data long integer/autonumber.
Message	Berisi data pesan yang berupa jadwal kereta api yang mengalami status. Tipe data *varchar*/memo
SenderID	Berisi data nomor telepon yang digunakan untuk proses pengiriman sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel SIMs. Tipe data varchar/text (20)
RecipientID	Berisi data nomor telepon yang digunakan untuk proses penerimaan sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel SIMs. Tipe data varchar/text (20)
DateSent	Berisi data waktu pada saat sms dikirim ke nomor yang dituju. Tipe data date/time
DateDelivered	Berisi data waktu pada saat sms terkirim ke nomor yang dituju. Tipe data date/time.
StatusID	Berisi data id dari status yang terjadi pada sms. Field ini berfungsi sebagai foreign key yang mengacu pada tabel MessageStatuses. Tipe data integer/number

b. Tabel SMSCenters

Tabel ini digunakan untuk menyimpan data nomor telepon SMS Center yang berasal dari provider perusahaan komunikasi. Kolom-kolom yang terdapat pada tabel SMSCenters diuraikan pada tabel 3.14 berikut.

Tabel 3.14 Penjelasan Tabel SMSCenters

Nama Field	Penjelasan
SMSCenterID	Berisi data nomor telepon SMS Center dari setiap provider yang dipakai dalam proses sms. Field ini merupakan primary key pada tabel SMSCenters. Tipe data varchar/text(20)
SMSCenterDescription	Berisi data deskripsi dari SMS Center yaitu data berbagai provider perusahaan komunikasi. Tipe data varchar/text(50)

c. Tabel SIMs

Tabel ini digunakan untuk menyimpan data nomor telepon, baik nomor telepon yang digunakan untuk mengirim sms maupun nomor telepon sebagai tujuan sms. Field–field yang terdapat pada tabel SIMs diuraikan pada tabel 3.15 berikut.

Tabel 3.15 Penjelasan Tabel TSIMs

Nama Field	Penjelasan
SIMID	Berisi data nomor telepon. Field ini merupakan primary key pada tabel SIMs.Tipe data *varchar/text* (20)
SIMDescription	Berisi data deskripsi dari sim yang akan dipakai sebagai *sender* atau *receiver* dalam proses sms. Tipe data varchar/text(100)
SMSCenterID	Berisi data nomor telepon SMS Center dari setiap provider yang dipakai dalam proses sms. Field ini merupakan primary-foreign key yang mengacu pada tabel SMSCenters. Tipe data varchar/text(20)

d. Tabel MessageStatuses

Tabel ini digunakan untuk menyimpan data status yang terjadi pada proses sms. Kolom-kolom yang terdapat pada tabel MessageStatuses diuraikan pada tabel 3.16 berikut.

Tabel 3.16 Penjelasan Tabel MessageStatuses

Nama Field	Penjelasan
MessageStatusID	Berisi data id dari status yang terjadi pada sms. Field ini berfungsi sebagai Primary key pada tabel MessageStatuses. Tipe data integer/number
MessageStatusDescription	Berisi data deskripsi status yang terjadi pada sms. Tipe data varchar/text(50)

e. Tabel RunningTexts

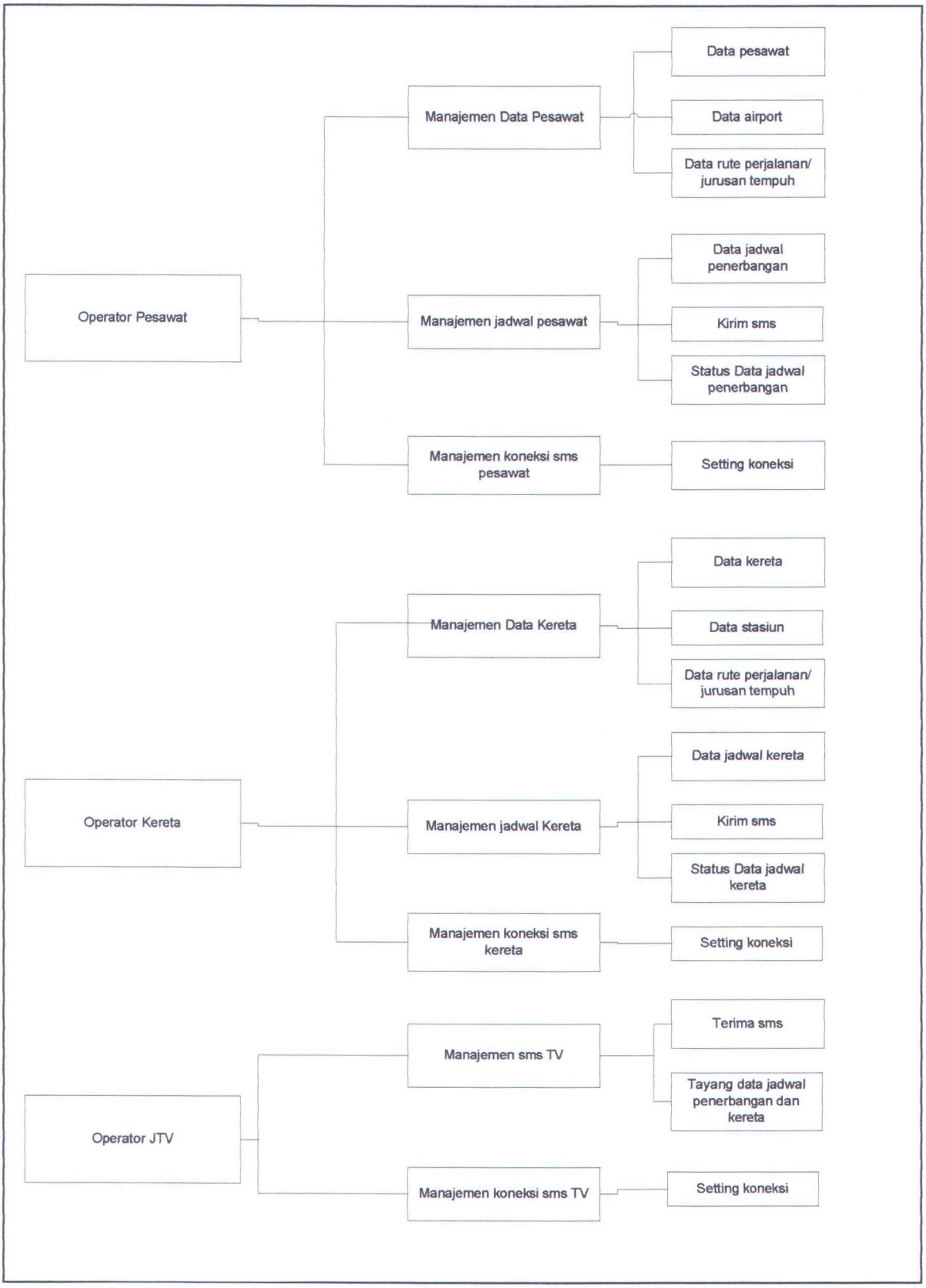
Tabel ini digunakan untuk menyimpan data sehubungan dengan sms yang diterima, dan isi data dari tabel ini akan ditayangkan dalam bentuk *Running Text* pada layar televisi. Field–field yang terdapat pada tabel RunningTexts diuraikan pada tabel 3.17 berikut.

Tabel 3.17 Penjelasan Tabel RunningTexts

Nama Field	Penjelasan
RunningTextID	Berisi data Id terhadap proses sms yang masuk. Field ini berfungsi sebagai primary key dari tabel RunningTexts, field ini auto increment dengan Tipe data long integer/autonumber.
SMSID	Berisi data Id terhadap proses sms. Field ini berfungsi sebagai primary-foreigner key dari tabel SMSs, Tipe data long integer/autonumber.
DateDisplayed	Berisi data waktu pada saat sms ditampilkan di layar televisi. Tipe data Date/time

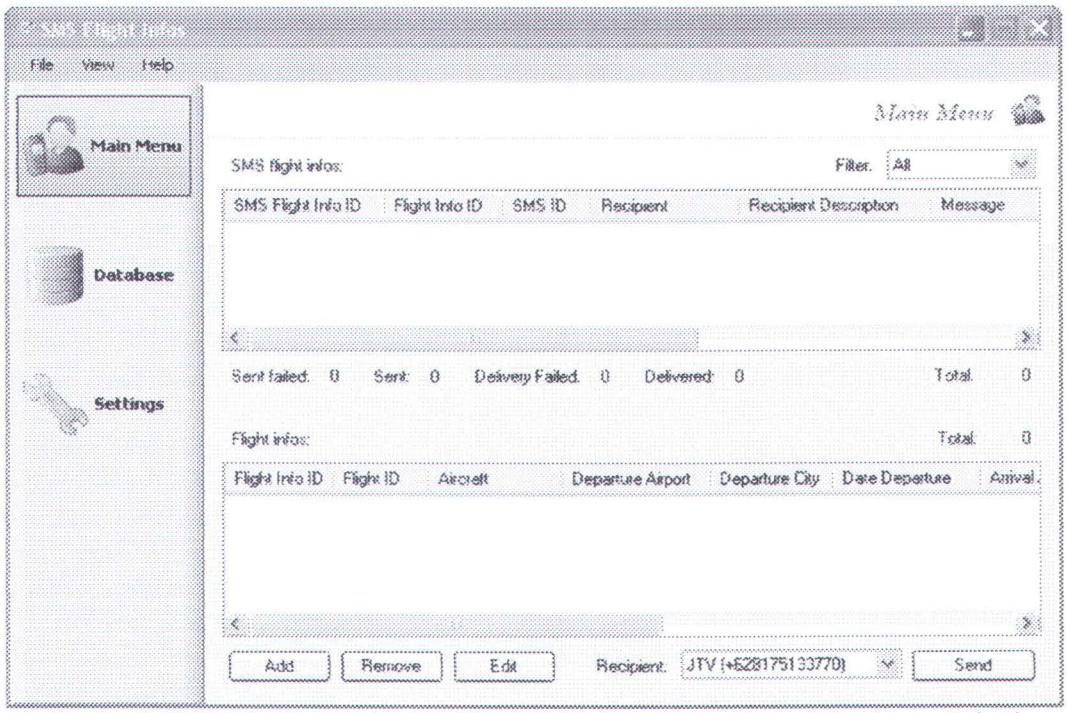
3.4 Pemodelan Antarmuka

Antarmuka merupakan komponen yang menghubungkan perangkat lunak dengan pengguna. Pada subbab ini merekap secara lebih umum dari desain antarmuka tersebut. Gambar 3.24 menunjukkan desain antarmuka perangkat lunak.



Gambar 3.24 Desain Antarmuka

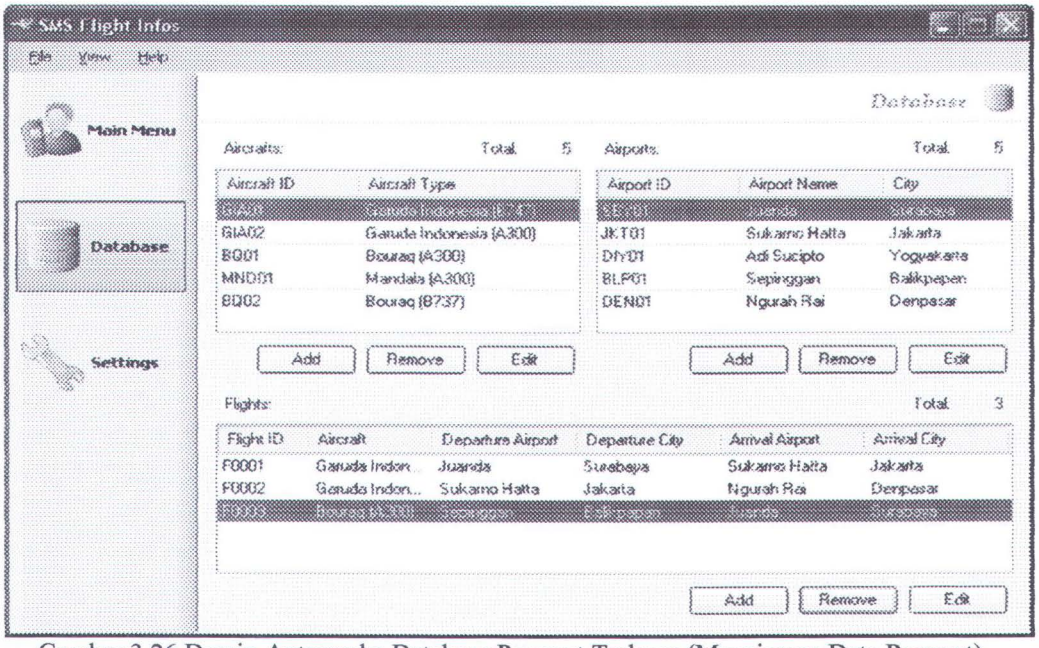
3.4.1 Pemodelan Antarmuka perangkat lunak pada Pesawat terbang



Gambar 3.25 Desain Antarmuka Pengiriman Jadwal Pesawat Terbang (Manajemen Jadwal Pesawat)

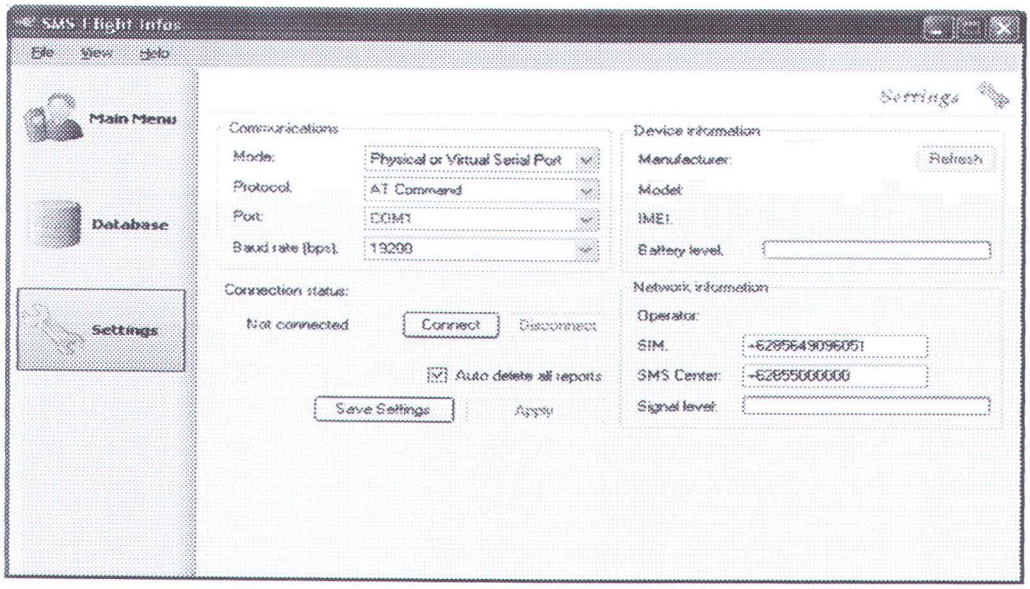
Aktivitas yang dilakukan oleh operator pesawat pada menu yang sesuai dengan gambar diatas adalah penambahan (*add*), penghapusan (*remove*), pengubahan (*edit*) yang dilakukan oleh operator pesawat terhadap data jadwal pesawat terbang, berikut pemasukan data status dan informasi yang terjadi pada pesawat terbang. Setelah itu operator pesawat dapat memilih nomor telepon tujuan sms pada menu *recipient* dan mengirimkannya dengan mengklik tombol *send.* Semua data jadwal penerbangan yang telah mengalami proses pengiriman atau *send* akan tertera di *SMS Flight Infos* yang dilengkapi dengan keterangan terhadap status sms, yaitu *sent failed, sent, delivery failed, delivered.* Dan pada menu *filter*,

operator dapat melihat total sms yang masuk berdasarkan penggolongan status sms yang ada.



Gambar 3.26 Desain Antarmuka Database Pesawat Terbang (Manajemen Data Pesawat)

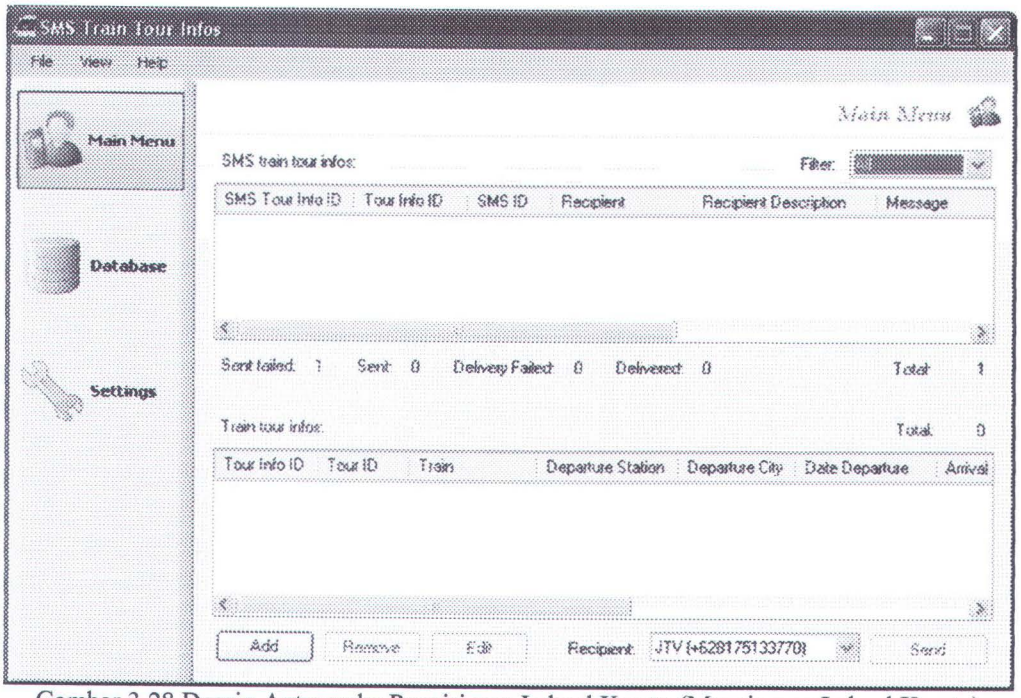
Aktivitas yang terjadi pada menu sesuai gambar diatas adalah penambahan (*add*), penghapusan (*remove*), pengubahan (*edit*) yang dilakukan oleh operator pesawat terhadap data pesawat terbang (*aircrafts*), data bandar udara (*airports*), data rute pesawat terbang (*Flight*). Data-data tersebut nantinya akan disimpan pada masing-masing table yang telah dibuat untuk masing-masing entry yaitu *Aircrafts, Airports, Flight.*



Gambar 3.27 Desain Antarmuka Setting Gateway (Manajemen Koneksi sms Pesawat)

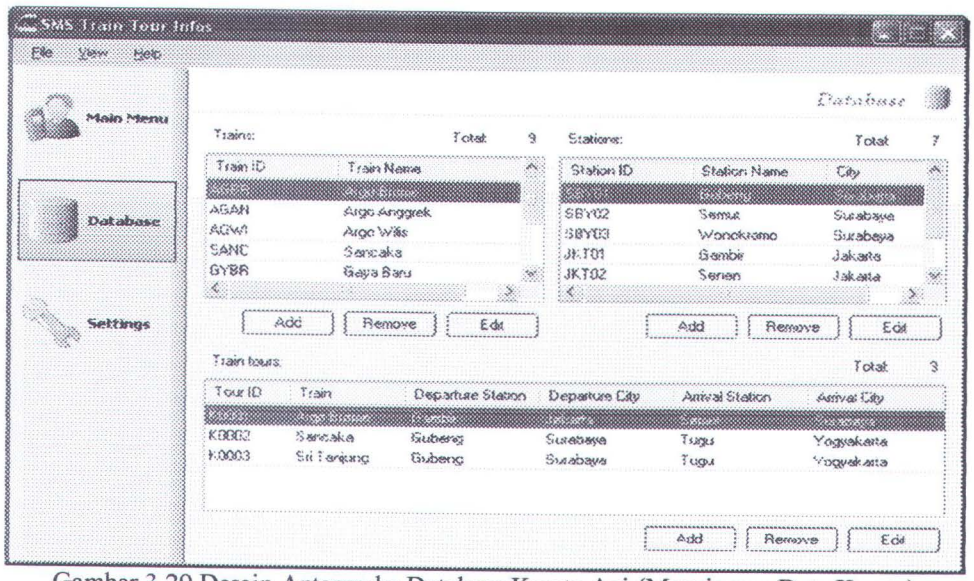
Operator pesawat sebelum melakukan proses pengiriman sms, terlebih dahulu melakukan setting koneksi untuk sms (SMS gateway), yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas.

3.4.2 Pemodelan Antarmuka perangkat lunak pada Kereta Api



Gambar 3.28 Desain Antarmuka Pengiriman Jadwal Kereta (Manajemen Jadwal Kereta)

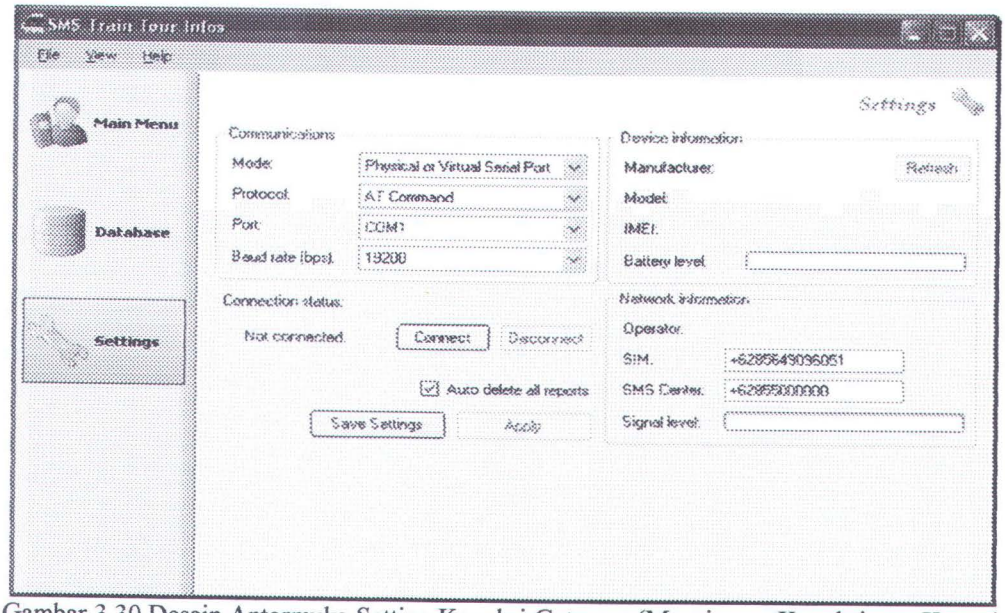
Aktivitas yang dilakukan oleh operator kereta pada menu yang sesuai dengan gambar diatas adalah penambahan (*add*), penghapusan (*remove*), pengubahan (*edit*) yang dilakukan oleh operator kereta terhadap data jadwal kereta api, berikut pemasukan data status dan informasi yang terjadi pada kereta api. Setelah itu operator kereta dapat memilih nomor telepon tujuan sms pada menu *recipient* dan mengirimkannya dengan mengklik tombol *send.* Semua data jadwal kereta api yang telah mengalami proses pengiriman atau *send* akan tertera di *SMS Traintour Infos* yang dilengkapi dengan keterangan terhadap status sms, yaitu *sent failed, sent, delivery failed, delivered.* Dan pada menu *filter*, operator dapat melihat total sms yang masuk berdasarkan penggolongan status sms yang ada.



Gambar 3.29 Desain Antarmuka Database Kereta Api (Manajemen Data Kereta)

Aktivitas yang terjadi pada menu sesuai gambar diatas adalah penambahan (*add*), penghapusan (*remove*), pengubahan (*edit*) yang dilakukan oleh operator kereta terhadap data kereta api (*Trains*), data stasiun (*stations*), data rute kereta api (*Train tours*). Data-data tersebut nantinya akan disimpan pada masing-masing

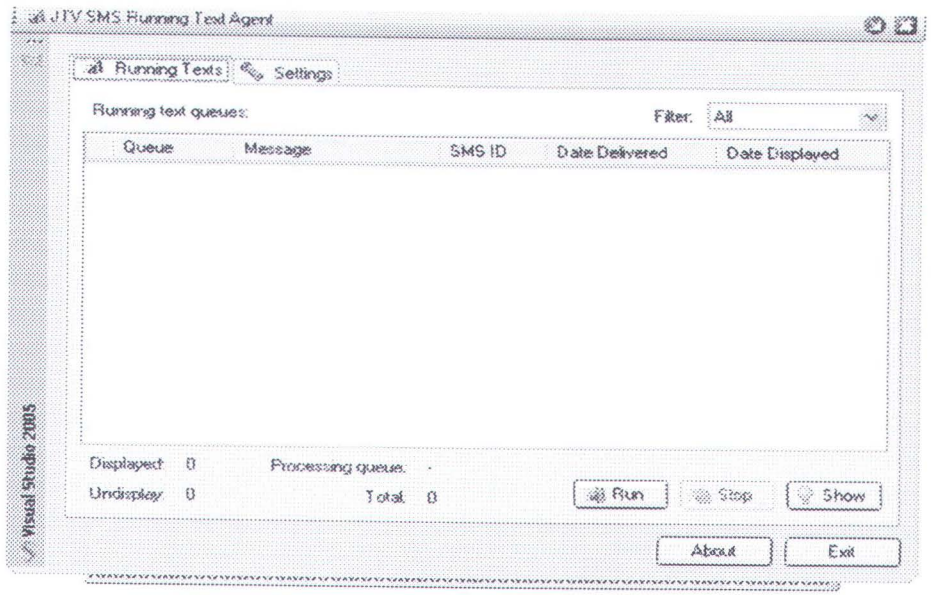
table yang telah dibuat untuk masing-masing entry yaitu *Trains, stations, Train tours.*



Gambar 3.30 Desain Antarmuka Setting Koneksi Gateway (Manajemen Koneksi sms Kereta)

Operator kereta sebelum melakukan proses pengiriman sms, terlebih dahulu melakukan setting koneksi untuk sms (SMS gateway), yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas.

3.4.3 Pemodelan Antarmuka perangkat lunak pada TV

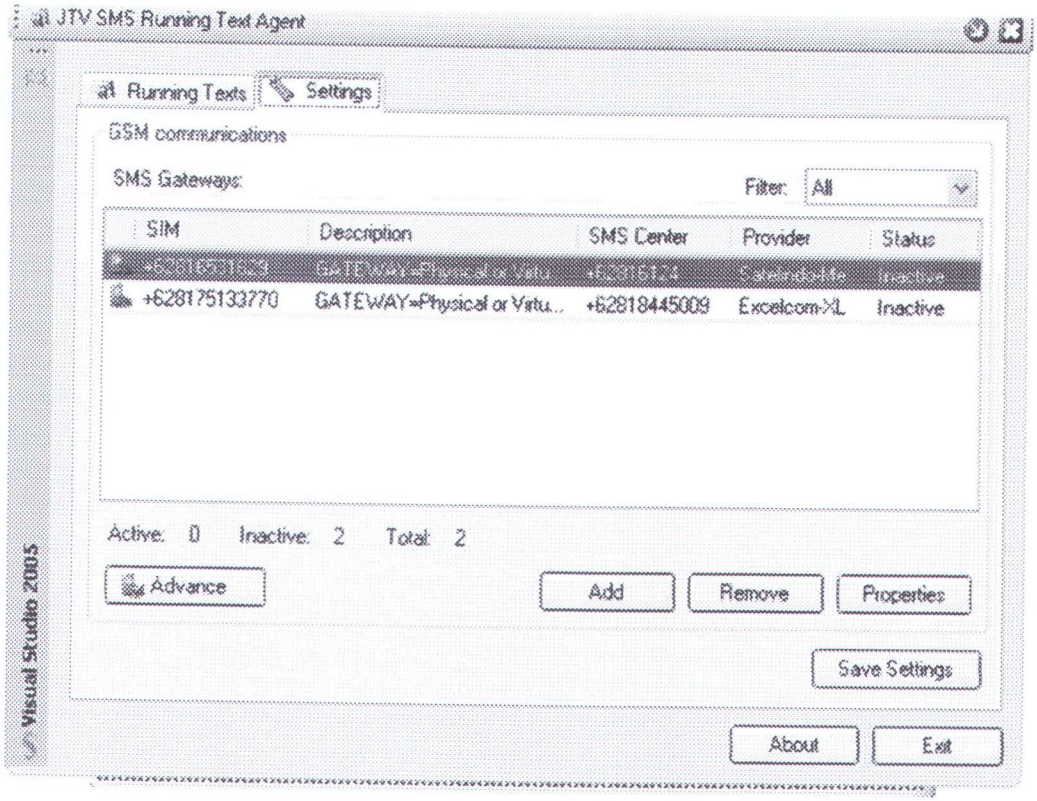


Gambar 3.31 Desain Antarmuka Penerima sms pada TV (Manajemen SMS TV)

Penerimaan sms dilakukan oleh sistem yang terdapat pada perangkat lunak dan sms yang masuk akan tertera pada *Running Text Queues.* Operator televisi bertugas untuk menampilkan sms yang masuk pada layar televisi. Untuk menampilkannya operator televisi dapat menekan tombol *run,* maka informasi data sms jadwal transportasi yang telah diterima oleh sistem yang terdapat pada perangkat lunak akan tersajikan dalam bentuk *running text.* Operator televisi juga dapat menghentikan *running text* yang sedang ditampilkan pada layar televisi, untuk tidak ditampilkan lagi yaitu dengan menekan tombol *stop* dan menyembunyikan *running text* yang sedang tayang pada layar televisi dengan menekan tombol *hide* kemudian tekan *show* untuk menampilkannya kembali.

Apabila pada proses manajemen sms TV terdapat event/delay ketika informasi telah diterima di stasiun televisi untuk ditampilkan di layar televisi sebagai *running text*, dimana delay tersebut ditangani oleh operator televisi secara manual. Operator televisi menangani queue sms yang diterima dan menyeleksi sms yang layak untuk ditayangkan. Disinilah event/delay dimana sms yang dikirimkan oleh operator kereta tidak dapat langsung ditampilkan di layar televisi, karena terdapat peran operator televisi. Ketergantungan pada operator televisi ini sangat memungkinkan sekali terjadinya human error, yaitu ketika sms yang masuk tidak ditindaklanjuti oleh operator televisi, oleh karena itu untuk meminimalisir human error ini, sms yang masuk apabila tidak ditindaklanjuti oleh operator televisi, maka dibuat persetujuan dari ketiga belah pihak, yaitu; operator kereta

api, operator pesawat, dan operator televisi terhadap pengaturan jadwal pengiriman sms, yang adalah setiap tiga puluh menit (pada menit ke 30), sehingga misalnya tidak ada sms yang masuk di stasiun televisi pada suatu jam, pihak operator televisi yang bertugas harus tetap bertanggung jawab untuk melakukan pengecekan terhadap sms yang masuk setaip menit ke 30 (satu jam dua kali).



Gambar 3.32 Desain Antarmuka koneksi sms pada TV (Manajemen Koneksi sms TV)

Operator televisi terlebih dahulu melakukan setting koneksi untuk sms (SMS gateway) sebelum proses sms terjadi, yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas. Selain itu operator televisi juga dapat melakukan penyaringan terhadap sms yang masuk, supaya tidak terjadi masuknya sms yang tidak diinginkan. Proses penyaringan terdapat pada menu *advance.*

BAB IV

IMPLEMENTASI

BAB IV

IMPLEMENTASI

Bab ini berisi tentang implementasi pembangunan perangkat lunak yang memberikan informasi pembatalan jadwal Pesawat Terbang dan Kereta Api menggunakan sms pada PT. JTV. Tahap implementasi pembangunan perangkat lunak dilakukan setelah tahap perancangan perangkat lunak sebagaimana yang telah dijelaskan di BAB III. Bagian implementasi perangkat lunak ini meliputi: implementasi data, implementasi proses, dan implementasi antarmuka.

4.1 Implementasi Data

Implementasi data meliputi implementasi desain *data model diagram* menjadi tabel. Terdapat kesamaan bentuk fisik, prinsip penyimpanan data, dan proses kerja pada basis data *FlightInformationSystem* dan *TrainInformationSystem*, untuk itu pada sub bab ini hanya akan dijelaskan implementasi data pada *TrainInformationSystem* dan *TVOnair* saja. Pada implementasi data ini, DBMS yang digunakan adalah Microsoft Access 2000.

```
CREATE TABLE Trains (
   TrainID VARCHAR ( 10 ) NOT NULL,
   TrainName VARCHAR ( 30 ),
   CONSTRAINT PK_TrainID PRIMARY KEY (TrainID)
   );
CREATE TABLE Stations (
   StationID VARCHAR ( 10 ) NOT NULL,
```

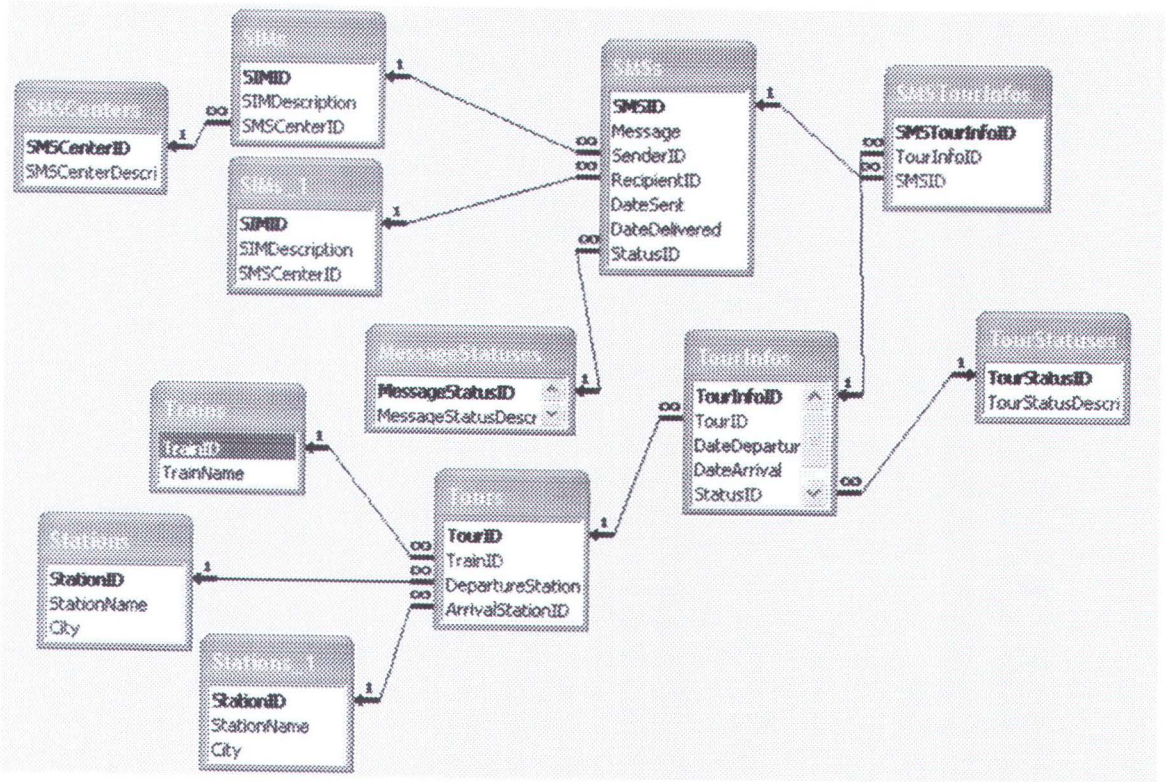
```
    StationName VARCHAR ( 30 ),
    City VARCHAR ( 30 ),
    CONSTRAINT PK_StationID PRIMARY KEY (StationID));
CREATE TABLE Tours (
    TourID VARCHAR ( 10 ) NOT NULL,
    TrainID VARCHAR ( 10 ) NOT NULL,
    DepartureStationID VARCHAR ( 10 ),
    ArrivalStationID VARCHAR ( 10 ),
    CONSTRAINT PK_Tours PRIMARY KEY (TourID));
CREATE TABLE TourInfos (
    TourInfoID Long Integer NOT NULL,
    TourID VARCHAR ( 10 ) NOT NULL,
    DateDeparture DATE,
    DateArrival DATE,
    StatusID Integer,
    Information VARCHAR ( 50 ),
    CONSTRAINT PK_TourInfos PRIMARY KEY (TourInfoID));
CREATE TABLE TourStatuses (
    TourStatusID Integer NOT NULL,
    TourStatusDescription VARCHAR ( 20 ),
    CONSTRAINT PK_TourStatuses PRIMARY KEY(TourStatusID));
CREATE TABLE SMSCenters (
    SMSCenterID VARCHAR ( 20 ) NOT NULL,
    SMSCenterDescription VARCHAR ( 50 ),
    CONSTRAINT PK_SMSCenters PRIMARY KEY(SMSCenterID));
CREATE TABLE SIMs (
    SIMID VARCHAR ( 20 ) NOT NULL,
    SIMDescription VARCHAR ( 100 ),
    SMSCenterID VARCHAR ( 20 ) NOT NULL,
    CONSTRAINT PK_SIMs PRIMARY KEY (SIMID));
CREATE TABLE MessageStatuses (
    MessageStatusID INTEGER NOT NULL,
    MessageStatusDescription VARCHAR ( 50 ),
    CONSTRAINT PK_MessageStatuses PRIMARY KEY
(MessageStatusID));
CREATE TABLE SMSs (
    SMSID LongInteger NOT NULL,
    Message Memo,
    SenderID VARCHAR ( 20 ) NOT NULL,
    RecipientID VARCHAR ( 20 ) NOT NULL,
    DateSent DATE,
    DateDelivered DATE,
    StatusID Integer NOT NULL,
    CONSTRAINT PK_SMSs PRIMARY KEY (SMSID));
CREATE TABLE SMSTourInfos (
    SMSTourInfoID Long Integer NOT NULL,
    TourInfoID Long Integer NOT NULL,
    SMSID LongInteger NOT NULL,
    CONSTRAINT PK_SMSTourInfos PRIMARY KEY (SMSTourInfoID));
```

Gambar 4.1 Skrip Database TrainInformationSystem

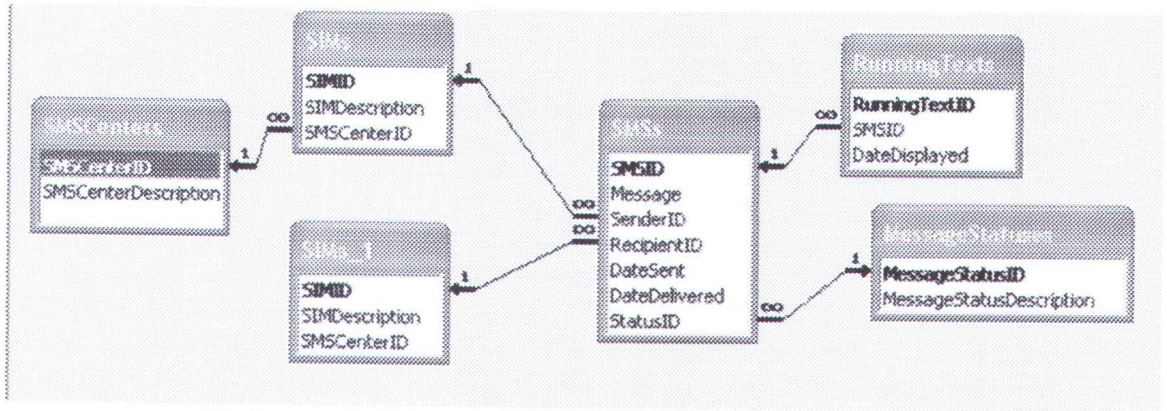
```
CREATE TABLE SMSCenters (
   SMSCenterID VARCHAR ( 20 ) NOT NULL,
   SMSCenterDescription VARCHAR ( 50 ),
   CONSTRAINT PK_SMSCenters PRIMARY KEY(SMSCenterID));
CREATE TABLE SIMs (
   SIMID VARCHAR ( 20 ) NOT NULL,
   SIMDescription VARCHAR ( 100 ),
   SMSCenterID VARCHAR ( 20 ) NOT NULL,
   CONSTRAINT PK_SIMs PRIMARY KEY (SIMID));
CREATE TABLE MessageStatuses (
   MessageStatusID INTEGER NOT NULL,
   MessageStatusDescription VARCHAR ( 50 ),
   CONSTRAINT PK_MessageStatuses PRIMARY KEY
(MessageStatusID));
CREATE TABLE SMSs (
   SMSID LongInteger NOT NULL,
   Message Memo,
   SenderID VARCHAR ( 20 ) NOT NULL,
   RecipientID VARCHAR ( 20 ) NOT NULL,
   DateSent DATE,
   DateDelivered DATE,
   StatusID Integer NOT NULL,
   CONSTRAINT PK_SMSs PRIMARY KEY (SMSID));
CREATE TABLE RunningTexts (
   RunningTextID Long Integer NOT NULL,
   SMSID LongInteger NOT NULL,
   DateDisplayed DATE,
   CONSTRAINT PK_RunningTexts PRIMARY KEY (RunningTextID));
```

Gambar 4.2 Skrip Database TVOnair

Dari sejumlah tabel basis data yang dibuat tersebut dapat dibuat hubungan antar tabel. Gambar 4.3 menunjukkan hubungan antar tabel basis data TrainInformationSystem, dan gambar 4.4 menunjukkan hubungan antar tabel basis data TVOnair.



Gambar 4.3 Hubungan Antar Tabel TrainInformationSystem



Gambar 4.4 Hubungan Antar Tabel Database TVOnair

4.2 Implementasi Proses

Implementasi proses meliputi implementasi dari setiap *usecase realization* dengan menggunakan bahasa Microsoft Visual Basic .Net 2005, dan *gateway* KylixSMS. Pada subbab ini, mengingat banyaknya proses yang

dimplementasikan, akan dibahas beberapa implementasi *usecase*. Tidak semua implementasi realisasi *usecase* akan dibahas pada buku Tugas Akhir ini.

4.2.1 Proses Koneksi SMS Gateway

Dimulai dengan mengkoneksikan handphone Siemens C35 pada komputer dengan bantuan kabel data. Kemudian mengatur bentuk komunikasi berupa *mode*, *protocol*, *port*, *baud rate*, yang sesuai dengan karakteristik dari handphone. Setelah itu proses koneksi sms gateway akan terjadi apabila tombol *connect* ditekan.

```
Private Sub btnConnect_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnConnect.Click
        If (Me.cboCommunicationBaudRate.SelectedIndex = 0) Then
            Me.prbConnectionProgress.Maximum =
Me.cboCommunicationBaudRate.Items.Count - 1
            Me.prbConnectionProgress.Value = 0
            Me.prbConnectionProgress.Visible = True
            For i As Integer =
Me.cboCommunicationBaudRate.Items.Count - 1 To 0 Step -1
                Me.cboCommunicationBaudRate.SelectedIndex = i
                If (i = 0) Then
                    Me.SetCommunicationSettingsEnabled(True)
                    Me.lblConnectionStatus.Text = "Not
connected."
                    MessageBox.Show(Me, "No match baud rate
found.", Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Exclamation)
                    Exit For
                End If

        Me.prbConnectionProgress.Value += 1
                If (Me.Connect() =
AxKylixSMS.ErrorCode.Succeed) Then
                    Me.SetDeviceNetworkInformation(True)
                    Me.lblConnectionStatus.Text = "Connected."
                    Exit For
                End If
            Next
            Me.prbConnectionProgress.Value =
Me.prbConnectionProgress.Maximum
            Me.prbConnectionProgress.Visible = False
        Else
            Dim err As AxKylixSMS.ErrorCode
```

```
            err = Me.Connect()
            If (err = AxKylixSMS.ErrorCode.Succeed) Then
                Me.SetDeviceNetworkInformation(True)
                Me.lblConnectionStatus.Text = "Connected."
            Else
                Me.SetCommunicationSettingsEnabled(True)
                Me.lblConnectionStatus.Text = "Not connected."
                MessageBox.Show(Me,
AxKylixSMS.GetErrorCodeToString(err), _
                    Me.Text, MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End If
        End If
    End Sub
```

Gambar 4.5 Penggalan Perintah untuk Mengkoneksikan sms gateway

Terdapat sebuah method yang dipanggil pada saat proses koneksi berlangsung, method tersebut bernama *connect( )*. Berikut ini adalah perintah yang terdapat pada method *connect( )*.

```
Private Function Connect() As AxKylixSMS.ErrorCode
        Me.SetDeviceNetworkInformation(False)
        Me.SetCommunicationSettingsEnabled(False)
        Me.lblConnectionStatus.Text = "Connecting..."
        Me.Refresh()

        Me.aksGateway.ConnectionMode =
Me.cboCommunicationMode.SelectedIndex + 1
        Me.aksGateway.ConnectionProtocol =
Me.cboCommunicationProtocol.SelectedIndex + 1
        Me.aksGateway.ConnectionParameter = _
            Me.cboCommunicationPort.Text + ", " +
Me.cboCommunicationBaudRate.Text

If (Me.aksGateway.Connect() = AxKylixSMS.ErrorCode.Succeed)
Then
            Return AxKylixSMS.ErrorCode.Succeed
        Else
            Return Me.aksGateway.GetLastError(0)
        End If
    End Function
```

Gambar 4.6 Penggalan Perintah untuk method *conect( )*

Setting koneksi yang telah dilakukan dapat memungkinkan untuk disimpan apabila memang diperlukan. Proses penyimpanan berupa operasi terhadap suatu file, dimana penyimpanan dilakukan terhadap sebuah file yang bernama *settings.ini*. Proses menyimpan dapat dilihat pada gambar 4.7 berikut.

```
Private Function SaveSettings() As Boolean
        Try
            Dim sw As New StreamWriter("settings.ini", False)

            sw.WriteLine("--- SMS Train Tour Infos ---")
            sw.WriteLine()
            sw.WriteLine(String.Format("    - Mode: {0}",
Me.cboCommunicationMode.Text))
            sw.WriteLine(String.Format("    - Protocol: {0}",
Me.cboCommunicationProtocol.Text))
            sw.WriteLine(String.Format("    - Port: {0}",
Me.cboCommunicationPort.Text))
            sw.WriteLine(String.Format("    - Baud rate: {0}",
Me.cboCommunicationBaudRate.Text))
            sw.WriteLine(String.Format("    - Auto delete all
reports: {0}", _
                Me.chkAutoDeleteAllReports.Checked.ToString()))
            sw.WriteLine(String.Format("    - SIM: {0}",
Me.txtNetworkSIM.Text))
            sw.WriteLine(String.Format("    - SMS Center: {0}",
Me.txtNetworkSMSCenter.Text))
            sw.Close()
            Return True
        Catch ex As Exception
            MessageBox.Show(Me, ex.Message, Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Error)
            Return False
        End Try
    End Function
```

Gambar 4.7 Penggalan Perintah untuk menyimpan setting koneksi

4.2.2 Proses Pengiriman Jadwal Kereta Api

Pada proses pengiriman jadwal kereta api langkah pertama yang harus dilakukan adalah menekan tombol *add* untuk mengisikan jadwal kereta api beserta status dan informasi yang menerangkan kondisi status tersebut, seperti terlihat dalam penggalan program pada gambar 4.8 berikut.

MILIK PERPUSTAKAAN ITS

```
Private Sub btnAddTrainTourInfo_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAddTrainTourInfo.Click
        If (Me.tblTrainTours.Count().Value = 0) Then
            MessageBox.Show(Me, "TrainTours table is empty!",
Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Error)
            Exit Sub
        End If

        Dim f As TrainTourInfoForm

        f = New TrainTourInfoForm("Add Train Tour Info", True,
"", "", Date.Now, Date.Now, "", "")
        f.ShowDialog()


Me.tblTrainTourInfos.Fill(Me.dstTrainInformationSystem.TrainTou
rInfos)
    End Sub
```

Gambar 4.8 Penggalan Perintah untuk memanggil form TrainTourInfo yang berfungsi untuk mengisikan jadwal kereta api beserta status dan informasi

Pada saat tombol *add* ditekan maka form *TrainTourInfo* akan dipanggil, form ini berfungsi untuk mengisi data jadwal kereta api, status yang terjadi, dan informasinya. Langkah awal adalah memilih id dari tour yang terdapat pada menu combo box *Tour ID*, Dimana data-data mengenai id tour yang dipilih tersebut diambil dari data yang telah tersimpan di dalam table *Tours*. Selanjutnya adalah memilih id dari status yang terdapat pada menu combo box *status*, id status yang dipilih diambil dari id status yang telah tersimpan di dalam table *TourStatuses*. Terdapat dua Penggalan perintah yang digolongkan berdasarkan fungsinya, yaitu *RefreshTours()* yang berfungsi untuk pengambilan data tour atau jadwal kereta api dan *RefreshTourStatuses()* yang berfungsi untuk pengambilan data status yang

terjadi pada kereta api. Penggalan perintah tersebut diterangkan pada gambar 4.9 berikut.

```
Private Sub RefreshTours()
        Dim tbl As TrainInformationSystemDataSet.ToursDataTable
        tbl = New
TrainInformationSystemDataSetTableAdapters.ToursTableAdapter().GetData
ByTourID(Me.cboTourID.Text)
        If (tbl.Count > 0) Then
            Me.txtTrainID.Text = tbl(0).TrainID
            Me.txtDepartureStationID.Text = tbl(0).DepartureStationID
            Me.txtArrivalStationID.Text = tbl(0).ArrivalStationID
        Else
            Me.txtTrainID.Text = ""
            Me.txtDepartureStationID.Text = ""
            Me.txtArrivalStationID.Text = ""
        End If

        Me.RefreshTrains()
        Me.RefreshDepartureStations()
        Me.RefreshArrivalStations()
        Me.RefreshTourStatuses()
    End Sub


Private Sub RefreshTourStatuses()
        Dim tbl As TrainInformationSystemDataSet.TourStatusesDataTable
        tbl = New
TrainInformationSystemDataSetTableAdapters.TourStatusesTableAdapter()_
            .GetDataByStatusID(Me.cboStatusID.Text)
        If (tbl.Count > 0) Then
            Me.txtStatus.Text = tbl(0).TourStatusDescription
        Else
            Me.txtStatus.Text = ""
        End If
    End Sub
```

Gambar 4.9 Penggalan Perintah untuk pengambilan data tour atau jadwal kereta api yaitu *RefreshTours()* dan pengambilan data status yang terjadi pada kereta api yaitu *RefreshTourStatuses()*

Apabila proses pengisian jadwal kereta api beserta status dan informasinya telah selesai maka tombol *OK* dalam form *TrainTourInfo* ditekan, kemudian data jadwal kereta api yang lengkap dengan status dan informasi akan tersimpan di dalam table *TourInfos* dan untuk menampilkan data tersebut akan diambil data yang berasal dari query *TrainTourInfos* kemudian akan ditampilkan pada data grid view Train tour infos. Proses penampilan data tersebut dilakukan oleh penggalan

perintah pada gambar 4.8. Dan untuk proses penyimpanan data pada table *TourInfos* sesuai dengan penggalan perintah pada gambar 4.10 dibawah ini

```
Private Sub btnOK_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnOK.Click
        Dim tbl As New
TrainInformationSystemDataSetTableAdapters.TourInfosTableAdapte
r()

        If (Me.newTrainTourInfo) Then
            Try
                tbl.Insert(Me.cboTourID.Text,
Me.txtDateDeparture.Text, Me.txtDateArrival.Text, _
                Me.cboStatusID.Text, Me.txtInformation.Text)
                Me.Close()
            Catch ex As Exception
                MessageBox.Show(Me, ex.Message, Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        Else
            Try
                tbl.Update(Me.cboTourID.Text,
Me.txtDateDeparture.Text, Me.txtDateArrival.Text, _
                    Me.cboStatusID.Text,
Me.txtInformation.Text, Me.oldTourInfoID, Me.oldTourID, _
                    Me.oldDateDeparture, Me.oldDateArrival,
Me.oldStatusID, Me.oldInformation)
                Me.Close()
            Catch ex As Exception
                MessageBox.Show(Me, ex.Message, Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If
    End Sub
```

Gambar 4.10 Penggalan Perintah proses yang terjadi pada saat button *OK* yang terdapat pada form TrainTourInfo ditekan

Dalam suatu pengiriman sms jadwal kereta api pasti dibutuhkan nomor tujuan untuk pengiriman sms. Oleh sebab itu dibuat combo box untuk menu *recipient* yang berfungsi untuk pemilihan nomor sms tujuan, dimana data nomor tujuan sms diambil dari table *SIMs* yang telah berisikan banyak pilihan data nomor tujuan sms. Penggalan perintah pengambilan data *recipient* pada table *SIMs* dapat dilihat pada gambar 4.11 berikut.

```
Dim tbl As New TrainInformationSystemDataSet.SIMsDataTable()

        tbl = New
TrainInformationSystemDataSetTableAdapters.SIMsTableAdapter().GetData()

   For i As Integer = 0 To tbl.Count - 1
     If (tbl(i).SIMDescription.IndexOf("TV;") >= 0) Then
     Me.cboSMSTrainTouInfosRecipient.Items.Add( _
     tbl(i).SIMDescription.Replace("TV;", "") + String.Format("({0})", _
                    tbl(i).SIMID))
     End If

   Next

   If (Me.cboSMSTrainTouInfosRecipient.Items.Count > 0) Then
         Me.cboSMSTrainTouInfosRecipient.SelectedIndex = 0
   End If
```

Gambar 4.11 Penggalan Perintah pengambilan data *recipient* pada table *SIMs*

Langkah selanjutnya adalah menekan button *send* untuk pengiriman sms jadwal kereta api beserta status dan informasi mengenai status yang terjadi pada kereta api. Setiap pengiriman sms jadwal kereta api di batasi sampai dengan maksimal 160 karakter agar sms tidak terpecah menjadi dua atau lebih untuk setiap data sms jadwal kereta api yang dikirim sesuai dengan bentuk pengiriman sms yang dilakukan oleh handphone pada umumnya untuk sekali pengiriman sms. Penggalan perintah pengiriman sms jadwal kereta api lengkap dengan status dan informasinya dapat dilihat pada gambar 4.12 berikut.

```
Private Sub btnSendSMSTrainTourInfos_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSendSMSTrainTourInfos.Click
        Dim tourInfoID As Integer
        Dim recipient As String
        Dim train As String
        Dim departureStation As String
        Dim dateDeparture As Date
        Dim arrivalStation As String
        Dim dateArrival As Date
        Dim status As String
        Dim information As String
        Dim message As String
        Dim temp As String
        Dim tbl As New TrainInformationSystemDataSet.SMSsDataTable()
        Dim tblSMSs As New
TrainInformationSystemDataSetTableAdapters.SMSsTableAdapter()
        Dim tblSMSTourInfos As New
TrainInformationSystemDataSetTableAdapters.SMSTourInfosTableAdapter



For i As Integer = 0 To Me.dgvTrainTourInfos.SelectedRows.Count - 1

Dim j As Integer
 j = Me.dgvTrainTourInfos.SelectedRows(i).Index
 tourInfoID = Me.dgvTrainTourInfos.Item( _
              Me.clmTrainTourInfosTourInfoID.Index, j).Value

 temp = (Me.cboSMSTrainTouInfosRecipient.Text).Replace("TV;", "")

 recipient = (temp.Substring(temp.IndexOf("(") + 1)).Replace(")", "")

 train=Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosTrain.Index,j).Value

 departureStation = _
Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosDepartureStation.Index,j).Value

 dateDeparture = _
Convert.ToDateTime(Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosDateDeparture
.Index, j).Value)

 arrivalStation = _
Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosArrivalStation.Index, j).Value

 dateArrival = _
Convert.ToDateTime(Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosDateArrival.
Index, j).Value)

 status = Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosStatus.Index,j).Value

 information = _
Me.dgvTrainTourInfos.Item(Me.clmTrainTourInfosInformation.Index, j).Value

 message = String.Format("Kereta Api {0} Asal {1} ({2}) Tujuan {3} ({4}) {5} {6}", train,
departureStation, dateDeparture.ToShortDateString + " " + _
dateDeparture.ToShortTimeString, arrivalStation,dateArrival.ToShortDateString
+ " " + dateArrival.ToShortTimeString, status, information)
```

```
            If (message.Length > 160) Then
                 message = message.Substring(0, 160)
            End If

             Me.aksGateway.RequestSendReport = 1
             Me.aksGateway.SendInterval = 1
             Me.aksGateway.SendRetryTimes = 1
             Me.aksGateway.SendTimeout = 15
             Me.aksGateway.SMSValidity = 6
             Me.messages(Me.currentMessage).Referrence = _
             Me.aksGateway.SendSMS(recipient, message)

 If (Me.messages(Me.currentMessage).Referrence > 0) Then
    blSMSs.Insert(message, Me.oldSIM, recipient, Date.Now, Nothing, 2)

 Else
   tblSMSs.Insert(message, Me.oldSIM, recipient, Nothing, Nothing, 1)

 End If

tbl = tblSMSs.GetData()
Me.messages(Me.currentMessage).SMSID = tbl(tbl.Count - 1).SMSID
tblSMSTourInfos.Insert(tourInfoID,Me.messages(Me.currentMessage).SMSID)
Me.currentMessage += 1
Me.RefreshSMSTourInfos()

Next

End Sub
```

Gambar 4.12 Penggalan Perintah pengiriman sms jadwal kereta api

Seperti yang terlihat pada gambar 4.12, terjadi pemanggilan terhadap procedure *RefreshSMSTourInfos()*. Dimana dalam Procedure *RefreshSMSTourInfos* ini akan menampilkan jadwal kereta setelah melewati proses pengiriman sms dan juga akan dicantumkan status report sms dari jadwal kereta api yang bersangkutan. Jadwal sms ditampilkan pada data grid view *SMS train tour infos* dimana data tersebut diambil dari query *SMSTrainTourInfos* dan akan ditampilkan berdasarkan penggolongan status report sms yaitu *sent, sent failed, delivery failed, delivered.* Lebih detail dapat dilihat pada gambar 4.13 berikut.

```
Private Sub RefreshSMSTourInfos()
        Select Case (Me.cboSMSTrainTourInfosFilter.SelectedIndex)
            Case 0

Me.tblSMSTrainTourInfos.Fill(Me.dstTrainInformationSystem.SMSTrainTourInfos)
            Case Else
                Me.tblSMSTrainTourInfos.FillByMessageStatusID( _
                    Me.dstTrainInformationSystem.SMSTrainTourInfos, _
                    Me.cboSMSTrainTourInfosFilter.SelectedIndex)
        End Select

 Dim tblSMSs As New TrainInformationSystemDataSetTableAdapters.SMSsTableAdapter()

   If (tblSMSs.CountByStatusID(1).HasValue) Then
    Me.lblTotalSMSSentFailed.Text = tblSMSs.CountByStatusID(1).Value.ToString

   Else
    Me.lblTotalSMSSentFailed.Text = "0"

   End If

   If (tblSMSs.CountByStatusID(2).HasValue) Then
     Me.lblTotalSMSSent.Text = tblSMSs.CountByStatusID(2).Value.ToString

   Else
     Me.lblTotalSMSSent.Text = "0"

   End If

If (tblSMSs.CountByStatusID(3).HasValue) Then
 Me.lblTotalSMSDeliveryFailed.Text =tblSMSs.CountByStatusID(3).Value.ToString

Else
 Me.lblTotalSMSDeliveryFailed .Text = "0"

End If

If (tblSMSs.CountByStatusID(4).HasValue) Then
  Me.lblTotalSMSDelivered.Text = tblSMSs.CountByStatusID(4).Value.ToString

Else
  Me.lblTotalSMSDelivered.Text = "0"

End If

If (tblSMSs.Count().HasValue) Then
  Me.lblTotalSMSTrainTourInfos.Text = tblSMSs.Count().Value.ToString

Else
  Me.lblTotalSMSTrainTourInfos.Text = "0"

End If

End Sub
```

Gambar 4.13 Penggalan Perintah untuk menampilkan jadwal kereta api yang telah mengalami proses pengiriman sms dilengkapi dengan status report sms

4.2.3 Proses Koneksi SMS TV

Dimulai dengan mengkoneksikan handphone Siemens C35 pada komputer dengan bantuan kabel data atau koneksi dengan menggunakan wavecom. Kemudian memilih menu settings dan menyortir satu data gateway yang ditampilkan di data grid view *SMS Gateways*. Penggalan perintah untuk kondisi sortir gateway terdapat pada gambar 4.14 berikut.

```
Private Sub dgvSMSGateways_SelectionChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
dgvSMSGateways.SelectionChanged
        If (Me.dgvSMSGateways.SelectedRows.Count > 0) Then
            Me.btnRemoveGateways.Enabled = True
            Me.mniRemoveGateways.Enabled = True
            If (Me.dgvSMSGateways.SelectedRows.Count = 1) Then
                Me.btnGatewayProperties.Enabled = True
                Me.mniGatewayProperties.Enabled = True
            Else
                Me.btnGatewayProperties.Enabled = False
                Me.mniGatewayProperties.Enabled = False
            End If
        Else
            Me.btnRemoveGateways.Enabled = False
            Me.mniRemoveGateways.Enabled = False
            Me.btnGatewayProperties.Enabled = False
            Me.mniGatewayProperties.Enabled = False
        End If
    End Sub
```

Gambar 4.14 Perintah sortir gateway

Setelah proses sortir salah satu data gateway yang ditampilkan di data grid view dilakukan, langkah selanjutnya adalah memilih proses yang akan dilakukan terhadap gateway, berupa proses penghapusan data gateway (*Remove gateways*) atau koneksikan gateway. Apabila memilih untuk menjalankan proses koneksi terhadap gateway, maka tekan button *properties* dan setelah itu akan memanggil form *GatewayProperties*. Berikut perintah klik button properties yang didalamnya

terdapat perintah untuk memanggil form *GatewayProperties* yang terdapat pada gambar 4.15.

```
'Event tombol Properties Click
    Private Sub btnGatewayProperties_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnGatewayProperties.Click, mniGatewayProperties.Click
        Dim aks As AxKYLIXSMSLib.AxKylixSMS
        Dim gw As Gateway

        gw =
Me.GetGateway(Me.dgvSMSGateways.SelectedRows(0).Index)
        Select Case (gw.Status)
            Case Gateway.GatewayStatus.Inactive
                aks = Me.GetAksGateway("")
            Case Gateway.GatewayStatus.Active
                aks = Me.GetAksGateway(gw.SIM)
            Case Else
                aks = Nothing
        End Select

        If (Not aks Is Nothing) Then
            Dim f As GatewayPropertiesForm

            f = New GatewayPropertiesForm(Me, gw,
Convert.ToInt32( _
                aks.Name.Replace("aksGateway", "")), False)

            Me.timCheckGatewayConnection.Enabled = False
            f.ShowDialog()
            Me.timCheckGatewayConnection.Enabled = True
            Me.RefreshGateways()
        Else
            MessageBox.Show(Me, String.Format("You can only
activate {0} gateway(s). Please inactivate " + _
                "at least one active gateway" +
ControlChars.NewLine + " before set up the other.", _
                MAX_ACTIVE_GATEWAYS.ToString()), Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub
```

Gambar 4.15 Penggalan Perintah klik button properties

Pada form *GatewayProperties* pilih menu settings dan untuk mengkoneksikan gateway, tekan button *connect*.

```
Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) _
        Handles btnConnect.Click

        If (Me.cboCommunicationBaudRate.SelectedIndex = 0) Then
            Me.prbConnectionProgress.Maximum = _
                 Me.cboCommunicationBaudRate.Items.Count - 1
            Me.prbConnectionProgress.Value = 0
            Me.prbConnectionProgress.Visible = True

           For i As Integer = Me.cboCommunicationBaudRate.Items.Count
                 - 1 To 0 Step -1
                Me.cboCommunicationBaudRate.SelectedIndex = i
                If (i = 0) Then
                  Me.SetCommunicationSettingsEnabled(True)
                  Me.lblConnectionStatus.Text = "Not connected."
                  MessageBox.Show(Me, "No match baud rate found.", _
                  Me.Text, MessageBoxButtons.OK,MessageBoxIcon.Exclamation)
              Exit For

                End If

                Me.prbConnectionProgress.Value += 1

               If (Me.Connect() = AxKylixSMS.ErrorCode.Succeed) Then
                   Me.SetDeviceNetworkInformation(True)
                   Me.lblConnectionStatus.Text = "Connected."

      Me.tbcGatewayProperties.SelectTab(Me.tbpGatewayPropertiesGeneral)
             Exit For

              End If

          Next
           Me.prbConnectionProgress.Value = Me.prbConnectionProgress.Maximum
           Me.prbConnectionProgress.Visible = False

         Else
             Dim err As AxKylixSMS.ErrorCode
            err = Me.Connect()

          If (err = AxKylixSMS.ErrorCode.Succeed) Then
              Me.SetDeviceNetworkInformation(True)
              Me.lblConnectionStatus.Text = "Connected."

      Me.tbcGatewayProperties.SelectTab(Me.tbpGatewayPropertiesGeneral)

          Else
                Me.SetCommunicationSettingsEnabled(True)
                Me.lblConnectionStatus.Text = "Not connected."
                MessageBox.Show(Me, AxKylixSMS.GetErrorCodeToString(err), _
                    Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Error)
           End If

        End If
    End Sub
```

Gambar 4.16 Penggalan Perintah untuk koneksi sms gateway

Di dalam prosedure gambar 4.16 terdapat pemangilan suatu method yaitu *connect( )*. Dan perintah yang terdapat di dalam method *connect( )* tersebut akan dijelaskan pada gambar 4.17 dibawah ini.

```
Private Function Connect() As AxKylixSMS.ErrorCode
        Me.SetDeviceNetworkInformation(False)
        Me.SetCommunicationSettingsEnabled(False)
        Me.lblConnectionStatus.Text = "Connecting..."
        Me.Refresh()

Me.myParentForm.GetAksGateways(Me.index).ConnectionMode = _
            Me.cboCommunicationMode.SelectedIndex + 1

Me.myParentForm.GetAksGateways(Me.index).ConnectionProtocol = _
            Me.cboCommunicationProtocol.SelectedIndex + 1

Me.myParentForm.GetAksGateways(Me.index).ConnectionParameter = _
Me.cboCommunicationPort.Text + ", " + Me.cboCommunicationBaudRate.Text

     If (Me.myParentForm.GetAksGateways(Me.index).Connect() = _
            AxKylixSMS.ErrorCode.Succeed) Then
         Return AxKylixSMS.ErrorCode.Succeed

     Else
     Return   Me.myParentForm.GetAksGateways(Me.index).GetLastError(0)

     End If

End Function
```

Gambar 4.17 Penggalan Perintah pada method *connect ( )*

Pada saat proses koneksi gateway berhasil default yang terjadi adalah menjalankan proses pada menu *general*, dimana menu ini terdapat dalam satu form yang sama dengan menu settings, yaitu form *GatewayProperties*. Dan langkah selanjutnya yang perlu diperhatikan adalah merubah status koneksi gateway tersebut menjadi active, supaya gateway tersebut dapat terpakai. Perubahan dilakukan pada combo box *GatewayStatus* dan proses perubahan status tersebut tertera dalam penggalan perintah yang terdapat pada gambar 4.15.

Apabila status telah berubah menjadi active maka untuk mengakhiri proses koneksi adalah menekan button OK. Penggalan perintah proses kerja yang terjadi ketika button OK ditekan tersebut akan dijelaskan pada gambar 4.18 dibawah ini.

```
Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) _
        Handles btnOK.Click

        If (Me.txtNetworkSIM.Text = "") Then
     Me.tbcGatewayProperties.SelectTab(Me.tbpGatewayPropertiesGeneral)
            Me.tltBaloonToolTip.ToolTipIcon = ToolTipIcon.Error
            Me.tltBaloonToolTip.ToolTipTitle = "SIM is empty!"

            If (Me.tltBaloonToolTip.Tag <> Me.txtNetworkSIM.Name) Then
                Me.tltBaloonToolTip.Tag = Me.txtNetworkSIM.Name
                Me.tltBaloonToolTip.Show("", Me.txtNetworkSIM)
            End If

   Me.tltBaloonToolTip.Show("Please fill this first before confirm.",
   Me.txtNetworkSIM,5000)

       Else
            If (Me.cboCommunicationBaudRate.SelectedIndex = 0) Then
                Me.cboCommunicationBaudRate.SelectedIndex = 7
            End If

Dim gw As New Gateway()

            gw.SIM = Me.txtNetworkSIM.Text
            gw.SMSCenter = Me.txtNetworkSMSCenter.Text
            gw.Description = Me.txtGatewayDescription.Text
            gw.SetModeFromString(Me.cboCommunicationMode.Text)
            gw.SetProtocolFromString(Me.cboCommunicationProtocol.Text)
            gw.SetPortFromString(Me.cboCommunicationPort.Text)
            gw.SetBaudRateFromString(Me.cboCommunicationBaudRate.Text)

If (Me.newGateway) Then
TVOnairDatabase.SIMs.Insert(gw.SIM, "GATEWAY=" + gw.GetModeToString()
+ ";" + gw.GetProtocolToString() + ";" + gw.GetPortToString() + ";" +
gw.GetBaudRateToString() + ";" + gw.Description, gw.SMSCenter)


Else
TVOnairDatabase.SIMs.Update(gw.SIM, "GATEWAY=" + gw.GetModeToString()
+ ";" + gw.GetProtocolToString() + ";" + gw.GetPortToString() + ";" +
gw.GetBaudRateToString() + ";" + gw.Description, gw.SMSCenter, _
Me.gateway.SIM, "GATEWAY=" + Me.gateway.GetModeToString() + ";" + _
Me.gateway.GetProtocolToString() + ";" + Me.gateway.GetPortToString()
+ ";" + Me.gateway.GetBaudRateToString() + ";" +
Me.gateway.Description, Me.gateway.SMSCenter)
```

```
gw.SetStatusFromString(Me.cboGatewayStatus.Text)
Select Case (gw.Status)
   Case TVSMSRunningTextAgent.Gateway.GatewayStatus.Inactive
        gw = New Gateway()
        Me.disconnectOnCancel = True

   Case TVSMSRunningTextAgent.Gateway.GatewayStatus.Active

Me.myParentForm.GetAksGateways(Me.index).SetSMSCNumber(gw.SMSCenter)
Me.disconnectOnCancel = False

End Select

        Me.myParentForm.SetGateways(Me.index, gw)
     End If

      Me.Close()
   End If
End Sub
```

Gambar 4.18 Penggalan Perintah pada saat button OK ditekan

4.2.4 Proses Manajemen SMS TV

Gateway yang telah active menunjukan bahwa proses sms sudah dapat dilakukan. Pada manajemen sms TV proses sms yang dilakukan adalah menerima jadwal sms transportasi yang masuk. Proses penerimaan sms akan dilakukan oleh function *NewSMSReceived( )* dimana penggalan programnya akan dijelaskan pada gambar 4.19.

```
Private Sub aksGateway_NewSMSReceived(ByVal sender As System.Object,
ByVal e As AxKYLIXSMSLib._DKylixSMSEvents_NewSMSReceivedEvent) Handles
aksGateway0.NewSMSReceived

   Me.SMSRunningTextFiltering(e.text.Replace("www.kylixsoft.com", ""),
   e.number,CType(sender, AxKYLIXSMSLib.AxKylixSMS).Tag,e.time,
            Date.Now)
End Sub
```

Gambar 4.19 Penggalan Perintah *NewSMSReceived( )*

Di dalam function *NewSMSReceived()*, terdapat pemanggilan sebuah fungsi yang bernama *SMSRunningTextFiltering()*, dimana Proses ini berfungsi untuk melakukan penyaringan terhadap sms yang masuk berdasarkan *Sender Prefix, Message Prefix, Description Prefix*. Data sms yang masuk walaupun dilakukan setting penyaringan sms atau tidak, akan dilakukan penyimpanan data sms terhadap sebuah tabel *SMSs*. Setelah itu proses yang terjadi adalah pemanggilan form *frmRunningText* dimana form ini berfungsi untuk menampilkan data sms transportasi yang masuk dalam bentuk text berjalan atau *Running Text*. Di dalam *frmRunningText* dilakukan pemanggilan terhadap sebuah fungsi yang bernama *AddMessage()*. Perintah terakhir didalam function *SMSRunningTextFiltering()* adalah pemanggilan fungsi *RefreshRunningTextQueues()*. Selengkapnya dapat dilihat pada gambar 4.20 berikut.

```
Private Function SMSRunningTextFiltering(ByVal message As String,
ByVal sender As String, ByVal recipient As String, ByVal dateSent As
Date, ByVal dateDelivered As Date)

        If (Me.senderPrefix <> "") Then
            If (sender.IndexOf(Me.senderPrefix) < 0) Then
                Return False
            End If
        End If

        If (Me.messagePrefix <> "") Then
            If (message.IndexOf(Me.messagePrefix) < 0) Then
                Return False
            End If
        End If

         If (Me.descriptionPrefix <> "") Then

        Dim dtb As New _
       TVOnairDataSet.SMSRunningTextDescriptionFilterDataTable()

      dtb = New
TVOnairDataSetTableAdapters.SMSRunningTextDescriptionFilterTableAdapte
r().GetData(Me.descriptionPrefix + "%")

            If (dtb.Count > 0) Then
                For i As Integer = 0 To dtb.Count - 1
                    If (sender = dtb(i).Sender) Then
                        Exit For
                    End If

                    If (i = dtb.Count - 1) Then
                        Return False
                    End If
                Next
            Else
                Return False
            End If
        End If

TVOnairDatabase.SMSs.Insert(message, sender, recipient, dateSent,
dateDelivered, Nothing)

Dim tblSMSs As New TVOnairDataSet.SMSsDataTable()
tblSMSs = New TVOnairDataSetTableAdapters.SMSsTableAdapter().GetData()
TVOnairDatabase.RunningTexts.Insert(tblSMSs(tblSMSs.Count - 1).SMSID,
Nothing)

Dim tblRunningTexts As New TVOnairDataSet.RunningTextsDataTable
tblRunningTexts = New _
TVOnairDataSetTableAdapters.RunningTextsTableAdapter().GetData()

Me.frmRunningText.AddMessage(tblRunningTexts(tblRunningTexts. _
Count - 1).RunningTextID, message)
Me.RefreshRunningTextQueues()
Me.dgvRunningTextQueues.FirstDisplayedScrollingRowIndex = _
Me.dgvRunningTextQueues.RowCount - 1

    Return True
End Function
```

Gambar 4.20 Penggalan Perintah *SMSRunningTextFiltering()*

```
Public Sub AddMessage(ByVal queue As Integer, ByVal message As
String)
        Me.queues(Me.queueCount).Queue = queue
        Me.queues(Me.queueCount).message = message
        Me.queueCount += 1

        If (Not Me.timRunningText.Enabled And
Me.messages(0).Left >= Me.Width) Then
            Me.UpdateMessage()
        End If
```

Gambar 4.21 Penggalan Perintah *AddMessage()* yang terdapat didalam *frmRunningText*

```
Private Sub RefreshRunningTextQueues()
        Select Case
(Me.cboRunningTextQueuesFilter.SelectedIndex)
            Case 0

Me.tblRunningTextQueues.Fill(Me.dstTVOnair.RunningTextQueues)
            Case 1

Me.tblRunningTextQueues.FillByDateDisplayedIsNotNull( _
                    Me.dstTVOnair.RunningTextQueues)
            Case 2

Me.tblRunningTextQueues.FillByDateDisplayedIsNull(Me.dstTVOnair
.RunningTextQueues)
        End Select
End Sub
```

Gambar 4.22 Penggalan Perintah *RefreshRunningTextQueues()*

Penggalan perintah pada gambar 4.22 diatas berfungsi untuk menampilkan data sms yang masuk pada data grid view.

Untuk menjalankan running text adalah dengan menekan button *Run*, maka proses kerja yang dilakukan adalah memanggil function *btnRunPauseRunningText_Click()* sesuai dengan yang terlihat pada gambar 4.23 dibawah ini.

```
Private Sub btnRunPauseRunningText_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnRunPauseRunningText.Click
        If (Me.btnRunPauseRunningText.Text = "Run") Then
            Me.btnRunPauseRunningText.Text = "Pause"
            Me.btnRunPauseRunningText.Image =
My.Resources.Resources.running_text_pause

            If (Me.frmRunningText.Opacity = 0) Then
                Me.btnShowHideRunningText.Text = "Hide"
                Me.btnShowHideRunningText.Image =
My.Resources.Resources.form_hide
                Me.frmRunningText.Opacity = 1
            End If

            Me.frmRunningText.RunningTextRun()
        ElseIf (Me.btnRunPauseRunningText.Text = "Pause") Then
            Me.btnRunPauseRunningText.Text = "Run"
            Me.btnRunPauseRunningText.Image =
My.Resources.Resources.running_text_run

            Me.frmRunningText.RunningTextPause()
        End If

        Me.btnStopRunningText.Enabled = True
End Sub
```

Gambar 4.23 Penggalan Perintah *btnRunPauseRunningText_Click()*

Didalam function *btnRunPauseRunningText_Click()* terdapat pemanggilan form *frmRunningText* yang kemudian memanggil fungsi untuk menjalankan running text yaitu *RunningTextRun()*, seperti yang terlihat pada gambar 4.24 dibawah ini

```
Public Sub RunningTextRun()
        Me.timRunningText.Enabled = True
    End Sub

Public Sub RunningTextPause()
        Me.timRunningText.Enabled = False
    End Sub
```

Gambar 4.24 Penggalan Perintah *RunningTextRun()*

Pada function *RunningTextRun()* akan memanggil sebuah fungsi timer yang berfungsi untuk mengatur kecepatan gerak *running text*, fungsi tersebut

bernama *timRunningText_Tick()*, untuk lebih jelasnya dapat dilihat pada gambar 4.25 dibawah ini.

```
Private Sub timRunningText_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) _
        Handles timRunningText.Tick
If (Me.currentMessageIndex < Me.messages.Length - 1 And _
    Me.messages(Me.currentMessageIndex).Text <> "" And _
    Me.logos(Me.currentMessageIndex).Right + _
                RUNNING_TEXT_SPACE <= Me.Width) Then
   If (Me.messages(Me.currentMessageIndex).Tag <> Nothing) Then

Me.myParentForm.UpdateRunningTextQueues(Me.messages(Me.currentMessageI
ndex).Tag)
Me.messages(Me.currentMessageIndex).Tag = Nothing
   End If
   Me.currentMessageIndex += 1
End If

If (Me.logos(Me.logos.Length - 1).Right + RUNNING_TEXT_SPACE <= 0)
            Then
      Me.UpdateMessage()
End If


    For i As Integer = 0 To Me.messages.Length - 1
        Me.messages(i).Left -= Me.mMotionPixel
        Me.logos(i).Left -= Me.mMotionPixel
    Next

End Sub
```

Gambar 4.25 Penggalan Perintah *timRunningText_Tick()*

Untuk menghentikan jalannya *running text* maka ditekan button *Stop*,dimana akan terjadi sebuah proses sesuai dengan penggalan program yang terdapat pada gambar 4.26 dibawah ini.

```
Private Sub btnStopRunningText_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnStopRunningText.Click
        If (Me.btnRunPauseRunningText.Text = "Pause") Then
            Me.btnRunPauseRunningText.Text = "Run"
            Me.btnRunPauseRunningText.Image =
My.Resources.Resources.running_text_run

            Me.frmRunningText.RunningTextPause()
        End If

        Me.frmRunningText.RunningTextRestart()
        Me.btnStopRunningText.Enabled = False
    End Sub
```

Gambar 4.26 Penggalan Perintah *btnStopRunningText_Click()*

Didalam *btnStopRunningText_Click()* terdapat pemanggilan dua function yaitu *RunningTextPause() dan RunningTextRestart()* dimana kedua fungsi ini terdapat dalam form *frmRunningText.* Fungsi *RunningTextPause()* akan memanggil fungsi *timRunningText_Tick()* (gambar 4.25), yang berfungsi sebagai timer. Dan fungsi *RunningTextRestart()* (gambar 4.27) akan memanggil fungsi *ArrangeMessages( )* (gambar 4.28), yang berfungsi untuk menyusun jumlah sms yang ditampilkan pada form *frmRunningText.*

```
Public Sub RunningTextRestart()
        Me.ArrangeMessages()
    End Sub
```

Gambar 4.27 Penggalan Perintah *RunningTextRestart()*

```
Private Sub ArrangeMessages()
        For i As Integer = 0 To Me.messages.Length - 1
            If (Me.messages(i).Text = "" Or i =
Me.messages.Length - 1) Then
                Me.logos(i).Width = 0
            Else
                Me.logos(i).Width = 40
            End If

            If (i = 0) Then
                Me.messages(i).Left = Me.Width
            Else
                Me.messages(i).Left = Me.logos(i - 1).Right
                If (Me.messages(i).Text <> "") Then
                    Me.messages(i).Left += RUNNING_TEXT_SPACE
                End If

            End If

            Me.logos(i).Left = Me.messages(i).Right
            If (Me.messages(i).Text <> "") Then
                Me.logos(i).Left += RUNNING_TEXT_SPACE
            End If
        Next
    End Sub
```

Gambar 4.28 Penggalan Perintah *ArrangeMessages()*

Untuk menampilkan atau menyembunyikan running text dengan menekan button *Show*, dimana akan terjadi sebuah proses sesuai dengan penggalan program yang terdapat pada gambar 4.29 dibawah ini.

```
    Private Sub btnShowHideRunningText_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnShowHideRunningText.Click
        If (Me.btnShowHideRunningText.Text = "Show") Then
            Me.btnShowHideRunningText.Text = "Hide"
            Me.btnShowHideRunningText.Image =
My.Resources.Resources.form_hide

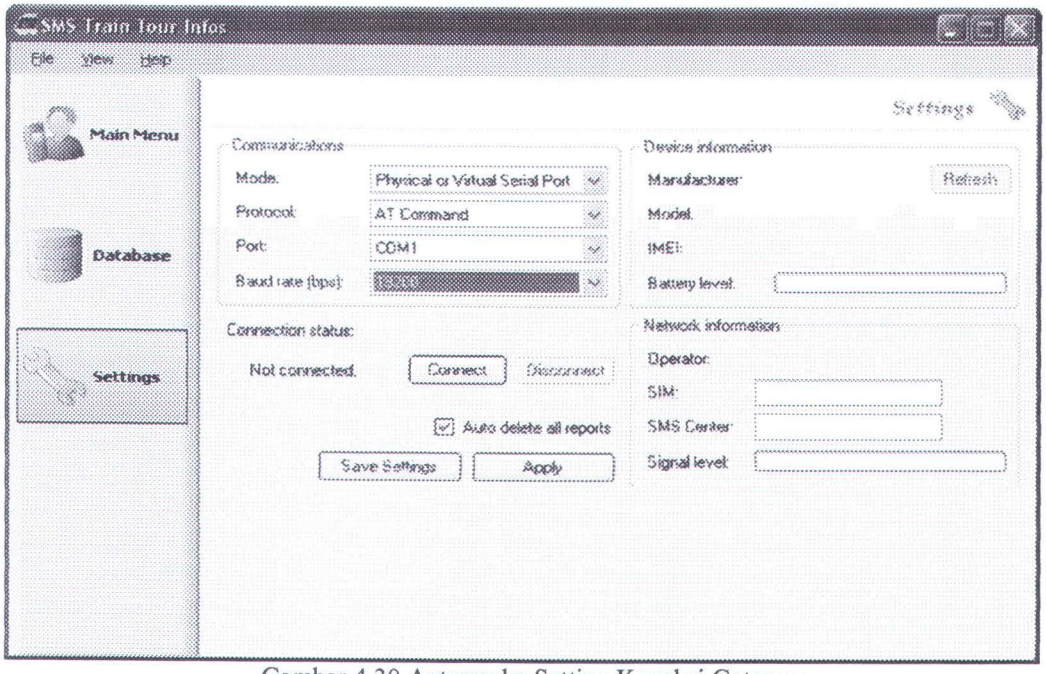
            Me.frmRunningText.Opacity = 1
        ElseIf (Me.btnShowHideRunningText.Text = "Hide") Then
            Me.btnShowHideRunningText.Text = "Show"
            Me.btnShowHideRunningText.Image =
My.Resources.Resources.form_show

            Me.frmRunningText.Opacity = 0
        End If
    End Sub
```

Gambar 4.29 Penggalan Perintah btnShowHideRunningText_Click()

4.3 Implementasi Antarmuka

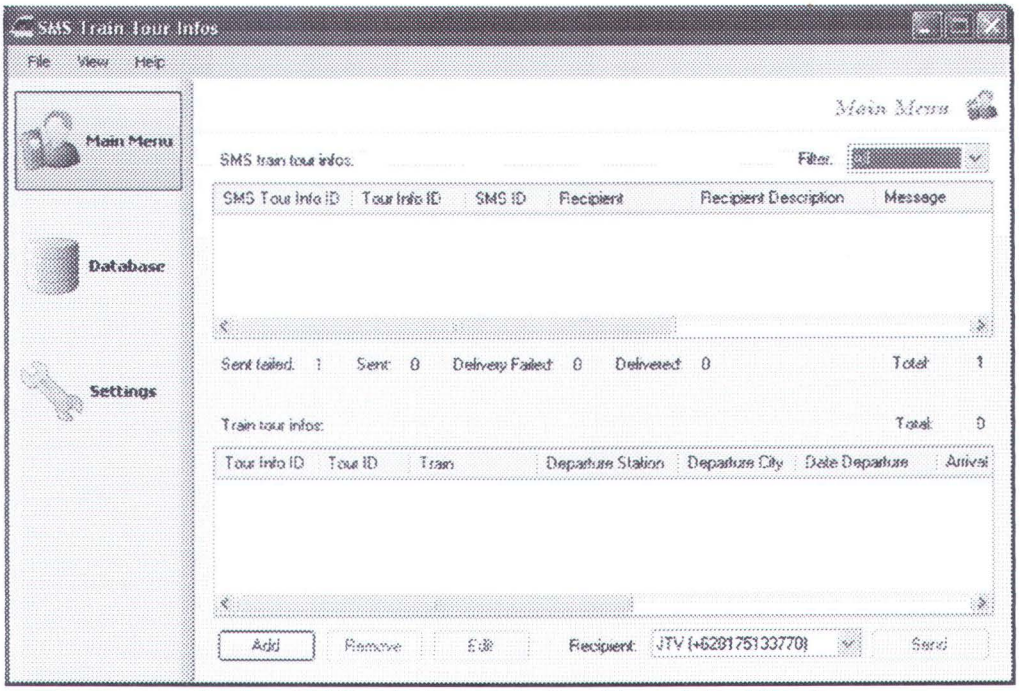
4.3.1 Proses Koneksi SMS Gateway



Gambar 4.30 Antarmuka Setting Koneksi Gateway

Operator kereta sebelum melakukan proses pengiriman sms, terlebih dahulu melakukan setting koneksi untuk sms (SMS gateway), yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas.

4.3.2 Proses Pengiriman Jadwal Kereta Api

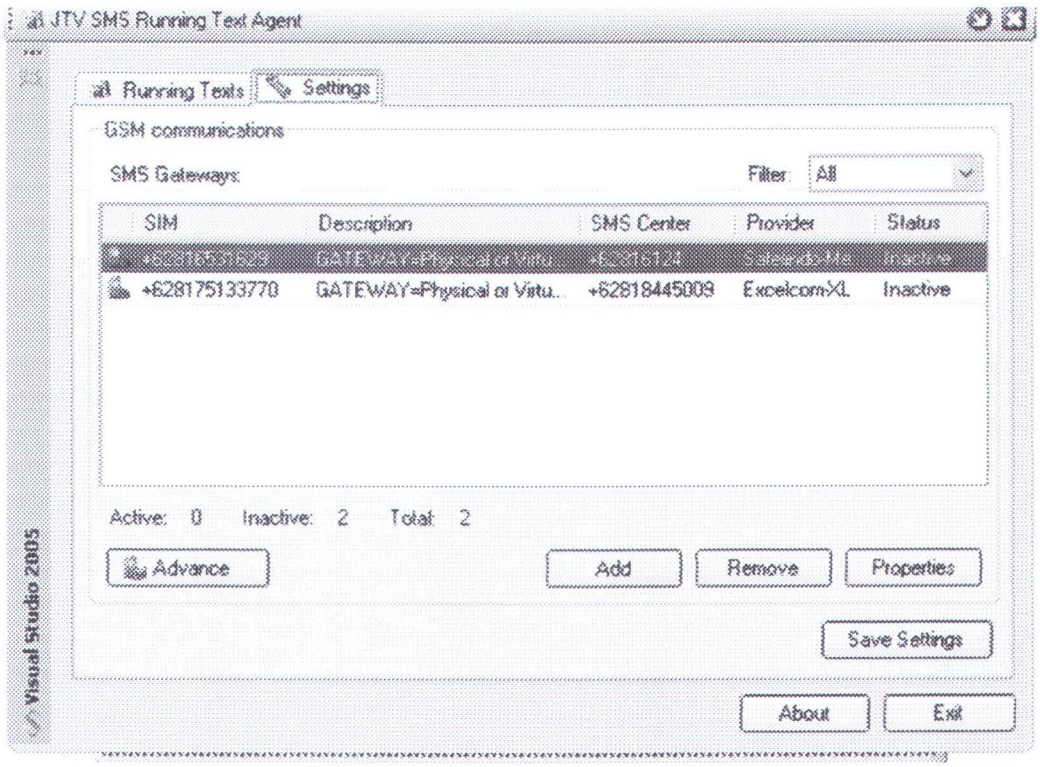


Gambar 4.31 Desain Antarmuka Pengiriman Jadwal Kereta

Aktivitas yang dilakukan oleh operator kereta pada menu yang sesuai dengan gambar diatas adalah penambahan (*add*), penghapusan (*remove*), pengubahan (*edit*) yang dilakukan oleh operator kereta terhadap data jadwal kereta api, berikut pemasukan data status dan informasi yang terjadi pada kereta api. Setelah itu operator kereta dapat memilih nomor telepon tujuan sms pada menu *recipient* dan mengirimkannya dengan mengklik tombol *send.* Semua data jadwal kereta api yang telah mengalami proses pengiriman atau *send* akan tertera

di *SMS Traintour Infos* yang dilengkapi dengan keterangan terhadap status sms, yaitu *sent failed, sent, delivery failed, delivered.* Dan pada menu *filter*, operator dapat melihat total sms yang masuk berdasarkan penggolongan status sms yang ada.

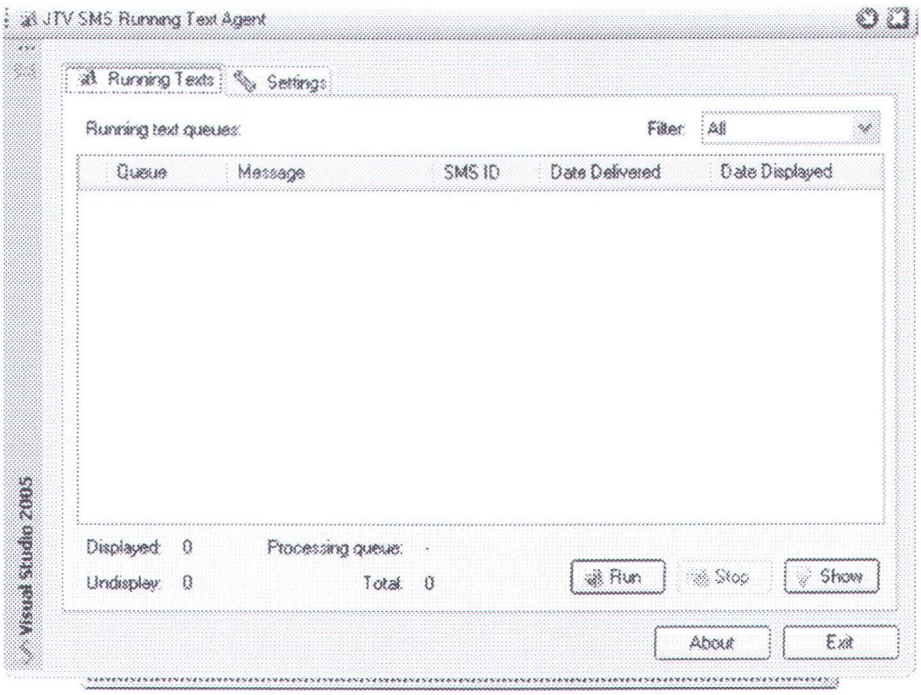
4.3.3 Proses Koneksi SMS TV



Gambar 4.32 Desain Antarmuka koneksi sms TV

Langkah awal yang dilakukan adalah melakukan setting koneksi untuk sms (SMS gateway) terlebih dahulu, yang terdapat pada menu *Settings* yang tertera sesuai dengan gambar diatas. Selain itu operator televisi juga dapat melakukan penyaringan terhadap sms yang masuk, supaya tidak terjadi masuknya sms yang tidak diinginkan. Proses penyaringan terdapat pada menu *advance.*

4.3.4 Proses Manajemen SMS TV



Gambar 4.33 Desain Antarmuka Manajemen SMS TV

Penerimaan sms dilakukan oleh sistem yang terdapat pada perangkat lunak dan sms yang masuk akan tertera pada *Running Text Queues*. Operator televisi bertugas untuk menampilkan sms yang masuk pada layar televisi. Untuk menampilkannya operator televisi dapat menekan tombol *run*, maka informasi data sms jadwal transportasi yang telah diterima oleh sistem yang terdapat pada perangkat lunak akan tersajikan dalam bentuk *running text*. Operator televisi juga dapat menghentikan *running text* yang sedang ditampilkan pada layar televisi, untuk tidak ditampilkan lagi yaitu dengan menekan tombol *stop* dan menyembunyikan *running text* yang sedang tayang pada layar televisi dengan menekan tombol *hide* kemudian tekan *show* untuk menampilkannya kembali.

BAB V

UJI COBA DAN EVALUASI

BAB V

UJI COBA DAN EVALUASI

Bab ini menjelaskan tentang proses pengujian terhadap sistem yang telah dibuat dan melakukan evaluasi terhadap hasil pengujian tersebut. Tujuan dilakukannya uji coba ini adalah untuk mengetahui apakah sistem yang telah diimplementasikan ini bisa berjalan sebagaimana mestinya

5.1 Lingkungan Uji Coba

Lingkungan uji coba adalah komputer tempat dimana uji coba sistem dilakukan. Pada uji coba kali ini digunakan dua buah personal computer (PC) dengan spesifikasi sebagai berikut:

- Spesifikasi Perangkat Keras I : NoteBook Pentium (R) dengan kecepatan 1.60 GHz, kapasitas RAM 512 Mb, kapasitas Hard disk 60 GB.
- Spesifikasi Perangkat Keras II: Pentium IV dengan kecepatan 2.40 GHz, kapasitas RAM 512 Mb, kapasitas Hard disk 80 GB untuk server televisi.
- VGA card TV Output : NVIDIA GeForce4 MX 4000 untuk perangkat keras II
- Sistem Operasi: MS Windows XP Professional version 2002 Service Pack 2.
- Database: MS Access 2000.
- Sambungan terminal data: Kabel Data SIM C35
- Port Komputer: Port 1 (COM1).
- Wavecom untuk server stasiun televisi.
- Satu buah televisi umum untuk menguji running text, apakah dapat muncul pada layar televisi.

- Terminal SMS Gateway: Ponsel Siemens C35.

Sedangkan spesifikasi perangkat keras minimum adalah Pentium III dengan kecepatan 400MHz dan RAM sebesar 256 Mb.

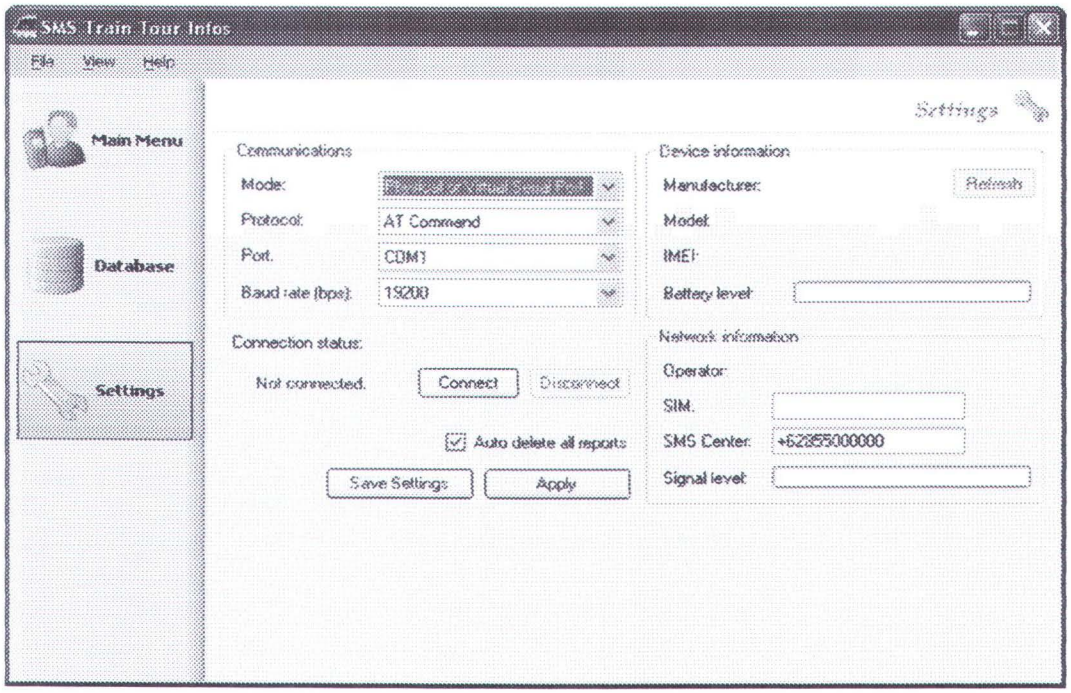
5.2 Pelaksanaan Uji Coba

Uji coba dilakukan dengan lima buah skenario yang berbeda guna menguji keberhasilan implementasi perangkat lunak.

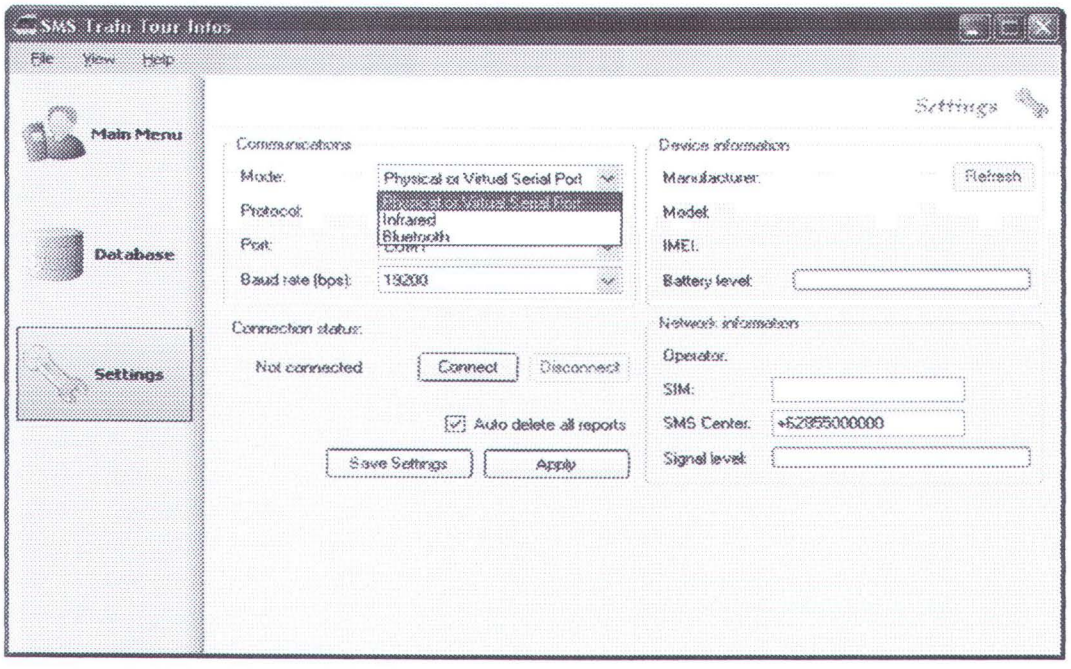
5.2.1 Skenario Proses Manajemen Koneksi sms Kereta

Skenario pertama ini berfungsi unutk menguji apakah aplikasi dapat menjalankan proses manajemen koneksi sms kereta api atau pesawat terbang dengan baik dan pengujian ini dilakukan pada komputer yang terdapat pada perusahaan kereta api maupun perusahaan pesawat terbang yang sudah terinstal aplikasi ini di dalamnya. Pengujian skenario Proses Manajemen Koneksi sms Kereta dapat dilihat pada bagian Lampiran tepatnya pada tabel 5.1

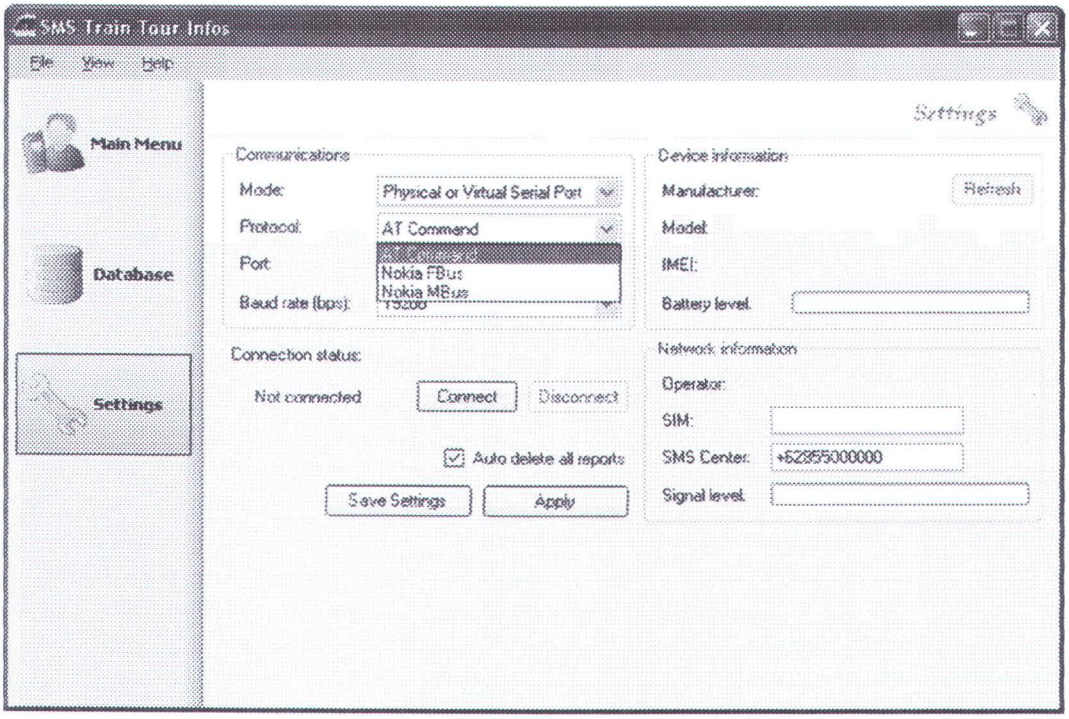
Gambar-gambar dibawah ini adalah menerangkan prosedur-prosedur yang terjadi sesuai dengan proses yang dilakukan pada tabel 5.1. Dimulai dari memilih menu *Settings* yang terdapat pada bagian pojok kiri bawah, sesuai dengan yang terlihat pada gambar 5.1 dibawah ini.



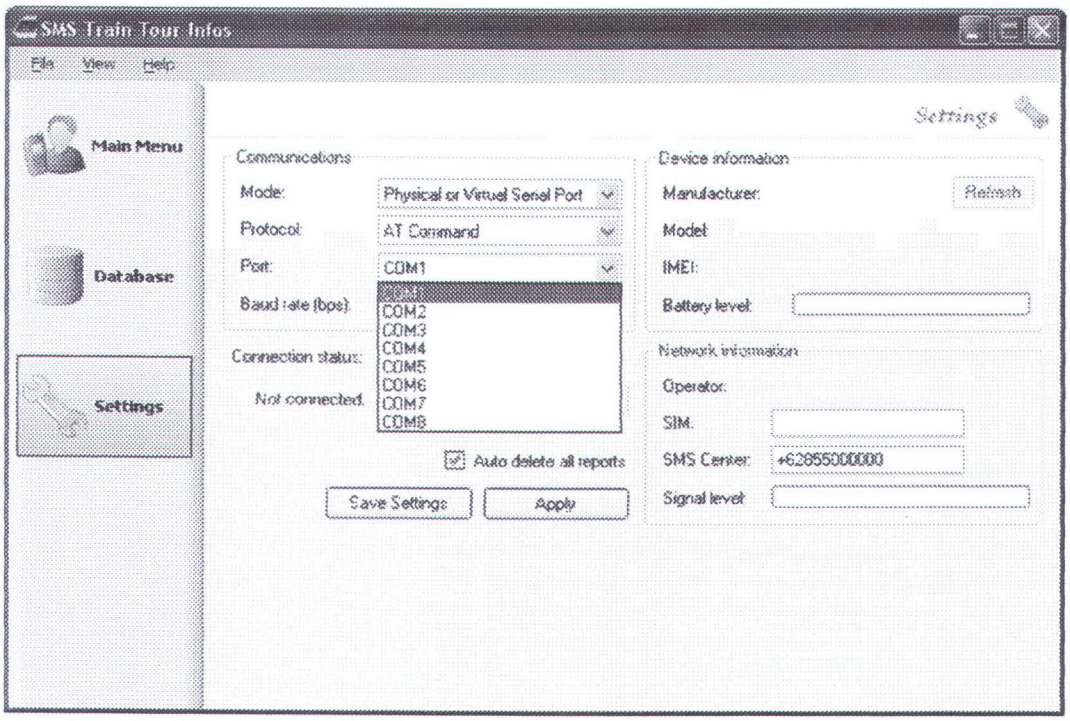
Gambar 5.1 Menu setting koneksi



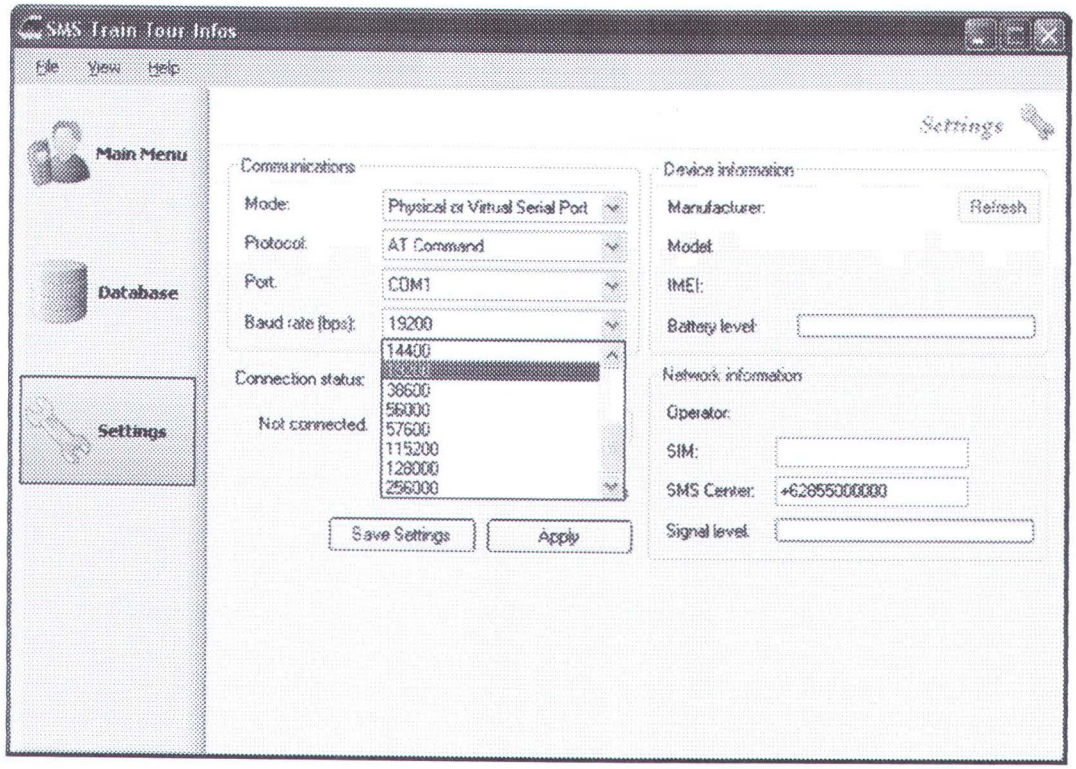
Gambar 5.2 Antarmuka pemilihan menu mode



Gambar 5.3 Antarmuka pemilihan menu protocol

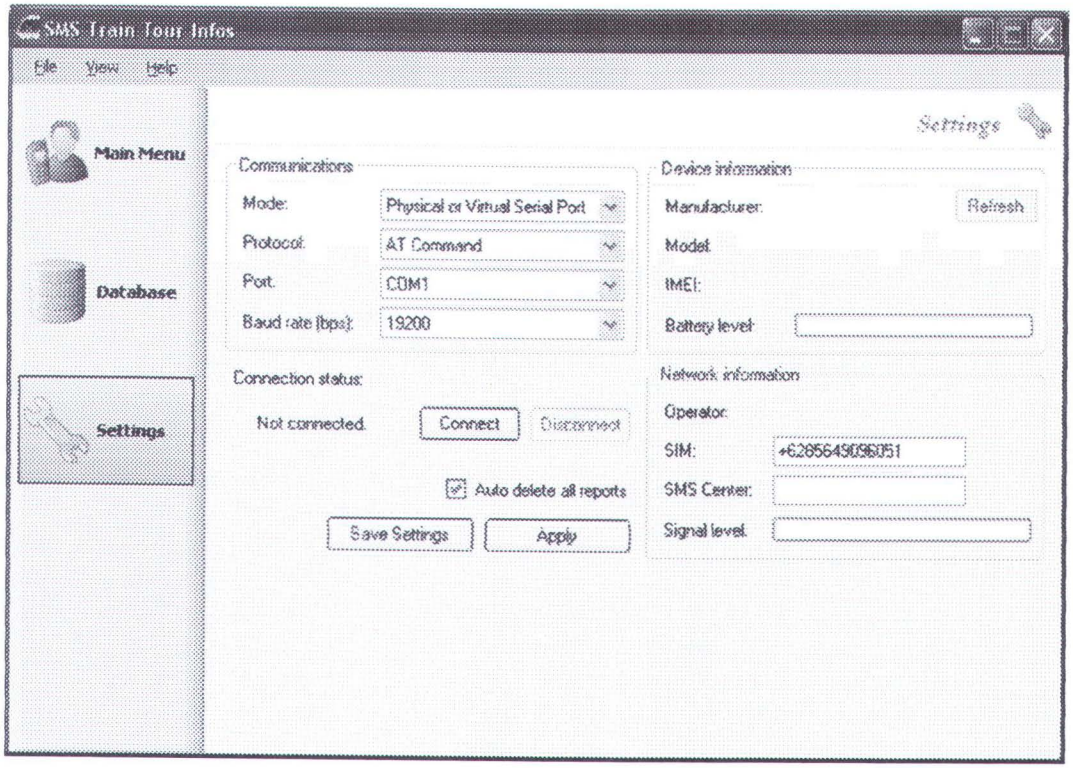


Gambar 5.4 Antar Muka pemilihan menu port



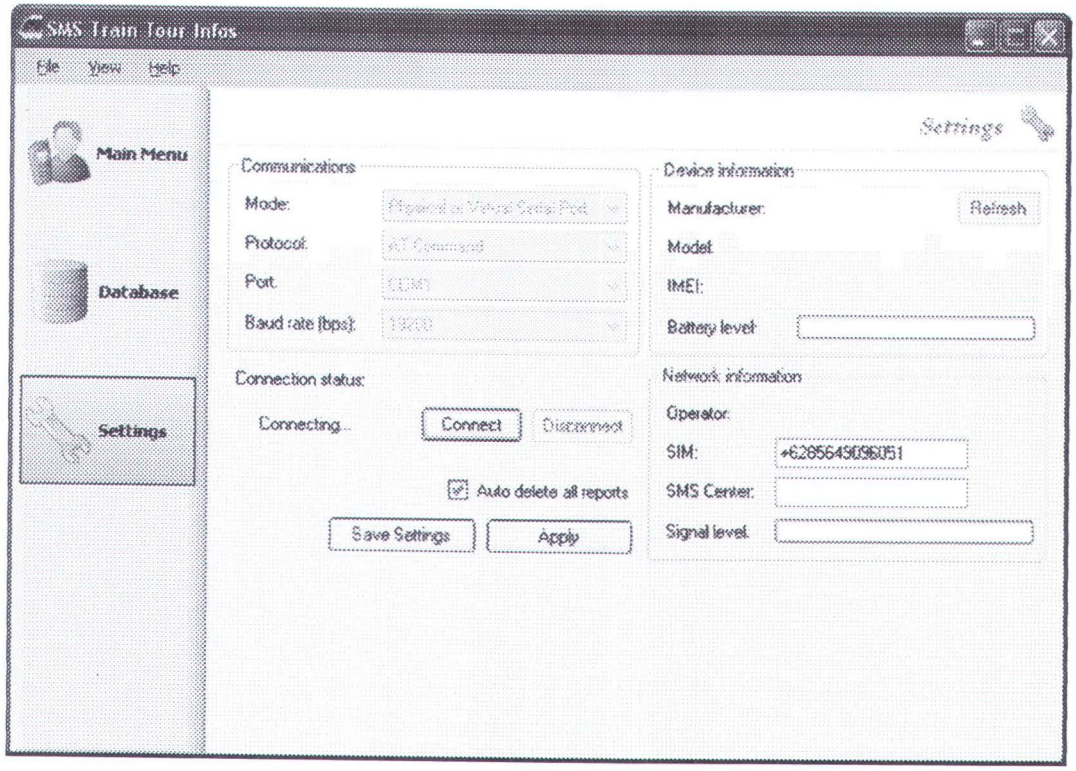
Gambar 5.5 Antar muka pemilihan menu Baud rate

Setelah selesai melakukan proses pengisian pada bagian *Communications* langkah selanjutnya adalah mengisi nomor *SIM* pada bagian *Network information*, nomor *SIM* yang diisikan harus sesuai dengan SIM yang terdapat di dalam ponsel seperti terlihat pada gambar 5.6.

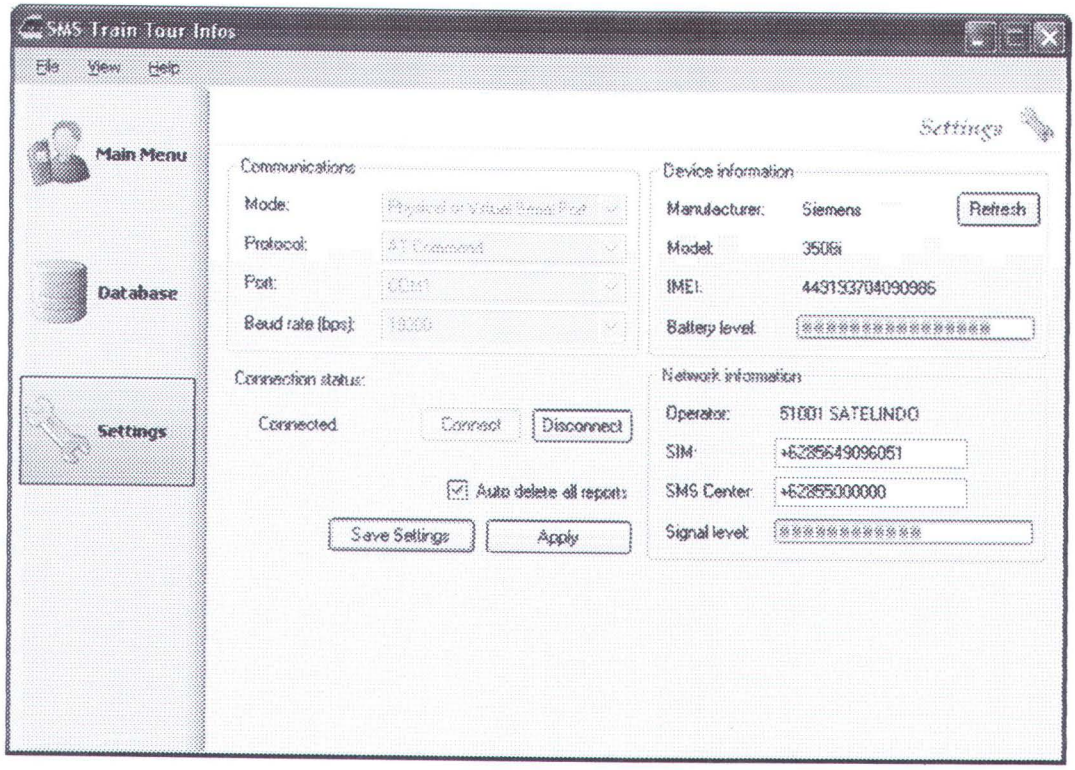


Gambar 5.6 Antar muka pengisian nomor SIM

Langkah terakhir yang harus dilakukan agar koneksi bisa terjadi adalah menekan button *Connect* (gambar 5.7). Apabila koneksi sudah terjadi maka secara otomatis bagian *Device information* dan *Network Information* akan menampilkan data spesifikasi dari ponsel C35 (gambar 5.8).

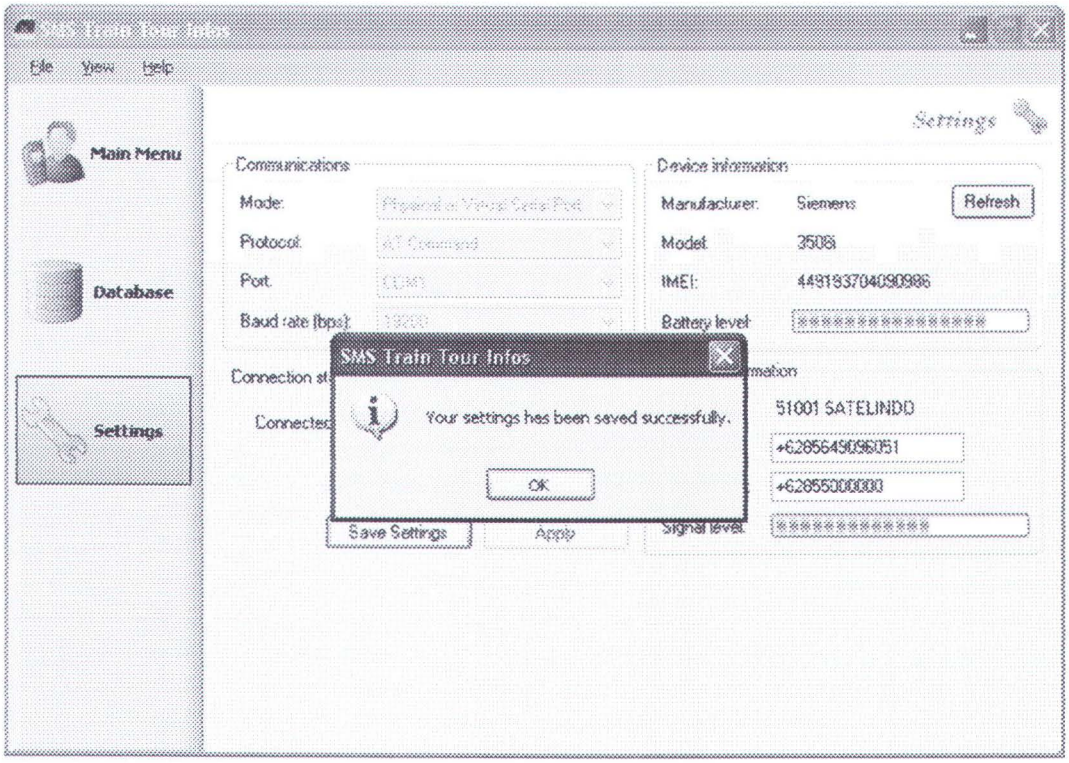


Gambar 5.7 Antar muka button connect ditekan



Gambar 5.8 Antar muka koneksi gateway terjadi

Setting koneksi diatas dapat juga disimpan apabila memang diperlukan. Penyimpanan dilakukan menggunakan metode operasi file dimana file yang dipakai untuk menyimpan setting tersebut bernama *settings.ini*. Gambar 5.9 dibawah ini menampilkan proses penyimpanan yang berhasil dilakukan.



Gambar 5.9 Antar muka Proses penyimpanan berhasil dilakukan

5.2.2 Skenario Proses Manajemen Data Kereta Api

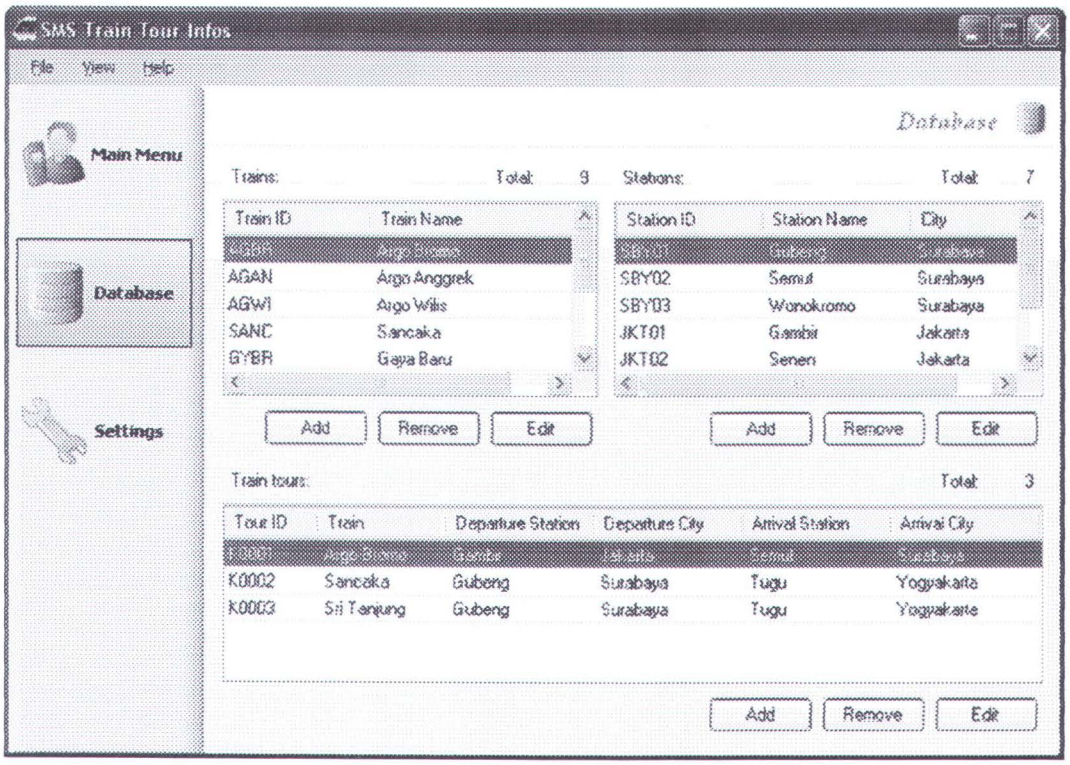
Skenario kedua tentang proses manajemen data kereta api ini digunakan untuk menguji apakah aplikasi bisa melakukan proses manajemen data dengan baik mulai dari proses melihat data, menambah data, mengedit data, sampai menghapus data. Skenario ini akan dibagi menjadi beberapa sub skenario berdasarkan jenis data yang ada dalam menu. Pengujian skenario Proses

Manajemen data kereta api dapat dilihat pada bagian lampiran tepatnya pada tabel 5.2

5.2.2.1 Data Kereta Api

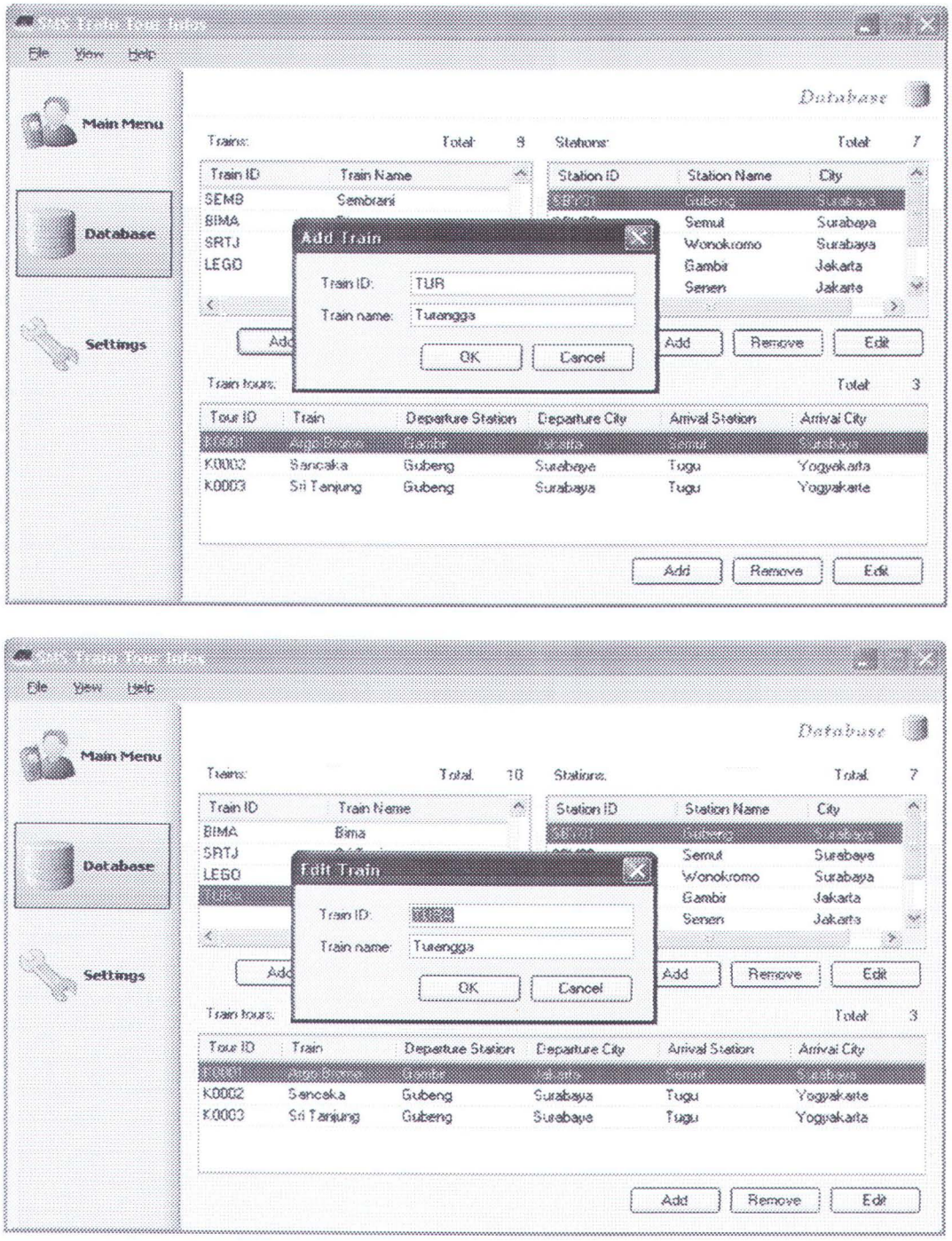
Sub skenario data kereta api digunakan untuk menguji apakah aplikasi bisa melakukan proses melihat data, menambah data, mengedit data, dan menghapus data kereta api.

Pertama, untuk memanajemen kereta api pengguna harus memilih menu *Database* yang berada pada menu sebelah kiri atau menubar bagian tengah. Setelah pengguna memilih menu *Database* maka muncul tampilan seperti yang terlihat pada gambar 5.10.



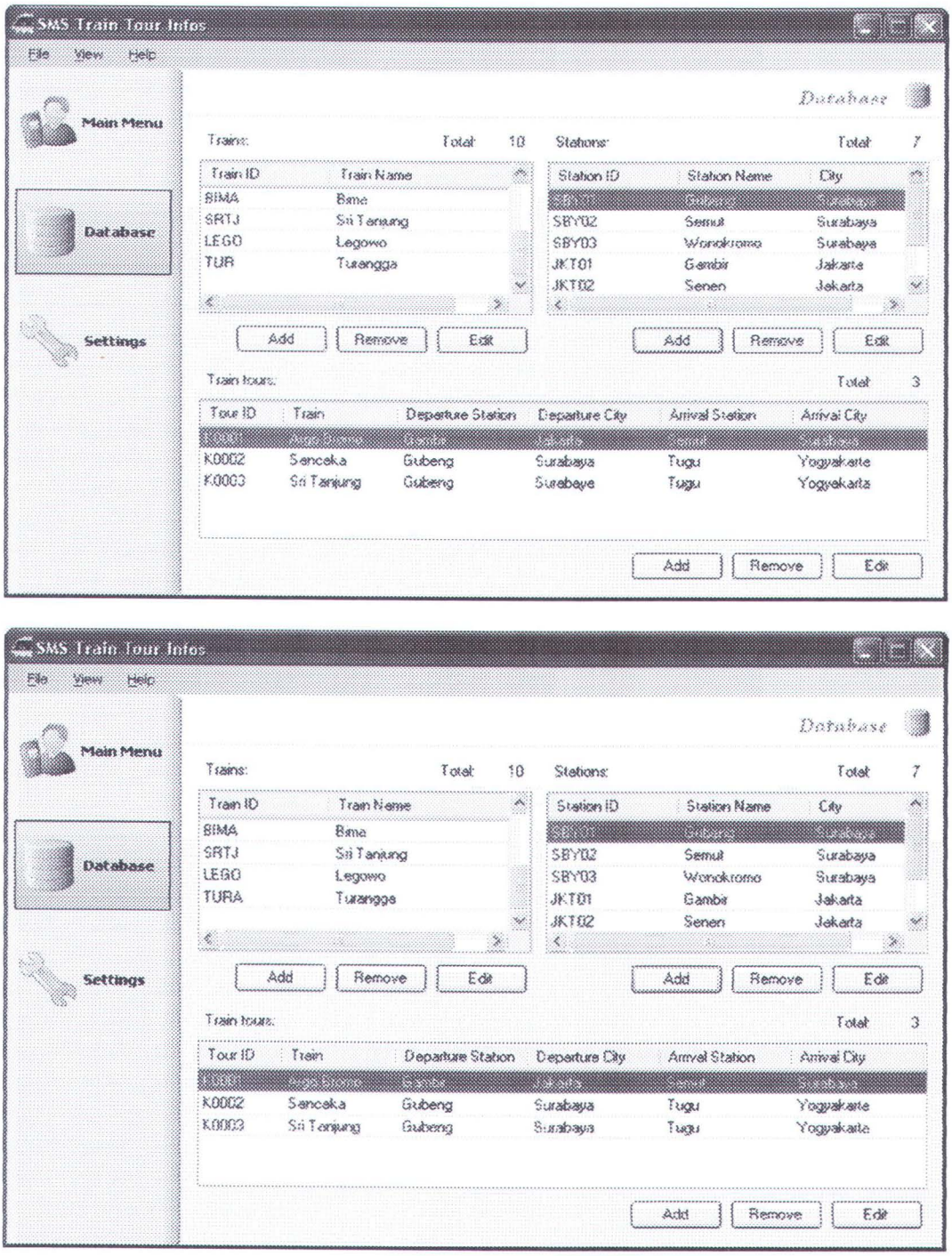
Gambar 5.10 Melihat Data Kereta Api

Kedua, pengguna menambahkan dan mengedit data Kereta Api, dengan menekan button *Add* atau *Edit* dan muncul tampilan seperti pada gambar 5.11 berikut.



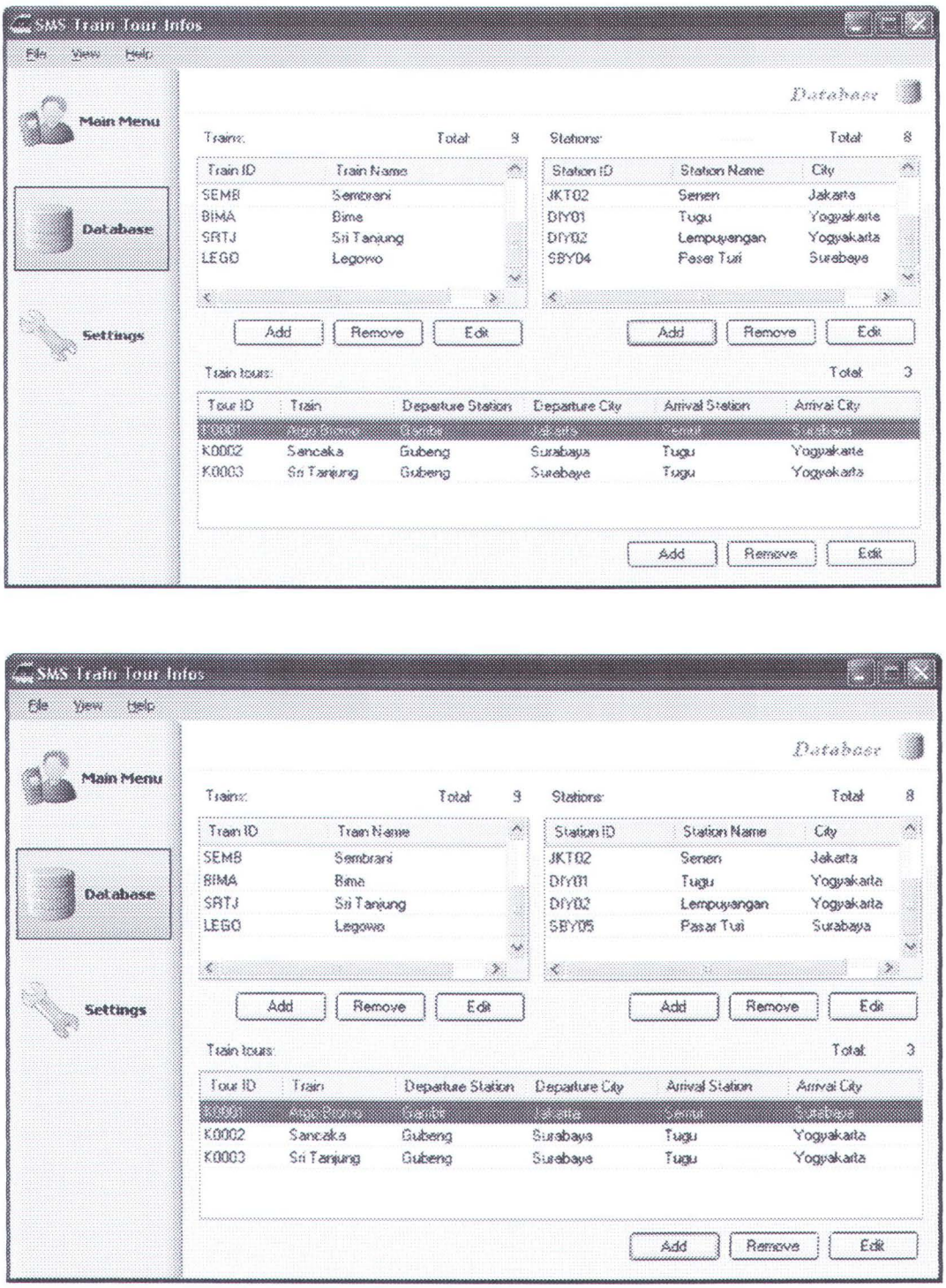
Gambar 5.11 Antar muka Add dan Edit data Kereta Api

Setelah proses dilakukan dengan menekan tombol OK, maka akan muncul tampilan pada gambar 5.12 di bawah dengan data yang telah di-*update*.



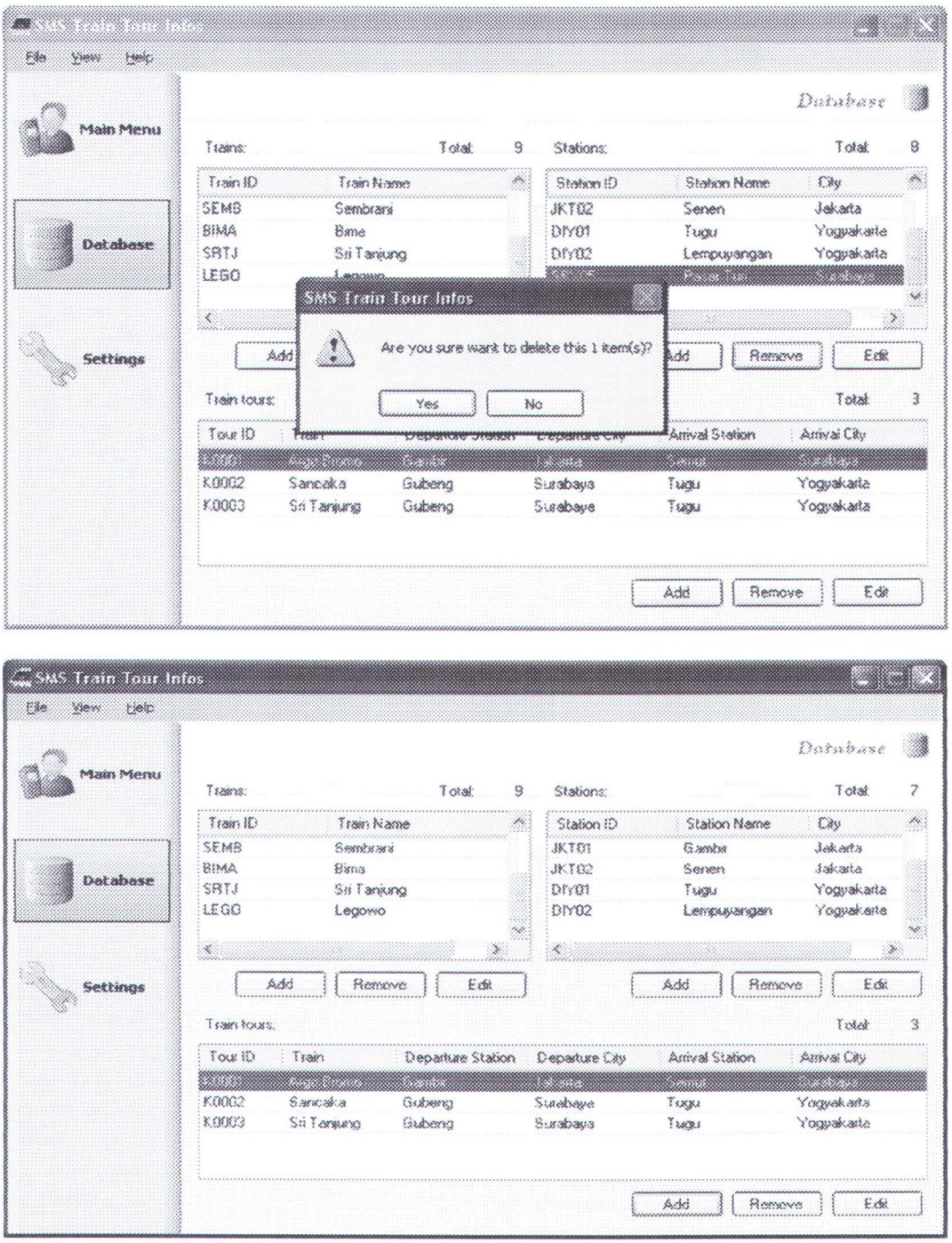
Gambar 5.12 Tampilan data kereta api setelah mengalami penambahan data dan pengubahan (edit) data

Setelah proses dilakukan dengan menekan tombol OK, maka akan muncul tampilan pada gambar 5.16 di bawah dengan data yang telah di-*update*.



Gambar 5.16 Tampilan data stasiun kereta api setelah mengalami penambahan data dan pengubahan (edit) data

Ketiga, pengguna dapat melakukan penghapusan terhadap data stasiun kereta api apabila memang diperlukan. Setelah button *Remove* ditekan maka keluar tampilan seperti tampak pada gambar 5.17 berikut.

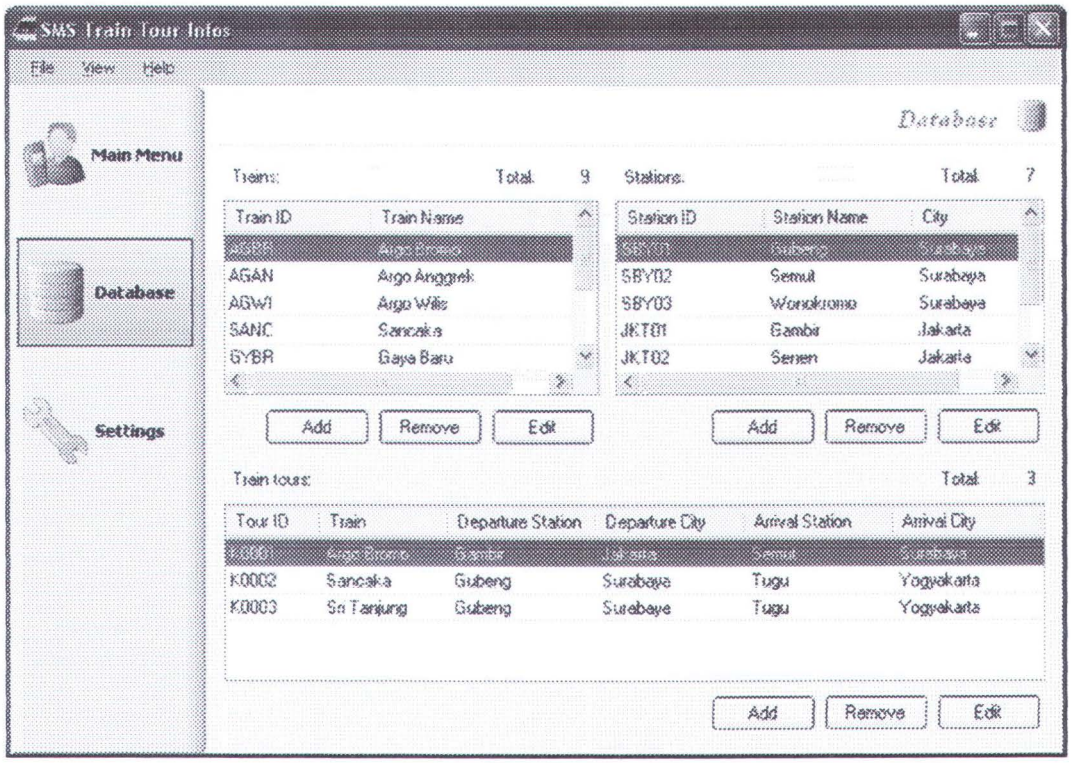


Gambar 5.17 Menghapus entry data stasiun kereta api dan tampilan setelah entry data stasiun kereta api terhapus

5.2.2.3 Data Jurusan Kereta Api

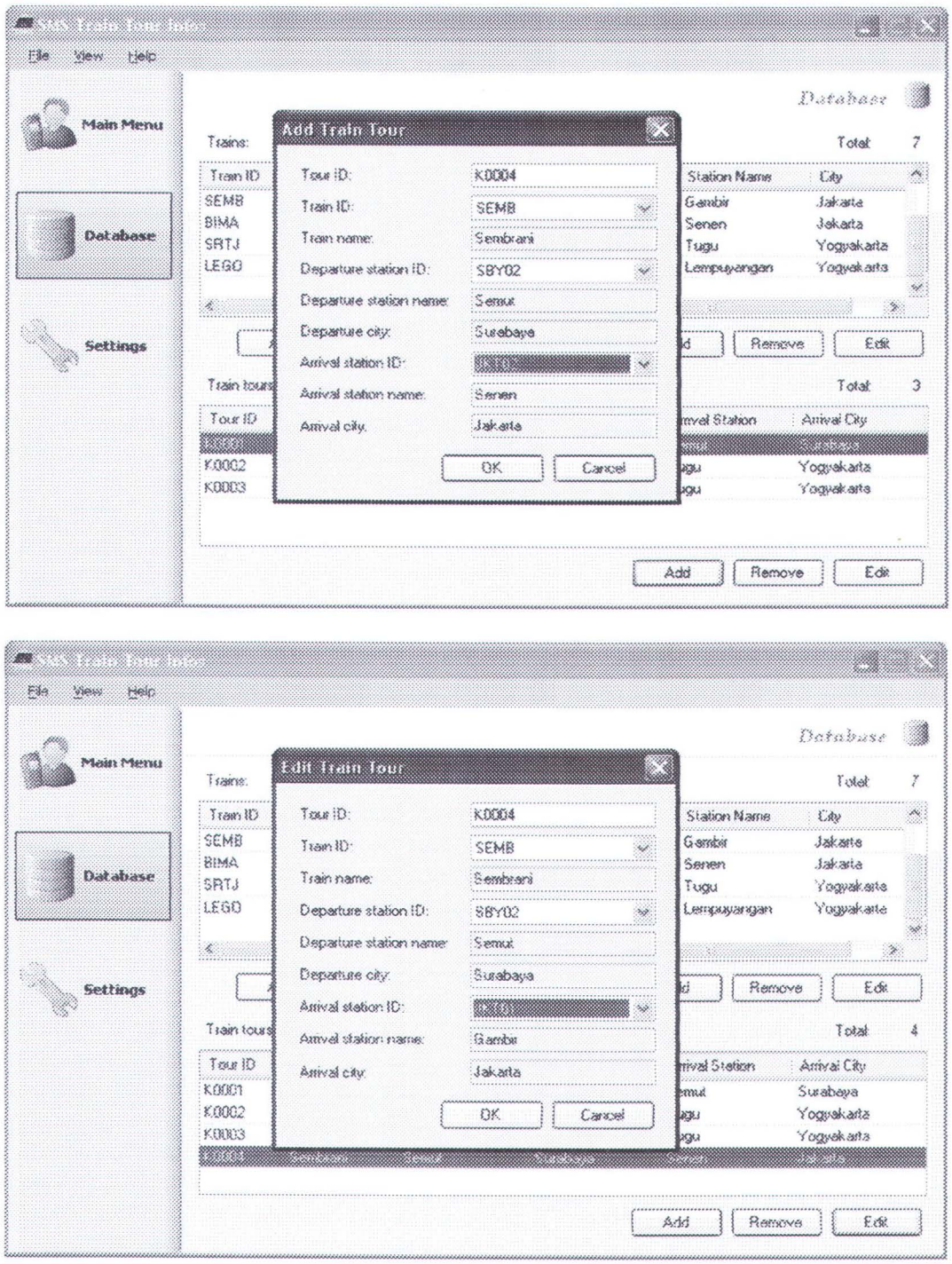
Sub skenario data jurusan kereta api digunakan untuk menguji apakah aplikasi bisa melakukan proses melihat data, menambah data, mengedit data, dan menghapus data jurusan kereta api.

Pertama, untuk memanajemen jurusan kereta api pengguna harus memilih menu *Database* yang berada pada menu sebelah kiri atau menubar bagian tengah. Setelah pengguna memilih menu *Database* maka muncul tampilan seperti yang terlihat pada gambar 5.18.



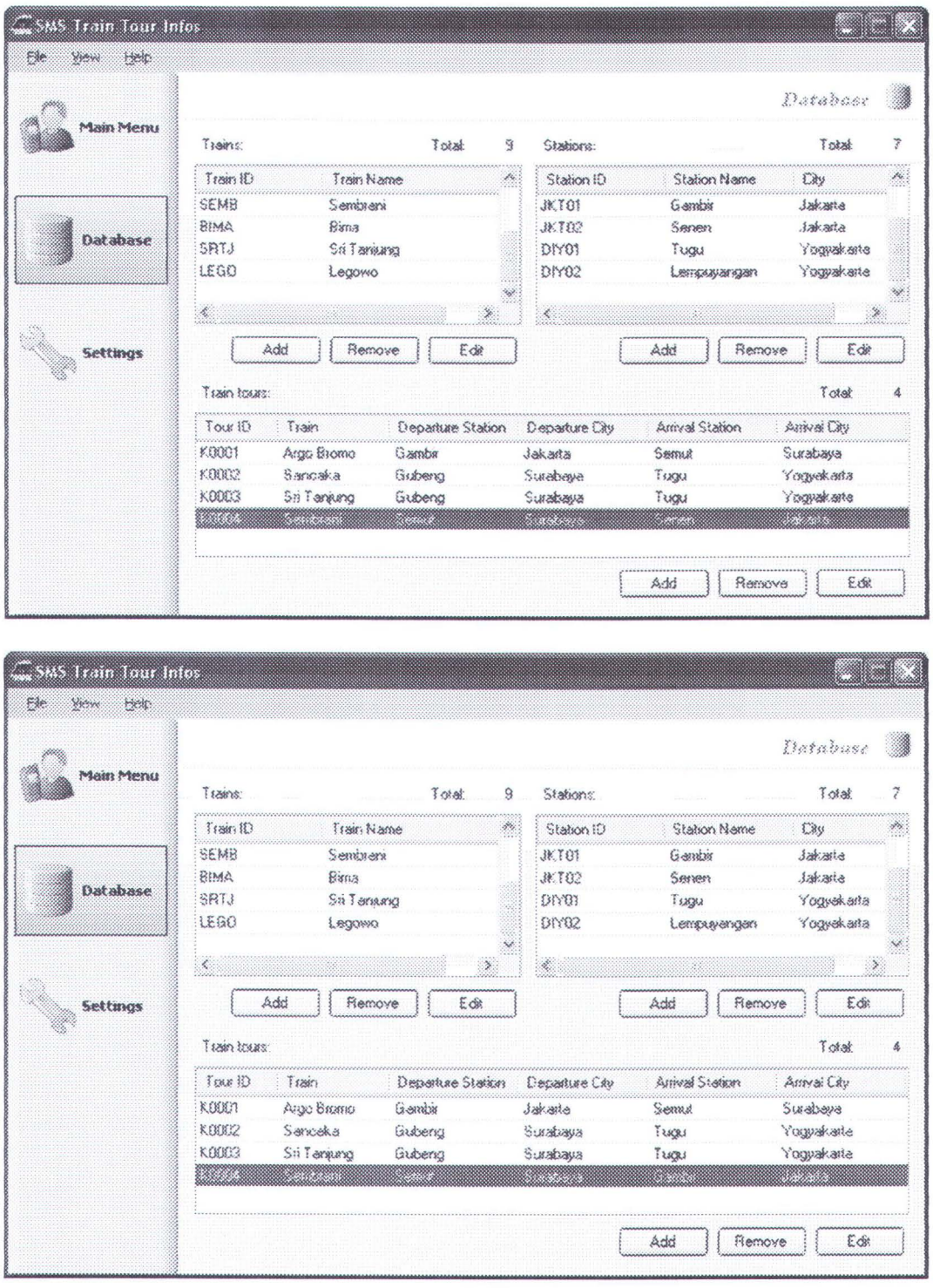
Gambar 5.18 Melihat Data jurusan Kereta Api

Kedua, pengguna menambahkan dan mengedit data jurusan kereta api, dengan menekan button *Add* atau *Edit* dan muncul tampilan seperti pada gambar 5.19 berikut.



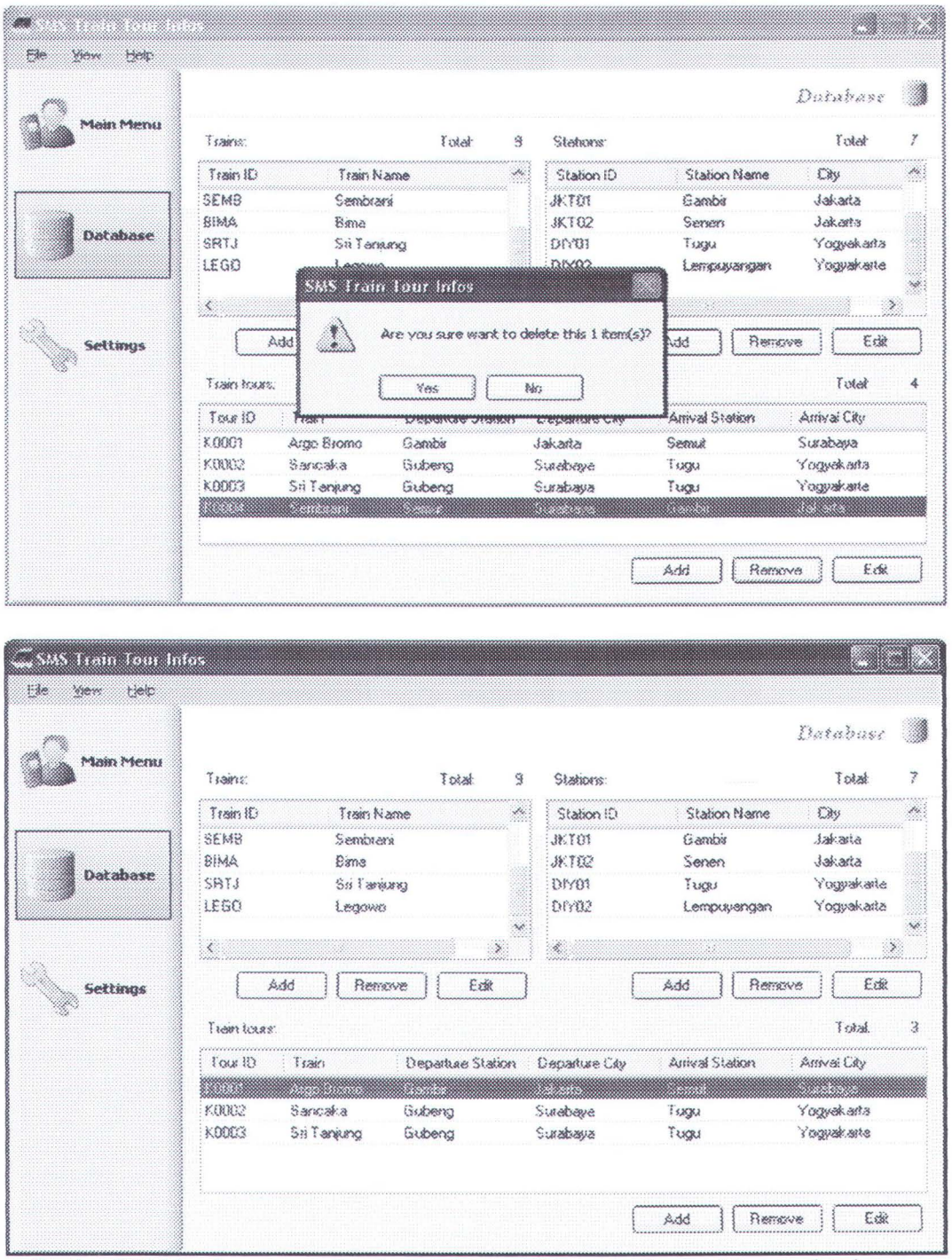
Gambar 5.19 Antar muka Add dan Edit data jurusan Kereta Api

Setelah proses dilakukan dengan menekan tombol OK, maka akan muncul tampilan pada gambar 5.20 di bawah dengan data yang telah di-*update*.



Gambar 5.20 Tampilan data jurusan kereta api setelah mengalami penambahan data dan pengubahan (edit) data

Ketiga, pengguna dapat melakukan penghapusan terhadap data jurusan kereta api apabila memang diperlukan. Setelah button *Remove* ditekan maka keluar tampilan seperti tampak pada gambar 5.21 berikut.

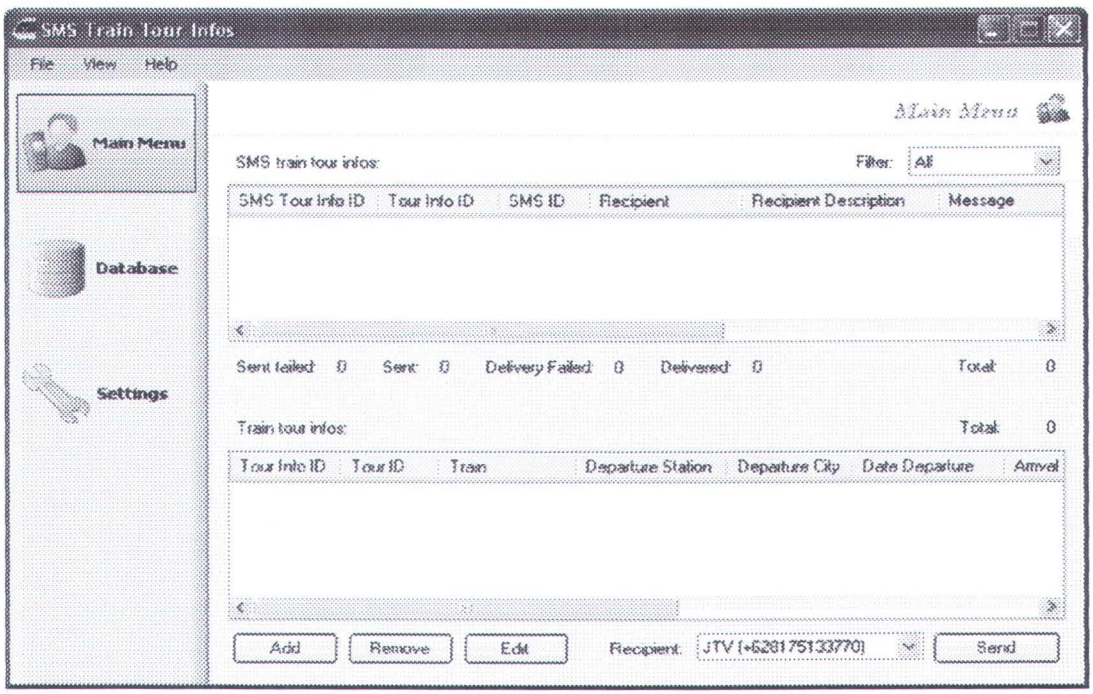


Gambar 5.21 Menghapus entry data jurusan kereta api dan tampilan setelah entry data jurusan kereta api terhapus

5.2.3 Skenario Proses Manajemen Jadwal Kereta Api

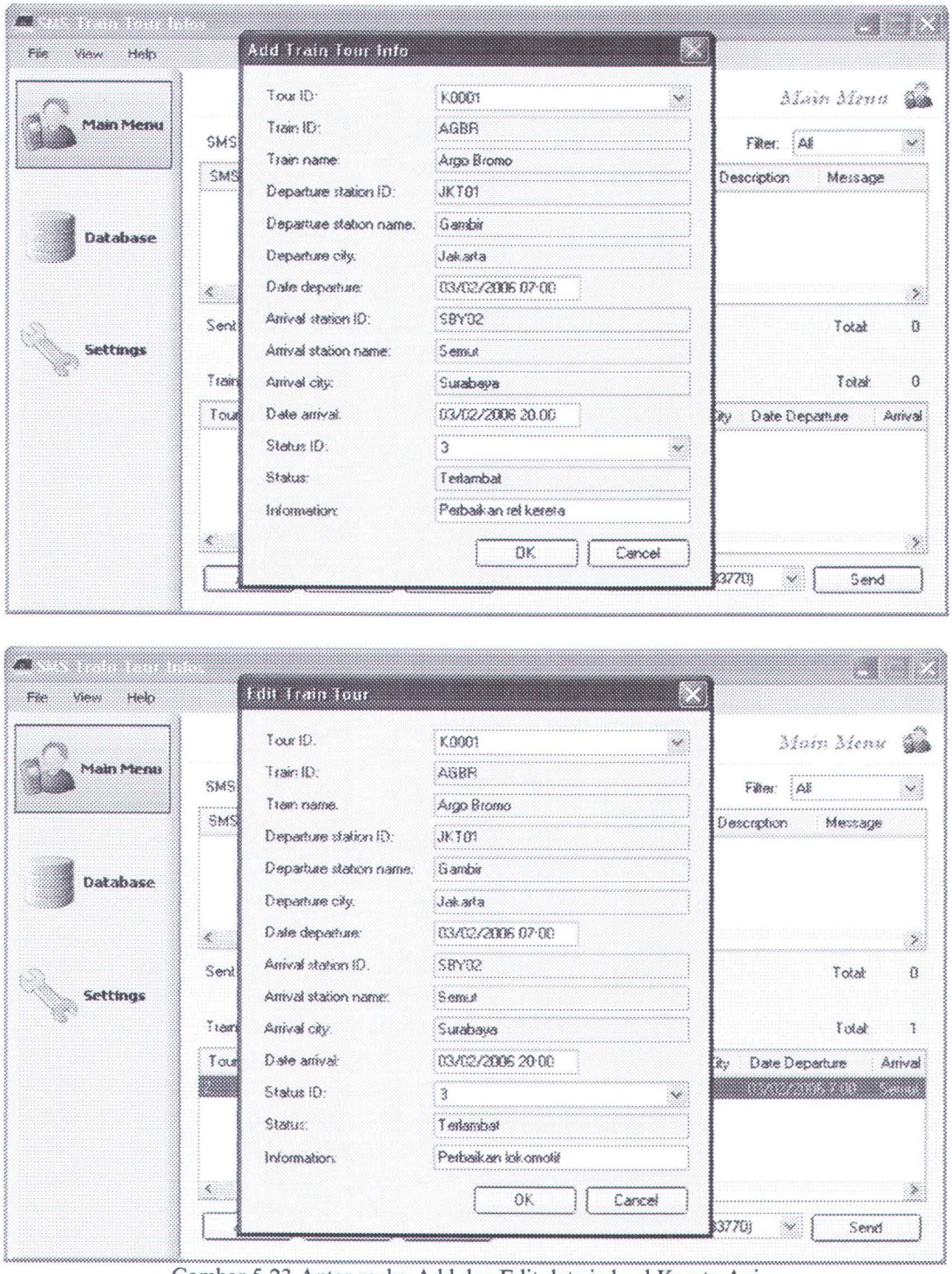
Skenario ketiga ini digunakan untuk menguji apakah aplikasi bisa melakukan penambahan, penghapusan, pengubahan jadwal kereta api dengan status dan informasi mengenai status yang terjadi pada kereta api. Selain itu juga untuk menguji apakah aplikasi dapat menjalankan proses kirim SMS dengan baik dan SMS tersebut sampai ke pihak stasiun televisi yaitu PT. JTV (Jawa Timur Televisi). Skenario Proses Manajemen Jadwal Kereta Api juga dapat dilihat pada bagian Lampiran yaitu pada tabel 5.3

Pertama, untuk memanajemen jadwal kereta api pengguna harus memilih menu *Main Menu* yang berada pada menu sebelah kiri atau menubar bagian atas. Setelah pengguna memilih menu *Main Menu* maka muncul tampilan seperti yang terlihat pada gambar 5.22.



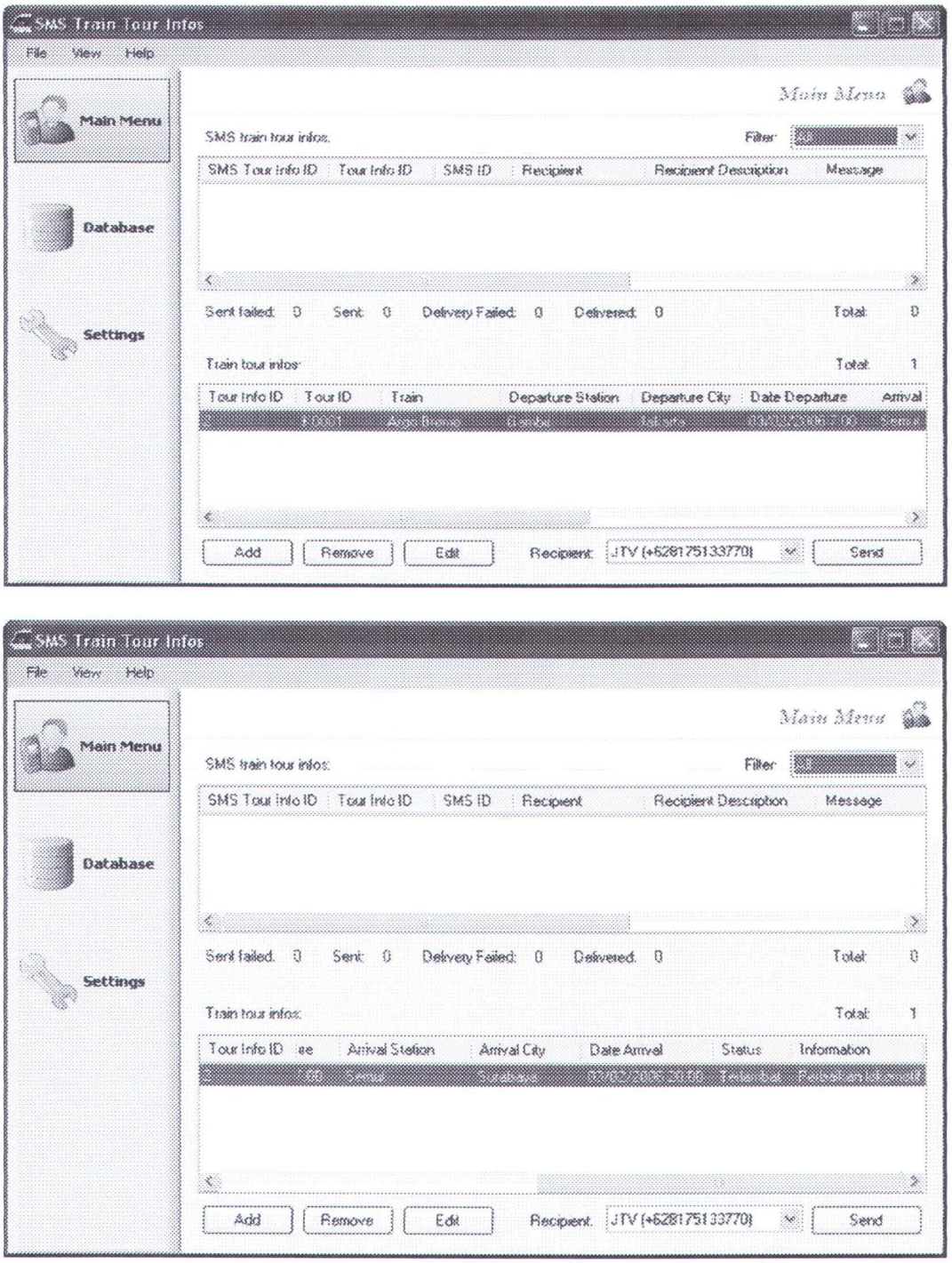
Gambar 5.22 Main Menu

Kedua, pengguna menambahkan dan mengedit data Jadwal kereta api, dengan menekan button *Add* atau *Edit* dan muncul tampilan seperti pada gambar 5.23 berikut.



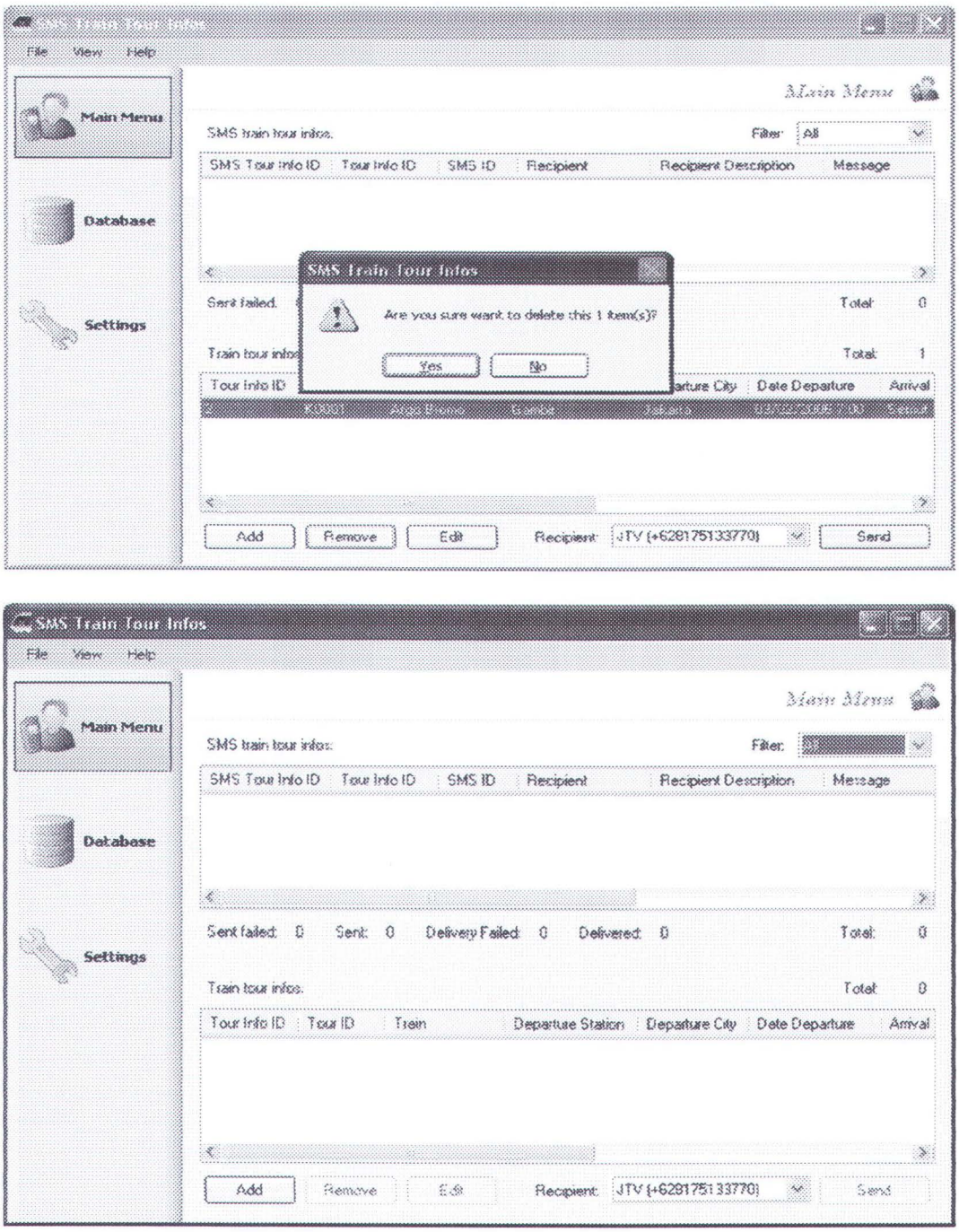
Gambar 5.23 Antar muka Add dan Edit data jadwal Kereta Api

Setelah proses dilakukan dengan menekan tombol OK, maka akan muncul tampilan pada gambar 5.24 di bawah dengan data yang telah di-*update*.



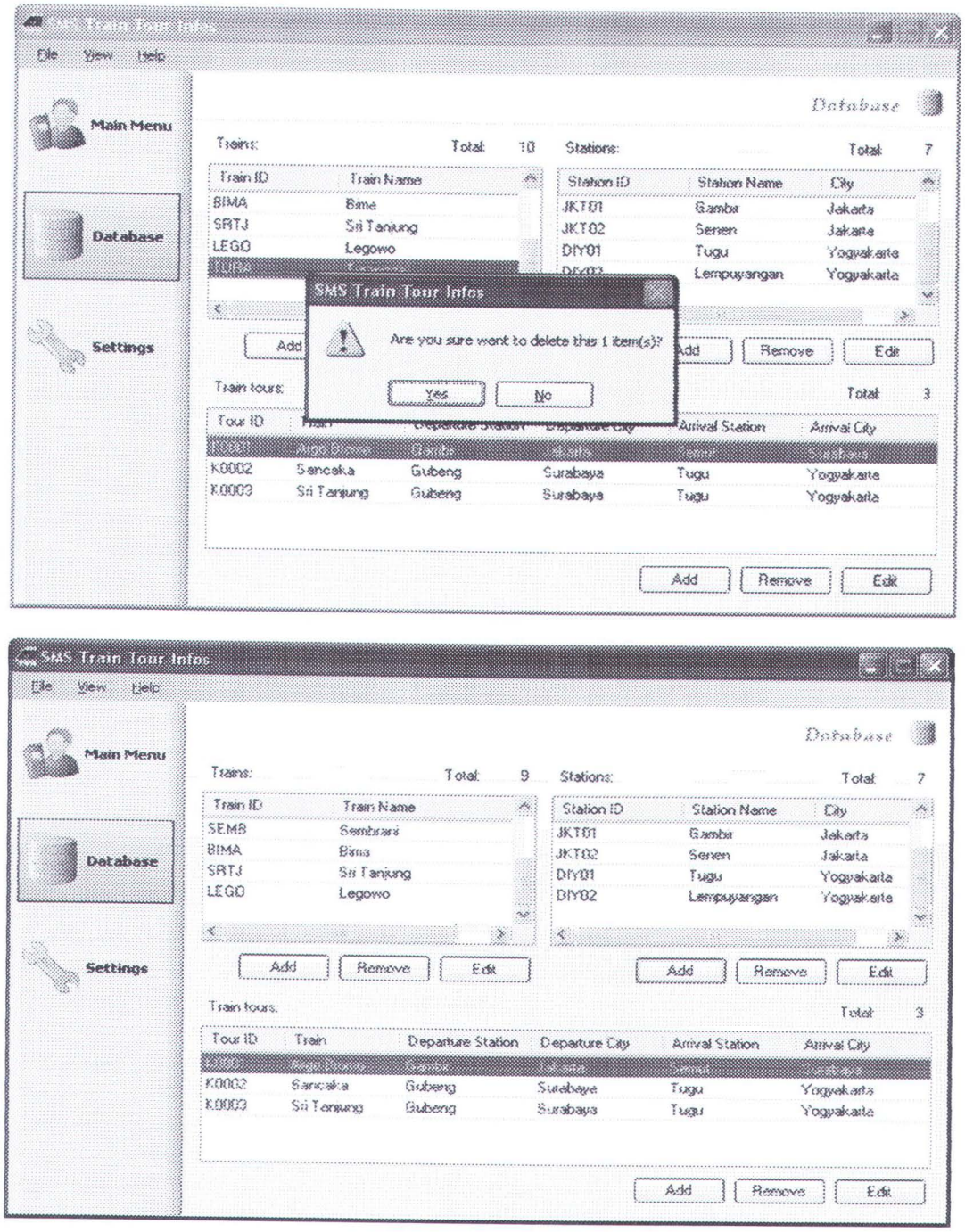
Gambar 5.24 Tampilan data jadwal kereta api setelah mengalami penambahan data dan pengubahan (edit) data

Ketiga, pengguna dapat melakukan penghapusan terhadap data jadwal kereta api apabila memang diperlukan. Setelah button *Remove* ditekan maka keluar tampilan seperti tampak pada gambar 5.25 berikut.



Gambar 5.25 Menghapus entry data jadwal kereta api dan tampilan setelah entry data jadwal kereta api terhapus

Ketiga, pengguna dapat melakukan penghapusan terhadap data kereta api apabila memang diperlukan. Setelah button *Remove* ditekan maka keluar tampilan seperti tampak pada gambar 5.13 berikut.

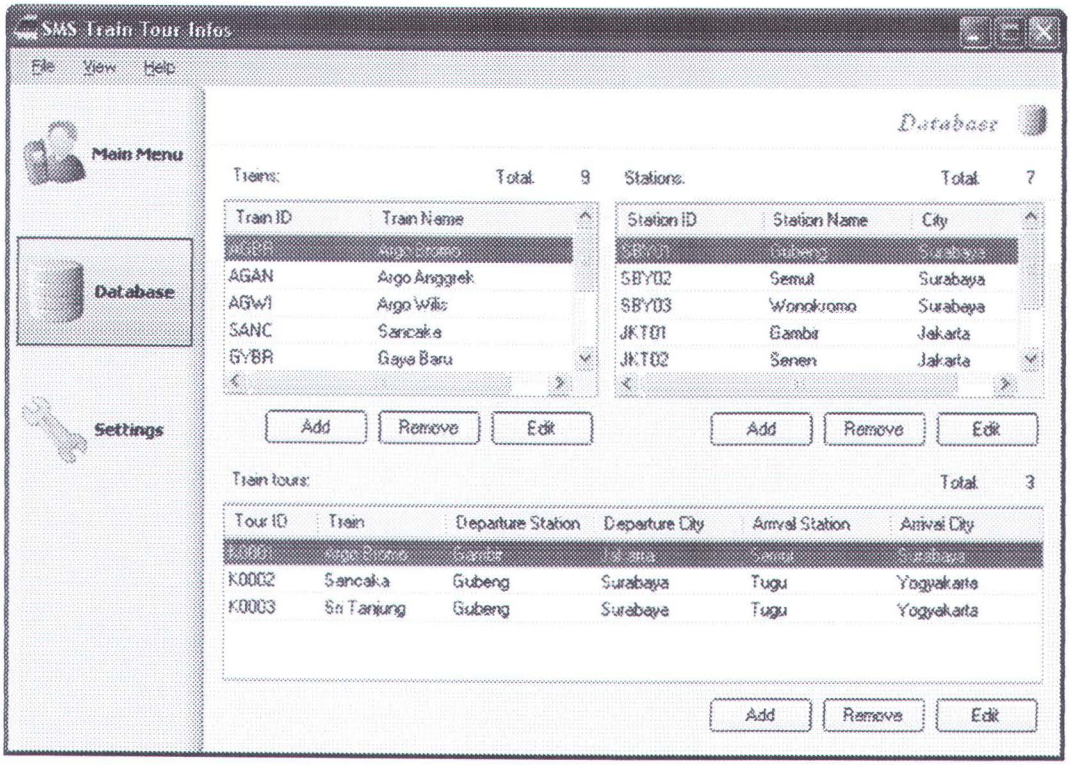


Gambar 5.13 Menghapus entry data kereta api dan tampilan setelah entry data kereta api terhapus

5.2.2.2 Data Stasiun Kereta Api

Sub skenario data stasiun kereta api digunakan untuk menguji apakah aplikasi bisa melakukan proses melihat data, menambah data, mengedit data, dan menghapus data stasiun kereta api.

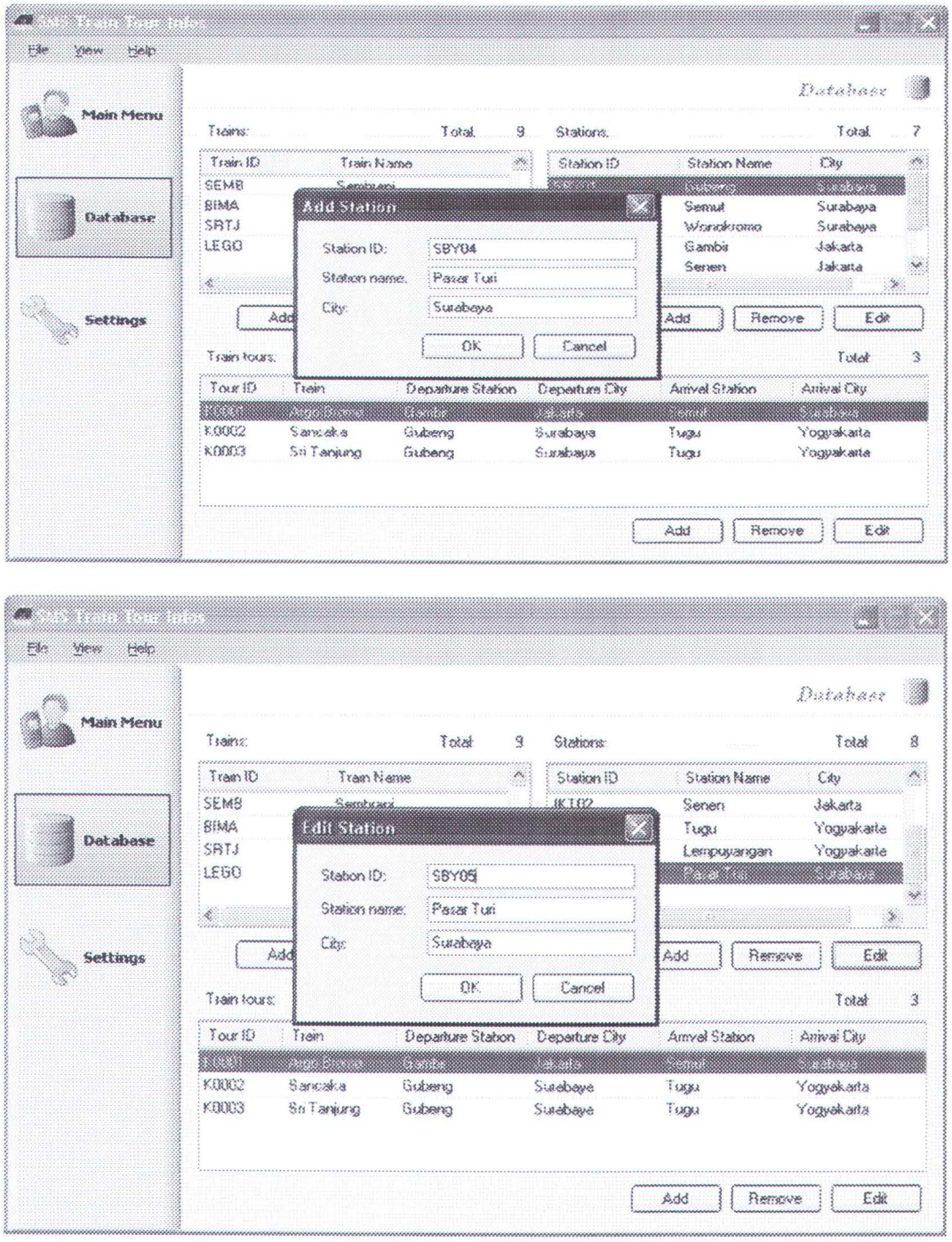
Pertama, untuk memanajemen stasiun kereta api pengguna harus memilih menu *Database* yang berada pada menu sebelah kiri atau menubar bagian tengah. Setelah pengguna memilih menu *Database* maka muncul tampilan seperti yang terlihat pada gambar 5.14.



Gambar 5.14 Melihat Data stasiun Kereta Api

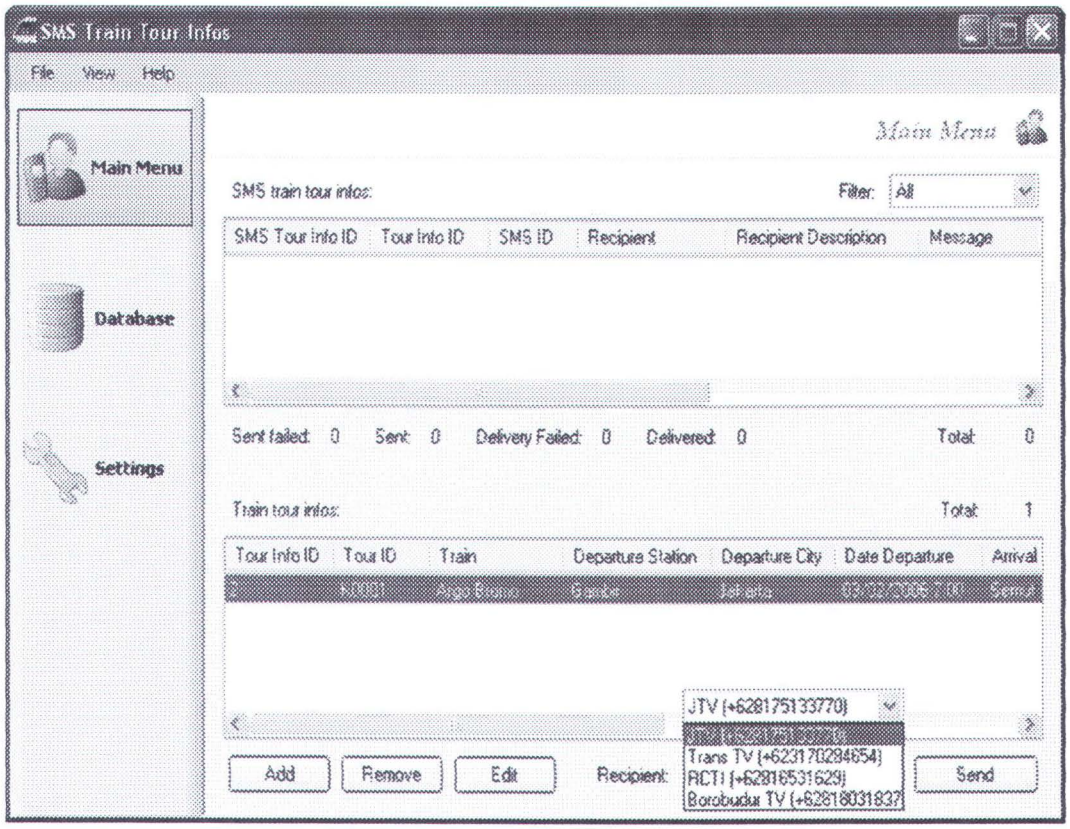
PERPUSTAKAAN ITS

Kedua, pengguna menambahkan dan mengedit data stasiun Kereta Api, dengan menekan button *Add* atau *Edit* dan muncul tampilan seperti pada gambar 5.15 berikut.



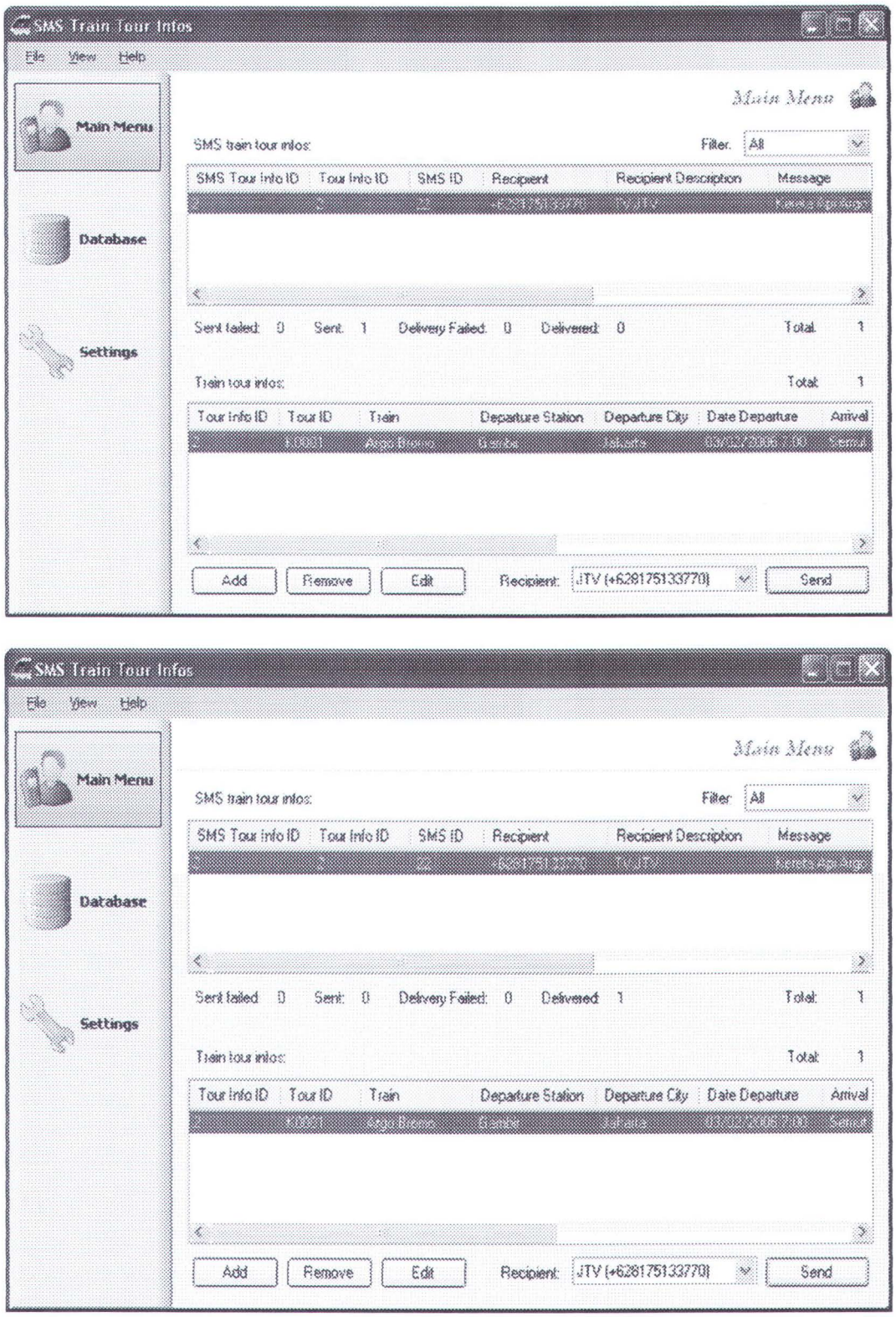
Gambar 5.15 Antar muka Add dan Edit data stasiun Kereta Api

Keempat, untuk mengirim SMS, pengguna harus memilih nomor tujuan sms terlebih dahulu yang terdapat pada combo box menu *Recipient*. Tampilan yang muncul seperti terlihat pada gambar 5.26 berikut.



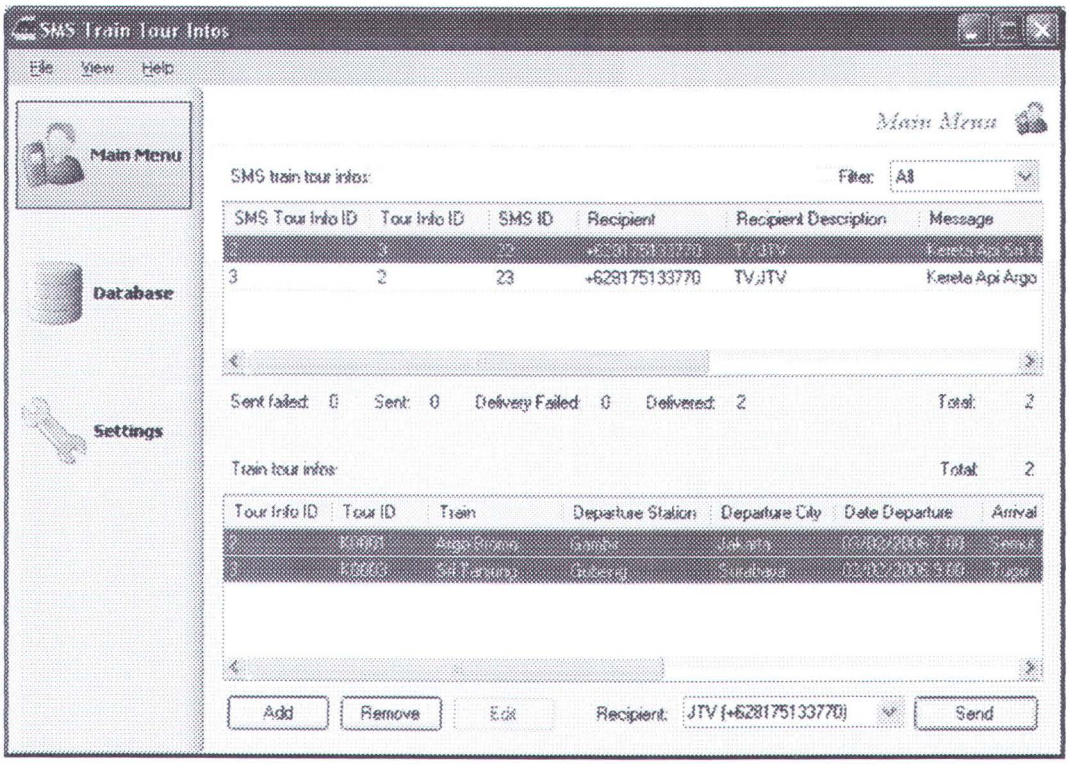
Gambar 5.26 Antar muka pemilihan nomor tujuan sms pada menu Recipient

Untuk mengirimkan jadwal kereta api lengkap dengan status dan informasinya, Pengguna menekan tombol *Send*. Setelah itu label *Sent* akan berubah nilai dari 0 menjadi 1 yang menunjukkan jumlah sms jadwal kereta api yang terkirim, apabila pihak nomor tujuan sms telah menerima pesan tersebut maka label *Delivered* akan berubah nilai dari 0 menjadi 1 dan label *Sent* juga akan berubah nilai menjadi 0 kembali. Hal ini menunjukkan bahwa sms jadwal kereta api telah berhasil sampai pada nomor tujuan. Tampilan yang muncul seperti terlihat pada Gambar 5.27 berikut.



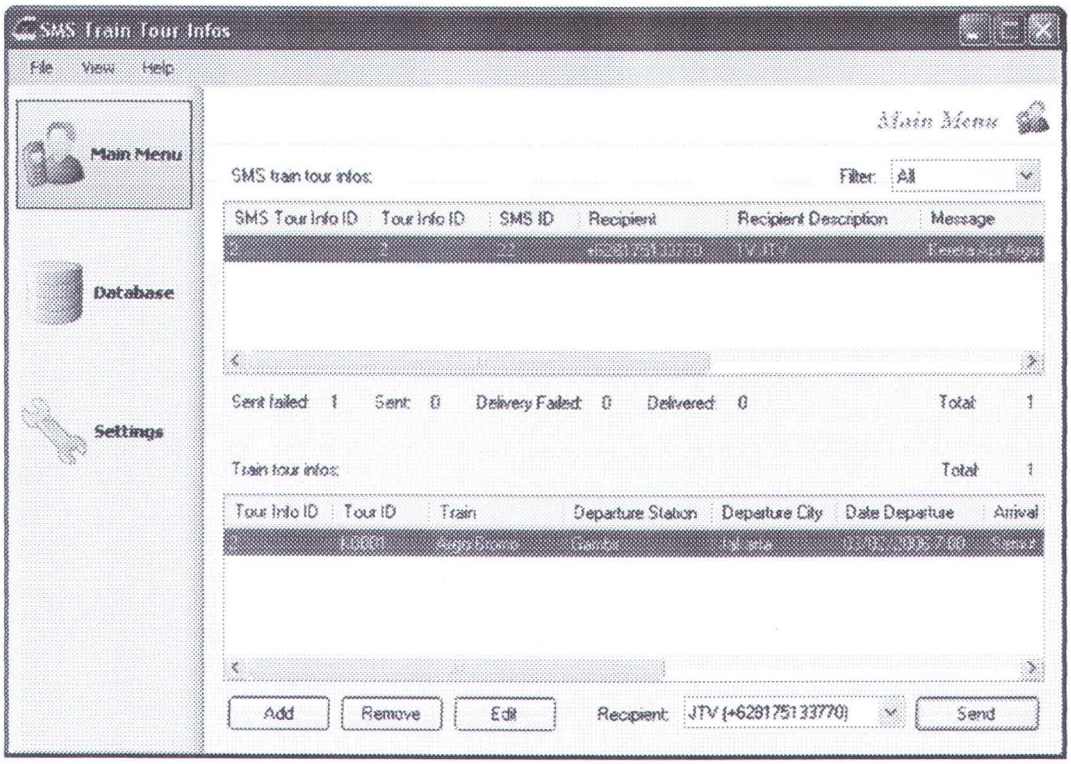
Gambar 5.27 Antar muka pada saat tombol Send ditekan dan terjadi perubahan status report dari Sent menjadi Delivered

Pengguna juga dapat mengirimkan dua atau lebih jadwal kereta api sekaligus dengan cara mem-blok atau mensortir kedua jadwal, yaitu dengan menekan tombol *shift* yang terdapat pada keyboard dan klik kiri pada mouse kedua jadwal tersebut dan kemudian mengirimkannya dengan menekan tombol *Send*. Proses pengiriman untuk dua atau lebih jadwal kereta api sekaligus tertera pada gambar 5.28 dibawah ini



Gambar 5.28 Antar muka pengiriman dua jadwal kereta api sekaligus

Status report *Sent failed* akan terisi apabila pada saat pengiriman sms jadwal kereta api ada gangguan terhadap koneksi atau setting koneksi belum dilakukan. Gambar 5.29 dibawah ini menunjukkan kondisi dari *Sent failed* yang terisi akibat belum ada setting terhadap koneksi gateway.



Gambar 5.29 Antar muka pada saat Sent failed terjadi

Status report *Delivery failed* akan terisi apabila pada saat pengiriman sms jadwal kereta api terdapat gangguan pada pihak yang dituju, dimana banyak sekali kemungkinan gangguan yang terjadi diantaranya yaitu provider dari pihak yang dituju mengalami gangguan sistem jaringan sehingga terjadi penundaan terhadap sms yang masuk, atau alat komunikasi pada pihak yang dituju dalam kondisi mati sehingga tidak memungkinkan untuk dapat menerima sms. Pada umumnya status *Delivery failed* ini bersifat sementara dan akan mengalami perubahan menjadi *Delivered* disaat kondisi jaringan provider nomor sms tujuan sudah stabil, atau alat komunikasi sudah diaktifkan kembali.

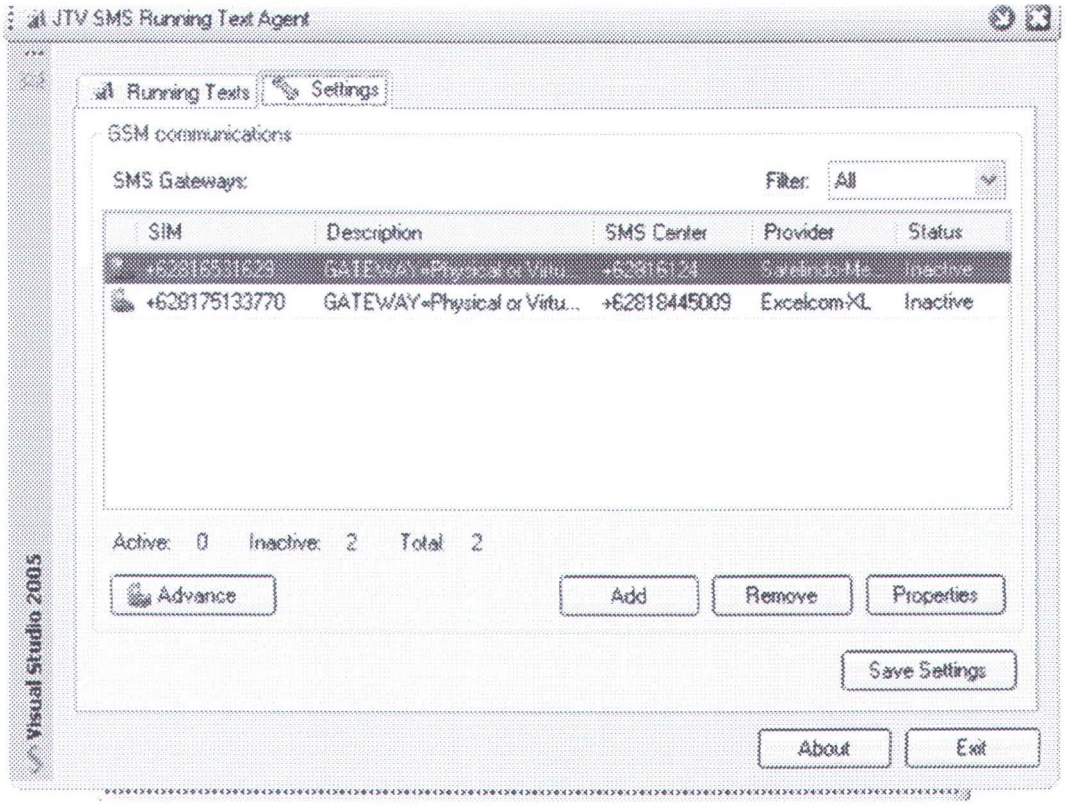
5.2.4 Skenario Proses Manajemen Koneksi sms TV

Skenario keempat ini digunakan untuk menguji apakah aplikasi bisa menjalankan proses koneksi sms gateway dengan baik, termasuk melakukan penyaringan sms yang masuk untuk menghindari hal yang tidak diinginkan. Pada uji coba proses skenario ini peralatan komunikasi yang digunakan untuk koneksi sms gateway adalah ponsel C35, namun pada implementasi sesungguhnya yaitu uji coba yang dilaksanakan pada PT. JTV (Jawa Timur Televisi) peralatan komunikasi yang digunakan untuk koneksi sms gateway adalah wavecom. Skenario Proses Manajemen Koneksi sms TV juga dapat dilihat pada bagian Lampiran yaitu pada tabel 5.4

Pada Skenario keempat ini akan dibagi menjadi beberapa sub skenario sesuai dengan yang terdapat pada menu *Settings*.

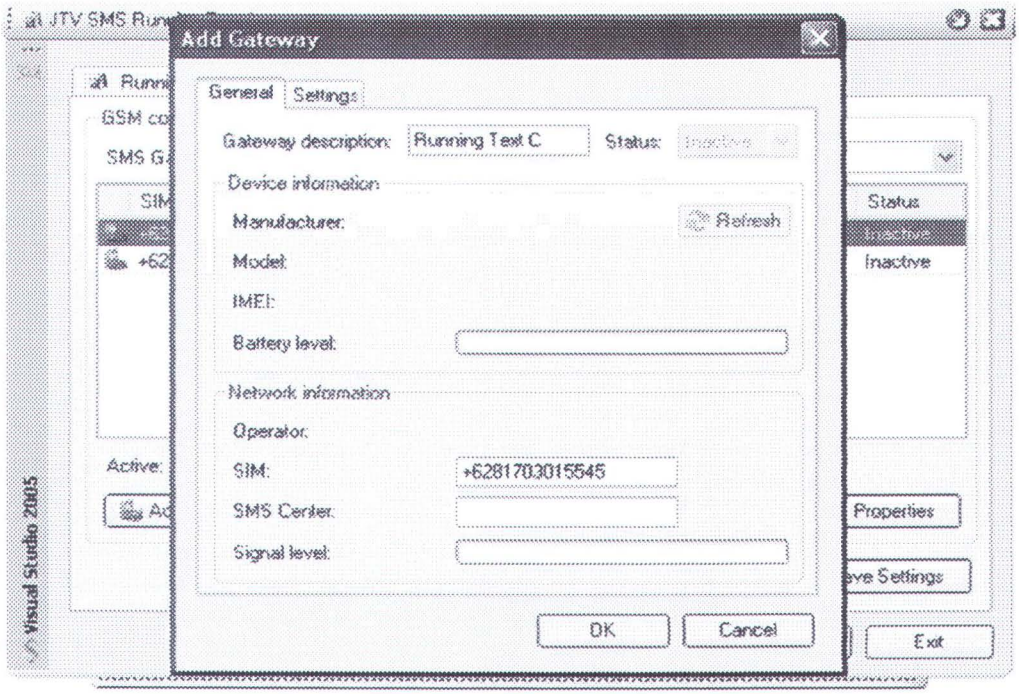
5.2.4.1 Koneksi Gateway

Pertama, untuk memanajemen koneksi gateway, pengguna harus memilih menu *Settings* pada menubar. Tampilan yang muncul seperti terlihat pada gambar 5.30 berikut.



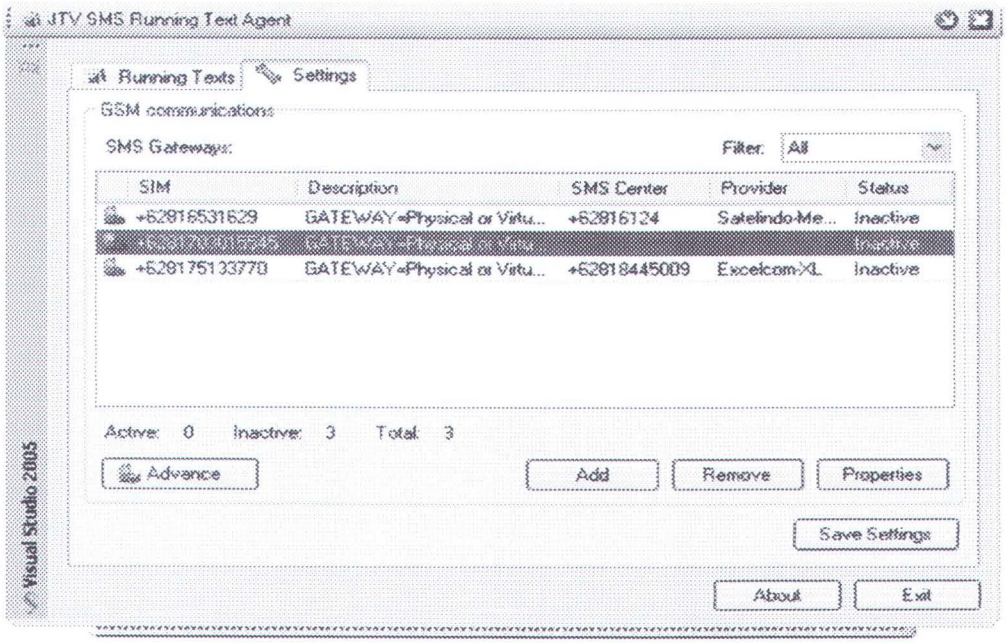
Gambar 5.30 Tampilan menu Settings

Pengguna dapat menambah daftar koneksi gateway dengan cara menekan tombol *Add* dan setelah itu pengguna mengisi *Gateway description* dimana penamaannya adalah bebas, dalam hal ini sebagai contoh untuk gateway yang ketiga diberi nama *Running Text C*. Pengguna juga diwajibkan untuk mengisi nomor SIM sesuai dengan nomor yang terdapat di dalam ponsel C35. Terakhir yang harus dilakukan oleh pengguna adalah menekan tombol *OK*, apabila pengisian sudah selesai dilakukan. Tampilan setelah tombol *Add* ditekan adalah seperti pada gambar 5.31 berikut.



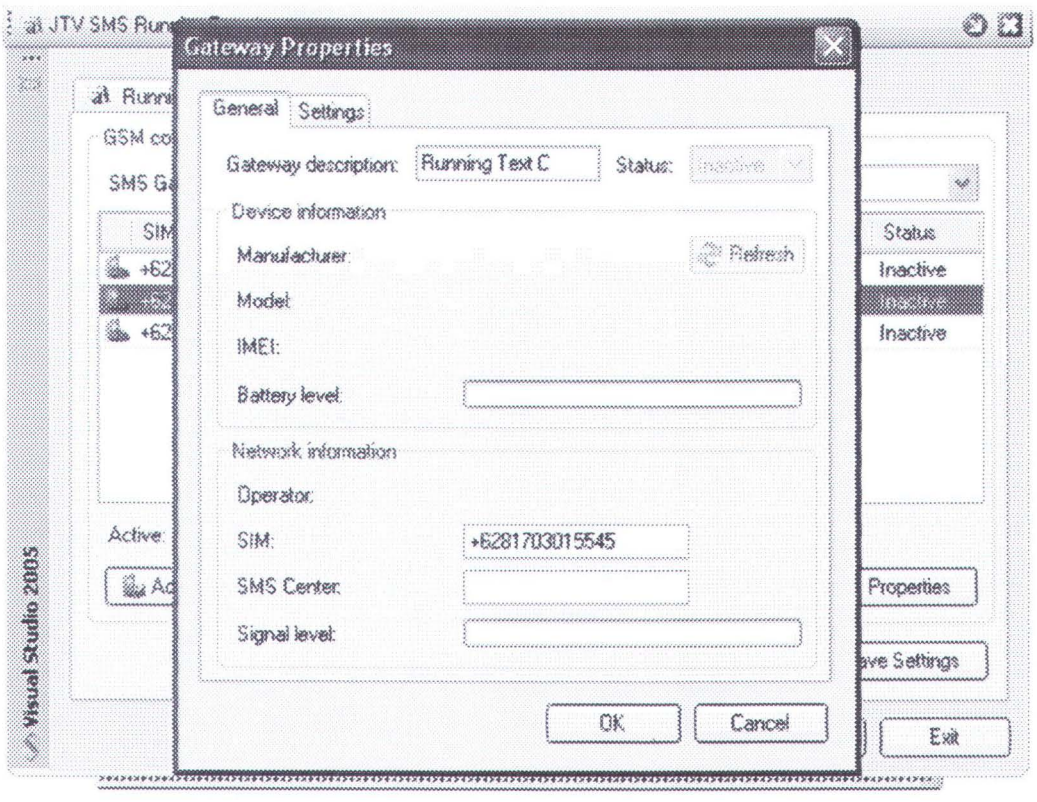
Gambar 5.31 Antar muka setelah tombol add ditekan

Daftar koneksi gateway akan bertambah satu sesuai dengan yang tertera pada gambar 5.32 dibawah ini



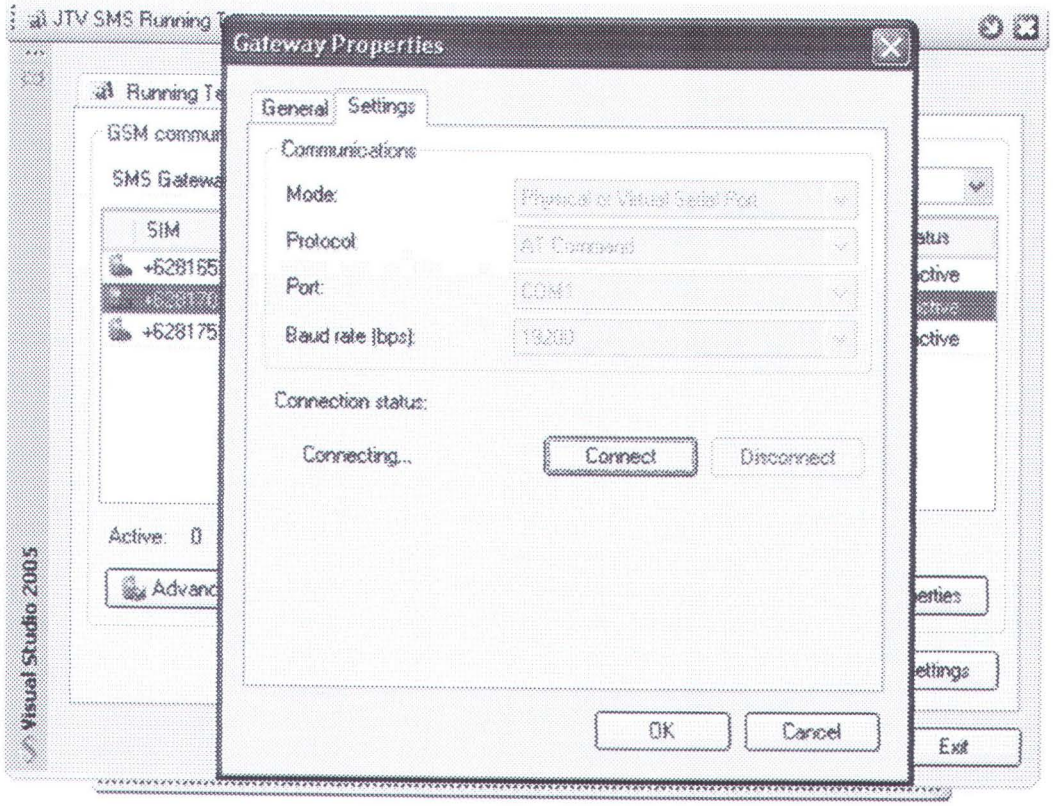
Gambar 5.32 Menu Settings setelah mengalami penambahan koneksi gateway

Selanjutnya apabila ingin mengaktifkan koneksi gateway yang ada dalam daftar, langkah pertama adalah pengguna memilih salah satu diantara tiga koneksi yang ada dan kemudian menyortirnya, sebagai contoh apabila yang akan diaktifkan adalah koneksi yang barusan ditambahkan, maka pengguna mensortir koneksi gateway tersebut dan kemudian tekan tombol *Properties*, yang juga bisa dilakukan dengan menekan klik kanan mouse pada koneksi gateway yang disortir dan kemudian memilih menu *Properties*, tampilannya dapat dilihat pada gambar 5.33 berikut.



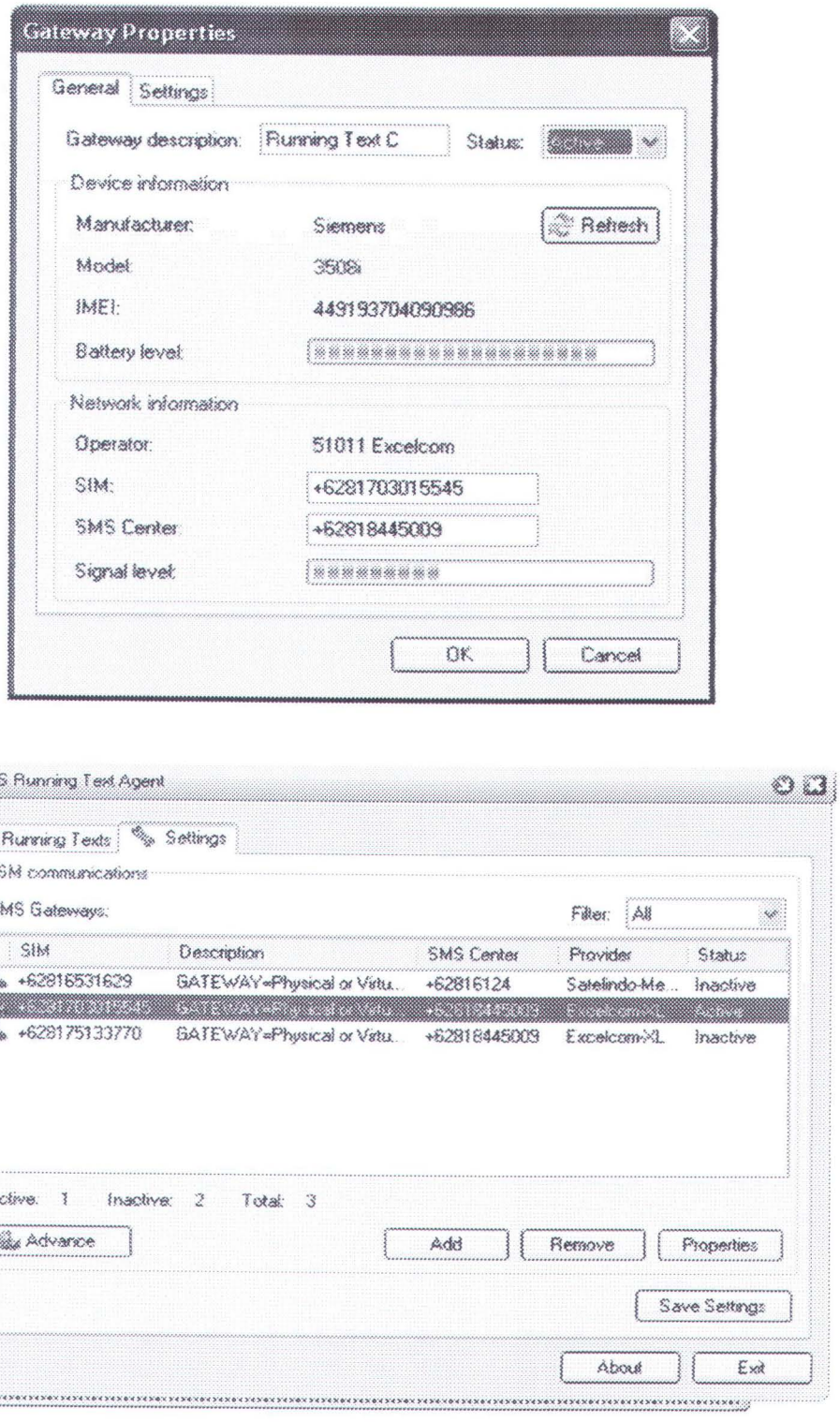
Gambar 5.33 Antar muka pemilihan menu Properties

Pada form *Gateway Properties* pilih menu *Settings* dan setelah itu tekan tombol *Connect* untuk mengkoneksikan gateway, tampilan yang muncul seperti pada gambar 5.34 berikut.



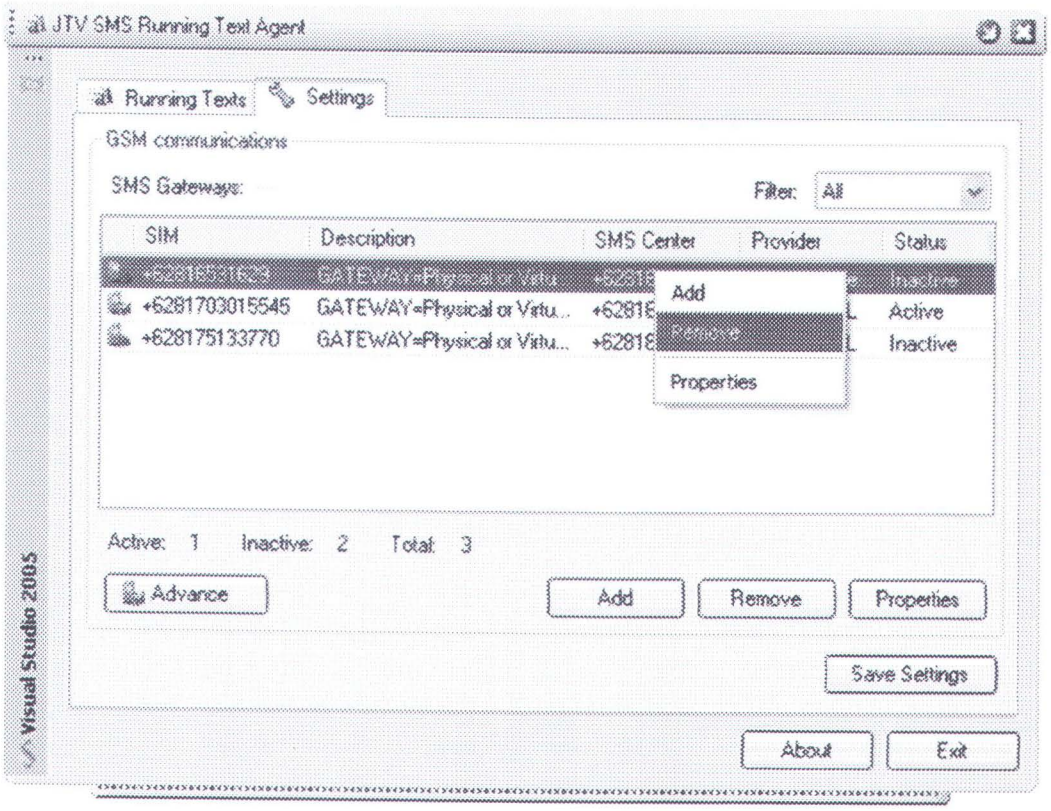
Gambar 5.34 Antar muka tombol Connect ditekan

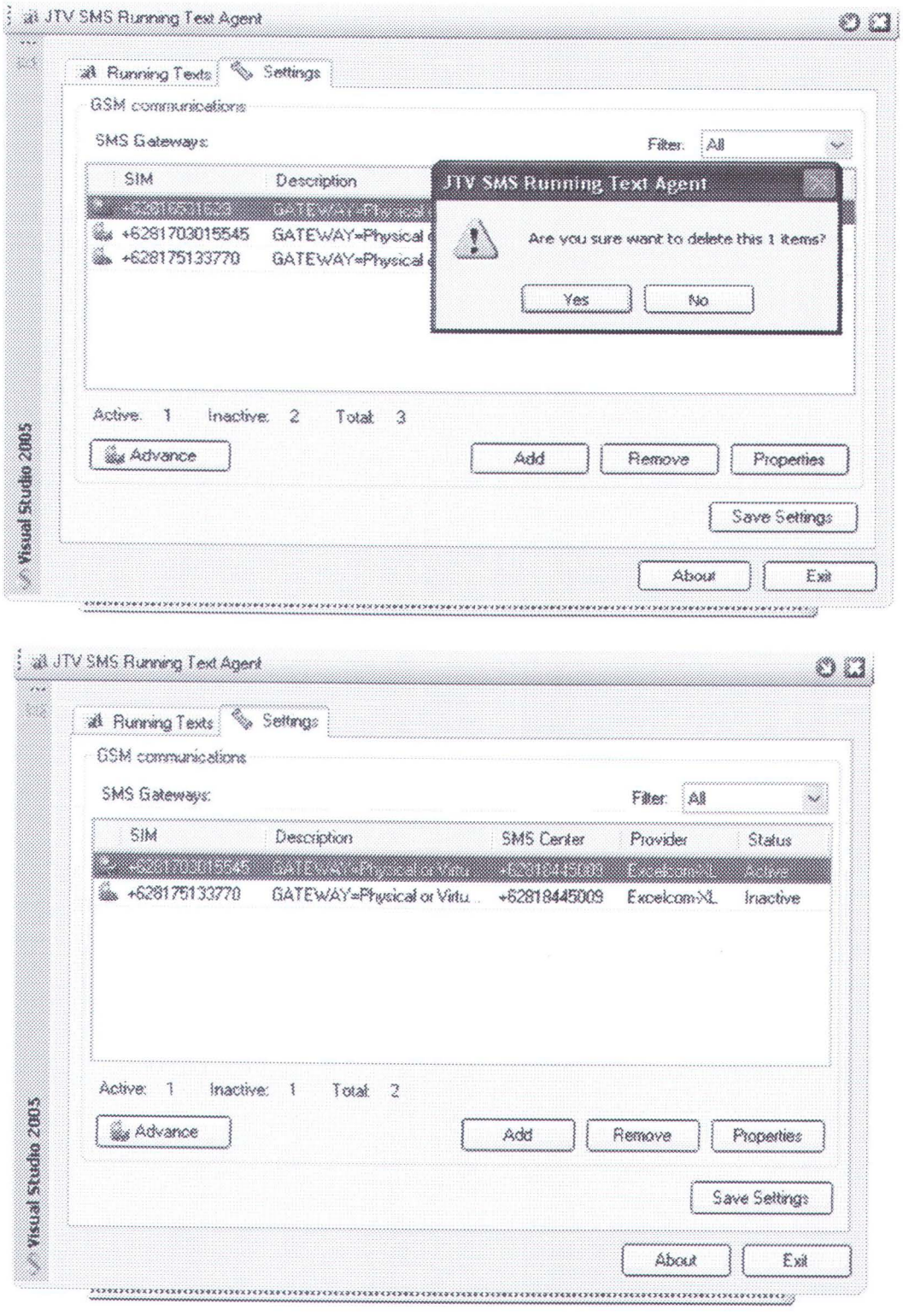
Default yang terjadi pada saat proses koneksi telah selesai dilakukan adalah kembali ke menu *General* dan setelah itu agar koneksi dapat dipakai maka dipilih *active* pada menu combo box *Status*. Terakhir pengguna menekan tombol OK untuk menyetujui setting koneksi yang telah dilakukan agar dapat dipakai, maka tampilan akan kembali pada menu awal *Settings* seperti tampak pada gambar 5.35 berikut.



Gambar 5.35 Antar muka koneksi gateway yang telah diaktifkan

Pengguna dapat menghapus koneksi gateway yang mungkin sudah tidak diperlukan kembali dengan cara menyortir terlebih dahulu koneksi yang ingin dihapus, setelah itu pilih menu *Remove* atau dapat juga dilakukan dengan klik kanan mouse pada koneksi yang telah disortir dan pilih menu *Remove*. Proses penghapusan koneksi dapat terlihat jelas pada gambar 5.36 dibawah ini

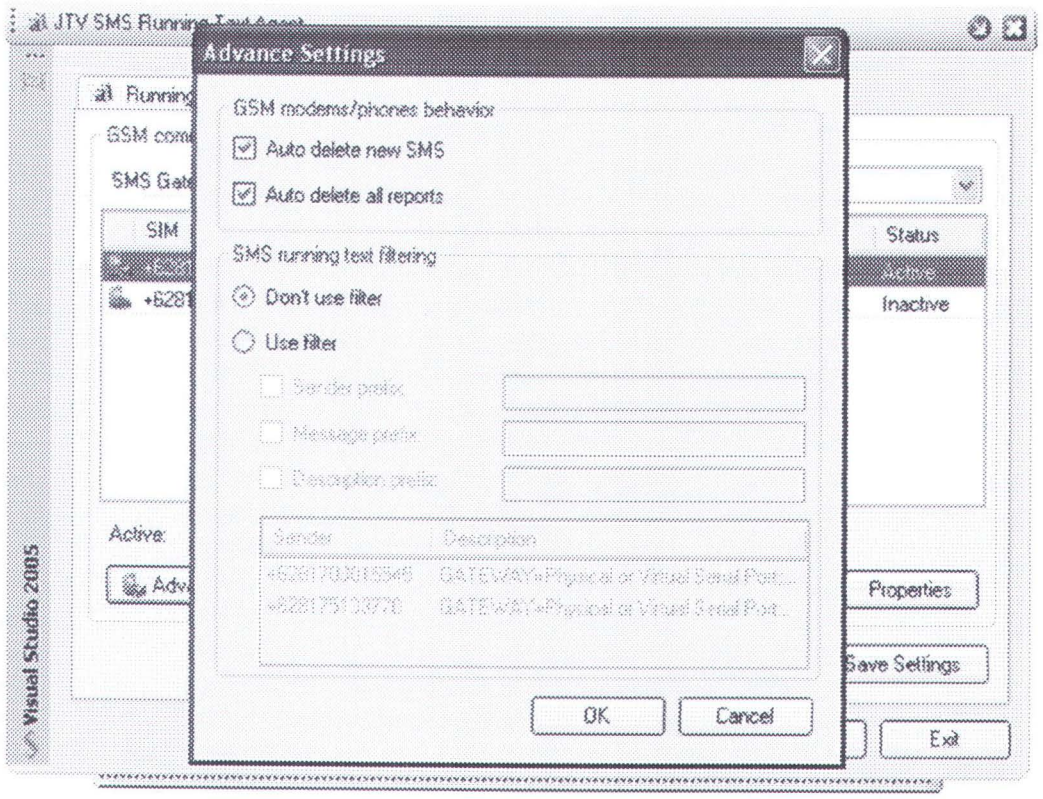




Gambar 5.36 Antar muka proses penghapusan koneksi

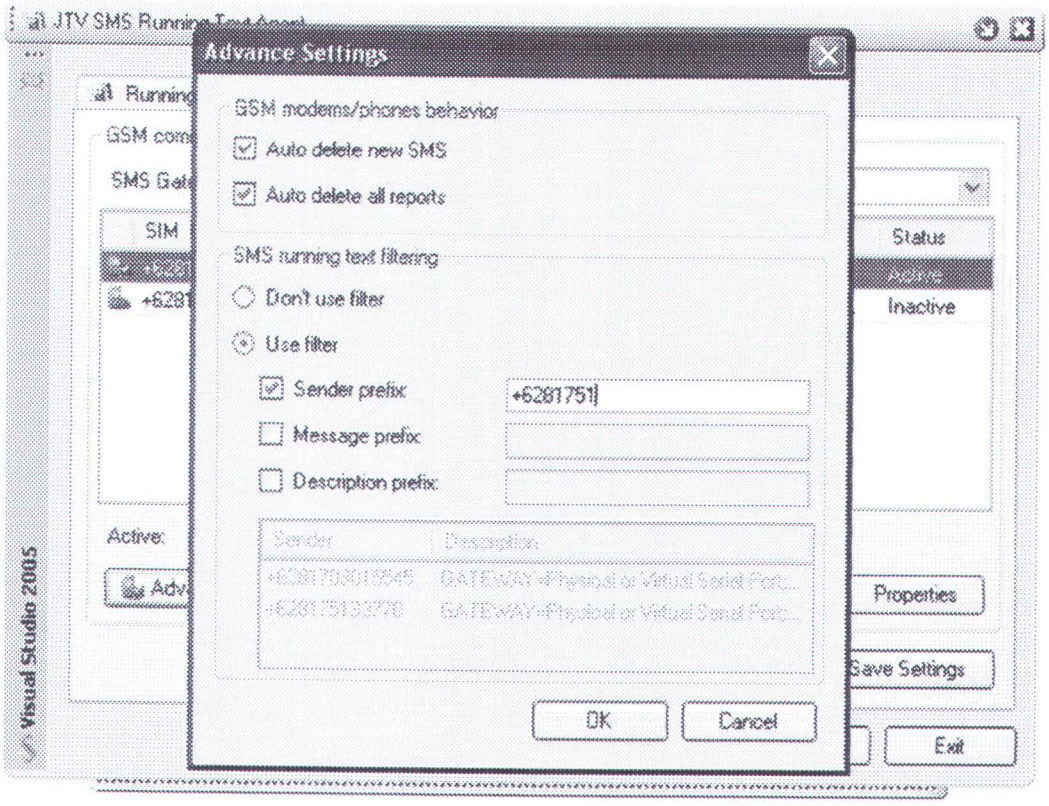
5.2.4.2 Penyaringan SMS yang masuk

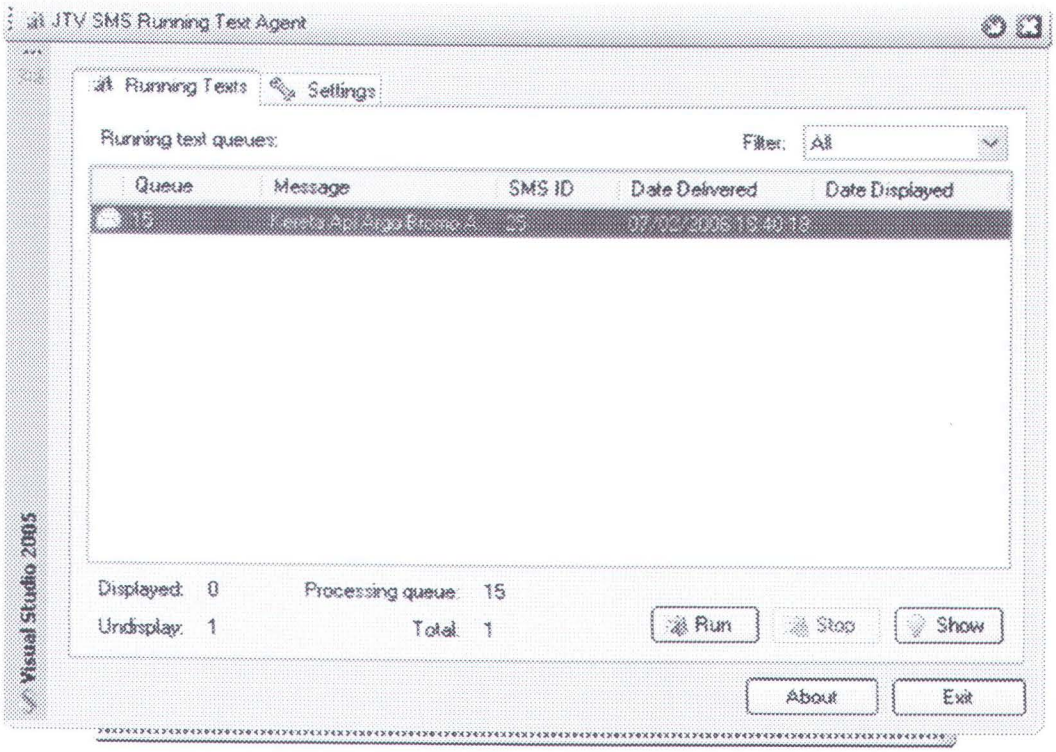
Kedua, untuk melakukan penyaringan terhadap sms yang masuk, pengguna harus memilih menu *Settings* pada menubar. Dan memilih tombol *Advance*, Dimana di dalam menu *Advance* ini pengguna dapat mengatur untuk batasan sms yang masuk, berdasarkan tiga kriteria batasan sesuai dengan yang terdapat pada aplikasi dalam menu *Advance*. Tiga kriteria batasan tersebut adalah *Sender prefix, Message prefix, Description prefix*. Apabila pengguna tidak menginginkan adanya penyaringan terhadap sms yang masuk, maka pengguna dapat memilih untuk mengaktifkan mode *Don't use filter*. Tampilan yang muncul seperti terlihat pada gambar 5.37 berikut



Gambar 5.37 Antar muka Menu Advance yang tidak menggunakan filter sms

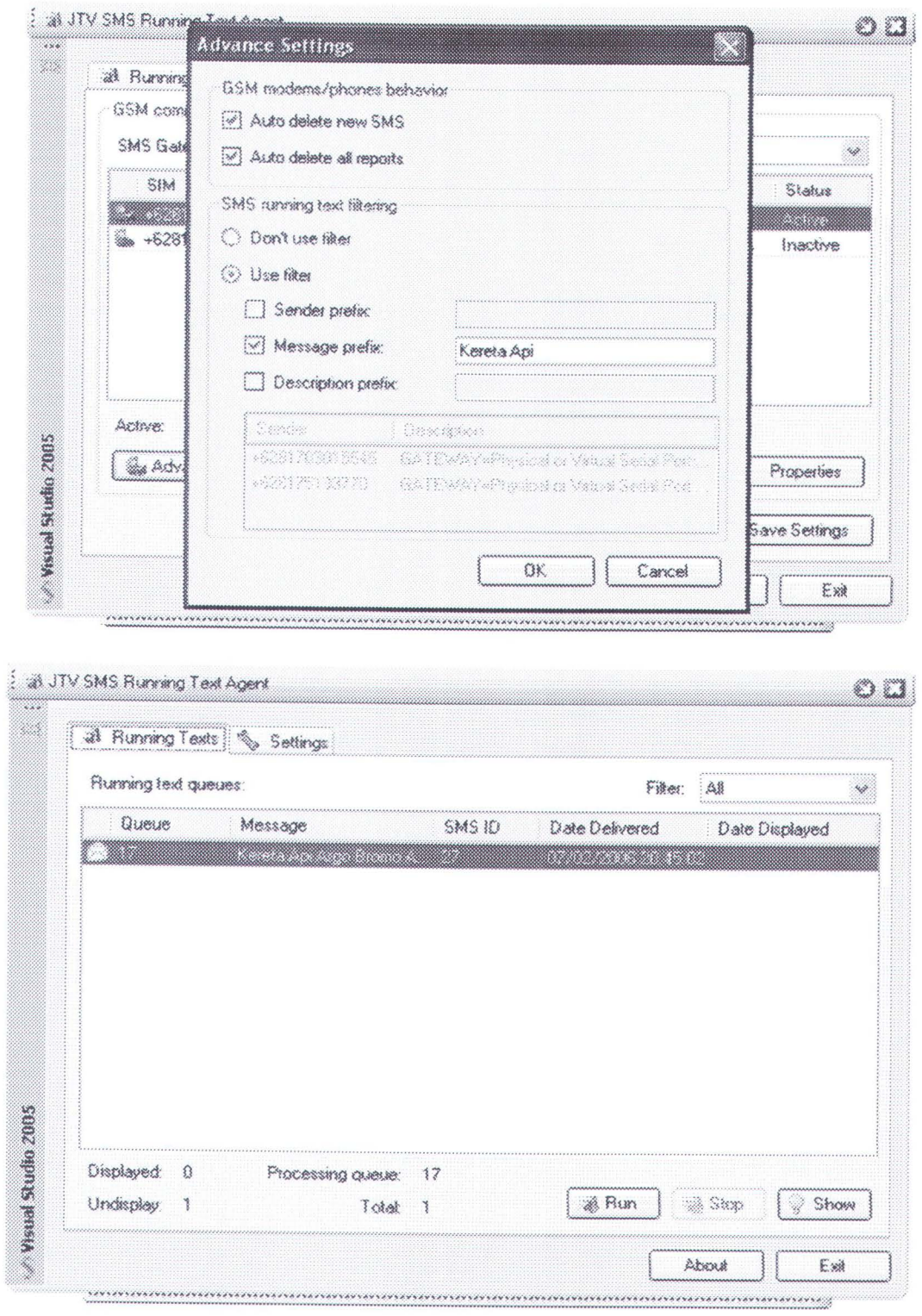
Apabila pengguna ingin menyaring sms yang masuk berdasarkan nomor awal atau nomor keseluruhan dari pihak pengirim, maka beri tanda pada bagian *Sender prefix* dan kemudian mengisikan nomor awal atau nomor keseluruhan yang digunakan sebagai batasan untuk sms yang masuk. Sebagai contoh gambar dibawah ini yang menggunakan *Sender Prefix* untuk menyaring sms yang masuk, yaitu menggunakan nomor awal dari pihak pengirim (+6281751), dimana sms yang bisa masuk hanyalah sms yang berasal dari nomor awal (+6281751). Untuk lebih jelasnya akan diterangkan pada gambar 5.38 dibawah ini.





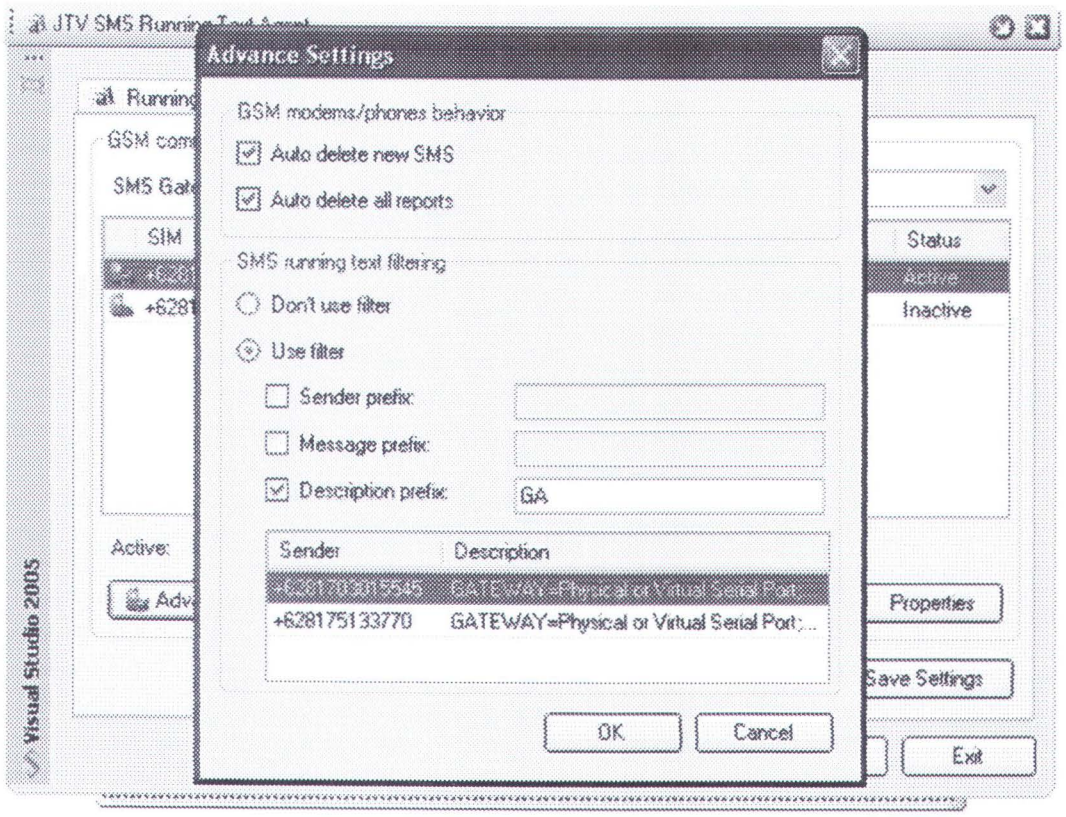
Gambar 5.38 Antar muka Menu Advance yang menggunakan Sender Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan

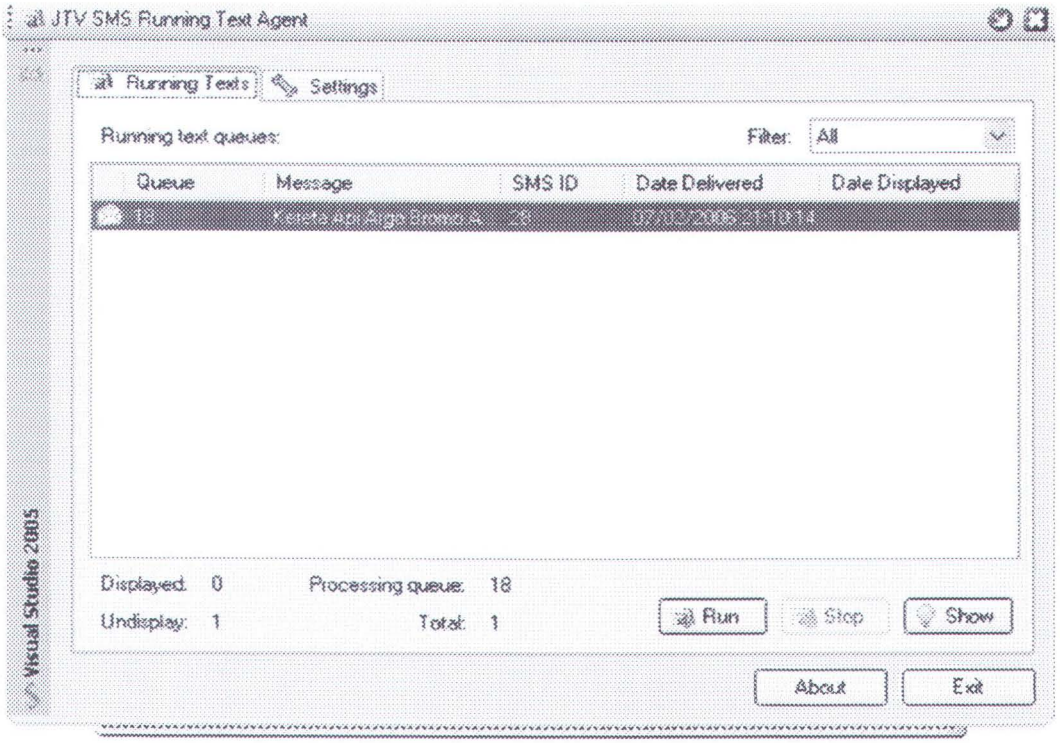
Pengguna ingin menyaring sms yang masuk berdasarkan kata kunci pesan, maka beri tanda pada bagian *Message prefix* dan kemudian mengisikan kata kunci pesan yang digunakan sebagai batasan untuk sms yang masuk. Sebagai contoh gambar dibawah ini yang menggunakan *Message Prefix* untuk menyaring sms yang masuk, yaitu menggunakan kata kunci pesan Kereta Api, dimana pesan sms yang bisa masuk hanyalah pesan yang kata awalnya adalah Kereta Api. Untuk lebih jelasnya akan diterangkan pada gambar 5.39 dibawah ini



Gambar 5.39 Antar muka Menu Advance yang menggunakan Message Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan

Pengguna ingin menyaring sms yang masuk berdasarkan deskripsi, maka beri tanda pada bagian *description prefix* dan kemudian akan muncul beberapa deskripsi yang ada yang telah diisikan sebagai contoh dan deskripsi tersebut nantinya akan berguna sebagai batasan untuk sms yang masuk. Gambar 5.40 dibawah ini menjelaskan penyaringan sms yang menggunakan *Description Prefix* dimana sms yang bisa masuk hanyalah sms yang sesuai dengan deskripsi yang ada.





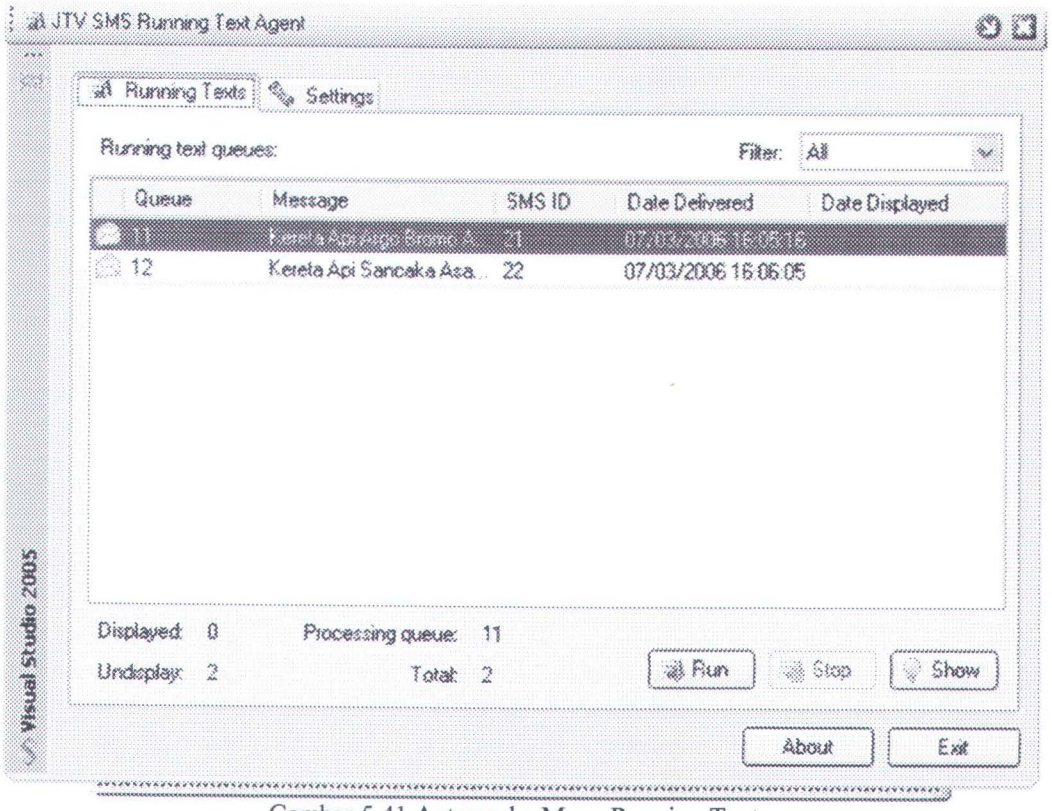
Gambar 5.40 Antar muka Menu Advance yang menggunakan Description Prefix untuk penyaringan sms yang masuk dan sms yang berhasil masuk dan telah mengalami proses penyaringan

5.2.5 Skenario Proses Manajemen SMS TV

Skenario kelima ini digunakan untuk menguji apakah aplikasi bisa melakukan penerimaan sms jadwal transportasi yang telah dikirim oleh operator perusahaan kereta api dan operator perusahaan pesawat terbang, sekaligus memunculkan sms jadwal transportasi tersebut dalam bentuk tulisan berjalan atau *running text* dengan baik. Skenario Proses Manajemen SMS TV juga dapat dilihat pada bagian Lampiran yaitu pada tabel 5.5

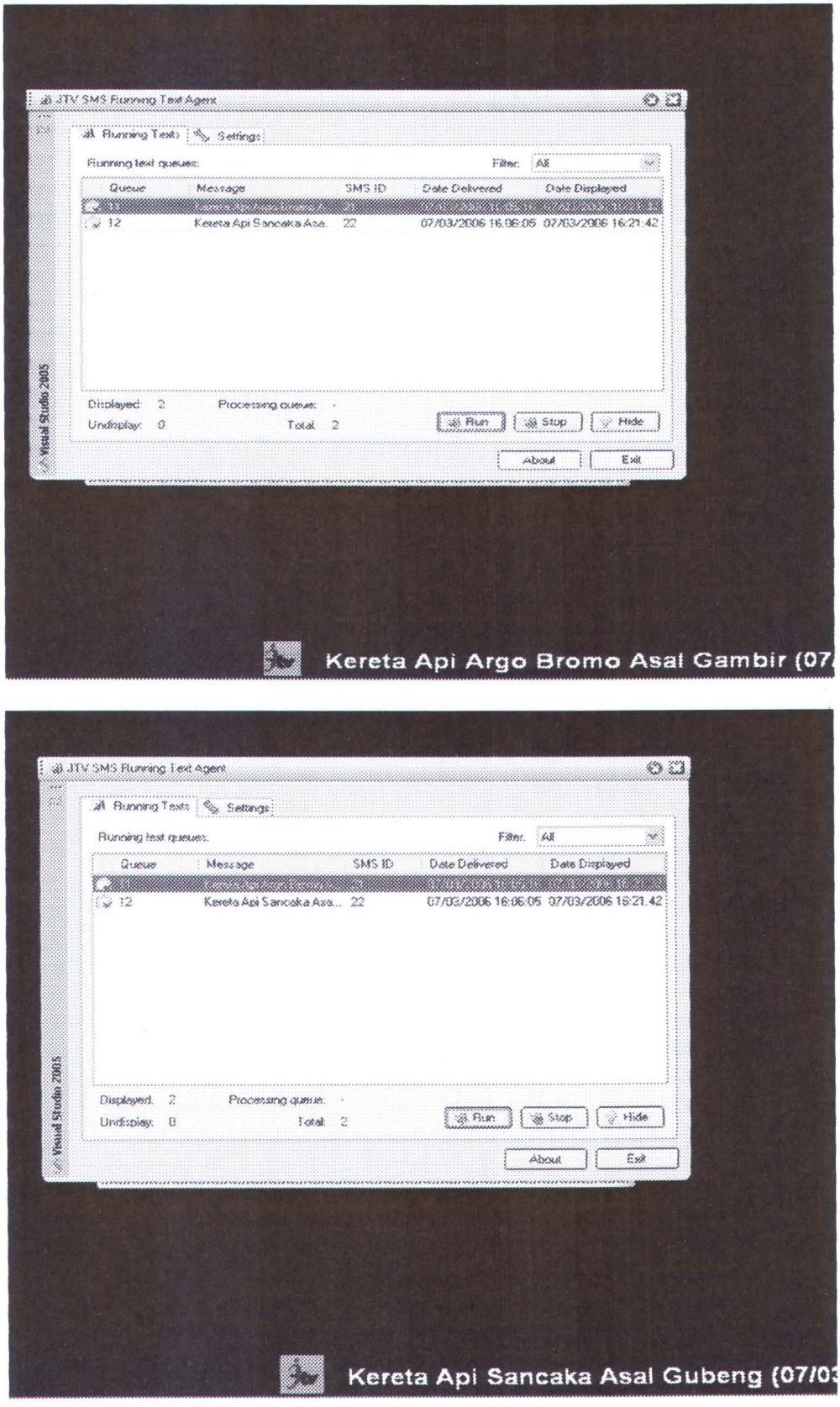
Pertama, untuk mengetahui daftar sms yang masuk, pengguna memilih menu *Running Texts* pada menubar. Tampilan yang muncul seperti pada gambar 5.41 berikut.

Gambar dibawah ini menunjukkan *running text* yang akan ditampilkan pada layar televisi. Background berwarna hitam adalah desktop pada komputer. Running text tersebut hanya terlihat setengah bagian vertikal pada desktop karena terjadi pembagian jumlah screen terhadap tampilan pada desktop dan tampilan pada layar televisi, dimana satu ukuran screen penuh yaitu 1024 x 768 dibagi menjadi dua, sehingga apabila pada desktop *running text* yang tampil adalah setengah bagian vertikal ke kanan, maka pada bagian layar televisi running text tersebut akan ditampilkan secara penuh, dimana running text akan berjalan dari pojok sebelah kanan bawah televisi, sampai dengan bagian pojok kiri bawah televisi, dan kemudian menembus ke desktop, dimana akan muncul dari bagian pojok kanan bawah desktop sampai dengan bagian tengah, dan berputar kembali untuk tampil pada layar televisi.



Gambar 5.41 Antarmuka Menu Running Texts

Kemudian pengguna dapat melakukan perintah terhadap aplikasi, dengan menekan tombol *Run* yang berfungsi untuk menjalankan text sms yang masuk. Seperti yang terlihat pada (gambar 5.41) diatas, apabila tombol *Run* ditekan maka pesan dengan nomor antrian 11 akan dijalankan terlebih dahulu, karena posisi sortir berada pada nomor antrian 11, setelah itu akan secara otomatis menjalankan pesan dengan nomor antrian 12. Pada Saat tombol *Run* ditekan akan berganti menjadi tombol *Pause*, yang apabila ditekan akan menghentikan jalannya tulisan tetapi tidak menghilangkan *running text* yang tampil pada desktop maupun pada layar televisi. Tombol *Stop*, digunakan untuk menghentikan sekaligus menghilangkan *running text* yang tampil pada desktop dan layar televisi. Tombol *Show* dan *Hide* digunakan untuk menayangkan dan menyembunyikan *running text* yang sedang berjalan pada desktop dan layar televisi.



Gambar 5.42 Antarmuka tampilan Running Text

Jumlah sms yang ditayangkan pada *running test* maksimal adalah 10 (sepuluh) sms, apabila jumlah sms yang masuk pada menu *Running Texts* lebih dari sepuluh, contoh sms kesebelas, maka sms yang pertama akan hilang dan yang ditampilkan adalah sms kedua sampai dengan sms kesebelas.

5.3 Kesimpulan Uji Coba

Dari sejumlah uji coba yang dilakukan, secara umum dapat disimpulkan bahwa aplikasi pemanfaatan sms untuk menyampaikan informasi pembatalan dan keterlambatan jadwal Pesawat Terbang dan Kereta Api melalui saluran televisi ini dapat berjalan sesuai dengan tujuan yang telah dirumuskan. Proses manajemen koneksi, manajemen data, jadwal dan manajemen sms tv serta penanganan delay sms berjalan sesuai dengan fungsinya.

BAB VI

PENUTUP

BAB VI

PENUTUP

Setelah dilakukan pembahasan teori, perancangan sistem, implementasi dan pengujian terhadap sistem pada bab-bab sebelumnya, maka pada bab penutup ini akan diambil kesimpulan dari Tugas Akhir " Pemanfaatan SMS Untuk Menyampaikan Informasi Pembatalan dan Keterlambatan Jadwal Pesawat Terbang dan Kereta Api Melalui Saluran Televisi".

6.1 Kesimpulan

1. Setelah melakukan uji coba dan evaluasi pada tugas akhir ini maka dapat diambil kesimpulan sebagai berikut. Aplikasi yang dibuat pada tugas akhir ini telah berhasil melakukan proses pengujian pengaturan koneksi gateway dengan baik sesuai dengan skenario proses manajemen koneksi sms kereta.
2. Aplikasi yang dibuat pada tugas akhir ini telah berhasil melakukan proses pengujian pada proses melihat data, menambah data, mengedit data, dan menghapus data kereta api, data stasiun kereta api, data jurusan kereta api dengan baik sesuai dengan skenario proses manajemen data kereta api.
3. Aplikasi yang dibuat pada tugas akhir ini telah berhasil melakukan proses pengujian dengan baik pada proses menambah data, mengedit data, dan menghapus jadwal kereta api lengkap dengan status dan informasi

mengenai status tersebut, dan juga proses pengiriman sms jadwal transportasi sesuai dengan skenario pada manajemen data kereta api.

4. Aplikasi yang dibuat pada tugas akhir ini telah berhasil melakukan proses pengujian dengan baik pada proses koneksi sms gateway, termasuk melakukan penyaringan sms yang masuk untuk menghindari hal yang tidak diinginkan sesuai dengan skenario proses manajemen koneksi sms TV.
5. Aplikasi yang dibuat pada tugas akhir ini telah berhasil melakukan penerimaan sms jadwal transportasi yang telah dikirim oleh operator perusahaan kereta api dan operator perusahaan pesawat terbang dan pengujian pemunculan sms jadwal transportasi dalam bentuk tulisan berjalan atau *running text* dapat berfungsi dengan baik sesuai dengan skenario proses manajemen sms televisi.
6. Dari uji coba terhadap aplikasi yang dibuat pada tugas akhir ini telah terbukti dapat menjalankan proses perubahan jadwal kereta api dan pesawat terbang yang mengalami pembatalan dan juga keterlambatan, secara *up to date*.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

[QUA98] Quatrani, Terry, Visual Modelling with Rational Rose and UML, Massachusetts : Addison Wesley Longman,1998

[MAT06] Matrox 2006, www.matrox.com,april 2006, for broadcast television

[LEA06] Creating a Graphical User Interface in VB.NET www.learnvisualstudio.**net**

[MAH05] Mahesh Chand , GDI+ in **VB.NET Tutorial**

LAMPIRAN

LAMPIRAN

TABEL PROSES SKENARIO UJI COBA PERANGKAT LUNAK

Tabel 5.1 Skenario Proses Manajemen Koneksi sms Kereta

No.	Tujuan	Prosedur	Data Input	Hasil Yang diharapkan	Hasil yang diperoleh
1.	Untuk menguji apakah aplikasi bisa menjalankan proses koneksi gateway dengan baik	Sebelum masuk ke sistem, pengguna harus melakukan pengecekan koneksi terhadap komputer dengan gateway yang tersedia melalui kabel serial	Ponsel C35	Kabel serial terpasang dengan ponsel C35 sebagai gateway	Terpasang
2.		Dari menu *Settings* dipilih menu *Mode* untuk menentukan bentuk koneksi yang sesuai dengan karakteristik ponsel C35	Dipilih *Physical or Virtual Serial Port* pada menu *Mode*	Koneksi gateway dari ponsel C35 terhubung melalui kabel serial terhadap port yang ada pada komputer	Status OK (gambar 5.2)
3.		Pemilihan menu *Protocol* yang berfungsi untuk menentukan aplikasi yang digunakan untuk melakukan komunikasi dengan terminal (ponsel)	*AT Command, Nokia Fbus,* dan *Nokia Mbus*	Dipilih yang sesuai dengan karakter ponsel C35	Bentuk perintah *AT Command* (gambar 5.3)
4.		Pemilihan menu *Port*	Banyak pilihan port yang terdapat pada menu *Port* dalam aplikasi	Port pada komputer yang dipilih harus sesuai dengan yang dipakai oleh kabel serial dalam menghubungkan komputer dengan ponsel	Port komputer yang digunakan adalah COM1 (gambar 5.4)
5.		Mengatur baudrate (bps) pada menu *Baud rate*	Ukuran baud rate	Ukuran baud rate yang sesuai untuk ponsel C35	19200 bps (gambar 5.5)
6.		Pengisian bagian *Network information*	Mengisi field *SIM*	Nomor *SIM* yang diisikan harus sesuai dengan SIM yang terdapat di dalam ponsel	Sesuai dengan gambar 5.6
7.		Menekan button *Connect* agar koneksi bisa terjadi	Tombol *Connect*	Koneksi bisa terjadi	Sesuai dengan gambar 5.8
8.		Penyimpanan setting koneksi	Tombol *Save settings*	Setting koneksi dapat disimpan dan akan disimpan terhadap sebuah file yang bernama *settings.ini*	Sesuai dengan gambar 5.9

Tabel 5.2 Skenario Proses Manajemen Data Kereta Api

No.	Tujuan	Prosedur	Data Input	Hasil Yang diharapkan	Hasil yang diperoleh
1.	Untuk menguji apakah aplikasi bisa melakukan proses melihat data, menambah data, mengedit data, dan menghapus data kereta api	Memilih menu *Database* yang berada pada menu sebelah kiri atau menubar bagian tengah	Klik menu *Database*	Form menu *Database* dapat ditampilkan	Sesuai dengan gambar 5.10
2.		Menambahkan dan mengedit data kereta api	Menekan tombol *Add* atau *Edit*	Untuk menambahkan dan melakukan pengubahan terhadap data kereta api	Sesuai dengan gambar 5.11
3.		Menyetujui proses tambah dan ubah data kereta api	Tekan tombol *OK*	Proses penambahan dan pengubahan data dapat dilakukan	Sesuai dengan gambar 5.12
4.		Melakukan proses penghapusan terhadap data kereta api	Tekan *Remove*	Data kereta api dapat terhapus	Sesuai dengan gambar 5.13
5.	Untuk menguji apakah aplikasi bisa melakukan proses melihat data, menambah data, mengedit data, dan menghapus data stasiun kereta api	Memilih menu *Database* yang berada pada menu sebelah kiri atau menubar bagian tengah	Klik menu *Database*	Form menu *Database* dapat ditampilkan	Sesuai dengan gambar 5.14
6.		Menambahkan dan mengedit data stasiun kereta api	Menekan tombol *Add* atau *Edit*	Untuk menambahkan dan melakukan pengubahan terhadap data stasiun kereta api	Sesuai dengan gambar 5.15
7.		Menyetujui proses tambah dan ubah data stasiun kereta api	Tekan tombol *OK*	Proses penambahan dan pengubahan data dapat dilakukan	Sesuai dengan gambar 5.16
8.		Melakukan proses penghapusan terhadap data stasiun kereta api	Tekan *Remove*	Data stasiun kereta api dapat terhapus	Sesuai dengan gambar 5.17
9.	Untuk menguji apakah aplikasi bisa melakukan proses	Memilih menu *Database* yang berada pada menu sebelah kiri atau menubar	Klik menu *Database*	Form menu *Database* dapat ditampilkan	Sesuai dengan gambar 5.18

	melihat data, menambah data, mengedit data, dan menghapus data jurusan kereta api.	bagian tengah			
10.		Menambahkan dan mengedit data jurusan kereta api	Menekan tombol *Add* atau *Edit*	Untuk menambahkan dan melakukan pengubahan terhadap data jurusan kereta api	Sesuai dengan gambar 5.19
11.		Menyetujui proses tambah dan ubah data jurusan kereta api	Tekan tombol *OK*	Proses penambahan dan pengubahan data dapat dilakukan	Sesuai dengan gambar 5.20
12.		Melakukan proses penghapusan terhadap data jurusan kereta api	Tekan *Remove*	Data jurusan kereta api dapat terhapus	Sesuai dengan gambar 5.21

Tabel 5.3 Skenario Proses Manajemen Jadwal Kereta Api

No.	Tujuan	Prosedur	Data Input	Hasil Yang diharapkan	Hasil yang diperoleh
1.	Untuk menguji apakah aplikasi bisa melakukan penambahan, penghapusan, pengubahan jadwal kereta api dengan status dan informasi mengenai status yang terjadi pada kereta api	Memilih menu *Main Menu*	Tekan menubar Main Menu	Untuk memanajemen jadwal kereta api dan sekaligus melakukan proses pengiriman jadwal	Sesuai dengan gambar 5.22
2.	Untuk menguji apakah aplikasi dapat menjalankan proses kirim SMS dengan baik dan SMS tersebut sampai ke pihak stasiun televisi yaitu PT. JTV (Jawa Timur Televisi).	Menambahkan dan mengedit data jadwal kereta api	Tekan *ADD* dan *EDIT*	Untuk menambahkan dan melakukan pengubahan terhadap data jadwal kereta api	Sesuai dengan gambar 5.23
3.		Menyetujui proses tambah dan ubah data jadwal kereta api	Tekan *OK*	Proses penambahan dan pengubahan data dapat dilakukan	Sesuai dengan gambar 5.24
4.		Melakukan Proses penghapusan terhadap data jadwal kereta api	Tekan *Remove*	Data jadwal kereta api dapat terhapus	Sesuai dengan gambar 5.25
5.		Pemilihan nomor tujuan pengiriman jadwal sms	Pilih pada menu *Recipient*	Nomor tujuan sms terpilih	Sesuai dengan gambar 5.26
6.		Mengirimkan jadwal dalam bentuk sms	Tekan *Send*	Jadwal kereta api dapat terkirim pada nomor yang dikehendaki	Sesuai dengan gambar 5.27
7.		Mengirimkan dua atau lebih jadwal kereta api sekaligus	Mensortir kedua jadwal, kemudian Tekan *Send*	Kedua jadwal kereta api dapat terkirim pada nomor yang dikehendaki	Sesuai dengan gambar 5.28

Tabel 5.4 Skenario Proses Manajemen Koneksi sms TV

No.	Tujuan	Prosedur	Data Input	Hasil Yang diharapkan	Hasil yang diperoleh
1.	Untuk menguji apakah aplikasi bisa menjalankan proses koneksi sms gateway dengan baik, termasuk melakukan penyaringan sms yang masuk untuk menghindari hal yang tidak diinginkan	Memilih menu *Settings*	Klik menu *Settings* pada tab bar	Form menu *Settings* akan ditampilkan	Sesuai dengan gambar 5.30
2.		Menambah daftar koneksi gateway	Tekan tombol *ADD* , memberi nama gateway tersebut pada *Gateway description*, kemudian mengisi field *SIM* yang kosong, sesuai dengan nomor yang terdapat pada ponsel	Proses penambahan daftar koneksi gateway dapat dilakukan	Sesuai dengan gambar 5.31
3.		Menampilkan daftar koneksi gateway yang ditambahkan	Tekan tombol *OK*	Daftar koneksi gateway terbukti dapat tertambah	Sesuai dengan gambar 5.32
4.		Mengaktifkan koneksi gateway yang ada dalam daftar	Sortir salah satu gateway, kemudian tekan *Properties*	Form menu *Properties* dapat ditampilkan	Sesuai dengan gambar 5.33
5.		Pilih menu *Settings*	Tekan tombol *Connect*	Proses koneksi gateway dapat dilakukan	Sesuai dengan gambar 5.34
6.		Proses *Connect* terjadi, Default pada aplikasi adalah kembali ke menu *general*	Klik combo box *Status* dan pilih *active*, kemudian tekan tombol *OK*	Koneksi gateway berhasil dilakukan	Sesuai dengan gambar 5.35
7.		Menghapus koneksi gateway	Sortir salah satu gateway yang terdapat pada daftar gateway, kemudian tekan *Remove*	Penghapusan koneksi gateway berhasil dilakukan	Sesuai dengan gambar 5.36
8.		Tidak perlu menggunakan metode penyaringan sms yang masuk	Klik *Don't use filter*	Tidak menggunakan metode penyaringan sms yang masuk	Sesuai dengan gambar 5.37
9.		Penyaringan sms yang masuk berdasarkan nomor awal atau	Klik *Sender prefix*, kemudian isi field yang ada sesuai	Penyaringan sms berdasarkan *Sender prefix* dapat dilakukan	Sesuai dengan gambar 5.38

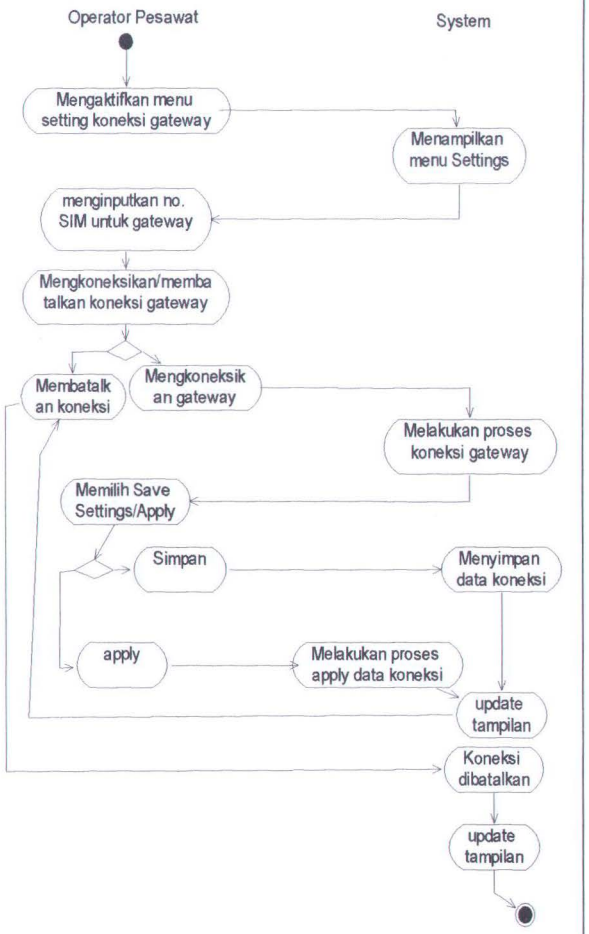
		nomor keseluruhan dari pihak pengirim	dengan ketentuan		
10.		Penyaringan sms yang masuk berdasarkan kata kunci pesan	Klik *Message prefix*, kemudian isi field yang ada sesuai dengan ketentuan	Penyaringan sms berdasarkan *Message prefix* dapat dilakukan	Sesuai dengan gambar 5.39
11.		Penyaringan sms yang masuk berdasarkan deskripsi	Klik *Description prefix*, isi field yang ada sesuai dengan ketentuan	Penyaringan sms berdasarkan *Description prefix* dapat dilakukan	Sesuai dengan gambar 5.40

Tabel 5.5 Skenario Proses Manajemen SMS TV

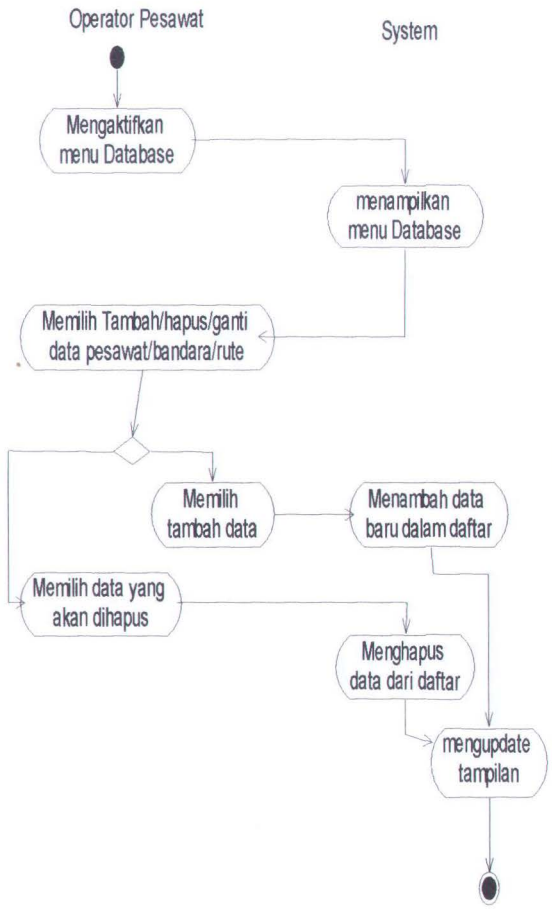
No.	Tujuan	Prosedur	Data Input	Hasil Yang diharapkan	Hasil yang diperoleh
1.	Untuk menguji apakah aplikasi bisa melakukan penerimaan sms jadwal transportasi yang telah dikirim oleh operator perusahaan kereta api dan operator perusahaan pesawat terbang	Memilih menu *Running Texts*	Klik menu *Running Texts* pada menu tab bar	Form menu *Running texts* dapat ditampilkan	Sesuai dengan gambar 5.41
2.	Untuk menguji pemunculan sms jadwal transportasi dalam bentuk tulisan berjalan atau *running text* dapat berfungsi dengan baik	Menjalankan text sms yang masuk	Tekan tombol *Run*	Text sms dapat berjalan sesuai dengan fungsinya	Uji coba terhadap aplikasi berdasarkan fungsi kinerjanya berhasil dilakukan, sesuai dengan gambar 5.42
3.		Menghentikan jalannya tulisan tetapi tidak menghilangkan *running text* yang tampil pada desktop maupun pada layar televisi	Tekan tombol *Pause*	Fungsi *Pause* dapat berjalan dengan baik	Uji coba terhadap aplikasi berdasarkan fungsi kinerjanya berhasil dilakukan, sesuai dengan gambar 5.42
4.		Menghentikan sekaligus menghilangkan *running text* yang tampil pada desktop dan layar televisi	Tekan tombol *Stop*	Fungsi *Stop* dapat berjalan dengan baik	Uji coba terhadap aplikasi berdasarkan fungsi kinerjanya berhasil dilakukan, sesuai dengan gambar 5.42
5.		Menayangkan dan menyembunyikan *running text* yang sedang berjalan pada desktop dan layar televisi	Tekan tombol *Show* dan *Hide*	Fungsi *Show* dan *Hide* dapat berjalan dengan baik	Uji coba terhadap aplikasi berdasarkan fungsi kinerjanya berhasil dilakukan, sesuai dengan gambar 5.42

LAMPIRAN ACTIVITY DIAGRAM USE CASE

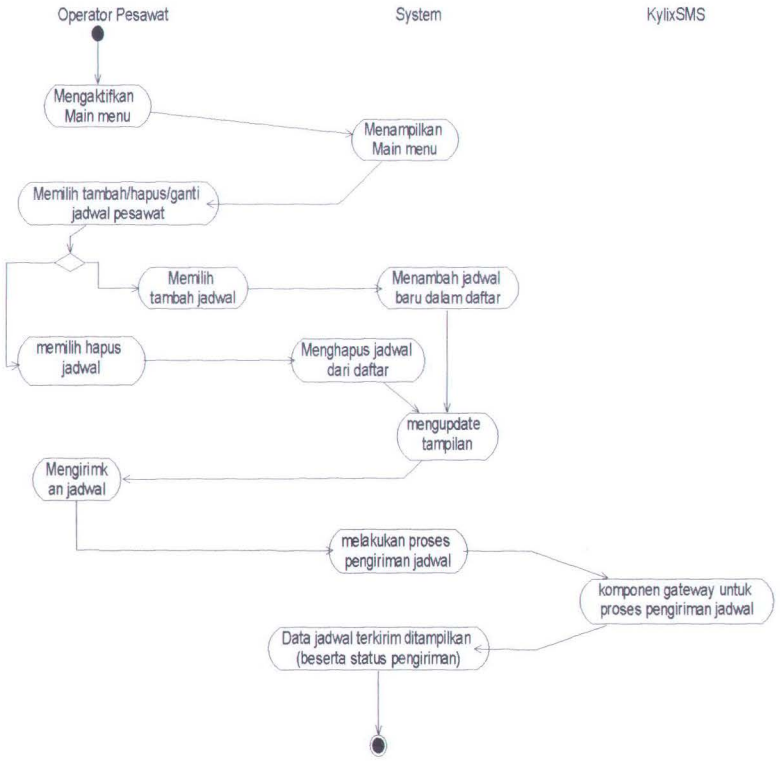
Manajemen Koneksi SMS Pesawat



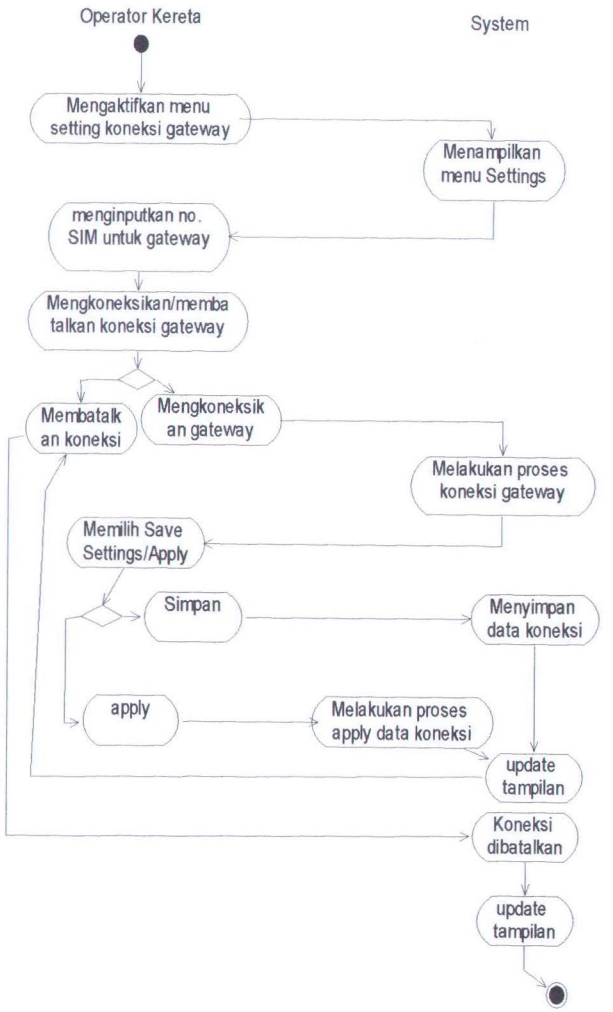
Manajemen Data Pesawat



Manajemen Jadwal Pesawat



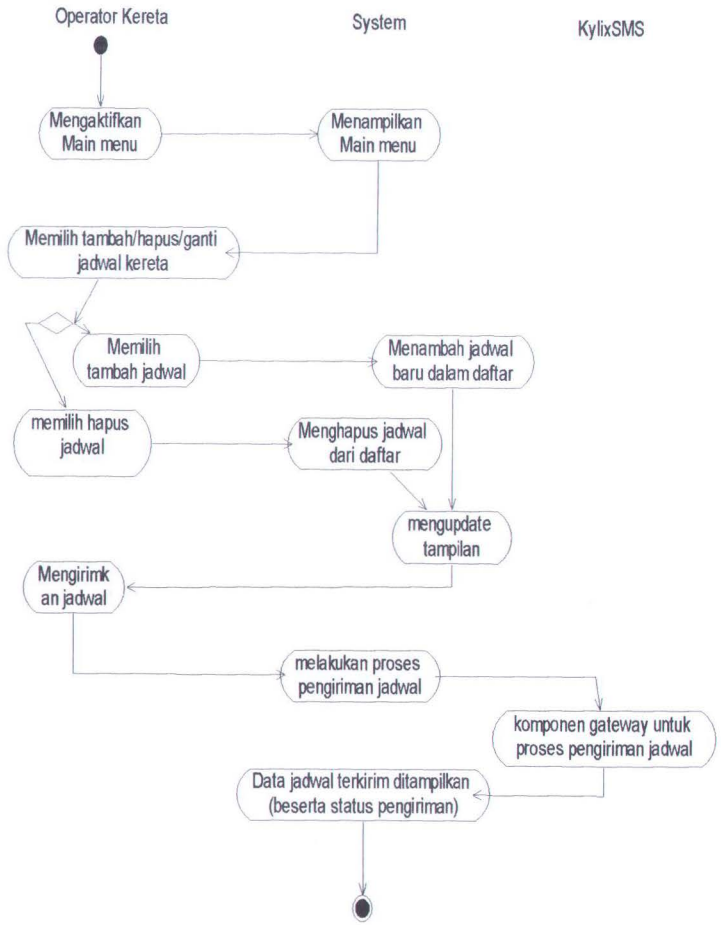
Manajemen Koneksi SMS Kereta



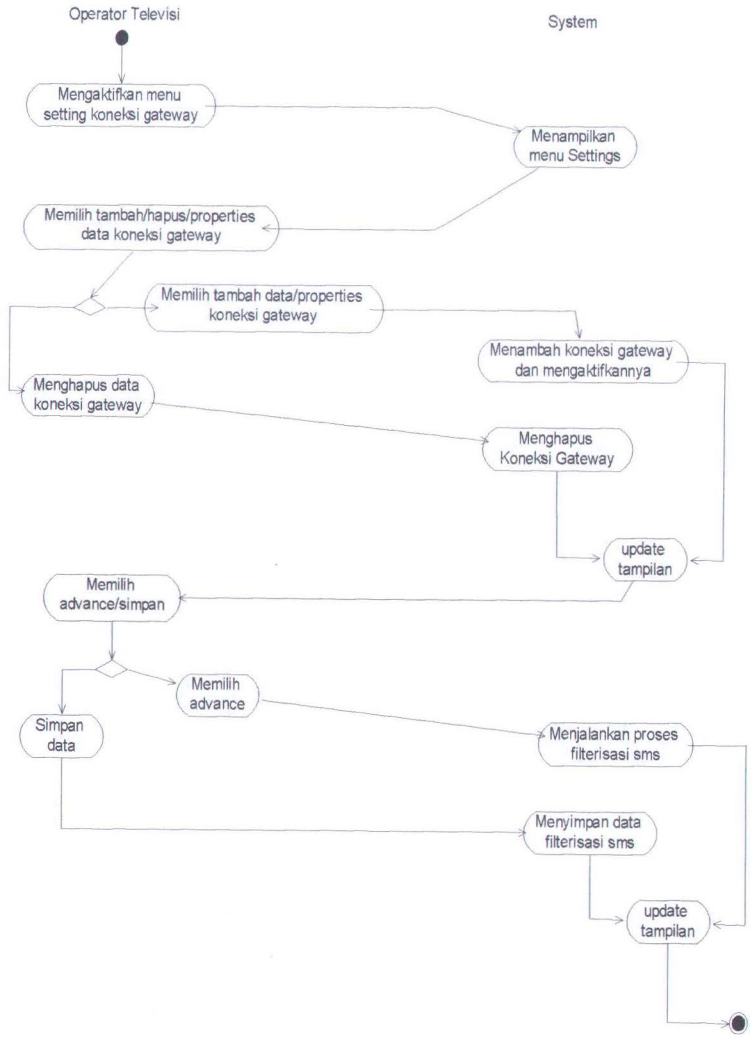
Manajemen Data Kereta



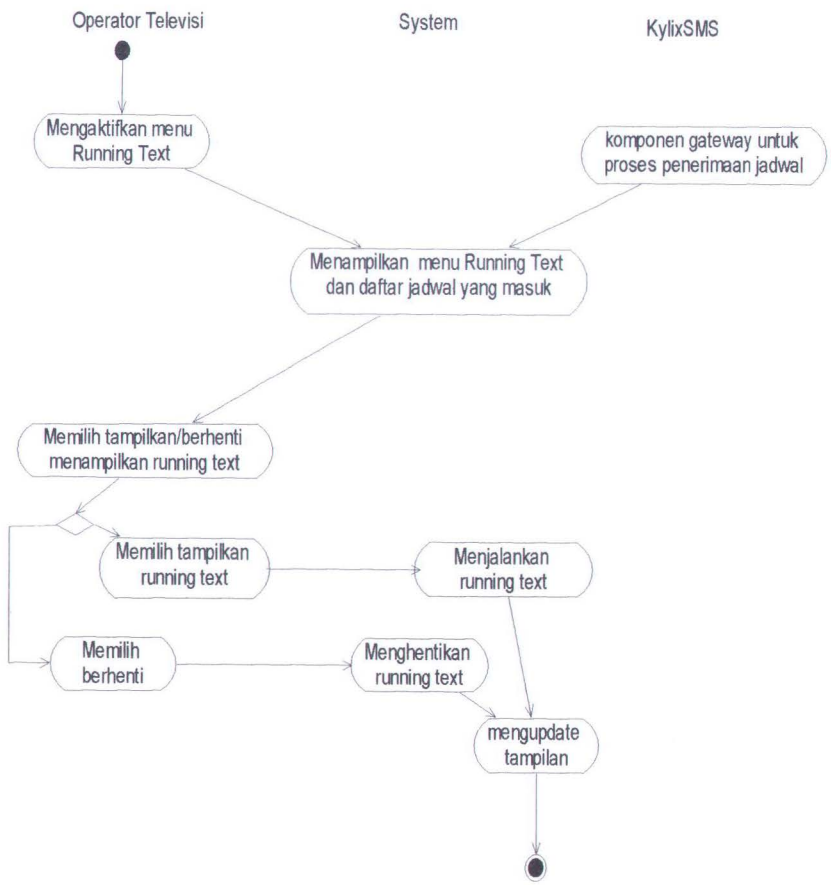
Manajemen Jadwal Kereta



Manajemen Koneksi SMS TV



Manajemen SMS TV



23.176/H/05

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK INTRUSION DETECTION SYSTEM MENGGUNAKAN METODE BAYESIAN NETWORK

TUGAS AKHIR

R514
005.1
Fai
p-1
2005

Disusun Oleh :

Nafis Faisal
NRP. 5199.100.014

PERPUSTAKAAN
I T S
Tgl. Terima 8-8-2005
Terima Dari H
No. Agenda Prp. 222164

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2005

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK INTRUSION DETECTION SYSTEM MENGGUNAKAN METODE BAYESIAN NETWORK

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan

Untuk Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui/Menyetujui

Dosen Pembimbing

Febriliyan Samopa, M.Kom

NIP 132.206.858

Surabaya, July 2005

ABSTRAK

Dengan semakin banyaknya pihak-pihak yang memanfaatkan jaringan internet sebagai jalur komunikasi data, maka banyak pihak yang merasakan bahwa keamanan jaringan adalah sesuatu keharusan. Banyak alat-alat yang dibangun untuk mencapai tujuan ini, salah satunya adalah dengan membangun suatu aplikasi pendeteksi intrusi (Intrusion Detection System/IDS)

Dalam penelitian ini akan dilakukan perancangan dan pembuatan aplikasi Sistem Pendeteksian Intrusi yang berfungsi untuk menganalisa packet-packet data yang datang dan pergi. Aplikasi ini menggunakan metode pendeteksian anomali menggunakan metode Naïve Bayes dalam mendeteksi suatu intrusi.

Walaupun terdapat kekurangan dalam ketepatan pendeteksian, tetapi jika dibandingkan dengan metode pendeteksian analisa signature, metode ini memiliki kelebihan-kelebihan tersendiri.

Dengan metode ini, suatu serangan yang belum pernah didefinisikan dapat dideteksi dengan mudah. Karena itu, metode ini sangat tepat untuk mendeteksi serangan-serangan yang merupakan zero-day attack.

Hasil dari uji coba yang dilakukan, menunjukkan bahwa aplikasi yang dibuat mampu melakukan pendeteksian intrusi dengan tingkat kesalahan false positive dan false negative yang kecil. Selain itu, karena kecilnya waktu komputasi metode Naïve Bayes, sehingga aplikasi mampu menangkap dan menganalisa paket-paket TCP dengan sangat cepat dan loss yang kecil.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT karena atas rahmat-Nya maka penulis dapat menyelesaikan Tugas Akhir yang merupakan syarat akademis bagi mahasiswa dalam rangka menyelesaikan program S1 di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Besar harapan penulis bahwa Tugas Akhir dengan judul "Perancangan dan Pembuatan Perangkat Lunak Intrusion Detection System Menggunakan Metode Bayesian Network" yang telah dikerjakan ini mampu memberikan kontribusi bagi perkembangan bidang Teknologi Informasi, khususnya studi mengenai *Intrusion Detection System* menggunakan metode *data mining*.

Penyusunan laporan Tugas Akhir ini tentunya tidak luput dari kesalahan dan kekurangan. Maka, segala saran dan kritik yang bersifat membangun maupun pengembangan lebih lanjut sangat diharapkan. Semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi kita semua.

Surabaya, Juni 2005

Nafis Faisal

UCAPAN TERIMA KASIH

Dalam kesempatan ini, penulis mengucapkan terima kasih bagi semua pihak yang telah membantu dalam pelaksanaan Tugas Akhir, khususnya kepada :

1. Allah SWT, yang telah memberikan kekuatan dan kesempatan untuk menyelesaikan tugas ini.
2. Ibu dan Bapak, yang selalu memberikan do'a dan kesabaran dalam membimbing penulis selama ini.
3. Bapak Febriliyan Samopa, S.Kom., M.Kom. selaku dosen pembimbing Tugas Akhir ini. Terima kasih untuk bimbingan, saran dan masukan selama pengerjaan Tugas Akhir.
4. Ibu Chastine Fatichah, S.Kom. selaku Koordinator Tugas Akhir. Terima kasih untuk semua waktu yang diluangkan dalam pengaturan keberhasilan presentasi Tugas Akhir.
5. Ibu Ir. Siti Rochimah, S.T. selaku dosen wali. Terima kasih untuk bimbingan yang telah diberikan.
6. Bapak Yudhi Purwananto, S.Kom., M.T., selaku Ketua Jurusan Informatika. Tugas Akhir ini tidak akan berhasil tanpa berjalannya proses-proses yang ada di lingkungan Teknik Informatika ITS.
7. Bapak dan Ibu Dosen Jurusan Teknik Informatika. Terima kasih untuk ilmu yang telah diberikan.
8. Seluruh staf tata usaha dan ruang baca yang telah sangat membantu penulis selama kuliah. Mas Yudi, Mbak Fatin, Mbak Eva serta staf-staf yang lain.

9. Mulyadi Santoso dan Wilmar Sulaiman, dua orang yang memberikan masukan-masukan yang sangat bermanfaat dalam penyelesaian tugas akhir ini.
10. Teman-teman ex-kantor di Maxxima Innovative Engineering, yang bersedia menyediakan koneksi internet gratis kepada penulis.
11. Kakakku Mbak Ulfa. Terima kasih atas motivasi yang diberikan kepada penulis.
12. Seluruh TA-wan Teknik Informatika ITS periode 2005 yang telah lulus dan akan lulus. Erzu, Kecu, Ray, Gede, Rahde, Ulfa, Inuk, Na'il, Imam, Wahib, Rono, Warna Agung, dan semuanya.
13. Teman-temanku tercinta di *c0f* yang tidak saya sebutkan namanya satu persatu. Terima kasih untuk semua dukungannya.
14. Seluruh mahasiswa Teknik Informatika ITS, senior dan yunior.

DAFTAR ISI

KATA PENGANTAR I

UCAPAN TERIMA KASIH II

DAFTAR ISI IV

DAFTAR GAMBAR VII

DAFTAR TABEL VIII

DAFTAR RUMUS IX

BAB 1 1

PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Tujuan dan Manfaat 2

1.3 Perumusan Masalah 3

1.4 Batasan Masalah 3

1.5 Metodologi Penelitian 3

1.6 Sistematika 4

BAB 2 6

DASAR TEORI 6

2.1 TCP/IP 6

2.1.1 Gambaran Umum 7

2.1.2 Arsitektur TCP/IP 9

2.1.3 Hubungan TCP /IP dengan Protokol yang lain. 13

2.2 IP *(Internet Protocol)* 15

2.2.1 Datagram IP 15

2.2.2 Routing pada IP 17

2.2.3 Alamat IP 19

2.3 TCP *(Transmission Control Protocol)* 20

2.3.1 Layanan dari TCP 20

2.3.2 Segment TCP 22

2.3.3 Koneksi I 24

2.4 Jenis-jenis Serangan 28

2.4.1 Denial of Service 30

2.4.2 User to Root 30

2.4.3 Remote to User 31

2.4.4 Probes 32

2.5 Klasifikasi Data 32

2.5.1 Bayesian Network 33

2.5.2 Naïve Bayes 35

2.5.3 Laplace Estimator 37

BAB 3 ..... **38**

PERANCANGAN SISTEM **38**

3.1 Gambaran Umum Sistem 38

3.2 Perancangan Data 41
3.2.1 Struktur Data Untuk Proses *Training* 42
3.2.2 Struktur Data Untuk Penyimpanan Koneksi pada Aplikasi 45

3.3 Flow Sistem Aplikasi Pendeteksian Intrusi 48

3.4 Perancangan Proses Aplikasi Pendeteksian Intrusi 49
3.4.1 DAD Level 0 49
3.4.2 DAD Level 1 51

3.5 Perancangan Antarmuka Aplikasi Pendeteksian Intrusi 58

BAB 4 ..... **60**

IMPLEMENTASI SISTEM **60**

4.1 Lingkungan Implementasi Aplikasi Pendeteksian Intrusi 60

4.2 Implementasi Data Aplikasi Pendeteksian Intrusi 60
4.2.1 Implementasi Data untuk Proses *Training* 60
4.2.2 Implementasi Struktur Data Penyimpanan Koneksi dalam Aplikasi 62

4.3 Implementasi Proses Aplikasi Pendeteksian Intrusi 63
4.3.1 Implementasi Proses DAD Level 0 63
4.3.2 Implementasi Proses DAD level 1 64

4.4 Implementasi Antarmuka Aplikasi Pendeteksian Intrusi 70
4.4.1 Implementasi Daftar Alert dan Fungsi Searching 70
4.4.2 Implementasi Daftar Alert *group by* IP Asal 71
4.4.3 Implementasi Daftar Alert *group by* IP Tujuan 71
4.4.4 Implementasi Daftar Alert *group by port* tujuan 72

BAB 5 ..... **73**

UJI COBA DAN EVALUASI **73**

5.1 Lingkungan Uji Coba Aplikasi Pendeteksian Intrusi 73
5.1.1 Perangkat Keras 73
5.1.2 Perangkat Lunak 73

5.2 Skenario Uji Coba 75
5.2.1 Uji Coba Pendeteksian Serangan Yang Telah Diketahui 75
5.2.2 Uji Coba Pendeteksian Serangan Baru 75
5.2.3 Uji Coba Kecepatan 76
5.2.4 Uji Coba Online 76

5.3 Pelaksanaan Dan Hasil Uji Coba 77
5.3.1 Pelaksanaan Dan Hasil Uji Coba Pendeteksian Serangan 77
5.3.2 Pelaksanaan Dan Hasil Uji Coba Pendeteksian Serangan Baru 80
5.3.3 Pelaksanaan Dan Hasil Uji Coba Kecepatan 82

5.4 Hasil Uji Coba Menggunakan Data Online 83

5.5 Evaluasi Akhir 85

BAB 6 ..... **86**

KESIMPULAN DAN SARAN **86**

6.1 Kesimpulan 86

6.2 Saran 87

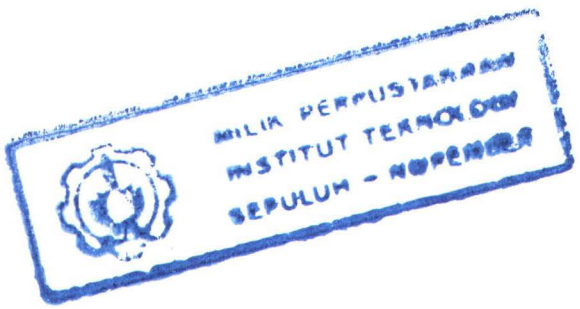
DAFTAR PUSTAKA **88**

DAFTAR GAMBAR

GAMBAR 2. 1 KONEKSI DUA JARINGAN 10
GAMBAR 2. 2 KONEKSI DUA JARINGAN 13
GAMBAR 2. 3 TCP/IP DAN PROTOKOL LAIN 14
GAMBAR 2. 4 DATAGRAM IP 16
GAMBAR 2. 5 ENKASULAPSI TCP *SEGMENT* 22
GAMBAR 2. 6 SEGMENT TCP 23
GAMBAR 2. 7 MEKANISME BUKA KONEKSI 25
GAMBAR 2. 8 MEKANISME TUTUP KONEKSI 27
GAMBAR 2. 9 DIAGRAM KONEKSI TCP 28
GAMBAR 2. 10 DIAGRAM JENIS SERANGAN 29
GAMBAR 2. 11 *BAYESIAN NETWORK* SEDERHANA 34
GAMBAR 2. 12 *NAÏVE BAYES* DENGAN BANYAK ATRIBUT 36
GAMBAR 3. 1 DESAIN KESELURUHAN APLIKASI PENDETEKSI INTRUSI 39
GAMBAR 3. 2 STRUKTUR DATA LIST KONEKSI DENGAN TARGET HOST TERTENTU 46
GAMBAR 3. 3 STRUKTUR DATA LIST KONEKSI DENGAN TARGET LAYANAN TERTENTU 47
GAMBAR 3. 4 FLOW APLIKASI PENDETEKSIAN SECARA SEDERHANA 48
GAMBAR 3. 5 DAD LEVEL 0 50
GAMBAR 3. 6 DAD LEVEL 1 PROSES PENDETEKSIAN 52
GAMBAR 3. 7 DAD LEVEL 2 PROSES SNIFFER 53
GAMBAR 3. 8 DAD LEVEL 2 PROSES PENYUSUNAN KONEKSI 54
GAMBAR 3. 9 DIAGRAM ALUR *CONNECTION CLEANING* 56
GAMBAR 3. 10 DIAGRAM SITEMAP ANTARMUKA *WEB CLIENT* 59
GAMBAR 4. 1 PERINTAH SQL UNTUK CREATE TABLE KONEKSI 61
GAMBAR 4. 2 FORMAT DATA TEXT DARI FILE TCPDUMP.LIST 62
GAMBAR 4. 3 IMPLEMENTASI STRUKTUR DATA PENYIMPAN KONEKSI PADA BAHASA C++ 63
GAMBAR 4. 4 PROGRAM UTAMA 64
GAMBAR 4. 5 FUNGSI UNTUK *CAPTURE* PAKET 65
GAMBAR 4. 6 FUNGSI UNTUK PARSING HEADER TCP 67
GAMBAR 4. 7 FUNGSI UNTUK REGISTER KONEKSI KEDALAM LIST 68
GAMBAR 4. 8 FUNGSI UNTUK MEMBERSIHKAN LIST 69
GAMBAR 4. 9 FUNGSI PENDETEKSIAN 69
GAMBAR 4. 10 IMPLEMENTASI HALAMAN DAFTAR ALERT BESERTA FUNGSI SEARCH 70
GAMBAR 4. 11 IMPLEMENTASI HALAMAN DAFTAR ALERT *GROUP BY* IP ASAL 71
GAMBAR 4. 12 IMPLEMENTASI HALAMAN DAFTAR ALERT *GROUP BY* IP ASAL 71
GAMBAR 4. 13 IMPLEMENTASI HALAMAN DAFTAR ALERT *GROUP BY PORT* TUJUAN 72
GAMBAR 5. 1 GAMBAR HASIL PENDETEKSIAN SECARA REALTIME DENGAN REPORT WEB BASED 84

DAFTAR TABEL

TABEL 3. 1 ATRIBUT-ATRIBUT KONEKSI 43
TABEL 5. 1 DAFTAR DATASET BESERTA KETERANGANNYA 77
TABEL 5. 2 PERCOBAAN KESATU MENGGUNAKAN DATASET 1 78
TABEL 5. 3 PERCOBAAN KEDUA MENGGUNAKAN DATASET 2 78
TABEL 5. 4 PERCOBAAN KETIGA MENGGUNAKAN DATASET 3 78
TABEL 5. 5 PERCOBAAN KELIMA MENGGUNAKAN DATASET 4 79
TABEL 5. 6 PERCOBAAN KELIMA MENGGUNAKAN DATASET 5 79
TABEL 5. 7 RANGKUMAN HASIL PERCOBAAN MISUSE 80
TABEL 5. 8 HASIL PERCOBAAN KESATU PENDETEKSIAN SERANGAN BARU 80
TABEL 5. 9 HASIL PERCOBAAN KEDUA PENDETEKSIAN SERANGAN BARU 80
TABEL 5. 10 HASIL PERCOBAAN KETIGA PENDETEKSIAN SERANGAN BARU 81
TABEL 5. 11 HASIL PERCOBAAN KETIGA PENDETEKSIAN SERANGAN BARU 81
.TABEL 5. 12 RANGKUMAN TINGKAT AKURASI DARI PENDETEKSIAN SERANGAN BARU 81
TABEL 5. 13 DAFTAR FILE DUMP DATA JARINGAN UNTUK PERBANDINGAN KECEPATAN 82
TABEL 5. 14 PERBANDINGAN KECEPATAN DALAM MEMBACA FILE DATA DUMP NETWORK 83
TABEL 5. 15 HASIL UJI COBA ONLINE MNGGUNAKAN NMAP 84



DAFTAR RUMUS

RUMUS 2. 1 TEOREMA BAYES 35
RUMUS 2. 2 JOINT DISTRIBUTION 36
RUMUS 2. 3 *LAPLACE ESTIMATOR* 37

BAB 1

PENDAHULUAN

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Internet telah menjadi jalur komunikasi utama masyarakat dunia. Informasi yang terdapat didalam internet bervariasi mulai dari hanya informasi sederhana sehari-hari hingga informasi-informasi penting yang melibatkan banyak perusahaan bahkan pemerintahan.

Internet didukung oleh adanya server-server yang terdapat pada berbagai belahan dunia. Dengan semakin bertambahnya server-server baru maka internet menjadi semakin kompleks yang pada akhirnya memunculkan persoalan-persoalan baru dalam beberapa hal.

Salah satu hal utama yang harus diberikan perhatian lebih adalah keamanan komunikasi data. Tidak semua implementasi internet tahan terhadap ancaman kemanan. Semakin kompleksnya sistem internet menjadikan semakin banyak ditemukan lobang-lobang keamanan baru sehingga pengguna internet harus mempersiapkan langkah-langkah untuk melakukan pencegahan atau paling tidak meminimalkan resiko kerugian dari ancaman-ancaman tersebut.

Biasanya, suatu server yang terkoneksi pada internet mempunyai seorang administrator yang tugasnya mengatur dan memonitor server tersebut. Seorang administrator biasanya menggunakan alat-alat (software pembantu) tertentu untuk memudahkan pelaksanaan tugasnya.

Salah satu alat yang penting dalam pengawasan server yaitu IDS (*Intrusion Detection System*/System Pendeteksian Intrusi). Biasanya alat seperti ini adalah suatu perangkat lunak *intelligent agent* yang dapat mengawasi adanya intrusi secara otomatis dalam batas-batas tertentu.

Pada dasarnya IDS terbagi dalam dua metode yaitu menggunakan pendeteksian dengan metode *misuse-signature based* dan menggunakan metode *anomaly based.* IDS dengan metode *misuse-signature based* memanfaatkan pengetahuan akan pola-pola yang menandakan suatu serangan sedangkan IDS dengan metode *anomaly based* lebih menekankan pada pendeteksian dengan melihat adanya suatu tingkah laku yang anomali dalam satu sesi transaksi data apabila dibandingkan dengan transaksi-transaksi yang lain.

Saat ini sudah terdapat beberapa perangkat lunak IDS yang memanfaatkan metode pendeteksian menggunakan *misuse-signature based* (misalnya : NFR, RealSecure, Dragon, Snort, dan Cisco Secure IDS) sedangkan untuk IDS dengan metode pendeteksian menggunakan *anomaly based* masih tidak ada yang bisa diimplementasikan dalam dunia nyata. Metode ini masih dalam tahap riset oleh para ilmuwan pada lingkungan akademisi maupun industri.

1.2 Tujuan dan Manfaat

Tujuan dari penelitian ini yaitu untuk membuat suatu aplikasi yang berfungsi melakukan pendeteksian serangan dengan menganalisa header dari paket-paket data yang diambil dari sebuah gateway.

Manfaat dari penelitian ini yaitu dapat membantu administrator jaringan untuk mengetahui serangan-serangan yang sedang atau akan terjadi.

1.3 Perumusan Masalah

- Membuat aplikasi untuk melakukan IP *packet Sniffing* yang dilanjutkan dengan IP *session assembling.*
- Membuat aplikasi yang cukup cepat untuk dapat melakukan analisa tiap-tiap koneksi IP yang ada dengan *loss packet* yang kecil.
- Membuat aplikasi untuk melakukan capture profile normal dan serangan dari data *training* yang ada.
- Mengintegrasikan knowledge yang didapat dari hasil capture profile normal dan serangan kedalam aplikasi pendeteksi.

1.4 Batasan Masalah

- Aplikasi yang akan dibuat pada sistem operasi FreeBSD
- Penelitian ini akan memanfaatkan *Packet Capture Library libPcap*
- Aplikasi ini dijalankan pada protokol TCP
- Aplikasi ini adalah *Network Intrusion Detection System*
- Aplikasi yang dibuat hanya menganalisa *header* dari tiap-tiap packet yang di-*capture* serta beberapa atribut dari tiap-tiap IP *session.*

1.5 Metodologi Penelitian

1. Studi Literatur
   - Studi dari literatur-literatur tentang jaringan dan *Packet Capture Library*
   - Studi literatur tentang *capture Session*
   - Studi terhadap aplikasi sejenis yang sudah ada

2. Perancangan Sistem / Aplikasi

   - Permodelan dan perancangan system, meliputi pembuatan alur proses serta komunikasi pada aplikasi.
   - Perancangan antarmuka yang sesuai sehingga dapat memudahkan pengguna dalam menggunakan aplikasi.

3. Pembuatan Aplikasi / Program

   Pada tahap ini, model dan rancangan sistem yang telah dibuat akan diimplementasikan dalam aplikasi.

4. Uji Coba, Evaluasi dan Modifikasi

   Pada tahap ini implementasi aplikasi diuji coba pada alat yang digunakan, serta dilakuakan evaluasi terhadap hasil uji coba tersebut dan jika diperlukan maka dilakukan modifikasi terhadap aplikasi

5. Penyusunan Laporan Penelitian

   Setelah tahap pengujian dan evaluasi maka hasil aplikasi yang sudah ada didokumentasikan dalam laporan Penelitian

1.6 Sistematika

Buku ini disusun dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Menjelaskan latar belakang, permasalahan yang diangkat beserta batasan-batasannya, tujuan dan manfaat yang ingin dicapai dari pembuatan penelitian ini serta metodologi yang dipakai.

BAB II Dasar Teori

Memaparkan hasil studi literatur berupa rangkuman materi tentang TCP/IP dan hal-hal yang berhubungan dengannya yang digunakan pada penelitian ini, juga materi tenyang *libpcap, library* yang dipakai untuk memungkinkan pengerjaan penelitian ini.

BAB III Perancangan Perangkat Lunak

Membahas rancangan aplikasi yang akan dibangun dimulai dari deskripsi umum sistem, batasan sistem, proses-proses yang terjadi, serta desain penyimpanan data fisik.

BAB IV Pembuatan Perangkat Lunak

Menunjukkan implementasi dari tiap rancangan yang telah dibuat yaitu lingkungan aplikasi sistem, implementasi proses, implementasi aplikasi.

BAB V Pengujian dan Evaluasi Perangkat Lunak

Membahas pengujian aplikasi, dari lingkungan pengujian, data yang diuji serta macam-macam skenario pengujian. Hasil dari tiap uji coba akan dievaluasi dan dianalisa dari segi hasil dan data yang diperoleh melalui penangkapan.

BAB VI Penutup

Menyatakan kesimpulan–kesimpulan yang didapat dari proses pembuatan Penelitian, beserta saran-saran untuk pengembangan selanjutnya.

BAB 2

DASAR TEORI

BAB 2
DASAR TEORI

Pada bagian ini akan dijelaskan tentang teori-teori yang menunjang dalam pembuatan dan perancangan penelitian ini. Teori penunjang tersebut meliputi : dasar-dasar teori jaringan komputer yang berhubungan dengan protokol TCP/IP, arsitekturnya serta hubungannya dengan protokol lain. Juga penjelasan tentang IP dan protokol-protokol yang dibentuk diatasnya dalam OSI layer.

Kemudian juga akan diulas beberapa jenis-jenis serangan yang telah banyak diketahui. Ulasan ini diikuti dengan pengklasifikasian serangan-serangan tersebut sesuai *behavior* masing-masing..

Pada bagian akhir, juga akan dijelaskan tentang metode *Naïve Bayes* yang merupakan salah satu dari metode *classification* untuk melakukan pengambilan *knowledge* dari *domain knowledge* tertentu. Pada kasus ini, metode ini digunakan untuk *capture* profile normal dan profile serangan dari data *training* yang sudah disiapkan yang pada akhirnya digunakan untuk menentukan apakah suatu IP *session* memiliki *likelyhood* normal ataukah serangan.

2.1 TCP/IP

Pada bagian ini akan dijelaskan mengenai protokol TCP/IP dan semua hal penting yang berkaitan dengan protokol tersebut.

2.1.1 Gambaran Umum

Protokol merupakan sekumpulan aturan yang mengatur dua atau lebih mesin dalam suatu jaringan dalam melakukan interaksi pertukaran format data. Protokol memiliki suatu fungsi yang spesifik satu sama lain pada sebuah hubungan telekomunikasi.

TCP/IP merupakan sekumpulan protokol yang dikembangkan untuk mengijinkan komputer-komputer agar dapat saling membagi sumber daya yang dimiliki masing-masing melalui media jaringan [12].

Protokol-protokol TCP/IP dikembangkan sebagai bagian dari riset yang dikembangkan oleh *Defense Advanced Research Projects Agency (DARPA)*. Pertama kalinya TCP/IP dikembangkan untuk komunikasi antar jaringan yang terdapat pada DARPA. Selanjutnya, TCP/IP dimasukkan pada distribusi perangkat lunak UNIX. Sekarang TCP/IP telah digunakan sebagai standar komunikasi *internetwork* dan telah menjadi protokol transport bagi internet, sehingga memungkinkan jutaan komputer berkomunikasi secara global.

TCP/IP memungkinkan komunikasi di antara sekumpulan interkoneksi jaringan dan dapat diterapkan pada jaringan *LAN* ataupun *WAN*. Tidak seperti namanya, TCP/IP tidaklah hanya memuat protokol di lapisan 3 dan 4 dari *OSI layer* (seperti IP dan TCP), tetapi juga memuat protokol-protokol aplikasi lainnya seperti *email, remote login, ftp, http*, dan sebagainya.

Protokol TCP/IP merupakan protokol yang memungkinkan komputer dengan berbagai ukuran, dari berbagai vendor dengan sistem operasi yang berbeda, untuk saling berkomunikasi. Realita yang terjadi sekarang, terutama yang

berhubungan dengan penggunaannya, bisa dibilang cukup mengagumkan. Karena sekarang ini penggunaan TCP/IP sudah sangat jauh dari tujuannya ketika pertama kali dibuat. Dari suatu proyek penelitian yang dibiayai pemerintah AS pada akhir tahun 1960-an untuk membuat suatu mekanisme jaringan pertukaran paket, telah berkembang menjadi suatu mekanisme komunikasi jaringan antar komputer yang paling banyak digunakan didunia saat ini. Benar-benar merupakan *open system* yang protokol dan implementasinya tersedia untuk publik dengan sedikit atau tanpa biaya. Merupakan dasar apa yang kita kenal dengan *Internet* yang menghubungkan komputer-komputer diseluruh dunia dalam suatu mekanisme pertukaran informasi [12].

TCP/IP dapat diterima oleh masyarakat dunia karena memiliki karakteristik sebagai berikut:

- Protokol TCP/IP dikembangkan menggunakan standar protokol yang terbuka.
- Standar protokol TCP/IP dalam bentuk *Request For Comment* (RFC) dapat diambil oleh siapapun tanpa biaya.
- TCP/IP dikembangkan dengan tidak tergantung pada sistem operasi atau perangkat keras tertentu.
- Pengembangan TCP/IP dilakukan dengan konsensus dan tidak tergantung pada vendor tertentu.
- TCP/IP independen terhadap perangkat keras jaringan dan dapat dijalankan pada jaringan *Ethernet*, *Token Ring*, jalur telepon dial-up, jaringan X.25, dan praktis jenis media transmisi apapun.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

- TCP/IP memiliki fasilitas *routing* yang memungkinkan sehingga dapat diterapkan pada *internetwork.*

2.1.2 Arsitektur TCP/IP

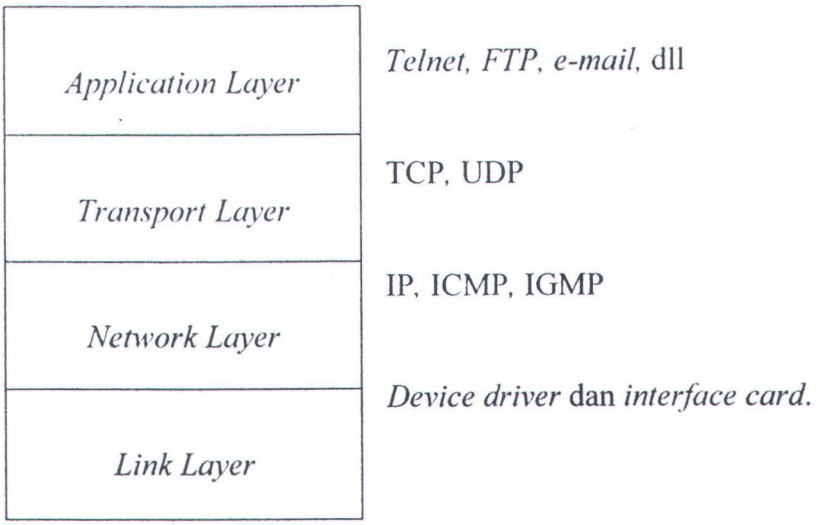
Seperti telah disebutkan sebelumnya, TCP/IP berisi kumpulan dari protokol-protokol yang melakukan fungsinya masing-masing secara spesifik. Protokol-protokol ini dikumpulkan berdasarkan fungsinya dalam lapisan-lapisan tertentu. TCP/IP memiliki 4 lapisan yang antara satu dengan lainnya memiliki protokol dengan fungsi yang saling melengkapi satu sama lain. TCP/IP secara umum terdiri dari 4 lapisan utama, dapat dilihat pada gambar 2.1 pada halaman berikutnya.

Lapisan-lapisan tersebut adalah:

a. Link Layer

Juga disebut *Network Access Layer.* Lapis ini merupakan lapis terbawah pada lapis TCP/IP. Fungsi protokol-protokol pada lapis ini adalah:

- Mendefinisikan bagaimana menggunakan jaringan untuk mengirimkan *frame*, yang merupakan unit data yang dilewatkan melalui media fisik.
- Protokol pada lapis ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.
- Pada lapis ini terdapat protokol-protokol seperti *Ethernet, Token Ring*, PPP, FDDI, ATM, X.25, dan SLIP.



Gambar 2. 1 Koneksi dua jaringan

b. Network Layer

Internet Layer Lapis ini bertanggung jawab atas routing yang ada pada jaringan. Protokol-protokol pada lapis ini menyediakan sebuah *datagram network service*. *Datagram* merupakan paket-paket informasi yang terdiri atas *header*, data, dan *trailer*. *Header* berisi informasi, seperti alamat tujuan yang dibutuhkan oleh jaringan untuk merutekan *datagram*. Sebuah *header* juga dapat berisi informasi lainnya seperti alamat asal dari pengirim. *Trailer* biasanya berupa nilai *checksum* yang digunakan untuk memastikan bahwa data tidak dimodifikasi pada saat transit.

Pada lapis ini terdapat protokol IP (*Internet Protocol*) yang berfungsi untuk menyampaikan paket data ke alamat yang tepat. ICMP, yang menyediakan kemampuan kontrol dan pesan. ARP, yang menentukan

MAC address dari dari alamat IP yang diketahui, serta RARP yang menentukan alamat IP jika diketahui alamat MAC.

c. Transport Layer

Lapis transport memiliki dua fungsi – *flow control*, yang disediakan oleh *sliding windows*; dan *reliability*, yang disediakan oleh *sequence number* dan *acknowledgement*. Dalam protokol TCP/IP terdapat 2 protokol transport yang utama yaitu : TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).

TCP menyediakan aliran data yang handal antara 2 *hosts*. Lapisan ini menitik beratkan pada hal-hal seperti : membagi-bagi data yang melewatinya dan mengaturnya sehingga dapat diterima dengan baik oleh lapisan dibawahnya, melakukan pemberitahuan tentang paket yang sudah diterima, mengatur waktu *timeout* untuk memastikan adanya pemberitahuan bahwa paket yang kita kirim sudah sampai dari pihak yang kita kirimi, dll. Karena kehandalan dalam aliran data yang dimiliki lapisan transport ini, maka lapisan aplikasi tidak perlu lagi memperhatikan detail-detail yang sudah diatur oleh lapisan transport.

Sedangkan UDP menyediakan layanan yang lebih sederhana. UDP cukup mengirimkan sutau paket data yang disebut *datagram* dari satu *hosts* ke yang lain, tapi tanpa suatu jaminan bahwa paket yang dikirim tersebut sampai ke tempat yang dituju. Maka lapis Aplikasi harus memberi tambahan supaya aliran data yang berjalan bisa lebih handal.

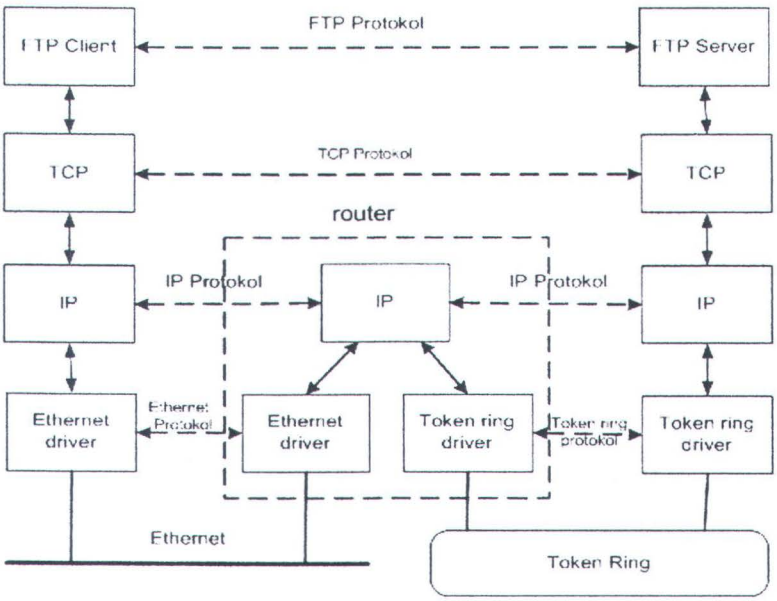
d. Application Layer

Lapis ini merupakan lapis teratas pada TCP/IP. Lapis ini menyediakan fungsi-fungsi bagi aplikasi-aplikasi pengguna. Lapis ini menyediakan layanan-layanan yang dibutuhkan oleh aplikasi-aplikasi user untuk berkomunikasi pada jaringan. Pada lapis ini terdapat beberapa protokol seperti TFTP, FTP, NFS untuk file transfer. SMTP dan POP3 sebagai protokol aplikasi *email*. Telnet dan FTP sebagai aplikasi *remote login*. SNMP sebagai protokol manajemen jaringan. Kemudian DNS, sebagai protokol aplikasi sistem penamaan di internet. Serta HTTP, sebagai protokol aplikasi web.

Dalam komunikasi, setiap lapisan yang terdapat pada tiap *host* berkomunikasi dengan lapisan yang sama pada *host* yang dituju. Komunikasi berlangsung melalui protokol-protokol yang dimiliki oleh masing-masing lapisan. Sama halnya yang terjadi pada internet. Setiap jaringan memiliki lapisan dan protokolnya yang berkomunikasi dengan lapisan dan protokol yang sama pada jaringan yang lain. Bedanya terletak pada adanya perangkat penghubung antar jaringan. Misal 2 buah jaringan berkomunikasi untuk melakukan FTP diinternet melalui sebuah *router*. Secara umum dapat dilihat pada gambar 2.2 di halaman sealnjutnya.

2.1.3 Hubungan TCP /IP dengan Protokol yang lain.

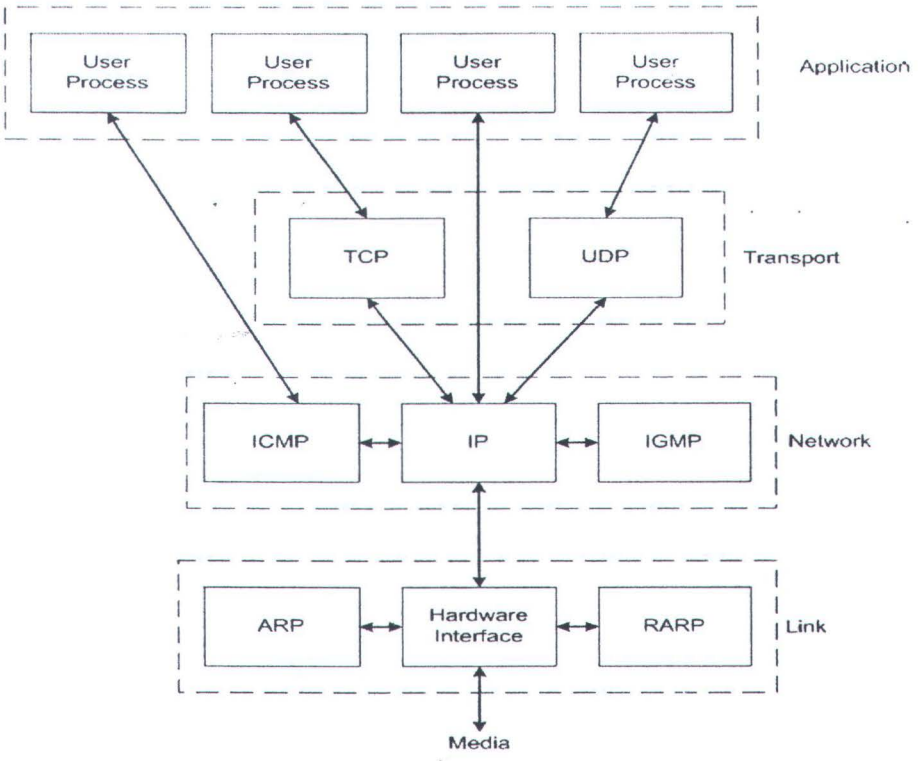
Selain protokol yang telah disebutkan sebelumnya, terdapat protokol-protokol lain dalam mekanisme komunikasi yang berhubungan dengan TCP/IP. Secara hierarki dapat dilihat pada gambar 2.2 di berikut ini:



Gambar 2. 2 Koneksi dua jaringan

TCP dan UDP merupakan protokol yang dominan pada lapisan transport, dan sama-sama menggunakan IP pada lapisan *network*. TCP menyediakan komunikasi yang handal walaupun menggunakan IP yang kurang handal. Sedangkan UDP dan kombinasinya dengan IP tidak menjamin sampainya *datagram* yang dikirimnya ke tempat tujuan.

IP merupakan protokol utama pada lapisan *network*. Setiap data TCP dan UDP ditranfer ke internet melalui IP. Pada gambar 2.3 di halaman selanjutnya terdapat juga aplikasi yang langsung mengakses IP. Ini dapat terjadi namun sangat jarang.



Gambar 2. 3 TCP/IP dan protokol lain

ICMP bisa dikatakan sebagai pembantu bagi IP. ICMP digunakan oleh IP untuk bertukar pesan *error* dan informasi penting lainya dengan IP pada jaringan/*host* tujuan. Walaupun demikian ICMP juga dapat diakses langsung oleh aplikasi pengguna. *Ping* dan *traceroute* adalah contohnya. Sedangkan IGMP digunakan untuk keperluan *multicasting* dalam hal ini mengirim *datagram* UDP kebeberapa *host* secara bersama-sama.

ARP (*Address Resolution Protocol*) dan RARP (*Reverse Address Resolution Protocol*) merupakan protokol yang digunakan untuk antarmuka jaringan tertentu saja (misal : *Ethernet* atau *token ring*), berfungsi untuk mengkonversi pengalamatan yang digunakan oleh IP dan pengalamatan yang digunakan oleh antarmuka jaringan.

2.2 IP *(Internet Protocol)*

IP merupakan protokol yang menjadi pekerja dalam lingkungan protokol-protokol TCP/IP. Data dari TCP, UDP, ICMP dan IGMP semua dikirim sebagai datagram IP. Yang mengagumkan adalah kenyataan bahwa IP menyediakan layanan pengiriman datagram yang *unreliable* dan *connectionless.*

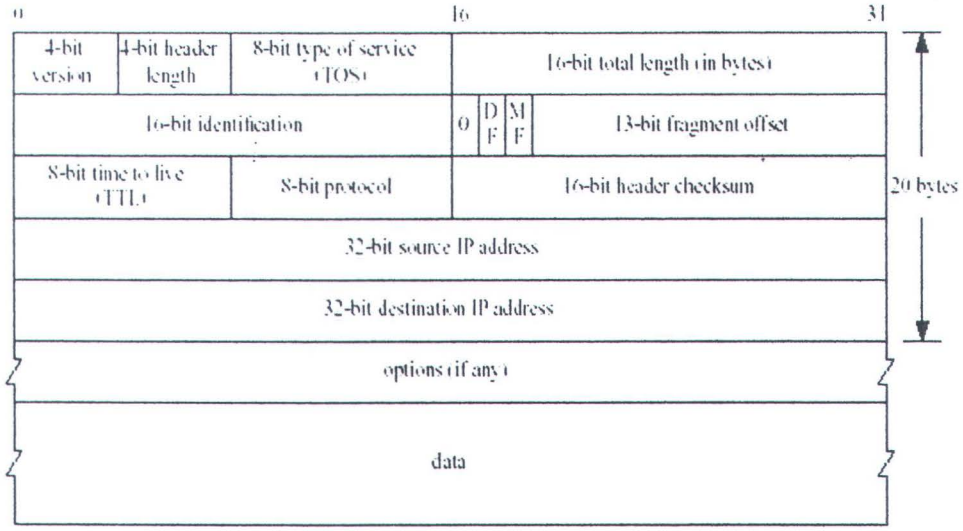
Unreliable berarti tidak ada jaminan akan sampainya IP datagram ke tempat tujuan. Namun IP menyediakan layanan untuk mengatasi jika terjadi hal-hal yang tidak diinginkan dalam koneksi, melalui algoritma *error handling* sederhana.

Connectionless berarti IP tidak menyediakan suatu informasi tentang status pengiriman datagramnya. Setiap *datagram* dikirim secara terpisah dengan yang lain, sehingga memungkinkan *datagram* tidak sampai atau *datagram* yang dikirim lebih lambat, sampai ditujuan lebih cepat. Selain itu *datagram* dapat juga sampai ketujuan secara acak [12].

2.2.1 Datagram IP

Sebuah datagram IP berisi *header* IP dan data serta dikelilingi oleh Media Access Control (MAC) *header* dan MAC *trailer*. IP *header* secara normal berukuran 20 *bytes* kecuali kolom *Options* memiliki nilai. Satu pesan dapat dikirim dalam urutan *datagram-datagram* yang disusun kembali menjadi pesan asli pada sisi penerima [10]. Datagram IP dapat dilihat pada gambar 2.4 pada halaman 17.

IP Header



Gambar 2. 4 Datagram IP

Field-field yang terdapat dalam IP *datagram* adalah sebagai berikut:

- VERSION – merupakan versi dari protokol IP yang digunakan untuk membuat *datagram*.
- IHL(*Internet Header Lenght*) – panjang *header*, dalam 32-bit words. Nilai minimum = 5 (20 bytes), nilai maksimum = 15 (60 bytes).
- TOS (*type of service*) –yang mengatur bagaimana datagram diperlakukan.
- *Total length* – panjang total (*header* + data) , panjng maksimum 65.535
- *Identification, flags, frag offset* – menyediakan fragmentasi dari datagram
- TTL -- *Time-To-Live*
- Protocol – protokol di lapis atas (layer 4) yang mengirimkan datagram

- *Header Checksum* – merupakan field yang berguna untuk pengecekan integritas *header*
- *Source* IP *address* dan *Destination* IP *address* -- 32-bit alamat IP
- *Options* – digunakan untuk pengetesan jaringan, *debugging*, keamanan, dan opsi lainnya. Beberapa opsi yang terdapat pada *options* : 0 Akhir dari daftar *options*, 1 Tidak ada operasi, 7 *Record route*, 68 *Timestamp*, 131 *Loose Source route*, 137 *trick source route*.

2.2.2 Routing pada IP

IP *Routing* terutama untuk sebuah *host* cukup sederhana. Jika tujuan langsung terhubung dengan *host* (*point to point*), atau dalam suatu jaringan, maka *datagram* IP langsung dikirim ke tujuan. Jika tujuan tidak langsung terhubung maka *host* akan mengirim *datagram* ke *router* terdekat dan *router* yang akan mengirimkannya ke tujuan [12].

Secara umum, IP dapat menerima *datagram* dari TCP, UDP, ICMP dan IGMP. IP memiliki *routing table* di memori yang akan melakukan pencarian apabila ada *datagram* yag diterima. Bila terdapat *datagram* yang diterima dari antarmuka jaringan, IP pertama kali memeriksa alamat yang dituju oleh *datagram* tersebut, apakah ke IP tersebut atau ke alamat *broadcast* dari IP. Apabila ya, maka *datagram* dikirim ke protokol yang dispesifikasikan oleh *header* IP. Bila tidak : pertama, bila IP yang menerima *datagram* tersebut juga berfungsi sebagi *router* maka *datagram* dilewatkan. Kedua, *datagram* ditolak.

Tiap entry pada *routing table* berisi informasi sebagai berikut :

- Alamat IP tujuan. Bisa berupa alamat jaringan atau alamat *host*, ditentukan oleh isi dari *flags*.
- Alamat IP dari *router* selanjutnya, atau alamat IP dari jaringan yang langsung terhubung. *Router* selanjutnya merupakan *router* yang langsung terhubung dengan kita sehingga kita bisa mngeirimkan *datagram* melaluinya.
- *Flags*. Menunjukkan apakah alamat IP yang dituju merupakan alamat jaringan atau alamat *host*.
- Spesifikasi dari antarmuka jaringan yang akan dilewati *datagram* yang akan dikirim.

IP Routing diselesaikan berdasarkan mekanisme *hop-by-hop*, hal ini karena IP tidak mengetahui rute yang harus dilalui untuk sampai ketujuan yang diinginkan kecuali tujuannya tersebut terhubung langsung. Yang diketahui IP hanyalah alamat dari *router* selanjutnya yang akan dilewati oleh *datagram*. Dengan asumsi bahwa *router* selanjutnya langsung terhubung dengan tujuan [1]. Maka IP *Routing* melakukan tindakan sebagai berikut :

1. Mencari di *routing table* entry yang cocok dengan alamat lengkap dari IP

   tujuan. Jika ditemukan maka paket dikirimkan ke *router* selanjutnya atau ke perangkat yang terhubung langsung dengan tujuan.

2. Mencari di *routing table* entry yang sesuai dengan ID jaringan tujuan. Jika ditemukan maka paket dikirimkan ke *router* selanjutnya atau ke perangkat yang terhubung langsung dengan tujuan.
3. Mencari di *routing table* entry dengan label "*default*". Jika ditemukan maka paket dikirimkan ke *router* selanjutnya atau ke perangkat yang terhubung langsung dengan tujuan.

Jika tidak ada yang ditemukan maka *datagram* tidak akan dikirim, dengan pesan error "*host unreachable*" atau "*network unreachable*" dikirim ke aplikasi.

2.2.3 Alamat IP

Alamat IP adalah 32-bit nilai yang unik yang mengidentifikasikan setiap peralatan yang terhubung ke jaringan TCP/IP. Alamat IP umumnya ditulis sebagai empat bilangan desimal yang dipisahkan oleh titik. Setiap desimal mewakili 8-bit byte dari 32-bit alamat, dan setiap desimal berada pada batas 0-255.

Pada alamat IP berisi bagian alamat jaringan dan bagian alamat *host*, tetapi format dari bagian itu tidak sama untuk setiap alamat IP. Jumlah dari bit alamat yang digunakan untuk mengidentifikasikan jaringan dan untuk mengidentifikasikan *host* bergantung pada *prefix length* dari alamat. *Prefix length* ditentukan oleh bit *mask* alamat.

Bit *mask* alamat bekerja seperti: panjang bit yang menentukan alamat jaringan pada alamat IP sama dengan panjang bit yang bernilai satu pada *mask*, sedangkan panjang bit host sama dengan panjang bit yang bernilai nol. Untuk contoh, jika alamat 172.22.12.4 mempunyai *mask* 255.255.255.0, dimana panjang bit yang bernilai satu adalah 24 dan yang bernilai 0 adalah 8. Jadi 24 bit pertama

digunakan untuk alamat jaringan yaitu 172.22.12 dan 8 bit sisanya digunakan untuk alamat host yaitu 4.

Untuk menuliskan alamat IP tidak harus dengan menyebutkan alamat dan *mask*-nya, kita bisa juga menuliskan alamat IP dengan format *alamat/prefix-length* dimana *prefiks-length* adalah panjang bit yang menyatakan alamat jaringan. Sebagai contoh alamat 172.22.12.8 *mask* 255.255.255.0 dapat diganti menjadi 172.22.12.8/24.

2.3 TCP *(Transmission Control Protocol)*

2.3.1 Layanan dari TCP

Walaupun TCP dan UDP sama-sama menggunakan menggunakan IP, tapi TCP menyediakan layanan yang berbeda dengan UDP. TCP menyediakan layanan yang *connection oriented, reliable* dan *byte stream. Connection oriented* memiliki arti bahwa dua aplikasi yang menggunakan TCP harus melakukan koneksi TCP sebelum melakukan pertukaran data [12].

Reliability dalam TCP dilakukan dengan menyediakan layanan-layanan sebagai berikut :

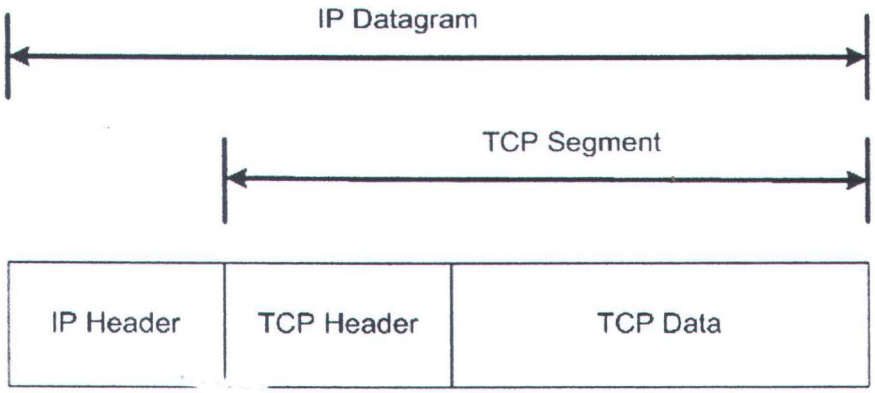
- Data aplikasi dibagi sesuai dengan ukuran yang menurut TCP paling baik untuk melakuakan pengiriman data. Berbeda dengan UDP yang mengirim *datagram* sesuai dengan ukuran aplikasi.
- Ketika melakukan pengiriman *segment*, TCP menjalankan *timer* dan menunggu respon *acknowledge* dari *segment* yang dikirim. Jika pesan *acknowledge* tidak diterima maka *segment* akan dikirm lagi.

- Jika TCP menerima data, maka TCP akan mengirim *acknowledgement* ke pengirim.
- TCP memiliki *checksum* pada *header*nya. *Checksum* ini berfungsi untuk melakukan pengecekan ada tidaknya modifikasi data ketika transit. Apabila ketika data yang diterima memiliki *checksum* yang invalid, TCP akan menolaknya dan tidak mengirimkan pesan *acknowledge*.
- Karena *segment* TCP dikirim dalam suatu IP datagram, yang mana IP datagram tidak menjamin data sampai ditempat tujuan atau data yang sampai tidak berurutan, TCP penerima perlu melakukan pengurutan data.
- Karena IP *datagram* dapat mengalami duplikasi, maka TCP akan menolak data yang terduplikasi.
- TCP juga mempunyai *flow control*. Setiap TCP yang berhubungan memiliki *buffer space* yang tertentu. TCP penerima hanya memperbolehkan pengirim mengirim data sebanyak penerima memiliki *buffer space*. Ini untuk menghindari *hosts* yang cepat mengambil semua *buffer space* host yang lebih lambat.

Selain itu TCP tidak melakukan interpretasi terhadap data yang diterima, apakah data *biner, ASCII* atau yang lain. Interpretasi terhadap *stream* sepenuhnya diserahkan pada aplikasi pada masing-masing *host*.

2.3.2 Segment TCP

Seperti telah disampaikan sebelumnya, data TCP dikirim sebagai datagram IP. Hal ini mengharuskan *segment* TCP harus dienkapsulasi didalam datagram IP. Enkasulapsi ini penting terutama untuk validitas data yang dikirimkan. Enkapsulasi data TCP ini dapat dilihat pada gambar 2.5 sebagai berikut :



Gambar 2. 5 Enkasulapsi TCP *Segment*

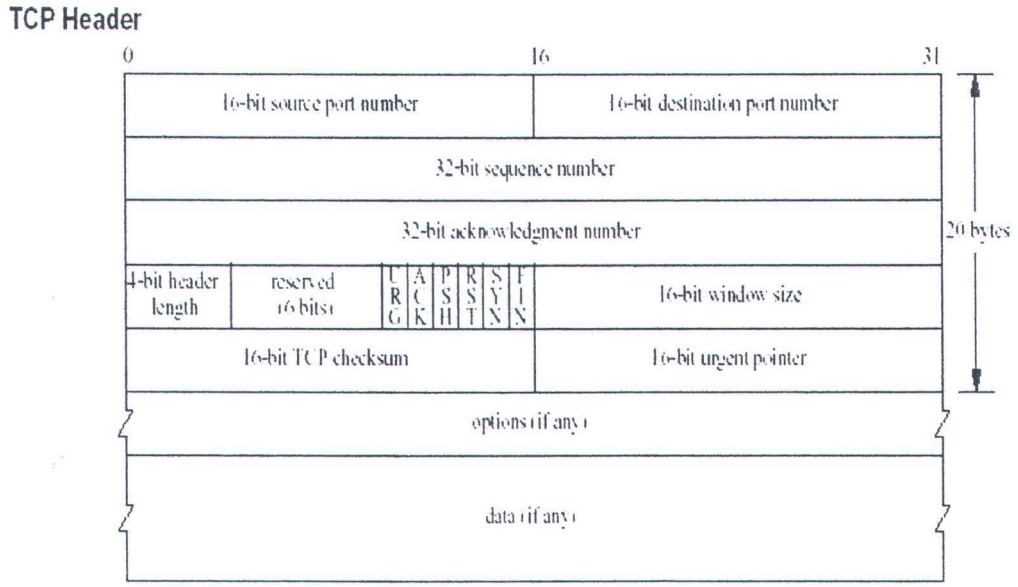
Data aplikasi yang dikirim oleh TCP disebut *segment*, seperti terlihat pada gambar diatas. *Segment* TCP dikirim sebagai *internet datagram*. IP *Header* membawa beberapa kolom-kolom informasi,termasuk alamat asal dan tujuan. TCP *header* secara normal berukuran 20 *bytes* diluar *options*. TCP *Header* berfungsi untuk menyediakan informasi-informasi yang spesifik untuk protokol TCP [9].

Segment TCP berisi *field-field* berikut:

- *Source Port* – angka yang menunjukkan port yang memanggil
- *Destination Port* – angka yang menunjukkan port yang dipanggil
- *Sequence Number* – angka yang digunakan untuk memastikan urutan yang benar dari data yang datang

- *Acknowledgment Number* – oktet TCP selanjutnya yang diharapkan
- HLEN – angka 32 bit words pada *header*
- *Reserved* -- diset 0
- *Code bits* – fungsi kontrol (misalnya setup dan terminasi dari suatu sesi)
- *Window* – angka dari oktet yang diterima oleh pengirim
- *Checksum* – *checksum* yang telah dikalkulasi dari kolom *header* dan data
- *Urgent pointer* – mengindikasikan akhir dari data yang penting
- *Options* -- ukuran maksimum dari *segment* TCP
- Data – data dari protokol lapis atas

Secara lengkap *segment* TCP dapat dilihat pada gambar 2.6 sebagai berikut :



Gambar 2. 6 Segment TCP

2.3.3 Koneksi I

Pada bagian ini, kita akan membahas lebih jauh tentang koneksi data serta *states* yang dimiliki TCP/IP dan penanganannya pada protocol TCP/IP.

2.3.3.1 Membuka koneksi

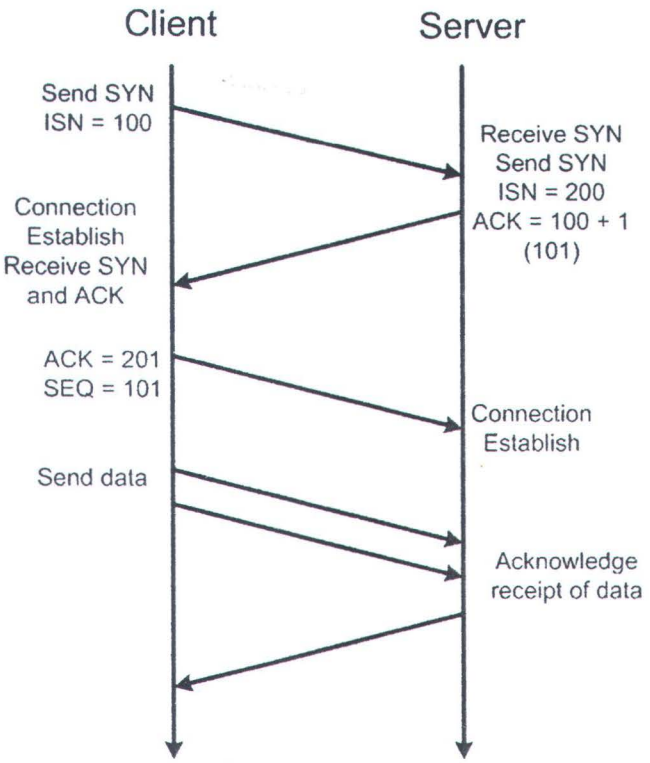
Koneksi TCP selalu dimulai dengan *3 way handshake,* yang menentukan dan menegosiasikan koneksi data yang akan dikirim. Suatu sesi koneksi dimulai dengan paket SYN lalu paket SYN/ACK dan paket ACK yang menyatakan selesainya suatu sesi koneksi. Ini berarti koneksi telah terjalan dan siap untuk mengirimkan data. Disebut *3 way handshake* karena untuk melakukan pembukaan koneksi melalui pengiriman 3 *segment* [12]. Adapun mekanisme pengirimannya adalah sebagai berikut :

- *Client* (pihak yang meminta koneksi) mengirimkan *segment* SYN untuk melakukan spesifikasi nomor *port* dari *server* yang akan dikoneksi. Segment ini juga berisi ISN (*Initial Sequence Number*) dari *client.* Ini adalah *segment* pertama. Contoh ISN *client* adalah 100.
- *Server* meresponnya dengan mengirimkan *segment* SYN miliknya yang berisi ISN server, misal 200. Ini adalah *segment* kedua. *Server* juga menjawab SYN dari *client* dengan mengirimkan ACK berisi ISN *client* ditambah 1.
- *Client* menjawab SYN dari *server* dengan mengirimkan ACK berisi ISN *server* ditambah 1 (*segment* ketiga).

Client sebagai pihak yang pertama mengirim SYN disebut melakukan *active open.* Sedangkan *server* yang mengirimkan SYN yang kedua disebut

melakukan *passive open*. Setiap akan membuka koneksi *client* dan *server* memilih ISN-nya masing-masing, sehingga setiap koneksi memiliki ISN yang berbeda. Ini untuk menghindari paket yang terhambat dijaringan, dikirimkan kembali dan menyebabkan interpretasi yang salah karena dianggap masih sebagai bagian dari koneksi yang berlangsung.

Setelah mekanisme *3 way handshake* koneksi sudah terjadi dan dapat dilanjutkan dengan pengiriman data. Setelah semua data dikirimkan maka penerima mengirimkan ACK bahwa data sudah diterima. Seluruh mekanisme pembukaan koneksi ini dapat dilihat pada gambar 2.7 di bawah ini :



Gambar 2. 7 Mekanisme Buka Koneksi

2.3.3.2 Menutup koneksi

Untuk melakukan penutupan koneksi diperlukan 4 *segment.* Ini terjadi karena adanya mekanisme *half* close yang dikarenakan oleh koneksi TCP yang *full duplex.* Mekanisme *full duplex* ini memungkinkan setiap ujung koneksi (*client* dan *server*) dapat saling mengirimkan data ke masing-masing secara bersamaan. Sehingga untuk menutup koneksi harus dilakukan penutupan *client* dan *server. Half close* berarti hanya satu ujung koneksi (*client* atau *server*) yang melakukan penutupkan koneksi. Aturan yang berlaku, tiap ujung konkesi (*client* dan *server*) dapat mengrimkan FIN. Pengiriman FIN menunjukkan bahwa si pengirim sudah selesai melakukan pengiriman data. Namun TCP tetap dapat melakukan pengiriman data selama penerima FIN yang pertama tadi belum mengirimkan FIN sebagai pertanda sudah tidak mengirimkan data lagi [12]. Secara lengkap mekanismenya dapat dijelaskan sebagai berikut :

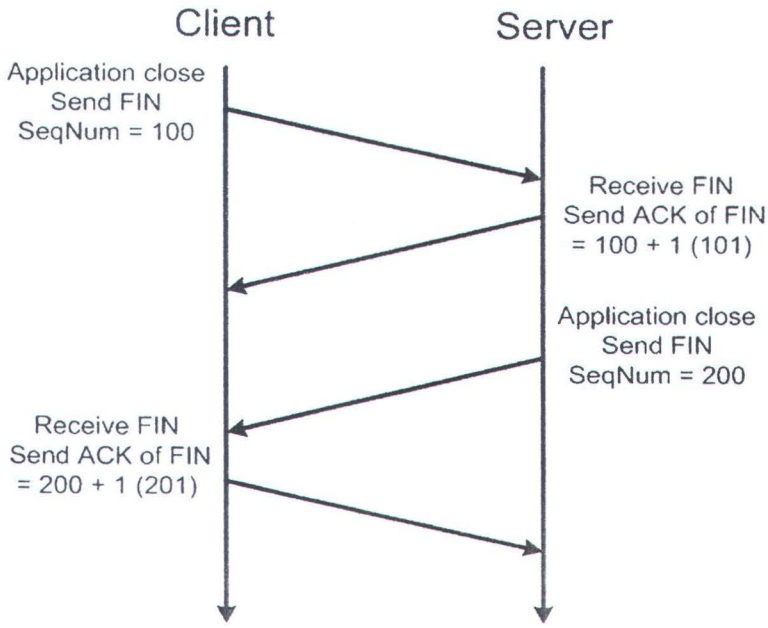
- Bila *client* akan menutup koneksi, maka *client* mengirimkan *segment* FIN ke *server*, menunjukkan bahwa *client* tidak akan mengirimkan data lagi.
- *Server* menerima FIN dari *client* dan menjawabnya dengan mengirimkan *segment* ACK yang berisi *sequence number* dari *client* ditambah 1.
- Dan apabila *server* juga telah mengirimkan data maka *server* mengirimkan *segment* FIN untuk menutup koneksi.

- *Client* menerima FIN dari *server* dan menjawabnya dengan mengirimkan *segment* ACK yang berisi *sequence number* dari *server* ditambah 1.

Secara umum mekanisme penutupan koneksi dapat dilihat pada gambar 2.8 di dibawah.

Sehingga dapat dilihat bahwa koneksi tidak pernah berhenti sampai ACK terakhir diterima/dikirim. Koneksi juga dapat dihentikan dengan mengirimkan RST (*reset*). Jika hal ini terjadi, maka koneksi akan terhenti setelah beberapa waktu tertentu.

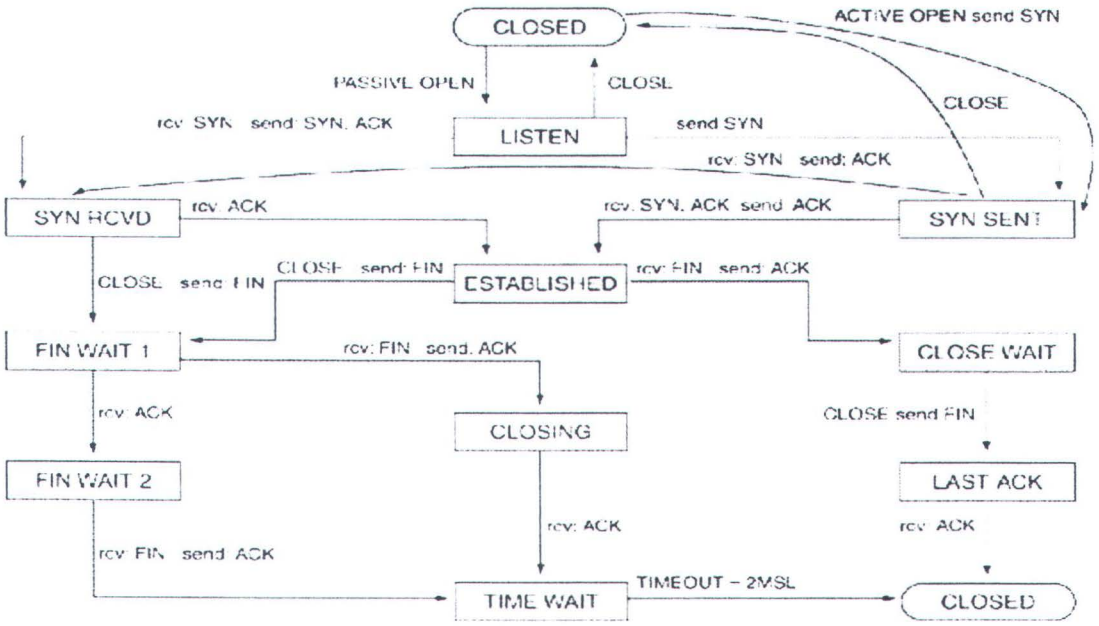
Sehingga dapat dilihat bahwa koneksi tidak pernah berhenti sampai ACK terakhir diterima/dikirim. Koneksi juga dapat dihentikan dengan mengirimkan RST (*reset*). Jika hal ini terjadi, maka koneksi akan terhenti setelah beberapa waktu tertentu.



Gambar 2. 8 Mekanisme Tutup Koneksi

Ketika koneksi TCP terhenti, koneksi memasuki *state* TIME_WAIT yang memiliki waktu 2 menit. Proses ini ditujukan untuk semua paket yang tidak mengikuti jalur tetap, dapat melalui aturan (*rule set*) yang kita miliki, bahkan ketika koneksi telah berhenti. Hal ini dimungkinkan oleh penggunaan suatu buffer waktu sehingga paket yang terjebak satu sama lain atau pada router yang lain tetap dapat mencapai *firewall* atau ujung dari koneksi [9].

Secara lengkap koneksi TCP dapat dilihat pada gambar 2.9 :



Gambar 2. 9 Diagram Koneksi TCP

2.4 Jenis-jenis Serangan

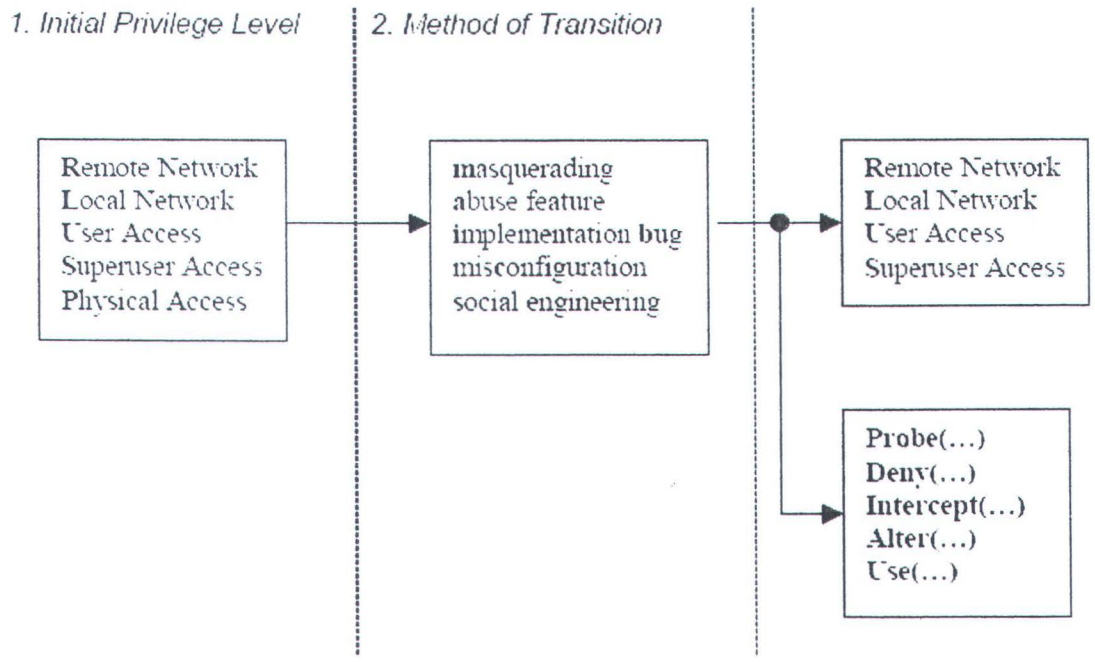
Penelitian ini membahas tentang pendeteksian intrusi. Untuk mendeteksi berbagai macam intrusi, terlebih dahulu harus dikenali bentuk-bentuk atau ciri-ciri dari masing-masing intrusi tersebut. Sehingga akan memudahkan kita dalam melakukan pendeteksiannya.

Untuk dapat mempermudah pengenalan dari pola yang ada pada berbagai macam serangan, dibuat suatu taxonomy untuk mengelompokkan macam-macam serangan yang ada.

Pengelompokan tersebut didasarkan pada beberapa feature [5] yaitu :

- Level privilege awal penyerangan
- Metode untuk melakukan exploit/metode untuk merubah privilege
- Jenis transisi antar level privilege
- Actions, tindakan yang dilakukan sebagai bagian dari penyerangan

Berikut ini gambar 2.10 dari jenis-jenis serangan untuk memudahkan mengelompokkan jenis-jenis serangan sesuai atribut-atribut seperti penjelasan diatas.



Gambar 2. 10 Diagram jenis serangan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Dari penjelasan diatas, serangan-serangan yang ada saat ini dapat dikelompokkan dalam empat besar kelompok serangan [5] , yang akan dijelaskan pada sub-sub-bab berikut.

2.4.1 Denial of Service

Sebuah serangan *denial of service* adalah bentuk serangan yang bertujuan untuk membuat proses komputing ataupun penggunaan memori yang melebihi batas kemampuan mesin sehingga pada akhirnya akan kesulitan untuk melayani atau bahkan menolak request yang sebenarnya legal. Terdapat bermacam-macam serangan denial of *service*. Beberapa dari serangan ini (misalnya *mailbomb*, *smurf*, *neptune*) memanfaatkan *service* yang memang diperbolehkan. Sebagian lainnya (*tear drop*, *ping of death*, *rose attack*) dengan sengaja membuat packet-packet salah sehingga membuat beberapa implementasi TCP/IP dari mesin-mesin tertentu menjadi *crash* .Ada juga beberapa macam *DoS* (*apache2*, *back*, *syslogd*) yang memanfaatkan kelemahan bug dari aplikasi-aplikasi daemon tertentu.

2.4.2 User to Root

Exploit user to root dimulai dengan adanya akses seorang penyerang ke suatu system dengan *account* normal *user* yang kemudian melakukan *exploit* kelemahan-kelemahan tertentu untuk mendapatkan akses *root* pada system.

Terdapat beberapa macam serangan *user to root*. Yang paling umum adalah *buffer overflow*. *Buffer overflow* terjadi ketika sebuah program melakukan *copy* data yang melebihi batas ke dalam sebuah *static buffer* tanpa melakukan pengechekan terlebih dahulu terhadap data tersebut. Sebagai contoh, seandainya

program meminta input nama lengkap dari user, programmer harus menentukan berapa byte buffer yang akan diperlukan untuk menyimpan nama lengkap tersebut. Seandainya diasumsikan program tersebut mengalokasikan 20 byte, maka seandainya dimasukkan input string sebanyak 35 byte, 15 byte yang terakhir akan men-*overwrite* stack, dimana ditempatkan instruksi selanjutnya yang akan dijalankan. Seorang penyerang dapat membuat data yang bertujuan untuk memanggil perintah sembarang yangakan dilakukan oleh sistem operasi. Programmer memang dapat menghilangkan *bug* ini dengan sedikit ketelitian, tetapi tetap saja terdapat beberapa tools yang mudah terkena jenis serangan ini. Kelompok lain dari serangan ini yaitu serangan yang memanfaat kelemahan program yang menjalankan perintah-perintah yang berhubungan dengan variabel *environment* dengan asumsi *environment* sesuai provilege yang mereka miliki. Contoh dari jenis serangan ini adalah *loadmodule*. Beberapa serangan user to root lainnya memanfaatkan *temporary file* yang tidak dimanajemen dengan baik. Dan yang terakhir adalah serangan dengan memanfaatkan *race condition* yang terjadi ketika beberapa *process/thread* mengakses resource yang sama. Walaupun cara-cara programming yang baik dapat mengurangi bug-bug semacam ini, pada kenyataannya bug-bug semacam ini selalu ada pada setiap versi UNIX maupun Windows.

2.4.3 Remote to User

Sebuah serangan *remote to user* terjadi jika seseorang mempunyai akses untuk melakukan pengiriman paket ke mesin tertentu melalui network, tetapi tidak mempunyai akses pada mesin tersebut, yang kemudian melakukan *exploit* pada

beberapa kelemahan system yang menjadikan dia sebagai *user* dari mesin tersebut. Beberapa dari serangan jenis ini memanfaatkan *buffer overflow* dari aplikasi-aplikasi server (sendmail, imap, named). Beberapa lainnya (dictionary, ftp-write, xsnoop) memanfaatkan konfigurasi system yang salah ataupun lemah. Juga terdapat beberapa serangan yang melibatkan *social engineering* misalnya *trojan horses.*

2.4.4 Probes

Dalam beberapa tahun terakhir, banyak sekali program yang dapat melakukan sca dan mencari kelemahan system secara otomatis ataupun untuk mengumpulkan informasi tentang mesin tertentu didistribusikan di internet. *Network probe* sangat membantu seorang penyerang untuk mengumpulkan informasi dan membuat rencana penyerangan selanjutnya. Seorang penyerang dengan sebuah peta mesin dan layanan yang ada pada suatu network dapat memanfaatkan informasi ini untuk mencari titik-titik yang lemah. Beberapa dari software *scanning* ini (satan, saint, mscan) bahkan sangat mudah dalam pemakaiannya sehingga memungkinkan seseorang yang tak berpengalaman sekalipun untuk melakukan pengechekan kelemahan dari ratusan bahkan ribuan komputer dalam satu *network.*

2.5 Klasifikasi Data

Pendeteksian intrusi dapat dianggap sebagai sebuah permasalahan klasifikasi : kita mencoba untuk mengelompokkan tiap-tiap audit record dalam

kelompok normal ataukah kelompok-kelompok lain, misalnya suatu jenis serangan tertentu.

Algoritma klasifikasi dapat memilih model yang paling tepat ketika dijalankan pada sebuah set record audit data yang telah dilabeli dengan label tertentu, misalnya label normal ataupun attack.

Algoritma ini akan memilih feature yang paling memberikan efek beda dalam pembentukan kelompok-kelompok data. Pemilihan feature-feature ini dapat dilakukan secara otomatis ataupun manual dengan bantuan seorang ahli.

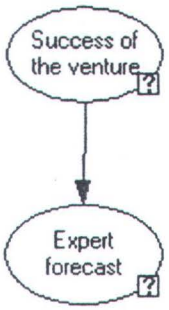
Terdapat banyak algoritma untuk melakukan klasifikasi, misalnya *Bayesian Network*, *Decision Tree*, *Divide and Conquer* dan masih banyak yang lainnya. Dalam penelitian ini hanya akan dibahas tentang *Bayesian Network*, dengan kekhususan pada jenis *Naïve Bayes*.

2.5.1 Bayesian Network

Bayesian Network (disebut juga *Belief Net*, *Bayesian Belief Net*, *Causal Probabilistic Network*, ataupun *Causal Network*) adalah suatu bentuk *Directed Acyclic Graph* dimana tiap-tiap node merepresentasikan suatu random variabel dan garis anak panah merepresentasikan keterkaitan probabilitas diantara node-node tersebut [1]. Struktur dari *Bayesian Network* dapat dibentuk dalam gambar, yang mana mengilustrasikan interaksi antara tiap-tiap variabel dalam pemodelan. Struktur dari directed graph dapat meniru hubungan causal yang terdapat dalam domain yang dimodelkan, walaupun bukanlah suatu keharusan. Akan tetapi seandainya strktur tersebut berbentuk causal, maka akan sangat berguna bagi seseorang untuk dapat memahami permasalahan dalam domain tersebut, dan akan

memudahkan dalam memprediksikan kejadian-kejadian karena perubahan-perubahan nilai-nilai variabel acak oleh manipulasi external.

Node-node dari *Bayesian Network* biasanya digambarkan berbentuk lingkaran atau elips. *Bayesian Network* pada gambar 2.11 dibawah ini merepresentasikan dua `variabel, Success dan Forecast, dan menunjukkan fakta bahwa terdapat ketergantungan hubungan diantara keduanya.



Gambar 2. 11 *Bayesian Network* sederhana

Sebuah *Bayesian Network* juga merepresentasikan hubungan quantitative diantara variabel-variabel dalam pemodelan. Yaitu merepresentasikan *joint probability distibution* diantara variabel-variabel tersebut. Setiap node memiliki distribusi *conditional probabilities* dari *predecessor* langsungnya. *Predecessor* adalah istilah untuk variabel diatasnya yang terhubung secara langsung. Node yang tidak mempunyai predecessor hanya memiliki *prior probability distribution.* Sebagai contoh, node Success pada contoh network sebelumnya memiliki *probability distribution* untuk kedua nilainya : Success dan Failure.

Node Forecast memiliki *probability distribution* sesuai nilai-nilainya (Good, Moderate, Poor) sesuai dengan nilai-nilai yang muncul pada *predecessor*-nya (node Success, dengan nilai Success dan Failure).

Struktur dari *Bayesian Network* dapat dibentuk dengan bantuan seorang ahli dalam bidang tertentu yang sedang dimodelkan. Selain itu, terdapat juga algoritma-algoritma untuk mencari struktur dari *Bayesian Network* secara otomatis dari suatu set data *training* yang telah ada. Kedua cara ini, manual dan otomastis dapat digabungkan apabila diperlukan untuk menemukan struktur yang paling tepat yang merepresentasikan domain knowledge tertentu.

Nama *Bayesian* berasal dari fakta bahwa *joint probability distribution* yang dihasilkan oleh suatu *Bayesian Network* adalah subyektif dan subyektifitas ini dapat berubah-ubah sesuai dengan adanya *evidence-evidence* baru dengan menggunakan Teorema Bayes.

2.5.2 Naïve Bayes

Bayesian Network didasarkan pada Teorema Bayes pada rumus yaitu :

$$\Pr[H \mid E] = \frac{\Pr[E \mid H] \times \Pr[H]}{\Pr[E]}$$

Rumus 2. 1 Teorema Bayes

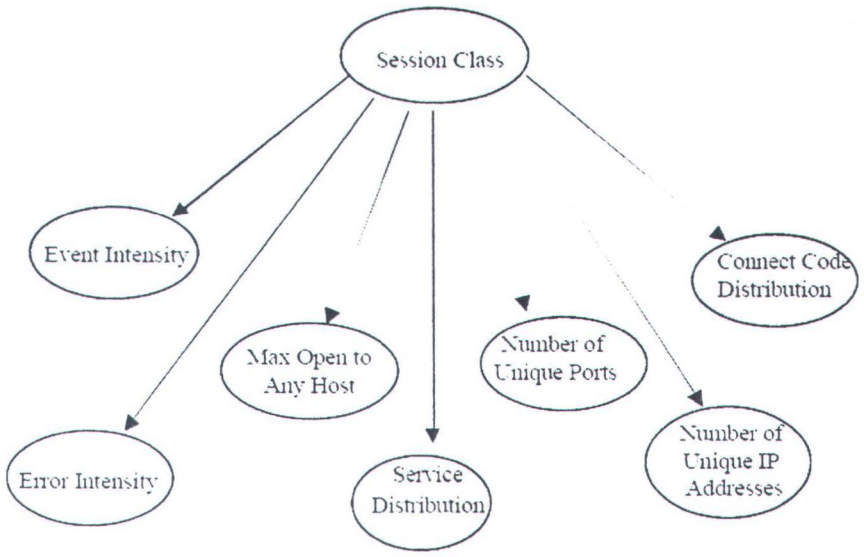
Dengan *Pr[A]* adalah probabilitas terjadinya event A, dan *Pr[A|B]* adalah probabilitas terjadinya event A setelah terjadi event B. *Pr[H]* adalah *prior probability*.

Disini hypothesis *P[H]* adalah normal atau attack, jadi misalnya *Pr[H=normal|E]* mempunyai arti probabilitas data normal dengan adanya *evidence-evidence* E. Disini E adalah satu set variabel *evidence* E_1, E_2, ... , E_n, sehingga rumus diatas menjadi :

$$\Pr[normal \mid E] = \frac{\Pr[E1 \mid normal] \times \Pr[E2 \mid normal] \times ... \times \Pr[En \mid normal] \times \Pr[normal]}{\Pr[E]}$$

Rumus 2. 2 Joint Distribution

Rumus 2.2 diatas adalah rumus *joint distribuiion* khusus untuk *Naïve Bayes*, yaitu suatu bentuk dari *Bayesian Network* yang tidak menghiraukan kenyataan akan adanya keterkaitan antar variabel dalam suatu *domain knowledge*, karena itulah metode ini disebut dengan *naïve*. Gambar 2.12 berikut ini gambar dari sebuah *Naïve Bayes*.



Gambar 2. 12 *Naïve Bayes* dengan banyak atribut

Walaupun begitu, dalam aplikasi nyata, metode bekerja sangat baik pada dataset-dataset nyata yang ada dan bahkan kadang kala mampu mengalahkan ketepatan metode-metode lain yang lebih kompleks. Selain itu metode ini mempunyai kelebihan dalam kecepatan komputasi, sehingga menjadi pilihan dalam pembuatan penelitian ini [15].

2.5.3 *Laplace Estimator*

Satu hal yang patut diperhatikan dalam metode ini yaitu adanya jenis nilai dari atribut tertentu yang hilang pada *training set*.

Seandainya terdapat atribut *connection status* dan atribut ini mempunyai dua jenis nilai, yaitu : "CLOSED" dan "ESTABLISHED". Kemudian seandainya data *training* set sama sekali tidak memasukkan data dengan record yang mengandung "CLOSED" sehingga *Pr[normal|E="CLOSED"]=0* maka kelika dalam aplikasi nyata terdapat data dengan nomor IP tersebut, *joint probability distribution*-nya akan dianggap nol karena nilai probabilitas *evidence* yang lain akan dikalikan dengan nol.

Kelemahan ini dapat dihindari dengan adanya *laplace estimator*, yaitu cara menghindari hal semacam diatas dengan menambahkan μ untuk semua semua jumlah data(penyebut) dan menambahkan *μ/jumlah kelompok nilai* [15]. Sebagai contoh untuk kasus diatas menjadi :

$$\Pr[\text{normal} \mid \text{E} = \text{"CLOSED"}] = \frac{0 + \mu/2}{\text{jumlah data} + \mu}$$

Rumus 2. 3 *Laplace estimator*

BAB 3

PERANCANGAN SISTEM

BAB 3
PERANCANGAN SISTEM

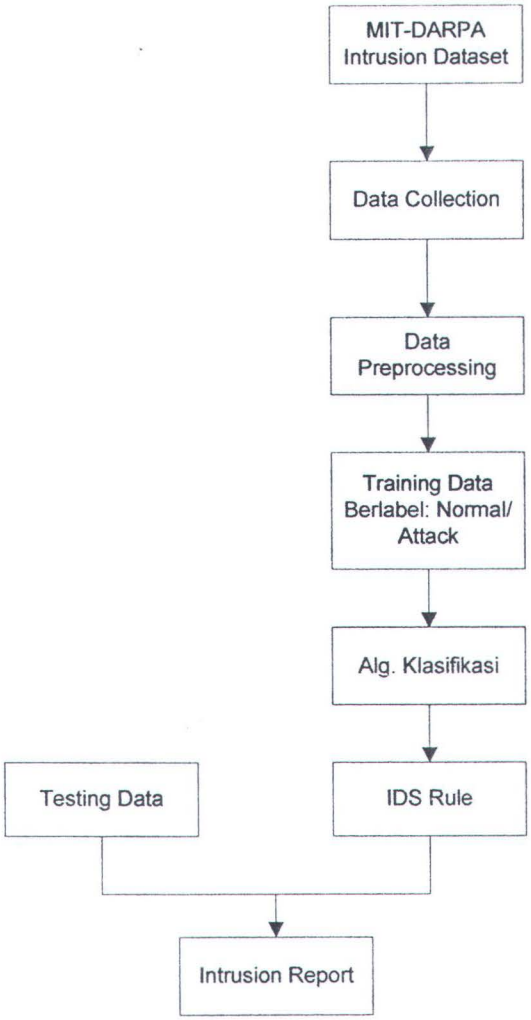
Pada bab ini dijelaskan mengenai perancangan dan pembuatan aplikasi. Perancangan aplikasi ditujukan untuk memberikan gambaran secara umum terhadap aplikasi yang dibuat. Hal ini berguna untuk menunjang pembuatan aplikasi sehingga kebutuhan akan aplikasi tersebut dapat diketahui. Selain itu perancangan aplikasi juga mempermudah dalam melakukan pengembangan lebih lanjut terhadap aplikasi yang dibuat. Tahap-tahap perancangan dan pembuatan aplikasi ini meliputi proses penentuan ruang lingkup sistem yang akan dibuat, analisa dan penentuan proses-proses yang terjadi dalam aplikasi yang akan dibuat serta mekanisme pengaturan data.

3.1 Gambaran Umum Sistem

Aplikasi yang akan dibuat ini merupakan aplikasi pendeteksi intrusi, yang tujuan utamanya adalah untuk memberikan peringatan dini kepada sistem administrator apabila terdapat usaha-usaha serangan.

Dalam membangun aplikasi pendeteksi intrusi ini terdapat tahap-tahap yang harus dilalui. Pertama-tama, perlu dilakukan proses pelatihan dengan data-data yang lampau, yang berisi koneski-koneksi normal dan koneksi-koneksi attack. Data ini haruslah menggambarkan data sebenarnya dari data network yang nyata. Kedua, dari pelatihan tersebut diperoleh *rule-rule* yang akan dipakai dalam pendeteksi oleh aplikasi dalam data network yang akan dicobakan. Ketiga adalahh

aplikasi pendeteksi itu sendiri. Berikut ini gambaran keseluruhan dari aplikasi pada penelitian ini.



Gambar 3. 1 Desain keseluruhan aplikasi pendeteksi intrusi

Karena aplikasi ini mendasarkan pendeteksian pada teorema bayes, maka aplikasi ini memerlukan *training* data terlebih dahulu untuk mengumpulkan *prior probabilities*. *Training* data ini dilakukan pada data network sesungguhnya, yang mana data-data tersebut masih harus diproses untuk dapat dianalisa. Proses ini disebut sebagai *preprocessing* data. Pada tahap ini dikumpulkan atribut-atribut yang akan digunakan sebagai atribut *classifier*. Setelah proses *training* data

dilakukan, maka *knowledge* yang telah didapat dapat digunakan untuk meramalkan suatu koneksi yang sedang terjadi termasuk kedalam kelompok normal ataukah intrusi.

Untuk aplikasi *training* pada penelitian ini terdapat sebuah program utama dan beberapa tools untuk mempercepat proses *training* data. Program utama ini mempunyai dua fungsi yaitu, pertama untuk mengumpulkan atribut-atribut dari data *raw tcpdump* ataupun langsung dari *ethernet card* untuk kemudian disimpan pada sebuah database untuk dipergunakan sebagai data *training*, kedua untuk melakukan proses pendeteksian setelah proses *training* data. Sedangkan tool-tool hanya digunakan untuk sekedar membantu kecepatan dan kemudahan proses development. Pada buku penelitian ini hanya akan dibahas program utama yang merupakan implementasi metode dan tujuan dari pokok bahasan utama pada penelitian ini.

Setiap proses klasifikasi, dibutuhkan atribut-atribut sebagai variabel pendeteksi. Atribut-atribut tersebut dikumpulkan dari packet-packet yang tertangkap oleh *libpcap* yang kemudian dikumpulkan menjadi satu *session* TCP. Ketika koneksi-koneksi ini selesai, program secara otomatis merekan atribut-atribut yang diperlukan Adapun definisi selesai hanya berlaku untuk sesi TCP. Setiap sesi TCP dianggap selesai ketika terdapat *flag* FIN dari client dan/atau dari server.

Atribut-atribut dari setiap koneksi data *training* disimpan kedalam database postgreSQL. Ketika data *training* telah terkumpul semua, dihitung

conditional probabilitas dari setiap atribut *given* klas *normal* dan *attack*, dan juga *prior probability* dari setiap klas.

Terdapat dua tahapan penting dalam proses *training* data. Tahap pertama adalah proses pengumpulan data. Tahapan kedua adalah data preprocessing. Disini data preprocesing adalah melakukan proses pelabelan data dengan label *normal/attack.*

Tahap paling akhir adalah proses pendeteksian itu sendiri. Seperti yang telah dijelaskan sebelumnya, program utama mempunyai dua fungsi, selain dapat digunakan untuk pengumpulan data, dapat juga digunakan untuk melakukan pendeteksian intrusi, yang mana merupakan tujuan utama dari penelitian ini.

Dalam pendeteksian, program ini memanfaatkan nilai-nilai yang telah dihasilkan melalui proses sebelumnya. Nilai-nilai ini tinggal dimasukkan dalam Teorema Bayes untuk mencari *P (*klas intrusi| *given events).* Dengan kata lain, pada proses ini dicari probabilitas normal dibanding probabilitas *attack* bedasar *event* yang terjadi dan data-data masa lalu (*training*).

Pada subbab-subbab berikut ini akan dijelaskan perancangan data dan perancangan proses dari hal-hal yang disebutkan diatas secara lebih mendetil yang disertai juga dengan diagram-diagram yang akan membantu untuk memahami proses perancangan program aplikasi secara keseluruhan

3.2 Perancangan Data

Dalam sub bab ini akan dijelaskan struktur data yang digunakan pada proses *training* dan struktur data yang digunakan pada aplikasi pendeteksi. Data yang dirancang disini disesuaikan dengan data *training* yang didapat dari dataset

milik Lincoln lab MIT-DARPA. Dataset ini adalah dataset standar untuk penelitian dibidang pendeteksian intrusi.

3.2.1 Struktur Data Untuk Proses *Training*

Dalam *training* data menggunakan dataset milik MIT-DARPA, pertama-tama aplikasi mengumpulkan data dari data *dump tcpdump*. Dari data ini didapat data-data koneksi. Data koneksi yang didapat adalah data tanpa label. Untuk melabeli label diperlukan informasi tambahan yang akan mencocokkan informasi tersebut dengan setiap data koneksi yang terkumpul.

3.2.1.1 Struktur Data Koneksi untuk Proses *Training*

Pada proses *training*, data yang diperlukan adalah atribut-atribut yang akan digunakan sebagai pendeteksian seperti yang telah dibahas pada sub bab 3.1.1 Selain itu, terdapat tambahan atribut-atribut yang akan berperan sebagai *key unique* dari setiap koneksi. *Key* ini akan diperlukan pada saat *preprocessing* data sebelum proses perhitungan probabilitas dilakukan.

Data-data ini akan diimplementasikan dalam bentuk tabel dalam suatu database. Aplikasi pendeteksi intrusi akan mengacu pada data tersebut ketika menghitung *prior probability* maupun *likelyhood* dari masing-masing atribut. Struktur data ini digambarkan pada tabel 3.1. pada halaman berikut. Terdapat delapan atribut yang akan disimpan dalam untuk keperluan data *training*. Tabel 3.1. juga terdapat keterangan yang menjelaskan asal dari setiap atribut-atribut tersebut.

3.2.1.2 Struktur Data Informasi untuk Pelabelan Data

Proses pelabelan data adalah proses memberikan label pada data koneksi yang telah dikumpulkan pada proses sebelumnya. Atribut *label* pada tabel 3.1 akan digenerate oleh proses pelabelan data. Untuk melakukan pelabelan data diperlukan suatu informai tambahan yang akan melakukan pencocokan melalui *key* dari setiap koneksi. Informasi ini harus disediakan untuk dapat memulai proses *training* data. Adapun informasi tambahan ini berisi beberapa hal sebagai berikut :

- IP asal
- *Port* asal
- IP tujuan
- *Port* tujuan
- Waktu koneksi
- Label

Dengan informasi diatas, proses pelabelan akan dapat dilakukan.

Tabel 3. 1 Atribut-atribut koneksi

No.	Nama Atribut	Jenis Atribut	Keterangan
1	*ip_src*	*Ignored*	Kelima atribut ini tidak dipakai dalam pendeteksian tetapi berfungsi sebagai *key* untuk pelabelan data. Selain itu juga sebagai informasi koneksi.
2	*port_src*		
3	*ip_dst*		
4	*port_dst*		
5	*timestamp*		

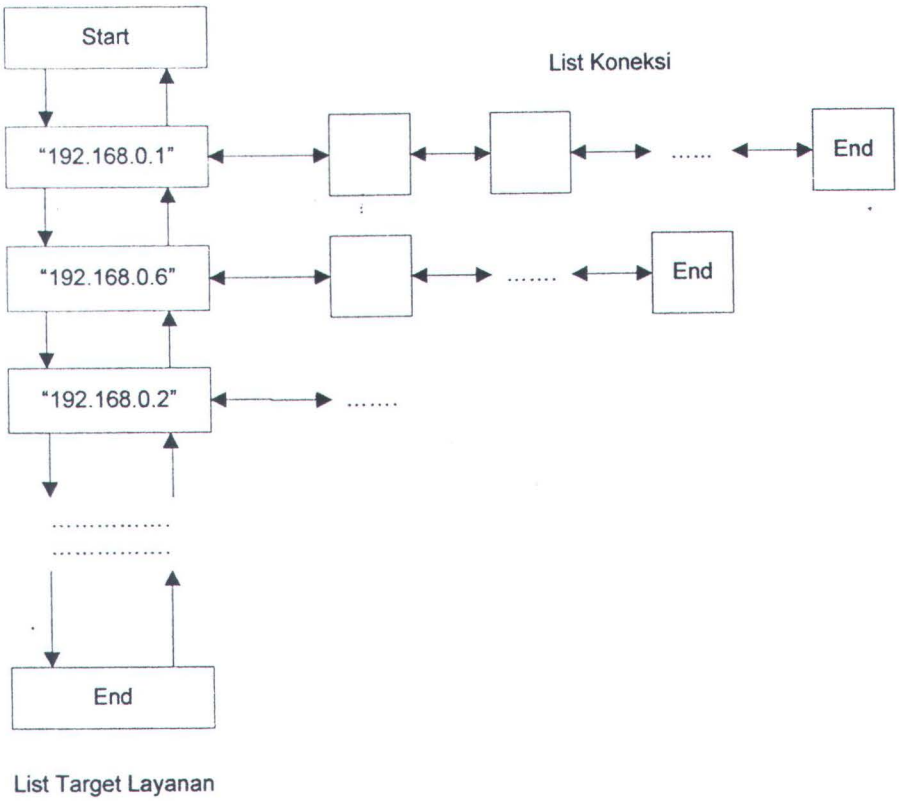
6	*conn_status*	Diskrit	Status koneksi paling akhir dari suatu koneksi. Macam-macam koneksi dikodekan dengan angka 0-9.
7	*p_svc_per_host*	Continue	Jumlah koneksi (persentase) dengan target host yang sama dalam waktu 2 detik terakhir.
8	*p_rej_per_host*	Continue	Jumlah koneksi (persentase) dengan target host yang sama yang ditolak oleh host tersebut dalam 2 detik terakhir.
9	*p_diff_host_per_svc*	Continue	Jumlah koneksi (persentase) yang mempunyai target layanan yang berbeda dalam 2 detik terakhir.
11	*p_rej_per_svc*	Continue	Jumlah koneksi yang memiliki target layanan yang sama yang ditolak oleh layanan tersebut dalam 2 detik terakhir.
12	*label*	Diskrit	Jenis klas dari koneksi-koneksi, apakah koneksi normal ataukah intrusi.

3.2.2 Struktur Data Untuk Penyimpanan Koneksi pada Aplikasi

Pada aplikasi, diperlukan suatu struktur data sementara untuk menyimpan koneksi-koneksi yang terjadi. Struktur ini haruslah sebuah struktur yang cepat dalam pengaksesannya dan juga cukup mudah dalam manajemennya. Karena kecepatan adalah suatu yang sangat penting maka struktur ini juga harus dapat diimplementasikan dalam memori. Karena penghitungan atribut-atribut pendeteksian didasarkan kepada target host maupun target *service* dan karena koneksi-koneksi yang ada diasumsikan sangat banyak sekali, maka waktu yang diperlukan untuk mencari dan kemudian menghitung atribut-atribut dari suatu koneksi tertentu akan menghabiskan waktu jika menggunakan list biasa. Begitu pula dalam pengecekan koneksi yang harus dilakukan untuk mencari koneksi-koneksi yang *error*.

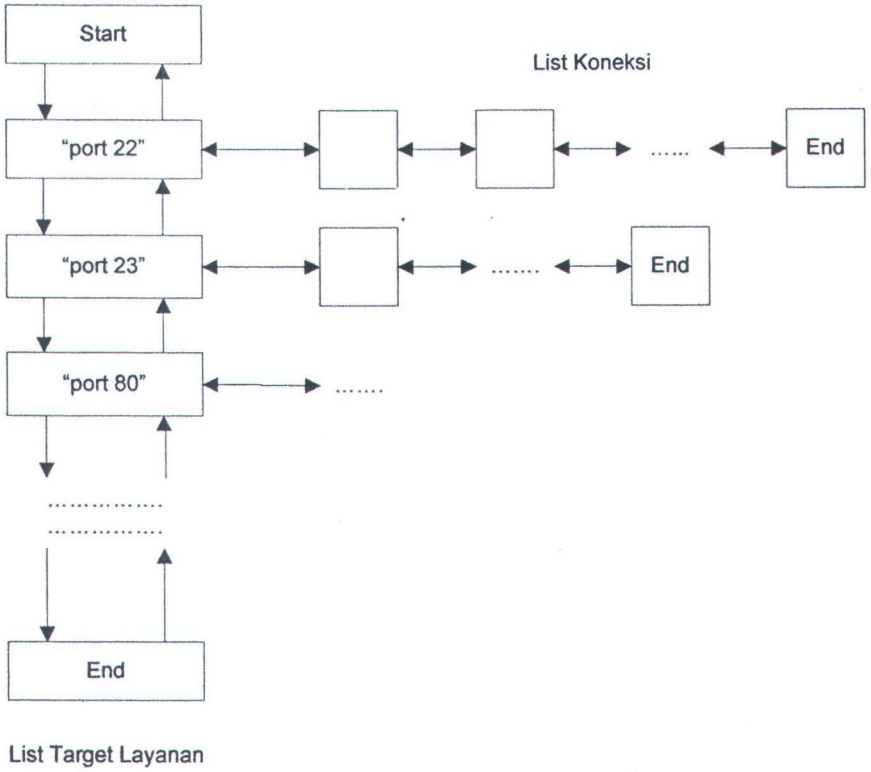
Untuk memenuhi persyaratan-persyaratan diatas, dibuatlah suatu struktur khusus untuk tempat penyimpanan koneksi. Struktur data ini adalah gabungan dari suatu *binary tree* dan *list*. Setiap *node* dari *binary tree* mempunyai *list* yang berisi item-item koneksi yang sedang terjadi. Untuk lebih jelasnya dapat dilihat pada gambar 3.2. dan 3.3.

Pada development selanjutnya, diharapkan pembuatan struktur ini dapat digantikan dengan bentuk-bentuk struktur data lainnya yang lebih cepat, misalnya *hash*. Sehingga, perancangan ini dibuat seflexible mungkin untuk perubahan-perubahan dimasa yang akan datang.



Gambar 3. 2 Struktur data list koneksi dengan target host tertentu

Dalam *binary tree* pencarian tidak dilakukan seperti halnya dalam list biasa, tetapi menggunakan sebuah metode *recursive* yang disebut *successive leaves.* Setiap node binary tree membagi data menjadi dua bagian, node anak sebelah kiri adalah data dengan nilai lebih kecil dan node anak sebelah kanan adalah data dengan nilai yang sama atau lebih besar. Karena sifat inilah binary tree lebih cepat dalam pencarian dan traversalnya. Pada gambar 3.2. dan 3.3. hanya dijelaskan bentuk data secara konsep, untuk memudahkan penjelasan. Gambar tersebut merepresentasikan bentuk secara fungsional chunk-chunk data yang disimpan dalam memory.



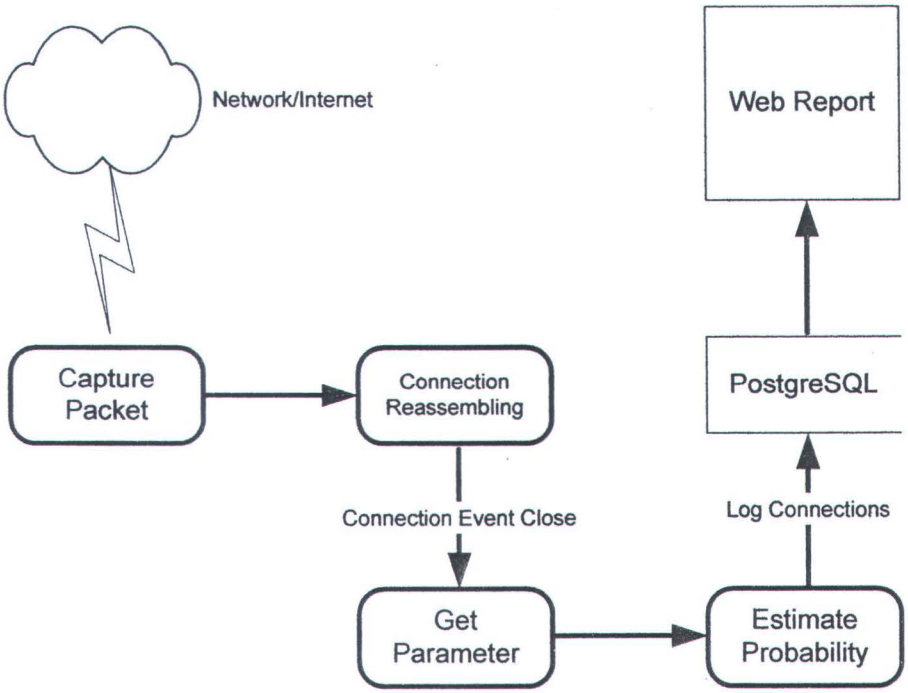
Gambar 3. 3 Struktur data list koneksi dengan target layanan tertentu

Terdapat dua buah list yang diperlukan. Pada gambar 3.2 menunjukkan struktur dari sebuah list yang mendaftar koneksi berdasar target host yang diakses. List utama berisi nilai-nilai host yang diakses. Setiap item dari list utama masing-masing mempunyai list yang berguna untuk menyimpan semua data koneksi yang sedang mengakses target host yang bersangkutan.

Gambar 3.3 adalah gambaran struktur data untuk menyimpan koneksi berdasar target layanannya. Pada list utama terdapat daftar layanan-layanan yang sedang diakses. Setiap item dari list utama masing-masing mempunyai list yang berguna untuk menyimpan semua data koneksi yang sedang mengakses layanan yang bersangkutan.

3.3 Flow Sistem Aplikasi Pendeteksian Intrusi

Untuk memudahkan development proses, maka haruş dipahami alur dari proses pendeteksian secara sederhana. Setelah itu, alur ini didapatkan, maka dilanjutkan dengan perancangan proses secara lebih detail pada sub-bab 3.4. Adapun alur dari sistem aplikasi sistem pendeteksian seperti terlihat pada gambar 3.4.



Gambar 3. 4 Flow aplikasi pendeteksian secara sederhana

Secara sederhana, gambar diatas menceritakan alur mulai dari penangkapan paket dari internet/network, kemudian paket-paket tersebut dikumpulkan menjadi koneksi-koneksi, yang kemudian ketika salah satu dari koneksi-koneksi tersebut menutup, maka akan dihitung parameter-parameter yang diperlukan untuk dapat menghitung pobabilitas dari koneksi tersebut. Dari probabilitas ini dapat ditentukan apakah koneksi tersebut suatu koneksi normal

ataukah serangan. Pada akhirnya, semua laporan disimpan dalam database dan ditampilkan melalui aplikasi web.

3.4 Perancangan Proses Aplikasi Pendeteksian Intrusi

Untuk membuat aplikasi pendeteksi intrusi pada penelitian ini terlebih dahulu dilakukan perancangan proses, serta perancangan alur. Perancangan proses berguna untuk mengintegrasikan semua proses yang terjadi dalam aplikasi serta menunjukkan alur perpindahan data pada aplikasi ini. Perancangan proses ini dilakukan dengan menggunakan Diagram Aliran Data (DAD) dengan menggunakan *Power Designer 6.1 32-bit ProcessAnalyst*. Proses-proses yang terjadi, mulai dari level 0 sampai level 2, dapat dijelaskan sebagai berikut:

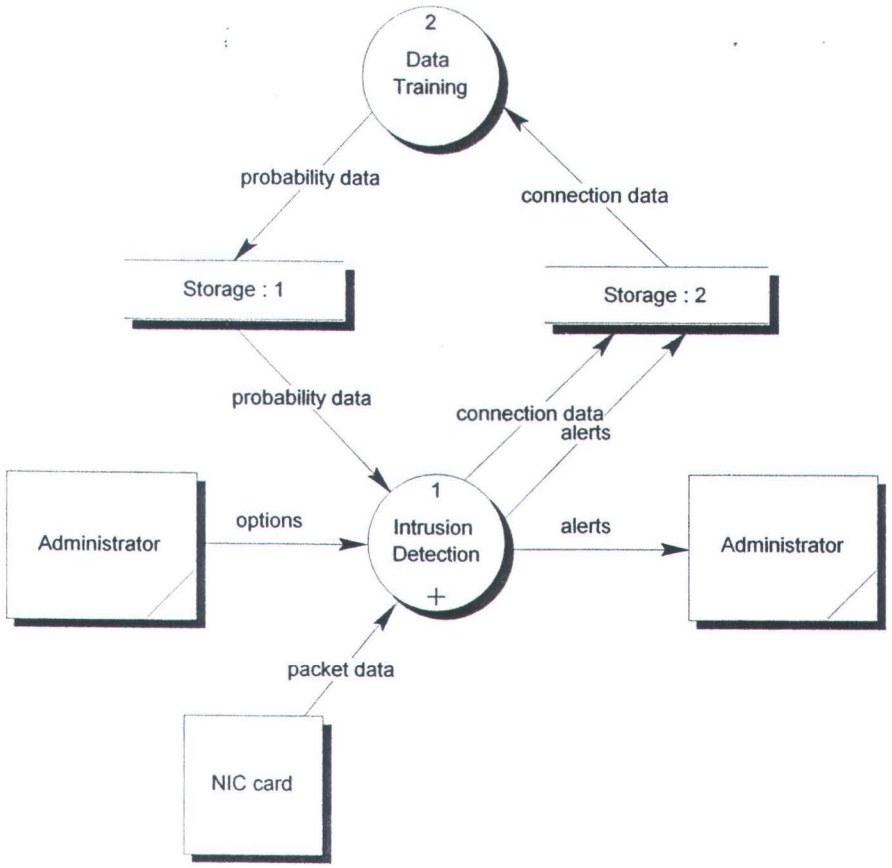
3.4.1 DAD Level 0

Pada proses Pendeteksian Intrusi pengguna memberikan masukan berupa nama *interface* dan opsi-opsi yang digunakan untuk mengatur mode aplikasi. Opsi-opsi ini antara lain :

- Mode *offline* | *real time*
- Mode *training* | *detection*
- Nama *NIC Card*

Gambar 3.5 dibawah ini adalah representasi dari DAD level 0. Mode *offline* adalah mode untuk pembacaan *raw data tcpdump,* sedangkan *real time* adalah mode untuk pembacaan data melalui *NIC card* secara langsung. Selain dari masukan user, proses ini juga menerima masukan data packet dari *NIC card* yang telah didefinisikan. Selain menghasilkan pesan *alert* kepada user yang kemudian

menyimpannya dalam database, proses ini juga menghasilkan data per koneksi yang disimpan pada database jika dipakai mode *training*.



Gambar 3. 5 DAD level 0

Mode *offline* adalah mode untuk pembacaan *raw data tcpdump,* sedangkan *real time* adalah mode untuk pembacaan data melalui *NIC card* secara langsung. Selain dari masukan user, proses ini juga menerima masukan data packet dari *NIC card* yang telah didefinisikan. Selain menghasilkan pesan *alert* kepada user yang kemudian menyimpannya dalam database, proses ini juga menghasilkan data per koneksi yang disimpan pada database jika dipakai mode *training*.

Pada DAD level 0 terdapat dua proses, yaitu Proses Pendeteksian dan pengolahan data, berikut ini akan dijelaskan masing-masing proses.

3.4.1.1 Proses Pendeteksian (1)

Proses pendeteksian diimplementasikan pada program utama, dijalankan dengan opsi-opsi tertentu untuk pengaturan mode pendeteksian, dan menerima input dari *network interface card* (NIC) berupa packet-packet datagram IP.

Proses tersebut selanjutnya mengolah packet-packet yang diterima menjadi satuan-satuan koneksi yang kemudian dapat menjadi input proses pendeteksian, yang akan dijelaskan pada level-level selanjutnya, atau untuk disimpan pada media storage yang pada akhirnya dapat digunakan sebagai input dari proses *Training* Data. Proses ini diturunkan pada level 1 yang akan dijelaskan sub-sub bab selanjutnya

3.4.1.2 Proses *Training* Data (2)

Pada proses ini data yang telah disimpan dalam media storage, yang dalam kasus ini menggunakan PostgreSQL, diolah untuk dilabeli sesuai kelas masing-masing. Proses ini mempunyai fungsi melabeli, dan menghitung data-data probabilitas setiap atribut. Probabilitas yang dikumpulkan adalah *conditional probability* dari setiap atribut klass (*attack / normal*). Dari proses pengolahan tersebut hasilnya akan dipakai oleh proses utama, yaitu proses pendeteksian, untuk mendeteksi data-data baru yang ditangkap / *capture.*

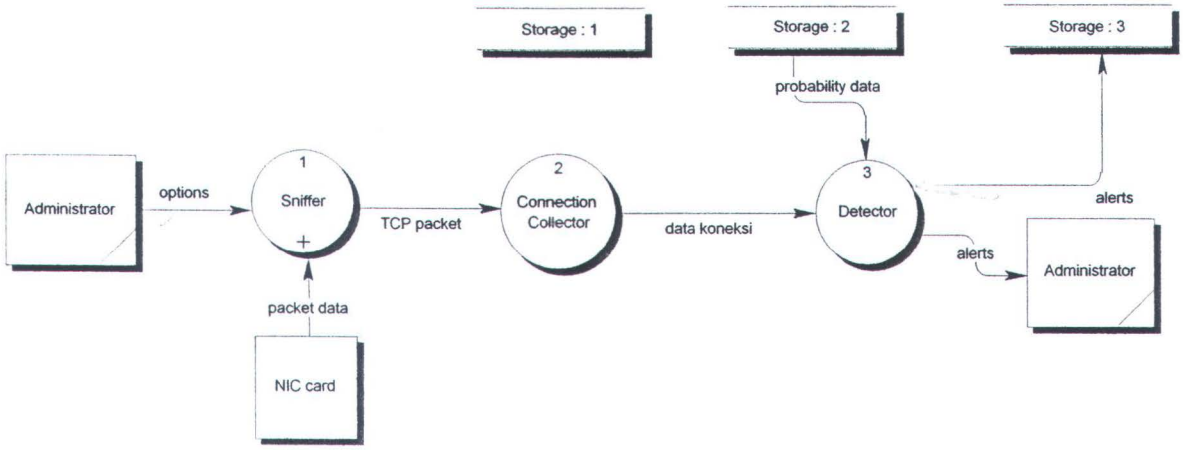
3.4.2 DAD Level 1

Pada level 1 yang merupakan kelanjutan dari proses *intrusion detection* pada level 0, terdapat beberapa proses yang mencakup proses *sniffer*, *connection*

collector, dan *detector*. Gambar 3.6 dibawah ini adalah gambar DAD level 1 yang merepresentasikan proses tersebut.

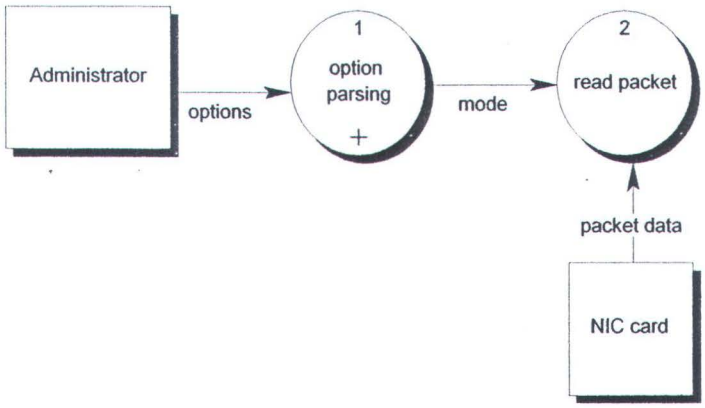
3.4.2.1 Proses *Sniffer* (1.1)

Pada proses ini dilakukan *capture* packet data dari NIC atau dari data raw hasil *dump* aplikasi *tcpdump*. Setelah itu proses akan mengeset variable-variabel inisialisasi yang dperlukan oleh sniffer untuk dapat berjalan. Untuk lebih jelasnya dapat dilihat dari diagram DAD level 2 pada gambar 3.7 pada halaman berikut.



Gambar 3. 6 DAD level 1 proses pendeteksian

Proses *option parsing* bertugas memparsing input dari argument command line yang dimasukkan oleh user. Proses *read packet* adalah implementasi *packet sniffer* sesungguhnya yang menggunakan *libpcap* sebagai library untuk meangkap packet dari NIC atau data *dump tcpdump*. Hasil dari proses *read packet* ini akan diberikan kepada proses penyusunan koneksi (*Connection Collector*, Gambar 3.6) yang mana pada proses penyusunan koneksi ini paket akan di-*reassembling* / dibangun menjadi koneksi-koneksi.



Gambar 3. 7 DAD level 2 proses sniffer

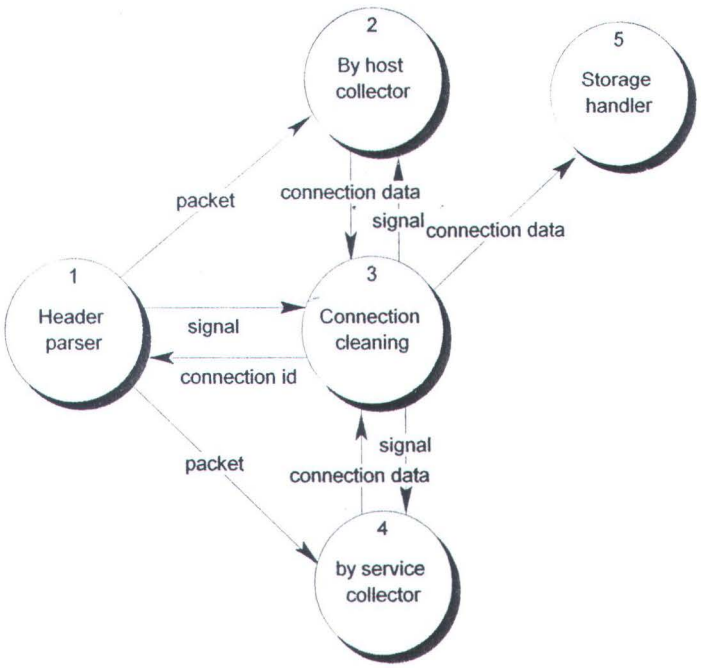
3.4.2.2 Proses Penyusunan Koneksi (1.2)

Proses ini bertanggung jawab untuk menyusun packet yang ditangkap oleh proses *sniffer*. Untuk itu diperlukan aturan-aturan RFC tentang TCP/IP supaya dapat dibentuk koneksi yang benar.

Pembentukan koneksi yang benar sangat penting artinya dalam penentuan hasil akhir pendeteksian. Pembentukan koneksi yang salah, akan menghasilkan pola data yang salah/sulit diprediksi sehingga kemampuan pendeteksian maupun pengumpulan pola akan sulit dilakukan.

Proses ini juga bertugas untuk menyimpan data koneksi kedalam database apabila dikehendaki, serta bertanggung jawab untuk memberikan data ke proses detection(1.3).

Gambar 3.8 pada halaman selanjutnya merupakan DAD level 2 dari proses penyusunan koneksi. Pada gambar tersebut, jalannya semua proses dapat dilihat dengan lebih jelas.



Gambar 3. 8 DAD level 2 proses penyusunan koneksi

Pada gambar 3.8 terlihat proses *header parser* menerima packet data dari proses sebelumnya. Oleh proses tersebut, header dibuka dan dilihat status koneksinya. Packet-packet akan dikumpulkan sesuai koneksi masing-masing. Bedasar aturan-aturan koneksi TCP/IP, *instance* koneksi akan dimasukkan kedalam list yang telah disediakan. Apabila aturan-aturan TCP/IP dilanggar makan, akan dimasukkan dikategorikan koneksi salah.

Terdapat dua proses yang menangani manajemen list koneksi. Satu proses khusus menangani list koneksi berdasarkan list koneksi berdasarkan target host, sedangkan yang lain menangani list koneksi berdasar target *service*.

Kemudian terdapat pula satu proses yang secara berkala melakukan pengecekan terhadap dua list tersebut. Proses pengecekan ini berfungsi untuk menghapus koneksi-koneksi yang statusnya telah selesai ataupun error. Selain menghapus, proses ini juga bertanggung jawab mengirimkan koneksi yang akan

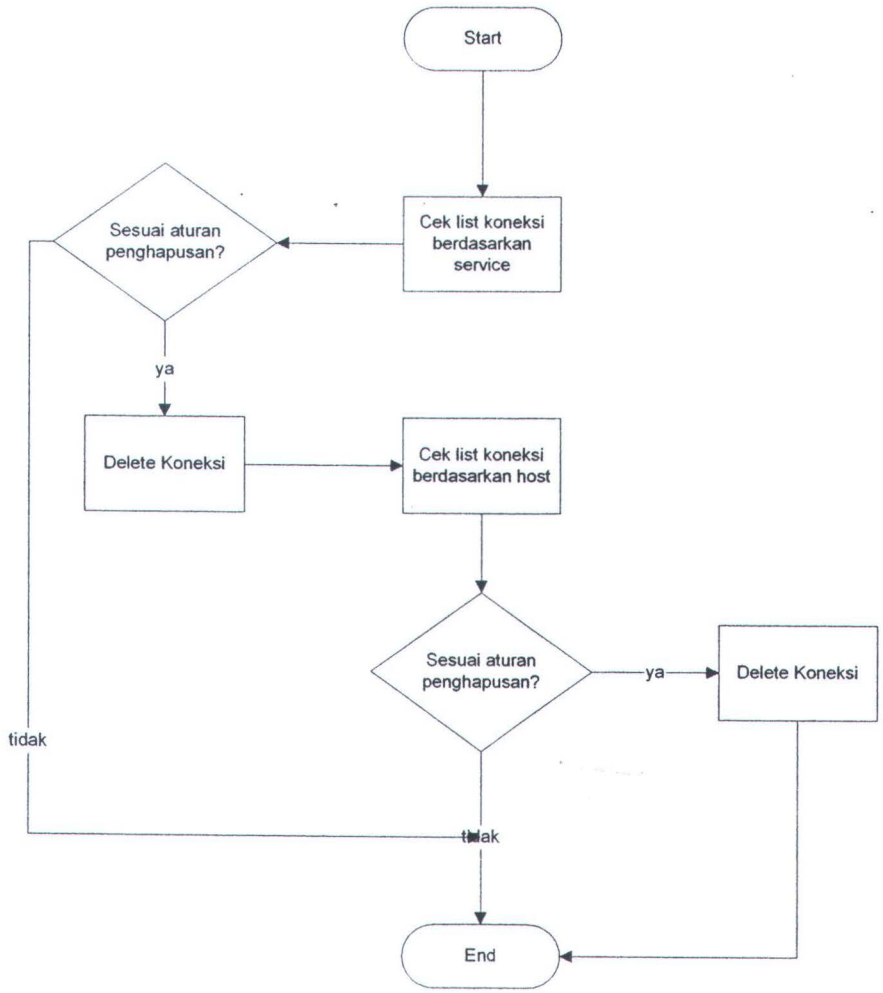
dihapus tersebut ke proses penyimpanan. Dan yang paling penting adalah proses ini juga berfungsi sebagai pengumpul atribut-atribut yang nantinya akan digunakan dalam proses pendeteksian. Proses ini dijalankan setiap 6 detik sekali, dengan asumsi dasar bahwa suatu permintaan koneksi, dalam implementasi TCP yang mengacu pada BSD *style*, akan menunggu jawaban *ack* dari *peer* selama 6 detik. Apabila koneksi tersebut tidak dijawab oleh *peer* maka TCP *entity* pada *kernel* akan mengirim permintaan request ulang. Proses ini akan mendelete semua koneksi yang berada dalam list yang memenuhi beberapa syarat yang telah ditentukan. Syarat-syarat tersebut antara lain:

- Suatu koneksi akan dihapus hanya jika status koneksinya bukan *established*
- Suatu koneksi akan dihapus hanya jika koneksi tersebut telah disimpan pada storage atau dideteksi
- Suatu koneksi akan dihapus hanya jika telah dikeluarkan dari kedua list koneksi yang ada

Penghapusan *instance* koneksi sangat penting karena jika koneksi-koneksi yang telah selesai/error tidak dihapus akan membuat beban pada memory yang pada akhirnya menyebabkan program *crash.*

Gambar 3.9 pada halaman selanjutnya adalah diagram alir dari proses *connection cleaning.* Gambar tersebut menjelaskan jalannya proses penghapusan *instance-instance* koneksi dari *instance* list-list dalam memori seperti yang telah dijelaskan sebelumnya.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 3. 9 Diagram alur *connection cleaning*

3.4.2.3 Proses Pendeteksian (1.3)

Pendeteksian menggunakan suatu metode yang berdasarkan teorema bayes. Proses ini mengimplementasikan teorema tersebut dalam kode-kode yang telah terintegrasi dengan proses-proses yang lain. Proses ini menerima input data-data *array conditional probability* dan *prior probability* dari storage, dan menghasilkan *posterior probability* yang bisa dikirim melalui standard output ataupun media storage.

Metode pendeteksian mendasarkan pada beberapa atribut yang telah diekstraksi dari data-data packet yang masuk. Atribut-atribut itu antara lain :

- Status Koneksi
- Jumlah koneksi ke target host yang sama dalam 2 detik terakhir yang mempunyai *service* yang sama dibagi jumlah koneksi ke target yang sama dalam 2 detik terakhir keseluruhan
- Jumlah koneksi ke target host yang sama dalam 2 detik terakhir yang hanya mengirimkan *SYN request* dibagi jumlah koneksi ketarget host yang sama dalam 2 detik terakhir secara keseluruhan.
- Jumlah koneksi ke target *host* yang sama yang ditolak(*reject*) oleh *host* tersebut selama 2 detik terakhir dibagi jumlah koneksi ke target *host* yang sama dalam 2 detik terakhir secara keseluruhan.
- Jumlah koneksi ke target *service* yang sama dalam 2 detik terakhir yang mempunyai target *host* yang berbeda dengan koneksi yang dianalisa dibagi jumlah koneksi yang mempunyai target *servcie* yang sama dalam 2 detik terakhir secara keseluruhan.
- Jumlah koneksi ke target *service* yang sama dalam 2 detik terakhir yang hanya mengirimkan *SYN request* dibagi jumlah keseluruhannya.
- Jumlah koneksi ke target *service* yang sama dalam 2 detik terakhir yang ditolak(*reject*) dibagi jumlah keseluruhannya.

Hasil perkalian *joint distribution* dari atribut-atribut itu akan diperbandingkan dengan data yang telah ada dari proses *training*. Hasil akhir akan menunjukkan kecenderungan kepada salah satu keadaan, yaitu normal atau intrusi.

3.5 Perancangan Antarmuka Aplikasi Pendeteksian Intrusi

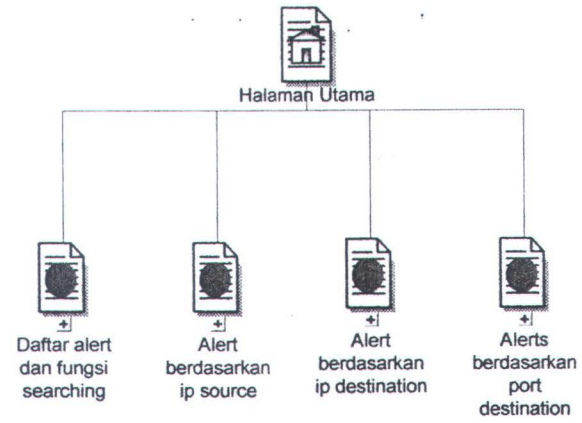
Aplikasi pendeteksi intrusi yang dibuat ini merupakan aplikasi berbasis Linux yang mana dijalankan dengan mode *console*, karena pengaksesannya menggunakan *putty*. Sehingga kalau dilihat dari sisi pengguna, aplikasi ini kurang *user friendly*. Ini disebabkan tampilan yang gelap dan kurang menarik, serta model penampilannya yang mengharuskan pengguna mengetikkan perintah-perintah supaya tampilan yang diinginkan dapat muncul di layar.

Oleh karena itu diperlukan aplikasi yang memudahkan pengguna untuk dapat melihat tampilan yang diinginkan serta memberikan kenyamanan dalam melihatnya. Dengan memperhatikan kemudahan akses data maka perancangan anatarmuka yang dibuat adalah palikasi antarmuka berbasis web.

Dalam merancang aplikasi antarmuka ini harus diperhatikan hal-hal yang diperlukan oleh pengguna dalam pembuatan antarmuka, antara lain :

- Menampilkan semua log pesan *alert* beserta atribut-atributnya dalam jangka waktu tertentu.
- Dapat menampilkan hasil pencarian berdasarkan IP asal dan/atau IP tujuan dan *port* tujuan dalam jangka waktu tertentu.
- Dapat menampilkan daftar alert berdasarkan banyaknya IP asal yang sama dalam jangka waktu tertentu.
- Dapat menampilkan daftar alert berdasarkan banyaknya IP tujuan yang sama dalam jangka waktu tertentu.
- Dapat menampilkan daftar alert berdasarkan banyaknya *port* tujuan yang sama dalam jangka waktu tertentu

Adapun site map dari halaman antarmuka digambarkan pada gambar 3.10 berikut ini.



Gambar 3. 10 diagram sitemap antarmuka *web client*

BAB 4

IMPLEMENTASI SISTEM

BAB 4
IMPLEMENTASI SISTEM

Pada pembahasan perancangan perangkat lunak telah digambarkan secara umum mengenai rancangan sistem Aplikasi *Sistem Pendeteksian Intrusi*. Pada bab ini akan dijelaskan mengenai lingkungan implementasi yang digunakan dan implementasi proses sesuai dengan yang dijelaskan pada bab sebelumnya

4.1 Lingkungan Implementasi Aplikasi Pendeteksian Intrusi

Dalam implementasi untuk Aplikasi *Sistem Pendeteksian Intrusi* ini, sistem perangkat keras yang digunakan adalah Athlon XP+ 2000 MHz, RAM 256 M dan Harddisk 40 GB 7200 rpm. Aplikasi ini berjalan dalam lingkungan FreeBSD dengan *gcc* dan *packet capture library* (libpcap) yang sudah terinstal.

4.2 Implementasi Data Aplikasi Pendeteksian Intrusi

Berdasarkan tahap perancangan terdapat dua bagian data yang akan diimplementasikan.

4.2.1 Implementasi Data untuk Proses *Training*

Pada tahap implementasi untuk data *training*, sesuai dengan tahap perancangan, terdapat dua bagian yang harus diimplementasikan. Satu bagian diimplementasikan ke dalam suatu *RDBMS* dan satu bagian dalam bentuk file. *RDBMS* yang dipilih adalah *postgreSQL* karena sifatnya yang *opensource* sesuai dengan komponen-komponen pendukung aplikasi lainnya.

Pada proses *training* diperlukan dua buah input. Satu adalah input data network, yang berbentuk *dump tcpdump* dan yang lain adalah input label dari data network tersebut, yang ditempatkan dalam sebuah file terpisah. Model data ini mengikuti file yang didistribusikan oleh Lincoln lab. MIT-DARPA yang digunakan dalam proses aplikasi penelitian ini.

4.2.1.1 Implementasi Struktur Data Koneksi

Data koneksi yang akan digunakan sebagai data *training* disimpan dalam bentuk table. Pada gambar 4.1 dibawah ini terdapat *script create table* dari tabel tersebut.

```
create table connections (
  ip_src               inet     ,
  ip_dst               inet     ,
  port_src             integer  ,
  port_dst             integer  ,
  timestamp            bigint   ,
  conn_status          smallint ,
  p_svc_per_host       real,
  p_s0_per_host        real,
  p_rej_per_host       real,
  p_diff_host_per_svc  real,
  p_s0_per_svc         real,
  p_rej_per_svc        real,
  label                smallint
)
```

Gambar 4. 1 Perintah SQL untuk create table koneksi

4.2.1.2 Implementasi Struktur Data Informasi untuk pelabelan

Informasi ini berisi data-data koneksi sesuai dengan file *dump tcpdump* yang telah dikumpulkan data koneksinya. Pada gambar 4.2 berikut terdapat format datanya.

```
1 01/23/1998 16:56:12 00:01:26 telnet 1754 23 192.168.1.30 192.168.0.20 0 -
2 01/23/1998 16:56:15 00:00:13 ftp 1755 21 192.168.1.30 192.168.0.20 0 -
.....
.....
14 01/23/1998 16:56:43 00:00:03 smtp 1783 25 192.168.1.30 192.168.0.20 0 -
15 01/23/1998 16:56:45 00:00:00 http 1784 80 192.168.1.30 192.168.0.40 1 phf
16 01/23/1998 16:56:49 00:00:14 ftp 43504 21 192.168.0.40 192.168.1.30 0 -
```

Gambar 4. 2 Format data text dari file tcpdump.list

Tidak semua informasi diperlukan oleh aplikasi ini. Yang diperlukan adalah informasi yang berfungsi membuat unique setiap satu koneksi dengan koneksi yang lain. Hanya kolom kedua, ketiga, ketujuh, kedelapan, kesembilan, kesepuluh, dan kesebelas yang diperlukan. Kolom kedua adalah tanggal koneksi, kolom ketiga adalah waktu koneksi, kolom ketujuh adalah port asal, kolom kedelapan adalah port tujuan, kolom kesembilan adalah alamat IP asal, kolom kesepuluh adalah alamat IP tujuan dan kolom kesebelas adalah label *normal/attack.*

4.2.2 Implementasi Struktur Data Penyimpanan Koneksi dalam Aplikasi

Untuk implementasi perancangan data list koneksi, diperlukan implementasi dari *associative array,* dan *linked list.* Dalam C++ telah tersedia *standard template library* yang berisi semua komponen-komponen yang diperlukan.

List koneksi ini terdiri dari dua jenis, satu untuk menyimpan koneksi yang urut bedasar target host, dan yang lain untuk menyimpan koneksi berdasar layanan target. Gambar 4.3 pada halaman berikut, adalah kode dari deklarasi list-list tersebut.

```
/* list untuk menyimpan koneksi */
typedef list<ConnItem*> ContainerQue;
typedef ContainerQue::iterator IterQue;
typedef ContainerQue::reverse_iterator IterQue_r;
/* map untuk menyimpan list koneksi */
/* berdasarkan host target */
typedef map<string,ConnList*> ContainerMap;
typedef ContainerMap::iterator IterMap;
typedef ContainerMap::reverse_iterator IterMap_r;
/* berdasarka service target */
typedef map<u_short,ConnList*> ContainerMap2;
typedef ContainerMap2::iterator IterMap2;
```

Gambar 4. 3 Implementasi struktur data penyimpan koneksi pada bahasa C++

4.3 Implementasi Proses Aplikasi Pendeteksian Intrusi

Pada sub bab ini, akan diberikan *code snippet* dari class-class yang dibentuk berdasarkan proses-proses yang telah dijelaskan sebelumnya. Untuk memudahkan proses development dan maintenance program, diputuskan untuk menggunakan konsep OOP sehingga terbentuk beberapa class sesuai dengan fungsi dan kegunaan masing-masing.

Adapun daftar file header yang berisi fungsi-fungsi implementasi beserta keterangannya dapat dilihat pada table pada halaman berikutnya.

4.3.1 Implementasi Proses DAD level 0

Proses ini diimplementasikan oleh program utama, yaitu bn_ids.cpp. Dalam gambar 4.4. pada halaman selanjutnya, terdapat *pseudo code* dari program utama.

```
1.      set option to command argument
2.      initiate NetworkSniffer
3.      check option
4.      if option offline=true
4.1     call sub sniffer from offline data
5       if option offline=false
5.1     call sub sniffer from realtime data
6.      stop
```

Gambar 4. 4 Program utama

Dari potongan kode program utama diatas, dapat dilihat bahwa program utama memanggil *instance* object NetwrokSniffer. Object NetworkSniffer ini adalah gerbang utama untuk masuk pada proses-proses yang digambarkan pada DAD level 1.

4.3.2 Implementasi Proses DAD level 1

Pada DAD level 1, terdapat 3 kelas utama yang terlibat dan beberapa kelas-kelas pendukungnya. Berikut ini hanya akan dibahas dari ketiga kelas tersebut.

4.3.2.1 Proses *Sniffer*

Proses ini diimplementasikan oleh kelas *NetworkSniffer* yang didefinisikan pada file header NetworkSniffer.h. Berikut ini definisi dalam NetworkSniffer.h

Untuk dapat menenggunakan library libpcap, harus disetting terlebih dahulu parameter-parameternya. Parameter-parameter tersebut antara lain id NIC, subnetmask jaringan, dan beberapa informasi yang berkenaan dengan jaringan.

Untuk melakukan capture, terdapat dua fungsi utama yang dipakai, yaitu *pcap_open_live() / pcap_open_offline* dan *pcap_loop()*.

Fungsi *pcap_open_live()* berguna untuk capture data real time dari suatu NIC, yang menjadi parameter input dari fungsi tersebut, sedangakn fungsi *pcap_open_offline()* berguna untuk capture data dari file dump *tcpdump*. Sedang fungsi *pcap_open_loop()* adalah sebuah API dari libpcap yang akan memanggil *callback function,* yang merupakan salah satu parameter dari fungsi ini, setiap kali ada sebuah paket data yang lewat.

Fungsi GotPacketFuncWrapper() adalah *callback function* sebagai parameter dari fungsi *pcap_loop*. Pada fungsi inilah semua proses capture packet dilakukan. Cuplikan *pseudo code* pada gambar 4.5. dibawah ini memberikan gambaran yang dilakukan oleh fungsi tersebut.

```
1.    While packet data captured by sniffer do
1.1   Set data_packet to packet data from sniffer
1.2   if data_packet=TCP data
1.2.1 call sub TCP_assembling
2.1   return
```

Gambar 4. 5 Fungsi untuk *capture* paket

Pada fungsi tersebut teedapat kelas ConnAssembler yang merupakan gerbang menuju proses *connection collector*. Semua packet yang terdeteksi sebagai TCP disimpan kedalan instance kelas *Protocol* yang selanjutnya dijadikan parameter untuk memnanggil fungsi *CollectPacket()* yang merupakan metode anggota kelas *ConnAssembler*. Instance dari kelas *ConnAssmebler* dan Protocol dinisialisasi pada *contructor* kelas *NetworkSniffer*.

4.3.2.2 Proses Penyusunan Koneksi

Penyusunan kembali paket-paket IP menjadi suatu koneksi TCP utuh adalah salah satu yang paling penting dari keseluruhan proses. Kesalahan pada tahap ini akan mempengaruhi hasil secara keseluruhan. Bahkan pada kasus-kasus tertentu, kesalahan dalam implementasi proses ini akan menyebabkan program *crash* bahkan juga Sistem Operasi yang menjadi host program. Untuk memudahkan pembahasan, maka proses ini diturunkan menjadi DAD level 2 seperti yang telah dijelaskan dalam Bab 3.

Pada proses ini diperlukan sebuah struktur data untuk menyimpan semua koneksi. Selain sebagai media penyimpanan, struktur ini juga memudahkan dalam penggalian atribut-atribut yang diperlukan dalam proses pendeteksian. Tedapat dua macam list yang dibutuhkan. Satu untuk menyimpan koneksi berdasarkan target host, dua untuk menyimpan koneksi berdasarkan target *service*. Untuk itu didefinisikan tiga macam struktur data yang memanfaatkan struktur data dari *standard template library* C++. Tedapat sebuah *map* untuk target host yang mempunyai *list* untuk setiap target host yang berbeda. Kemudian sebuah *map* lagi yang mempunyai *list* untuk setiap target *service* yang sama. *Semua list* ini, yang terdapat dalam *map* target host maupun *map* target *service*, mengacu pada *instance* koneksi yang sama untuk menghemat komputasi dan alokasi memori.

Penjelasan lebih mendetail beserta beberapa implementasi fungsi-fungsi dari proses ini diberikan sub bab sub bab berikut ini.

4.3.2.2.1 Proses Pemilahan Header

Pada fungsi dibawah ini dilakukan pengecekan header bedasar aturan-aturan TCP. Pada gambar 4.6 berikut ini adalah *pseudo code dari* header parsing.

```
1.    Set flag to TCP header flag
2.    if flag=SYN and no connection or
      last_status=CONN_CLOSED or last_status=CONN_ERROR
2.1     call sub insert connection
2.2     set conn_status to CONN_REQUEST
3.    if flag=SYN and ACK and last_status=CONN_REQUEST
3.1     set conn_status to CONN_SYN_ACK_SENT
3.2     update connection
4.    if flag=ACK and last conn_status=SYN_ACK_SENT
4.1     set conn_status to CONN_ESTABLISHED
4.2     update connection
5.    if flag=FIN and last_status=CONN_ESTABLISHED
5.1     conn_status=CONN_CLOSED
5.2     update connection
6.    return
```

Gambar 4. 6 Fungsi untuk parsing header TCP

Pada kode diatas terdapat dua fungsi yang penting yaitu, *UpdateConn()* dan *InsertConn()*. Fungsi-fungsi itulah yang berfungsi menangani perubahan dan pembentukan instance koneksi baru . Selain itu, fungsi ini juga bertugas memanggil proses pengecekan list koneksi setiap interval enam detik.

4.3.2.2.2 Proses Penyimpanan Koneksi

Proses ini dimulai dengan peyimpanan koneksi ke *list* berdasarkan target host kemudian dilanjutkan dengan penyimpanan kedalam *list* bedasarkan target *service*. Setiap langkah tersebut dilakukan dengan melakukan penelusuran pada setiap list hingga ditemukan *instance* koneksi yang berkesesuaian. Penelusuran ini sangat cepat dilakukan karena struktur data *map* adalah sebuah *binary tree link list*.

Pada gambar 4.7. berikut diberikan *pseudo code* dari fungsi penyimpanan *list* berdasarkan host. Untuk fungsi penyimpanan berdasarkan *service* langkah-langkah yang dilakukan adalah sama.

```
1.  Get listOfConnectionByHost
2.  Get currentConnection
2.  loop listOfConnectionByHost from start to stop do
 2.1     if currentConnection is found
 2.1.1   update connection atributes with new data
 2.2     return
3.  if currentConnection is not found
 3.1     call sub insert new connection into
                  listOfConnectionByHost

4.  return
```

Gambar 4. 7 Fungsi untuk register koneksi kedalam list

4.3.2.2.3 Proses Pengecekan List Koneksi

Pada Proses ini dilakukan pengecekan semua koneksi yang terdapat dalam kedua *list*. Proses ini sangat penting karena koneksi yang telah menutup koneksi tidak langsung dihapus melainkan karena masih dibutuhkan untuk penghitungan salah satu atribut pendeteksian yang menghitung jumlah koneksi dalam dua detik terakhir. Karena itu proses ini sangat penting artinya karena berguna sebagai pembersih memori. *Instance* yang sudah melewati waktu enam detik dan tidak direferensi oleh *list* yang lain akan dihapus oleh proses ini. Selain itu, untuk *instance* yang tidak pernah mengirimkan tanda menutup koneksi akan dapat dihapus dari *list* dan kemudian disimpan oleh proses penyimpanan atau dideteksi oleh proses pendeteksian.

Terdapat dua fungsi untuk masing-masing *list*. Pada gambar 4.8 berikut ini hanya diberikan *pseudo code* fungsi penghapusan *list* koneksi berdasarkan target host.

```
1.    for each connection in listOfConnectionByHost
 1.1    if connection.start_time > 6 second and conn_status<>
           ESTABLISHED
 1.1.1    if not connection had not been saved
 1.1.1.1    call sub saved connection
 1.1.2    delete connection

2.    return
```

Gambar 4. 8 Fungsi untuk membersihkan list

4.3.2.3 Proses Pendeteksian

Proses pendeteksian adalah proses paling akhir dari system. Hasil keluaran proses ini adalah tujuan dari pembuatan program. Proses ini diimplementasikan kelas *Bayes* yang didefinisikan pada file header *Bayes.hpp*. Pada gambar 4.9 berikut ini terdapat *preudo code* dari *Bayes::Detect()*.

```
1.    get connection from connection assembler
2.    get probabilities for every connection attributes
3.    call sub bayes joint distribution calculation
4.    return result
```

Gambar 4. 9 Fungsi pendeteksian

Dari cuplikan kode diatas, dapat dilihat bahwa untuk menghitung hasil probabilitas sesuai dengan teorema bayes diperlukan nilai conditional probabilitas untuk semua atribut terhadap kelas *normal* maupun *attack* pun juga nilai probabilitas *prior* untuk masing-masing kelas. Untuk mendapatkan nilai-nilai probabilitas tersebut didapat dengan memanggil fungsi-fungsi tertentu yang dibuat khusus sebagai API untuk menghitung nilai-nilai probabilitas tersebut.

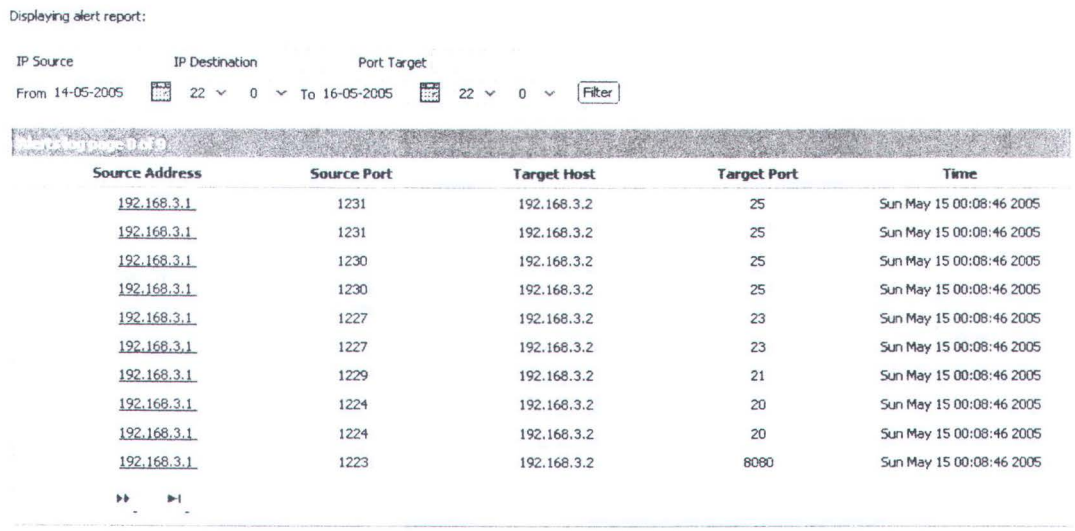
4.4 Implementasi Antarmuka Aplikasi Pendeteksian Intrusi

Aplikasi antarmuka pada penelitian ini digunakan sebagai media untuk menampilkan laporan yang diperoleh aplikasi pendeteksi.Aplikasi client ini menggunakan bahasa *php* sebagai server side programming. Bahasa ini dipilih karena kemudahan penggunaannya dan kecepatannya dalam waktu komputasi.

Berdasarkan perancangan antarmuka pada bab 3, dilakukan implementasi antarmuka sebagai berikut ini.

4.4.1 Implementasi Daftar Alert dan Fungsi Searching

Tampak pada gambar 4.10., terdapat daftar semua koneksi yang dicurigai sebagai serangan. Pada halaman tersebut juga terdapat fasilitas untuk memfilter data berdasarkan jangka waktu tertentu, IP asal, IP tujuan, dan *port* tujuan.


Displaying alert report:

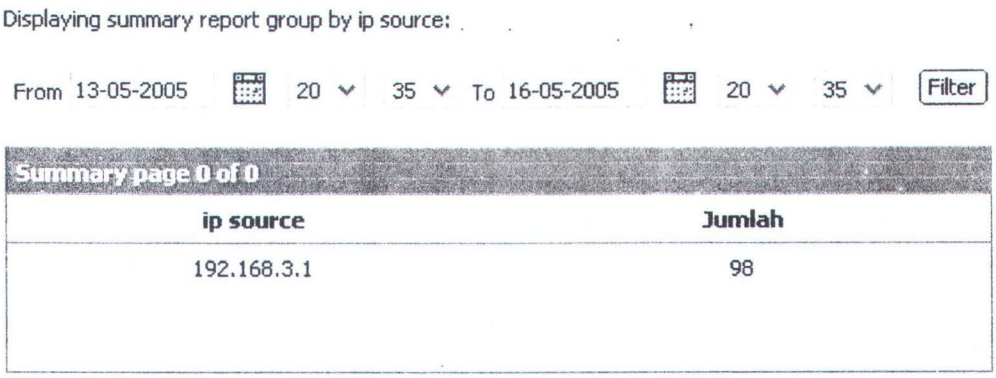
IP Source IP Destination Port Target

From 14-05-2005 22 0 To 16-05-2005 22 0 Filter

Source Address	Source Port	Target Host	Target Port	Time
192.168.3.1	1231	192.168.3.2	25	Sun May 15 00:08:46 2005
192.168.3.1	1231	192.168.3.2	25	Sun May 15 00:08:46 2005
192.168.3.1	1230	192.168.3.2	25	Sun May 15 00:08:46 2005
192.168.3.1	1230	192.168.3.2	25	Sun May 15 00:08:46 2005
192.168.3.1	1227	192.168.3.2	23	Sun May 15 00:08:46 2005
192.168.3.1	1227	192.168.3.2	23	Sun May 15 00:08:46 2005
192.168.3.1	1229	192.168.3.2	21	Sun May 15 00:08:46 2005
192.168.3.1	1224	192.168.3.2	20	Sun May 15 00:08:46 2005
192.168.3.1	1224	192.168.3.2	20	Sun May 15 00:08:46 2005
192.168.3.1	1223	192.168.3.2	8080	Sun May 15 00:08:46 2005

Gambar 4. 10 Implementasi halaman daftar alert beserta fungsi search

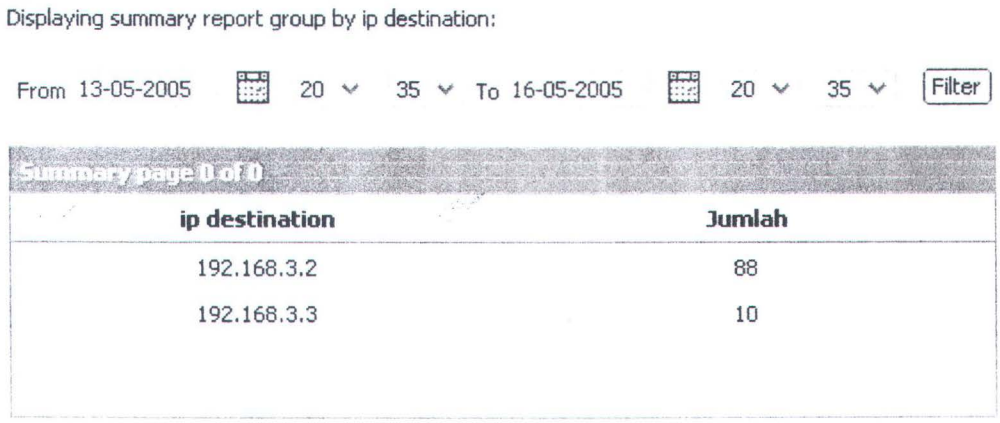
4.4.2 Implementasi Daftar Alert *group by* IP Asal



Gambar 4. 11 Implementasi halaman daftar alert *group by* IP asal

Pada gambar 4.11 diatas, terdapat implementasi daftar alert yang didasarkan pada banyaknya IP asal yang sama. Daftar tersebut direquest berdasarkan jangka waktu tertentu.

4.4.3 Implementasi Daftar Alert *group by* IP Tujuan



Gambar 4. 12 Implementasi halaman daftar alert *group by* IP asal

Pada gambar 4.12 diatas, terdapat implementasi daftar alert yang didasarkan pada banyaknya IP tujuan yang sama. Seperti halnya daftar alert berdasarkan alamat IP asal, daftar tersebut juga dapat direquest berdasarkan jangka waktu tertentu.

4.4.4 Implementasi Daftar Alert *group by port* tujuan

Pada gambar 4.11 di halaman seanjutnya, terdapat implementasi daftar alert yang didasarkan pada banyaknya *port* tujuan yang sama. Daftar tersebut direquest berdasarkan jangka waktu tertentu.

Halaman ini mempunyai desain dan fungsi yang sama dengan dua halaman sebelumnya. Halaman-halaman ini dibuat untuk mempermudah proses analisa bagi petugas Administrator jaringan.

Displaying summary report group by port destination:

From 13-05-2005 20 35 To 16-05-2005 20 35 Filter

port destination	Jumlah
20	10
21	2
22	2
23	10
25	20
53	6
59	5
79	5
80	3
110	10

Gambar 4. 13 Implementasi halaman daftar alert *group by port* tujuan

BAB 5

UJI COBA DAN EVALUASI

BAB 5
UJI COBA DAN EVALUASI

Suatu aplikasi baru dikatakan berguna apabila bisa dijalankan sesuai dengan tujuan awal. Pada bab ini akan dijelaskan tahapan-tahapan pengujian yang dilakukan terhadap aplikasi yang penelitian ini.

5.1 Lingkungan Uji Coba Aplikasi Pendeteksian Intrusi

Pada subbab ini akan diterangkan terlebih dahulu kondisi lingkungan (perangkat keras maupun perangkat lunak) yang digunakan untuk melakukan uji coba.

5.1.1 Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan aplikasi pada sesi uji coba ini mempunyai konfigurasi sebagai berikut :

- Processor : AMD Athlon XP 2000+ 2 Ghz
- Memori : 96 Mb PC 2700
- Hardisk : 40 Gb 7200 rpm

5.1.2 Perangkat Lunak

Perangkat lunak yang digunakan adalah :

- Sistem Operasi FreeBSD 5.1.
- Perangkat lunak Tcpdump.
- Perangkat lunak data mining framework weka.

- Compiler gcc 3.2.2.
- Perangkat lunak IDS Snort 2.0.

5.1.2.1 Perangkat Lunak Tcpdump

Perangkat lunak ini Tcpdump adalah suatu *tools* standar untuk menangkap dan menganalisa data jaringan. Aplikasi ini dapat diperoleh di http://www.tcpdump.org. Perangkat ini digunakan dalam salah satu skenario yang akan dijelaskan pada sub-bab selanjutnya.

5.1.2.2 Framework Weka

Framework Weka adalah sebuah kumpulan tools dan *library java* untuk membantu periset dalam bidang data mining. Perangkat lunak ini dikembangkan oleh tim dari Universitas Waikato Selandia Baru, dan dapat di-download di http://www.cs.waikato.ac.nz/ml/weka. Dalam framework ini sudah terdapat kumpulan class-class yang merupakan implementasi dari beberapa metode datamining yang sudah dikenal, misalnya *decision tree* [15]. Selain itu framework ini juga menyediakan fasilitas pengujian dan pembandingan antara dua atau beberapa metode datamining menggunakan input yang sama.

5.1.2.3 Snort

Snort merupakan software IDS *opensource* yang paling aktif selama tahun-tahun terakhir. Snort dikembangkan pertma kali oleh *Marty Roesch*, yang kemudian dilanjutkan oleh tim *sourcefire* dan dapat di-download di http://www.snort.org. Snort dalam pendeteksian intrusi mengunakan beberapa

preprocessor dan juga dengan *detection engine* yang melakukan pencarian string menggunakan sistem seperti halnya *regex*.

5.2 Skenario Uji Coba

Uji coba dijalankan dengan memakai 3 skenario utama :

- Uji coba pendeteksian serangan yang telah diketahui.
- Uji coba pendeteksian serangan baru.
- Uji coba kecepatan.
- Uji coba data online.

5.2.1 Uji Coba Pendeteksian Serangan Yang Telah Diketahui

Uji coba ini dimaksudkan untuk mengukur tingkat akurasi Naïve Bayes sebagai pendeteksi dalam mendeteksi serangan-serangan yang telah diketahui. Uji coba ini bersifat offline dan sebagai input, menggunakan beberapa dataset yang merupakan hasil dari *preprocess* dataset online milik Lincoln Lab. MIT-DARPA 1998 menjadi data yang telah berbentuk *text file*. Pada uji coba ini akan diujicobakan dataset yang berisi beberapa jenis-jenis serangan yaitu, *portsweep*, *ipsweep*, *guess_passwd*, *satan*, serta *nmap*. Selain itu, sebagai perbandingan, akan disertakan juga uji coba pendeteksian dengan metode *decision tree*. Metode *decision tree* adalah sebuah metode yang banyak sekali digunakan pada bidang data mining.

5.2.2 Uji Coba Pendeteksian Serangan Baru

Uji coba pada tahap ini dimaksudkan untuk membuktikan kemampuan metode *Naïve Bayes* sebagai pendeteksi anomali. Seperti halnya sesi uji coba

pertama, uji coba ini juga bersifat offline menggunakan data *text file* hasil dari *preprocess* dataset online milik Lincoln Lab. DARPA-MIT 1998. Pada sesi ini akan diujicobakan beberapa serangan yang mana serangan-serangan tersebut tidak terdapat pada data training. Seperti pada uji coba pertama, pada sesi ini juga disertakan uji coba pendeteksian menggunakan *decision tree* untuk membuktikan bahwa metode *misuse detection* tidak dapat mendeteksi bentuk-bentuk serangan yang baru.

5.2.3 Uji Coba Kecepatan

Adalah hal yang sulit untuk mengetahui kecepatan sebenarnya dari sebuah aplikasi network seperti halnya IDS. Kecepatan disini berarti berapa banyaknya data maksimal yang dapat dianalisa oleh IDS tanpa *dropping packet* Karena sulitnya pengukuran, maka akan dibuat beberapa perbandingan dengan *tcpdump*. Selain itu, sebagai pembanding lainnya akan dipergunakan *Snort* untuk membaca data yang sama. Diharapkan dari dua perbandingan ini dapat diambil kesimpulan yang proporsional mengenai kecepatan dari aplikasi ini.

5.2.4 Uji Coba Online

Pada percobaan menggunakan data online, aplikasi dijalankan pada FreeBSD kemudian dilakukan *probing* menggunakan software *local port sentry* dan *Lan Guard* yang dijalankan pada sebuah *workstation* windows XP untuk membuktikan bahwa aplikasi berjalan sesuai harapan.

5.3 Pelaksanaan Dan Hasil Uji Coba

Dari skenario yang telah disiapkan maka dilakukan ujicoba. Dan pada bagian ini, ujicoba yang dilakukan dan hasilnya akan dijelaskan satu persatu.

5.3.1 Pelaksanaan Dan Hasil Uji Coba Pendeteksian Serangan

Pada tahap ini dilakukan dalam lima kali percobaan. Masing-masing menggunakan satu dataset yang sebagian besar berisi koneksi normal dan sebagian kecil berisi salah satu dari lima serangan. Tabel 5.1 berikut ini merangkum dari dataset yang dipergunakan pada tahap ini.

Tabel 5. 1 Daftar dataset beserta keterangannya

Dataset	Jenis Koneksi	Jumlah Record
Dataset 1	normal	194,556
	portsweep	2,082
Dataset 2	normal	194,556
	ipsweep	2,496
Dataset 3	normal	194,556
	nmap	463
Dataset 4	normal	194,556
	guess_passwd	53
Dataset 5	normal	194,556
	satan	3178

Dari setiap dataset akan diujicobakan dengan dua metode, yaitu metode *Naïve Bayes* dan *Decision Tree* sebagai metode pembanding. Kemudian dari setiap model yang telah dibangun akan dilakukan *stratified ten-folds cross-validation* [15]. Evaluasi menggunakan cara ini adalah sebuah standar yang banyak dipakai dalam mengevaluasi percobaan-percobaan dalam bidang data mining. Dalam *ten-folds cross-validation,* sebuah dataset dibagi menjadi sepuluh bagian sama besar (kurang lebih) yang kemudian dilakukan percobaan sebanyak

sepuluh kali yang masing-masing menggunakan sembilan bagian dan satu bagian sebagai data test. Pembagian menjadi sepuluh bagian dilakukan dengan menjaga perbandingan jumlah klas sesuai dengan dataset aslinya, hal ini disebut secara *stratified*. Adapun hasil dari percobaan-percobaan menggunakan dataset1 s/d dataset 5 ditunjukkan oleh tabel 5.2 s/d tabel 5.6.

Tabel 5. 2 Percobaan kesatu menggunakan dataset 1

Dataset 1		Naïve Bayes		Decision Tree	
Jenis Koneksi	**Jumlah Koneksi**	**Detect**	**Miss**	**Detect**	**Miss**
portsweep	2,082	2,075	7	2,078	4
normal	194,556	194,410	146	194,551	5
Train Time		0.39 detik		4.27 detik	
Test Time		0.92 detik		0.63 detik	

Tabel 5. 3 Percobaan kedua menggunakan dataset 2

Dataset 2		Naïve Bayes		Decision Tree	
Jenis Koneksi	**Jumlah Koneksi**	**Detect**	**Miss**	**Detect**	**Miss**
ipsweep	2,496	859	1,637	858	1,638
normal	194,556	194,416	140	194,418	138
Train Time		1.09 detik		8.45 detik	
Test Time		1.34 detik		1.47 detik	

Tabel 5. 4 Percobaan ketiga menggunakan dataset 3

Dataset 3		Naïve Bayes		Decision Tree	
Jenis Koneksi	**Jumlah Koneksi**	**Detect**	**Miss**	**Detect**	**Miss**
nmap	463	457	6	457	6
normal	194,556	194,504	52	194,555	1
Train Time		0.34 detik		24.44 detik	
Test Time		7.59 detik		0.55 detik	

Tabel 5. 5 Percobaan kelima menggunakan dataset 4

Dataset 4		Naïve Bayes		Decision Tree	
Jenis Koneksi	**Jumlah Koneksi**	**Detect**	**Miss**	**Detect**	**Miss**
guess_passwd	53	51	2	49	4
normal	194,556	194,385	171	194,554	2
Train Time		4.91 detik		28.44 detik	
Test Time		7.55 detik		0.63 detik	

Tabel 5. 6 Percobaan kelima menggunakan dataset 5

Dataset 5		Naïve Bayes		Decision Tree	
Jenis Koneksi	**Jumlah Koneksi**	**Detect**	**Miss**	**Detect**	**Miss**
satan	3,178	3,140	38	3,135	43
normal	194,556	194,418	138	194,556	0
Train Time		4.48 detik		40.66 detik	
Test Time		6.99 detik		1.38 detik	

Dari lima percobaan yang disajikan oleh tabel diatas, dapat dilihat bahwa metode *Naïve Bayes* mempunyai tingkat *false positive* yang lebih tinggi dari pada metode *Decision Tree.* Sedangkan dalam hal tingkat pendeteksian, kedua metode tersebut mempunyai nilai yang hampir sama. Sedangkan waktu yang dibutuhkan untuk *training* model jauh lebih cepat metode naïve bayes, sedangkan untuk waktu *test,* beberapa percobaan menunjukkan *decision tree* mengungguli *naïve bayes*. Hal ini karena pada tahap 3 s.d 5 (tabel 5.4 s/d tabel 5.6) dataset memakai variabel masih dalam bentuk continues. Sedang pada tahap 1 dan 2 (tabel 5.2 dan tabel 5.3) yang variabelnya telah di rubah menjadi variabel diskrit, maka kecepatan tesnya hampir sama.

Untuk memperjelas perbandingan kedua metode tersebut, tabel 5.7 berikut ini memberikan rangkuman perhitungan nilai *detection rate* serta *false positive* dari *Naïve Bayes* (disingkat NB) dan *Decision Tree* (DT). Dalam bidang IDS,

selain tingkat keakurasian pendeteksian, nilai *false positive* juga harus diperhitungkan. *False positive* yang tinggi akan menunjukkan banyaknya koneksi legal yang terdeteksi sebagai suatu serangan. Nilai *false positive* didapatkan dari jumlah koneksi yang terdeteksi sebagai *attack* dibagi jumlah total koneksi normal.

Tabel 5. 7 Rangkuman hasil percobaan misuse

Data	Detection Rate		False Positive	
	NB	DT	NB	DT
1	99.66 %	99.81 %	0.075 %	0.003 %
2	34.42 %	34.36 %	0.072 %	0.071 %
3	98.7 %	98.7 %	0.027 %	0 %
4	96.23 %	92.45 %	0.088 %	0.001 %
5	98.8 %	98.65 %	0.071 %	0 %
Rata-rata	85.56 %	84.79 %	0.067 %	0.015 %

5.3.2 Pelaksanaan Dan Hasil Uji Coba Pendeteksian Serangan Baru

Pada tahap ini dilakukan dalam empat kali percobaan. Masing-masing menggunakan satu dataset sebagai *training* dan satu dataset dipergunakan sebagai test. Dataset yang dipergunakan sebagai test berisi serangan yang belum dikenal pada tahap *training*. Hasil keempat percobaan ditampilkan pada tabel 5.8 s/d 5.11.

Tabel 5. 8 Hasil percobaan kesatu pendeteksian serangan baru

Dataset			Naïve Bayes		Decision Tree	
Training	Test	Jumlah Koneksi	Detect	Miss	Detect	Miss
guess_passwd	portsweep	2,082	2,075	7	1,791	291
	normal	194,556	194,376	180	194,553	3
Train time			0.89 detik		10.95 detik	
Test time			0.84 detik		1 detik	

Tabel 5. 9 Hasil percobaan kedua pendeteksian serangan baru

Dataset			Naïve Bayes		Decision Tree	
Training	Test	Jumlah Koneksi	Detect	Miss	Detect	Miss
guess_passwd	nmap	463	457	6	0	463
	normal	194,556	194,376	180	194,553	3
Train time			0.91 detik		8.97 detik	
Test time			0.83 detik		0.94 detik	

Tabel 5. 10 Hasil percobaan ketiga pendeteksian serangan baru

Dataset			Naïve Bayes		Decision Tree	
Training	Test	Jumlah Koneksi	Detect	Miss	Detect	Miss
port_sweep	guess_passwd	53	51	2	6	47
	normal	194,556	194,413	143	194,553	3
Train time			0.34 detik		3.11 detik	
Test time			0.88 detik		0.55 detik	

Tabel 5. 11 Hasil percobaan ketiga pendeteksian serangan baru

Dataset			Naïve Bayes		Decision Tree	
Training	Test	Jumlah Koneksi	Detect	Miss	Detect	Miss
port_sweep	nmap	463	457	6	0	463
	normal	194,556	194,413	143	194,553	3
Train time			0.34 detik		3.98 detik	
Test Time			0.92 detik		0.98 detik	

Rangkuman dari beberapa buah percobaan diatas disajikan pada tabel 5.12 dibawah ini.

.Tabel 5. 12 Rangkuman tingkat akurasi dari pendeteksian serangan baru

Percobaan	Detection Rate		False Positive	
	NaïveBayes	DecisionTree	NaïveBayes	DecisionTree
1	99.66 %	86.02 %	0.093 %	0.002 %
2	98.7 %	0 %	0.093 %	0.002 %
3	96.23 %	11.32 %	0.074 %	0.002 %
4	98.7 %	0 %	0.074 %	0.002 %
Rata-rata	98.32 %	24.34 %	0.084 %	0.002 %

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

5.3.3 Pelaksanaan Dan Hasil Uji Coba Kecepatan

Dalam uji coba tahap ini, akan dilakukan pengetesan kecepatan aplikasi IDS yang dibuat. Sebuah IDS haruslah cepat dalam menangkap paket serta analisanya. Artian cepat disini adalah bahwa aplikasi harus dapat menangkap semua paket dan menganalisanya pada jaringan yang cukup sibuk, minimal pada sebuah *medium-size* LAN. Karena sulitnya pengetesan secara *realtime* dan terbatasnya *resource* yang ada, maka penulis memutuskan untuk sekali lagi memakai data simulasi milik Lincoln Lab. MIT-DARPA yang masih berbentuk *raw tcpdump data*. Data ini masih berbentuk binary file yang merupakan dump data simulasi jaringan US *Airforce* menggunakan aplikasi Tcpdump. Setiap satu file berisi dump data jaringan selama sehari penuh. File-file yang akan digunakan sebagai simulasi dapat dilihat pada tabel 5.13. Pada jaringan ini terdapat beberapa ratus client dan beberapa *firewall* serta *router*. Diasumsikan bahwa jaringan ini dapat merepresentasikan sebuah jaringan LAN dengan ukuran yang sedang sehingga cukup representatif jika dipergunakan sebagai data simulasi.

Tabel 5. 13 Daftar file dump data jaringan untuk perbandingan kecepatan

Nama File	Besar File
week_one/wednesday/tcpdump	316 MB data network
week_two/monday/tcpdump	152 MB data network
week_two/tuesday/tcpdump	314 MB data network

Selanjutnya dengan input file-file tersebut, akan diperbandingkan kecepatan aplikasi dengan kecepatan Tcpdump itu sendiri serta Snort. Hasil dari perbandingan ditampilkan pada tabel 5.14 pada halaman 83.

Tabel 5. 14 Perbandingan kecepatan dalam membaca file data dump network

Nama File	Naïve bayes	Tcpdump	Snort
week_one/wednesday/tcpdump	00:10:36	00:00:37	00:01:08
week_two/monday/tcpdump	00:08:41	00:00:26	00:00:49
week_two/tuesday/tcpdump	00:11:46	00:00:30	00:01:04

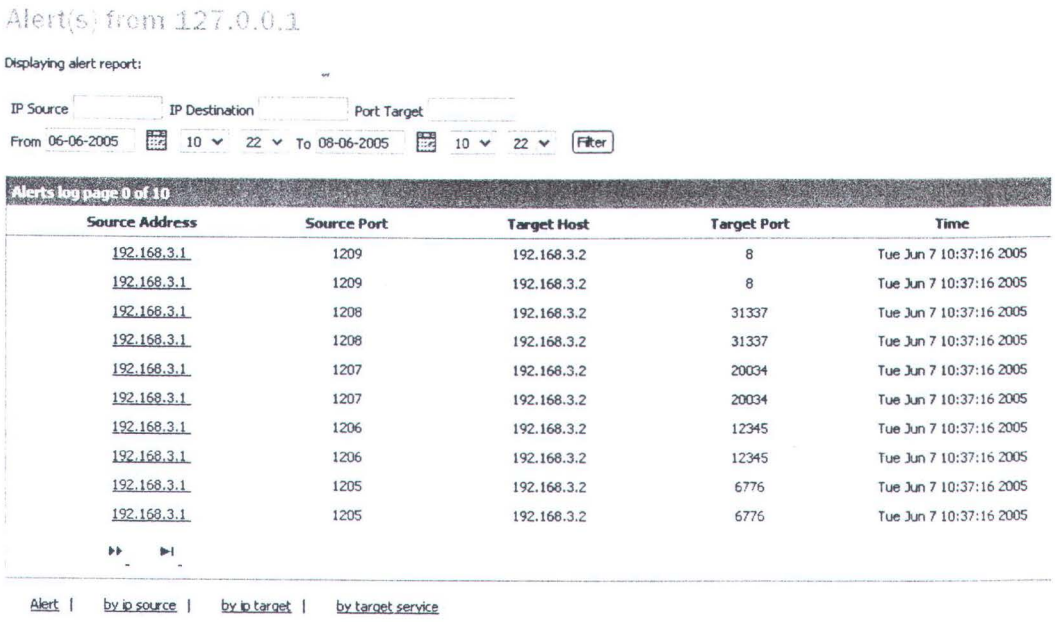
Dari tabel 5.14 diatas, dapat dilihat bahwa aplikasi penelitian ini mempunyai performance yang paling buruk, jauh ditinggalkan oleh aplikasi-aplikasi lainnya. Hal ini dikarenakan adanya proses penulisan ke database PostgreSQL untuk setiap koneksi yang tertangkap oleh aplikasi. Sehingga menambah waktu keseluruhan proses analisa. Selain itu, kedua program pembanding diatas (tcpdump dan snort) dibuat dengan menggunakan bahasa C yang tentu saja lebih cepat dari C++ serta telah mengalami life development cycle selama beberapa tahun (tcpdump lebih dari sepuluh tahun sedangkan snort selama tiga tahun terakhir).

5.4 Hasil Uji Coba Menggunakan Data Online

Pada percobaan menggunakan data online, aplikasi dijalankan pada freeBSD kemudian dilakukan *probing* menggunakan software *local port sentry* dan *Lan Guard* yang dijalankan pada sebuah *workstation* windows XP untuk membuktikan bahwa aplikasi berjalan sesuai harapan.

Probing yang dijalankan yaitu *stealth scan* dan TCP *scan.* Dari hasil *Stealth scan* adalah sejenis *port* scanning dengan hanya melakukan *half open connection*. Sedangkan TCP *scan* adalah scanning TCP. Pada percobaan ini dapat diketahui bahwa aplikasi berjalan sesuai harapan. Aplikasi dapat mendeteksi

semua jenis *scanning/probing* dari kedua software tersebut. Gambar 5.1 dibawah ini menunjukkan halaman web report dari aplikasi.



Gambar 5. 1 Gambar hasil pendeteksian secara realtime dengan report web based

Dari percobaan didapatkan informasi bahwa dari semua operasi *scan* yang dilakukan tidak semuanya terdeteksi sebagai alert. Hanya setengah lebih dari seluruh koneksi *scan* yang dapat dideteksi. Berikut ini hasil dari pendeteksian secara online ditampilkan pada tabel 5.15.

Tabel 5. 15 Hasil uji coba online mnggunakan Nmap

Jenis Serangan	Jumlah Koneksi	Terdeteksi
TCP scan Lan Guard	98	82
Stealth Scan	73	60

Dari percobaan secara online, ternyata hasil yang didapatkan kurang memuaskan, hal ini mungkin dikarenakan process preprocess paket TCP hingga menjadi data yang dapat dilakukan pendeteksian kurang sempurna. Perhitungan data atribut yang berhubungan dengan perhitungan koneksi selama dua detik

terakhir sangatlah *errorprone*. Selain itu pemilihan data raining akan sangat mempengaruhi hasil akhir secara keseluruhan. Pada percobaan online diatas dilatih dengan data *syn attack* dengan *prior* normal 99.967 % dan *prior* attack 0.033 %.

5.5 Evaluasi Akhir

Dari beberapa kali hasil uji coba seperti yang telah didokumentasikan diatas, dapat diketahui bahwa aplikasi pendeteksi dalam penelitian ini dapat melakukan *sniffing* dan analisa header paket TCP dengan kecepatan yang cukup memuaskan.

Adapun fungsi utama dari aplikasi ini, yaitu sebagai aplikasi pendeteksi intrusi, juga dapat dilakukan dengan cukup baik. Meskipun mempunyai nilai *false positive* yang sedikit lebih tinggi dari metode *misuse dtection* dalam perbandingan, metode ini mempunyai keunggulan dalam mendeteksi serangan baru yang belum dikenal sebelumnya.

BAB 6

KESIMPULAN DAN SARAN

BAB 6
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari pelaksanaan uji coba yang telah dilakukan maka dapat diperoleh kesimpulan :

- Aplikasi dapat melakukan *sniffing* paket TCP dan menyimpannya.dalam *session-session* TCP yang berkesesuaian.
- Aplikasi dapat melakukan penangkapan paket (*packet sniffing*) sekaligus melakukan analisa koneksi dengan kecepatan cukup memuaskan.
- Aplikasi dapat menangkap profile normal dari data jaringan melalui atribut-atribut khusus setiap koneksi.
- Aplikasi dapat melakukan pendeteksian serangan yang telah diketahui seperti halnya metode *misuse* ataupun mendeteksi serangan baru yang belum diketahui.
- Hanya jenis-jenis tertentu dari *attack* yang dapat dideteksi oleh aplikasi ini, berkenaan dengan terbatasnya variabel pendeteksi hanya pada header paket TCP. Jenis-jenis yang dapat dideteksi adalah *probe* dan DoS yang mana perilaku koneksinya berbeda dengan perilaku koneksi normal.

6.2 Saran

Dari pengujian aplikasi maka beberapa hal yang dapat dikembangkan dikemudian hari adalah sebagai berikut :

- Melakukan penelitian lebih lanjut untuk menambah variabel-variabel yang berkenaan dengan content dari data paket maupun informasi spesific dari host-host yang ada sehingga jenis serangan yang dapat dikenali menjadi semakin banyak (mis. jenis-jenis *user to root attack*).
- Menambah atribut-atribut yang dapat menangani protocol-protocol selain TCP.
- Menambahkan fungsi *decrypt* untuk dapat membaca paket-paket yang ter-*encrypt*.
- Menambahkan metode *misuse* untuk mengurangi *false positive*.
- Mengoptimalisasi kode program untuk bagian-bagian yang penting agar dapat menambah kecepatan jalannya program. Hal ini dapat dilakukan dengan mengganti bagian-bagian tertentu dengan *pure* C dan mengganti struktur data manajemen koneksi.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[1] Alder, Raven; Babbin, Jacob; Doxtater, Adam; Foster, James C.; Kohlenberg, Toby; Rash, Michael, "*Snort 2.1 Intrusion Detection Second Edition*", Syngress Publishing, May 2004.

[2] Charniak, Eugene, "*Bayesian Networks Without Tears*", http://www.aaai.org, 1996.

[3] Fyodor, Nmap Homepage, http://www.insecure.org, 2005.

[4] Gfi Lan Guard, Gfi Lan Guard Web Site, http://www.gfi.com, 2005.

[5] Kendall, Kristopher, "A Database of Computer Attacks for the Evaluation of Intrusion detection System", Master's Thesis, Department of electrical Engineering and Computer Science, Massachussets Institute of Technology, Juni 1999.

[6] Lee, Wenke; Stolfo, Salvatore J. Stolfo; Mok, Kui W., "A Data Mining Framework for Building Intrusion Detection Models", *In Proceedings of The IEEE Symposium on Security and Privacy*, 1999.

[7] Liberty, Jesse; Horvath, Davith B., "*Teach Yourself C++ for Linux in 21 Days*", Sams Publishing, 46290, April 2000.

[8] PCAP_Programming.html , http://www.tcpdump.org.

[9] RFC:791 "Internet Protocol Darpa Internet Program Protocol Specification", Information Sciences Institute University of Southern California., September 1981.

[10] RFC:793 "Transmission Control Protocol Darpa Internet Program Protocol Specification", Information Sciences Institute University of Southern California., September 1981

[11] Roesch, Marty; Sourcefire Team, Snort Official Site, http://www.snort.org, 2005.

[12] Stevens, W. Richard, "TCP/IP Illustrated, Volume 1",Addison Wesley Longman, Inc., 1994.

[13] Syarif, Iwan, "Aplikasi Data Mining Untuk Pendeteksian Intrusi Pada Sistem Jaringan Dengan algoritma Clustering Nearest Neighbour", Thesis Program Pasca Sarjana, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya, Agustus 2003.

[14] Tanenbaum, Andrew S., "Computer Networks 3e",Prentice Hall, Inc., 1996.

[15] Witten, Ian H., "Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann Publisher, 2000.

18-511/ITS/H/2003

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

PERANCANGAN DAN PEMBUATAN SITUS ECOMMERCE DENGAN MENGGUNAKAN LOTUS DOMINO 5

TUGAS AKHIR

R51f
004.678
As'a
p-1
2002

PERPUSTAKAAN
ITS
Tgl. Terima 16-7-2003
Terima Dari H
No. Agenda Prp. 118249

Oleh :

ABRAHAM AS'AD
NRP : 5196.100.045

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2002

PERANCANGAN DAN PEMBUATAN SITUS ECOMMERCE DENGAN MENGGUNAKAN LOTUS DOMINO 5

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Komputer

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui/Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Ir. Joko Lianto Buliali, MSc	Dwi Sunaryono, S. Kom

SURABAYA

FEBRUARI 2002

ABSTRAK

ABSTRAK

Lotus Notes/Domino adalah suatu produk yang merupakan gabungan dari fungsi Application Development, Document Database, dan Messaging System yang terintegrasi.Pengembangan aplikasi berbasis web dimungkinkan karena elemen perancangan yang dibuat di dengan Domino Designer dapat langsung ditampilkan ke web browser. Dengan kemampuan Domino Server yang mampu menjadi Web Server memungkinkan aplikasi dapat diakses dari luar sistem.

Dalam Tugas Akhir ini dilakukan pembuatan aplikasi toko buku online BookStore.Inc dengan memanfaatkan kelebihan yang dipunyai oleh Lotus Notes/Domino. Aplikasi BookStore.Inc dapat diakses melalui jaringan intranet maupun internet.

Aplikasi yang dikembangkan telah dilakukan uji coba terhadap fungsionalitas dan operasionalitas aplikasi yang ditekankan pada kemampuan aplikasi menangani fungsi-fungsi standar suatu situs eommerce yaitu: browse katalog, pendaftaran pelanggan, proses pembelian dan validasi kartu kredit(simulasi) serta aspek keamanan. Hasil uji coba menunjukkan bahwa fungsionalitas dan operasionalitas aplikasi tersebut telah berjalan dengan baik.

KATA PENGANTAR

KATA PENGANTAR

Bismillahirrohmanirrohiim,

Alhamdulillah, segala puji dan syukur hanya milik Allah SWT, yang memiliki apa yang di langit dan di bumi, hanya kepadaNya-lah segala puji, Dia-lah yang mempunyai segala ilmu dan Maha Penolong dan hanya karena limpahan kasih-sayang, berkah dan taufiqNya-lah penulis dapat menyelesaikan Tugas Akhir ini dengan baik dan lancar.

Shalawat dan salam semoga kepada terlimpah kepada Rasulullah Muhammad SAW, makhluk-Nya yang terbaik, sekaligus penutup para Nabi dan Rasul yang paling mulia.

Tugas akhir ini berjudul "**Perancangan Dan Pembuatan Situs Ecommerce Dengan Menggunakan Lotus Domino 5**", semoga Tugas Akhir ini dapat memberikan manfaat kepada pembaca, khususnya kepada mereka yang tertarik pada pengembangan aplikasi web dengan menggunakan teknologi Lotus Notes/Domino.

Pada kesempatan ini, dengan segala kerendahan hati penulis ingin menyampaikan terima kasih yang sebesar-besarnya atas berbagai dukungannya dalam menyelesaikan Tugas Akhir ini. Terima kasih penulis sampaikan terutama kepada:

1. Keluarga di Lamongan, terutama almarhum aba H. Mashuri dan Ibu Hj. Siti Zulaikhah sebagai orang tua penulis yang telah mencurahkan segala perhatian dan kasih sayangnya yang tak terputus kepada penulis serta selalu mendoakan kebahagiaan dan kesuksesan penulis. Tak ketinggalan juga untuk mbak Indah,

dik Pipit , my lovely sisters, atas dorongan semangat dan doa yang diberikan kepada penulis. Juga kepada de Kasri, de Ilyas, de Tuminah, kak Tarjo, mbak Kus, de Nemu, de Bun, kak Maksum dan sanak-saudara penulis yang lain di Lamongan.

2. Keluarga di Surabaya, mbak Indah dan mas Mufti atas kebaikan dan bantuan serta semangat yang terus diberikan kepada penulis, dik Pipit, dik Afa, Muhib, mbak Cici atas segala kebaikan dan perhatian yang telah mereka berikan kepada penulis.
3. Bapak Dr. Ir. Joko Lianto, M.Sc, selaku pembimbing pertama penulis yang telah memberikan bimbingannya. Terima kasih penulis sampaikan atas diskusi dan wejangan yang diberikan kepada penulis.
4. Bapak Dwi Sunaryono, S.kom, selaku pembimbing kedua penulis, yang telah memberikan ide pembuatan tugas akhir penulis, serta atas bimbingan, dorongan dan semangat, diskusi, kesempatan, nasehat dan kebaikan-kebaikannya selama penulis menyelesaikan Tugas Akhir ini. Hanya Alloh yang akan membalas dengan kebaikan yang lebih besar.
5. Bapak dan Ibu Dosen Teknik Informatika yang lain, untuk segala ilmu yang dilimpahkan serta bimbingannya selama masa perkuliahan.
6. Seluruh Staff Tata Usaha Teknik Informatika (Mas Yudi, Mbak Irna, Mas Sugeng, Pak Mu'in), atas segala pelayanan administrasi kemahasiswaan yang diberikan.
7. Teman-temanku arek-arek Blok U-167, Aji, Andik, Darlis, Henry, Ali, Ocal, Andri, atas tumpangan dan fasilitas yang telah diberikan kepada penulis, juga

atas bantuan moril dan materiil, semangat dan nasehat yang telah diberikan kepada penulis. Hanya Allah yang akan memberikan balasan atas kebaikan antum sekalian.

8. Para labman dan labgirl di laboratorium Sistem Informasi, atas fasilitas yang telah diberikan, Heri Akhmadi (98), atas hiasan bacaan qur'annya, Hera , Daning, trims buat snacknya, Heri 95, Julius, terima kasih pinjaman printernya, Arif Basofi yang telah memberikan bahan-bahan dan dasar teori serta sebagai teman berdiskusi tentang Lotus Notes, juga atas segala bantuan dan dukungan yang telah mereka berikan kepada penulis.
9. Teman-teman C0C seperjuangan Tugas Akhir, Arif Basofi, Julius, Henry, Nova, Thatit, Danar, Agung, Anank, Ferdi, Martanti, Mardiana, Wayan, Wisdarmanto, Partono, Yuliana dan teman-teman yang lain, atas kerjasama dan kebersamaan diwaktu susah dan senang saat mengerjakan TA
10. Seluruh angkatan C-0C, Puji, Cahyono, Dikhi, Aan, Azzah, Candra, Monica, Dhias, Binar dan teman-teman yang lain yang tidak bisa penulis sebutkan satu persatu di sini, atas kebersamaan yang indah yang telah dan Insya Allah akan terus terjalin serta kerjasama selama masa perkuliahan. Vivat C-0C.
11. Buat kakak-kakak angkatan penulis serta adik angkatan penulis baik yang penulis kenal maupun belum yang telah mewarnai kampus kita ini dengan kegiatan yang bermanfaat, Insya Allah.

Tidak ada kesempurnaan dalam sebuah karya manusia. Penulis menyadari masih banyak kekurangan dalam penyusunan Tugas Akhir ini. Untuk itu pula jika

MILIK PERP... INSTITUT TEKNOLOGI SEPULUH NOPEMBER

ada kritik dan saran dari pembaca, Insya Allah akan diterima penulis dengan lapang dada.

Akhir kata penulis berdo'a semoda Allah SWT selalu membuka hati kita dengan cahayaNya dan mengajarkan ilmuNya kepada kita dan semoga Allah SWT menghindarkan kita dari ilmu yang tidak bermanfaat.

Surabaya, Januari 2002

Penulis

DAFTAR ISI

DAFTAR ISI

ABSTRAK I

KATA PENGANTAR II

DAFTAR ISI VI

DAFTAR GAMBAR IX

DAFTAR TABEL XI

BAB I 1

PENDAHULUAN 1

1.1 PENDAHULUAN 1
1.2 PERUMUSAN MASALAH 2
1.3 BATASAN PERMASALAHAN 3
1.4 TUJUAN DAN MANFAAT 3
1.5 METODOLOGI TUGAS AKHIR 4
1.6 SISTEMATIKA PEMBAHASAN 5

BAB II 7

TEORI PENUNJANG 7

2.1 INTERNET 7
2.1.1 E-Business & E-commerce *8*
2.2 LOTUS NOTES / DOMINO 10
2.2.1 Komponen Notes dan Domino *11*
2.2.1.1 Program - Program Client dan Server 12
- Domino Mail Server 15
- Domino Application Server 16
- Domino Enterprise Server 16

2.2.1.2 Notes Object Services (NOS) 16
2.2.1.3 Database dan File 17
2.2.2 Notes *17*
2.2.2.1 Item-Item Notes 19
2.2.2.2 Tipe-Tipe Notes 19
2.2.2.3 Data Notes 20
2.2.2.4 Administrasi Notes 22
2.2.2.5 Elemen Perancangan Notes 22
2.2.2.6 Hierarki Notes 24
2.2.3 Bahasa Pemrograman *24*
- Agent dan Action 25

2.3 NOTES DAN WEB 26
2.3.1 Domino Web Server *27*
2.3.2 Arsitektur Domino Web Server *31*
Komponen-Komponen HTTP Server Task 34
2.4 KEAMANAN LOTUS NOTES / DOMINO 36
2.5 UNIFIED MODELING LANGUAGE (UML) 40
2.5.1 Diagram UML [CRA - 98] *41*

BAB III 42

PERANCANGAN SISTEM APLIKASI 42

3.1 USE CASE VIEW 43
2.2.2 Deskripsi Aktor *43*
2.2.3 Diagram Use Case *44*
3.1.2.1 Diagram Use Case Manajemen Data Buku 46
3.1.2.2 Diagram Use Case Manajemen Pelanggan 47
3.1.2.3 Diagram Use Case Pendaftaran Pelanggan 48
3.1.2.4 Diagram Use Case Browse Katalog 49
3.1.2.5 Diagram Use Case Add to Cart 49
3.1.2.6 Diagram Use Case Pembelian Buku 51
3.1.3 MODEL OBJEK STATIS *53*
3.1.3.1 Diagram Class Manajemen Data Buku 54
3.1.3.2 Diagram Class Manajemen Pelanggan 55
3.1.3.3 Diagram Class Pendaftaran Pelanggan 56
3.1.3.4 Diagram Class Browse Katalog 57
3.1.3.5 Diagram Class Add to Cart 58
3.1.3.6 Diagram Class Pembelian Barang 59
3.1.4 MODEL OBJEK DINAMIS *60*
3.1.4.1 Diagram Sequence Login 61
3.1.4.2 Diagram Sequence Manajemen Data Buku 62
3.1.4.3 Diagram Sequence Manajemen Pelanggan 63
3.1.4.4 Diagram Sequence Pendaftaran Pelanggan 64
3.1.4.5 Diagram Sequence Browse Katalog 65
3.1.4.6 Diagram Sequence Add to Cart 66
3.1.4.7 Diagram Sequence Pembelian Barang 69
- Diagram Sequence Sub Proses Cek Out 69
- Diagram Sequence Sub Proses Validasi Pembayaran 70

BAB IV 72

PEMBUATAN APLIKASI 72

4.1 SETTING ACL (ACCES CONTROL LIST) 72
- Role 72
- User 72

4.2 FORM 73
4.2.1 Form DisplayKatalog *73*
4.2.2 Form Registrasi *75*
4.2.3 Form EntriKartuKredit *77*
4.2.4 Form LupaPassword *78*
4.2.5 Form KeranjangBelanja *78*
4.2.7 Form HalamanUtama *79*
4.2.8 Form Konfirmasi *80*
4.2.9 Form Order *81*
4.2.10 Form OrderStatusConfirm *82*
4.2.11 Form OrderThanks *83*
4.2.12 Form CaraPembayaran *83*
4.2.13 Form CaraPemesanan *84*
4.2.14 Form CaraPengiriman *85*
4.2.15 Form RegChoice *85*
4.2.16 Form StatusPesanan *86*
4.2.17 Form ViewKatalogperKategori *87*
4.2.18 Form SSSearchTemplateDefault *88*
4.3 VIEW 89
- View Admin/SemuaDokumen 89
- View KatalogBuku 89

- View BukuTerbaru 90
- View KatalogHasilPencarian 90
- View KatalogperKategori 90
- View KatalogBuku 90
- View (CartDetailByID) 90

4.4 AGENT 90
4.4.1 CekStokBuku *90*
4.4.2 AddtoCart *92*
4.4.3 UpdateCart *93*
4.4.4 ClearCart *95*
4.4.5 ProsesKartuKredit *96*
4.4.6 DaftarAnggota *98*
4.5 SCRIPT LIBRARIES 101

BAB V 102

UJI COBA PERANGKAT LUNAK 102

5.1 LINGKUNGAN UJI COBA 102
5.2 UJI COBA APLIKASI 103
5.2.1 Proses Browsing Katalog dan Pencarian Buku *104*
5.2.2 Proses Pendaftaran Pelanggan *107*
5.2.2 Proses Pembelian Buku *110*
5.2.2 Proses Manajemen Dokumen *117*
5.2.2.1 Manajemen Buku 119
5.2.2.2 Manajemen Kartu Kredit 120
5.2.2.3 Manajemen Order 121
5.2.2.4 Manajemen Pelanggan 122

BAB VI 125

PENUTUP 125

6.1 KESIMPULAN 125
6.2 KEMUNGKINAN PENGEMBANGAN 126

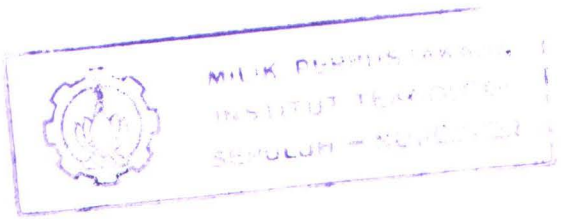
DAFTAR PUSTAKA 128

DAFTAR GAMBAR

DAFTAR GAMBAR

GAMBAR 2.1. KOMPONEN-KOMPONEN HARDWARE/SOFTWARE NOTES DAN DOMINO 11
GAMBAR 2.2. KOMPONEN-KOMPONEN PROGRAM CLIENT 15
GAMBAR 2.3. STRUKTUR LOGIKAL NOTES 18
GAMBAR 2.4. ARSITEKTUR DOMINO WEB SERVER 31
GAMBAR 2.5. TINGKATAN/LEVEL KEAMANAN NOTES/DOMINO 37
GAMBAR 3.1. AKTOR SISTEM TOKO BUKU ONLINE BOOKSTORE.INC 43
GAMBAR 3.2. DIAGRAM USE CASE UTAMA SISTEM BOOKSTORE.INC 45
GAMBAR 3.3. DIAGRAM USE CASE MANAJEMEN BUKU 47
GAMBAR 3.4. DIAGRAM USE CASE MANAJEMEN PELANGGAN 47
GAMBAR 3.5. DIAGRAM USE CASE PENDAFTARAN PELANGGAN 48
GAMBAR 3.6. ACTIVITY DIAGRAM PENDAFTARAN PELANGGAN 49
GAMBAR 3.7. DIAGRAM USE CASE BROWSE KATALOG 49
GAMBAR 3.8. DIAGRAM USE CASE ADD TO CART 50
GAMBAR 3.9. DIAGRAM ACTIVITY ADD TO CART 51
GAMBAR 3.10. DIAGRAM USE CASE PEMBELIAN BARANG 52
GAMBAR 3.11. DIAGRAM ACTIVITY PEMBELIAN BARANG 53
GAMBAR 3.12. DIAGRAM CLASS PADA USE CASE MANAJEMEN DATA BUKU 55
GAMBAR 3.13. DIAGRAM CLASS PADA USE CASE MANAJEMEN PELANGGAN 56
GAMBAR 3.14. DIAGRAM CLASS PADA USE CASE PENDAFTARAN PELANGGAN 57
GAMBAR 3.15. DIAGRAM CLASS PADA USE CASE BROWSEKATALOG 58
GAMBAR 3.16. DIAGRAM CLASS ADD TO CART 59
GAMBAR 3.17. DIAGRAM CLASS PADA USE CASE PEMBELIAN BARANG 60
GAMBAR 3.18.. DIAGRAM SEQUENCELOGIN NOTES 61
GAMBAR 3.19. DIAGRAM SEQUENCELOGIN WEB 62
GAMBAR 3.20. DIAGRAM SEQUENCE MANAJEMEN DATA BUKU 63
GAMBAR 3.21. DIAGRAM SEQUENCE MANAJEMEN PELANGGAN 64
GAMBAR 3.22. DIAGRAM SEQUENCE PENDAFTARAN PELANGGAN 65
GAMBAR 3.23. DIAGRAM SEQUENCE BROWSE KATALOG 66
GAMBAR 3.24. DIAGRAM SEQUENCE ADD TO CART SUB PROSES CEK SESSION 67
GAMBAR 3.25. DIAGRAM SEQUENCE ADD TO CART SUB PROSES TAMBAH ITEM 67
GAMBAR 3.26. DIAGRAM SEQUENCE ADD TO CART SUB PROSES EDIT ITEM KERANJANG 68
GAMBAR 3.27. DIAGRAM SEQUENCE ADD TO CART SUB PROSES HAPUS ITEM 68
GAMBAR 3.28. DIAGRAM SEQUENCE PEMBELIAN BUKU SUB PROSES CEK OUT 70
GAMBAR 3.29. DIAGRAM SEQUENCE PEMBELIAN BUKU SUB PROSES VALIDASI PEMBAYARAN DENGAN KARTU KREDIT 71
GAMBAR 3.30. DIAGRAM SEQUENCE PEMBELIAN BUKU SUB PROSES VALIDASI PEMBAYARAN DENGAN TRANSFER BANK 71
GAMBAR 4.1. FORM DISPLAYKATALOG 74
GAMBAR 4.1. FORM REGISTRASI 76
GAMBAR 4.3. FORM ENTRIKARTUKREDIT 77
GAMBAR 4.4. FORM LUPAPASSWORD 78
GAMBAR 4.5. FORM KERANJANG BELANJA 79
GAMBAR 4.6. FORM HALAMANUTAMA 80
GAMBAR 4.7. FORM KONFIRMASI 81
GAMBAR 4.8. FORM ORDER 81
GAMBAR 4.9. FORM ORDERSTATUSCONFIRM 82
GAMBAR 4.10. FORM ORDERTHANKS 83
GAMBAR 4.11. FORM CARAPEMBAYARAN 84
GAMBAR 4.12. FORM CARAPEMESANAN 84
GAMBAR 4.13. FORM CARAPENGIRIMAN 85
GAMBAR 4.14 FORM REGCHOICE 86
GAMBAR 4.15 FORM STATUSPESANAN 87

GAMBAR 4.16 FORM VIEWKATALOGPERKATEGORI 88
GAMBAR 4.17 FORM $$SEARCHTEMPLATEDEFAULT 89
GAMBAR 5.1. HALAMAN UTAMA 105
GAMBAR 5.2. TAMPILAN HASIL PENCARIAN JUDUL "XML" 106
GAMBAR 5.3 TAMPILAN KATALOG KATEGORI KOMPUTER 106
GAMBAR 5.4. HALAMAN KATALOG BUKU 107
GAMBAR 5.5.. TAMPILAN UTAMA MENU PENDAFTARAN ANGGOTA 108
GAMBAR 5.6. HALAMAN PENDAFTARAN PELANGGAN 108
GAMBAR 5.7. TAMPILAN HALAMAN KONFIRMASI PENDAFTARAN BERHASIL 110
GAMBAR 5.8. HALAMAN DESKRIPSI BUKU "XML FOR ASP.NET DEVELOPERS" 111
GAMBAR 5.9. HALAMAN KERANJANG BELANJA 111
GAMBAR 5.10 FORM LOGIN 112
GAMBAR 5.11. HALAMAN ORDER 113
GAMBAR 5.12. HALAMAN KONFIRMASI PEMBELIAN BERHASIL 114
GAMBAR 5.13. HALAMAN DAFTAR ORDER PENGGUNA 116
GAMBAR 5.14. HALAMAN STATUS ORDER 117
GAMBAR 5.15. MENU MANAJEMEN DOKUMEN PADA NAVIGATOR KIRI 118
GAMBAR 5.16. HALAMAN ADMINISTRATOR 118
GAMBAR 5.17. HALAMAN MANAJEMEN BUKU 119
GAMBAR 5.18. HALAMAN EDIT BUKU 119
GAMBAR 5.19. HALAMAN DESKRIPSI BUKU MASTERING LINUX,2E 120
GAMBAR 5.20. HALAMAN UTAMA MANAJEMEN KARTU KREDIT 120
GAMBAR 5.21. HALAMAN EDIT KARTU KREDIT 121
GAMBAR 5.22. HALAMAN UTAMA MANAJEMEN ORDER 121
GAMBAR 5.23. HALAMAN EDIT ORDER 122
GAMBAR 5.24. HALAMAN UTAMA MANAJEMEN PELANGGAN 122
GAMBAR 5.25. HALAMAN EDIT PELANGGAN 123



DAFTAR TABEL

DAFTAR TABEL

TABEL 2.1. TABEL 12 CLASS NOTES DASAR 19
TABEL 2.2. TABEL ACCESS CONTROL LIST (ACL) DOMINO 40
TABEL 3.1. DAFTAR CLASS PADA USE CASE MANAJEMENDATA BUKU 54
TABEL 3.2. DAFTAR CLASS PADA USE CASE MANAJEMEN PELANGGAN 55
TABEL 3.3. DAFTAR CLASS PADA USE CASE PENDAFTARAN PELANGGAN 56
TABEL 3.4. DAFTAR CLASS PADA USE CASE BROWSE KATALOG 57
TABEL 3.5. DAFTAR CLASS PADA USE CASE ADD TO CART 58
TABEL 3.6. DAFTAR CLASS PADA USE CASE PEMBELIAN BARANG 59

BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

Pada bab ini dijelaskan beberapa hal dasar yang meliputi latar belakang, perumusan masalah, tujuan dan manfaat, batasan permasalahan, metodologi serta sistematika penyusunan dan pembahasan dalam buku tugas akhir ini. Dari uraian tersebut diharapkan gambaran umum permasalahan dan pemecahan karya tugas akhir ini dapat dipahami.

1.1 Pendahuluan

Internet merupakan suatu jaringan komputer global. Internet tidak mengenal batas negara, dan juga tidak ada keterbatasan ruang dan waktu. Dengan demikian Internet merupakan suatu media yang sangat efektif dan juga relatif murah untuk mengiklankan suatu produk .

Dengan semakin banyaknya pengguna internet, maka semakin terbuka pula kesempatan untuk menjual produk melalui internet. Saat ini hampir semua perusahaan menyadari besarnya keuntungan yang bisa didapat melalui internet. Kesempatan untuk menjangkau pasar global dengan biaya yang relatif sedikit membuat setiap perusahaan berlomba untuk membuka situs perusahaan mereka di internet. Hampir setiap situs perusahaan yang menjual barang atau jasa,menyediakan fasilitas untuk pembelian barang lewat internet di situs mereka. Untuk menampilkan data perusahaan dan produk yang dijual diperlukan adanya suatu situs web. Untuk membuat situs web, ada banyak cara dan berbagai macam teknologi yang dapat dipakai . Tugas Akhir ini mencoba untuk membuat sebuah

situs web yang dapat menampilkan berbagai macam informasi baik tentang perusahaan maupun produk-produk yang ditawarkan oleh perusahaan, disertai dengan fasilitas untuk melakukan pembelian produk secara online dengan menggunakan teknologi yang sudah terbukti kehandalan dan keamanannya yaitu Lotus Notes/Domino. versi /release 5.

Lotus Domino Designer R5 adalah suatu lingkungan pengembangan aplikasi web terpadu *(integrated development environment)* yang handal dan mempunyai tampilan yang task-oriented dan kumpulan tool-tool pengembangan *(development tools)* yang terpadu sehingga mempermudah dan mempercepat web developer dalam membuat aplikasi internet. Sebagai contoh , adanya elemen perancangan Outline dan Frameset akan mempermudah dalam pembuatan navigasi/penjelajahan halaman-halaman web dalam suatu situs. Kelebihan lainnya adalah , antara lain kemudahan akses ke database, pembuatan page Html yang WYSIWYG seperti halnya Microsoft Frontpage dan software web development lainnya, support terhadap bahasa – bahasa pemrograman web yang sudah ada seperti Java, Java script, Xml, Html dan juga mendukung CORBA/IIIOP sehingga dapat menulis aplikasi Java client dan Java applet untuk browser yang dapat mengakses data yang ada di Domino server, serta tingkat keamanan berlapis yang diterapkan di Domino.

1.2 Perumusan Masalah

Masalah-masalah yang timbul dan akan dikembangkan solusinya dalam pembuatan Tugas Akhir ini adalah :

1. Membuat sebuah situs web yang dapat menyediakan informasi perusahaan, produk yang ditawarkan, dan menyediakan fasilitas pembelian barang secara online.
2. Membuat suatu situs ecommerce yang mempunyai sistem navigasi atau penjelajahan situs yang efisien dan mudah dimengerti.
3. Membuat situs ecommerce yang terjamin keamanannya.

1.3 Batasan Permasalahan

Dalam pembuatan situs ecommerce ini situs yang dibuat adalah situs yang menawarkan dan menjual barang kepada pembeli secara online, sedangkan pelelangan di internet (e-bid) tidak termasuk dalam bahasan tugas akhir ini. Diamping itu terdapat hal-hal yang mungkin akan menjadi suatu kendala , diantaranya dalam sistem pembayaran , apakah menggunakan alat transaksi seperti kartu kredit yang lazim digunakan di Internet ataukah menggunakan metode transfer ke rekening di bank, karena di Indonesia penggunaan kartu kredit masih belum memasyarakat dan sedikitnya bank yang mampu menjadi penjamin.

1.4 Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah terwujudnya sebuah situs ecommerce yang mampu menampilkan informasi tentang produk yang akan dijual dengan sistem penjelajahan situs yang baik dan mudah dimengerti serta mampu menangani proses pembelian secara online.

Manfaat yang didapatkan dari pembuatan tugas akhir ini adalah dapat mengoptimalkan penggunaan Lotus Notes/Domino yang sebelumnya kebanyakan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH – NOPEMBER

hanya digunakan untuk proses workflow dalam suatu perusahaan untuk pembuatan sebuah situs yang menarik.

1.5 Metodologi Tugas Akhir

Dalam pembuatan Tugas Akhir ini digunakan metodologi sebagai berikut:

1 Studi Literatur

Tahap ini merupakan tahap awal yang harus ditempuh penulis, dimana didalam tahap ini selain menentukan topik dan judul untuk Tugas Akhir, juga untuk mencari referensi , situs–situs, dan data yang berhubungan.

2 Perancangan Sistem

Tahap ini merupakan tahap dimana rancangan dari web page dibuat. Rancangan diusahakan sedetail mungkin sehingga memudahkan dalam pembuatannya.

3 Pembuatan Aplikasi

Pada tahap ini web page dibuat dengan menggunakan software Lotus Notes/Domino R5 .

4 Uji Coba dan Evaluasi

Tahap ini dilakukan untuk melakukan test apakah homepage yang diinginkan sudah tercapai dan untuk mencari kesalahan-kesalahan yang mungkin terdapat selama pembuatan homepage.

5 Penyusunan Buku TA

Merupakan langkah terakhir, dimana proses pembuatan aplikasi dari awal sampai akhir akan didokumentasikan. Dokumentasi ini nantinya akan digunakan untuk mengembangkan aplikasi ini lebih lanjut.

1.6 Sistematika Pembahasan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis yaitu:

- BAB I : PENDAHULUAN, berisi penjelasan tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, metodologi pelaksanaan tugas akhir dan sistematika pembahasan.

- BAB II : TEORI PENUNJANG, akan dibahas dasar ilmu yang mendukung dalam pembahasan tugas akhir ini, mulai pengenalan Internet, Ecommerce, teknologi Lotus Notes / Domino beserta komponennya, serta pemodelan obyek dengan UML.

- BAB III : PERANCANGAN SISTEM APLIKASI, bab ini akan membahas perancangan dari aplikasi toko buku online BookStore.Inc dengan menggunakan notasi UML.

- BAB IV : PEMBUATAN APLIKASI, bab ini berisi pembuatan aplikasi yang berupa aplikasi web dan Notes Client dengan menggunakan Lotus Notes / Domino berdasarkan perancangan yang telah dibuat.

- BAB V : UJI COBA PERANGKAT LUNAK, bab ini membahas dilakukannya uji coba pada lingkungan uji coba yang ada serta dengan skenario yang jelas sebagai bentuk implementasi aplikasi ini.

- BAB VI : PENUTUP, berisi kesimpulan yang didapat dari pembuatan Tugas Akhir ini serta dilengkapi dengan saran yang memberikan adanya

kemungkinan pengembangan agar nantinya Tugas Akhir ini dapat lebih baik.

BAB II
TEORI PENUNJANG

BAB II

TEORI PENUNJANG

Pada bab ini akan dibahas tentang Internet, Ecommerce serta teknologi yang akan digunakan untuk membangun aplikasi Ecommerce untuk tugas akhir ini yaitu Lotus Notes dan Domino Release 5.

2.1 Internet

Internet adalah jaringan komputer yang terbesar di dunia. Jaringan ini menghubungkan jutaan komputer dimana terjadi pertukaran informasi antar pengguna. Internet menurut rancangannya adalah terdesentralisasi, dan tidak ada seseorang atau sebuah badan yang "memiliki" Internet. Walaupun demikian, banyak perusahaan dan institusi yang memiliki dan mengatur jaringan mereka sendiri yang merupakan bagian dari Internet. Tiap komputer di Internet , atau *host,* adalah independen. Pada tahun1994, diperkirakan terdapat 30 juta pengguna Internet. Pada awal tahun 2000, jumlah pengguna diperkirakan hampir sepuluh kali lipat dari tahun 1994, yang berarti hampir 300 juta pengguna di seluruh dunia. Jumlah pengguna Internet dikatakan bertambah dua kali lipat tiap 10 bulan. Pertumbuhan lalu lintas di Internet bertambah dua kali lipat tiap tiga bulan.Diperkirakan akan ada 128 juta pembeli lewat *Web* pada tahun 2002. Pendapatan dari *E-commerce* adalah $ 43 milyar pada tahun 1998. Diperkirakan bahwa pendapatan dari *E-commerce* akan mencapai $ 1,4 triliun pada tahun 2003.

2.1.1 E-Business & E-commerce

E-business adalah transformasi dari proses bisnis kunci melalui penggunaan teknologi. Internet. *Web* telah mengubah setiap aspek dari kehidupan manusia, demikian juga dengan pengoperasian bisnis. Dengan mengadaptasikan teknologi Internet kedalam *core* bisnisnya, maka berkembanglah apa yang disebut dengan *e-business* , dimana sekarang perusahaan besar dan kecil menggunakan *web* untuk berkomunikasi dengan partner mereka, mengakses sistem data di *backend server*, dan melakukan transaksi perdagangan.

Untuk melakukan *e-business* seperti tersebut diatas maka dibuatlah situs-situs web yang digunakan untuk bertransaksi, yang dinamakan situs transaksi (*transactional site*). Situs transaksi sendiri dapat digolongkan menjadi tiga kategori :

- ***Business-to-Business (B2B)*** : Situs transaksi tipe ini menyediakan akses kepada anggota rantai supply eksternal *(external supply chain)* yaitu partner dan *vendor* untuk menambah, menghapus, dan mengedit basisdata perusahaan. Beberapa contoh diantaranya pemesanan produk, penyimpanan ramalan penjualan, dan pengaksesan dan pengeditan data. Aplikasi pada sistem *host* memproses transaksi, mengupdate data dan melakukan hal-hal lain yang sesuai dengan perancangan dari aplikasi. Situs kategori ini biasanya mempunyai beberapa beberapa bentuk autentikasi, misalnya : *login* dan *ID* , permintaan informasi yang berbasis form atau fungsi *query* yang telah

ditentukan, dan fungsi pemasukan data yang berbasis form untuk detil transaksi.

- ***Business-to-Consumer (B2C)*** : Menyediakan akses kepada konsumen untuk menambah, menghapus, dan mengedit suatu data di basisdata peusahaan. Situs ini berbeda dengan situs dinamik biasa dimana di situs ini disediakan fungsi penambahan, penghapusan, dan pengeditan data. Pelanggan dapat memperbaharui informasi pribadi mereka, ditambahkan ke basisdata, mengisi form informasi, form aplikasi, dan lain-lain. Informasi ini kemudian diproses oleh aplikasi yang bersangkutan di *host*. Seperti juga pada situs *business-to-business* , situs ini juga mempunyai beberapa bentuk autentikasi, misalnya : *login* dan *ID* , permintaan informasi yang berbasis form atau fungsi *query* yang telah ditentukan, dan fungsi pemasukan data yang berbasis form untuk detil transaksi.
- ***E-Commerce (e-store)*** : Pengimplementasian e-commerce melibatkan transaksi perdagangan dan pembelian produk atau layanan. Situs business-to-business dan business-to-customer menyediakan data dan informasi dan menginjinkan penambahan, penghapusan, dan pengubahan data, tetapi tidak ada fungsi belanja. Situs E-commerce menyediakan keranjang belanja, katalog produk, fungsi pengiriman barang, pelacakan pesanan, pemrosesan pesanan dan pembayaran .

  Fungsi –fungsi yang terdapat pada situs E-commerce diatas dapat dibuat dengan menggunakan Lotus Domino.

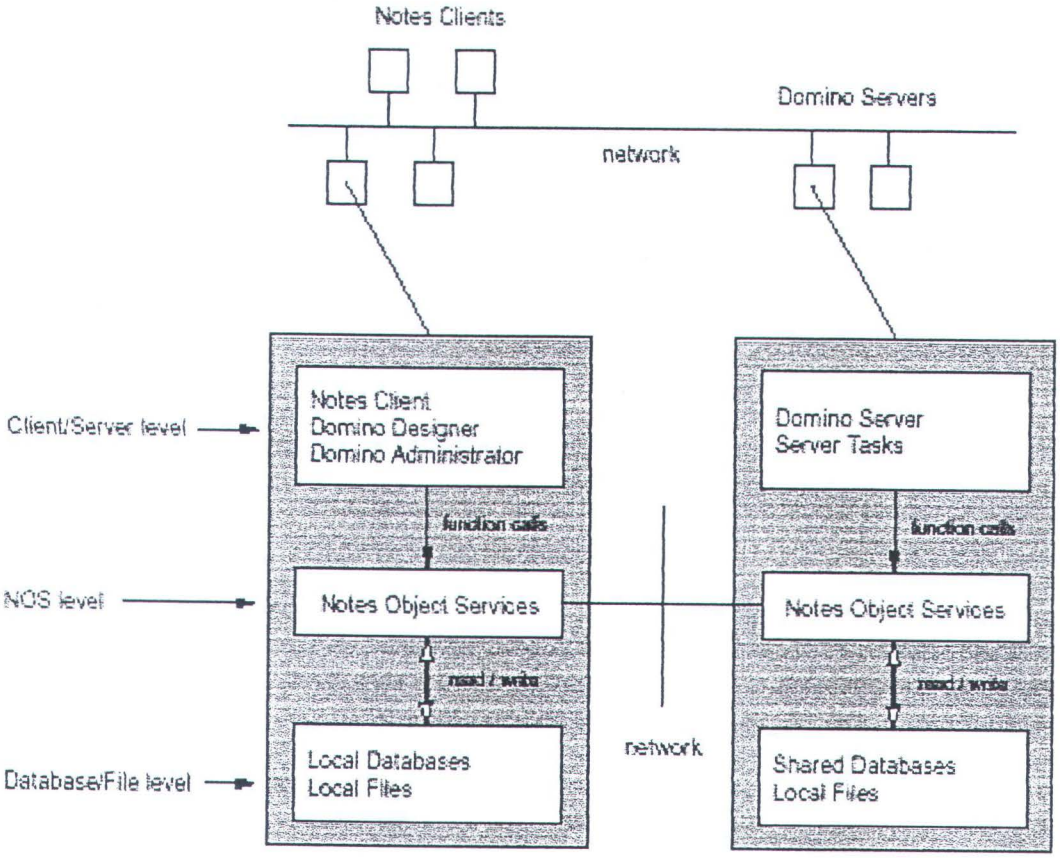
2.2 Lotus Notes / Domino

Lotus Notes merupakan *document database,* yaitu database yang lebih mengutamakan dalam mengorganisir berbagai dokumen, serta agar dapat terakses pada suatu grup atau perkumpulan orang. Database Notes menyimpan informasi dalam sebuah database, yang berisi obyek-obyek yang diketahui sebagai kumpulan dokumen (data Notes) serta elemen-elemen perancangan. Sebuah dokumen merupakan sebuah obyek yang dapat berisi text, grafik, video, audio maupun bentuk data *rich text* lain. Sebagai aplikasi yang memiliki sifat *multiplatform,* dimana dapat digunakan pada berbagai macam jenis mesin komputer dan sistem operasi, Lotus Notes telah melakukan berbagai pengembangan dengan memberikan perbaikan dan penambahan fasilitas dan fitur-fitur baru yang ditujukan untuk mempermudah pemakaian dan menyesuaikan dengan teknologi informasi saat ini, diantaranya dapat mendukung Web Client dan Notes Client yang merupakan *independent-server.*

Pada saat fasilitas HTTP ditambahkan ke dalam sistem, Notes dan Domino merupakan *true application server* untuk internet yang pertama kali. Web server yang dibangun dengan Domino server ini, memiliki keistimewaan dapat sekaligus menjadi *firewall* bagi keluar masuknya data dari dan ke perusahaan yang bersangkutan, yang menggunakan aplikasi internet. Fasilitas internet juga dikembangkan dengan menambahkan beberapa protokol internet diantaranya HTTP, POP3, IMAP, NNTP. Lotus Notes sebenarnya kombinasi akan tiga hal, yaitu *Application Development, Document Database* dan *Messaging System* yang terintegrasi.

2.2.1 Komponen Notes dan Domino

Notes dan Domino terbagi atas dua komponen, yaitu komponen perangkat keras Notes dan Domino *(Notes and Domino Hardware)* dan komponen perangkat lunak Notes dan Domino *(Notes and Domino Software)*. Komponen *hardware* terdiri atas komputer *Notes Client*, komputer *Domino Server* serta *Network* yang saling terhubung diantaranya. Sedangkan komponen *software* terletak pada komputer *client* dan *server*. Gambaran arsitektur komponen-komponen *hardware* dan *software* Notes dan Domino, seperti terlihat dibawah ini.



Gambar 2.1. Komponen-komponen hardware/software Notes dan Domino

MILIK PERPUS
INSTITUT
SEPULUH

Persamaan arsitektur *software* tiga-level diatas adalah telah digunakannya pada kedua komputer baik *client* maupun *server*.

Masing-masing komponen *software* diatas termasuk dalam salah satu level berikut :

1. Program-program *client* dan *server*
2. Notes Object Services (NOS)
3. Kumpulan database dan file.

2.2.1.1 Program - Program Client dan Server

Program-program client dan server telah mengunakan NOS *(Notes Object Service)* untuk membuat *(create)*, modifikasi *(modify)*, membaca *(read)* serta mengelola *(maintain)* berbagai database dan file.

A. Program – Program Client

Terdapat tiga jenis program-program *client*, yang masing-masing memiliki fungsi yang berbeda-beda, dan memberikan akses secara interaktif dengan berbasiskan *window* pada berbagai database dan file lokal ke komputer *client*, serta dalam *sharing* database, yaitu :

- **Notes Client**, diperuntukkan bagi *user-user* Notes, menyediakan akses secara interaktif pada data user. Notes client ini mencakup tatap muka *(interface)* untuk Domino server dan *Web Browser*, oleh karenanya *client* dapat digunakan untuk menghubungkan ke Domino server ataupun *Web Browser*. Pengguna dapat berinteraksi dengan database Notes seperti membaca maupun

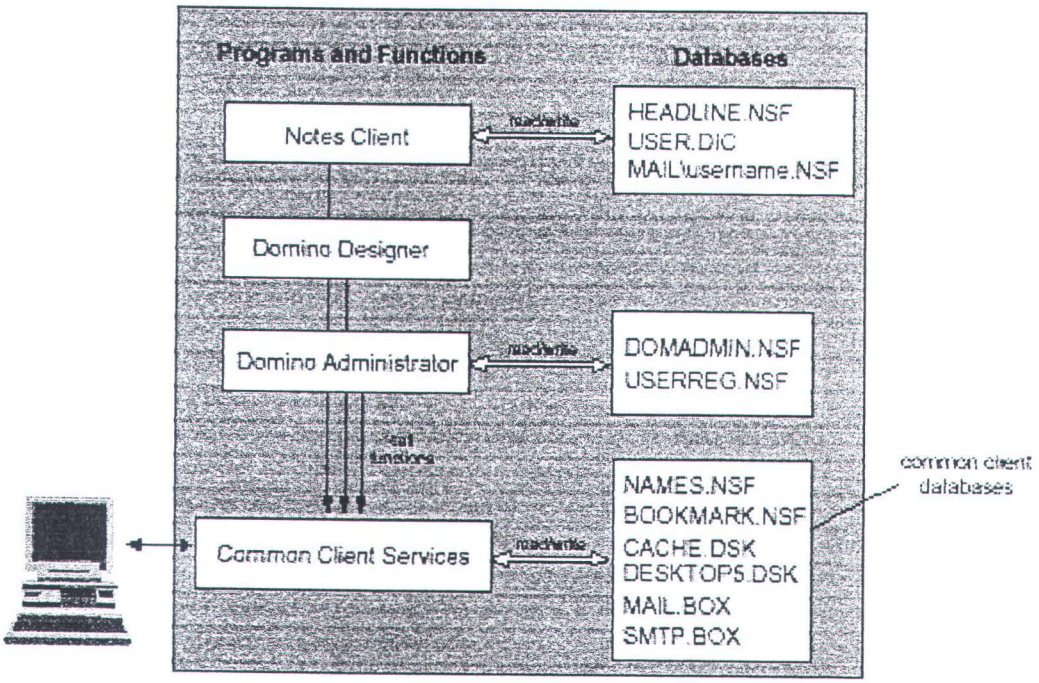
mengirim mail pada *internet mail server*, membaca dan mengirimkan topik ke *internet newsgroup*, mencari direktori-direktori internet, menampilkan HTML dari beberapa Web server serta menggunakan sertifikat X.509 untuk keamanan *(security)*. Beberapa file dan database secara umum ter-*share* oleh semua *client*, database-database tersebut menyediakan fungsi-fungsi spesifik pada Notes client, seperti database MAIL*username*.NSF, adalah file mail user. Untuk memudahkan mail, file ini biasanya tersimpan baik dalam *workstation* pengguna dan dalam *home mail server* pengguna.

- **Domino Designer**, diperuntukkan bagi pengembang aplikasi, menyediakan akses pada elemen-elemen perancangan.Domino Designer menyediakan *IDE* (*Integrated Development Environment*) atau Lingkungan Pengembangan Terpadu yang memiliki sifat *stand-alone*, dimana para pengembang aplikasi Notes menggunakannya dalam membangun aplikasi yang terjamin keamanannya oleh kemampuan dan tingkat pengamanan bertingkat yang dipunyai oleh Lotus Notes. Dengan menggunakan Notes client ataupun *Web browser*, *user* dapat menampilkan aplikasi-aplikasi tersebut. *IDE* (*Integrated Development Environment*) memberikan pengembang aplikasi Notes kemampuan untuk mengakses :
  - Elemen-elemen perancangan Notes, seperti form dan view.
  - Elemen-elemen perancangan Web, seperti frameset dan page.
  - Bahasa Formula Notes, yang meliputi @command dan @function LotusScript, JavaScript dan Java.

- **Domino Administrator**, diperuntukkan bagi sistem administrator, menyediakan akses ke data administrasi. Domino Administrator menyediakan *interface* administrasi yang terintegrasi, dimana sistem administrator menggunakannya untuk mengatur dan mengawasi user, database serta server. Beberapa file dan database secara umum ter-*share* oleh semua *client*, database-database tersebut menyediakan fungsi-fungsi spesifik pada Domino Administrator, diantaranya :

  - DOMADMIN.NSF, adalah database Domino Administrator, yang berisi hal-hal fungsional yang diminta dalam menjalankan Domino Administrator.
  - USERREG.NSF, adalah database *User Registration Queue*, yang menyimpan informasi mengenai user Notes, yang sedang menunggu proses pendaftaran/registrasi .

Gambar 2.2 berikut ini menunjukkan program-program *client* dan komponen-komponen *software* yang digunakan. Meskipun tidak ditunjukkan, NOS *(Notes Object Service)* telah digunakan oleh program-program *client* dan juga oleh servis-servis *client* umumnya, yang mendukung dalam membaca dan menulis database serta dalam mengakses layanan-layanan sistem lain, seperti manajemen memori dan *time-date service*.



Gambar 2.2. Komponen-komponen program client

B. Program – Program Server

Sedang dalam komputer *server*, program Domino Server mendukung koneksi antara komputer *client* dan *server*, serta mengatur bagian dari tugas-tugas *server*. Selain itu juga menampilkan tugas-tugas dari *schedule-driven* database, seperti *routing message* ke mailbox dan meng-update *user account*, ataupun koneksi dengan berbagai macam tipe *client*, sebagai contoh Notes, *Web Browser* dan CORBA-pada server. Program-program server meliputi :

- **Domino Mail Server**

Domino Mail Server mengkombinasikan dukungan penuh terhadap standar internet mail dengan kemampuan *messaging* dan kehandalannya dalam

skala besar, serta kemampuan dari Lotus Notes. Domino Mail Server terintegrasi melalui servis lintas platform yang meliputi akses Web, *group scheduling*, *collaboration* serta *newsgroups* yang semuanya dapat terakses melalui Web browser atau client lain yang standar. Domino Mail Server hanya untuk digunakan pada *messaging*.

- **Domino Application Server**

Domino Application Server merupakan teknologi terdepan yang terintegrasi antara *messaging* dan *application server*. Server dapat dengan mudah mengintegrasikan sistem *back-end* dengan sistem *front-end* bisnis prosesnya.

- **Domino Enterprise Server**

Domino Enterprise Server adalah server yang dibutuhkan ketika akan terjadinya misi yang kritis, penyebaran dalam skala luas dengan kemampuan akses secara terus-menerus dan kemampuan maksimum dalam semua kondisi.

2.2.1.2 Notes Object Services (NOS)

Notes Object Services (NOS), yang ditulis sebagai library dari fungsi-fungsi C dan digolongkan kedalam berbagai layanan *(service)* yang diperlukan oleh Notes, merupakan jantung dari semua *software* Notes. Program-program *client* dan *server*, seperti Notes Client, Domino Designer, Domino Administrator, program Domino Server serta tugas-tugas *server*, telah memanggil NOS untuk *create* dan *modify* database Notes, mengakses jaringan, menyusun dan menjalankan *formula* dan *script* yang melekat dalam database, mengalokasikan memori, mendapatkan informasi waktu *(time-date)* dan seterusnya.

2.2.1.3 Database dan File

Seperti yang telah diketahui, bahwa komputer *server* dan *client* memiliki bentuk database tersendiri. Komputer-komputer *server* memiliki sekumpulan database yang ter-*share* sehingga dapat diakses oleh komputer lain melalui suatu jaringan, sedang komputer *client* juga memiliki sekumpulan database lokal sendiri, dan baik keduanya memiliki file-file lokal.

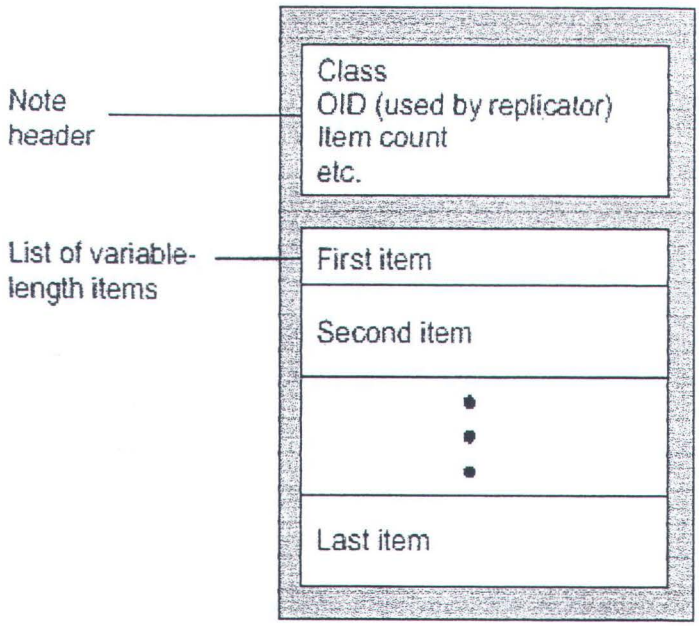
Sebuah database ataupun file adalah lokal, jika hanya terakses oleh program yang berjalan dalam komputer yang sama. Database dalam komputer *client* adalah lokal sebab program-program *client* tidak mempunyai pelaksanaan logis dalam program Domino Server, untuk merespon datangnya permintaan database. Hanya program-program yang berjalan pada komputer *client* saja yang dapat mengakses database dalam komputer *client*.

Ketika database kebanyakan terdiri atas data dalam jaringan Notes, beberapa data memisahkan file-file non-database, seperti file ID dan NOTES.INI. File-file tesebut telah ada dalam komputer *client* dan *server* dan selalu lokal, sebab tidak diantara keduanya baik program *client* maupun *server* mengandung kebutuhan logis untuk meminta ataupun menyediakan akses secara *share* ke file-file non-database.

2.2.2 Notes

Sebuah Notes adalah struktur data sederhana yang menyimpan elemen-elemen perancangan (*form, views* dan seterusnya), data yang dibentuk user *(documents)* dan informasi administrasi, seperti database Access Control List

(ACL). Karena struktur data Notes yang sama menyimpan semua tipe informasi tersebut, maka Notes hanya membutuhkan sebagian single servis NOS untuk *create, read, update* dan *replicate* kebanyakan informasi dalam database Notes. Dibawah ini adalah gambar struktur logikal Notes.



Gambar 2.3. Struktur Logikal Notes

Pada gambar struktur logikal Notes diatas (gambar 2.3), tiap Notes memiliki header kecil yang diikuti daftar item *variable-length*, yang disebut dengan *field*. Header menyimpan informasi umum tentang Notes, termasuk nilai yang mengindikasikan class Notes, seperti *document, form* ataupun *view* serta originator ID-nya (OID). OID berisi Note unik, Universal ID (UNID), yang esensial untuk replikasi. Dalam daftar item memiliki sebuah nama, atribut flag, sebuah nilai serta sebuah tipe nilai, seperti *text* atau *number*.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH NOPEMBER

2.2.2.1 Item-Item Notes

Setiap note mengandung sekumpulan item yang ditentukan oleh class Notes. Sebagai contoh, semua *form* Notes berisi sekumpulan item yang sama, meskipun nilai item berbeda dari *form* Notes ke *form* Notes. Sama halnya juga dengan semua *view* Notes yang berisi sekumpulan item yang sama, meskipun nilai item berbeda dari *view* Notes ke *view* Notes. Bagaimanapun dokumen Notes berbeda, sebab semua dokumen tidak mengandung sekumpulan item yang sama. Karena kumpulan item dalam dokumen bergantung pada *form* yang digunakan untuk membentuk dokumen, sehingga dua dokumen Notes kemungkinan dapat memiliki daftar item yang sangat berbeda.

2.2.2.2 Tipe-Tipe Notes

Terdapat beberapa tipe Notes, yang ditentukan oleh beberapa class Notes yang mana nilainya tersimpan dalam setiap header Notes. Untuk Notes yang lain merupakan variasi dalam class. Untuk itu Notes membedakan tipe dengan ada atau tidaknya item spesifik atau nilai item dalam Notes. Berikut daftar tabel 12 class Notes dasar beserta tipe Notes yang dimiliki tiap class, juga yang mengorganisir class oleh class group.

Tabel 2.1. Tabel 12 Class Notes Dasar

Class Group	NOTE_CLASS	NOTE TYPES
Data	DOCUMENT	Document
Administration	ACL*	Access Control List
	REPLFORMULA	Replication formula
	FIELD	Shared field

	FILTER	Agent, database script, outline, script library
	FORM	Form, frameset, page, subform
	VIEW	Folder, navigator, view
Design-Element	DESIGN*	Design Collection (terstruktur seperti view)
	ICON	Icon
	INFO*	Menggunakan dokumen database ini untuk Help
	HELP	Menggunakan dokumen database ini untuk Help
	HELP_INDEX*	Index untuk Help

* Tiap database hanya mengandung kejadian tunggal dari class tersebut.

2.2.2.3 Data Notes

Data Notes atau *document*, secara khusus terdiri dari sebagian besar database Notes. Setiap dokumen dapat diasosiasikan dengan *form* Notes yang digunakan untuk membuat dokumen dan untuk itu digunakan secara default untuk *view* atau modifikasi dokumen. Sebagai contohnya, beberapa user dapat menggunakan *form* Topik Utama untuk membangun dokumen Topik Utama dalam suatu database. Tiap dokumen topik utama berisi item-item yang didefinisikan oleh *form* Topik Utama. Sebagai tambahan, dokumen Topik Utama diasosiasikan dengan nama dari *form* yang digunakan untuk membangunnya. Asosiasi ini memastikan bahwa *form* yang tepat tersebut, selanjutnya akan digunakan untuk menampilkan dokumen ketika beberapa user menginginkan untuk *view* atau memodifikasikannya.

Konsep *document oriented* adalah sebenarnya dari penerapan konsep dokumen dalam dunia sehari-hari yang berbentuk kertas ke bentuk digital. Konsep

ini sangat berbeda sekali dengan konsep database relasional. Dimana relasional mengenal field kunci yang akan menghubungkan antara satu tabel dengan tabel yang lainnya. Sedangkan dalam Domino, tidak terdapat field kunci. Kita dapat mencari hubungan antar satu dokumen dengan dokumen yang lain dengan menggunakan *View*.

Konsep dokumen ini adalah ibarat sebuah lembar formulir yang kosong berisi kotak-kotak tempat mengisi data yang jika sudah terisi maka form itu disebut dengan dokkumen yang akhirnya akan disimpan dalam lemari file. Kita bisa menganalogikan kotak-kotak tempat isian itu adalah sebagai field-field yang kita definisikan sedangkan tempat penyimpanannya adalah databasenya.

Satu buah dokumen sebenarnya dapat terlihat sebagai satu buah record dalam database relasional. Akan tetapi dalam dokumen tersebut kita dapat memasukkan data-data (seperti dibahas diatas) yang belum tentu terdapat field yang sama dengan data tersebut. Jadi bisa terjadi kemungkinan sebuah dokumen memiliki hanya lima buah field akan tetapi pada saat terjadi transaksi dokumen tersebut menampung data lebih dari 20 item

Ketika menampilkan dokumen, Notes menggunakan model *late-binding* untuk mengaplikasikan *form* ke dokumen. Pendekatan ini menyediakan fleksibilitas lebih daripada dengan melakukan model yang mengikat erat sebuah dokumen pada satu *form* spesifik. Sebagai contohnya, meski tiap dokumen memiliki default *form* yang diasosiasikan dengannya, *form-form* alternatif dapat diaplikasikan pada dokumen sehingga isinya dapat ditampilkan dalam beberapa cara yang berbeda. Sebagai tambahan, pada keseluruhan penggunaan nilai *field*

dan / atau hasil dari komputasi formula, Notes dapat mengontrol secara dinamik *form* mana untuk digunakan. Baik keduanya dari Notes client dan Web browser mendukung model unik *late-binding* dalam menampilkan informasi.

Meskipun *form* dan dokumen biasanya tersimpan secara terpisah, sebuah *form* dokumen dapat tersimpan dalam dokumen itu sendiri. *Form* ini secara aktual tersimpan sebagai kumpulan item yang dimiliki dokumen. Menyimpan sebuah *form* dengan cara ini memungkinkan untuk meng-*copy* dokumen dari satu database ke database lainnya, yang tidak mengandung kebutuhan form untuk *view* dan *edit* dokumen. Bagaimanapun option ini, memiliki kekurangan didalamnya jika digunakan secara terus-menerus, hal ini memberikan arti menambah ukuran dari database.

2.2.2.4 Administrasi Notes

Terdapat dua tipe Notes yang dibentuk dan diatur oleh manajer database, yaitu Access Control List Notes (ACL) dan Replication Formula Notes. Setiap database hanya memiliki satu ACL note, yang harus mendaftarkan hak akses berbagai *user, server* dan *group* ke Notes lain dalam database.

2.2.2.5 Elemen Perancangan Notes

Database desaigner dapat membuat beberapa elemen-perancangan Notes sebagai berikut :

- *Field Notes*, yang mendefinisikan *shared field* atau *field* yang dapat berada pada lebih dari satu *form*. Sebuah field adalah sebuah area dalam sebuah form

yang berisikan sebuah tipe informasi yaitu teks, multi teks (mampu menyimpan berbagai bentuk tipe, grafik, dan obyek multimedia), multiple-choice (keyword), angka, tanggal dan waktu, dan nama user. Field-field dalam sebuah form mendefinisikan bentuk dan macam informasi yang dimuat dalam sebuah dokumen. Anda bahkan dapat menghubungkan field-field dengan akses kontrol section yang dapat mengatur hak untuk mengedit field-field dengan orang-orang yang sudah ditentukan

- Filter Notes, menjaga kode dari aplikasi.
- Form Notes, yang mengontrol dalam *creating, viewing* dan *modifying* berbagai dokumen individual.
- View Notes, yang menspesifikasikan bagaimana untuk meng-index dokumen dalam database serta menyediakan akses pada subset spesifik dari dokumen.
- Design-Collection Notes, yang sama halnya dengan *view*, index elemen-elemen perancangan, *replication formula Notes* serta *help-index Notes*.
- Icon Notes, yang berisi database icon.
- Info Notes, yang berisi tentang *(About)* dokumen database tersebut, yang muncul ketika user pertama kali membuka database.
- Help Notes, berisi informasi bantuan mengenai aplikasi.
- Help-Index Notes, yang meng-index help Notes.
- Private design-element Notes, yang berisi elemen-elemen perancangan, dimana user individual menambahkan ke database. Tiap Notes tersebut berisi sebuah *field* yang mengindikasikan "real"class – sebagai contoh *form, view* dan seterusnya dari Notes. Ketika perancang aplikasi me-refresh elemen-

elemen perancangan dalam database, elemen-perancangan private Notes tidak berubah.

2.2.2.6 Hierarki Notes

Untuk mendukung aplikasi yang mengkategorikan dan mensubkategorikan dokumen, data indivual Notes dapat disusun secara hierarki. Sebuah Notes dapat sebagai Notes utama, respon ke Notes utama atau respon ke respon Notes. Terdapat batasan 32 level antara Note utama dengan berbagai respon paling dalam.

Untuk mengatur hubungan dalam hierarki Notes, Notes meletakkan informasi kedalam sebuah header Notes serta daftar itemnya. Masing-masing dokumen dalam database Notes adalah teridentifikasi secara unik dengan UNID. Dokumen respon dan respon-ke-respon berisi *field* spesial yang dinamakan $Ref, yang mengandung UNID dari dokumen "parent". Field ini melayani dalam mengatur hubungan dari hierarki respon.

2.2.3 Bahasa Pemrograman

Bahasa pemrograman yang dapat digunakan dalam membuat dan mengembangkan aplikasi ataupun menjalankan agents adalah diantaranya formula, LotusScript, Java dan JavaScript.

Formula dan LotusScript yang terdapat sebuah aplikasi notes adalah dapat mendefinisikan isi dokumen-dokumen dan view-view, menentukan bagaimana data-data ditampilkan, dan menjalankan workflow yang kompleks dan tugas-tugas yang dijalankan secara otomatis.

LotusScript adalah sebuah bahasa pemrograman yang dapat dibangun ke dalam hampir semua disain-disain komponen, dan khususnya untuk memproses banyak dokumen, melakukan program-program iterasi dan percabangan dalam sebuah program, melakukan tugas-tuigas yang sulit mungkin akan membutuhkan Formula (@function) yang sangat panjang., dan juga berguna untuk menghubungkan ke non-Notes database dengan menggunakan metode ODBC. Lotus Script juga berguna untuk menjalankan tugas-tugas pada saat-saat tertentu seperti membuka, menutup, menyimpan dokumen atau memasukkan atau mengedit sebuah field.

Dalam Lotus Notes Release 4.5 dan Domino server, terdapat tambahan beberapa kelas dari kelas-kelas dalam LotusScript program yang memberikan kemampuan lebih kepada pemrogram Lotus Notes.

Bahasa Java digunakan dalam pemrograman di aplikasi Domino adalah dikarenakan dukungan penuh terhadap multi-platform yang juga tidak mengenal mesin. Dan juga bahasa Java sudah sangat dikenal dan hampir digunakan untuk bahasa pemrograman diseluruh dunia.

- **Agent dan Action**

Database agent, dan form, subform, view, dan action folder adalah seluruhnya adalah kemampuan disain yang dapat mengotomatisasi pada aplikasi-aplikasi.

Contohnya adalah sebuah agent yang mengotomasi proses .pendaftaran user baru di sebuah situs.

2.3 Notes dan Web

Bagian ini menjelaskan tentang arsitektur dimana Notes menggunakannya untuk menyediakan servis-servis pada web dan membicarakan bagaimana komponen - komponen dari internet saling terintegrasi dalam Notes client.

World Wide Web mengalami revolusi dengan membuatnya lebih mudah bagi orang dalam mengakses serta mempublikasikan informasi dalam sebuah cara yang tidak dapat digambarkan secara geografis dan batas-batas organisasional.

Pertama kali web server dibentuk menurut standar yang membuat seluruh internet mengarah pada sebuah penyimpanan dokumen besar, yatu :

- URL (Uniform Resource Locator), merupakan penamaan yang telah ditentukan, yang menunjukkan lokasi dari dokumen pada internet.
- HTTP (Hypertext Transfer Protocol), merupakan protokol komunikasi untuk mencapai dokumen yang jauh.
- HTML (Hypertext Markup Language), merupakan format standar dalam menampilkan text dokumen.

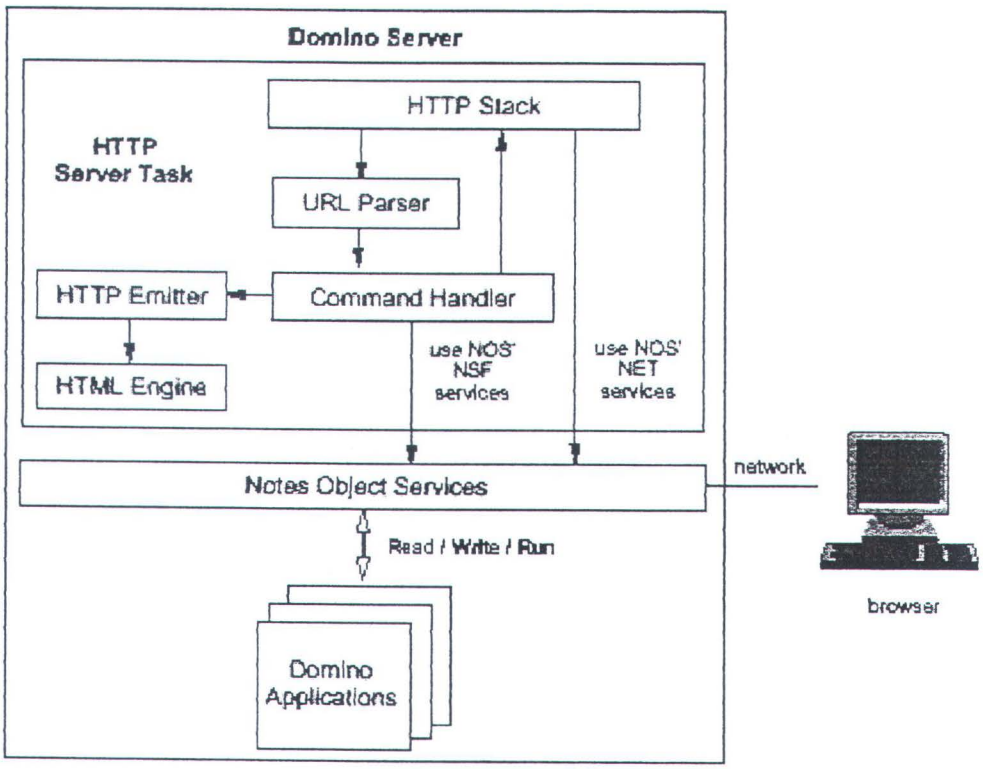
Generasi pertama *web server* adalah berdasarkan *file server* yang diterima *file request* melalui koneksi jaringan dan direspon dengan mengambil file dari hard disk serta meng-*copy*-nya ke jaringan.

Generasi berikutnya dari *web server* adalah dimana server-server aplikasi yang secara dinamik dihasilkan dari isi berbasis *web client* pada *request user*. *Web server* tersebut dapat menyediakan informasi dari berbagai sumber, seperti database, serta tidak membutuhkan data yang telah dikodekan dalam HTML.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

2.3.2 Arsitektur Domino Web Server

Berikut gambaran arsitektur Domino Web Server :



Gambar 2.4. Arsitektur Domino Web Server

Domino Web Server adalah aplikasi TCP/IP yang mengimplementasikan protokol HTTP. Domino menjawab permintaan URL dari Web client dengan mengirimkan kembali page dari data dalam HTML. Selain itu juga menangani permintaan URL dan form HTML yang men-*trigger* program eksekusi menurut spesifikasi CGI (*Common Gateway Interface*). Pada aspek ini, Domino Server menunjukkan seperti halnya server HTTP lainnya, menanggapi menurut cara standar pada permintaan URL standar.

Jantung dari server aplikasi Web Domino adalah tugas server HTTP (*HTTP server task*). Permintaan dari web client secara langsung dilanjutkan ke

HTTP task untuk diproses. Tugas *(task)* memiliki semua fasilitas biasa untuk mengakses halaman HTML yang tersimpan dalam sistem file dari host platform dan menjalankan program-program CGI. Ditulis dalam C/C++, *HTTP task* menggunakan NOS untuk mengintegrasikan dengan Domino Server.

HTTP task adalah *multithreaded.* Task mendengar permintaan web client dan mengirim respon, serta menjalankan tugas normal server dengan identitas sebuah file ID yang terletak disuatu tempat pada host file system. Dalam mengimplementasikan keamanan untuk web client, urutan melayani permintaan harus mengasumsikan identitas dari pengguna yang masuk (*logged-in user*).

Gambar 2.4, menggambarkankan komponen inti dari HTTP task. Susunan HTTP meliputi semua kode yang sepakat dengan didalam dan diluar batas komunikasi HTTP. URL parser menangani datangnya *Domino URL call.* Pengedar HTML menyediakan *outgoing flow* dari informasi yang merupakan hasil dari URL call. Pengedar HTML menggunakan HTML engine, yang bertindak sebagai sumber untuk standar yang mendefinisikan format sebagaimana mestinya dari beberapa informasi yang diterjemahkan kedalam HTML.

Pada pusat penanganan perintah, yang merupakan *link* secara langsung antara HTTP task dan NOS, layer ini menerjemahkan perintah-perintah kedalam format yang ekuivalen dengan format dari perintah yang dibuat dari Notes client atau disebut dari Dbserver task.



Proses detailnya adalah sebagai berikut :

Ketika HTTP task menerima permintaan dari web browser, sebuah koneksi dibuat pada HTTP stack, yang mengatur koneksi antara web client dan server. Pada saat ini SSL *(Secure Socket Layer)* dapat diimplementasikan. Kemudian HTTP stack mengirim permintaan secara langsung ke URL parser, yang menentukan jika hal tersebut standar URL atau spesial Domino URL, kemudian jika permintaan tersebut adalah standar URL maka URL parser akan mengirim informasi ke HTTP stack yang akan memprosesnya sebagai perintah HTTP sederhana. Jika permintaan tersebut adalah dalam bentuk Domino URL, maka URL parser akan memecah URL menjadi bagian yang berbeda, melakukan serangkaian pemeriksaan (*check*), menyediakan perintah (*command*) yang implisit yang diperlukan serta meminta command handler yang sesuai. Command handler ini mengatur semua detail yang berhubungan dengan tiap perintah, membuktikan identitas yang benar untuk tujuan keamanan, pengaksesan NOS, mengeksekusi formula dan script dan mengambil informasi.

HTTP task dapat dianggap sebagai emulator Notes client dengan integrasi yang erat dengan NOS dan berjalan dalam Domino Server. Setelah data dari Domino Server didapatkan sesuai dengan *action*/perintah yang dikirimkan melalui command handler, informasi dikirimkan kembali ke HTTP server. Hasil tersebut kemudian dilewatkan ke HTML Emitter, yang akan menggunakan HTML Engine untuk menerjemahkan informasi kedalam halaman HTML, kemudian halaman

dikirim ke HTTP stack, yang membentuk koneksi ke client dan mengirimkan data kepada user.

Komponen-Komponen HTTP Server Task

Berikut komponen-komponen dari HTTP task secara lebih detail.

- **HTTP Stack**

  HTTP stack ditulis dalam C dan merupakan gateway ke HTTP task. Semua informasi masuk ke dalam Domino dari Web dan keluar dari Domino ke Web melalui layer ini. HTTP stack memiliki empat fungsi utama, yaitu :

  1. Connection Management

     Disini membentuk koneksi TCP/IP ke Web dan menangani semua interaksi HTTP untuk koneksi.

  2. URL Dispatch

     Disini menggunakan protokol HTTP untuk mengirim text HTML ke browser.

  3. HTTP Command Redirection

     Task ini mengeksekusi command URL sederhana untuk task seperti menampilkan file HTML, menjalankan servlet atau melakukan beberapa action yang tidak ada hubungan dengan Domino Server. Servis ini mulai ketika parser menolak URL sebagai non-Domino URL.

  4. SSL Implementation

- **URL Parser**

  URL parser membedakan Domino URL dari standar URL dan menterjemahkan Domino URL kedalam command dimana command handler

dapat mengetahuinya. Sebuah Domino URL menspesifikasikan sebuah *path* atau replika ID dari database Notes (NSF) atau berupa server command seperti ?OpenServer.

Pertama, parser memecah URL menjadi tiga bagian, yaitu *mandatory object identifier*, *optional command* dan *argument*.

Kedua, parser menerjemahkan Domino URL kedalam perintah-perintah dimana command handler dapat memahaminya. Untuk melakukan ini, parser menghubungkan obyek yang dikenal dengan perintah yang sesuai untuk obyek itu. Berikut contoh dari Domino URL tertentu:

http://www.lotus.com/site/app.nsf/ViewUtama?OpenView&start=10

Contoh URL ini mereferensi **ViewUtama** dalam database **site/app.nsf** pada server **www.lotus.com.** View ini dipanggil dengan menuliskan nama dari view yaitu **ViewUtama**. Bagian perintah (setelah tanda tanya (?)) secara sintaks adalah opsional, jika kosong, maka URL parser akan menentukan tipe dari obyek dan menyediakan perintah implisit yang sesuai untuk tipe obyeknya. Perintah spesifik pada contoh diatas adalah ?OpenView dan memiliki argumen dengan memanggil &start yang menspesifikasikan bahwa Domino Server harus mulai pada dokumen ke-10 dari view ViewUtama.

- **Command Handler**

  Command Handler menggunakan empat cache untuk meningkatkan kinerja secara efisien, yaitu *database cache, design cache, HTMLcache* dan *static cache*.

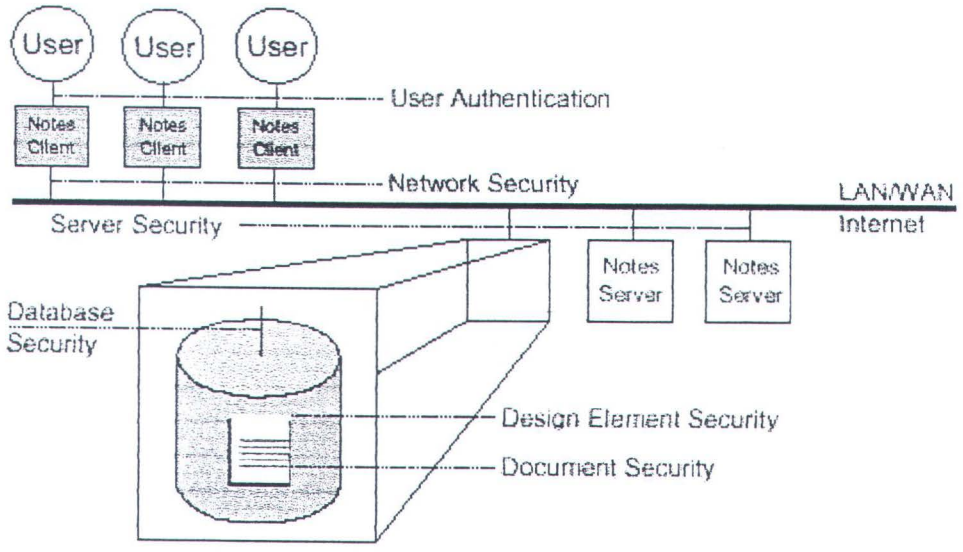
- **HTML Emitter dan Engine**

  Proses DominoWeb Server berakhir dengan membuat HTML untuk dikirim kembali ke browser. Sebuah *intelligent HTML emitter* digunakan oleh semua command handler. Emitter menggunakan table-driven HTML engine yang memiliki pengetahuan mengenai sintaks dan semantik (keterkaitan) dari HTML, yang memastikan bahwa output menyesuaikan diri pada standar HTML.

2.4 Keamanan Lotus Notes / Domino

Lotus Notes/Domino memiliki model keamanan yang berlapis, dimana baik administrator sistem dan pengembang aplikasi (*application developer*) dapat menyesuaikan untuk memenuhi kebutuhan spesifik organisasional. Model keamanan yang berlapis ini mirip dengan sistem keamanan rumah. Dimana pada sistem kemanan rumah, pengunjung tidak dapat memasuki halaman rumah tanpa adanya akses ke gerbang/pintu depan. Pengunjung tidak dapat memasuki rumah tanpa kunci pintu depan, dan seterusnya hingga dapat masuk bagian-bagian rumah. Dengan ide ini dapat dipastikan bahwa tidak dapat memasuki bagian-bagian dari rumah tanpa melalui tiap lapisan keamanan.

Lotus Notes/Domino memiliki fitur-fitur keamanan yang melindungi komponen-komponen utama, yaitu pengguna (*user*), jaringan (*network*), server, database, elemen-elemen perancangan (*design elements*) dan dokumen. Keamanan menjangkau sistem secara vertikal dari level pengguna hingga ke level dokumen, serta secara horisontal yang melalui setiap komponen. Setiap komponen memiliki mekanisme keamanan yang digunakan secara konsisten melalui semua komponen pada level tersebut. Sebagai contoh, semua pengguna yang mencoba untuk menggunakan Notes client harus ter-otentikasi. Semua database memiliki sebuah ACL untuk mencegah akses yang tidak diinginkan ke data. Semua dokumen dapat membatasi siapa saja yang dapat membaca (*read*) dan menulis (*write*) dokumen tersebut. Berikut gambar level keamanan dari Notes/Domino.



Gambar 2.5. Tingkatan/level keamanan Notes/Domino

- **Keamanan Pengguna (*User Security*)**

  Untuk mencegah seseorang dari menyamar sebagai pengguna dan mendapatkan hak akses ke data pengguna tersebut, maka Notes/Domino menyediakan file ID pada setiap pengguna. Sebuah file ID mengandung kunci enkripsi dan sertifikat dimana Domino Server menggunakannya untuk memeriksa dan membuktikan keaslian/kebenaran dari file dan secara tak langsung keaslian dari orang yang memiliki file tersebut. Untuk menyediakan tambahan keamanan, Notes/Domino membolehkan pengguna untuk menggunakan password dalam meng-enkripsi bagian yang sensitif dari file ID. Hal yang harus diperhatikan adalah bahwa pengguna tidak membocorkan password yang telah dien-enkripsi file ID-nya. Jika pengguna yang tak berhak mendapatkan password dan meng-*copy* file ID tersebut, maka merubah password pada file ID yang asli tidak memperbaiki bahaya yang terjadi. Pengguna yang tidak berhak masih dapat menggunakan ID untuk membaca mail yang ter-enkrip serta mengakses server, kecuali pemeriksaan pasword (*password-checking*) telah di aktifkan (*enable*) pada server.

- **Keamanan Jaringan (*Network Security*)**

  Lotus Notes/Domino menyediakan beberapa fasilitas keamanan jaringan dengan meng-enkripsi pesan-pesan (*messages*) yang dikirim antara klien dan komputer server. Notes/Domino tidak dapat menjalankan tipe lain dari keamanan jaringan seperti mencegah pengguna yang tidak berhak dari membobol masuk atau menyadap pada jaringan. Perlindungan yang terbaik terhadap penyadapan adalah dengan meng-enkripsi lalu-lintas jaringan.

- **Keamanan Server (*Server Security*)**

  Keamanan server terutama dikendalikan oleh informasi yang tersimpan dalam direktori Domino. Direktori Domino menspesifikasikan pengguna, server serta grup atau para pengguna dan server mana yang diberikan atau diabaikan aksesnya pada server dan/atau spesifik port server. Selain itu, setiap server dapat menspesifikasikan dengan sesuai pengguna, server dan grup mana yang dapat menggunakan koneksi melalui luar (*passthru*), membuat database baru dan/atau membuat replika database pada server. ACL database juga dapat membatasi siapa saja yang dapat membuat replika database.

- **Keamanan Database (*Database Security*)**

  Setiap database memiliki ACL-nya sendri yang menspesifikasikan pengguna dan server mana yang dapat mengakses database dan tugas/task mana yang dapat mereka lakukan.

- **Keamanan Dokumen (*Document Security*)**

  Meskipun beberapa pengguna memiliki akses untuk membaca dan/atau mengubah dokumen dalam database, dokumen individual dapat membatasi akses pengguna. Membatasi akses dapat dicapai dengan mengunakan field-field *Reader, Author* dan *Signer*, kemudian formula-formula *hide-when* serta kunci enkripsi yang dilampirkan pada sebuah field khusus dalam dokumen. Selain itu, sebuah kunci enkripsi dapat meng-enkripsi seluruh dokumen.

Tabel 2.2. Tabel Access Control List (ACL) Domino

	Manager	Designer	Editor	Author	Reader	Depositor	No Access
Create Document	√	√	√	Option		√	
Delete Document	Option	Option	Option	Option			
Create Personal Folder/View	√	√	Option	Option	Option		
Create Shared Folder/ View	√	√	Option				
Read Public Document	√	√	√	√	√	Option	Option
Write Public Document	√	√	√	Option	Option	Option	Option
Create Personal Agent	√	√	Option	Option	Option		
Create LotusScript Agent	√	option	option	Option	Option		

2.5 UNIFIED MODELING LANGUAGE (UML)

UML atau Unified Modeling Language adalah bahasa untuk menspesifikasikan, memvisualisasikan dan membangun produk sistem perangkat lunak. UML merupakan notasi yang bertujuan untuk memodelkan sistem dengan menggunakan pendekatan yang berorientasi objek.

UML merupakan standar industri untuk model yang berorientasi objek yang dibangun dari tiga metoda pendahulunya, yaitu metoda Booch (oleh Grady Booch), metoda OMT (*Object Modeling Technique* oleh James Rumbaugh) dan metoda OOSE (*Object Oriented Software Engineering* oleh Ivar Jacobson).

Server aplikasi web dapat menyaring dokumen dan merubahnya berdasarkan user yang berhak/otentifikasi.

Untuk merubah dokumen, server dapat :

- Merubah penampilan dari dokumen
- Membatasi akses pada beberapa isi dokumen
- Merubah apa yang bisa dilakukan *user* dengan dokumen.
- Menyesuaikan dengan software user

2.3.1 Domino Web Server

Lotus Domino adalah server aplikasi web yang menyediakan sekumpulan servis yang terintegrasi, dimana para pengembang menggunakannya dalam mengamankan, serta membangun aplikasi internet maupun intranet secara interaktif. Program Domino Server menjalankan tugas-tugas server. Salah satu tugasnya adalah dalam HTTP, dimana mengubah server Domino kedalam bentuk server aplikasi Web yang nyata. Bagaimanapun berbeda dengan Web file server tradisional dalam tiga hal penting, yaitu :

- **Isi Dinamik**

  Domino secara otomatis membentuk halaman HTML dari penyimpanan obyek secara langsung, *user identity*, *user preference*, tipe *client* dan input data. Berbeda dengan penyimpanan file, obyek memisahkan form dari isinya, untuk itu susunan halaman *(page)* berasal dari *page data*. Ketika pengguna meminta sebuah dokumen atau page, Domino secara dinamik menggabungkan form dan isi. Pada saat adanya permintaan, Domino juga dapat

mengintegrasikan informasi dari sumber lain seperti database relational. Para pengembang menggunakan penyimpanan efisien dan menggunakan metode ini untuk merancang berbagai aplikasi yang dapat di-*custom*-isasi, seperti dapat menggantikan identitas dan profil pengguna. Domino menggunakan sebuah direktori untuk mengatur informasi serta otentikasi pengguna. Setelah pengguna terotentikasi, ACL memberikan hak pengguna untuk mengakses pada informasi yang dibutuhkan.

- **Interaksi Pengguna**

  Domino bukan hanya memberikan pelayanan meng-kastem infomasi tetapi juga membiarkan pengguna untuk berinteraksi dengan aplikasi. Dengan menggunakan sintaks URL, penguna *Web* dapat memberikan perintah-perintah pada *Domino Server*. Perintah-perintah tersebut kemungkinan membentuk, mengubah atau menghapus informasi, atau menjalankan *agent* pada server. Para pengembang aplikasi dapat membuat *point-and-click UI (User Interface)* untuk aplikasi dan program yang dilakukan dengan menggunakan *simple @function*, daripada membangun program CGI atau Perl. Kemampuan mengubah (edit) berbeda dengan solusi lain karena mereka tidak membutuhkan pengertian end user untuk mengetahui dimana informasi tersimpan dalam sistem file. Level akses yaitu *No Access, Depositor, Reader, Author dan Editor*. Para pengembang dapat mendefinisikan *access role* lebih dulu dalam mem-*filter* akses dari *class* yang dimiliki pengguna. Hanya Domino yang dapat menyampaikan isi dinamik berdasarkan pada *class* pengguna, juga dapat mendefinisikan aplikasi secara fungsional berdasarkan

pada *class* pengguna. Fleksibilitas ini sangat berguna dalam aplikasi bisnis yang digunakan oleh berbagai macam pengguna, yang harus berpatisipasi dengan cara yang jelas.

- **Pemrosesan Workflow dan Page**

Ketika pengguna menyampaikan informasi kemudian menekan tombol untuk menjalankan *agent*, servis Domino yang banyak dapat dipanggil kedalam *action* untuk mengumpulkan informasi lebih banyak, proses informasi atau memasukkan informasi melalui proses *workflow serial* atau *paralel*. Pelayanan-pelayanan pengintegrasian perusahaan dapat memasukkan informasi ke *relasional, host* dan sistem proses transaksi. Pelayanan agent dapat bertindak pada informasi ke perawatan rutin aplikasi secara otomatis, sebagai contoh penambahan pengguna baru. Pelayanan *messaging* dapat me-*route* informasi dari individu ke individu atau departemen ke departemen hingga proses pekerjaan selesai. Perlengkapan lain seperti faksimili dan *pager* dapat digunakan dalam pemrosesan informasi. Fitur tersebut sangat berguna ketika membangun aplikasi transaksi bisnis yang mempengaruhi sistem yang ada dan mengotomatiskan berbagai aktivitas bisnis setiap hari.

Yang membedakan Domino dari Web Application Server dan Web File Server adalah pelayanan-pelayanannya yang memungkinkan dalam menampilkan isi secara dinamik dan dalam mengatur isi sebagai aliran melalui proses yang telah didefinisikan. Hal ini termasuk penyimpanan obyek,

direktori, keamanan, *workflow, agent, enterprise integration* dan *messaging services.*

- **Model Obyek Domino Web Server**

Arsitektur yang membuat Domino menjadi server aplikasi Web yang efisien, kaya pelayanan berdasarkan pada konsep yang konsisten dikeseluruhan Domino: perancangan yang *object-oriented.* Domino Web server dibangun melalui serangkaian obyek yang merupakan elemen-elemen model dari database Notes dan struktur dokumen, serta mendefinisikan penyajian dari data tersebut. Selain itu juga lingkungan *runtime* untuk aplikasi, menyediakan metode yang mengimplementasikan akan kebutuhan konversi dan perhitungan yang memungkinkan *browser* untuk menjalankan aplikasi. *Web Server* yang lain tidak memiliki model data yang dibangun sendiri, penyajian paradigma atau lingkungan eksekusi aplikasi,sehingga para *programmer* harus membangun model-model data, memutuskan bagaimana menampilkan informasi dan menyesuaikan arsitekturnya pada kemampuan yang terbatas dari spesifikasi CGI atau Web Server vendor API.

Sebuah badan standarisasi industri, OMG (*Object Management Group*) kemudian menetapkan standar UML v1.0 pada November 1997

2.5.1 Diagram UML [CRA - 98]

UML memiliki sembilan jenis diagram sistem, yaitu :

- Aktifitas / Activity
- Class
- Collaboration
- Component
- Deployment
- Object
- Use case
- Sequence
- State

Kesembilan diagram di atas dapat dikelompokkan ke dalam tiga kategori, yaitu statis, dinamis dan arsitektural.

Diagram statis (*Static Diagram*) menggambarkan struktur dari suatu sistem dan fungsi serta tanggungjawabnya. Yang termasuk ke dalam diagram statis adalah Use case, Class dan Object diagram.

Diagram dinamis (*Dynamic Diagram*) menggambarkan interaksi atau hubungan-hubungan yang didukung oleh sistem tersebut. Yang termasuk di dalam kategori diagram dinamis ini diantaranya Activity, Collaboration, Sequence, State dan Use Case diagram.

Dan diagram arsitektural (*Architectural Diagram*) menggambarkan realisasi dari sistem tersebut menjadi komponen-komponen yang dapat dijalankan dan dieksekusi. Yang termasuk ke dalamnya adalah Component dan Deployment diagram.

BAB III
PERANCANGAN SISTEM APLIKASI

BAB III

PERANCANGAN SISTEM APLIKASI

Pada bab ini dijelaskan mengenai perancangan sistem aplikasi penjualan buku secara online BookStore.Inc, yang merupakan prototipe dari sebuah aplikasi *e-commerce* di internet.

Rancangan aplikasi ini ditujukan untuk memberikan gambaran secara umum terhadap aplikasi yang dibuat. Rancangan ini berguna untuk menunjang aplikasi yang akan dibuat sehingga kebutuhan akan aplikasi tersebut dapat diketahui sebelumnya. Dengan rancangan aplikasi juga akan mempermudah untuk mengadakan pengembangan lebih lanjut terhadap aplikasi yang dibuat.

Perancangan dari aplikasi toko buku online BookStore.Inc akan dilakukan menggunakan notasi UML (*Unified Modelling Language*) yang didukung oleh tool Rational Rose versi 2000 Enterprise Edition, dengan menggunakan pendekatan pembuatan perangkat lunak bolak-balik (*Round-Trip Software Development*)[1] **[DTK-96]**, dimana rancangan yang telah selesai memiliki kemungkinan untuk berubah apabila telah memasuki tahap implementasi nantinya.

Pendekatan dengan UML ini, karena database Notes menyimpan informasi berisi obyek-obyek, yang berupa kumpulan dokumen (data Notes) beserta elemen-elemen perancangan yang ada di dalamnya.

---

[1] Diperkenalkan oleh Grady Booch dalam bukunya Object-Oriented Analysis and Design with Applications, 1994

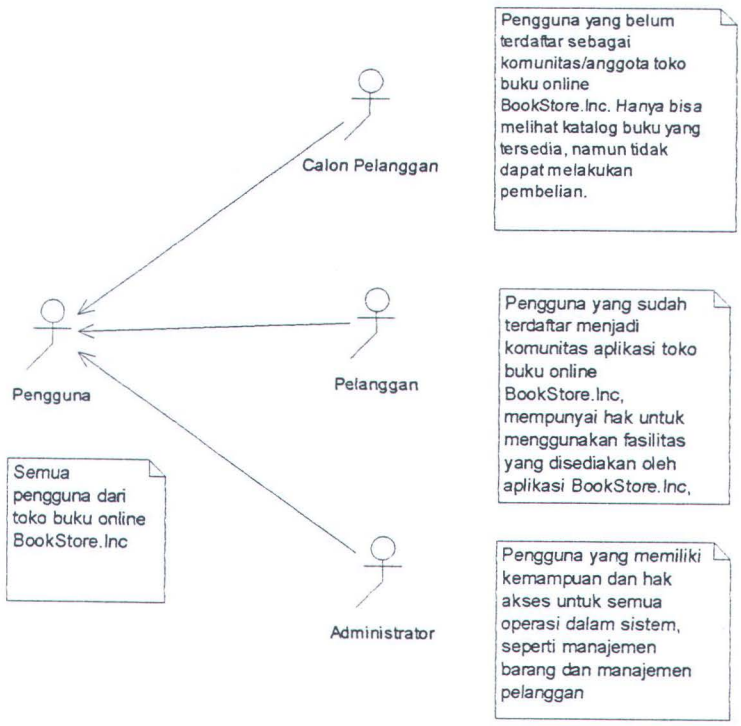
MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH – NOPEMBER

3.1 USE CASE VIEW

Pemodelan use case view menggambarkan urutan dari transaksi-transaksi yang terkait yang dilakukan oleh aktor dengan sistem. Dengan use case kita dapat mengetahui proses-proses yang terjadi di dalam sistem.

2.2.2 Deskripsi Aktor

Terdapat beberapa aktor yang berhubungan dengan toko buku online BookStore.Inc, keterangan berikut merupakan penjelasannya:



Gambar 3.1. Aktor sistem toko buku online BookStore.Inc

- Pengguna

  Semua pengguna atau pengakses situs BookStore.Inc.

- Calon Pelanggan

  Pengguna yang belum terdaftar sebagai komunitas/anggota toko buku online BookStore.Inc. Hanya bisa melihat katalog buku yang tersedia, namun tidak dapat melakukan pembelian

- Pelanggan

  Pengguna yang sudah terdaftar menjadi komunitas aplikasi toko buku online BookStore.Inc, mempunyai hak untuk menggunakan fasilitas yang disediakan oleh aplikasi BookStore.Inc, antara lain melakukan pembelian barang.

- Administrator

  Pengguna yang memiliki kemampuan dan hak akses untuk semua operasi dalam sistem, seperti manajemen barang dan manajemen pelanggan.

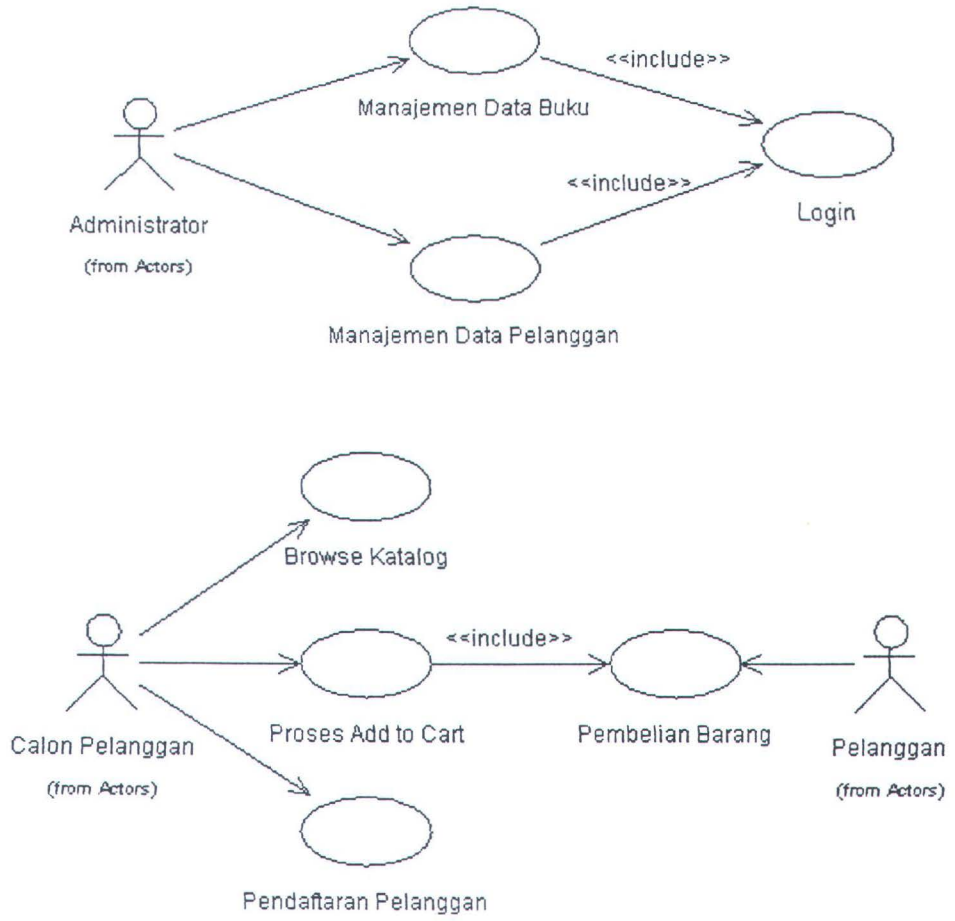
2.2.3 Diagram Use Case

Di dalam sistem toko buku online BookStore.Inc terdapat beberapa proses utama yaitu proses browse katalog, proses addtocart, proses pendaftaran pelanggan, proses pembelian buku, proses manajemen data buku, dan proses manajemen data pelanggan.

Proses browse katalog, addtocart, pendaftaran pelanggan, dan pembelian buku hanya dapat dijalankan dari web dengan menggunakan web browser. Sedangkan untuk proses pengiriman barang/buku tidak dimasukkan dalam proses aplikasi ini dengan asumsi proses pengiriman merupakan proses yang akan otomatis dijalankan setelah barang dibeli oleh pelanggan di situs ini.

Sedangkan proses manajemen data buku dan manajemen data pelanggan hanya dapat dilakukan oleh administrator dan dapat dijalankan baik lewat web dengan menggunakan web browser maupun dengan menggunakan Notes Client

Gambar 3.2. berikut ini menggambarkan diagram use case utama aplikasi BookStore.Inc



Gambar 3.2. Diagram Use Case Utama Sistem BookStore.Inc

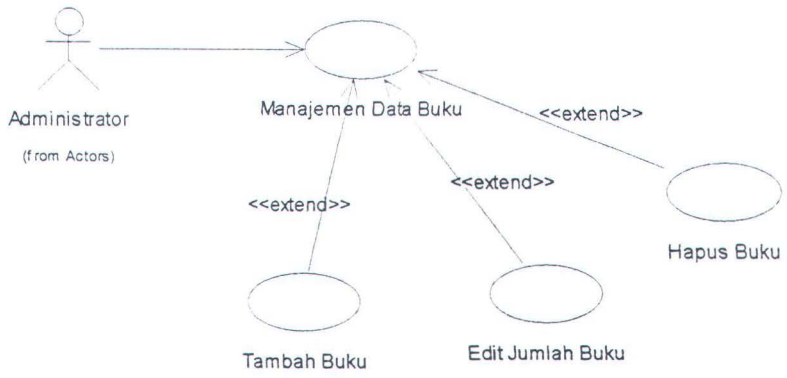
Dari gambar diatas dapat dilihat bahwa ada dua proses utama yang dapat dilakukan oleh administrator yaitu proses manajemen data buku dan manajemen data pelanggan.

Sedangkan calon pelanggan dapat melakukan tiga proses utama, yaitu proses browse katalog, proses addtocart, dan proses pendaftaran pelanggan. Browse katalog memungkinkan calon pelanggan untuk melihat-lihat buku yang ditawarkan oleh toko buku online BookStore.Inc. Proses addtocart memungkinkan calon pelanggan untuk menambah, mengurangi dan mengupdate jumlah buku ke dalam Keranjang Belanja, serta session recovery dimana pelanggan yang telah mengisi Keranjang Belanja kemudian tidak ingin melakukan pembelian pada saat itu dapat melakukannya pada saat lain tanpa harus melakukan proses addtocart lagi. Pendaftaran pelanggan memungkinkan calon pelanggan untuk menjadi anggota komunitas aplikasi toko buku online BookStore.Inc.

Pelanggan dapat melakukan proses pembelian barang yang merupakan proses inti dari aplikasi ini, yang di dalamnya termasuk pula proses add to cart, cek out untuk konfirmasi pembelian dan proses validasi pembayaran.

3.1.2.1 Diagram Use Case Manajemen Data Buku

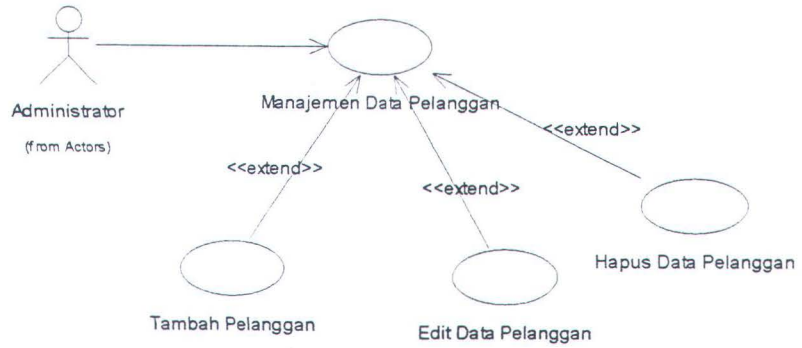
Manajemen terhadap data buku hanya bisa dilakukan oleh administrator. Manajemen yang dimaksud meliputi aktifitas-aktifitas manambah, menghapus dan memodifikasi data dari buku. Proses ini dapat dilihat pada gambar 3.3 di bawah ini.



Gambar 3.3. Diagram Use Case Manajemen Buku

3.1.1.2 Diagram Use Case Manajemen Pelanggan

Manajemen terhadap pelanggan juga hanya bisa dilakukan oleh administrator. Manajemen yang dimaksud meliputi aktifitas-aktifitas menambah, menghapus, dan mengedit data data dari buku. Proses ini dapat dilihat pada gambar 3.4 di bawah ini .

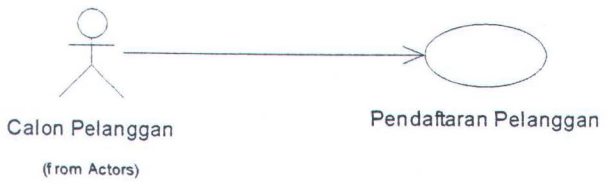


Gambar 3.4. Diagram Use Case Manajemen Pelanggan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH – NOPEMBER

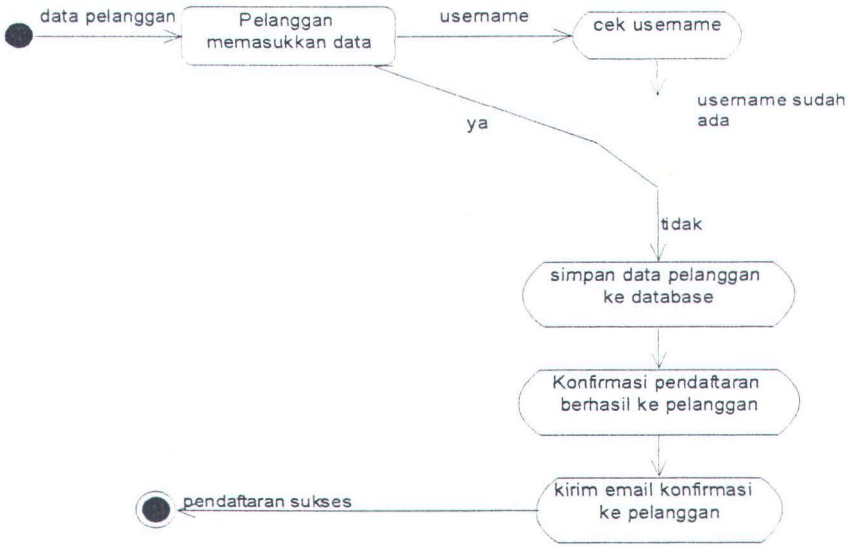
3.1.1.3 Diagram Use Case Pendaftaran Pelanggan

Use case pendaftaran pelanggan dirancang untuk menggambarkan proses pendaftaran calon pelanggan menjadi anggota komunitas aplikasi toko buku online BookStore.Inc. Calon pelanggan memasukkan informasi yang berguna bagi sistem seperti username, password, nama, alamat, nomor telepon, dan email. Username yang diberikan oleh pengguna merupakan suatu yang unik, oleh karenanya sistem perlu untuk melakukan pengecekan apakah username tersebut sudah ada dalam daftar database. Secara lebih detail, proses ini dapat dilihat pada gambar 3.5 dibawah ini.



Gambar 3.5. Diagram Use Case Pendaftaran Pelanggan

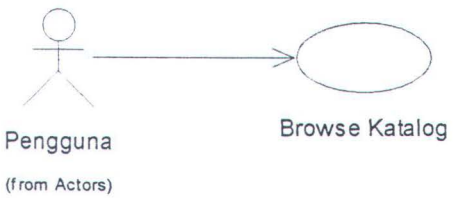
Selain diagram di atas, dibuat juga activity diagram untuk menjelaskan proses pendaftaran pelanggan seperti terlihat pada gambar 3.6. berikut ini.



Gambar 3.6. Activity Diagram Pendaftaran Pelanggan

3.1.1.4 Diagram Use Case Browse Katalog

Didalam use case ini semua pengguna termasuk calon pelanggan, pelanggan dan administrator dapat menjalankan proses ini.

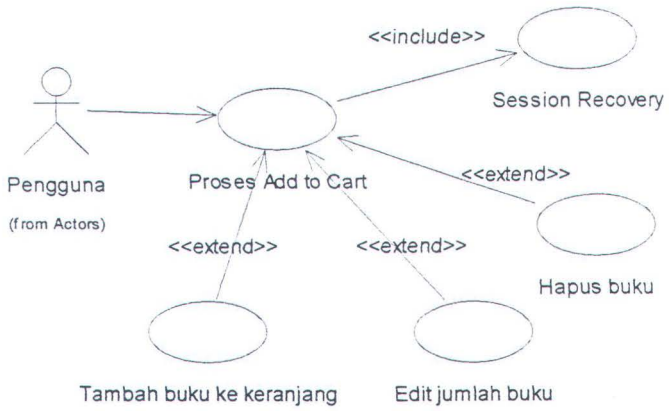


Gambar 3.7. Diagram Use Case Browse Katalog

3.1.1.5 Diagram Use Case Add to Cart

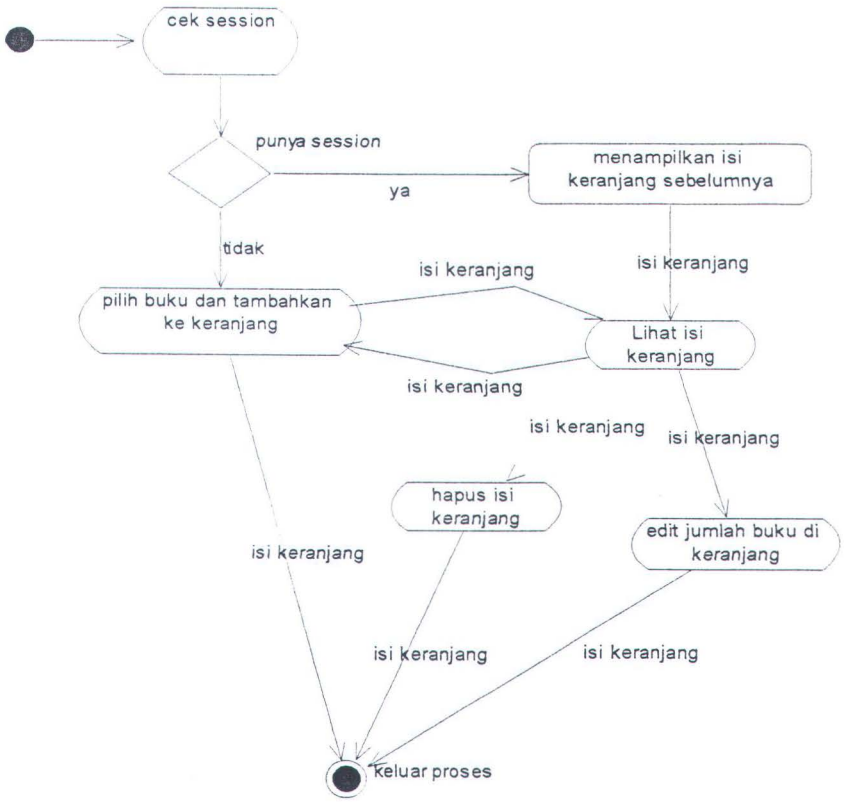
Didalam use case ini terdapat subproses-subproses menambah, menghapus, dan mengedit jumlah buku di dalam keranjang belanja, serta session recovery. Session recovery mengecek apakah pengguna pernah melakukan proses

add to cart sebelumnya. Jika sudah pernah maka isi keranjang belanja sebelumnya akan ditampilkan. Proses ini dapat dilihat pada gambar 3.8 di bawah ini .



Gambar 3.8. Diagram Use Case Add to Cart

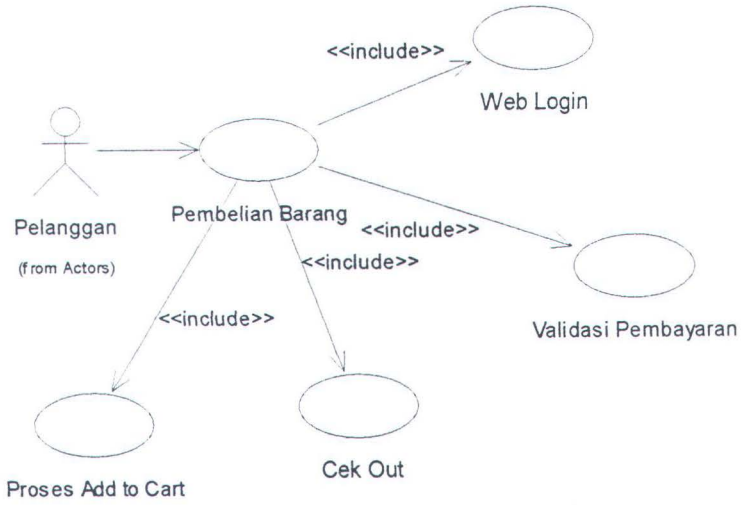
Aliran kejadian dari use case ini dapat digambarkan melalui activity diagram pada gambar3.9.



Gambar 3.9. Diagram Activity Add to Cart

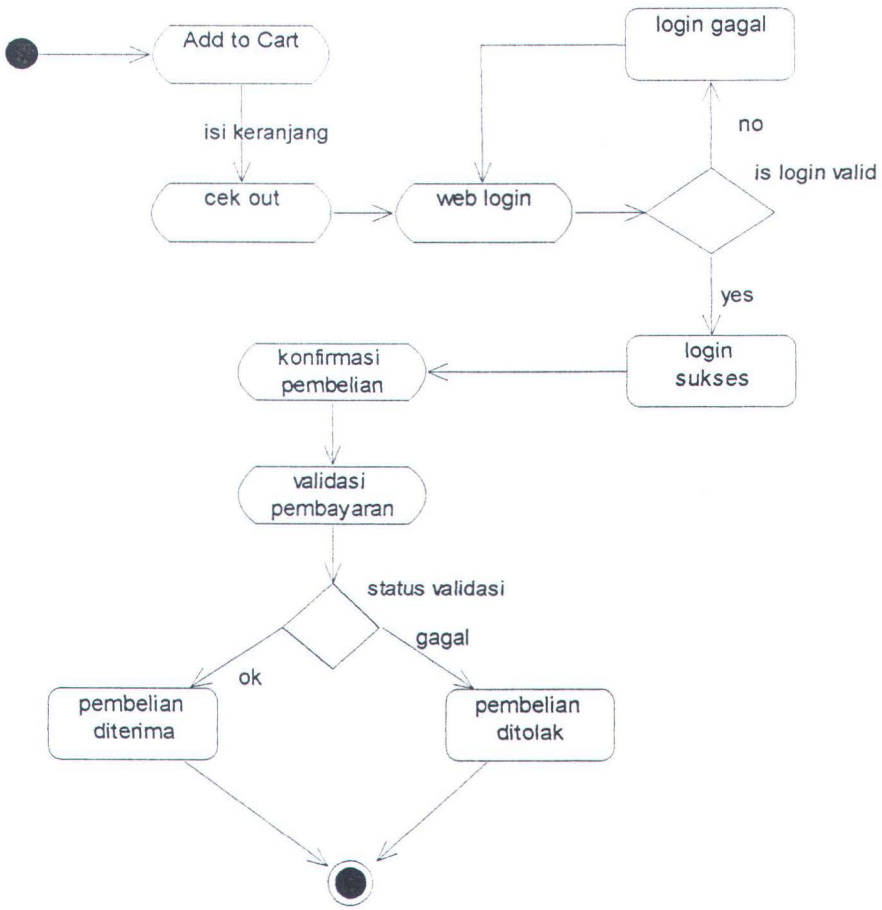
3.1.1.6 Diagram Use Case Pembelian Buku

Pada use case ini terdapat subproses-subproses yaitu add to cart, cek out, validasi pembayaran dan web login. Setelah pengguna memasukkan item buku kedalam keranjang, maka pengguna dapat menuju sub proses cek out. Untuk cek out pengguna harus mempunyai username dan password yang terdaftar pada aplikasi, untuk itu dijalankan sub proses web login. Setelah login selesai maka sub proses validasi pembayaran akan dijalankan. Secara lebih detail, dapat dilihat pada gambar 3.10. berikut ini.



Gambar 3.10. Diagram Use Case Pembelian Barang

Aliran kejadian dari use case ini dapat digambarkan melalui activity diagram pada gambar3.11.



Gambar 3.11. Diagram activity pembelian barang

3.1.3 MODEL OBJEK STATIS

Berdasarkan diagram use case yang telah dibuat sebelumnya maka kemudian dibuat model objek statis, yang diwujudkan dalam bentuk diagram class. Class yang dibuat dikelompokkan dalam tiga tipe class yaitu:

a. *Boundary class*, yaitu class yang menjembatani sistem dengan pengguna, seperti form, class ini terdapat pada user service.

b. *Control class*, yaitu class yang mengontrol class lain, berupa class-class yang berada pada business service.

c. *Entity class*, yaitu class menyimpan data-data yang dibutuhkan sistem beserta karakteristik data-data itu, barada pada data service.

3.1.3.1 Diagram Class Manajemen Data Buku

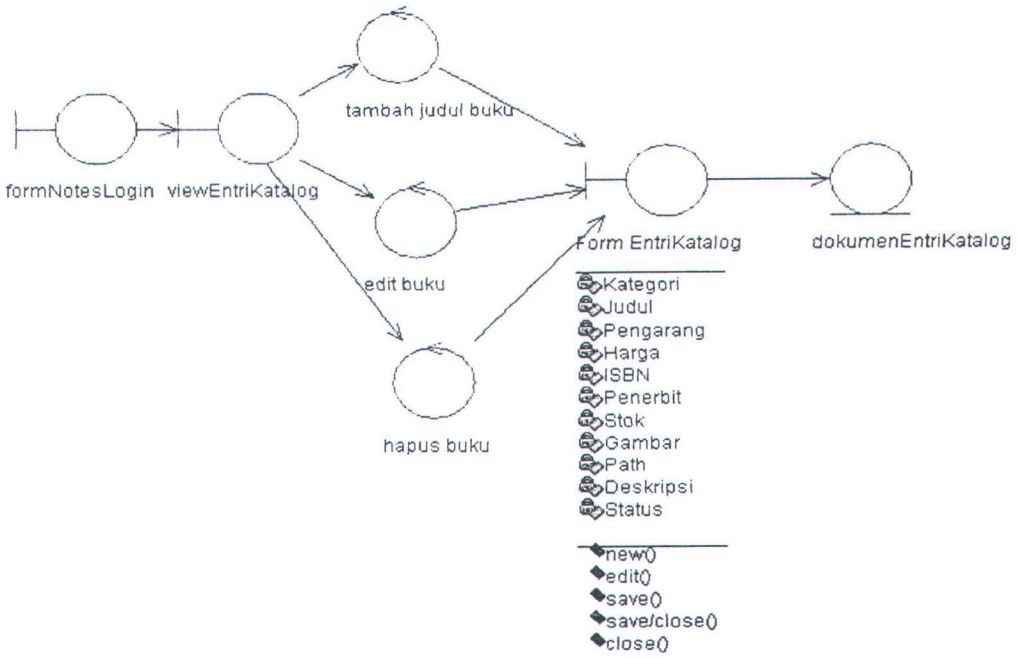
Berdasarkan diagram use case manajemen buku, dibuat rancangan class seperti tampak pada tabel di bawah ini.

Tabel 3.1. Daftar Class pada Use Case ManajemenData Buku

Nama Class	Kelompok	Keterangan
FormNotesLogin	Boundary Class	Antar muka login Notes
ViewEntriKatalog	Boundary Class	View data buku
FormEntriKatalog	Boundary Class	Antar muka input, edit data buku
Tambah buku	Control Class	Menambah buku
Edit buku	Control Class	Mengedit buku
Hapus buku	Control Class	Menghapus data buku
DokumenEntriKatalog	Entity	Dokumen data buku

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.12.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 3.12. Diagram Class pada Use Case Manajemen Data Buku

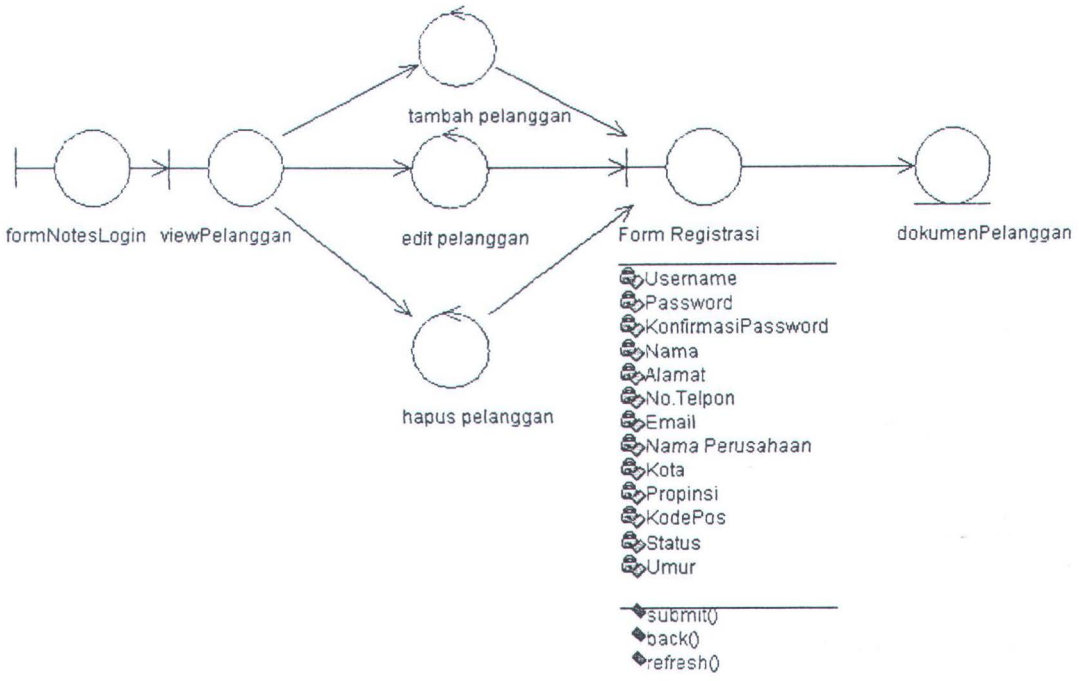
3.1.3.2 Diagram Class Manajemen Pelanggan

Berdasarkan diagram use case manajemen pengguna, dibuat rancangan class seperti tampak pada tabel 3.2. di bawah ini

Tabel 3.2. Daftar Class pada Use Case Manajemen Pelanggan

Nama Class	Kelompok	Keterangan
FormNotesLogin	Boundary Class	Antar muka login Notes
ViewPelanggan	Boundary Class	View data pelanggan
FormRegistrasi	Boundary Class	Antar muka input, edit data pelanggan
Tambah pelanggan	Control Class	Menambah pelanggan
Edit pelanggan	Control Class	Mengedit pelanggan
Hapus pelanggan	Control Class	Menghapus data pelanggan
Dokumen Pelanggan	Entity	Dokumen data pelanggan

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.13.



Gambar 3.13. Diagram Class pada Use Case Manajemen Pelanggan

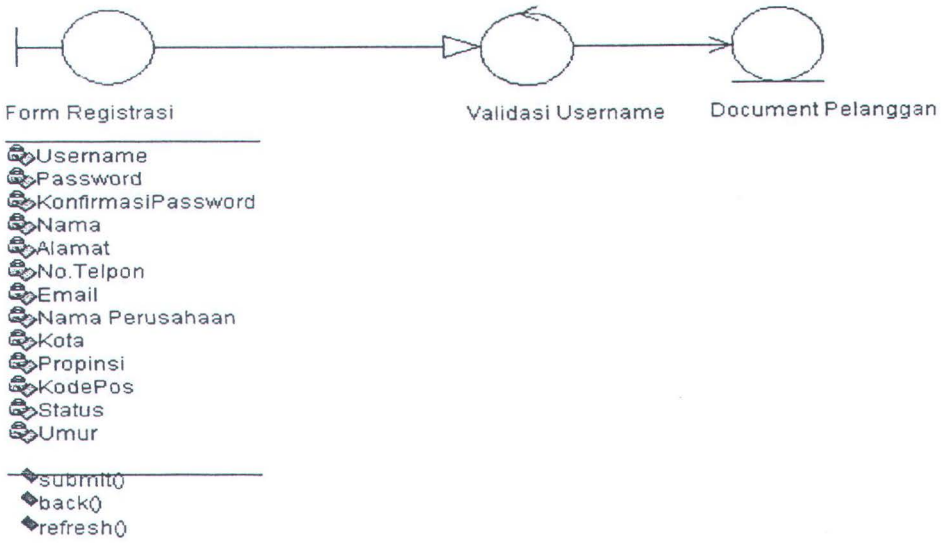
3.1.3.3 Diagram Class Pendaftaran Pelanggan

Berdasarkan diagram use case dan activity diagram pendaftaran pelanggan, dirancang beberapa class sebagai berikut.

Tabel 3.3. Daftar Class pada Use Case Pendaftaran Pelanggan

Nama Class	Kelompok	Keterangan
FormRegistrasi	Boundary Class	Form yang berfungsi sebagai antar muka pendaftaran
Validasi Username	Control Class	Class yang berfungsi untuk memvalidasi username dan data pengguna.
Document Pelanggan	Entity Class	Dokumen untuk menyimpan data pelanggan.

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.14. berikut ini.



Gambar 3.14. Diagram Class pada Use Case Pendaftaran Pelanggan

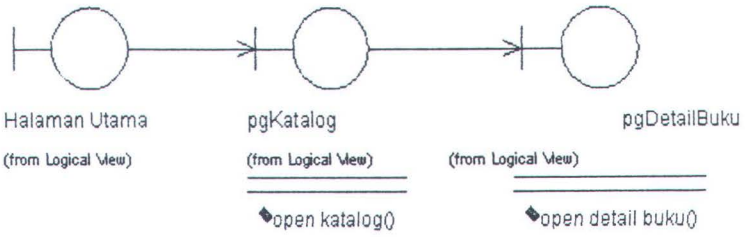
3.1.3.4 Diagram Class Browse Katalog

Berdasarkan dari diagram use case browse katalog , dirancang beberapa class sebagai berikut.

Tabel 3.4. Daftar Class pada Use Case Browse Katalog

Nama Class	Kelompok	Keterangan
HalamanUtama	Boundary Class	Form yang antar muka utama
PgKatalog	Boundary Class	Class yang berfungsi untuk memvalidasi username dan data pengguna.
PgDetailBuku	Boundary Class	Dokumen untuk menyimpan data pelanggan.

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.15. berikut ini.



Gambar 3.15. Diagram Class pada Use Case BrowseKatalog

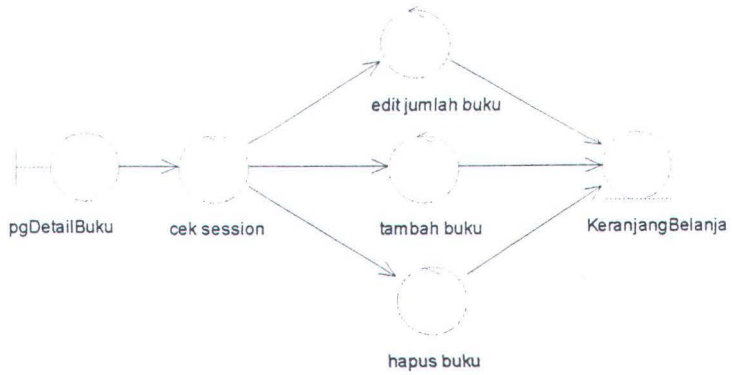
3.1.3.5 Diagram Class Add to Cart

Berdasarkan diagram use case dan activity diagram Add to Cart, dirancang beberapa class sebagai berikut.

Tabel 3.5. Daftar Class pada Use Case Add To Cart

Nama Class	Kelompok	Keterangan
PgDetailBuku	Boundary Class	Form yang berfungsi sebagai antar muka pendaftaran
Cek session	Control Class	Mengecek session pengguna, apakah keranjang sudah ada isinya.
Tambah buku	Control Class	Menambah buku ke keranjang
Hapus buku	Control Class	Menghapus buku dari keranjang
Edit jumlah buku	Control Class	Mengedit jumlah buku di keranjang
Keranjang Belanja	Entity	Menyimpan data keranjang belanja

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.16 berikut ini.



Gambar 3.16. Diagram Class Add to Cart

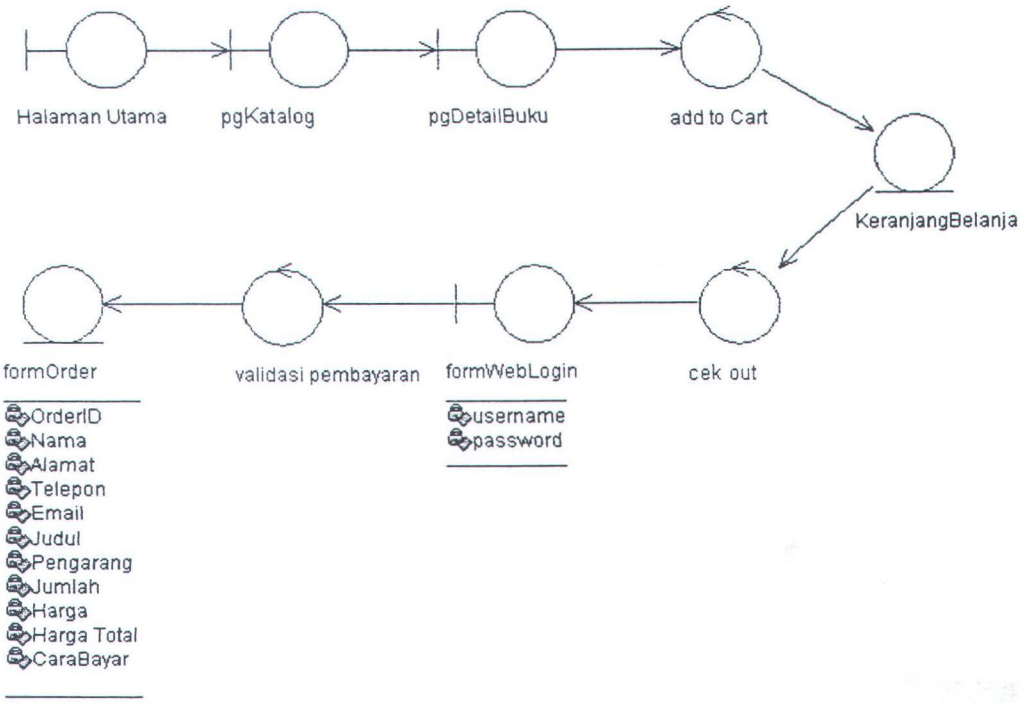
3.1.3.6 Diagram Class Pembelian Barang

Berdasarkan diagram use case dan activity diagram pembelian barang di atas, dirancang beberapa class berikut.

Tabel 3.6. Daftar Class pada Use Case Pembelian Barang

Nama Class	**Kelompok**	**Keterangan**
Halaman Utama	Boundary Class	Antar muka utama dari aplikasi Book Store.Inc
PgKatalog	Boundary Class	Halaman katalog, untuk mencari dan mem-*browse* buku yang diinginkan
PgDetailBuku	Boundary Class	Halaman detil dari buku yang dipilih
FormWebLogin	Boundary Class	Halaman web login
Add to Cart	Control Class	Melakukan proses penambahan, penghapusan, peng-update-an jumlah buku di keranjang belanja
Cek Out	Control Class	Proses cekout dari aplikasi
Validasi Pembayaran	Control Class	Melakukan proses validasi pembayaran
Keranjang Belanja	Entity Class	Menyimpan data keranjang belanja
FormOrder	Entity Class	Menyimpan data pelanggan dan daftar belanjanya

Bentuk diagram class dari use case ini adalah seperti pada gambar 3.17. berikut ini.



Gambar 3.17. Diagram Class pada Use Case Pembelian Barang

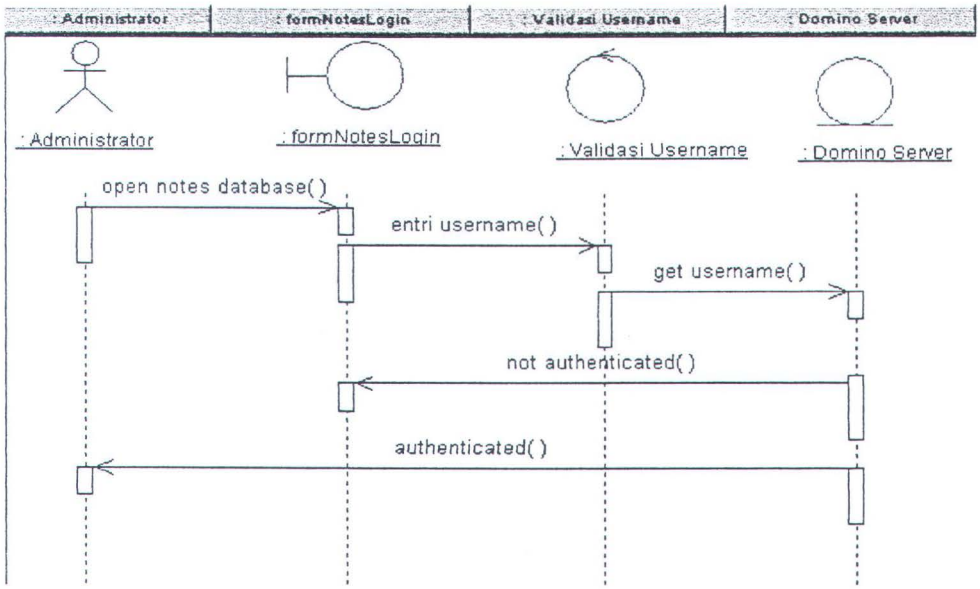
3.1.4 MODEL OBJEK DINAMIS

Dalam subbab ini dijelaskan interaksi yang terjadi antar objek/class, yang digambarkan dalam sequence diagram. Penjelasan masing-masing diagram dilakukan berdasarkan masing-masing use case seperti penjelasan pada model objek statis.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH – NOPEMBER

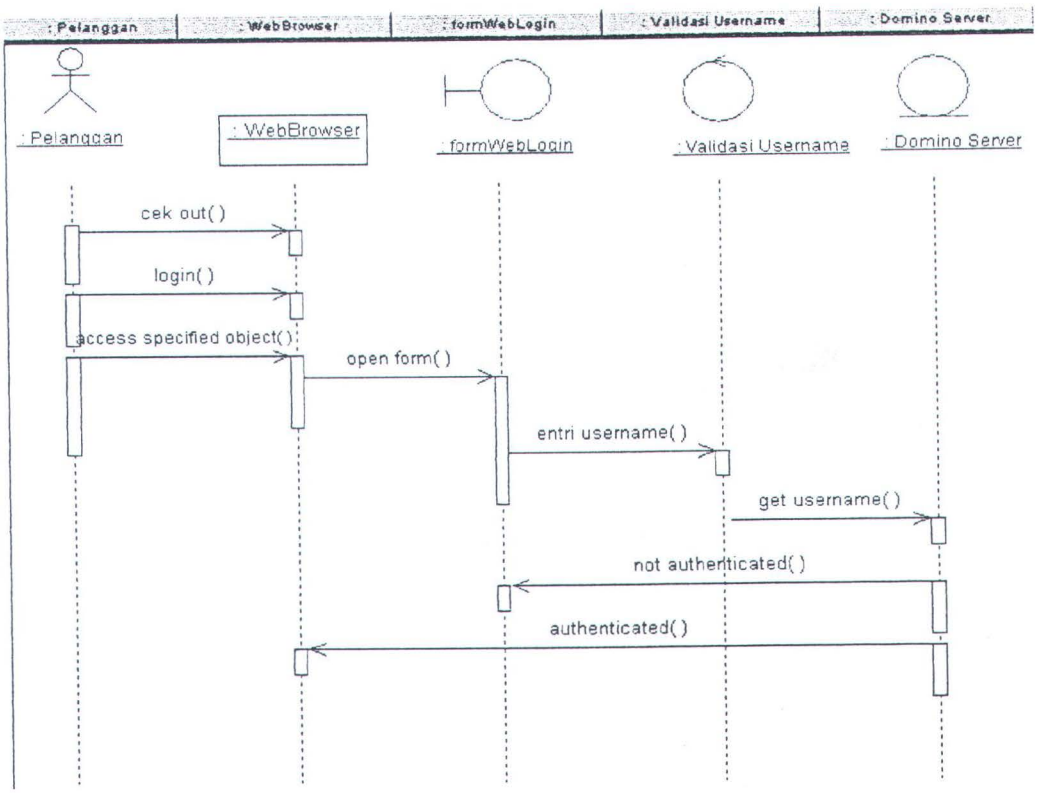
3.1.4.1 Diagram Sequence Login

Aplikasi BookStore.Inc mempunyai dua buah sistem login yaitu login Notes dan login web. Interaksi yang terjadi antar obyek/class pada saat login Notes dapat dilihat pada diagram sequence pada gambar 3.18 berikut ini.



Gambar 3.18.. Diagram SequenceLogin Notes

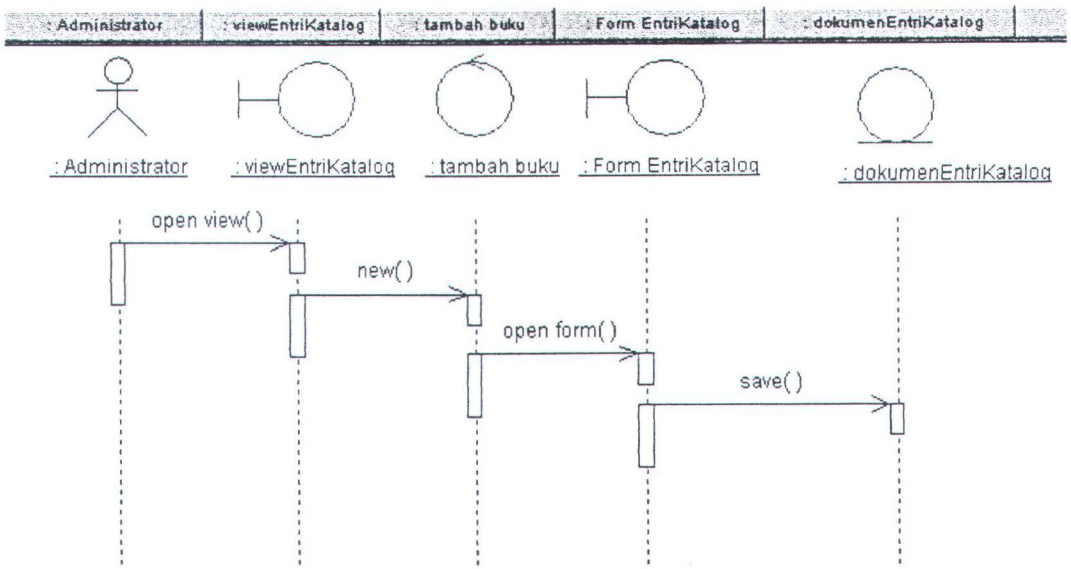
Sedangkan interaksi yang terjadi antar obyek/class pada saat login web dapat dilihat pada diagram sequence pada gambar 3.19.



Gambar 3.19. Diagram SequenceLogin Web

3.1.4.2 Diagram Sequence Manajemen Data Buku

Interaksi yang terjadi antar abjek/class pada use case manajemen data buku dapat dilihat pada sequence diagram pada gambar 3.20.

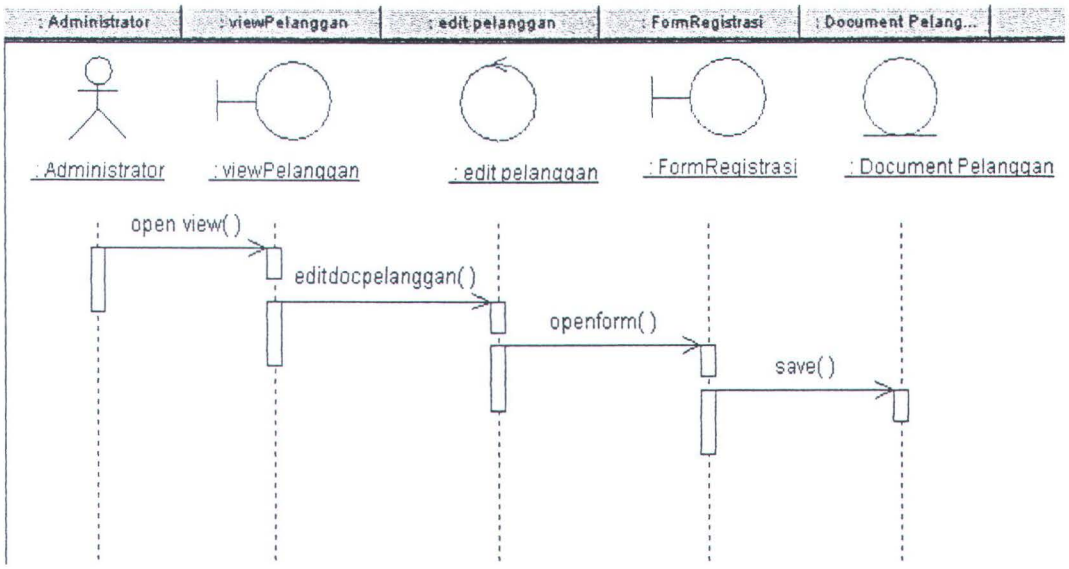


Gambar 3.20. Diagram Sequence Manajemen Data Buku

Gambar di atas merupakan salah satu dari interaksi yang terdapat pada use case ini yaitu pada aktifitas menambah data buku. Administrator melakukan manajemen buku seperti menambah, menghapus dan merubah data buku.

3.1.4.3 Diagram Sequence Manajemen Pelanggan

Interaksi yang terjadi antar abjek/class pada use case manajemen pelanggan dapat dilihat pada sequence diagram berikut ini.

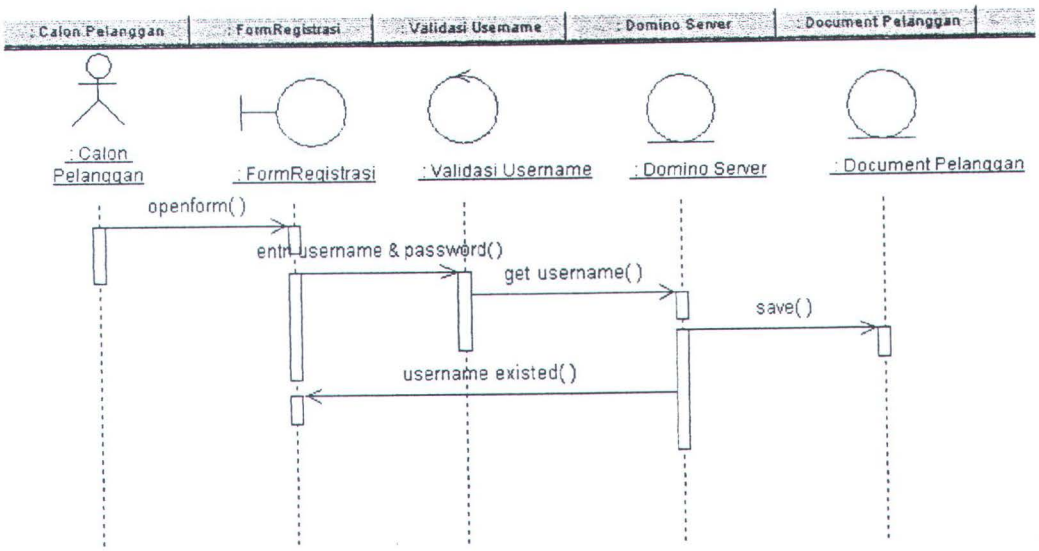


Gambar 3.21. Diagram Sequence Manajemen Pelanggan

Gambar diatas. merupakan salah satu dari interaksi yang terdapat pada use case ini yaitu pada aktifitas mengedit data pengguna. Administrator dapat melakukan manajemen pelanggan seperti menambah, menghapus dan merubah data pelanggan.

3.1.4.4 Diagram Sequence Pendaftaran Pelanggan

Interaksi yang terjadi antar abjek/class pada use case ini dapat dilihat pada sequence diagram pada gambar 3.22.

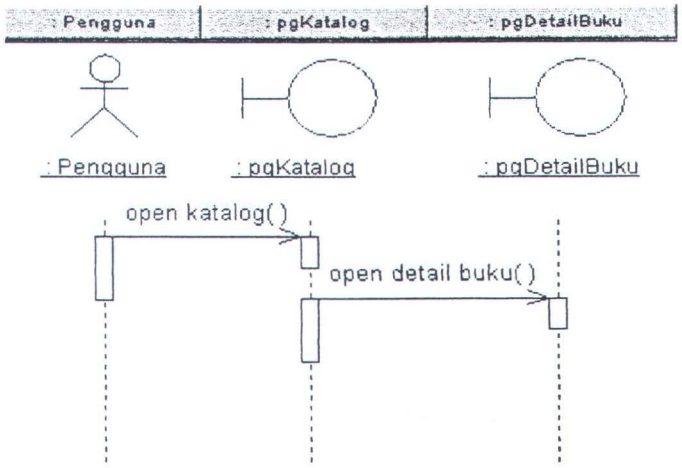


Gambar 3.22. Diagram Sequence Pendaftaran Pelanggan

Dari gambar 3.22 di atas dapat dilihat bahwa calon pelanggan harus memasukkan informasi yang berguna bagi sistem seperti nama, alamat, username, dan password, dan email. Username yang diberikan oleh pengguna merupakan suatu yang unik, oleh karenanya sistem perlu untuk melakukan pengecekan apakah username tersebut sudah ada di database. Jika ada maka pengguna harus memasukkan username lain. Jika tidak ada maka data pelanggan ditambahkan ke database dan disimpan dalam dokumen pelanggan.

3.1.4.5 Diagram Sequence Browse Katalog

Interaksi yang terjadi antar abjek/class pada use case browse katalog dapat dilihat pada diagram sequence berikut ini.



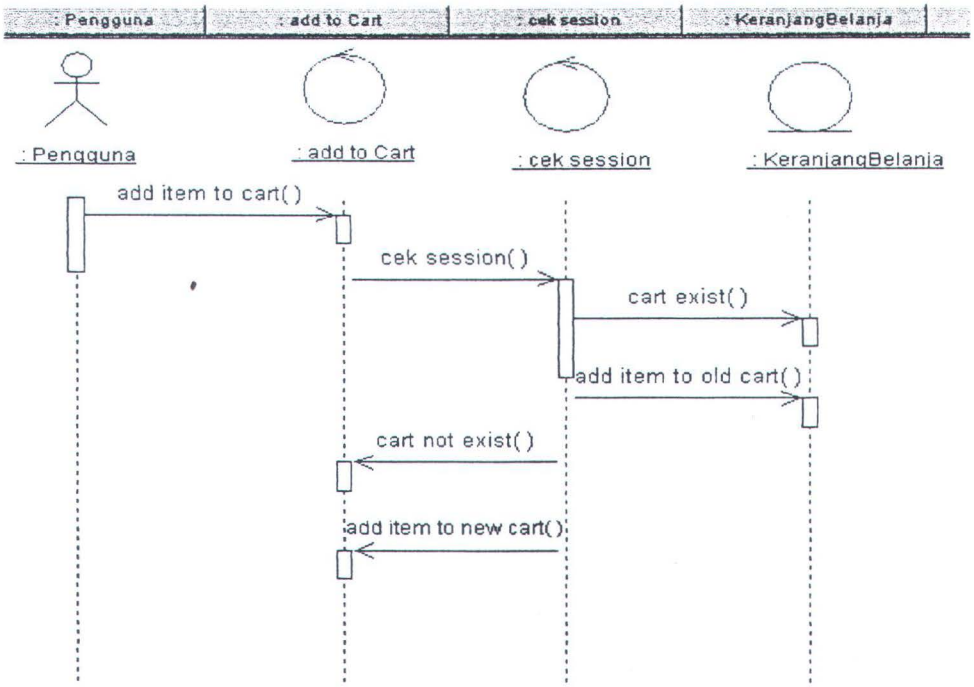
Gambar 3.23. Diagram Sequence Browse Katalog

3.1.4.6 Diagram Sequence Add to Cart

Interaksi yang terjadi antar abjek/class pada use case add to cart dapat dilihat pada diagram sequence berikut ini. Proses add to cart terdiri dari empat sub proses yaitu cek session, tambah item, edit jumlah item, dan hapus item. Sub proses cek session adalah proses yang dijalankan pertama kali. Sub proses ini mengambil data keranjang pengguna dari session sebelumnya . Jika tidak ditemukan maka sistem akan membuat keranjang baru.

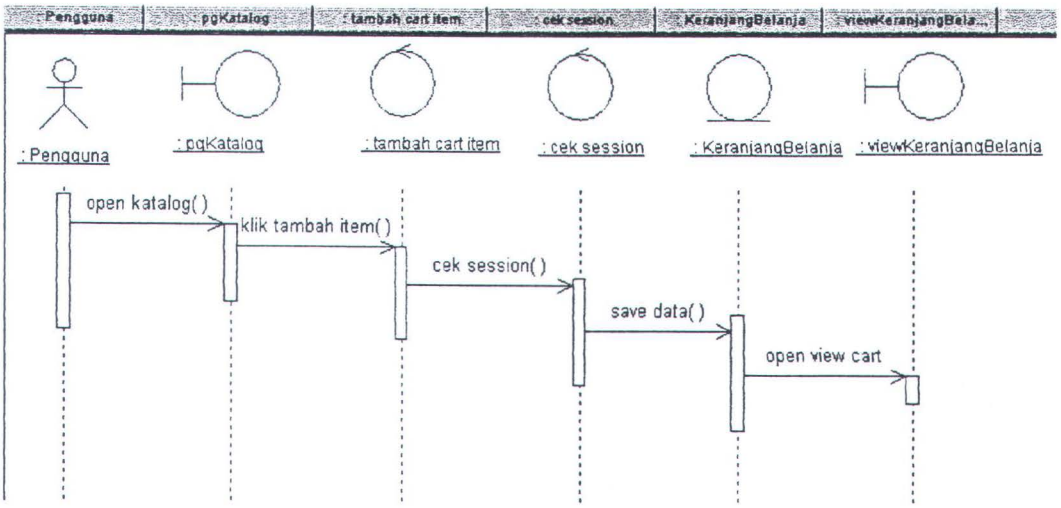
Diagram sequence pada gambar 3.24 berikut ini menggambarkan sub proses cek session.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



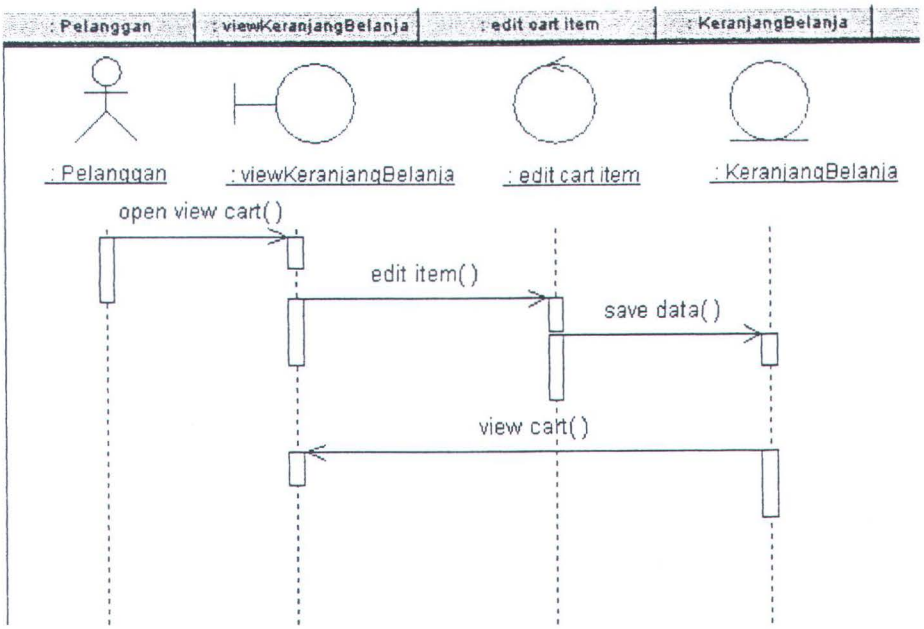
Gambar 3.24. Diagram Sequence Add to Cart sub proses Cek Session

Dibawah ini adalah diagram sequence untuk menambah item ke keranjang belanja.



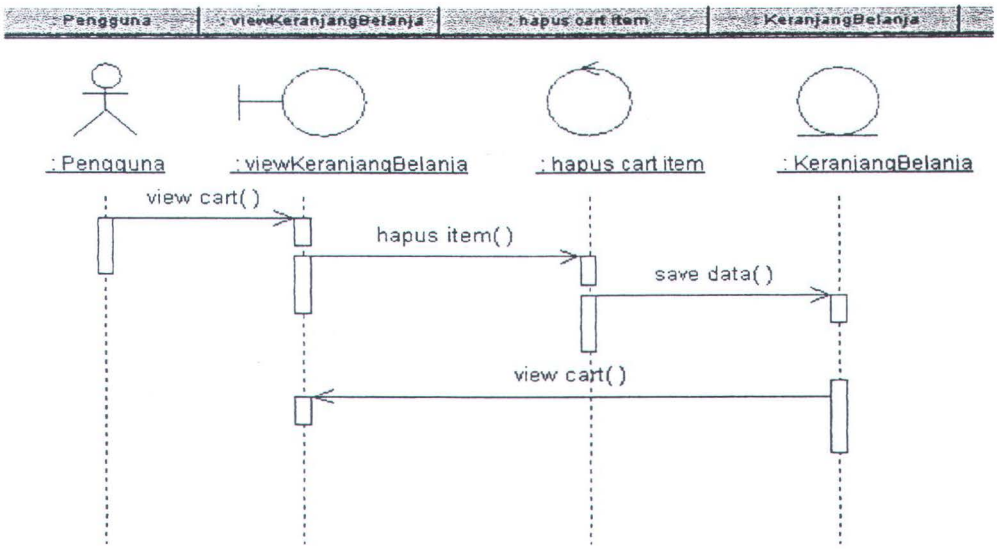
Gambar 3.25. Diagram Sequence Add to Cart sub proses Tambah Item

Dibawah ini adalah diagram sequence untuk mengedit item di keranjang belanja.



Gambar 3.26. Diagram Sequence Add to Cart sub proses Edit Item Keranjang

Berikut ini adalah diagram sequence untuk menghapus item di keranjang belanja.



Gambar 3.27. Diagram Sequence Add to Cart sub proses Hapus Item

3.1.4.7 Diagram Sequence Pembelian Barang

Interaksi yang terjadi antar abjek/class pada use case ini dapat dilihat dari masing-masing sub proses yang terjadi yaitu add to cart, cek out, web login, dan validasi pembayaran.

Sub proses pertama yang harus dijalankan adalah add to cart, kemudian dilanjutkan dengan sub proses cek out, yang akan menjalankan sub proses web login dan kemudian diakhiri dengan sub proses validasi pembayaran.

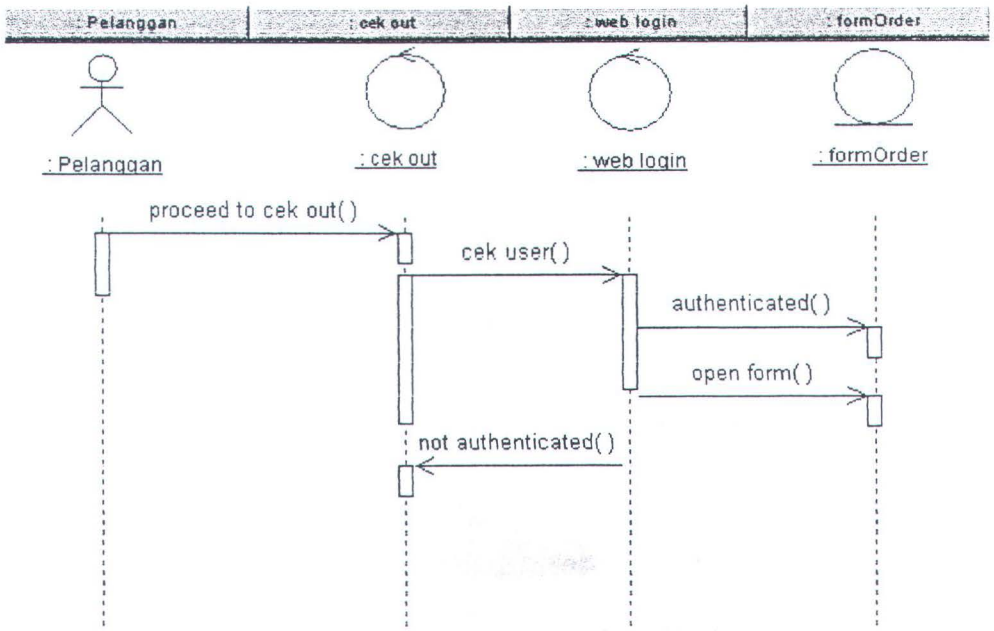
Sub proses Add to Cart dapat dilihat pada gambar 3.24, gambar 3.25, dan gambar 3.26 dan 3.27. Sedangkan sub proses web login dapat dilihat pada gambar 3.19.

Di dalam sub bab ini akan digambarkan diagram sequence untuk sub proses cek out dan validasi pembayaran.

- **Diagram Sequence Sub Proses Cek Out**

Cek out merupakan subproses yang dijalankan oleh pengguna jika pengguna sudah memasukkan item ke dalam keranjang belanja dan ingin melakukan pembayaran. Semua barang belanjaan yang terdapat di dalam keranjang belanja dianggap sebagai barang yang telah dipesan dan akan dihitung berapa jumlah tagihan yang harus dibayar. Gambar 3.28. menjelaskan tentang subproses tersebut di atas.

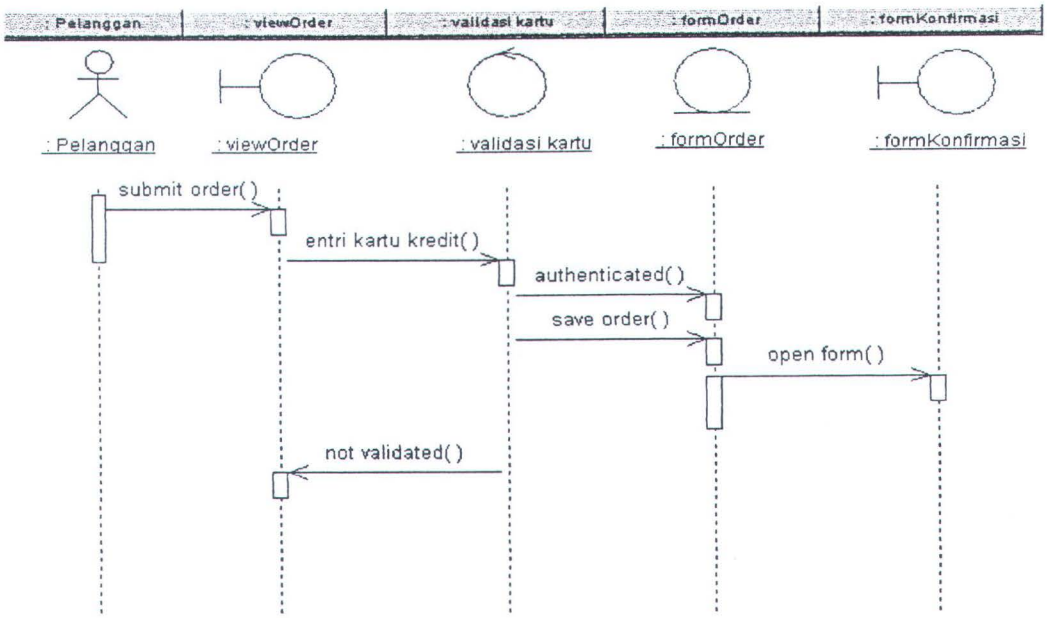
MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 3.28. Diagram Sequence Pembelian Buku sub proses Cek Out

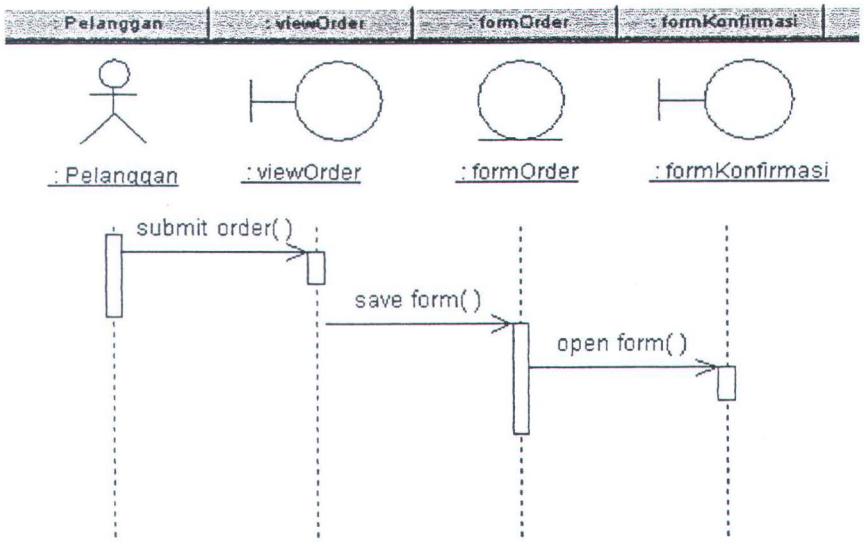
- **Diagram Sequence Sub Proses Validasi Pembayaran**

Validasi Pembayaran merupakan subproses terakhir dalam proses pembelian buku. Di dalam sistem BookStore.Inc cara pembayaran dibagi menjadi dua, yaitu pembayaran dengan kartu kredit dan pembayaran lewat transfer bank. Untuk proses validasi pembayaran dengan kartu kredit di dalam implementasinya nanti hanyalah simulasi dari proses validasi sebenarnya. Gambar 3.29. menjelaskan tentang subproses validasi pembayaran lewat kartu kredit.



Gambar 3.29. Diagram Sequence Pembelian Buku sub proses Validasi Pembayaran dengan Kartu Kredit

Gambar 3.30. berikut ini menjelaskan tentang subproses validasi pembayaran lewat transfer bank.



Gambar 3.30. Diagram Sequence Pembelian Buku sub proses Validasi Pembayaran dengan Transfer Bank

BAB IV
PEMBUATAN APLIKASI

BAB IV

PEMBUATAN APLIKASI

Pada bab ini akan dijelaskan mengenai implementasi dari perancangan aplikasi toko buku online BookStore.Inc . Untuk membuat aplikasi ini digunakan tool Lotus Domino Designer Release 5 . Database yang dibuat disimpan di direktori server yaitu Lotus/Domino/Data. Hal ini agar database dapat diakses dari luar mesin (workstation) dimana aplikasi ini berada, baik dengan menggunakan Notes Client maupun dengan web browser.

Untuk implementasi database BookStore.Inc, dibuat suatu database baru dengan nama TokoBuku.nsf dari Lotus Domino Designer. Untuk keperluan keamanan maka pada ACL (*Access Control List*) database ini diatur user mana saja yang berhak mengakses dan tingkat akses yang diperbolehkan untuk user itu.

4.1 Setting ACL (Acces Control List)

Pada database ini ACL yang dibuat adalah sebagai berikut :

- **Role**

  [SiteManager] Role ini diberikan kepada user administrator. Role ini diperlukan untuk menjalankan form EntriKatalog dan melihat dan mengedit status dan tanggal pemesanan pada form Order.

- **User**

  Anonymous User ini adalah semua pengguna yang mengakses situs BookStore.Inc. Tingkat akses yang diberikan adalah

	Author, artinya pengguna dapat membaca dokumen dan menulis dokumen yang telah ditentukan untuk pengguna (dalam hal ini form Pendaftaran). User ini dapat melihat-lihat katalog dan melakukan pendaftaran, memasukkan item ke keranjang belanja, namun tidak dapat melakukan transaksi cek out untuk membeli barang.
WebUserGroup	User yang bertipe grup. Artinya semua pengguna yang sudah terdaftar menjadi anggota komunitas BookStore.Inc termasuk kedalam kelompok user ini. Grup ini juga memiliki tingkat akses Author. Grup ini dapat melakukan semua proses yang dapat dilakukan oleh user anonymous ditambah dengan proses cek out untuk pembelian barang.
Admin	User ini memiliki tingkat akses tertinggi yaitu Manager. User ini juga diberikan role [Site Manager]. User ini dapat mengubah data dan rancangan dari aplikasi ini.

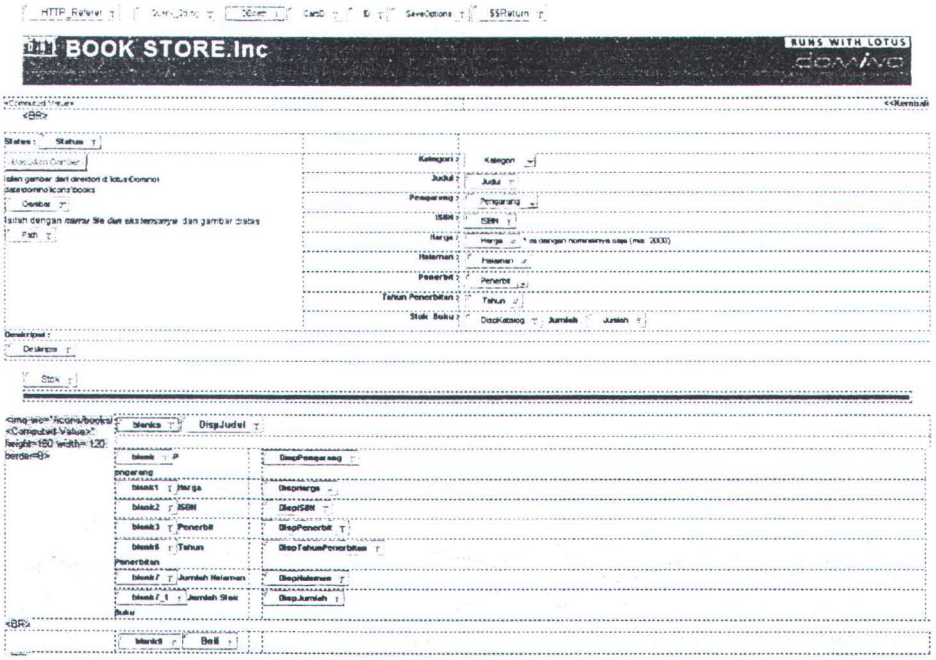
4.2 Form

Inti dari aplikasi ini ada pada form. Form dapat digunakan untuk keperluan pengisian data dan juga untuk mengatur tampilan aplikasi pada web browser.

4.2.1 Form DisplayKatalog

Form ini digunakan untuk mengisi data buku dan menampilkan informasi buku kepada pengguna. Form ini dapat dijalankan dengan menggunakan Notes Client ataupun dengan web browser. Form ini hanya dapat diakses oleh

Administrator. Pada gambar 4.1 berikut ini dapat dilihat tampilan dari rancangan form DisplayKatalog :



Gambar 4.1. Form DisplayKatalog

Form ini dibagi menjadi dua tampilan yang dibatasi oleh garis melintang ditengah form. Bagian atas adalah tampilan untuk administrator dalam menambah dan mengedit buku. Sedangkan bagian bawah adalah tampilan untuk pengguna di web browser. Bagian atas tidak akan tampil bila yang membuka adalah pengguna biasa. Hal ini bisa dilakukan dengan memanfaatkan fasilitas Hide yang telah disediakan oleh Domino Designer.

Form bagian atas terdiri dari field DBPath untuk mencari path dari database tokobuku.nsf, field ID untuk menyimpan ID dari dokumen, field Status untuk mengeset status dari buku itu di web, apakah termasuk buku baru atau lama, field Kategori untuk mengeset kategori dari buku, field Judul untuk mengisi judul buku, ISBN untuk mengisi ISBN, field Harga untuk mengisi harga buku, field

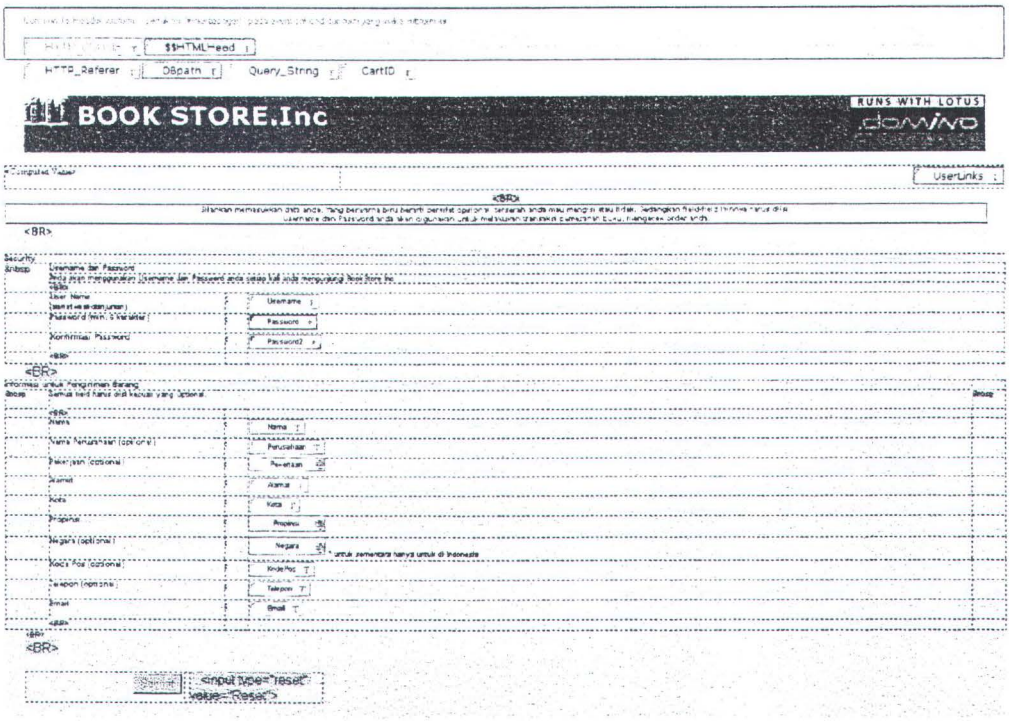
Penerbit untuk mengisi nama dari penerbit buku, field Gambar untuk menyimpan gambar buku, field Path untuk menyimpan path/letak dari file gambar buku, agar lebih terstruktur maka semua file gambar buku diletakkan di direktori domino/data/domino/icons/books, field Stok sebagai keterangan stok buku yang berisi nilai ada dan habis yang isinya tergantung kepada jumlah buku, field Jumlah Buku yang menyimpan informasi jumlah buku yang tersedia, dan field Deskripsi untuk pengisian deskripsi lengkap dari buku, field Options untuk mengatur apakah form bisa disimpan atau tidak

Sedangkan pada bagian bawah form yang diperuntukkan untuk pengguna di web, pengguna dapat melakukan proses penambahan item kedalam keranjang belanjanya dengan menekan field Beli, yang pada intinya merupakan tombol untuk melakukan pembelian. Apabila buku yang diinginkan sedang habis maka pengguna dapat memasukkan alamat email ke field yang telah disediakan pada form ini. Jika stok buku sudah ada maka sistem memberitahu pengguna lewat email bahwa buku sudah tersedia. Proses ini dinamakan dengan proses notifikasi.

Sedangkan action atau operasi yang ada pada form ini adalah New untuk membuat data buku baru, Edit untuk mengedit data buku, Save untuk menyimpan data buku, Save/Close untuk menyimpan dan keluar dari aplikasi form EntriKatalog, Close untuk keluar dari aplikasi EntriKatalog.

4.2.2 Form Registrasi

Form ini dgunakan oleh pengguna untuk mendaftarkan diri menjadi anggota komunitas toko buku online BookStore.Inc. Gambar 4.2 berikut ini adalah sebagian tampilan dari rancangan form Registrasi.



Gambar 4.1. Form Registrasi

Form ini mempunyai field–field Username untuk pengisian usernama, password untuk pengisian password, password2 untuk konfirmasi password, nama untuk mengisi nama pengguna, perusahaan untuk mengisi nama perusahaan, alamat untuk mengisi alamat dari pengguna, kota untuk mengisi kota dari pengguna, propinsi untuk mengisi propinsi dari pengguna, negara untuk mengisi negara dari pengguna, kodepos untuk mengisi kode pos dari alamat pengguna, telepon untuk mengisi nomer telepon dari pengguna, email untuk pengisian alamat email dari pengguna, pekerjaan untuk pengisian pekerjaan pengguna, dan umur untuk pengisian umur pengguna.

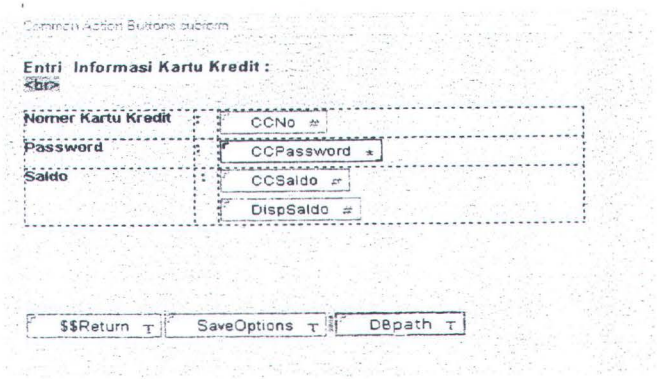
Form ini mempunyai dua action, yaitu submit untuk memproses data pengguna dan reset untuk mengosongkan field-field yang diisi oleh pengguna.

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Ada beberapa field yang harus diisi oleh pengguna yaitu field username, password, password2, nama, alamat, kota, propinsi dan email. Data ini diperlukan untuk pengiriman barang dan email konfirmasi kepada pengguna. Selain field-field diatas sifatnya adalah opsional.

4.2.3 Form EntriKartuKredit

Untuk keperluan simulasi validasi kartu kredit, maka dibuat form untuk pengisian data kartu kredit yang dimiliki oleh pelanggan. Gambar 4.3 dibawah ini adalah tampilan dari rancangan form EntriKartuKredit.

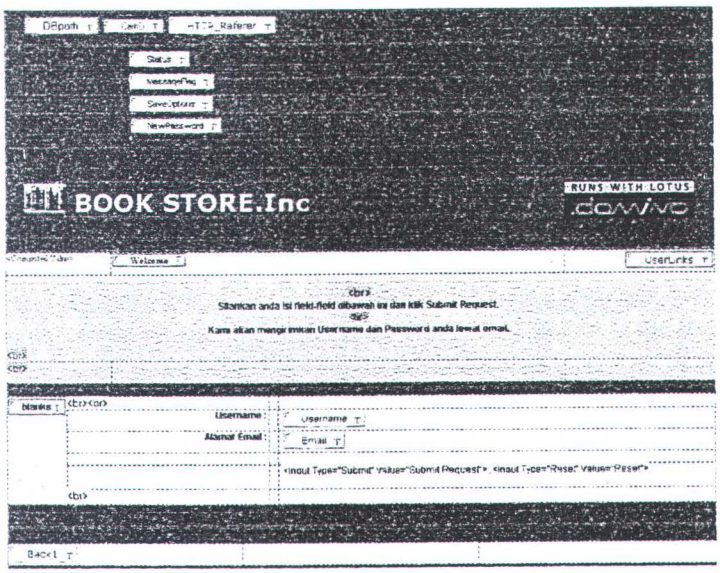


Gambar 4.3. Form EntriKartuKredit

Form ini sebagaimana juga form EntriKatalog dan form Registrasi menggunakan subform CommonActionButtons yang berisi *action*/perintah yang bisa dijalankan pada form yang menggunakan subform ini. Action yang ada pada subform CommonActionButtons yaitu New untuk membuat dokumen baru, Edit untuk mengedit data dokumen, Save untuk menyimpan dokumen, Save/Close untuk menyimpan dan menutup form, dan Close untuk menutup form.

4.2.4 Form LupaPassword

Form ini diperuntukkan bagi pelanggan yang tidak ingat password dari usernamenya. Gambar 4.4 dibawah ini adalah tampilan dari rancangan form LupaPassword.



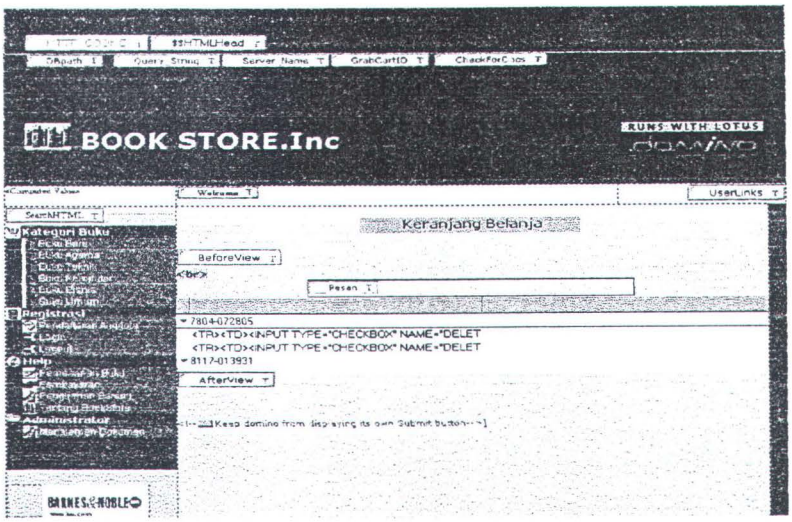
Gambar 4.4. Form LupaPassword

Form ini memiliki dua field yang harus diisi oleh pelanggan, yaitu username dan alamat email dari pelanggan yang telah didaftarkan di database BookStore.Inc. Sistem akan mengecek kebenaran dari kombinasi username dan alamat email yang telah terdaftar di database dengan yang diisikan pelanggan, jika cocok maka sistem akan menampilkan username dan password kepada pelanggan, serta mengirimkan email konfirmasi kepada pelanggan.

4.2.5 Form KeranjangBelanja

Form ini adalah tempat bagi pengguna untuk melihat isi keranjang belanja, dan mengedit isi dari keranjang belanjanya, serta untuk melakukan proses cek out

dari aplikasi. Gambar 4.5 dibawah ini adalah sebagian dari tampilan rancangan form KeranjangBelanja.

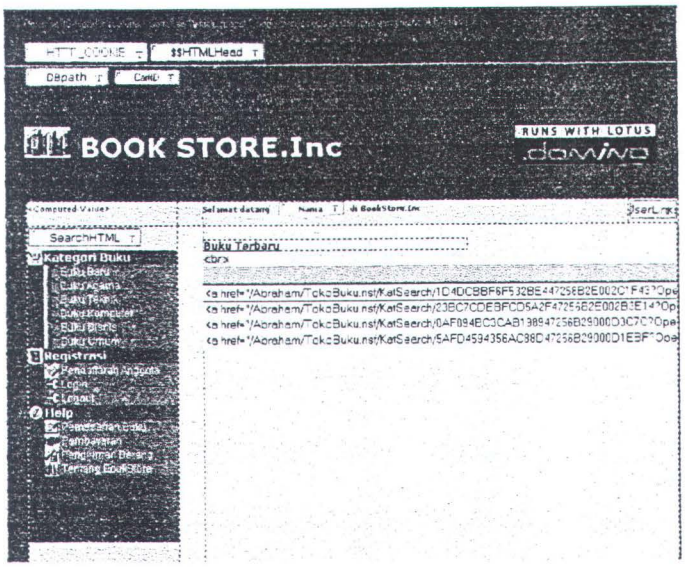


Gambar 4.5. Form Keranjang Belanja

Didalam form ini terdapat outline untuk navigasi aplikasi, *shared field* SearchHtml untuk melakukan pencarian buku, dan sebuah *embedded view* dengan *shared field* BeforeView ,field Pesan, dan *shared field* AfterView untuk menampilkan isi dari keranjang belanja pengguna..

4.2.7 Form HalamanUtama

Form ini adalah tampilan utama dan tampilan *default* yang akan dilihat pengguna apabila mengakses situs ini.Gambar 4.6 berikut ini adalah sebagian tampilan dari rancangan form HalamanUtama.



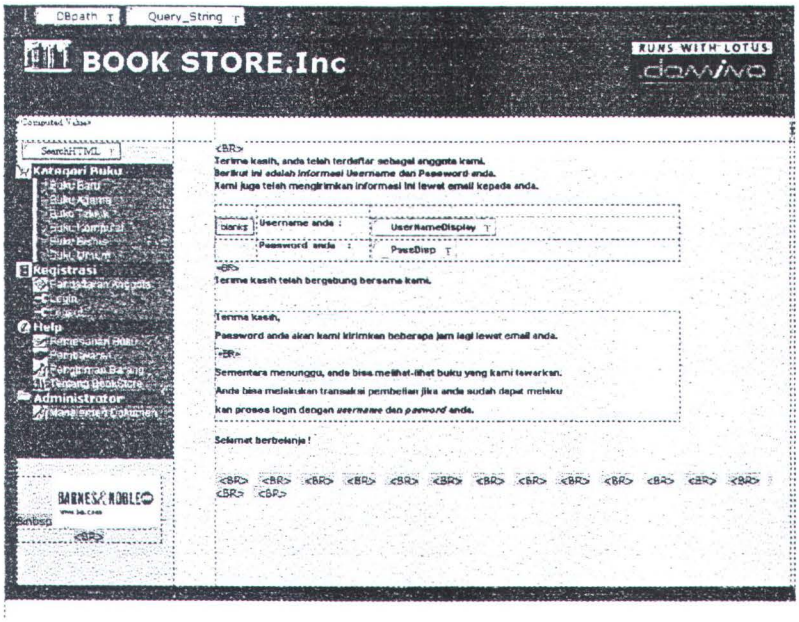
Gambar 4.6. Form HalamanUtama

Di dalam form ini terdapat *embedded view* yang akan menampilkan buku-buku terbaru yang dipunyai oleh situs ini.

4.2.8 Form Konfirmasi

Form ini adalah merupakan halaman konfirmasi yang akan ditampilkan setelah pengguna melakukan proses pendaftaran atau setelah pengguna menjalankan form LupaPassword. Gambar 4.7 berikut ini adalah sebagian tampilan dari rancangan form Konfirmasi.

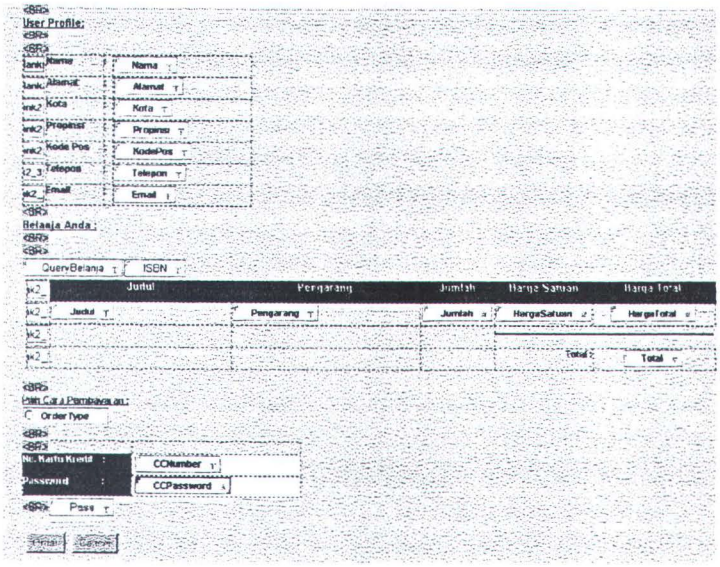
MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 4.7. Form Konfirmasi

4.2.9 Form Order

Form ini digunakan untuk menyimpan informasi pesanan(*order*) yang dilakukan oleh pelanggan. Gambar 4.8 dibawah ini adalah sebagian tampilan dari rancangan form Order.

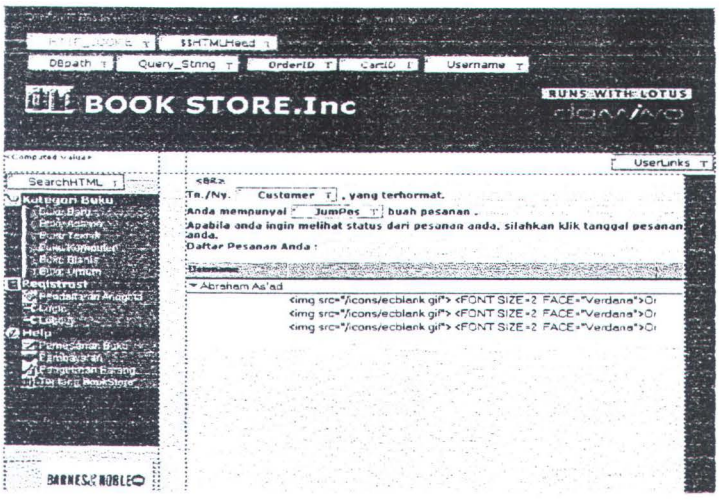


Gambar 4.8. Form Order

Pada form ini ditampilkan data pelanggan dan isi dari keranjang belanja pengguna. Form ini adalah semacam kuitansi pembelian yang juga akan dikirim kepada pengguna lewat email. Di dalam form inilah dilakukan proses validasi pembayaran. Pembayaran dapat dilakukan dengan kartu kredit atau transfer bank. Pelanggan yang memilih pembayaran dengan menggunakan kartu kredit harus melalui proses validasi kartu kredit. Didalam aplikasi ini disediakan simulasi validasi kartu kredit pelanggan. Pelanggan memasukkan nomer kartu kredit dan password, kemudian sistem akan mengecek kebenaran data kartu kredit yang diisikan pelanggan dan juga mengecek saldo yang ada pada kartu kredit pelanggan.

4.2.10 Form OrderStatusConfirm

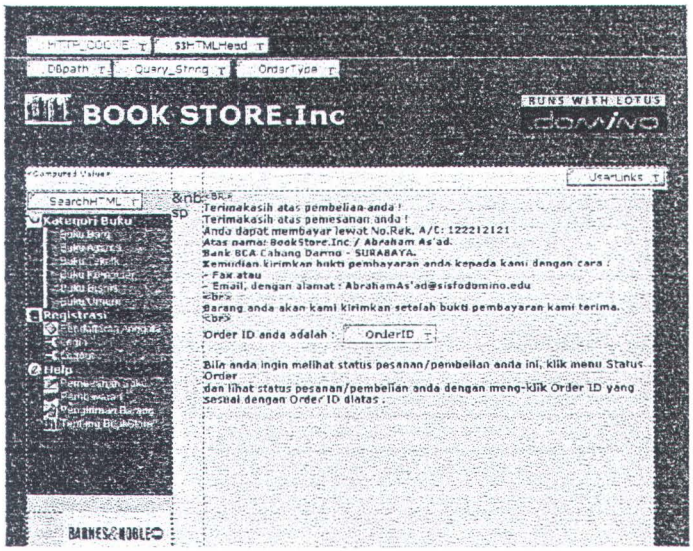
Form ini menampilkan pesanan yang pernah dibuat oleh pelanggan. Untuk mengakses form ini pelanggan harus memasukkan username dan password yang dipunyai pelanggan. Gambar 4.9 dibawah ini adalah sebagian tampilan dari rancangan form OrderStatusConfirm.



Gambar 4.9. Form OrderStatusConfirm

4.2.11 Form OrderThanks

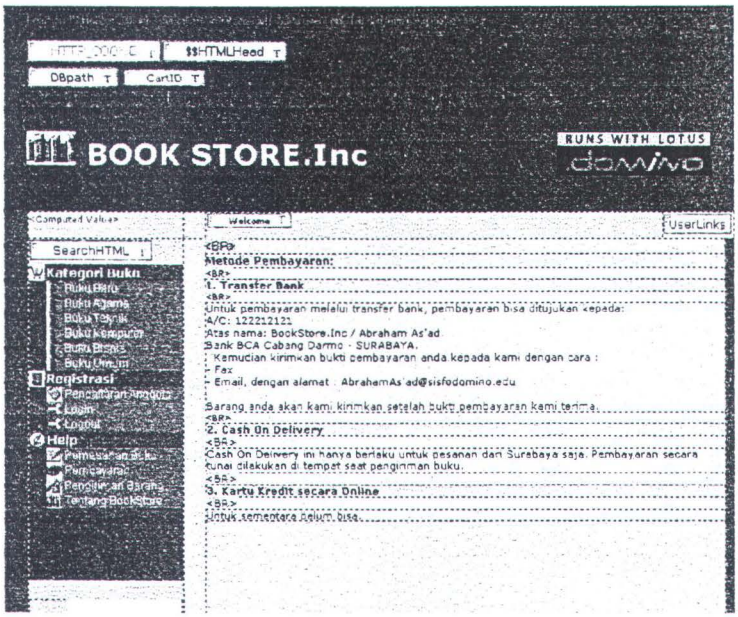
Form ini adalah merupakan halaman konfirmasi yang akan ditampilkan setelah pengguna melakukan proses pembelian di situs ini. Gambar 4.10 dibawah ini adalah sebagian tampilan dari rancangan form OrderThanks.



Gambar 4.10. Form OrderThanks

4.2.12 Form CaraPembayaran

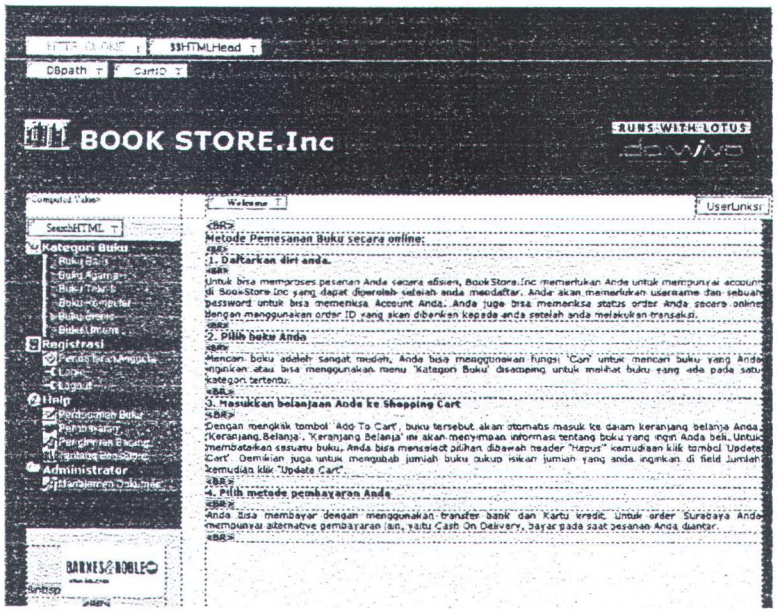
Form ini menampilkan informasi cara pembayaran yang dapat dilakukan di situs ini . Gambar 4.11 berikut ini adalah sebagian tampilan dari rancangan form CaraPembayaran.



Gambar 4.11. Form CaraPembayaran

4.2.13 Form CaraPemesanan

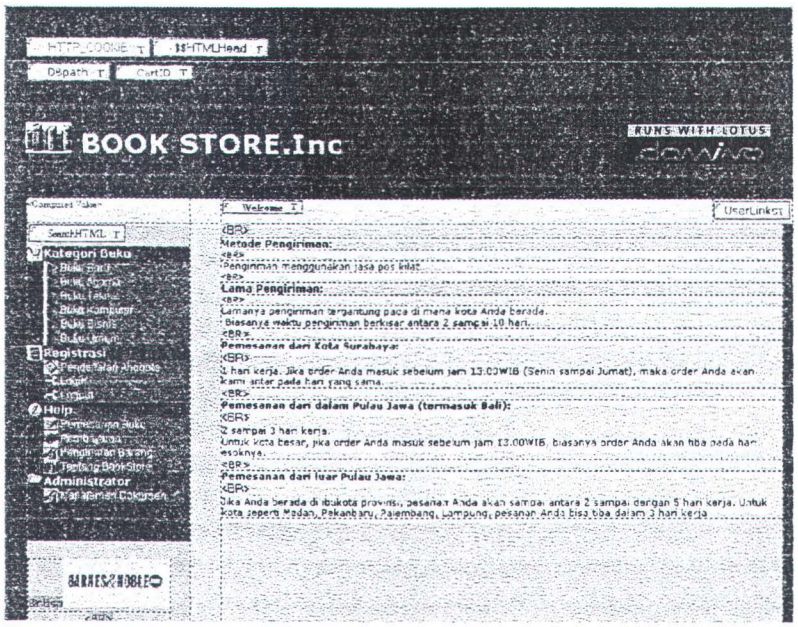
Form ini menampilkan informasi cara pemesanan buku. Gambar 4.12 dibawah ini adalah sebagian tampilan dari rancangan form CaraPemesanan.



Gambar 4.12. Form CaraPemesanan

4.2.14 Form CaraPengiriman

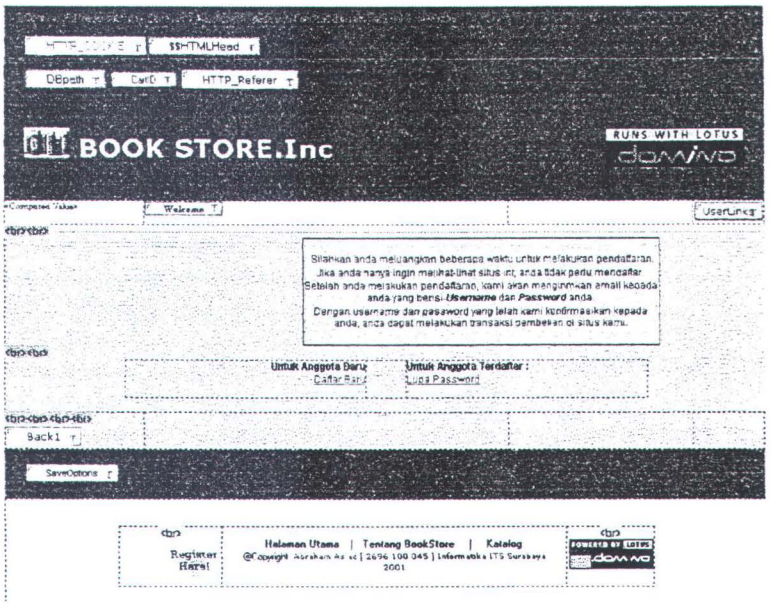
Form ini menampilkan informasi cara pengiriman buku ke alamat pengguna. Gambar 4.13 dibawah ini adalah sebagian tampilan dari rancangan form CaraPengiriman.



Gambar 4.13. Form CaraPengiriman

4.2.15 Form RegChoice

Form ini adalah halaman utama yang akan ditampilkan kepada pengguna, apabila pengguna hendak melakukan pendaftaran. Di form ini terdapat pilihan bagi pengguna baru dan pengguna lama (pelanggan). Untuk pengguna baru operasi yang dapat dilakukan adalah daftar baru, sedangkan untuk pelanggan disediakan menu lupa password untuk meminta informasi tentang password yang dipunyai pelanggan. Gambar 4.14 berikut ini adalah sebagian tampilan dari rancangan form RegChoice.

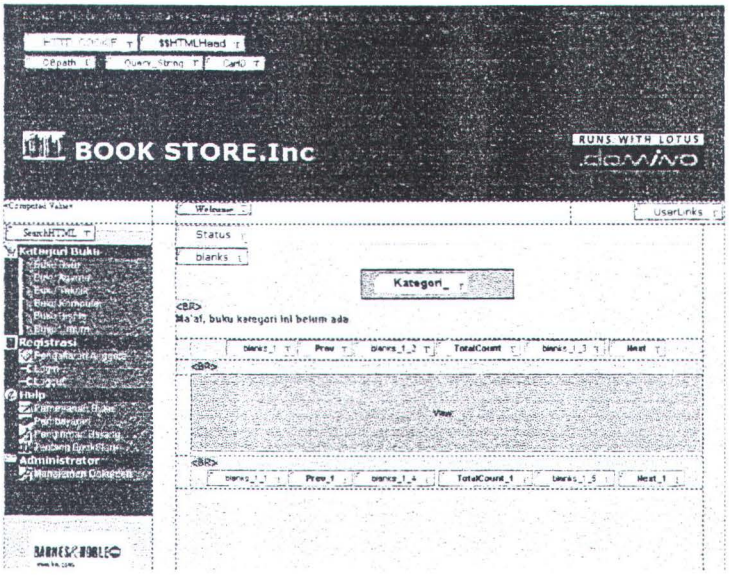


Gambar 4.14 Form RegChoice

4.2.16 Form StatusPesanan

Form ini menampilkan informasi status pesanan pelanggan. Status ini dapat berupa Menunggu Pembayaran, apabila pelanggan melakukan pembayaran lewat transfer bank, Dalam Pengiriman, jika pelanggan sudah membayar barang dan barang sedang dalam pengiriman ke alamat pelanggan, juga Dibatalkan apabila dalam waktu tertentu pesanan belum dibayar oleh pelanggan. Gambar 4.15 berikut ini adalah sebagian tampilan dari rancangan form StatusPesanan.

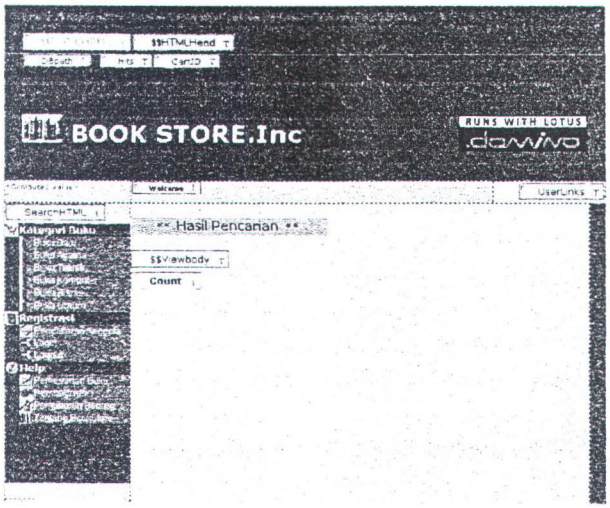
MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 4.16 Form ViewKatalogperKategori

4.2.18 Form $$SearchTemplateDefault

Form ini menampilkan hasil pencarian buku yang dilakukan pengguna. Dari navigator yang ada di sebelah kiri halaman utama, pengguna dapat mengisikan kriteria pencarian menurut berdasarkan kategori yang sudah disediakan yaitu berdasarkan judul buku, nama pengarang, ISBN dan semua kategori. Gambar 4.17 berikut ini adalah sebagian tampilan dari rancangan form $$SearchTemplateDefault.



Gambar 4.17 Form $$SearchTemplateDefault

4.3 View

Untuk melihat dokumen yang telah disimpan dalam database dibuat beberapa view. View ini juga digunakan untuk manajemen data/dokumen di dalam database. Di dalam view dapat dilakukan proses penghapusan dokumen/data yang sudah tidak diperlukan. View-view yang dibuat dalam database ini diantaranya adalah :

- **View Admin/SemuaDokumen**

  View ini digunakan untuk melihat semua dokumen yang tersimpan di dalam database.

- **View KatalogBuku**

  View ini digunakan untuk melihat dokumen yang telah disimpan melalui form EntriKatalog.

- **View BukuTerbaru**

  View ini digunakan untuk melihat dokumen buku yang termasuk dalam kategori baru.

- **View KatalogHasilPencarian**

  View ini digunakan untuk menampilkan hasil pencarian .

- **View KatalogperKategori**

  View ini digunakan untuk menampilkan katalog buku yang tersusun menurut kategori buku.

- **View KatalogBuku**

  View ini digunakan untuk melihat dokumen yang telah disimpan melalui form EntriKatalog.

- **View (CartDetailByID)**

  View ini digunakan untuk menampilkan isi keranjang belanja.

4.4 Agent

Agent dibutuhkan untuk melaksanakan suatu proses. Berikut ini adalah beberapa agent yang melakukan proses utama yang dibuat dalam aplikasi ini.

4.4.1 CekStokBuku

Agent ini digunakan untuk menangani proses notifikasi. Agent ini dijadwalkan berjalan sehari sekali. Proses yang dijalankan adalah agent mengecek view notifikasi untuk mendapatkan judul buku yang dipesan oleh pengguna kemudian mengecek stok buku di database, apabila stok buku sudah ada maka agent mengirim email kepada pengguna kemudian data pengguna di view

notifikasi dihapus. Agent ini ditulis dengan menggunakan Lotus Script. *Listing* programnya adalah sebagai berikut :

```
Dim db As NotesDatabase
Dim s As NotesSession
Dim view As NotesView

Sub Initialize
        Set s = New NotesSession
        Set db= s.CurrentDatabase

        Dim vdoc As NotesDocument
        Dim cdoc As NotesDocument
        Dim cview As NotesView
        Dim ISBN As Variant
        Dim  vstok As Variant
        Dim Email As String
        Dim Judul As String
        Dim pdoc As NotesDocument

        Set view = db.GetView("NotificationView")
        Set cview = db.GetView("KatalogJudul")

        Set vdoc = view.GetFirstDocument

        While Not ( vdoc Is Nothing )
                ISBN = vdoc.GetItemValue( "ISBN" )(0)
                Email = vdoc.GetItemValue( "Email" )(0)
                Judul = vdoc.GetItemValue( "Judul" )(0)
                Set cdoc = cview.getDocumentbykey(ISBN,True)
                vstok = cdoc.GetItemValue("Stok")(0)
                If vstok="1" Then
                        Call SendEmail(Email,Judul)
                        Set pdoc = vdoc
                        Set vdoc = view.GetNextDocument( vdoc )

                        Call pdoc.Remove( True )
                Else
                        Set vdoc = view.GetNextDocument( vdoc )
                End If
        Wend
End Sub
Sub sendEmail(Email As String, Judul As String)
        Dim docMemo As New NOTESDOCUMENT(db)
        Dim bodyField As New NOTESRICHTEXTITEM(docMemo, "Body")


        Call docMemo.ReplaceItemValue("SendTo", Email)
        Call docMemo.ReplaceItemValue("Subject","Buku yang anda cari telah
        tersedia")
        Call docMemo.ReplaceItemValue("From","webmaster@bookstore.net")
        Call docMemo.ReplaceItemValue("Form","Memo")
```

```
        Call bodyField.APPENDTEXT("Buku :  """ &Judul &"""  ,yang anda
inginkan telah tersedia di toko kami.")
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT("Silahkan membelinya lewat situs kami.")
        Call bodyField.ADDNEWLINE(1)
        Call bodyField.APPENDTEXT("Terima kasih atas perhatian anda." )
        Call bodyField.ADDNEWLINE(3)
        Call bodyField.APPENDTEXT("TTD" )
        Call bodyField.ADDNEWLINE(1)
        Call bodyField.APPENDTEXT("Webmaster BookStore.Inc" )
        Call docMemo.Send(False)
End Sub
```

4.4.2 AddtoCart

Agent ini digunakan untuk menangani proses add to cart. Agent ini dijalankan pada waktu pengguna menekan tombol AddttoCart pada form EntriKatalog(forWeb). Agent ini pertama akan mengecek apakah keranjang belanja pengguna sudah terisi, jika sudah maka agent menambahkan item ke keranjang pengguna, jika belum maka agent membuat keranjang belanja baru dan memasukkan item yang diinginkan pengguna kedalam keranjang. Agent ini ditulis dengan menggunakan Lotus Script. *Listing* programnya adalah sebagai berikut :

```
Sub Initialize
        Dim s As New NotesSession
        Dim db As NotesDatabase
        Dim doc As NotesDocument, oiDoc As NotesDocument, cDoc As
        NotesDocument
        Set db = s.CurrentDatabase

        Set doc = s.DocumentContext

        Dim vCartID As Variant, vISBN As Variant, vItemID As Variant,  vJumlah As
        Variant, vPosn As Variant, vPath As Variant
        Dim vOrderKey(1) As String

        'Determin CartID
        vCartID = Evaluate ({ @Middle (@LowerCase(Query_String) + "&";
        "&cartid="; "&") }, doc)
        If vCartID(0) = "" Then
        vCartID=        Evaluate({@Middle(@LowerCase(HTTP_COOKIE)        +
        ";";"cartid=";";")}, doc)
        End If
```

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

```
'Get ISBN from URL
vISBN = Evaluate ({ @Middle (@LowerCase(Query_String) + "&"; "&isbn=";
"&") }, doc)

'Get the catalog document using the ISBN as the key
Set cDoc = db.GetView ("LookUpISBN").GetDocumentByKey (vISBN(0))

' Get the existing order item document, if it exists.
vOrderKey(0)=vCartID(0)
vOrderKey(1)=vISBN(0)
Set oiDoc = db.GetView ("LookUpOrderISBN").GetDocumentByKey
(vOrderKey)

' Create new order item document in the event one was not found
If oiDoc Is Nothing Then
        Set oiDoc = db.CreateDocument
        oiDoc.Form = "OrderItem"
        oiDoc.CartID = vCartID
        oiDoc.Jumlah = 1
        oiDoc.ISBN = vISBN
        oiDoc.Judul = cDoc.Judul
        oiDoc.Pengarang = cDoc.Pengarang

Else
        oiDoc.Jumlah=oiDoc.Jumlah(0) + 1

End If
```

4.4.3 UpdateCart

Agent ini digunakan untuk mengupdate keranjang belanja. Agent ini dijalankan pada waktu pengguna menekan tombol Recalculate pada form KeranjangBelanja. Agent ini mengambil nilai dari variabel CGI *Request_Content* yang mengandung data yang dikirimkan pada waktu pengguna menekan tombol Recalculate (yang menjalankan perintah post document *(http post)*), diantaranya yaitu nomer ISBN dari buku, dan jumlah yang disi pengguna di field jumlah serta nilai dari *check box* delete. Setelah nilai yang diinginkan diambil, maka agent akan mengubah dokumen menurut data yang dikirimkan. Apabila pengguna mengisi check box delete maka buku dihapus dari keranjang, jika pengguna mengubah isi dari jumlah maka jumlah buku akan diubah menurut data yang

diisikan pengguna, kemudian keranjang belanja akan ditampilkan kembali *(refresh)*. Agent ini ditulis dengan menggunakan Lotus Script. *Listing* programnya adalah sebagai berikut :

```
Sub Initialize
        Dim s As New NotesSession, db As NotesDatabase
        Dim doc As NotesDocument, oiDoc As NotesDocument

        Set db = s.CurrentDatabase
        Set doc = s.DocumentContext

        Dim vCartID As Variant, vPath As Variant, vContent As Variant
        Dim vDelete As Variant, vJumlah As Variant, vISBN As Variant, vSessionID As
        Variant
        Dim emptyQuantity As Integer

    'get CartID

        vCartID=        Evaluate({@Middle(@LowerCase(HTTP_COOKIE)        +
        ";";"cartid=";";")}, doc)
        If vCartID(0) = "" Then
        vCartID = Evaluate ({@Middle (@LowerCase(HTTP_Referer) + "&";
        "&cartid="; "&")}, doc)
        End If

    ' ambil nilai request_content

        vContent = Evaluate({@Explode(Request_Content;"&")}, doc)

        Forall arg In vContent
                If emptyQuantity Then
                        arg = "ISBN="
                        emptyQuantity = False
                End If
                If arg = "JUMLAH=" Then
                        emptyQuantity = True
                End If
        End Forall

        doc.tmpContent = vContent
        vDelete=Evaluate({@Trim(@Right(tmpContent;"DELETE="))},doc)
        vJumlah=Evaluate({@Trim(@Right(tmpContent;"JUMLAH="))},doc)
        vISBN=Evaluate({@Trim(@Right(tmpContent;"ISBN="))},doc)

        'Update Order Item documents

        For x=0 To Ubound(vISBN)
                Set oidoc = db.GetDocumentByUNID(vISBN(x))
                If Not oidoc Is Nothing Then
                        If Cint(vJumlah(x))>0 Then
                                oidoc.Jumlah = Cint(vJumlah(x))
```

```
                    Call oidoc.Save(True,True)
                End If
            End If
        Next

        'Delete an Order Item documents if user mark  delete

        If Not vDelete(0) ="" Then
            Forall v In vDelete
                Set oidoc = db.GetDocumentByUNID(v)
                If Not oidoc Is Nothing Then
                    Call oidoc.Remove(True)
                End If
            End Forall
        End If

        'Set URL path for return - in this case a simply redisplay of the cart

        vPath=Evaluate({@ReplaceSubstring (@Subset (@DbName; -1); "\" : " "; "/" :
        "+")})
        Print "[" + vPath(0) + "/cart?ReadForm&CartID=" + vCartID(0) + "]"

End Sub
```

4.4.4 ClearCart

Agent ini dijalankan setelah pelanggan selesai melakukan proses pembelian. Agent ini akan menghapus isi dari keranjang belanja pelanggan kemudian mengirimkan email konfirmasi pembelian kepada pelanggan. Agent ini ditulis dengan menggunakan Lotus Script. *Listing* programnya adalah sebagai berikut :

```
Dim db As NotesDatabase
Dim s As NotesSession

Sub Initialize
    Dim dc As NotesDocumentCollection
    Dim doc As NotesDocument

    Set s = New NotesSession
    Set db = s.CurrentDatabase
    Set doc = s.DocumentContext

    Set dc = db.GetView( "(CartDetailByID)"
    ).GetAllDocumentsByKey(doc.CartID,True)
' Ganti field Status dan OrderId dari dokumen OrderItem
    Call dc.StampAll ("Status", "MenungguPembayaran")
```

```
        Call dc.StampAll ("OrderID", doc.OrderID)
        Call sendEmail
        Call doc.ReplaceItemValue("SaveStatus", "1")
End Sub
Sub sendEmail
        Dim doc As NotesDocument

        Set s = New NotesSession
        Set db = s.CurrentDatabase
        Set doc = s.DocumentContext

        Dim docMemo As New NOTESDOCUMENT(db)
        Dim bodyField As New NOTESRICHTEXTITEM(docMemo, "Body")

        Dim userAddress As String
        userAddress = doc.GetItemValue("Email")(0)

        Call doc.ReplaceItemValue("ErrorText", "ok")
        Call docMemo.ReplaceItemValue("SendTo", userAddress)
        Call docMemo.ReplaceItemValue("Subject","Daftar belanja anda")
        Call docMemo.ReplaceItemValue("From","webmaster@bookstore.net")
        Call docMemo.ReplaceItemValue("Form","Memo")

        Call bodyField.APPENDTEXT("Terima kasih , ")
        Call bodyField.ADDNEWLINE(1)
        Call bodyField.APPENDTEXT("Anda telah membeli di tempat kami.")
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT("Order ID anda adalah : " &
        doc.GetItemValue("OrderID")(0))
        Call bodyField.ADDNEWLINE(3)
        Call bodyField.APPENDTEXT(" Berikut ini adalah daftar belanja anda : ")
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT(" Judul Buku : " &
        doc.GetItemValue("Judul")(0) &" Jumlah buku :" &
        doc.GetItemValue("Jumlah")(0)& " Harga :
        "&doc.GetItemValue("HargaTotal")(0))
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT("Harga Total : " &
        doc.GetItemValue("Total")(0))
        Call docMemo.Send(False)
End Sub
```

4.4.5 ProsesKartuKredit

Agent ini adalah sebagai simulasi validasi kredit. Setelah pelanggan mengisikan nomer kartu dan password dan menekan tombol submit pada form Order maka agent ini akan dijalankan. Agent ini akan mengecek nomer kartu dan password pada database dengan nomer kartu dan password yang diisikan oleh

pelanggan. Jika valid maka agent akan menghapus isi keranjang belanja dan mengirim email konfirmasi pembelian kepada pelanggan. Agent ini ditulis dengan menggunakan Lotus Script. *Listing* programnya adalah sebagai berikut :

```
Sub Initialize
        Dim db As NotesDatabase
        Dim view As NotesView
        Dim docKartu As NotesDocument
        Dim Saldo As Variant
        Dim  Harga As Variant
        Dim dc As NotesDocumentCollection


        Set s = New NotesSession
        Set db = s.CurrentDatabase
        Set doc = s.DocumentContext

        Set view = db.GetView("VKartuKredit")
        If doc.GetItemValue("CCNumber")(0) = "" Then
        Call doc.ReplaceItemValue("ErrorText", "Nomer kartu kredit harus diisi !")
        Else
                Set docKartu =
                view.Getdocumentbykey(doc.GetItemValue("CCNumber")(0),True)
                If docKartu Is Nothing Then
                        Call doc.ReplaceItemValue("ErrorText", "Nomer kartu kredit
                        tidak ada !")
                Else
                        If docKartu.GetItemValue("CCPassword")(0) =
                        doc.GetItemValue("CCPassword")(0) Then
                                Saldo = docKartu.GetItemValue("CCSaldo")(0)
                                Harga =  doc.GetItemValue("PHargaTotal")(0)
                                If Saldo < Harga Then
                                        Call      doc.ReplaceItemValue("ErrorText",
                                        "Saldo anda  tinggal : Rp."+Saldo+ ", tidak
                                        cukup untuk melakukan transaksi ini !")
                                Else
                                        Saldo = Saldo-Harga
                                        Call
                                        docKartu.ReplaceItemValue("CCSaldo",
                                        Saldo)
                                        Call  docKartu.ComputeWithForm(  False,
                                        False )
                                        Call docKartu.Save( False, True )
                        'Clear cart
                                        Set  dc  =  db.GetView(  "(CartDetailByID)"
                                        ).GetAllDocumentsByKey(doc.CartID,True)
                            ' Ganti field Status dan OrderId dari dokumen OrderItem
                                        Call dc.StampAll ("Status",
                                        "DalamPengiriman")
                                        Call dc.StampAll ("OrderID", doc.OrderID)
                                        Call sendEmail
```

```
                                    Call doc.ReplaceItemValue("Status",
                                    "DalamPengiriman")
                                    Call doc.ReplaceItemValue("SaveStatus",
                                    "1")
                            End If
                        Else
                            Call doc.ReplaceItemValue("ErrorText", "Password
                            anda salah !")
                        End If
                    End If
                End If
            End Sub
```

4.4.6 DaftarAnggota

Agent ini memproses pendaftaran pelanggan. Agent ini akan dijalankan setelah pengguna menekan tombol submit pada form Registrasi. Agent ini akan mengecek username yang diisikan oleh pengguna ke database Names.nsf di Domino Server. Jika username ada maka agent akan mengirimkan pesan kesalahan kepada pengguna. Jika tidak ada maka username pelanggan dan semua data pada form Registrasi akan disimpan di database names.nsf. Setelah disimpan agent akan mengirimkan email berisi username dan password pelanggan kepada pelanggan. Agent ini ditulis dengan menggunakan Lotus Script. Agent ini menggunakan *library* CommonRoutines. *Listing* programnya adalah sebagai berikut :

```
Option Public
Option Declare

'Use declaration and routines in commonRoutines Library

Use "CommonRoutines"
'Declare variable to handle registration document
'access reader access.
Dim ReaderString(1) As String
Dim vpath As Variant
Dim vCartID As Variant
Sub Initialize
```

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

```
        'Set doc equal to submitted web form
        Set doc = s.DOCUMENTCONTEXT

        'intialize Reader security
        ReaderString(0) = "[SiteManager]"
        ReaderString(1) = doc.Username(0)
        'Register user by call registration sub
        Call registerNewUser

End Sub
Sub registerNewUser

        Call validateUser 'Check for duplicate names
        If(userMatchFlag = True) Then
                Call doc.ReplaceItemValue("Status","0")
                Print "Error - Username yang anda pilih : " &
                doc.GetItemValue("Username")(0) _
                & "  telah digunakan.  Silahkan memasukkan username yang lain."
        Else

                Call AddUserToGroup  'Add user to proper access group

                If (retGroupStatus = False) Then
                        Call doc.ReplaceItemValue("Status", "0")
                        Print "An error occured while processing this group request,
                        your registration " _
                        & "will not be processed"
                Else
                        Set docPerson = dbNab.CreateDocument

                        Call docPerson.ReplaceItemValue("form", "Person")
                        Call docPerson.ReplaceItemValue("Type", "Person")
                        Call docPerson.ReplaceItemValue("LastName",
                        doc.GetItemValue("Username")(0))
                        Call docPerson.ReplaceItemValue("FullName",
                        doc.GetItemValue("Username")(0))
                        Call docPerson.ReplaceItemValue("HTTPPassword",
                        doc.GetItemValue("Password")(0))
                        Call docPerson.ReplaceItemValue("OfficePhoneNumber",
                        doc.GetItemValue("Telepon")(0))
                        Call docPerson.ReplaceItemValue("InternetAddress",
                        doc.GetItemValue("Email")(0))
                        Call docPerson.ComputeWithForm( False, False )
                        ret=docPerson.Save( False, True )

                        If (ret = True) Then
                        ' refresh the database
                                Dim nabFullNameView As NotesView
                                Set nabFullNameView =
                                dbNAB.GetView("($LDAPCN)")

                                Call nabFullNameView.Refresh

                                'Call doc.ReplaceItemValue("SaveOptions", "1")
                                'Call doc.Save(False, False)
```

```
                    Call sendEmail

                    Call doc.ReplaceItemValue("Status", "1")
                    ' Terimakasih '

                    vCartID=
                    Evaluate({@Middle(@LowerCase(HTTP_COOKIE)
                    + ";";"cartid=";";")}, doc)
                    If vCartID(0) = "" Then
                    vCartID = Evaluate ({@Middle
                    (@LowerCase(HTTP_Referer) + "&"; "&cartid=";
                    "&")}, doc)
                    End If
                    vPath=Evaluate({@ReplaceSubstring (@Subset
                    (@DbName; -1); "\" : " "; "/" : "+")})
                    Print "[" + vPath(0) +
                    "/Konfirmasi?OpenForm&UN="+doc.GetItemValue(
                    "Username")(0)+"&Pwd="+doc.GetItemValue("Pas
                    sword")(0)+"&CartID=" + vCartID(0) + "]"

                Else

                    Call doc.ReplaceItemValue("Status", "0")
                    Print "An error occured while processing this
                    request, your registration" _
                    & "will not be processed"
                End If
            End If
        End If
End Sub
Sub sendEmail
        Dim docMemo As New NOTESDOCUMENT(db)
        Dim bodyField As New NOTESRICHTEXTITEM(docMemo, "Body")

        Dim userAddress As String
        userAddress = doc.GetItemValue("Email")(0)

        Call docMemo.ReplaceItemValue("SendTo", userAddress)
        Call docMemo.ReplaceItemValue("Subject","Konfirmasi pendaftaran di
        BookStore.Inc")
        Call docMemo.ReplaceItemValue("From","webmaster@bookstore.net")
        Call docMemo.ReplaceItemValue("Form","Memo")
        Call bodyField.APPENDTEXT("Terima kasih ! ")
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT("Anda telah terdaftar menjadi anggota
BookStore.Inc.")
        Call bodyField.ADDNEWLINE(2)
        Call bodyField.APPENDTEXT("Username anda adalah: " &
        doc.GetItemValue("Username")(0))
        Call bodyField.ADDNEWLINE(1)
        Call bodyField.APPENDTEXT("Password anda adalah : " &
        doc.GetItemValue("Password")(0))
        Call docMemo.Send(False)
End Sub
```

4.5 Script Libraries

Script libraries adalah program yang dapat digunakan oleh semua agent yang ada di dalam suatu database. Untuk menggunakan deklarasi variabel dan fungsi-fungsi yang ada di *script libraries* digunakan perintah *use "nama library"*. Di dalam aplikasi ini dibuat sebuah *script library* yang diberi nama CommonRoutines. Di dalam CommonRoutines terdapat fungsi ValidasiGroup yang mengecek grup BookStoreWebUser yang digunakan untuk menampung anggota aplikasi ini dan Validasi User yang mengecek kevalidan username yang dimasukkan pengguna, *library* ini digunakan oleh agent DaftarUser.

BAB V
UJI COBA PERANGKAT LUNAK

BAB V

UJI COBA PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai uji coba dari aplikasi yang telah dibuat. Uji coba ini dilakukan dengan tujuan bahwa aplikasi dapat melakukan proses transaksi sebagaimana lazimnya sebuah situs *E-commerce*.Proses-proses yang di uji coba adalah proses *browsing* katalog dan pencarian buku, pendaftaran pelanggan, pembelian buku dan validasi pembayaran lewat kartu kredit serta proses manajemen dokumen.

5.1 Lingkungan Uji Coba

Uji coba dilakukan pada jaringan intranet. Aplikasi yang dibuat dijalankan pada komputer client yang mengakses komputer server. Oleh karena aplikasi yang dibuat merupakan aplikasi dari sistem client / server, maka terdapat persyaratan minimal sistem aplikasi pada client maupun pada server agar dapat berjalan secara optimal.

Kebutuhan minimum yang dibutuhkan untuk menjalankan aplikasi yang telah dibuat pada komputer server adalah :

- Sistem Operasi — Windows NT 4.0 dengan Service Pack 3.
- Perangkat Lunak — Lotus Domino Server, Lotus Domino Administrator

Sedangkan kebutuhan minimum yang dibutuhkan untuk menjalankan aplikasi pada komputer client adalah :

- Sistem Operasi — Windows NT, Windows 98 atau Windows 95.
- Perangkat Lunak — Lotus NotesClient untuk administrator, Microsoft Internet Explorer 4.01 (web browser lain yang mendukung java applet) untuk semua pengguna

5.2 Uji coba Aplikasi

Setelah semua sarana dan perangkat lunak pendukung sudah siap maka dilakukan uji coba aplikasi yang telah dibuat. Sebelum pelaksanaan uji coba dilakukan maka pertama kali perlu dilakukan pengaktifan server. Pada ujicoba ini Domino Server diinstall dan dijalankan di komputer Lotus di Laboratorium Sistem Informasi. Komputer ini mempunyai spesifikasi :

Hardware :

Prosesor : Intel Pentium III 500 Mhz

Memori : 128 MB

Software :

Sistem Operasi : Windows NT Server, Enterprise Edition Version 4

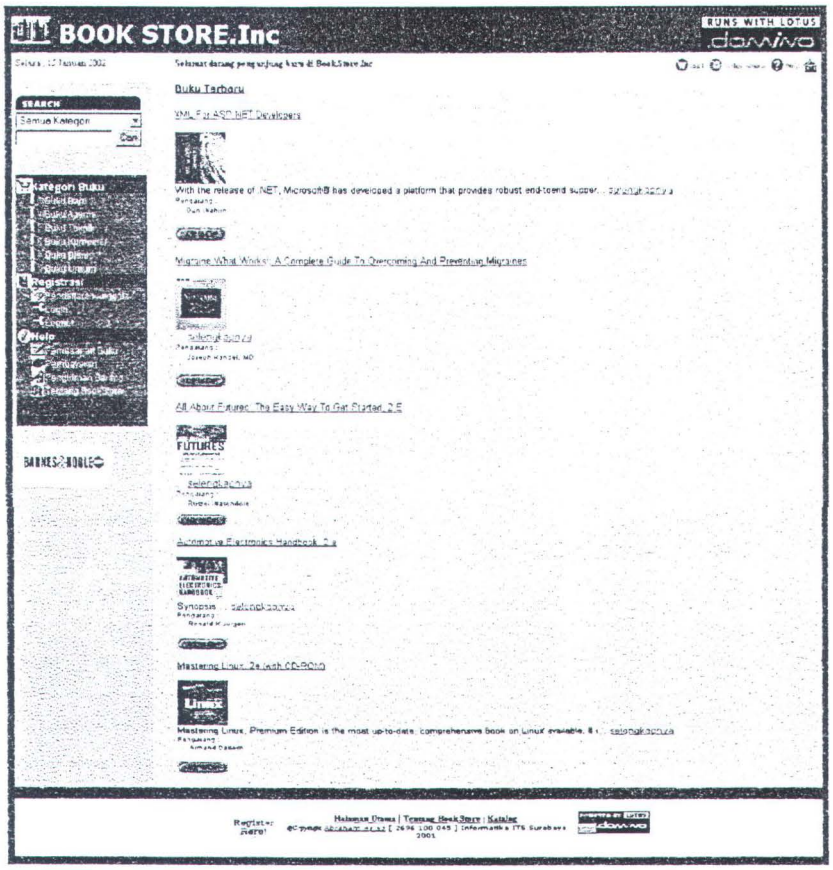
Perangkat Lunak : Lotus Domino Server 5, Lotus Domino Administrator 5

Sedangkan untuk komputer client, menggunakan dua komputer yang ada di Laboratorium Komputing. Web browser yang digunakan adalah Microsoft Internet Explorer 5 yang mendukung java applet. Jaringan yang digunakan adalah jaringan intranet yang sudah terpasang kampus teknik informatika. Sebelum membuka situs BookStore.Inc, maka pengguna harus mengeset web browser agar dapat menerima cookie, yang digunakan untuk menyimpan data keranjang belanja pengguna, dan mengeset LAN Setting pada Internet Options dengan memilih pilihan-*bypass proxy for local addresses*. Hal ini dilakukan agar waktu pengaksesan situs dapat lebih cepat.

5.2.1 Proses Browsing Katalog dan Pencarian Buku

Proses browsing katalog dan pencarian buku dapat dilakukan dari hampir seluruh halaman dari aplikasi ini. Hal ini karena navigator kiri yang merupakan navigator yang berisi kategori katalog dan field pencarian dapat ditemukan di hampr semua halaman BookStore.Inc.

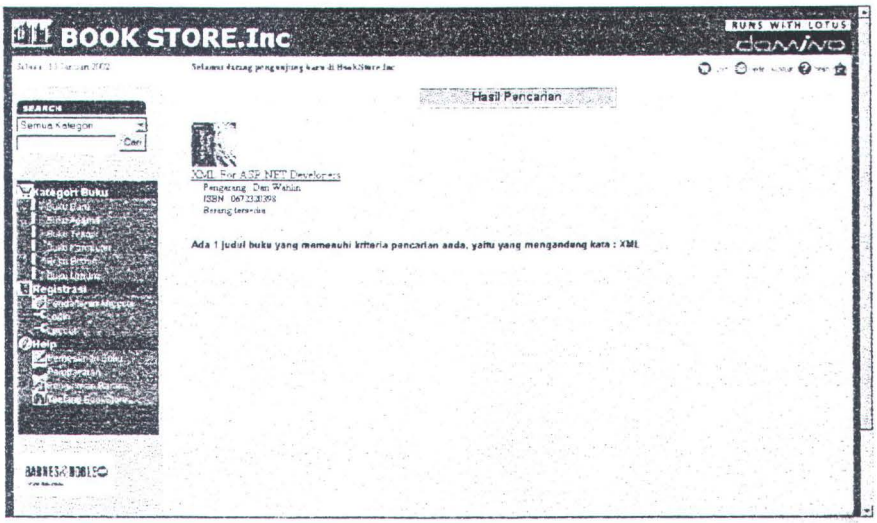
Gambar 6.1 berikut ini adalah tampilan halaman utama dari BookStore.Inc.



. **Gambar 5.1**. Halaman Utama

Dapat dilihat pada gambar itu bahwa situs ini mempunyai fasilitas Pencarian Buku yang dapat dilihat pada navigator sebelah kiri atas dibawah logo BookStore.Inc. Fasilitas Pencarian Buku ini memiliki alternatif pencarian buku berdasarkan empat kategori yaitu, semua kategori yang merupakan default kategori, kategori judul, pengarang, dan ISBN.

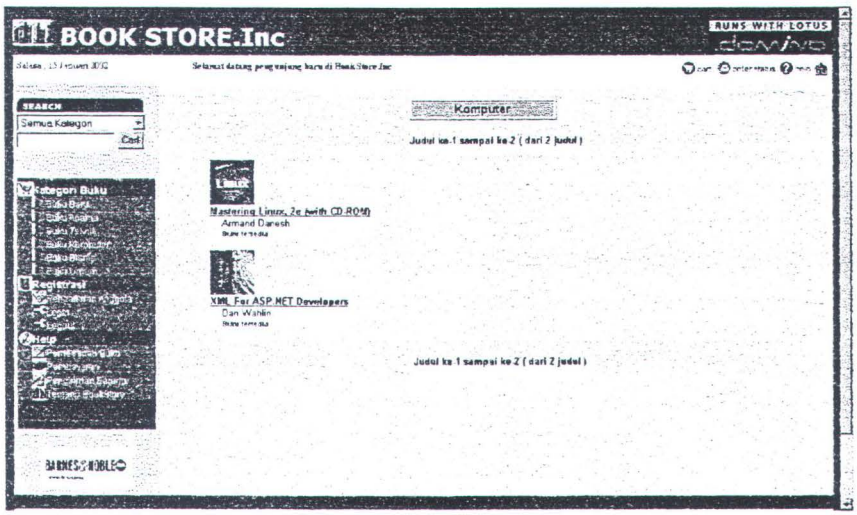
Berikut ini adalah tampilan yang muncul setelah penulis mengetikkan kata "XML" pada field isian pencarian dan memilih kategori berdasarkan judul buku.



Gambar 5.2. Tampilan Hasil Pencarian judul "XML"

Selain dengan fasilitas pencarian, pengguna juga dapat mencari buku yang diinginkan dengan memilih salah satu kategori buku yang diinginkan pada navigator disebelah kiri halaman utama.

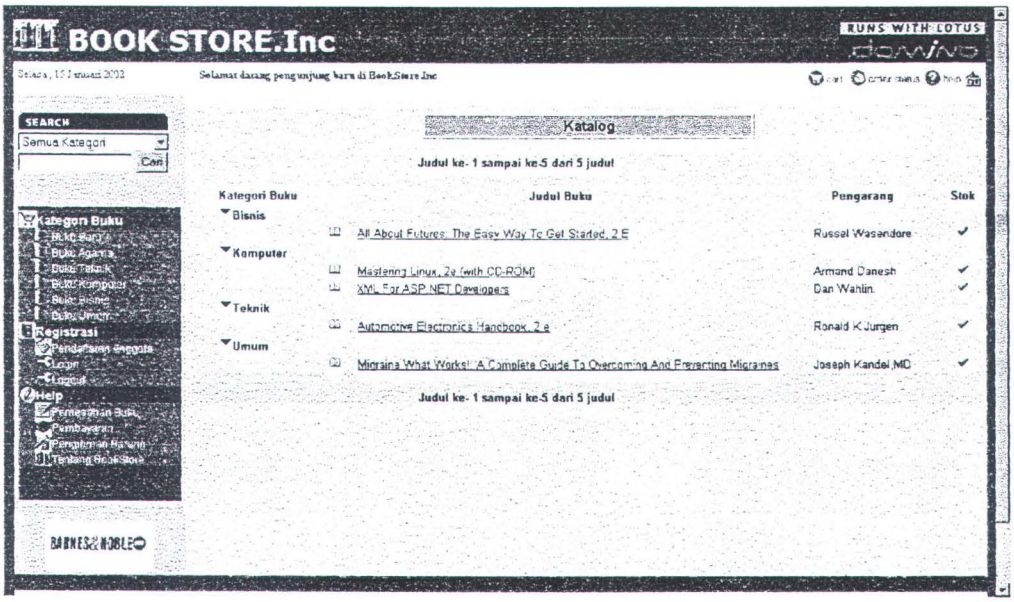
Berikut ini adalah tampilan yang muncul setelah penulis meng-klik kategori Komputer pada navigator sebelah kiri.



Gambar 5.3 Tampilan Katalog Kategori Komputer

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Apabila pengguna ingin melihat seluruh daftar buku maka pengguna dapat meng-klik menu/link Katalog pada navigator sebelah bawah halaman utama. Berikut ini adalah tampilan yang muncul setelah penulis meng-klik Katalog Buku pada navigator bawah.

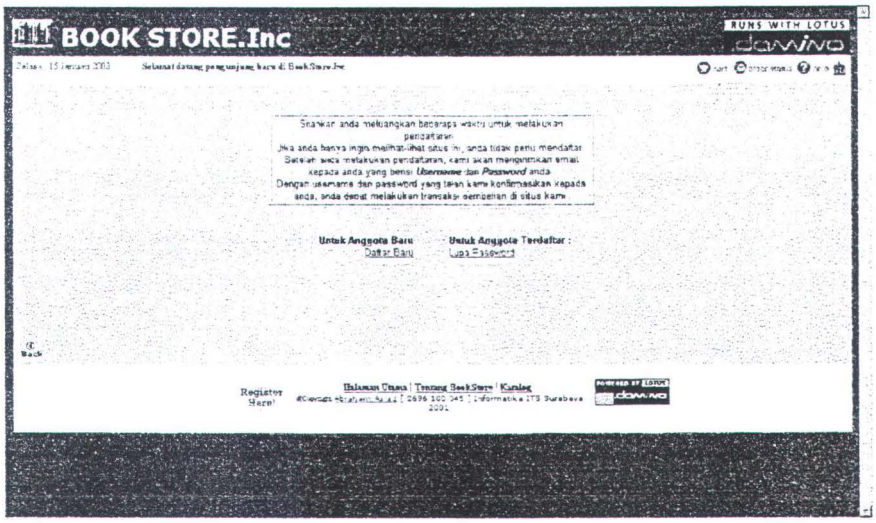


Gambar 5.4. Halaman Katalog Buku

Tampilan katalog buku diatas berisi semua buku yang tersedia di situs ini. Dari tampilan diatas penulis juga dapat melihat stok buku. Pada gambar diatas stok semua buku adalah tersedia, yang dilambangkan dengan tanda centang.

5.2.2 Proses Pendaftaran Pelanggan

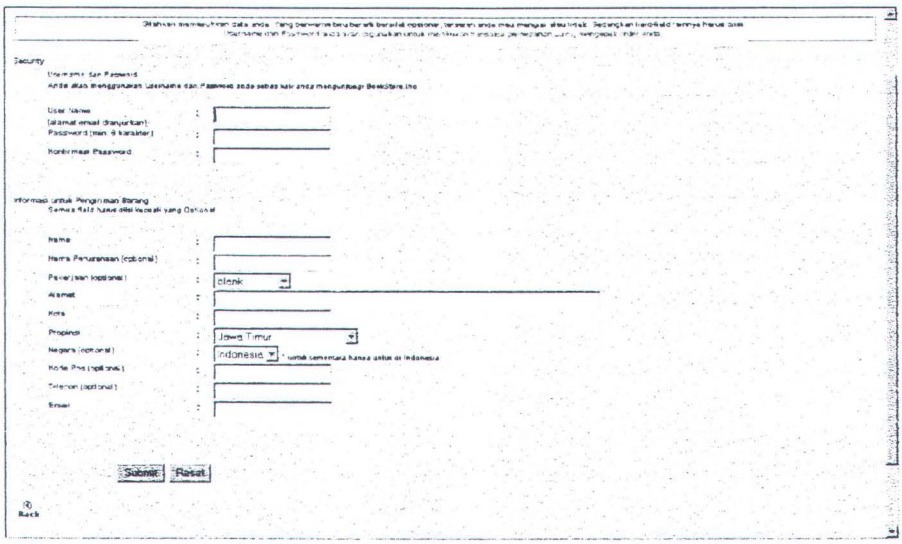
Untuk melakukan proses pendaftaran pengguna dapat meng-klik menu Pendaftaran Anggota pada navigator sebelah kiri atau dengan meng-klik link Register Here pada navigator bawah. Berikut ini adalah gambar tampilan pilihan pendaftaran yang muncul setelah penulis meng-klik menu pendaftaran anggota.



Gambar 5.5.. Tampilan utama menu pendaftaran anggota

Pada gambar diatas dapat dilihat ada dua menu yang dapat dijalankan pengguna, yaitu menu Daftar Baru, yang diperuntukkan bagi pengguna baru yang belum terdaftar sebagai anggota, dan menu Lupa Password yang diperuntukkan bagi anggota yang lupa dengan password yang dipunyainya.

Gambar berikut ini adalah tampilan dari halaman pendaftaran pelanggan baru yang mucul setelah penulis meng-klik menu Daftar Baru.

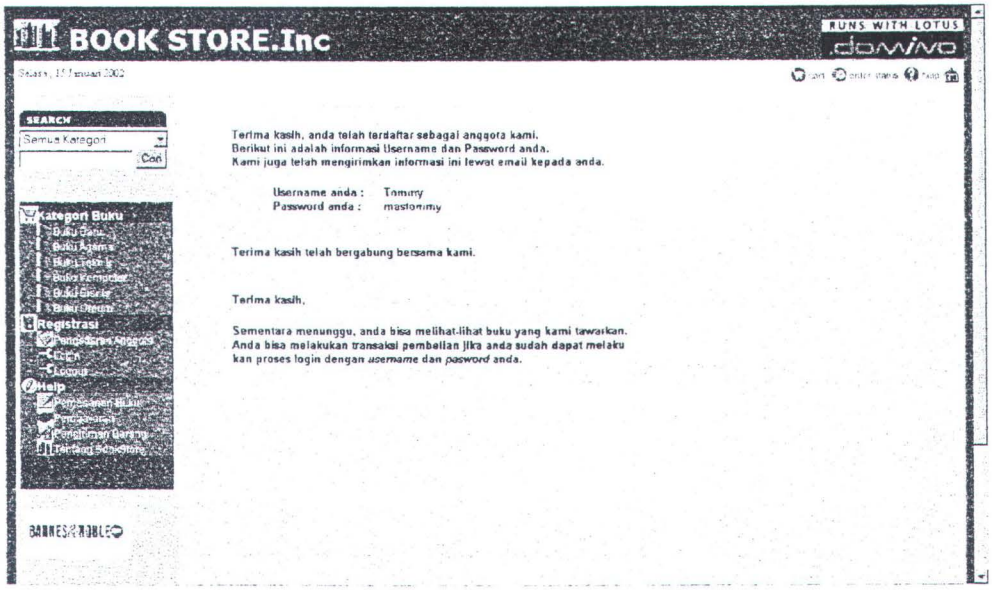


Gambar 5.6. Halaman Pendaftaran Pelanggan

Pada halaman ini terdapat field-field yang harus diisi dan ada field-field yang bersifat opsional. Ketika penulis mencoba men-submit form ini tanpa mengisi username yang merupakan field yang harus diisi, penulis mendapatkan peringatan yang memberitahukan bahwa field username harus diisi yang berupa *pop up message* . Demikian juga jika penulis tidak mengisi field yang harus diisi lainnya.

Kemudian penulis mengisikan data sebagai berikut : Username = Tommy, Password = mastommy, Konfirmasi Password = mastommy, Nama = Tommy J. Pisa, Perusahaan = PT. Maju Jaya, Pekerjaan = Mahasiswa, Alamat = Jl. Maleo 9, Kota = Jakarta, Propinsi = DKI Jakarta, Negara = Indonesia, Kode Pos = 56009, Telepon = 021- 32221376, email = abrahamas'ad@sisfodomino.edu.

Setelah itu penulis menekan tombol submit maka dalam waktu enam detik kemudian muncul halaman konfirmasi pendaftaran berhasil yang berisi username dan password pelanggan. Berikut ini adalah gambar tampilan halaman konfirmasi yang muncul setelah penulis berhasil melakukan pendaftaran yang menampilkan username yang telah didaftarkan penulis yaitu "tommy" dengan password "mastommy".



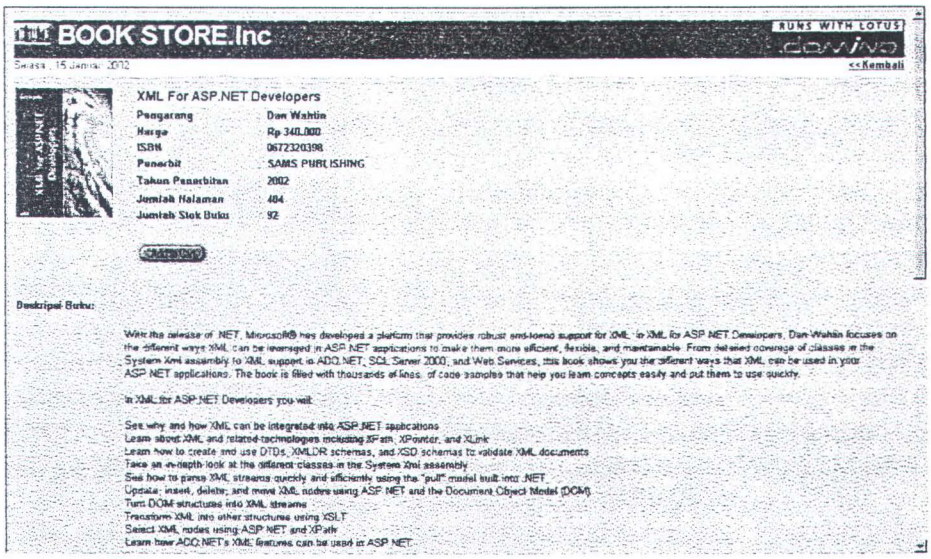
Gambar 5.7. Tampilan Halaman Konfirmasi Pendaftaran Berhasil

5.2.2 Proses Pembelian Buku

Setelah berhasil melakukan pendaftaran maka penulis dapat melakukan proses pembelian buku dan validasi pembayaran buku. Pada aplikasi ini semua pengguna baik yang sudah terdaftar maupun yang belum terdaftar dapat memasukkan buku ke dalam keranjang belanja. Akan tetapi untuk melakukan proses selanjutnya yaitu proses cek out untuk melakukan pembayaran hanya pengguna yang sudah terdaftar sebagai anggota yang dapat melakukannya.

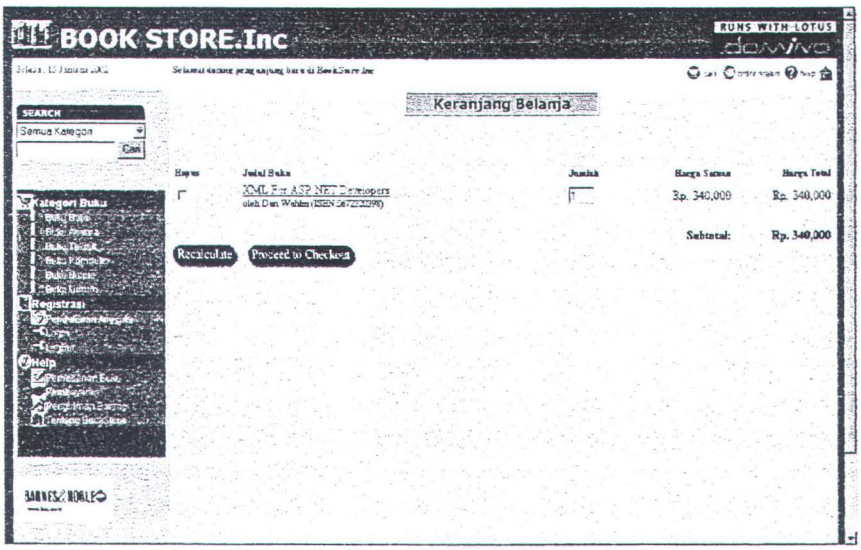
Untuk memasukkan buku kedalam keranjang belanja penguna harus memilih buku, kemudian menekan tombol AddtoCart yang terdapat pada deskripsi buku. Khusus untuk buku baru yang ada di halaman utama, pengguna dapat langsung menekan tombol AddtoCart yang ada di bawah deskripsi singkat masing-masing buku.

Pada uji coba ini penulis memilih buku "XML for ASP.Net Developers" sebanyak satu buah. Berikut ini adalah gambar tampilan halaman deskripsi buku "XML for ASP.Net Developers".



Gambar 5.8. Halaman Deskripsi Buku "XML for ASP.Net Developers"

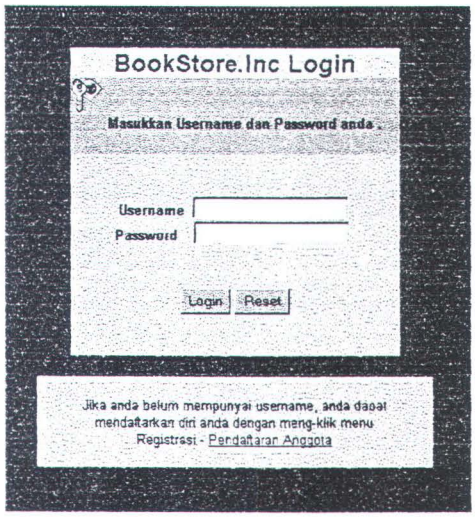
Setelah peulis menekan tombol AddtoCart maka halaman keranjang belanja akan ditampilkan. Berikut ini adalah gambar halaman keranjang belanja penulis.



Gambar 5.9. Halaman Keranjang Belanja

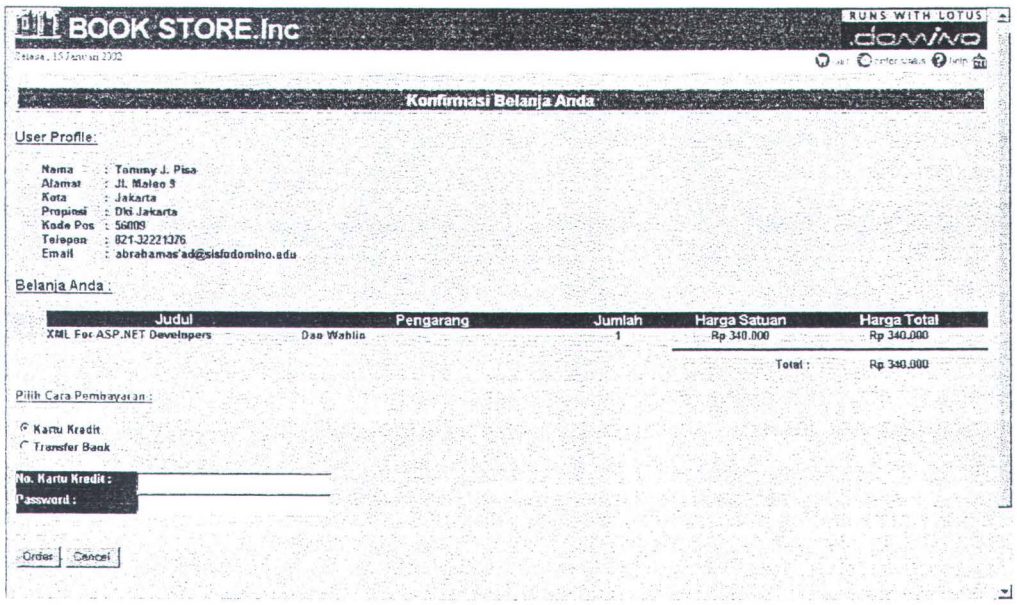
Di halaman keranjang belanja ini pengguna dapat mengubah jumlah item yang ingin di beli. Untuk itu pengguna dapat mengisi field Jumlah dengan angka yang diinginkan kemudian menekan tombol Recalculate. Untuk menghapus buku keranjang, pengguna cukup mengisi check box Hapus disamping kiri judul buku dan menekan tombol Recalculate. Dalam uji coba ini penulis dapat mengubah jumlah buku yang akan dibeli dan menghapus buku dari keranjang belanja dengan cara diatas dengan baik.

Untuk melakukan proses pembayaran, pengguna menekan tombol Cek Out. Setelah pengguna menekan tombol Cek Out maka akan muncul form Login apabila pengguna belum melakukan proses login. Jika pengguna sudah login ke dalam sistem BookStore.Inc maka akan langsung ditampilkan halaman Order. Karena penulis belum melakukan proses login maka setelah menekan tombol Proceed to check out di form keranjang belanja, muncul tampilan form login Berikut ini adalah gambar form Login.



Gambar 5.10 Form Login

Pada uji coba ini penulis mengisikan username "tommy" dan password "mastommy" yang telah didaftarkan pada proses pendaftaran anggota diatas. Setelah penulis menekan tombol Login maka kemudian mucul tampilan form Order. Apabila pengguna salah dalam mengisikan username dan password maka pengguna tidak dapat melanjutkan ke form Order ini. Berikut ini adalah gambar halaman order yang muncul setelah penulis berhasil melakukan proses login.



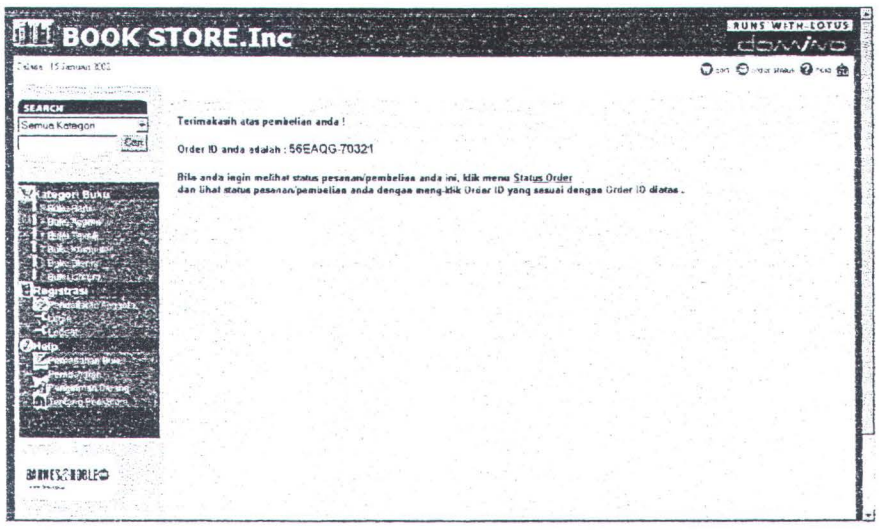
Gambar 5.11. Halaman Order

Pada form diatas informasi pelanggan seperti nama dan alamat ditampilkan secara otomatis. Demikian juga dengan buku yang akan dibeli berserta jumlah totalnya.Informasi pelanggan diperoleh dengan cara look up ke database names.nsf yang digunakan untuk menyimpan data pelanggan.

Proses selanjutnya adalah validasi pembayaran. Aplikasi ini memiliki dua macam cara pembayaran, yaitu dengan cara transfer ke rekening bank dan dengan menggunakan kartu kredit. Untuk validasi kartu kredit, pada aplikasi ini dibuat

satu simulasi validasi kartu kredit. Validasi ini dilakukan dengan membandingkan data kartu kredit yang diisikan pengguna dengan data yang tersimpan di database.

Penulis memilih cara pembayaran lewat kartu kredit. Penulis mengisikan data nomor kartu kredit dan passwordnya, yaitu dengan nomor "2002" dan password "2002". Untuk diketahui informasi kartu kredit ini disimpan pada database BookStore.Inc, dimana data yang disimpan yaitu nomor kartu, password, dan saldo. Pada saat penulis menggunakan kartu kredit ini, saldo yang masih tersisa adalah sebesar Rp 9.660.000,00. Setelah penulis menuliskan nomor kartu dan password diatas, kemudian penulis menekan tombol Order. Beberapa detik kemudian, muncul halaman konfirmasi pembelian berhasil yang berisi ucapan terimakasih kepada pelanggan dan Order ID dari pembelian pelanggan yang berguna untuk melihat status dari pembelian itu, seperti pada gambar dibawah ini.



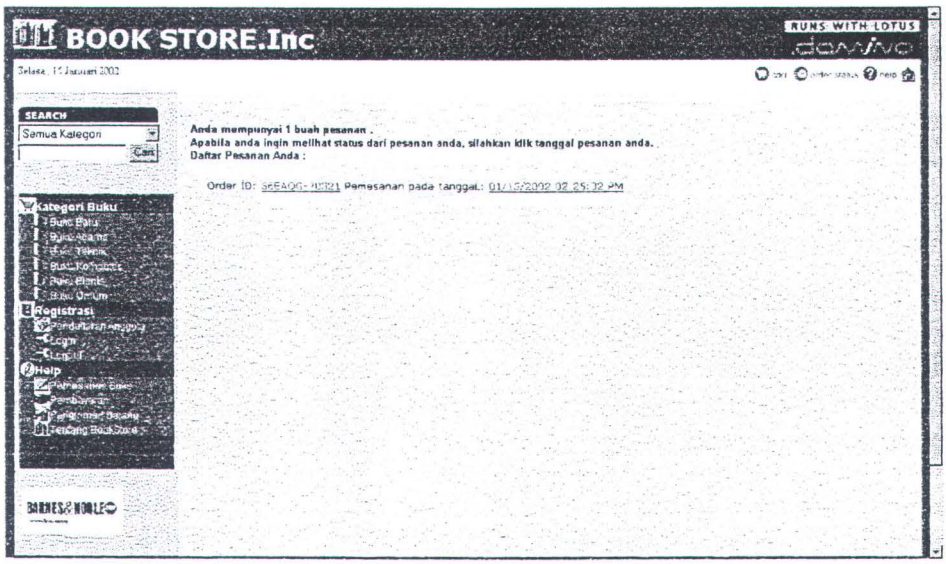
Gambar 5.12. Halaman Konfirmasi Pembelian Berhasil

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Setelah proses diatas penulis melihat data Kartu Kredit penulis dengan nomor 2002, ternyata saldo dari kartu kredit itu telah berkurang menjadi Rp. 9.320.000,00, berkurang Rp. 340.000 sesuai dengan harga buku XML for ASP.Net Developers yang dibeli oleh penulis. Dengan demikian proses validasi kartu kredit telah benar-benar berhasil dijalankan. Demikian juga dengan jumlah buku dari XML for ASP.Net Developers telah berkurang dari 92 menjadi 91. Hal ini membuktikan bahwa proses update buku secara otomatis dapat berjalan dengan baik.

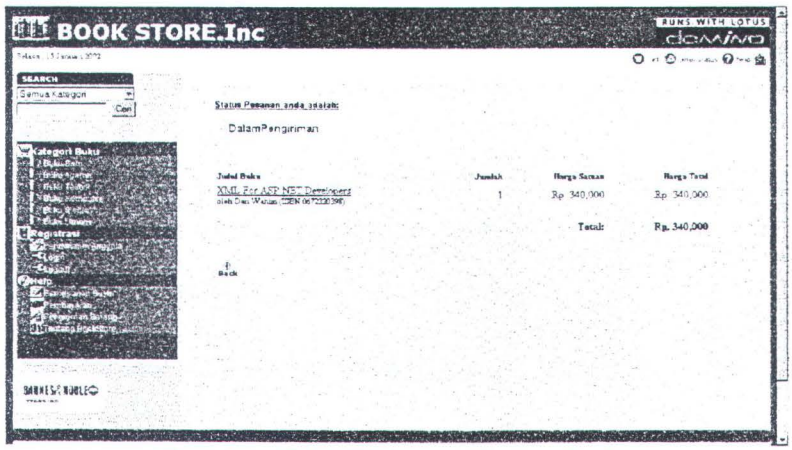
Apabila pelanggan salah dalam mengisikan nomor kartu kredit dan password, maka pelanggan tidak dapat meneruskan pembelian yang berarti jumlah buku yang telah dipilih akan tetap seperti semula. Apabila pelanggan tidak memiliki kartu kredit, pelanggan dapat memilih alternatif kedua, yaitu dengan transfer bank. Jika pilihan ini dipilih pelanggan, maka pelanggan dapat meneruskan pembelian, dan juga akan muncul halaman konfirmasi bahwa pemesanan telah berhasil dilakukan oleh pelanggan. Perbedaan dari penggunaan kartu kredit dan transfer bank adalah, jika menggunakan kartu kredit maka status dari ordernya adalah "Dalam Pengiriman", dan jika menggunakan transfer bank, status order pelanggan adalah "Menunggu Pembayaran". Apabila dalam jangka waktu tertentu pelanggan belum membayar maka pesanan(order) pelanggan dianggap dibatalkan dan status pesanannya menjadi "Dibatalkan". Status Order sendiri memiliki empat nilai, yaitu MenunggуPembayaran, Dalam Pengiriman, Dibatalkan,dan SudahTerkirim.

Apabila pengguna ingin melihat status dari pesanan (order) yang telah dilakukan maka pengguna dapat meng-klik link Status Order pada halaman konfirmasi pembelian berhasil atau dari link Order Status pada navigator kanan atas. Berikut ini adalah gambar tampilan yang muncul setelah penulis meng-klik link Status Order.



Gambar 5.13. Halaman Daftar Order Pengguna

Pada halaman ini semua order yang pernah dilakukan oleh pelanggan akan ditampilkan menurut OrderID dan tanggal pembelian/pemesanan. Pada halaman diatas penulis hanya mempunyai satu buah order. Pelanggan dapat melihat status dari pesanan yang ditampilkan dihalaman daftar order. Berikut ini adalah halaman Status Pesanan yang muncul setelah penulis meng-klik Order ID penulis.



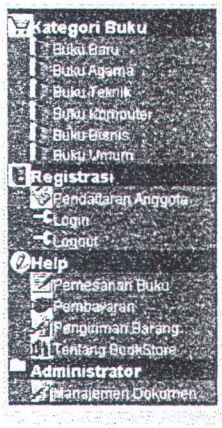
Gambar 5.14. Halaman Status Order

Pada gambar diatas tampak bahwa status order penulis adalah Dalam Pengiriman. Hal ini karena penulis memakai cara pembayaran lewat kartu kredit. Jika penulis memakai cara transfer bank maka status order diatas adalah Menunggu Pembayaran.

5.2.2 Proses Manajemen Dokumen

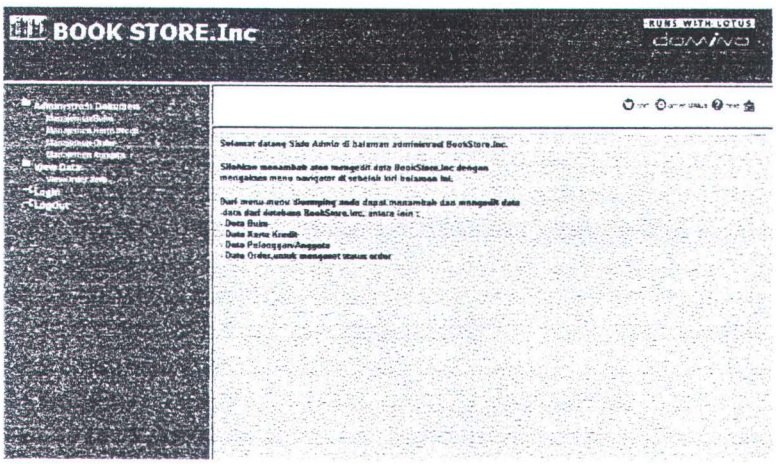
Untuk melakukan manajemen dokumen, pengguna harus login sebagai administrator. Jika pengguna belum login sebagai administrator situs ini, dalam hal ini yang memiliki role sebagai [SiteManager], maka menu manajemen dokumen tidak akan kelihatan.

Pada ujicoba ini oenulis login sebagai administratro dengan mengisikan username "Sisfo Admin" dan dengan password "admindomino". Username ini juga memiliki hak akses sebagai manager/ administrator pada Lotus Domino Server.



Gambar 5.15. Menu Manajemen Dokumen pada navigator kiri

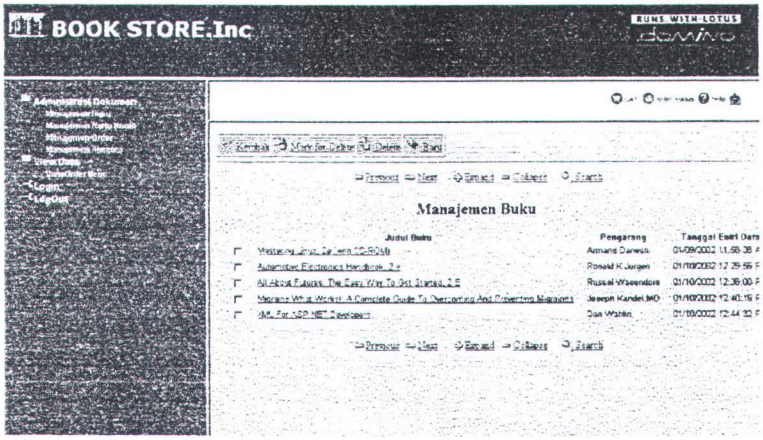
Setelah penulis meng-klik menu Manajemen Dokumen maka muncul halaman web baru yang berisi halaman administrator. Pada halaman ini ada menu Administrasi Dokumen yang terdiri dari menu manajemen buku,manajemen kartu kredit, manajemen order, dan manajemen pelanggan, dan juga menu View Order Item yang akan menampilkan semua data keranjang belanja pengguna baik yang sudah dibeli maupun yang belum dibeli. Dibawah ini adalah gambar halaman administrator.



Gambar 5.16. Halaman Administrator

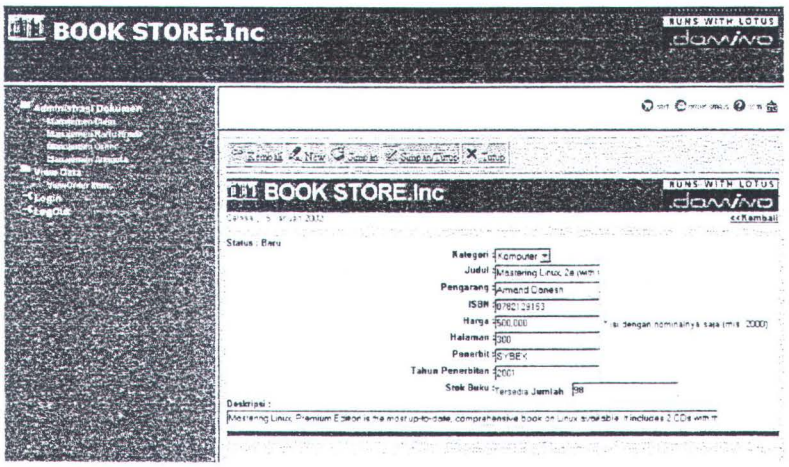
5.2.2.1 Manajemen Buku

Berikut ini adalah tampilan utama manajemen buku. Halaman ini berisi semua buku yang ada di database BookStore.Inc



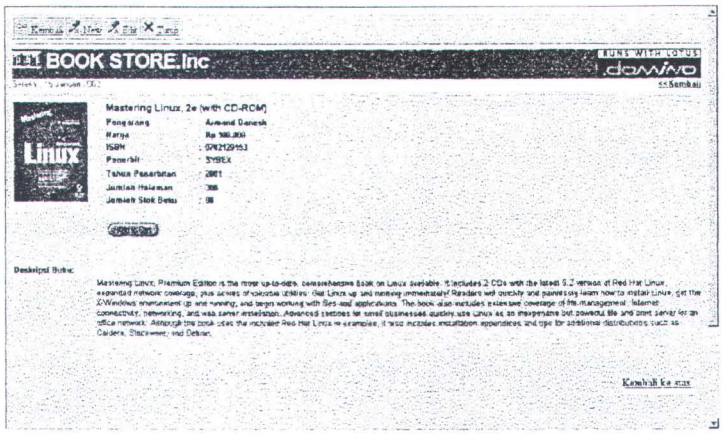
Gambar 5.17. Halaman Manajemen Buku

Pada ujicoba ini penulis mengedit buku "Mastering Linux.2 e" dengan mengubah harga buku dari Rp. 450.000 menjadi Rp. 500.000. Berikut ini adalah tampilan halaman edit buku Mastering Linux, 2e.



Gambar 5.18. Halaman Edit Buku

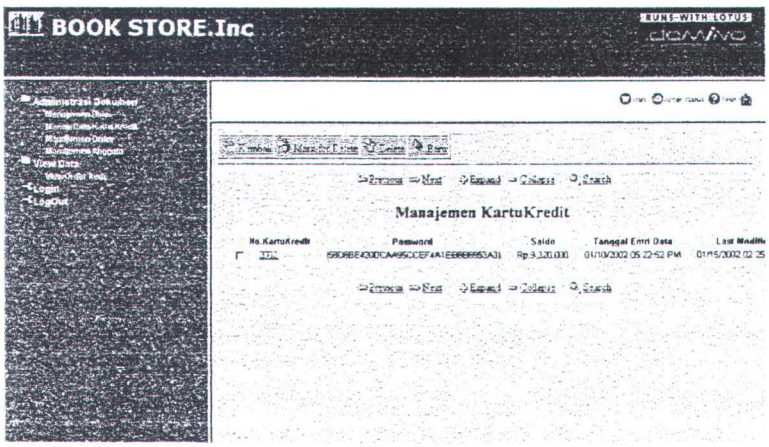
Perubahan yang terjadi dapat dilihat pada deskripsi buku Mastering Linux,2 e, seperti pada gambar dibawah ini.



Gambar 5.19. Halaman Deskripsi buku Mastering Linux,2e

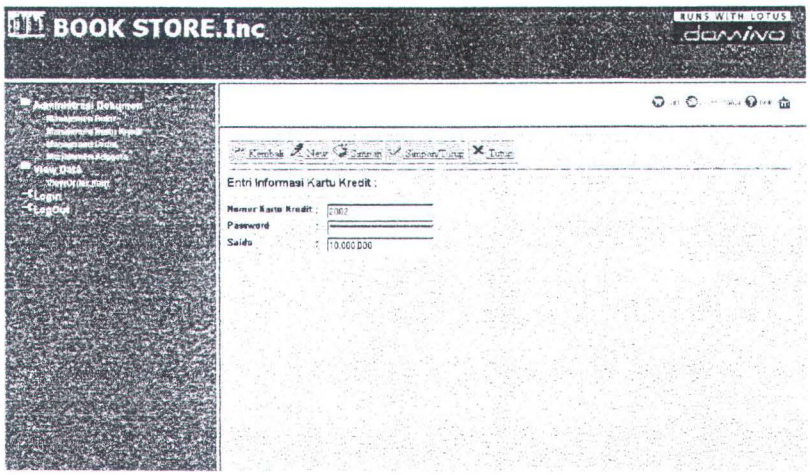
5.2.2.2 Manajemen Kartu Kredit

Berikut ini adalah tampilan utama manajemen kartu kredit. Halaman ini berisi semua data kartu kredit yang ada didatabase BookStore.Inc.



Gambar 5.20. Halaman Utama Manajemen Kartu Kredit

Pada uji coba ini penulis menguah data saldo kartu kredit dengan nomor 2002 dari saldo semula sebesar Rp.9.320.000 menjadi Rp. 10.000.000. Berikut ini adalah halaman edit kartu kredit.

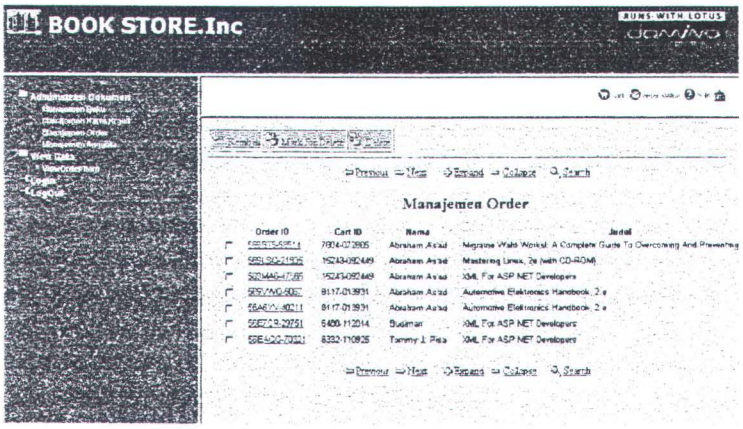


Gambar 5.21. Halaman Edit Kartu Kredit

Perubahan yang terjadi dapat dilihat pada halaman utama kartu kredit diatas.

5.2.2.3 Manajemen Order

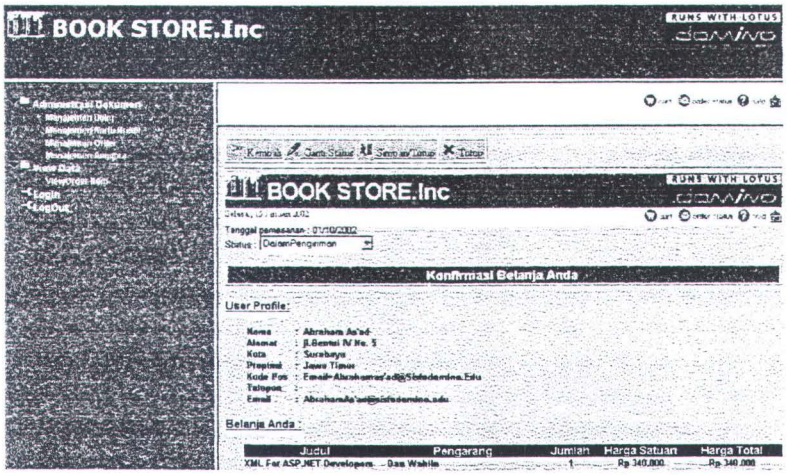
Berikut ini adalah tampilan utama manajemen order. Halaman ini berisi semua order yang telah dipesan atau dibeli oleh pelanggan.



Gambar 5.22. Halaman Utama Manajemen Order

Pada ujicoba ini penulis mengubah status order dengan order ID "569MA6-47565" atas nama Abraham As'ad yang sebelumnya berstatus "Menunggu Pembayaran" menjadi "Dalam Pengiriman".

Dibawah ini adalah gambar halaman edit order.

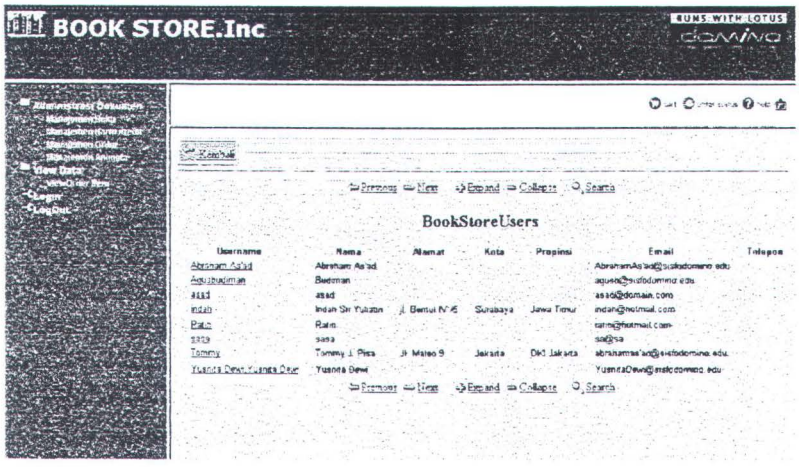


Gambar 5.23. Halaman Edit Order

Untuk melihat hasil perubahan yang terjadi dapat dilihat pada halaman utama manajemen order diatas.

5.2.2.4 Manajemen Pelanggan

Berikut ini adalah tampilan utama manajemen pelanggan. Halaman ini berisi semua pelanggan yang terdaftar di BookStore.Inc.



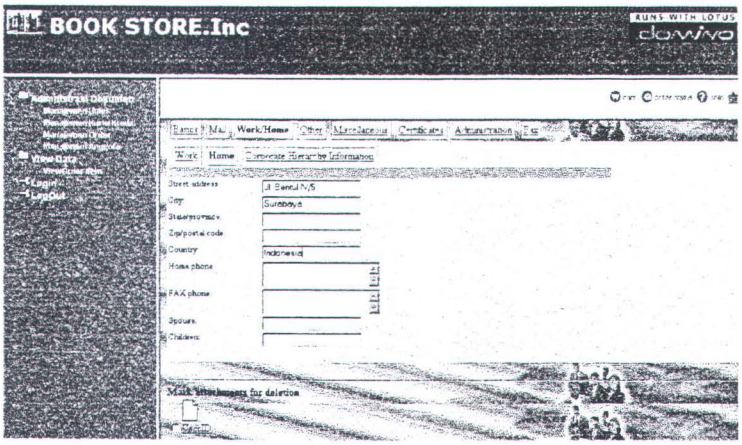
Gambar 5.24. Halaman Utama Manajemen Pelanggan

Pada ujicoba ini penulis mengubah alamat dari username Abraham As'ad yang sebelumnya kosong menjadi "Jl. Bentul IV/5", kota menjadi "Surabaya" dan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

Negara menjadi "Indonesia". Karena penulis menggunakan login "sisfo admin" yang juga merupakan administrator dari Domino Server dimana database names.nsf tersimpan, maka penulis tidak harus login lagi.

Dibawah ini adalah gambar halaman edit pelanggan.



Gambar 5.25. Halaman Edit Pelanggan

Untuk melihat hasil perubahan yang terjadi dapat dilihat pada halaman utama manajemen pelanggan diatas.

Dari ujicoba manajemen dokumen ini penulis dapat melakukan semua proses dengan baik, hanya untuk proses manajemen buku, penulis/admin tidak dapat menambah dan mengedit data yang berupa gambar, dalam hal ini gambar buku, lewat web karena Lotus Notes/Domino 5 tidak mendukung import gambar lewat web. Lotus Notes/Domino 5 dapat memasukkan file gambar sebagai *attachment* tetapi tidak dapat memasukkan gambar itu sendiri kedalam database lewat web.Hal ini hanya dapat dilakukan dengan Notes Client.

Untuk manajemen pelanggan, karena data pelanggan disimpan di database tersendiri yaitu names.nsf yang merupakan database default dari Lotus Notes yang menyimpan daftar pengguna dari Lotus Notes, maka untuk melakukan

pengubahan data pelanggan, administrator situs BookStore.Inc harus login sebagai Adminitrator Notes.

Setelah melakukan semua uji coba ini rata-rata waktu yang dibutuhkan untuk melakukan proses yang menjalankan agent yang memiliki proses yang agak rumit seperti pada proses Pendaftaran Anggota, dan proses Submit Order(termasuk didalamnya adalah Validasi Pembayaran) pada form Order adalah berkisar 6-8 detik. Sedangkan waktu untuk loading halaman yang berfungsi sebagai tampilan seperti halaman katalog , deskripsi buku adalah berkisar antara 1-3 detik. Proses pencarian buku berjalan cukup cepat antara 2-4 detik.

Sistem security untuk form-form yang *restricted* hanya untuk pelanggan seperti Order Status,dan form Order dapat berjalan dengan baik. Pengguna yang tidak memiliki username dan password tidak dapat membuka form-form tersebut. Demikian juga untuk form yang khusus untuk administrator, pengguna biasa tidak dapat membuka apalagi menjalankan form tersebut. Masa *idle* untuk username yang login adalah 1 jam, jadi jika pelanggan yang telah login meninggalkan situs ini selama satu jam maka sistem akan meminta pelanggan untuk login lagi.

BAB VI
PENUTUP

BAB VI

PENUTUP

Bab ini menjelaskan kesimpulan dan kemungkinan pengembangan lebih lanjut dari aplikasi toko buku online BookStore.Inc.

6.1 Kesimpulan

Kesimpulan yang didapat dari tugas akhir ini dalah :

- Pembuatan halaman web yang statis dapat dengan mudah dilakukan dengan menggunakan elemen perancangan dari Lotus Domino Designer yaitu page dan frameset. Halaman web dapat dipercantik tampilannya dengan menggunakan Html yang dapat ditulis langsung di page dengan memakai fasilitas passthru html-nya Domino Designer. Dan juga grafik, obyek OLE seperti file musik, animasi dapat dimasukkan ke halaman web dengan mudah.

- Adanya elemen Outline dan Navigator , subform, dan shared field di Domino Designer mempermudah pembuatan navigasi situs.

- Pembuatan aplikasi interaktif situs dengan pengguna dapat menggunakan bahasa pemrograman Formula,untuk perintah sederhana, Lotus Script untuk pembuatan operasi yang lebih sulit dan kompleks, serta Java dan Javascript.

- Untuk menampilkan semua dokumen dari suatu form yang mengandung gambar, terdapat kesulitan karena keterbatasan *view*[1] yang tidak dapat

[1] View adalah elemen yang dipunyai Lotus untuk menampilkan isi dari form (dokumen dari form)

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH – NOPEMBER

menampilkan gambar yang tersimpan didokumen, sehingga harus diperlukan trik tersendiri untuk menampilkannya. Dalam kasus penulis adalah untuk pembuatan tampilan katalog buku, penulis menyimpan semua file gambar buku dalam satu direktori dan menggunakan Html di view katalog buku untuk menampilkannya.

- Untuk pembuatan proses pendaftaran dan validasi user diperlukan proses yang agak panjang dan rumit karena agar server bisa mengenali username dan password pengguna maka username dan password itu harus disimpan di database tersendiri yang disediakan oleh Lotus.

- Adanya tingkat-tingkat pengamanan yang bisa diterapkan di Lotus Domino mempermudah penerapan keamanan(*security*) aplikasi yang dibuat mulai dari tingkat jaringan, server sampai di tingkat dokumen. Dalam tugas akhir ini tingkat pengamanan yang diterapkan adalah pada database dengan pengesetan ACL dan Role, pada form, pada view, dan dokumen.

- Dalam aplikasi ini, apabila ada dua user atau lebih yang login pada satu workstation, keranjang belanja yang dipakai dan disimpan dalam cookie adalah satu. Sehingga jika salah satu pengguna melakukan order buku, maka isi keranjang belanja akan dihapus dan pengguna yang lain harus memasukkan buku yang diinginkan ke dalam keranjang lagi.

6.2 Kemungkinan Pengembangan

Dalam pembuatan Tugas Akhir ini terdapat beberapa kemungkinan pengembangan yang perlu diperhatikan untuk pengembangan lebih lanjut dari apliaksi ini.

- Untuk menangani barang yang lebih banyak, aplikasi ini dapat menggunakan database relasional seperti Oracle untuk penyimpanan data dalam jumlah besar.
- Penggunaan SSL untuk keamanan apabila situs ini akan dikembangkan menjadi lebih besar.
- Menggunakan fasilitas validasi kartu kredit yang diberikan oleh vendor kartu kredit, misalnya MasterCard.
- Dapat ditambah fasilitas create mail account di situs ini, dengan memanfaatkan fungsi Domino Server sebagai mail server.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[BOU-00] Boutault, Robert, Isaac Allotey-Pappoe, Peter EiBak, Eric Feng, Jennifer Minge, *Developing E-business Applications Using Lotus Domino for AS/400*, IBM Corporation , Dept. HZ8 Building 678, P.O. Box 12195, Research Triangle Park, NC 27709-2195, 2000

[INT-97] International Technical Support Organization, *Developing Web Applications Using Lotus Notes Designer for Domino 4.6*, IBM Corporation , Dept. HZ8 Building 678, P.O. Box 12195, Research Triangle Park, NC 27709-2195, 1997

[JOS-01] Joseph Denny Hartanto, Tugas Akhir - *Workflow Automation Pada Divisi Kurir PT XYZ Dengan Aplikasi SistemLotus Notes*, 2001.

[LAR-98] Larman, Craig, *Applying UML and Patterns: Introduction to Object-Oriented Analysis and Design*, Prentice-Hall,Inc., 1998.

[LOT-00] Lotus Development Corporation, *Lotus Inside Notes – The Architecture of Notes and Domino Server*, 55 Cambridge Parkway, Cambridge, 2000.

[LOT-99] Lotus Development Corporation, *Lotus Education*, 55 Cambridge Parkway, Cambridge, 1999.

[LUT-00] Luthfi, Tugas Akhir - *Perancangan dan Pembuatan Perangkat Lunak Sistem Informasi Sumber Daya Manusia di ARCO Indonesia*, 2000

[TKA-96] Tkach, Daniel, Walter Fang, Andrew So, *Visual Modeling Technique: Object Technology Using Visual Programming,* Addison Wesley Inc., 1996.

[QUA-98] Quatrani, Terry, *Visual Modeling with Rational Rose and UML,* Addison Wesley Longman, Inc., 1998.

[YUL-01] Yuli Prapto Handoyo, *Pembuatan Prototipe Virtual Store dengan menggunakan Objek Terdistribusi*, 2001.

16.213/H/02

TUGAS AKHIR

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE



RSIf
005.1
Dam
p-1
2001

Disusun oleh :

Fitri Damayanti
2695100069

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2001

PERPUSTAKAAN
I T S
Tgl. Terima 03/01/02
Terima Dari H
No. Agenda Prp. 21-4298

ITS

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui,

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Ir. Arif Djunaidy, M.Sc.	**Rully Soelaiman, S.Kom**
NIP. 131 633 403	NIP. 132 085 802

Surabaya
Agustus, 2001

Bismillahhirrohmanirrohim

"Hai orang-orang yang beriman, bertakwalah kepada Allah sebenar-benar takwa kepada-Nya, dan janganlah sekali-kali kamu mati melainkan dalam keadaan beragama Islam (QS:3.102)"

"Dan hendaklah ada diantara kamu segolongan umat yang menyeru kepada kebajikan, menyuruh kepada yang ma'ruf dan mencegah dari yang munkar, merekalah orang-orang yang beruntung (QS:3.104)"

Kupersembahkan Untuk

Ayahanda, Ibunda dan Suami Tercinta

KATA PENGANTAR

KATA PENGANTAR

Alhamdulillahi Rabbil'aalamin segala puji bagi Allah serta Sholawat dan salam bagi Nabi Muhammad atas berkat dan rahmat Allah SWT telah diselesaikan Tugas Akhir

Tugas Akhir ini tidak akan mampu diselesaikan tanpa dukungan dari semua pihak baik moril maupun materiil. Pada kesempatan ini saya menyampaikan terima kasih pada semua pihak yang telah memberikan dukungan dan dorongan.

- Bapak Dr.Ir.Arif Djunaidy,M.Sc selaku dosen pembimbing dan ketua jurusan Teknik Informatika, yang selalu meluangkan waktu untuk memberikan bimbingan serta dorongan dan motivasi dalam menyelesaikan tugas akhir ini.
- Bapak Rully Soelaiman,S.Kom selaku dosen pembimbing yang telah memberikan ide, bimbingan dan bantuan selama pengerjaan tugas akhir ini.
- Ibu Nanik Suciati,M.Kom selaku dosen wali.
- Seluruh staf pengajar Teknik Informatika atas ilmu yang diberikan selama masa perkuliahan.
- Staf tata usaha jurusan Teknik Informatika atas bantuannya.
- Ayahanda tersayang yang selalu memberikan kasih sayang, bimbingan, dorongan, dan do'a.
- Ibunda tercinta yang tidak henti-hentinya menyayangi, mendo'akan dan memberi semangat serta nasehat dengan ikhlas tanpa pamrih.
- Sang suami tersayang dan tercinta yang dengan setia mendengarkan keluh-kesah dan omelanku,yang memberikan motivasi dan do'a, dan selalu setia dalam suka maupun duka.
- Bapak dan Ibu Muin atas kasih sayang , nasehat dan bimbingannya.
- Adik-adikku Ahmad dan Wawan atas bantuan dan do'anya. Rajin-rajinlah beribadah dan belajar semoga menjadi anak yang Sholeh. Amin.

- Naim dan Yuli atas bantuan dan motivasinya. Cepetan Tugas Akhirnya dikerjakan. Semoga cepet nyusul (lulus dan mendapat jodoh).
- Anak-anakku Lyha dan Ita yang setia mendengarkan keluh-kesahku dan selalu memberi motivasi, semoga cepet lulus.
- Mbah Imam atas ilmu yang diberikan, serta bantuan dan motivasinya.
- Mukib, Yuni, Eny, Isty, Hida, Izzah, Saroh, Elok atas bantuan dan kebersamaan kita selama ini.
- Mbak Zaenab, Mbak Uswah, Mbak Niswah, Mbak Nurul yang telah membimbing dan mengarahkan untuk menuju jalan yang lurus dan benar.
- Pram, Eko, Fauzi, Anib, Salim, Hardi, Gandi dan teman-temanku C-0B lainnya atas bantuannya.
- Teman-teman kostku GK-70 atas bantuan dan motivasinya .
- Pihak-pihak yang ikut membantu terwujudnya tugas akhir ini.

Tugas akhir ini belum sampai pada tahap sempurna, maka saran dan kritik dari pembaca sangat diharapkan. Semoga karya ini dapat memberikan manfaat dan dicatat sebagai ilmu yang bermanfaat.

Amin ya Robbal Alamiin.

Surabaya, 7 Agustus 2001

Fitri Damayanti

ABSTRAKSI

ABSTRAKSI

Dalam perkembangan teknologi komputer dewasa ini, banyak aplikasi yang menggunakan citra sebagai sumber informasi maupun dalam hal komunikasi data seringkali menggunakan citra sebagai sarana komunikasi, secara khusus adalah citra wajah. Sistem pengenalan wajah merupakan sistem yang mudah digunakan dan lebih berpotensi untuk tidak dapat ditembus.

Dalam tugas akhir ini, dirancang dan dibuat satu sistem pengenalan wajah dengan masukan berupa citra wajah yang berukuran tertentu, dengan menggunakan metode Fisherface. Sistem ini terdiri dari dua tahapan utama, yaitu tahap pelatihan dan tahap pengenalan. Tahap pelatihan dimaksudkan untuk melatih citra wajah yang telah ada didalam basis data untuk dicari nilai Transformasi Fisher's Linear Discriminant (FLD) dan bobot wajah pelatihan. Sedangkan tahap pengenalan akan mengenali wajah berdasarkan informasi hasil pelatihan. Uji coba dilakukan dengan menggunakan dua tipe data pelatihan, yaitu: data pelatihan terurut dan data pelatihan terkendali. Data pelatihan terurut diambil dari data pelatihan sesuai dengan nomor urutan datanya tanpa memandang variasi data yang ada. Data pelatihan terkendali dipilih secara manual dan mempertimbangkan variasi data yang ada tanpa memandang urutan indeks data.

Hasil uji coba dengan data pelatihan terurut menunjukkan 80% sedangkan dengan data pelatihan terkendali menunjukkan 88%. Kemampuan sistem pengenalan wajah ini sangat tergantung dari jumlah variasi data pelatihan yang digunakan dan juga tergantung pada kemiripan wajah tiap-tiap orang. Jumlah variasi yang memadai akan memberikan tingkat keberhasilan pengenalan wajah tinggi bila dibandingkan dengan variasi data pelatihan yang kurang memadai. Sistem yang telah berhasil dibuat dapat dimanfaatkan sebagai dasar bagi pengembangan pengaman elektronik yang berbasis pada pengenalan wajah.

DAFTAR ISI

DAFTAR ISI

Kata Pengantar i

Abstraksi iii

Daftar Isi iv

Daftar Gambar vii

Daftar Tabel viii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Perumusan Masalah 2

1.3 Tujuan dan Manfaat 3

1.4 Batasan Masalah 3

1.5 Metodologi Pengerjaan Tugas Akhir 3

1.6 Sistematika Penulisan 5

BAB II PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE 6

2.1 Teori Dasar 6

2.1.1 Wajah Dipandang Sebagai Sebuah Vektor 6

2.1.2 Ruang Lingkup Gambar 7

2.2 Transformasi Matriks 8

2.2.1 Matriks Normalisasi 9

2.2.2 Matriks Kovariansi 9

2.3 Nilai Karakteristik dan Vektor Karakteristik 10

2.3.1 Definisi Nilai Karakteristik dan Vektor Karakteristik 11

2.3.2 Penentuan Nilai Karakteristik dan Vektor Karakteristik 11

2.3.3 Transformasi Kemiripan 12

2.3.4 Metode Cyclic Jacobi untuk Matriks Simetris 14

2.4 Metode Fisherface 15

2.4.1 Algoritma Metode Fisherface 18
2.4.2 Matriks Bobot Wajah 23
2.4.2.1 Matriks Bobot Wajah Pelatihan 23
2.4.2.2 Matriks Bobot Wajah Uji Coba 24

BAB III PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK 25
3.1 Tujuan dan Sasaran Sistem 25
3.2 Perancangan Perangkat Lunak 25
3.2.1 Perancangan Data 26
3.2.1.1 Data Masukan 26
3.2.1.2 Data Proses 26
3.2.1.3 Data Keluaran 29
3.2.2 Perancangan Proses 29
3.2.3 Perancangan Antarmuka 34
3.3 Pembuatan Perangkat Lunak 35
3.3.1 Implementasi Data 35
3.3.2 Implementasi Proses 37
3.3.3 Implementasi Antarmuka 41

BAB IV UJI COBA DAN EVALUASI SISTEM 43
4.1 Lingkungan Uji Coba 43
4.2 Data Uji Coba 44
4.3 Uji Coba dan Evaluasi Hasil 44
4.3.1 Uji Coba dengan Data Pelatihan terurut 46
4.3.2 Uji Coba dengan Data Pelatihan Terkendali 48
4.3.3 Evaluasi Hasil Uji Coba 50
4.3.4 Perbandingan Dengan Sistem Lain 51

BAB V KESIMPILAN 53

DAFTAR PUSTAKA 54
LAMPIRAN A : DATA PELATIHAN A-1
LAMPIRAN B : PETUNJUK PEMAKAIAN PERANGKAT LUNAK B-1

DAFTAR GAMBAR

DAFTAR GAMBAR

Gambar 2.1	Formasi vektor dari sebuah gambar
Gambar 2.2	Basis ruang lingkup gambar
Gambar 3.1	DFD level 0 subsistem pelatihan
Gambar 3.2	DFD level 1 subsistem pelatihan
Gambar 3.3	DFD level 2 subsistem pelatihan Metode PCA
Gambar 3.4	DFD level 2 subsistem pelatihan Metode FLD
Gambar 3.5	DFD level 0 subsistem pengenalan wajah
Gambar 3.6	DFD level 1 subsistem pengenalan wajah
Gambar 3.7	Menu utama
Gambar 3.8	Tampilan sebagian wajah pelatihan
Gambar B.1	Tampilan dari sebagian wajah training
Gambar B.2	Tampilan matriks hasil dari proses training
Gambar B.3	Tampilan hasil dari testing
Gambar B.4	Tampilan matriks dari hasil proses testing

DAFTAR TABEL

DAFTAR TABEL

Tabel 4.1	Hasil uji coba pengenalan wajah dengan data pelatihan terurut
Tabel 4.2	Hasil uji coba pengenalan wajah dengan data pelatihan terkendali
Tabel 4.3	Perbandingan keberhasilan pengenalan berdasarkan metode pemilihan data pelatihan yang digunakan
Tabel A.1	Data pelatihan

BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Sistem pengaman elektronik mengalami kemajuan pesat dalam beberapa dekade terakhir ini. Perkembangan ini didukung oleh kemajuan teknologi dibidang hardware, khususnya komputer. Dukungan yang diberikan komputer seperti kemudahan dibidang perancangan, simulasi, dan implementasi sistem pengaman elektronik tersebut.

Aplikasi sistem pengenalan biometrik (sistem pengenalan dengan penggunaan ciri khas pada diri manusia, seperti wajah, sidik jari, dan telapak tangan) untuk sistem pengaman mendapatkan perhatian yang serius akhir-akhir ini. Perhatian ini muncul karena ciri-ciri yang digunakan tersebut merupakan ciri yang unik untuk setiap manusia dan tidak dapat dipalsukan. Diantara ketiga ciri yang disebutkan diatas, ciri wajah merupakan ciri yang lebih kompleks karena banyaknya informasi yang terdapat padanya. Informasi-informasi itu meliputi bentuk wajah, jarak kedua mata, warna kulit, panjang dan lebar alis mata, dan informasi lainnya. Dengan demikian, penggunaan wajah sebagai kode akses untuk sistem pengaman elektronik lebih berpotensi untuk tidak dapat ditembus dibandingkan dengan penggunaan ciri yang lain.

Tugas akhir ini berisi tentang sistem pengenalan wajah dengan menggunakan metode Fisherface. Metode ini mampu menghasilkan presentase

total tingkat pengenalan yang lebih tinggi. Hal ini dapat terjadi karena dengan menerapkan metode Fisherface untuk mereduksi dimensi citra wajah ke dimensi ciri sebagai data masukan sistem, data-data pada citra wajah yang dapat mengganggu tidak ikut diproses lagi. Dengan demikian hanya ciri-ciri wajah yang penting saja yang disimpan dan diolah sistem untuk melakukan pengenalan terhadap objek wajah. Selain itu bila terdapat variasi ciri wajah yang akan dikenali, seperti ekspresi, tetap mampu dikenal dengan baik. Dapat dikatakan bahwa metode Fisherface yang mereduksi dimensi data masukan sistem dapat meningkatkan kinerja sistem, terutama dengan mempercepat proses pelatihan dan meningkatkan prosentase tingkat pengenalan objek.

1.2 PERUMUSAN MASALAH

Masalah utama dari pengenalan wajah adalah diberikan sebuah gambar atau citra yang berisi satu atau banyak pas foto wajah orang, kemudian gambar tersebut diidentifikasi masing-masing individu yang ada dengan menggunakan basis data dari wajah yang sudah ada. Kesulitan yang muncul dalam pengenalan pola adalah besarnya variasi pada input sistem. Perbedaan-perbedaan posisi dan arah pandangan pada input yang ada dari wajah orang yang sama harus tetap dikenali sebagai wajah orang yang sama pula. Metode Fisherface akan mereduksi data masukan sistem yang berupa dimensi citra wajah ke dimensi ciri, data yang mengganggu tidak ikut diproses. Setelah proses pelatihan, kemudian proses uji coba dilakukan sebuah foto yang tidak dikenal diinputkan ke dalam program dan dibandingkan dengan semua basis data.

1.3 TUJUAN DAN MANFAAT

Tujuan dari tugas akhir ini adalah mencoba mengenali suatu pas foto yang diinputkan dalam program dari pas foto yang tidak dikenal sebelumnya berdasarkan basis data yang ada dengan menggunakan metode Fisherface.

Manfaat yang bisa diperoleh dari pembuatan sistem ini adalah dasar bagi pembuatan sistem yang lebih besar dalam bidang pembuatan sistem keamanan di berbagai bidang.

1.4 BATASAN MASALAH

Agar pembahasan menjadi terfokus, dibuat ruang lingkup pembahasan sebagai berikut :

- Foto yang diinputkan harus merupakan foto oval dari wajah saja (pas foto)
- Foto merupakan gambar dengan skala keabuan (gray scale) dan mempunyai suatu ukuran tertentu, yaitu : 92 * 112 piksel.
- Foto yang dipergunakan berekstensi BMP.

1.5 METODOLOGI PENGERJAAN TUGAS AKHIR

Pembuatan Tugas Akhir ini akan dilaksanakan dengan metodologi sebagai berikut :

- Studi Literatur

  Studi literatur dilakukan dengan mencari dan mempelajari literatur-literatur yang berkaitan dengan *metode Fisherface* , teori-teori yang berhubungan

dengan sistem yang akan dibangun, desain sistemnya, dan bahasa pemrogramannya.

- Perancangan Sistem Perangkat Lunak

  Melakukan perancangan sistem mulai dari Input/output, *pre-processing* data, teknik pengolahan data, perancangan struktur data, dan perancangan struktur pemrogramannya

- Pembuatan Perangkat Lunak

  Setelah semua rancangan aplikasi tersedia maka selanjutnya diimplementasikan kedalam *source code* sesuai bahasa pemrograman yang dipilih.

- Uji Coba dan Evaluasi

  Pengujian suatu perangkat lunak dilakukan guna mengetahui permasalahan yang terjadi setelah perangkat lunak tersebut selesai dikerjakan.

- Perbaikan dan Penyempurnaan Perangkat Lunak

  Apabila dalam proses pengujian ditemukan kesalahan tau kekurangan maka perlu dilakukan perbaikan dan penyempurnaan. Setelah itu dilakukan lagi pengujian sampai aplikasi bisa benar-benar sempurna dan sudah layak dipakai sesuai dengan tujuannya.

- Penulisan Tugas Akhir

  Pada tahap akhir dibuat suatu dokumentasi lengkap tentang aplikasi yang telah dibuat, dalam hal ini dibuat dalam bentuk Buku Tugas Akhir.

1.6 SISTEMATIKA PENULISAN

Uraian sistematika penulisan pada tugas akhir ini bertujuan supaya perancangan dan pembuatan perangkat lunak yang dibahas menjadi mudah dipahami, jelas dan sistematis untuk tiap-tiap bab atau sub bahasan. Secara kronologis uraian dalam tugas akhir ini disusun sebagai berikut :

BAB I berisi pendahuluan, menguraikan tentang latar belakang, permasalahan, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan Tugas Akhir.

BAB II berisi pengenalan wajah menggunakan metode Fisherface, yang menguraikan teori-teori dan konsep-konsep dasar mengenai pengenalan wajah dari sebuah pas foto yang diinputkan dalam program berdasarkan basis data yang ada dengan menggunakan metode Fisherface.

BAB III menjelaskan tentang modul-modul perancangan dan pembuatan perangkat lunak yang meliputi prosedur-prosedur atau fungsi-fungsi program.

BAB IV menjelaskan tentang analisis hasil dari implementasi program.

BAB V menjelaskan tentang penutup yang berisi kesimpulan dan kemungkinan pengembangan lebih lanjut.

BAB II
PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE

BAB II
PENGENALAN WAJAH MENGGUNAKAN METODE FISHERFACE

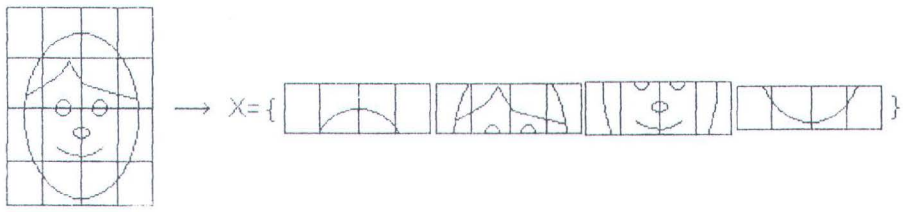
Dalam bab ini akan dibahas konsep-konsep dan teori-teori dasar tentang pengenalan wajah dengan menggunakan metode Fisherface yang berhubungan erat dengan pembuatan sistem perangkat lunak dan implementasi program. Dalam bab ini juga dibahas metode serta algoritma dari metode Fisherface

2.1 TEORI DASAR

Gambar merupakan kumpulan dari angka-angka, yang angka-angka tersebut membentuk sebuah matriks. Sehingga sebelum membahas lebih lanjut akan dikenalkan dulu tentang vektor dan matriks serta rumus-rumus untuk menghitungnya.

2.1.1 Wajah Dipandang Sebagai Sebuah Vektor

Sebuah wajah, yang juga merupakan sebuah gambar, dapat dipandang sebagai sebuah vektor. Apabila lebar dan tinggi gambar adalah w dan h piksel, maka banyaknya komponen dari vektor ini adalah w*h. Setiap piksel dikodekan oleh satu komponen vektor. Konstruksi vektor dari sebuah gambar dibentuk oleh penggabungan sederhana, yaitu baris dari sebuah gambar diletakkan saling bersebelahan dengan baris-baris yang lain, seperti yang terlihat pada gambar 2.1.



Gambar 2.1 Formasi Vektor dari Sebuah Gambar

2.1.2 Ruang Lingkup Gambar

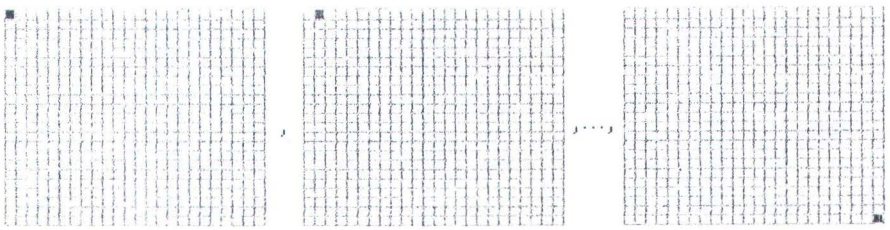
Vektor wajah yang telah dideskripsikan sebelumnya merupakan bagian dari sebuah ruang. Ruang ini adalah sebuah ruang lingkup gambar (*image space*), yaitu ruang dari semua gambar keseluruhan yang mempunyai dimensi w*h piksel. Basis dari ruang lingkup gambar dikomposisikan oleh vektor-vektor berikut, seperti terlihat pada gambar 2.2.

Semua wajah mirip satu sama lain. Mereka semua mempunyai dua mata, satu hidung, satu mulut, dua telinga, dan lain sebagainya yang terletak pada tempat yang sama. Akibatnya semua vektor wajah terletak pada tempat-tempat yang amat berdekatan dalam ruang lingkup gambar.

Oleh karena itu, sebuah ruang lingkup gambar yang penuh bukanlah sebuah ruang yang bagus untuk mendeskripsikan sebuah wajah. Tugas yang akan dipresentasikan di sini adalah bertujuan untuk membangun sebuah ruang lingkup wajah (*face space*) yang lebih dapat mendeskripsikan wajah. Vektor basis dari ruang lingkup wajah disebut sebagai komponen utama (*principal component*).

Dimensi dari ruang lingkup gambar adalah w*h. Tentu saja semua piksel dari sebuah wajah tidak relevan, dan setiap piksel bergantung pada tetangganya.

Sehingga dimensi dari ruang lingkup wajah adalah kurang dari dimensi ruang lingkup gambarnya. Dimensi dari ruang lingkup wajah tidak dapat ditentukan, tetapi hal ini sudah dapat dipastikan bahwa dimensi dari ruang lingkup wajah akan jauh lebih kecil daripada ruang lingkup gambarnya.



Gambar 2.2 Basis Ruang Lingkup Gambar

2.2 TRANSFORMASI MATRIKS

Diberikan sebuah basis data wajah pelatihan yang terdiri dari K gambar. Ukuran gambar diasumsikan terdiri dari N*N piksel, dimana setiap piksel dikodekan dengan menggunakan 8 bit atau 256 tingkat keabuan (*gray level*). Setiap gambar dapat dipandang sebagai sebuah vektor kolom dengan ukuran (N*N)x1.

Misalkan setiap gambar diberi notasi $\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_K$ yang merupakan vektor kolom N^2x1. Bila sebuah gambar dalam wajah pelatihan dinotasikan dalam bentuk matriks, maka akan diperoleh matriks yang berordo N^2xK.

$\Gamma_n = (\Gamma_1 \quad \Gamma_2 \quad \Gamma_3 \quad \ldots \quad \Gamma_K)^T$, dimana T adalah matriks transpose. Sedang isi dari tiap vektor kolom adalah

$$\Gamma_n = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & K & \gamma_{1,K} \\ \gamma_{2,1} & \gamma_{2,2} & K & \gamma_{2,K} \\ M & M & & M \\ \gamma_{N^*N,1} & \gamma_{N^*N,2} & K & \gamma_{N^*N,K} \end{bmatrix}$$

2.2.1 Matriks Normalisasi

Sebuah gambar harus sudah dinormalisasi terlebih dahulu untuk mendefinisikan sebuah lokasi penting dalam gambar tersebut. Normalisasi berpengaruh langsung pada dasar pengenalan wajah apabila semua wajah memiliki latar belakang sama dan variansi antar gambar sebagian besar merupakan sebuah variansi pengukuran fitur. Misalkan :

- $\Gamma = [\gamma_{nk}]$ adalah matriks berordo N^2xK yang mendefinisikan himpunan wajah pelatihan.
- Vektor kolom ke-k dari matriks ini, γ_k, berkorespondensi pada wajah ke-k dari wajah pelatihan.
- Vektor γ_k dinormalisasi dengan rumus :

$$|\gamma_k| = \sqrt{\gamma_k^T \cdot \gamma_k}$$

2.2.2 Matriks Kovariansi

Matriks kovariansi adalah sebuah matriks simetris berordo N^2xN^2 dan merupakan sebuah pengukuran yang tidak hanya pada sebuah variansi, tetapi *ko-variansi* dari vektor-vektor kolom.

- Wajah rata-rata, m_x, merupakan operasi rata-rata sederhana dari K kolom yang merepresentasikan sebuah wajah :

$$m_x = \frac{1}{K} \sum_{n=1}^{K} \Gamma_n$$

- Deviasi merupakan penyimpangan pengamatan terhadap rata-rata wajah :

$$\Phi_n = \Gamma_n - m_x$$

- Misalkan $\Phi_n = \Gamma_n - m_x$ dan A = (Φ_1 Φ_2 ...Φ_K) adalah matriks wajah ternormalisasi, maka matriks kovariansi di atas dapat ditulis menjadi :

  $C_x = AA^T$ adalah matriks *outer product* berordo N^2xN^2, atau

  $L = A^TA$ adalah matriks *inner product* berordo KxK, $K << N^2$.

- Matriks C_x dan L adalah matriks *simetris*, sebab unsur-unsur yang berada pada posisi simetris relatif terhadap diagonal utama adalah sama, sehingga dapat diilustrasikan bahwa $\Phi_p^T\Phi_q = \Phi_q^T\Phi_p$, untuk p,q $\in$ [1, K]. Jadi dapat disimpulkan bahwa, matriks A nxn dikatakan simetris apabila $A^T = A$.

2.3 NILAI KARAKTERISTIK DAN VEKTOR KARAKTERISTIK

Dalam metode Fisherface dihitung juga nilai karakteristik dan vektor karakteristik, sehingga subbab ini akan membahas definisi vektor karakteristik dan nilai karakteristik sekaligus cara menghitungnya.

PERPUS
ITS

2.3.1 Definisi Nilai Karakteristik dan Vektor Karakteristik

Definisi 1 : Misalkan A dalah sebuah matriks nxn. Sebuah matriks bukan nol, P, yang berukuran nx1 sedemikian rupa sehingga $AP = \lambda P$ dinamakan vektor karakteristik bagi A, sedangkan skalar λ dinamakan nilai karakteristik bagi A yang bersesuaian dengan P.

Persamaan pendefinisinya, yaitu :

$AP = \lambda P$ ekivalen dengan

$\lambda IP = AP$ atau $(\lambda I - A) P = 0.$

Persamaan terakhir akan mempunyai solusi bukan nol jika dan hanya jika

$\det(\lambda I - A) = 0.$

Hal ini memberi petunjuk tentang bagaimana cara memperoleh nilai karakteristik dan vektor karakteristik. Akan tetapi, apabila matriks A berukuran besar, maka cukup banyak memerlukan perhitungan sehingga cara demikian sangatlah rumit dan tidak efisien.

2.3.2 Penentuan Nilai Karakteristik dan Vektor Karakteristik

Untuk mendapatkan nilai karakteristik dan vektor karakteristik sekaligus, digunakan metode *Cyclic Jacobi* atau *Rotation Jacobi* yang telah dikembangkan oleh Jacobi [SAM-97]. Metode ini hanya bisa dikerjakan untuk matriks simetris.

Untuk mempermudah pencarian vektor karakteristik P yang bersesuaian dengan nilai karakteristik λ pada matriks berukuran besar, maka sebelum membahas metode Cyclic Jacobi, akan dibahas dulu definisi-definisi matematika

dasar dan aljabar linier. Sebab tanpa dasar ini, maka kita akan kesulitan untuk mendapatkan pengertian vektor karakteristik yang bersesuaian dengan nilai karakteristik dalam metode Cyclic Jacobi.

2.3.3 Transformasi Kemiripan

Transformasi Kemiripan (*Similarity Transform*) adalah untuk mentransformasikan matriks A ke matriks B yang *similar* dengan matriks A yang telah dinormalisasi, dimana B adalah matriks diagonal yang dikomposisikan dari nilai karakteristik $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n$. Tujuannya adalah untuk menunjukkan bahwa nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari A.

Definisi 2 : Kemiripan (*Similarity*)

A dan B adalah dua matriks nxn dikatakan *similar* bila $B = P^{-1}AP$.

Bukti bahwa nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari A dapat diilustrasikan sebagai berikut :

Bukti : $B = P^{-1}AP = \begin{bmatrix} \lambda_1 & 0 & 0 & \Lambda & 0 \\ 0 & \lambda_2 & 0 & \Lambda & 0 \\ 0 & 0 & O & & M \\ M & M & & O & M \\ 0 & 0 & \Lambda & \Lambda & \lambda_n \end{bmatrix}$

Misalkan $P_1, P_2, \ldots, P_n$ adalah kolom dari P, atau $P = [P_1\ P_2 \ldots P_n]$ adalah vektor karakteristik yang bersesuaian dengan nilai karakteristik, maka

$$A[P_1 P_2 P_3 \Lambda\ P_n] = [P_1 P_2 P_3 \Lambda\ P_n] \begin{bmatrix} \lambda_1 & 0 & 0 & \Lambda & 0 \\ 0 & \lambda_2 & 0 & \Lambda & 0 \\ 0 & 0 & O & & M \\ M & M & & O & M \\ 0 & 0 & \Lambda & \Lambda & \lambda_n \end{bmatrix}$$

Dengan menulis kembali dalam istilah kolom-kolom, maka

$[\ AP_1\ \ AP_2\ \ AP_3\ \ \ldots\ \ AP_n\ ] = [\ \lambda_1 P_1\ \ \lambda_2 P_2\ \ \lambda_3 P_3\ \ \ldots\ \ \lambda_n P_n\ ]$

Jadi diperoleh $A P_i = \lambda_i P_i$, I = 1, 2, 3, ... , n.

Yang berarti bahwa vektor P_i adalah vektor karakteristik yang bersesuaian dengan nilai karakteristik λ_i. Jadi terbukti bahwa, jika A dan B adalah 2 matriks *similar*, maka mereka mempunyai nilai karakteristik yang sama.

Dalam implementasi program, untuk mendapatkan nilai karakteristik dan vektor karakteristik dengan menggunakan matriks P ***invers***, P^{-1}, akan mendapatkan kesulitan, sebab nilai karakteristik di sini harus tidak boleh nol, sehingga harus diusahakan agar P adalah ***matriks ortogonal***.

Agar P *ortogonal*, maka A harus *simetris*. Pada sub bab sebelumnya sudah dibuktikan bahwa A adalah matriks simetris. Oleh karena A matriks nxn adalah matriks ***simetris***, maka matriks $\mathbf{A^T = A}$.

Dari uraian di atas, maka terbukti bahwa P adalah matriks yang dapat didiagonalkan secara ortogonal. Oleh karena P adalah *matriks ortogonal*, maka matriks $\mathbf{P^{-1} = P^T}$. Akibatnya, matriks diagonal B pada uraian sebelumnya yang semula ditulis :

$\mathbf{B = P^{-1}AP}$ *ekivalen dengan* $\mathbf{B = P^TAP}$

Persamaan yang ditulis terakhir ini adalah persamaan untuk mendapatkan kemudahan dalam menentukan nilai karakteristik dan vektor karakteristik dengan menggunakan *Metode Cyclic Jacobi*.

2.3.4 Metode Cyclic Jacobi untuk Matriks Simetris

Setelah kita mendapatkan gambaran tentang aljabar linier yang erat kaitannya dengan matriks, maka sekarang ini kita akan membahas metode Cyclic Jacobi.

Matriks rotasi bidang Rpq, yang mana elemen-elemen matriksnya adalah :

$r_{pp} = \cos\theta$ $\qquad$ $r_{pq} = \sin\theta$

$r_{qp} = \sin\theta$ $\qquad$ $r_{pq} = -\cos\theta$

$$R_{pq} = \begin{bmatrix} 1 & & & & & & \\ & \Lambda & & & & & \\ & & \cos\theta & \Lambda & \sin\theta & & \\ & & & \Lambda & & & \\ & & \sin\theta & \Lambda & -\cos\theta & & \\ & & & & & \Lambda & \\ & & & & & & 1 \end{bmatrix}$$

Semua elemen-elemen matriks yang tidak dispesifikasikan adalah 0 dan semua entries diagonal adalah 1, sin, atau cos. Sudut rotasi tidak masuk dalam pembicaraan kita. Matriks ini digunakan untuk mentransformasikan matriks A satu iterasi pada satu saat. Misalkan A matriks simetris dan kenyataannya bahwa matriks di atas berpengaruh hanya pada baris dan kolom p dan q.

Pada subbab sebelumnya telah dijelaskan bahwa terdapat ketentuan nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari matriks simetri A.

Dengan menggunakan rumus pada metode cyclic Jacobi yang bersesuaian dengan ketentuan similaritas di atas, maka diperoleh persamaan

$$D = R_{pq}^{T} A R_{pq}$$

dimana R adalah perkalian semua matriks rotasi R_{pq} dan R matriks ortogonal, sehingga dari persamaan di atas, dapat ditulis kembali sebagai :

$$AR = RD$$

dimana kolom R memberikan vektor karakteristik dari A yang bersesuaian dengan nilai karakteristiknya

2.4 METODE FISHERFACE

Metode Fisherface adalah metode untuk mentransformasikan vektor citra dari ruang citra dimensi-n ke ruang ciri dimensi-m[BEL-97]. Dengan demikian jika pada tugas akhir ini digunakan citra wajah dengan ukuran 92x112 piksel maka dimensi ruang citra sebesar 10304 piksel. Dimensi citra ini akan direduksi dengan Fisherface hingga memiliki dimensi sebesar m, dengan $m<n$.

Salah satu metode yang digunakan di sini adalah metode Principal Component Analys (PCA), yang juga diketahui sebagai transformasi Karhunen Loeve. Metode PCA ini bertujuan untuk mengurangi dimensi dari sebuah ruang sehingga menghasilkan basis baru yang lebih baik dalam mendeskripsikan berbagai kumpulan "model".

Sedangkan tujuan dari metode Fisherface adalah mereduksi dimensi sekaligus memperbesar rasio jarak antar kelas (between-class scatter) dengan jarak intra kelas (within-class scatter) dari vektor ciri dengan anggapan bahwa semakin besar rasio, vektor ciri yang dihasilkan semakin tidak sensitif baik terhadap perubahan ekspresi maupun perubahan cahaya[BEL-97].

Metode Fisherface dikembangkan untuk citra dalam berbagai variasi cahaya dan ekspresi wajah. Dasar metode Fisherface ini adalah Fisher's Linear Discriminant (FLD). Metode ini ditemukan oleh Robert Fisher pada tahun 1936 untuk klasifikasi taksonomi dan menjadi salah satu teknik yang banyak digunakan dalam pengenalan pola (pattern recognition)[BEL-97].

FLD merupakan salah satu contoh metode class spesific , karena metode ini berusaha untuk membentuk jarak (scatter) antar kelas dan intra kelas sehingga dapat menghasilkan klasifikasi yang lebih baik. FLD ini membuat matriks transformasi W dapat memaksimalkan rasio antara determinan between-class scatter (S_B) dengan within-class scatter (S_W) dari vektor-vektor ciri melalui fungsi:

$$W_{opt} = \arg\max_{w} \frac{\left|WS_B W^T\right|}{\left|WS_w W^T\right|}$$

$$= [w_1;w_2;...;w_m]$$

dimana $[w_1;w_2;...;w_m]$ merupakan m buah vektor eigen (dalam bentuk vektor baris) dari rasio antara S_B dengan S_W, yang bersesuaian dengan m buah nilai eigen terbesar. Jika w_i adalah vektor eigen dari rasio antara matriks S_B dengan matriks S_W dan d_i merupakan nilai eigen yang bersesuaian, maka :

$$S_B w_i^T = d_i S_W w_i^T$$

dimana i : 1...m dan $d_1 > d_2 > ... > d_m$.

Jika x_i, i = 1...N, adalah vektor citra dimensi-n dan masing-masing vektor citra merupakan anggota salah satu dari C kelas citra wajah $\{X_1, X_2, ..., X_C\}$ dan vektor μ adalah rata-rata vektor citra yang dapat diperoleh dari persamaan

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

maka matriks S_B dan matriks S_W dapat diperoleh melalui persamaan berikut :

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)^T (\mu_i - \mu)$$

$$S_W = \sum_{i=1}^{C} \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)$$

dimana N_i adalah jumlah anggota kelas X_i dan μ_i adalah rata-rata citra anggota kelas X_i, i = 1... C.

Suatu citra wajah dengan lebar dan tinggi masing-masing l dan t piksel memiliki jumlah piksel sebanyak l x t. Tiap-tiap piksel dikodekan dengan nilai 0-255 sesuai dengan nilai tingkat keabuannya. Maka dapat dibentuk vektor citra wajah berdasarkan nilai keabuan tersebut, yaitu :

$$g_i = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1l} \\ a_{21} & a_{22} & \dots & a_{2l} \\ \dots & \dots & \dots & \dots \\ a_{t1} & a_{t2} & \dots & a_{tl} \end{bmatrix}$$

dimana i = 1...N (banyaknya citra wajah)

Setiap g_i adalah anggota salah satu kelas wajah X. Jika terdapat C buah kelas wajah X maka terdapat X_j, dimana j = 1...C. Untuk setiap kelas wajah X_j terdapat N_j buah citra wajah, dimana j = 1...C. Dengan demikian jumlah semua citra wajah adalah $N_1+N_2+...+N_C = N$.

Dari vektor citra wajah di atas dapat dibentuk suatu vektor baris citra wajah, yaitu: $x_i = [\, a_{11} \;\; a_{12} \;\; \ldots \;\; a_{ii} \,]$ (1 * n)

Dengan demikian vektor citra dikatakan berada dalam ruang citra dimensi-n,dimana i = 1...N. Selanjutnya adalah membentuk matriks input berdimensi N*n yang berisi kumpulan vektor baris citra yang akan digunakan dalam pelatihan dan pengujian.

$$\text{input} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_N \end{bmatrix} \qquad (N*n)$$

2.4.1 Algoritma Metode Fisherface

Pada dasarnya metode Fisherface terdiri dari 4 langkah utama, yaitu Metode Principal Component Analys (PCA), Transformasi Principal Component Analys (PCA), Metode Fisher's Linear Discriminant (FLD), dan Transformasi Fisher's Linear Discriminant (FLD). Detail langkah dapat dijelaskan sebagai berikut :

Metode PCA

1. Menormalisasi matriks dari wajah-wajah pelatihan

$$X_i = \frac{Xi}{\sqrt{\sum_{i=1}^{n} Xi^2}}$$

Dimana n adalah dimensi dari ruang citra.

2. Mencari rata-rata vektor citra

$$\mu_{PCA} = \frac{1}{N}\sum_{i=1}^{N} X_i \qquad (1*n)$$

Dimana N adalah banyaknya data wajah yang akan mengalami pelatihan dan n adalah dimensi dari ruang citra.

3. Untuk i= 1 ...N, mencari selisih vektor citra dengan rata-rata vektor citra, μ_{PCA}

$$\Phi_i = X_i - \mu_{PCA} \qquad (1*n)$$

Menyimpan Φ dalam matriks A, sehingga

$$A = [\Phi_1; \Phi_2; \ldots; \Phi_N] \qquad (N*n)$$

4. Mencari matriks total scatter, S_T

$$S_T = A\,A^T \qquad (N*n)*(n*N)= (N*N)$$

5. Mencari vektor eigen dan nilai eigen dari matriks S_T, kemudian mengurutkan berdasarkan nilai eigen, dari yang terbesar.

$$[v,d]=eig(S_T)$$

Nilai eigen dari S_T : $d=[d_1 \quad d_2 \quad \ldots \quad d_n]$ (1*N)

Dimana $d_1 > d_2 > \ldots > d_n$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$v = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \qquad (N*N)$$

6. Mereduksi vektor eigen dari S_T sesuai dengan nilai (N-C)

$$v' = X_i^T V D$$

$$(n*N)*(N*N)*(N*(N-C))=(n*(N-C))$$

dimana :

- X_i adalah matriks wajah yang telah dinormalisasi dengan ordo (N*n)
- V adalah matriks yang berisi vektor karakteristik dengan ordo (N*N)
- D adalah matriks diagonal yang elemen diagonalnya adalah $(\lambda_i)^{-1/2}$ dengan ordo (N*(N-C))

Matriks transformasi W_{PCA} adalah :

$$W_{PCA} = v'^T \qquad ((N-C)*n)$$

Transformasi PCA

7. Transformasi vektor citra dimensi-n menjadi vektor input dimensi-(N-C), untuk input masukan metode FLD

$$InputFLD = A\ W_{PCA}^{T} \qquad (N*n)*(n*(N-C))=(N*(N-C))$$

Atau untuk i = 1…N :

$$X'i = \Phi i\ W_{PCA}^{T} \qquad (1*n)*(n*(N-C)) = (1*(N-C)$$

Jadi :

$$input = \begin{bmatrix} x'_1 \\ x'_2 \\ \cdots \\ x'_N \end{bmatrix}$$

Metode FLD

8. Mencari rata-rata input FLD

$$\mu_{FLD} = \frac{1}{N}\sum_{i=1}^{N} X'_i \qquad (1*(N-C))$$

9. Untuk I = 1..C, mencari rata-rata kelas X_I

$$\mu_i = \frac{1}{N_i}\sum_{i=1}^{Ni} X'_i \qquad (1*(N-C))$$

dimana N_i adalah jumlah anggota kelas Xi

menyimpan μi , i = 1..N, dalam matriks μ_k , sehingga :

$$\mu_k = \begin{bmatrix} x'_1 \\ x'_2 \\ \cdots \\ x'_C \end{bmatrix} \qquad (C*(N-C))$$

10. Mencari matriks *between-class scatter*,S_B

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu_{FLD})^T(\mu_i - \mu_{FLD})$$

$$((N-C)*C)*(C*(N-C)) = ((N-C)*(N-C))$$

11. Mencari matriks *within-class scatter*,S_W

$$S_W = \sum_{i=1}^{C} \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)$$

((N-C)*N)*(N*(N-C)) = ((N-C)*(N-C))

12. Mencari rasio antara S_B dengan S_W

$$\text{Rasio} = \frac{S_B}{S_W} \qquad ((N-C)*(N-C))$$

13. Mencari vektor eigen dan nilai eigen dari matriks rasio, kemudian mengurutkan berdasarkan nilai eigen, dari yang terbesar.

[v,d]=eig(rasio)

Nilai eigen dari rasio : $d=[d_1 \quad d_2 \quad \dots \quad d_{(N-C)}]$ (1*(N-C))

Dimana $d_1 > d_2 > \dots > d_{(N-C)}$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$v = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1(N-C)} \\ w_{21} & w_{22} & \cdots & w_{2(N-C)} \\ \cdots & \cdots & \cdots & \cdots \\ w_{(N-C)1} & w_{(N-C)2} & \cdots & w_{(N-C)(N-C)} \end{bmatrix} \qquad ((N-C)*(N-C))$$

14. Mereduksi vektor eigen dari rasio sesuai dengan nilai m = (C – 1)

$$v' = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ w_{(N-C)1} & w_{(N-C)2} & \cdots & w_{(N-C)m} \end{bmatrix} \qquad ((N-C)*m)$$

Matriks transformasi W_{FLD} adalah :

$$W_{FLD} = v'^T \qquad (m*(N-C))$$

Transformasi FLD

15. Transformasi vektor citra dimensi-(N-C) menjadi vektor ciri dimensi-m

$$W_{opt} = W_{PCA}{}^{T} W_{FLD}{}^{T}$$

2.4.2 Matriks Bobot Wajah

Matriks bobot wajah (weight face) adalah matriks yang berisi bobot dari tiap-tiap gambar yang digunakan untuk tahap pengenalan wajah. Matriks bobot wajah terdiri dari matriks bobot wajah pelatihan dan matriks bobot wajah uji coba.

2.4.2.1 Matriks Bobot Wajah Pelatihan

Misalkan matriks bobot wajah pelatihan, dinotasikan sebagai Ω, maka matriks bobot pelatihan untuk wajah pelatihan dari sejumlah n gambar adalah :

$$\Omega = X\ W_{opt}$$

$$(Nxm) = (Nxn)\ x\ (nxm)$$

dimana :

- Ω adalah matriks bobot wajah pelatihan dengan ordo (Nxm).
- X adalah matriks wajah pelatihan yang telah dinormalisasi dengan ordo (Nxn).
- W_{opt} adalah matriks hasil dari transformasi FLD.

2.4.2.2 Matriks Bobot Wajah Uji Coba

Misalkan sebuah wajah dalam tahap uji coba (testing) akan dihitung bobotnya, maka matriks bobot wajah uji coba , dinotasikan sebagai Ω_t, dengan persamaan sebagai berikut :

$$\Omega_t = X_t W_{opt}$$

$$(1xm) = (1xn) \times (nxm)$$

dimana :

- Ω_t adalah matriks bobot wajah ujicoba dengan ordo (1xm).
- X_t adalah matriks dari wajah uji coba yang telah dinormalisasi dengan ordo (1xn).
- W_{opt} adalah matriks hasil dari transformasi FLD.

BAB III
PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

BAB III
PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Dalam bab ini dibahas tentang perancangan input/output sistem, perancangan diagram alir data, perancangan stuktur data, dan perancangan proses.

3.1 TUJUAN DAN SASARAN SISTEM

Sistem pengenalan wajah ini dibuat dengan tujuan untuk mengimplementasikan metode Fisherface untuk pengenalan pola wajah kedalam bentuk perangkat lunak. Sasaran yang ingin dicapai adalah menguji kemampuan metode tersebut untuk mengenali sejumlah pola wajah yang bervariasi. Untuk memenuhi tujuan dan sasaran sistem diatas, maka ditempuh langkah-langkah seperti yang akan dijelaskan pada sejumlah sub bab dibawah.

3.2 PERANCANGAN PERANGKAT LUNAK

Perancangan perangkat lunak yang dilakukan meliputi perancangan data, perancangan proses, dan perancangan antarmuka. Perancangan data terdiri dari data masukan, data proses, dan data keluaran. Perancangan proses meliputi dari perancangan diagram alir data (DAD). Perancangan antarmuka terdiri form masukan, form proses, form keluaran.

3.2.1 Perancangan Data

Perancangan data bertujuan untuk mengetahui kebutuhan perangkat lunak yang akan dibangun. Pada bagian ini ditentukan data-data yang akan terlibat langsung dalam sistem yang akan dibangun. Perancangan data dapat digolongkan dalam tiga bagian, diantaranya adalah :

3.2.1.1 Data Masukan

Data Masukan dibagi menjadi dua bagian, yaitu : data masukan untuk proses pelatihan dan data masukan untuk proses uji coba. Data masukan untuk proses pelatihan berupa kelompok foto yang ukurannya sudah dijelaskan dalam bab I. Kelompok foto direpresentasikan dalam suatu tabel dengan format no record gambar, gambar. Sedangkan data masukan untuk proses uji coba berupa foto yang tidak terdapat dalam data pelatihan.

3.2.1.2 Data Proses

Data proses adalah data yang digunakan oleh sistem selama proses berlangsung. Data-data ini dapat dibagi lagi menjadi beberapa bagian, yaitu : data proses pelatihan dan data proses uji coba. Proses uji coba baru bisa dilakukan setelah proses pelatihan selesai, karena hasil dari proses pelatihan akan digunakan dalam proses uji coba.

- *Data Proses Pelatihan, terdiri dari* :
  - *Data Hasil Transformasi PCA*
    - *Data Hasil Normalisasi*

      Data ini diperoleh dari normalisasi matriks pelatihan. Kemudian dipakai untuk menghitung rata-rata vektor citra.

- *Data Hasil Perhitungan Matriks Kovarian*

  Dari hasil rata-rata vektor citra, kemudian dicari selisih vektor citra dengan rata-rata vektor citra. Selanjutnya dijadikan matriks kovarian (S_T).

- *Data Hasil Perhitungan Vektor Eigen dan Nilai Eigen dari Matriks Kovarian (S_T)*

  Vektor eigen dan nilai eigen diperoleh dengan menggunakan algoritma Jacobi. Kemudian mengurutkannya berdasarkan nilai eigen, dari yang terbesar. Selanjutnya vektor eigen direduksi sesuai dengan nilai pengurangan antara banyaknya foto dengan banyaknya kelas. Perkalian antara matriks wajah pelatihan yang dinormalisasi dengan transpose vektor eigen yang telah direduksi nantinya dipakai sebagai inputan untuk transformasi FLD(W_{PCA}).

- *Data Hasil Transformasi FLD*

  - *Data Hasil Rata-rata Input FLD*

    Data ini diperoleh dari perhitungan rata-rata input FLD, yang selanjutnya data ini dipakai untuk meghitung matriks between-class scatter (S_B).

  - *Data Hasil Rata-rata Kelas*

    Data ini diperoleh dengan menghitung rata-rata untuk masing-masing kelas,yang selanjutnya data ini dipakai untuk menghitung matriks between-class scatter (S_B) maupun matriks within-class scatter (S_W).

- *Data Hasil Perhitungan Matriks Between-class scatter (S_B)*

  Data ini diperoleh dari pengurangan antara data hasil dari rata-rata kelas dengan data hasil dari rata-rata input FLD.

- *Data Hasil Perhitungan Matriks Within-class scatter (S_W)*

  Data ini diperoleh dari pengurangan antara data hasil dari metode PCA dengan data hasil dari rata-rata kelas.

- *Data Hasil Perhitungan Vektor Eigen dan Nilai Eigen dari Rasio antara S_B dengan S_W*

  Mencari vektor eigen dan nilai eigen dari perbandingan antara S_B dengan S_W dengan menggunakan algoritma Jacobi. Kemudian mengurutkannya berdasarkan nilai eigen, dari yang terbesar. Selanjutnya vektor eigen direduksi sesuai dengan nilai banyaknya kelas dikurangi 1. Hasil tersebut selanjutnya disebut (W_{FLD}). Perkalian antara transpose data hasil dari transformasi PCA (W_{PCA}) dengan transpose data hasil dari transformasi FLD (W_{FLD}) atau yang disebut W_{opt} ini yang nantinya akan digunakan untuk menghitung matriks bobot wajah pelatihan maupun matriks bobot wajah uji coba.

- *Data Proses Uji Coba, terdiri dari :*

  - *Data Hasil Perhitungan Matriks Bobot Wajah Uji Coba*

    Data ini didapat dari proses perkalian antara matriks wajah uji coba yang telah dinormalisasi dengan data dari proses pelatihan (W^{T}_{opt}). Data ini selanjutnya dibandingkan dengan matriks bobot pelatihan.

- *Data Hasil Perhitungan Jarak Euclidean*

  Data ini diperoleh dengan menghitung perbandingan bobot wajah uji coba dengan bobot wajah pelatihan. Rumus dari jarak Euclidean adalah $|x_{1i} - x_{2i}| = (\Sigma(x_{1i} - x_{2i})^2)^{1/2}$.

3.2.1.3 Data Keluaran

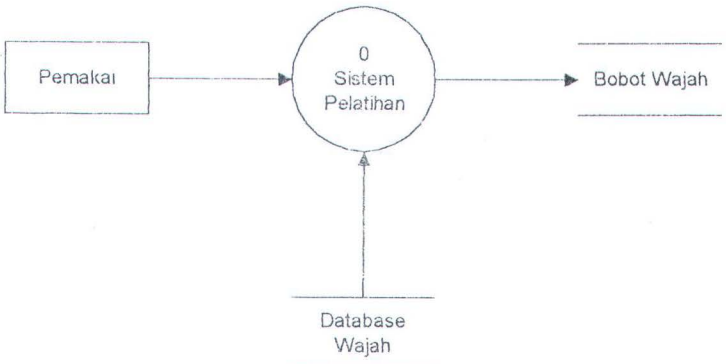
Data keluaran diperoleh dari pengurutan data hasil perhitungan jarak euclidean. Dari jarak terpendek akhirnya didapat foto dari data pelatihan yang mirip dengan foto dari data uji coba yang dimasukkan.

3.2.2 Perancangan Proses

Sistem pengenalan wajah ini terdiri dari dua subsistem yaitu subsistem proses pelatihan wajah-wajah yang ada didalam basis data dan subsistem proses uji coba atau pengenalan. Pembagian sistem pengenalan wajah ini menjadi dua subsistem didasarkan pada kondisi bahwa proses uji coba hanya bisa dilakukan setelah subsistem pelatihan dijalankan.

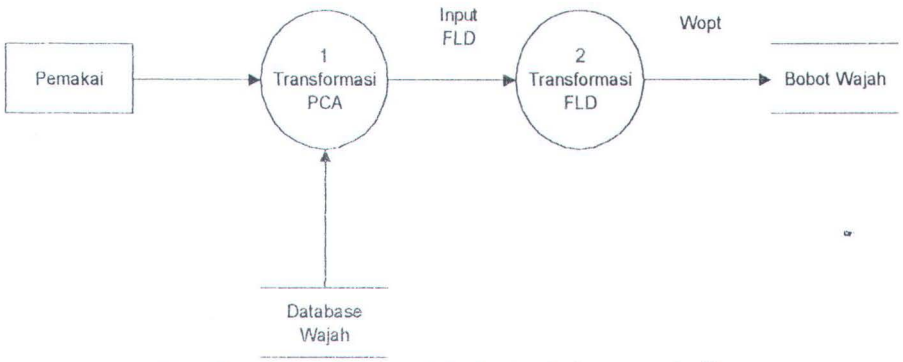
Bagian ini juga akan membahas perancangan sistem pengenalan wajah yang akan direpresentasikan dalam gambar, yaitu menggunakan Diagram Aliran Data (DAD) atau Data Flow Diagram (DFD). Dengan menggunakan diagram aliran data, dapat memperjelas semua proses dan aliran data yang terjadi dalam sistem pengenalan wajah ini, dari diagram aliran data juga dapat diketahui data yang diolah dan dihasilkan oleh suatu proses, dengan demikian aliran data mulai awal proses sampai akhir proses dapat diketahui dengan jelas.

Subsistem pelatihan merupakan bagian dari sistem pengenalan wajah dengan menggunakan metode Fisherface, subsistem pelatihan ini harus dilakukan dahulu sebelum subsistem uji coba atau pengenalan wajah dilakukan. Secara keseluruhan subsistem pelatihan dapat dilihat pada gambar 3.1 dibawah ini, yang menggambarkan kontek diagram atau DFD level 0 dari subsistem pelatihan



Gambar 3.1 DFD level 0 dari subsistem pelatihan

Data flow diagram diatas menggambarkan sistem pelatihan yang mengambil beberapa input citra dari database wajah kedalam proses pelatihan dan menghasilkan bobot wajah dari masing-masing wajah dari database wajah yang digunakan. Selanjutnya proses pelatihan ini dipecah-pecah menjadi beberapa subproses seperti gambar 3.2 yang menggambarkan DFD level 1.

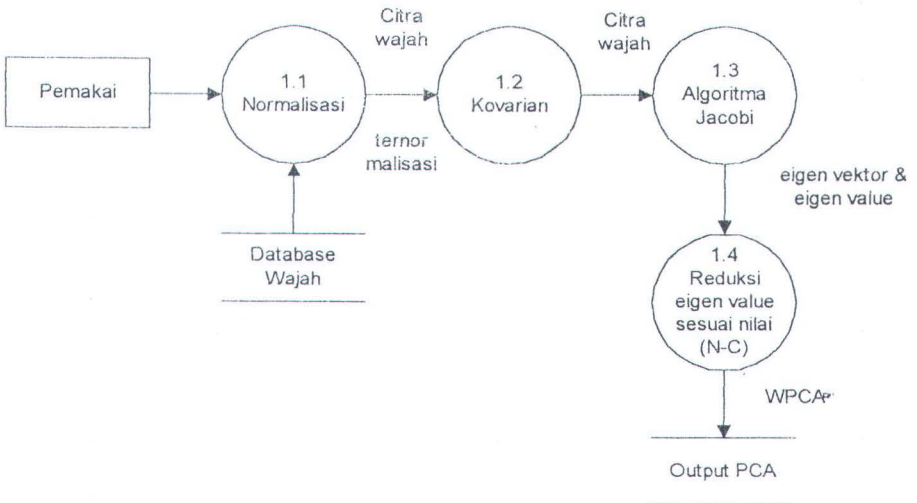


Gambar 3.2 DFD level 1 dari subsistem pelatihan

Proses pelatihan terdiri dari transformasi PCA dan transformasi FLD. Hasil atau output dari transformasi PCA tersebut menjadi inputan untuk melakukan transformasi FLD. Dalam transformasi PCA dipecah-pecah menjadi beberapa proses seperti gambar 3.3.

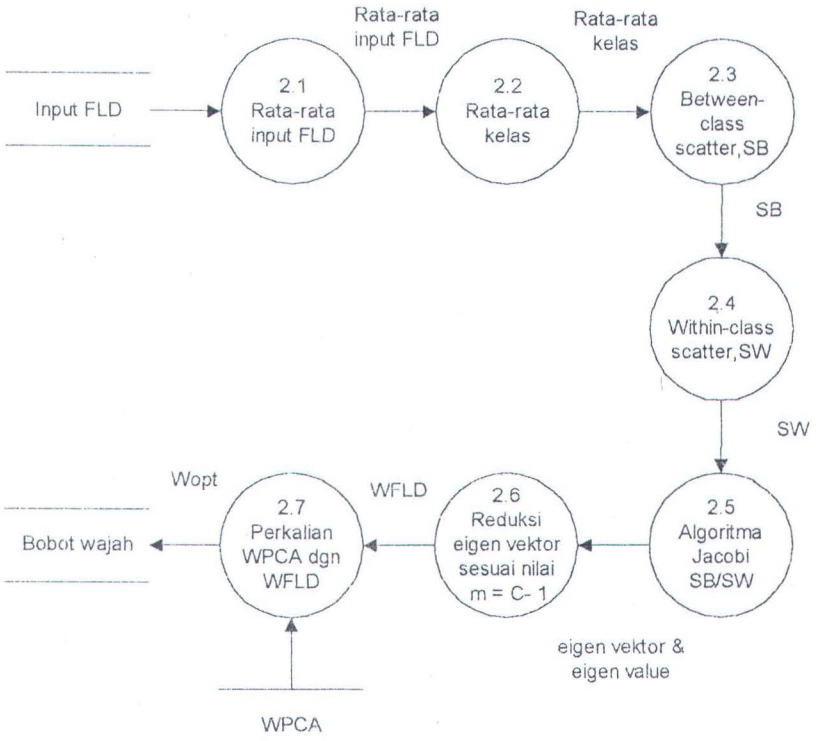
Proses dalam tranformasi PCA terdiri dari proses-proses sebagai berikut :

- Menormalisasi wajah-wajah pelatihan.
- Menghitung matriks kovariansi dengan menggunakan *outer product*.
- Menghitung vektor karakteristik dan nilai karakteristik untuk matriks kovariansi.
- Mengurutkan vektor karakteristik yang bersesuaian dengan nilai karakteristik dengan urutan *descending(naik)*.
- Mereduksi vektor karakteristik dengan nilai N-C(W_{PCA}),dimana N adalah banyak wajah atau data dalam basis data wajah, C merupakan jumlah kelas dalam basis data tersebut.



Gambar 3.3 DFD level 2 dari subsistem pelatihan Metode PCA

Setelah proses transformasi PCA selesai dijalankan hasil dari proses tersebut menjadi inputan untuk proses FLD. Output atau hasil dari transformasi FLD tersebut yang digunakan untuk menghitung bobot wajah-wajah pelatihan dan bobot wajah-wajah uji coba. Secara lebih lengkap proses transformasi FLD dapat dilihat pada gambar 3.4.

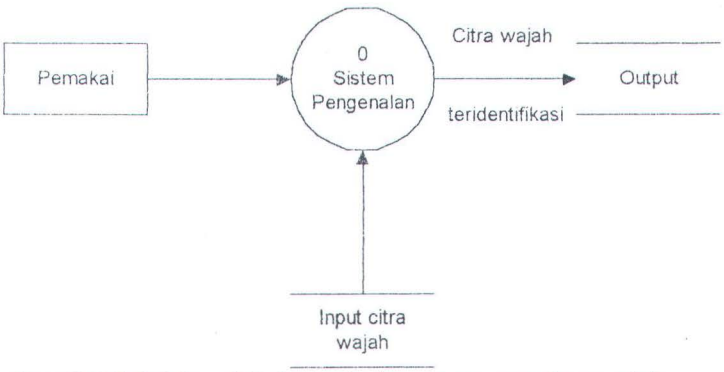


Gambar 3.4 DFD level 2 dari subsistem pelatihan Metode FLD

Proses dalam tranformasi FLD terdiri dari proses-proses sebagai berikut :

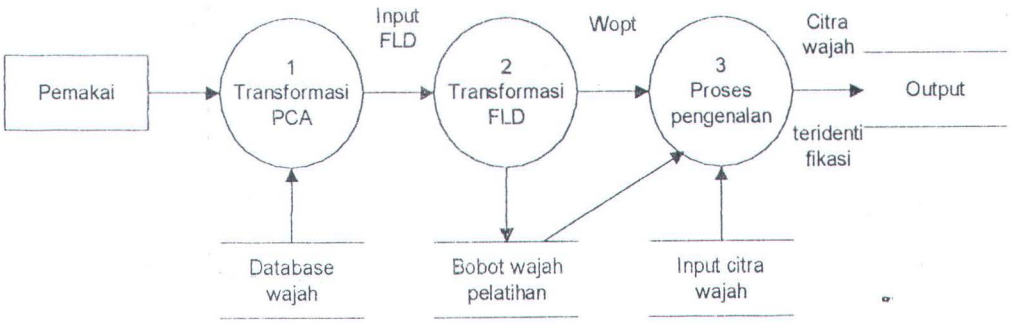
- Mencari rata-rata inputFLD.
- Mencari rata-rata kelas dari inputFLD.
- Mencari matriks *between-class scatter*,S_B.
- Mencari matriks *within-class scatter*,S_W.
- Mencari rasio antara S_B dengan S_W

- Menghitung vektor karakteristik dan nilai karakteristik untuk matriks rasio.
- Mengurutkan vektor karakteristik yang bersesuaian dengan nilai karakteristik dengan urutan *descending*.
- Mereduksi vektor karakteristik dengan nilai C-1.(W_{FLD}).



Gambar 3.5 DFD level 0 dari subsistem pengenalan wajah

Gambaran dari subsistem pengenalan wajah secara keseluruhan dapat digambarkan dengan Diagram Aliran Data level 0 (Data Flow Diagram). Dari gambar DFD level 0 dapat dilihat cara kerja sistem pengenalan wajah secara umum menerima suatu input citra yang akan dikenali, dan kemudian menghasilkan output suatu citra yang teridentifikasi.



Gambar 3.6 DFD level 1 dari subsistem pengenalan wajah

Seperti dijelaskan dalam dekripsi sistem diatas bahwa proses pengenalan atau uji coba baru bisa dijalankan setelah proses training selesai dijalankan. Secara lebih lengkap proses pengenalan atau uji coba dapat dilihat pada gambar 3.6.

Proses dalam pengenalan atau uji coba terdiri dari proses-proses sebagai berikut :

- Mengambil sebuah wajah uji coba.
- Menormalisasi wajah uji coba tersebut.
- Menghitung bobot wajah uji coba.
- Membandingkan bobot wajah uji coba dengan bobot wajah pelatihan dengan menggunakan rumus jarak Euclidean.
- Mengurutkan jarak yang dihasilkan dengan urutan ascending, kemudian mengambil jarak minimum terdekat.

3.2.3 Perancangan Antarmuka

Pada bagian ini akan dijelaskan struktur rancangan antarmuka dari sistem. Adapun hirarki menunya adalah sebagai berikut :

- Menu Open Database

  Menu ini digunakan untuk membuka tabel atau database yang berisi wajah-wajah pelatihan yang akan diproses.

- Menu Inisialisasi

  Menu ini digunakan untuk membaca foto kedalam bentuk matriks atau angka.

- Menu Training

  Menu ini digunakan untuk melakukan proses pelatihan dari wajah-wajah pelatihan yang diinputkan tersebut.

- Menu Open File

  Pada menu ini wajah-wajah uji coba dapat dimasukkan.

- Menu Testing

  Menu ini akan melakukan proses pengenalan dari wajah-wajah uji coba yang dimasukkan.

3.3 PEMBUATAN PERANGKAT LUNAK

Subbab ini dibagi menjadi tiga bagian, yaitu implementasi data, implementasi proses, dan implementasi antarmuka.

3.3.1 Implementasi Data

Data berupa foto yang kemudian dirubah dalam bentuk matriks atau angka. Disini bisa menambah kelas dan menghapus kelas. Setiap kelas harus mempunyai jumlah data yang sama.

Prosedur dibawah ini digunakan untuk menambah kelas.

```
procedure TFMain.btnAddClick(Sender: TObject);
var Pic: TPicture;
begin
  if dlgOpenImg.Execute then
  try
    Pic := TPicture.Create;
    Pic.LoadFromFile(dlgOpenImg.FileName);
    tblData.Append;
    dbImg.Picture := Pic;
    tblData.Post;
    AddData(Pic);
    LoadData;
```

```
    ShowData;
  finally
    Pic.Free;
  end;
end;

procedure TFMain.AddData(Pic: TPicture);
var len: integer;
begin
  len := Length(ImgData);
  SetLength(ImgData,len+1);
  ImgData[len] := TImage.Create(Self);
  ImgData[len].Picture := Pic;
  ShowData;
end;
```

Prosedur dibawah ini digunakan untuk menghapus kelas.

```
procedure TFMain.btnDeleteClick(Sender: TObject);
var idx: integer;
begin
  tblData.Delete;
  LoadData;
  ShowData;
end;

procedure TFMain.LoadData;
var JmlRec,i: integer;
begin
  FreeData;
  SetLength(ImgData,0);
  scbData.Refresh;
  With tblData do
  begin
    JmlRec := RecordCount;
    SetLength(ImgData,JmlRec);
    i := 0;
    tblData.First;
    While not tblData.Eof do
    begin
      ImgData[i] := TImage.Create(Self);
      ImgData[i].Picture := dbImg.Picture;
      Inc(i);
      Next;
    end;
  end;
end;
```

Prosedur untuk menampilkan foto-foto yang digunakan sebagai masukan dari sistem

```
procedure TFMain.ShowData();
var Len,i,a,b: integer;
begin
  With tblData do
  begin
    a := 0; b := 0;
```

```
    Len := Length(ImgData);
    for i:=0 to Len-1 do
    begin
      ImgData[i].Parent := scbData;
      ImgData[i].Width := 92;
      ImgData[i].Height := 112;
      ImgData[i].Left := 5 + (i*100);
      if a > Ni-1 then
      begin
        a := 0;
        Inc(b);
      end;
      ImgData[i].Top := 5 + (b*120);
      ImgData[i].Left := a * 100;
      Inc(a);
      Next;
    end;
  end;
  stBar.SimpleText := '   Jumlah Item :' +
IntToStr(Length(ImgData));
end;
```

Prosedur untuk membaca foto ke dalam bentuk matriks atau angka

```
procedure TFMain.Inisialisasi;
var i,j,k :integer;
begin
     setlength(imgdata,tbldata.recordcount);
     for k:=0 to tbldata.recordcount-1 do
     begin
         for i:=1 to 112 do
             for j:=1 to 92 do
                 D[k+1,(i-1)*92+j]:=
                 getrvalue(imgdata[k].picture.bitmap.
                 canvas.pixels[i-1,j-1]);
     end;
end;
```

3.3.2 Implementasi Proses

Dalam proses terdapat prosedur jacobi, prosedur normalisasi matriks, prosedur untuk mencari rata-rata kelas, prosedur mencari between-class scatter, prosedur mencari within-class scatter, prosedur membandingkan bobot wajah pelatihan dengan bobot wajah uji coba.

Prosedur Pencarian Matriks Eigen dengan Algoritma Jacobi.

Prosedur ini digunakan untuk mencari vektor eigen dan nilai eigen suatu matriks, baik untuk matriks kovarian maupun matriks rasio antara between-class scatter dengan within-class scatter

```
procedure jacobi(var a: glnpnp;n:integer;var d:glnp; var v : glnpnp; var
nrot : integer);
label 99;
const
     nmax = 200;
var
     j,iq,ip,i:integer;
     tresh,theta,tau,t,sm,s,h,g,c : real;
     b,z : array [1..nmax] of real;
begin
     for ip := 1 to n do
         begin
              for iq := 1 to n do
                  begin
                       v[ip,iq] := trunc(0.0)
                  end;
              v[ip,ip] := trunc(1.0)
         end;
     for ip := 1 to n do
         begin
              b[ip] := a[ip,ip];
              d[ip] := b[ip];
              z[ip] := trunc(0.0)
         end;
     nrot := 0;
     for i:= 1 to 50 do
         begin
              sm := trunc(0.0);
              for ip := 1 to n-1 do
              begin
                   for iq := ip+1 to n do
                   begin
                        sm := sm + abs(a[ip,iq])
                   end;
              end;
              if (sm = trunc(0.0)) then goto 99;
              if (i < 4) then tresh := 0.2*sm/sqr(n)
              else
              tresh := trunc(0.0);
              for ip := 1 to n-1 do
              begin
                   for iq := ip + 1 to n do
                   begin
                        g := trunc(100.0)* abs(a[ip,iq]);
                          if ((i>4) and ((abs(d[ip]+g))=
                            abs(d[ip])) and ((abs(d[iq]+g)) =
                            abs(d[iq]))) then a[ip,iq] := trunc(0.0)
                        else
                        if (abs(a[ip,iq])>tresh) then
```

```
                                begin
                                  h := d[iq]-d[ip];
                                  if ((abs(h)+g) = abs(h)) then
                                  begin
                                       t:=a[ip,iq]/h
                                  end
                                  else
                                  begin
                                        theta := 0.5*h/a[ip,iq];
                                            t:=trunc(1.0)/(abs(theta)+
                                            sqrt(trunc(1.0)+sqr(theta)));
                                       if (theta<trunc(0.0)) then t:=-t
                                  end;
                                  c:=trunc(1.0)/sqrt(1+sqr(t));
                                  s:=t*c;
                                  tau:=s/(trunc(1.0)+c);
                                  h:=t*a[ip,iq];
                                  z[ip]:=z[ip]-h;
                                  z[iq]:=z[iq]+h;
                                  d[ip]:=d[ip]-h;
                                  d[iq]:=d[iq]+h;
                                  a[ip,iq]:=trunc(0.0);
                                  for j:=1 to ip-1 do
                                  begin
                                       g:=a[j,ip];
                                       h:=a[j,iq];
                                       a[j,ip]:=g-s*(h+g*tau);
                                       a[j,iq]:=h+s*(g-h*tau)
                                  end;
                                  for j := ip+1 to iq - 1 do
                                  begin
                                       g:= a[ip,j];
                                       h:= a[j,iq];
                                       a[ip,j]:= g-s*(h+g*tau);
                                       a[j,iq]:= h+s*(g-h*tau)
                                  end;
                                  for j:= iq +1 to n do
                                  begin
                                       g:= a[ip,j];
                                       h:= a[iq,j];
                                       a[ip,j]:= g-s*(h+g*tau);
                                       a[iq,j]:= h+s*(g-h*tau)
                                  end;
                                  for j:= 1 to n do
                                  begin
                                       g:= v[j,ip];
                                       h:= v[j,iq];
                                       v[j,ip]:= g-s*(h+g*tau);
                                       v[j,iq]:= h+s*(g-h*tau)
                                  end;
                                  nrot := nrot+1
                             end
                             end
                        end
                   end;
                   for ip := 1 to n do
                   begin
                        b[ip]:=b[ip]+z[ip];
                        d[ip]:=b[ip];
                        z[ip]:=trunc(0.0)
                   end
              end
end
```

Prosedur normalisasi matriks

Normalisasi matrik menggunakan rumus : $\frac{Xi}{\sqrt{\sum_{i=1}^{n} Xi^2}}$

Normalisasi untuk matriks wajah pelatihan maupun untuk matriks wajah uji coba.

X_i adalah matriks wajah, n besar dari dimensi matriks wajah.

```
for i:= 1 to N do
     begin
         temp := 0;
         for j:= 1 to Dim do
             temp := temp + D[i,j]*D[i,j];
         temp:= sqrt(temp);
         for j:= 1 to Dim do
             D[i,j]:= D[i,j]/temp;
     end;
```

{Mencari Rata-rata Kelas menggunakan rumus $\mu_i = \frac{1}{N_i} \sum_{i=1}^{Ni} X'_i$ }

Dalam proses ini mencari rata-rata kelas masing-masing kelas, dimana N_i adalah jumlah wajah atau anggota setiap kelas.

```
for k:= 1 to N-C do
     begin
          for i:= 1 to C do
          begin
               temp3:=0;
               for j:=(i-1)*Ni+1 to (Ni*i) do
               begin
                    temp3:=temp3+H[j,k];
               end;
               R[i,k]:=temp3/Ni;
          end;
     end;
```

{Mencari $S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu_{FLD})^T (\mu_i - \mu_{FLD})$ }

Mencari matriks jarak antar kelas (between-class scatter), diperoleh dari rata-rata kelas dikurangi dengan rata-rata matriks hasil dari transformasi PCA

```
for i:=1 to N-C do
     begin
          for j:= 1 to C do
          begin
```

```
                F[j,i]:=R[j,i]-E[i];
            end;
        end;
```

{Mencari $S_W = \sum_{i=1}^{C} \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)$}

Mencari matriks jarak intra kelas (within-class sctter), diperoleh dari matriks hasil dari transfomasi PCA dikurangi dengan matriks rata-rata kelas.

```
for k:=1 to N-C do
    begin
        for i:=1 to C do
        begin
            for j:=(i-1)*Ni+1 to (Ni*i) do
            begin
                H[j,k]:=H[j,k]-R[i,k];
            end;
        end;
    end;
```

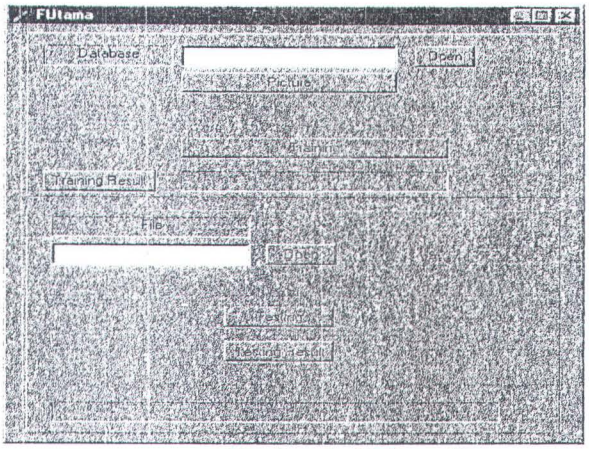
{Jarak Euclidean menggunakan rumus $|x_{1i} - x_{2i}| = (\Sigma(x_{1i} - x_{2i})^2)^{1/2}$}

Proses ini digunakan untuk menghitung perbandingan bobot wajah ujicoba dengan wajah pelatihan.

```
for i:=1 to N do
    begin
        templ:=0;
        for j:= 1 to C-1 do
        begin
            templ:=templ+(sqr(L[j]-H[i,j]));
        end;
        E[i]:=sqrt(templ);
        zl[i]:=E[i];
    end;
```

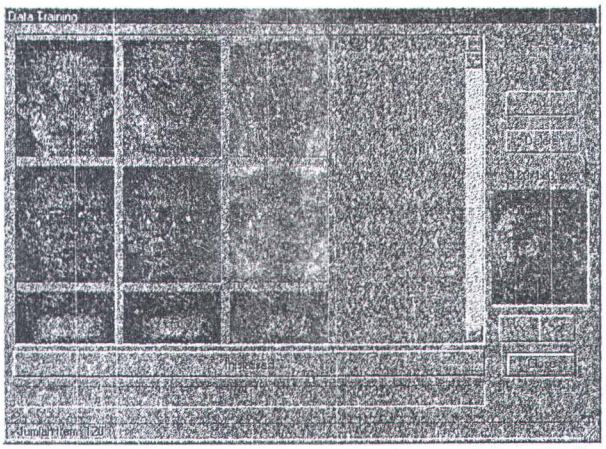
3.3.3 Implementasi Antarmuka

Pada bagian ini akan dijelaskan implementasi rancangan antarmuka seperti yang dijelaskan sebelumnya. Adapun prototype menu utama seperti ditunjukkan oleh gambar 3.7 berikut ini :



Gambar 3.7 Menu Utama

Setelah mengisi Open Database akan tampil form seperti gambar 3.8



Gambar 3.8 Tampilan sebagian wajah pelatihan

Add dan *Delete* digunakan untuk menambah dan mengurangi kelas pelatihan. Tombol Inisialisasi digunakan untuk merubah wajah pelatihan kedalam bentuk matriks. Setelah proses inisialisasi selesai, kembali ke menu utama. Tombol Training digunakan untuk melakukan proses pelatihan dari wajah-wajah pelatihan. Open file wajah yang akan diuji coba dilakukan setelah proses training selesai. Wajah yang mirip dengan wajah uji coba akan tampil setelah proses testing selesai.

BAB IV
UJI COBA DAN EVALUASI HASIL

BAB IV

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Bab ini menjelaskan tentang hasil uji coba dan evaluasi yang dilakukan terhadap perangkat lunak sistem pengenalan wajah dengan menggunakan metode Fosherface. Uji coba yang dilakukan terhadap sistem ini dimaksudkan untuk mengetahui kemampuan dan kelemahan sistem. Hasil pengujian akan dijadikan dasar bagi evaluasi terhadap sistem ini. Hasil evaluasi terhadap sistem ini bisa digunakan untuk pengembangan sistem lebih lanjut, sehingga bisa dijadikan sistem pengenalan yang lebih baik dibandingkan dengan sistem ini. Kriteria yang digunakan untuk evaluasi isstem ini adalah sebagai berikut :

- Waktu yang digunakan selama proses pelatihan berlangsung.
- Keberhasilan sistem dalam mengenali citra wajah dengan mengukur prosentase keberhasilan pengenalan terhadap jumlah citra wajah keseluruhan.

4.1 LINGKUNGAN UJI COBA

Perangkat keras yang dipakai untuk uji coba perangkat lunak ini adalah komputer dengan prosesor Intel Pentium II 450 Mhz dengan memori utama sebesar 128 MB.

4.2 DATA UJI COBA

Uji coba terhadap sistem ini dilakkan dengan menggunakan citra wajah yang berukuran 92 x 112 piksel. Citra wajah tersebut berada dalam bentuk grey level dengan tingkat keabuan 256. Data uji coba diambil dari basis data wajah yang dimiliki oleh laboratorium riset Olivetti. Basis data ini terdiri dari 400 obyek wajah dari 40 orang dengan 10 variasi untuk setiap orang. Data ini diindeks dengan angka 1 sampai 10. Sebagian dari data ini digunakan untuk proses pelatihan dan sisanya sebagai data uji coba.

4.3 UJI COBA DAN EVALUASI HASIL

Subbagian ini menjelaskan tentang uji coba dan evaluasi hasil yang dilakukan terhadap sistem pengenalan wajah ini. Pengujian yang dilakukan terhadap sistem pengenalan wajah ini menggunakan dua tipe data pelatihan, yaitu: data pelatihan terurut dan data pelatihan terkendali.

a. Data pelatihan terurut yang terdiri dari :

Data pelatihan terurut diambil dari data pelatihan sesuai dengan nomor urutan datanya tanpa memandang variasi data yang ada. Data pelatihan terurut terdiri

- Data pelatihan terurut 1

  Indeks nomor 1 sampai dengan 3 dari 10 variasi data yang ada digunakan sebagai data pelatihan,sedangkan sisanya untuk pengujian.

- Data pelatihan terurut 2.

  Indeks nomor 1 sampai dengan 4 dari 10 variasi data yang ada digunakan sebagai data pelatihan,sedangkan sisanya untuk pengujian

- Data pelatihan terurut 3.

  Indeks nomor 1 sampai dengan 5 dari 10 variasi data yang ada digunakan sebagai data pelatihan,sedangkan sisanya untuk pengujian

b. Data pelatihan terkendali.

Data pelatihan ini diambil dengan memilih dan mempertimbangkan variasi data yang ada tanpa memandang urutan indeks data. Data pelatihan terkendali ini diambil secara manual. Data pelatihan terkendali terdiri dari :

- Data pelatihan terkendali 1.

  3 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan tujuh data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

- Data pelatihan terkendali 2.

  4 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan enam data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

- Data pelatihan terkendali 3.

  5 data dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada tanpa memandang urutan indeks data. Sedangkan untuk pengujiannya digunakan lima data sisanya. Data yang terpilih untuk tipe ini sebagai data pelatihan dan data uji coba bisa dilihat pada lampiran.

4.3.4 Perbandingan Dengan Sistem

Berdasarkan hasil studi literature terhadap berbagai tugas akhir sarjana di Jurusan Teknik Informatika, terdapat dua tugas akhir mengenai pengenalan wajah yang dapat dijadikan sebagai bahan perbandingan dengan sistem pengenalan wajah yang dibuat dalam tugas akhir ini. Kedua tugas akhir yang relevan tersebut adalah :

"PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK PENGENALAN WAJAH DENGAN MENGGUNAKAN JARINGAN SYARAF YANG DIDASARKAN PADA KEPUTUSAN PROBABILISTIK" [SUB-99]

Kemampuan pengenalan yang dimiliki oleh jaringan syaraf PDBNN ini tergantung dari jumlah variasi data pelatihan yang digunakan untuk pelatihan. Pengambilan data pelatihan tanpa memandang variasi data yang mewakili, seperti yang dilakukan pada uji coba, menghasilkan tingkat keberhasilan 56,5% untuk subnet yang mewakili bernilai tertinggi dan 69,5% untuk subnet yang mewakili bernilai positif. Sedangkan pengambilan data dengan pemilikan yang mewakili variasi dari data yang ada, menghasilkan tingkat keberhasilan 72% untuk subnet yang mewakili bernilai tertinggi dan 85% untuk subnet yang mewakili bernilai positif. Waktu pelatihan untuk jaringan pendeteksi wajah dan pencari lokasi mata 4 menit dan waktu pelatihan jaringan syaraf 2 menit. Waktu rata-rata yang di[illegible]an untuk menemukan area wajah yang memiliki nilai log-likehood tertinggi 25 detik dan untuk menemukan posisi kedua mata 6 detik.

"PENGENALAN WAJAH DENGAN MENGGUNAKAN JARINGAN SYARAF KONVOLUSIONAL"[FAT-98]

Dari sebanyak 280 data yang diujicobakan, 168 data dapat dikenali sedangkan 112 data lainnya gagal. Prosentase keberhasilan dari uji coba ini sebesar 71,5% untuk data pelatihan terurut dan 82% untuk data pelatihan terkendali. Jumlah data pelatihan yang semakin besar tidak menjamin meningkatnya kemampuan jaringan syaraf konvolusional dalam melakukan klasifikasi. Hal ini mungkin dipengaruhi oleh variasi data pelatihan. Data pelatihan yang tidak terlalu variatif akan menyempitkan jangkauan dari kelas tersebut. Akan tetapi data pelatihan dengan variasi yang tinggi akan mengurangi ketepatan dalam generalisasi kelas tersebut. Waktu yang dibutuhkan untuk data pelatihan terurut untuk pelatihan jaringan SOM 10 menit dan untuk pelatihan jaringan konvolusional 19 jam. Sedangkan untuk data pelatihan terkendali untuk pelatiha jaringan SOM 52 menit dan untuk pelatihan jaringan konvolusional 26 jam.

BAB V
KESIMPULAN

BAB V

KESIMPULAN

Dalam bab ini dibahas mengenai beberapa kesimpulan yang dapat diambil dari keseluruhan tugas akhir yang telah dibuat. Kesimpulan diambil berdasarkan hasil ujicoba yang telah dilakukan terhadap sistem pengenalan wajah dengan data yang tersedia.

Dari ujicoba yang telah dilakukan sistem pengenalan wajah ini dapat diambil beberapa kesimpulan :

1. Hasil yang diperoleh dalam proses pengenalan wajah dengan menggunakan Metode *Fisherface* untuk data pelatihan terurut 80% sedangkan untuk data pelatihan terkendali 88%. Hasil ini menunjukkan bahwa pengenalan wajah dengan menggunakan Metode *Fisherface* cukup baik untuk diterapkan dalam sistem pengenalan wajah.
2. Hasil perbandingan dengan 2 sistem pengenalan wajah yang lain. Dengan menggunakan jaringan syaraf yang didasarkan pada keputusan probabilistik untuk data pelatihan terurut 69,5% sedangkan untuk data pelatihan terkendali 85%. Dengan menggunakan jaringan syaraf konvolusional untuk data pelatihan terurut 71,5% sedangkan untuk data pelatihan terkendali 82%.
3. Untuk setiap tambahan input wajah baru dilakukan pelatihan ulang sehingga konsekuensinya dibutuhkan memori yang cukup besar.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[SUB-99] Agus Subhan , "Perancangan dan Pembuatan Perangkat Lunak Pengenalan Wajah Dengan Menggunakan Jaringan Syaraf yang didasarkan pada Keputusan Probabilistik", *Tugas Akhir*, Sarjana Jurusan Teknik Informatika FTI-ITS, 1999

[ANT-98] Howard Anton, *Elementary Linear Algebra*, Drexel University Press, 1998

[BEL-97] Belhumeur P.N, J.P.Hespanha, and D.J.Kriegman, "Face Recognition Using Class Specific Linear Projection" , *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol 19, No 7, July 1997

[FAT-98] Aminudin Al Fathhoni, "Pengenalan Wajah Dengan Menggunakan Jaringan Syaraf Konvolusional", *Tugas Akhir*, Sarjana Jurusan Teknik Informatika FTI-ITS, 1998

[SET-00] Setyati Endang, "Pengenalan Wajah dengan Menggunakan PCA", *Tesis Pasca Sarjana*, Jurusan Teknik Informatika, 2000

[SAM-97] Sami Romdhani, *Face Recognition by Using PCA*, 1997

[WIL-89] William H, Teukolsky, Saul A, Vetterling, William T, and Flannery Brian P, *Numerical Recipes in C*, Cambridge University Press, 1989

LAMPIRAN

LAMPIRAN A

DATA PELATIHAN

Gambar yang digunakan untuk pelatihan maupun untuk uji coba terdapat pada tabel A.1. Gambar tersebut terdiri dari 40 orang yang berbeda, dengan setiap orang memiliki 10 variasi.

Pola	1	2	3	4	5	6	7	8	9	10
1										
2										
3										

Pola	1	2	3	4	5	6	7	8	9	10
4										
5										
6										
7										
8										
9										

Pola	1	2	3	4	5	6	7	8	9	10
10										
11										
12										
13										
14										
15										

4.3.1 Uji Coba dengan Data Pelatihan Terurut

Hasil yang dicapai dari uji coba dengan data pelatihan terurut dapat dilihat dalam tabel 4.1, yang kesemua data tersebut terurut sesuai dengan nomor indeksnya.

Tabel 4.1
Hasil uji coba pengenalan wajah dengan data pelatihan terurut

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
1	3	43%	4	57%	5	83%	1	17%	4	80%	1	20%
2	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
3	2	29%	5	71%	2	33%	4	67%	2	40%	3	60%
4	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
5	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
6	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
7	5	71%	2	29%	5	83%	1	17%	5	100%	0	0%
8	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
9	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
10	3	43%	4	57%	3	50%	3	50%	3	60%	2	40%
11	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
12	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
13	4	57%	3	43%	4	67%	2	33%	4	80%	1	20%
14	3	43%	4	57%	2	33%	4	67%	3	60%	2	40%
15	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
16	3	43%	4	57%	4	67%	2	33%	3	60%	2	40%
17	2	29%	5	71%	1	17%	5	83%	1	20%	4	80%
18	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
19	2	29%	5	71%	5	83%	1	17%	4	80%	1	20%

Tabel 4.1

Hasil uji coba pengenalan wajah dengan data pelatihan terurut

(lanjutan)

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
20	3	43%	4	57%	2	33%	4	67%	2	40%	4	80%
21	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
22	7	100%	0	0%	6	100%	0	0%	5	100%	1	20%
23	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
24	6	86%	1	14%	5	83%	1	17%	5	100%	2	40%
25	4	57%	3	43%	4	67%	2	33%	3	60%	2	40%
26	3	43%	4	57%	4	67%	2	33%	3	60%	0	0%
27	3	43%	4	57%	2	33%	4	67%	1	20%	4	80%
28	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
29	4	57%	3	43%	3	50%	3	50%	4	80%	1	20%
30	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
31	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
32	4	57%	3	43%	3	50%	3	50%	3	60%	2	40%
33	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
34	5	71%	2	29%	4	67%	2	33%	4	80%	1	20%
35	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
36	4	57%	3	43%	3	50%	3	50%	3	60%	2	40%
37	3	43%	4	57%	2	33%	4	67%	2	40%	3	60%
38	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
39	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
40	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
Rata-rata		67%		33%		72%		28%		80%		20%
Waktu komputasi	Pelatihan : 1,5 menit				2,5 menit				3,5 menit			
	Pengenalan : 10 detik				20 detik				30 detik			

4.3.2 Uji Coba dengan Data Pelatihan Tekendali

Hasil yang dicapai dari uji coba dengan data pelatihan terkendali dapat dilihat dalam tabel 4.2, yang kesemua data tersebut dipilih dari 10 variasi data dengan pertimbangan variasi data yang ada anpa memandang urutan indeks data..

Tabel 4.2
Hasil uji coba pengenalan wajah dengan data pelatihan terkendali

Pola	Tipe data terkendali 1				Tipe data terkendali 2				Tipe data terkendali 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
1	3	43%	4	57%	5	83%	1	17%	4	80%	1	20%
2	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
3	3	43%	4	57%	4	67%%	2	33%	3	60%	2	40%
4	6	86%	1	14%	5	83%	1	17%	5	100%	0	0%
5	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
6	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
7	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
8	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
9	4	57%	3	43%	5	83%	1	17%	5	100%	0	0%
10	5	71%	2	23%	4	67%	2	33%	3	60%	2	40%
11	4	57%	3	43%	3	50%	3	50%	4	80%	1	20%
12	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
13	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
14	3	43%	4	57%	5	83%	1	17%	5	100%	0	0%
15	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
16	5	71%	2	29%	4	67%	2	33%	4	80%	1	20%
17	2	29%	5	71%	5	83%	1	17%	5	100%	0	0%
18	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%

Tabel 4.2
Hasil uji coba pengenalan wajah dengan data pelatihan terkendali
(lanjutan)

Pola	Tipe data terurut 1				Tipe data terurut 2				Tipe data terurut 3			
	Keberhasilan		Kegagalan		Keberhasilan		Kegagalan		Keberhasilan		Kegagalan	
	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase	Jumlah	Prosentase
19	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
20	4	57%	3	43%	5	83%	1	17%	4	80%	1	20%
21	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
22	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
23	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
24	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
25	4	57%	3	43%	4	67%	2	33%	4	80%	1	20%
26	3	43%	4	57%	3	50%	3	50%	4	80%	1	20%
27	3	43%	4	57%	3	50%	3	50%	3	60%	2	40%
28	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
29	4	57%	3	43%	3	50%	3	50%	5	100%	0	0%
30	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
31	5	71%	2	29%	4	67%	2	33%	3	60%	2	40%
32	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
33	7	100%	0	0%	6	100%	0	0%	5	100%	0	0%
34	5	71%	2	29%	4	67%	2	33%	5	100%	0	0%
35	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
36	5	71%	2	29%	5	83%	1	17%	4	80%	1	20%
37	3	43%	4	57%	3	50%	3	50%	2	40%	3	60%
38	6	86%	1	14%	5	83%	1	17%	4	80%	1	20%
39	5	71%	2	29%	5	83%	1	17%	4	80%	1	20%
40	6	86%	1	14%	6	100%	0	0%	5	100%	0	0%
Rata-rata		76%		24%		83%		17%		88%		12%
Waktu komputasi	Pelatihan : 1,5 menit Pengenalan : 10 detik				2,5 menit 20 detik				3,5 menit 30 detik			

4.3.3 Evaluasi Hasil Uji Coba

Prosentase keberhasilan pengenalan wajah seperti yang terlihat pada tabel 4.1 dan 4.2 sangat tergantung pada pemilihan data pelatihan yang digunakan. Pemilihan data pelatihan yang mewakili semua variasi atau mendektai semua variasi dalam data uji coba memberikan hasil pengenalan yang meningkat jika dibandingkan dengan mengambil secara acak atau urut data pelatihan tanpa memandang variasi yang ada. Tabel berikut memberikan perbandingan tingkat keberhasilan dari pemilihan data dari uji coba yang dilakukan.

Tabel 4.3
Perbandingan keberhasilan pengenalan berdasarkan metode pemilihan data pelatihan yang digunakan.

Tipe Data Uji Coba	Uji Coba Terurut			Uji Coba Terkendali		
	Presentase Keberhasilan	Waktu Komputasi		Presentase Keberhasilan	Waktu Komputasi	
		Pelatihan	Pengenalan		Pelatihan	Pengenalan
Jumlah Data 3	67%	1,5 menit	10 detik	76%	1,5 menit	10 detik
Jumlah Data 4	72%	2,5 menit	20 detik	83%	2,5 menit	20 detik
Jumlah Data 5	80%	3,5 menit	30 detik	88%	3,5 menit	30 detik

Waktu yang diperlukan untuk melakukan proses pelatihan berbanding lurus dengan banyaknya data yang ditraining. Semakin banyak jumlah data yang ditraining, maka semakin banyak pula waktu yang dibutuhkan. Dari pengamatan hasil uji coba baik dengan data pelatihan terurut maupun data pelatihan terkendali, ada beberapa data yang pengenalannya kurang baik. Perbedaan-perbedaan yang signifikan ini merupakan penyebab faktor kegagalan dalam mengenali citra wajah yang diinputkan ke dalam program tersebut, yaitu :

- Adanya kemiripan wajah antara satu orang dengan orang yang lain.
- Adanya ekspresi wajah yang bervariasi, sedangkan data untuk pelatihan belum mewakili variasi data yanga ada.

ITS

Pola	1	2	3	4	5	6	7	8	9	10
16										
17										
18										
19										
20										
21										

Pola	1	2	3	4	5	6	7	8	9	10
22										
23										
24										
25										
26										
27										

Pola	1	2	3	4	5	6	7	8	9	10
28										
29										
30										
31										
32										
33										

Pola	1	2	3	4	5	6	7	8	9	10
34										
35										
36										
37										
38										
39										

40	

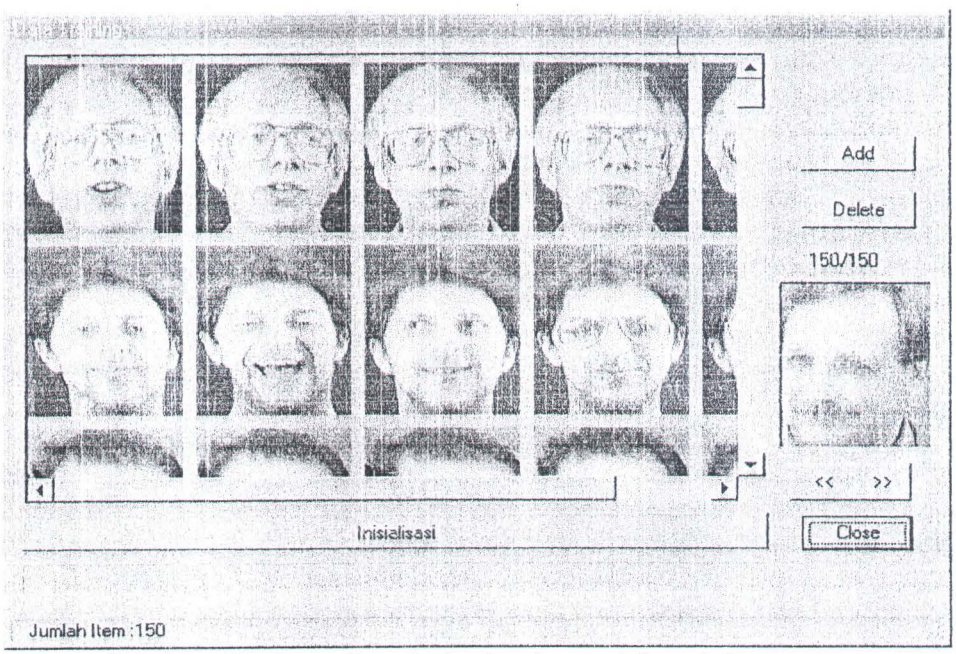
LAMPIRAN B

PETUNJUK PEMAKAIAN PERANGKAT LUNAK

Langkah-langkah dalam implementasi program :

Langkah-langkah dalam tahap training:

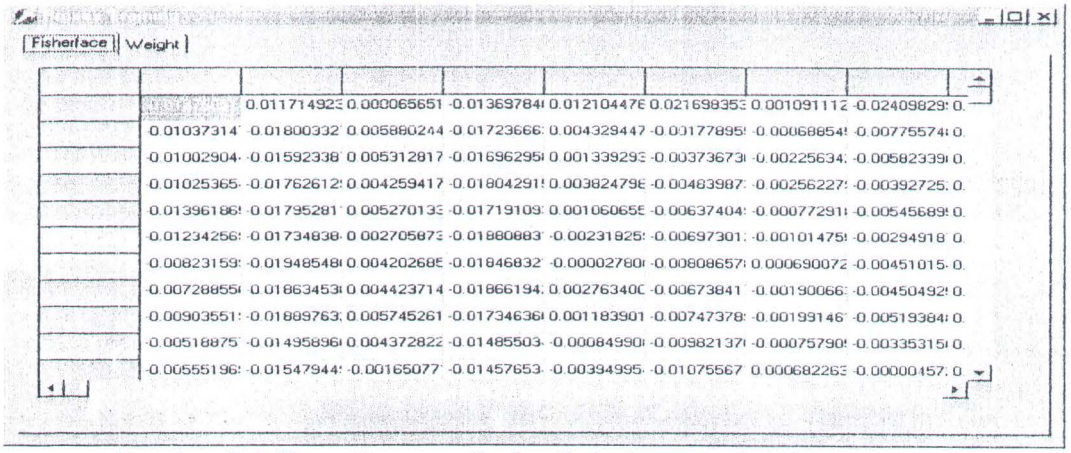
Langkah 1: Aktifkan file prjFace,kemudian click tombol *Open*, dalam tombol open ini dipilih tabel mana yang akan ditraining. Apabila ingin menambah record tabel tekan tombol *Add* dan apabila ingin menghapus record tabel tekan tombol *Delete*.



Gambar B.1 Tampilan dari sebagian wajah training

Langkah 2: Click tombol *Inisialisai*.Tombol ini digunakan untuk membaca gambar yang akan ditraining.

Langkah 3: Click tombol *Training*.Setelah proses training selesai, macam-macam matriks hasil training dapat dilihat dengan menekan tombol *Training Result*.

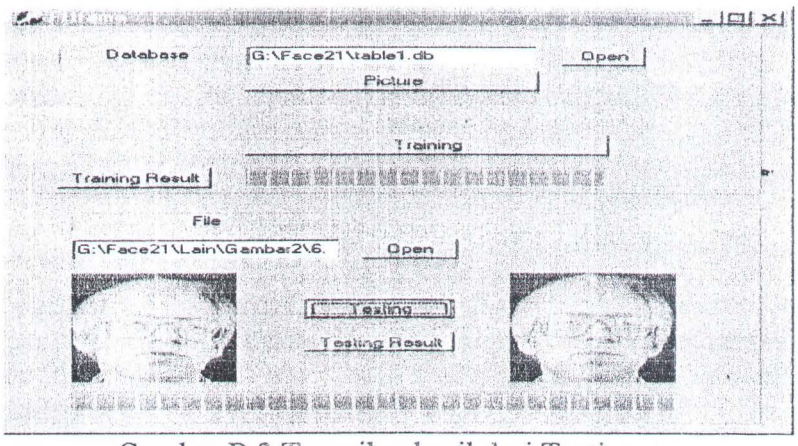


Gambar B.2 Tampilan matriks hasil dari proses training

Langkah-langkah dalam tahap testing:

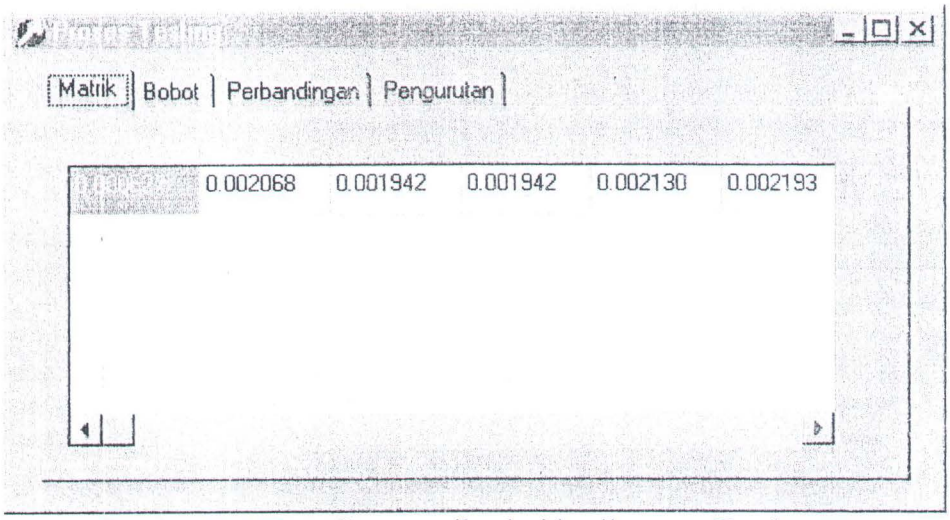
Langkah 1: Click tombol *Open*, tombol ini digunakan untuk menginputkan gambar yang akan diuji coba.

Langkah 2 : Click tombol *Testing*, tombol ini digunakan untuk melakukan proses testing. Setelah proses testing selesai, maka akan tampak gambar yang hampir mendekati dengan gambar ujicoba.



Gambar B.3 Tampilan hasil dari Testing

Langkah 3 : Macam-macam matriks hasil testing dapat dilihat dengan menekan tombol *Testing Result*.



Gambar B.4 Tampilan matriks dari hasil proses Testing

17.564/H/03

MILIK PERPUSTAKAAN ITS

PERANCANGAN DAN PEMBUATAN SISTEM OLAP UNTUK ANALISIS STUDI KELAYAKAN PROYEK

Studi Kasus Pemerintah Kabupaten Kediri

TUGAS AKHIR

R514
005.1
Dam
p-1
2002

Oleh :

RULY DAMARATRI
5198.100.079

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2002

PERPUSTAKAAN ITS
Tgl. Terima 13-8-2002
Terima dari 1
No. Agenda Prp. 215805

PERANCANGAN DAN PEMBUATAN SISTEM OLAP UNTUK ANALISIS STUDI KELAYAKAN PROYEK

Studi Kasus Pemerintah Kabupaten Kediri

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui/Menyetujui

Pembimbing I	Pembimbing II
Dosen	Kabid. Data Bappeda Kab. Kediri
Dr. Ir. Joko Lianto Buliali, M.Sc	Drs. Sugeng Sundoro, M.M
NIP. 131996151	NIP. 510056704

SURABAYA
Juli, 2002

ABSTRAK

ABSTRAK

Untuk memperbaiki kinerja pemerintah, maka penataan kelembagaan negara, instansi pemerintah, entitas ekonomi, dan satuan kerja sektor publik, merupakan keharusan untuk dikelola dalam sebuah konsep ***good governance****. Mengingat pengeluaran pembangunan memerlukan dana yang relatif besar, maka perlu diperhatikan beberapa faktor yang mendasarinya yaitu antara lain: kemampuan anggaran negara, skala prioritas pembangunan serta analisis biaya dan manfaatnya. Karena itu perlu dilakukan evaluasi atas usulan proyek-proyek pembangunan untuk diseleksi dan ditentukan kelayakannya, agar tujuannya dapat dicapai dan dapat memberikan manfaat yang optimal. Rangkaian kegiatan proyek-proyek pembangunan meliputi aktivitas perencanaan, pelaksanaan, dan pengawasan proyek. Aktivitas perencanaan proyek adalah keseluruhan proses pengidentifikasian rencana kegiatan dan keuangan proyek serta tujuan yang ingin dicapai.*

Sebuah aplikasi berbasis teknologi ***OLAP*** *mampu menjadi* ***analysis tool*** *yang dapat memanipulasi tampilan dalam berbagai sudut pandang data, merubah urutan data, dan melakukan perubahan parameter dengan menggunakan fungsi-fungsi rekapitulasi dalam sebuah aliran data* ***global to detil*** *atau sebaliknya, akan dapat memberi bantuan kepada pengguna dalam proses Analisis menentukan kebijakan dalam menentukan kelayakan proyek-proyek pembangunan terutama ditinjau dari aspek keuangan.*

Tugas Akhir ini akan merupakan sebuah penerapan aplikasi OLAP yang nantinya terdiri atas 3 modul yaitu User Management yang akan mengatur hak akses dari para pengguna, OLAP Entri yang akan menjembatani sistem manual dengan sistem OLAP yang akan dibuat, dan modul OLAP Report yang merupakan penerapan dari OLAP yang sesungguhnya sebagai alat bantu Analisis.

KATA PENGANTAR

KATA PENGANTAR

Segala puji syukur ke hadirat Tuhan Yang Maha Esa, Maha Kasih dan Penyayang yang telah memberikan berkat dan rahmat, serta kasih setiaNya, sehingga penulis telah dapat menyelesaikan penyusunan tugas akhir ini, dengan judul ***"Perancangan dan Pembuatan Sistem OLAP untuk Analisa Studi Kelayakan Proyek Pemerintah Daerah"***, sebagai salah satu syarat untuk menyelesaikan program strata satu (S-1) pada Jurusan Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan tugas akhir ini, penulis berusaha untuk menerapkan ilmu didapat selama masa-masa perkuliahan dengan tidak terlepas dari niat baik dari berbagai pihak berupa petunjuk, bimbingan, bantuan, dan dukungan yang ditujukan kepada penulis.

Dengan tidak lupa akan kodratnya sebagai manusia, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan dan terdapat banyak kekurangan. Oleh karena itu, dengan segala kerendahan hati penulis masih dan akan tetap terus mengharapkan saran serta kritik yang membangun dari rekan-rekan pembaca. Meskipun demikian, penulis berharap bahwa tugas akhir ini dapat memberikan manfaat kepada pembacanya.

Surabaya, Juli 2002

Penulis

UCAPAN TERIMA KASIH

UCAPAN TERIMA KASIH

Penyelesaian Tugas Akhir ini tak terlepas dari bantuan banyak pihak, untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapa yang sudah begitu mengasihi penulis, mencukupi setiap kebutuhan, memberikan mujijat serta hikmat yang melimpah, menyertai penulis sehingga Tugas Akhir ini dapat diselesaikan sampai tuntas tanpa kekurangan apapun. Terima kasih untuk ayat saat teduhnya (Roma 10:10) yang menghilangkan segala *nervous* dan membuat penulis semakin yakin dan percaya diri. *I Love You, Father!*
2. Bapak dan Ibu yang telah banyak memberi dukungan baik berupa materi maupun doa serta pengorbanannya.
3. Anang Krisnocahyo, sebagai seorang kakak yang telah memberikan banyak dukungan dan nasehat.
4. Ari Sih Rahmat, sebagai seorang adik yang selalu siap membantu ditengah kesibukan penulis.
5. Yudy "KOKO" Setyawan, atas dukungan, ispirasi dan doanya, terima kasih atas perhatian, kesabaran dan kebersamaanya dalam membantu penulis menyelesaikan tugas akhir ini. Minta maaf kalo gara-gara Tugas akhir ini Nonik selalu bikin Koko kuatir. *God Bless Us!*
6. Agus Zainal SKom, MKom, selaku Ketua Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember

7. Bapak Dr.Ir. Joko Lianto Buliali,M.Sc serta Bapak Sugeng Sundoro, MM, selaku pembimbing yang telah sangat banyak membantu memberikan saran dan masukan dalam penyelesaian Tugas Akhir ini. Terima kasih Pak atas bantuan dan dukungannya.
8. Bapak Dwi Sunaryono, SKom selaku dosen wali yang telah memberikan nasehat, semangat dan dorongan kepada penulis sehingga penulis dapat menyelesaikan studinya dengan cepat.
9. Seluruh Dosen-dosen Teknik Informatika yang tidak dapat penulis sebutkan satu-persatu. Terima kasih atas bimbingan dan ilmunya selama masa perkuliahan.
10. Seluruh staf dan karyawan Teknik Informatika: Mas Yudi, Mas Sugeng, Mas Sholeh, Mbak Irna, Mbak Davi dan yang lainnya, yang telah membantu dalam kemudahan administrasi.
11. Mas Ilham yang telah meluangkan waktunya untuk membantu penulis dalam pengambilan data dan pencarian informasi di Bappeda Kediri.
12. Seluruh staf Bappeda yang telah banyak direpotkan selama pengerjaan tugas akhir ini.
13. Bang Romeo dan Mbak Betty yang sudah sangat membantu dalam memberikan "referensi" dalam pengerjaan tugas akhir ini. Minta maaf kalo penulis sudah terlalu merepotkan. Semoga sukses selalu.
14. Semua "Philean – PhiLeo Chorale" atas kebersamaanya, kegilaan, semangat dan doanya serta pelayanan-pelayanan yang telah ada yang membuat penulis menjadi bersemangat melewati semua masalah-masalah.

15. Rekan-rekan serta seluruh staf Indonesian Airlines yang ikut memberikan dukungan serta semangat. Terima kasih untuk Pak Edi Surono dan Pak Messy yang telah memberikan ijin, Farah dan Feranita yang selalu memberikan perhatian, serta rekan-rekan lain yang sudah menemani hari-hari penulis menjelang ujian Tugas Akhir.
16. Intan Hapsari Eltri, sahabatku yang baik yang telah banyak memberikan semangat selama penulis menjadi mahasiswa, yang rela menyediakan waktunya untuk manampung segala keluh kesah. Semoga persahabatan ini sampai selamanya.
17. Dina Kurniasari, Akuntansi FE Unair yang telah mengajari analisa laporan keuangan. Maaf kalo sering ngrepotin.
18. Mbak Icha yang selalu menanyakan "*Kapan maju seminar?*". Terima kasih untuk perhatiannya.
19. Mas Tinus yang selalu menawarkan bantuannya. Minta maaf untuk setiap penolakan atas tawaran yang diberikan. Semoga Mas bisa bantu lain waktu.
20. Bapak dan Ibu Iswanto sekeluarga, Mbak Ni, serta teman-teman Kos Bhaskara IV 23-25 yang telah banyak membantu dan menemani penulis sampai akhir studinya.
21. Mas Setya dan Mbak Iin yang selalu memberi dukungan dan doa, semoga kalian rukun selalu.
22. Ratna, Citra, dan Brata serta *SeTech-ers* yang lain atas kebersamaan dan kesabarannya dalam menyelesaikan permasalahan-permasalahan "proyek".

23. Cewek-cewek "Wonder Woman" Premi, Dewi, Raras, Citra, Nana, Ifah, dan Nita atas persahabatan dalam suka duka serta kebersamaannya selama ini. Semoga sampai kapanpun kita bisa tetap jadi *wonder woman.*
24. Rekan-rekan C0E semuanya tanpa terkecuali yang telah banyak membantu penulis selama masa perkuliahan. Terima kasih semuanya, semoga semuanya juga bisa cepat lulus.
25. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah secara langsung maupun tidak langsung telah memberikan dukungan dan bantuan kepada penulis dalam menyelesaikan studi.

DAFTAR ISI

DAFTAR ISI

ABSTRAK iii
KATA PENGANTAR iv
UCAPAN TERIMA KASIH v
DAFTAR ISI ix
DAFTAR GAMBAR xii
DAFTAR TABEL xiv

BAB I PENDAHULUAN 1

1.1 LATAR BELAKANG 1
1.2 TUJUAN 4
1.3 PERMASALAHAN 5
1.4 BATASAN PERMASALAHAN 5
1.5 METODOLOGI 6
1.6 SISTEMATIKA PEMBAHASAN 8

BAB II TEORI PENUNJANG 10

2.1 FEASIBILITY STUDY 10
2.1.1 Analisis Keuangan dalam *Feasibility Study* 11
2.1.2 Indikator-Indikator Keuangan 11
2.1.2.1 Return On Total Assets (ROA) *12*
2.1.2.2 Return On Investment (ROI) *13*
2.1.2.3 Internal Rate of Return (IRR) *13*
2.1.2.4 Payback Periode (PBP) *13*
2.1.2.5 Service Coverage (SC) *14*
2.1.3 Penentuan Prioritas (*Ranking*) Proyek 14
2.1.3.1 Benefit, Disbenefit, Cost *14*
2.1.3.2 Benefit Cost Analysis *16*
2.1.3.3 Analisis Benefit Cost Ratio untuk Membandingkan Alternatif *17*
2.2 ONLINE ANALYTICAL PROCESSING 18
2.2.1 OLAP 18
2.2.2 Multidimensional Data Cube 19
2.2.3 Kategorisasi OLAP 22
2.2.3.1 ROLAP *22*
2.2.3.2 MOLAP *24*
2.2.3.3 MQE *26*
2.2.3.4 HOLAP *27*
2.3 METODOLOGI SDLC 27

BAB III PENGENALAN PROSES PERENCANAAN BAPPEDA 31

3.1 SISTEM YANG ADA SEKARANG 31
3.1.1 Kebutuhan Data untuk Analisis 33
3.1.1.1 Data Struktur Proyek *33*
3.1.1 Elemen Perencana 35
3.1.2 Sarana Pendukung 36
3.2 SISTEM YANG DIHARAPKAN 36

BAB IV ANALISIS KEBUTUHAN SISTEM 39

4.1 ANALISIS KEBUTUHAN 39
4.1.1 Analisis User Management 41
4.1.2 Analisis Kebutuhan Entri Data terhadap Sistem yang Akan Dibuat 42
4.1.2.1 Kebutuhan Data Struktur Proyek *42*
4.1.2.2 Kebutuhan Data Keuangan Proyek *43*
4.1.3 Analisis Fitur OLAP Report 43
4.2 ANALISIS SARANA PENDUKUNG 44

BAB V PERANCANGAN SISTEM 46

5.1 DESAIN PROSES 46
5.3.1 DAD Level 0 46
5.3.2 DAD Level 1 47
5.3.3 DAD Level 2 49
5.3.3.1 DAD Level 2 Login *49*
5.3.3.2 DAD Level 2 User Management *50*
5.3.3.3 DAD Level 2 OLAP Entri *51*
5.3.3.4 DAD Level 2 OLAP Report *52*
5.3.4 DAD Level 3 54
5.3.4.1 DAD Level 3 Penentuan Cube dan Layout *54*
5.3.4.2 DAD Level 3 Data Display Customizing *55*
5.2 DESAIN DATA 57
5.2.1 Desain Data *User Management* 57
5.2.2 Desain Data OLAP Entri 58
5.2.3 Desain Data OLAP Report 59
5.2.3.1 Desain Data Struktur Database *59*
5.2.3.2 Desain Data Penyimpanan Display Analisis *61*
5.2.3.3 Desain Query Data Cube *61*
5.3 DESAIN ANTARMUKA 63
5.3.1 Desain Antarmuka *User Management* 63
5.3.2 Desain Antarmuka OLAP Entri 63
5.3.3 Desain Antarmuka OLAP Report 64
5.4 DESAIN ALGORITMA PERHITUNGAN INDIKATOR KEUANGAN 65

BAB VI PEMBUATAN APLIKASI SISTEM *FEASIBILITY STUDY* BERBASIS OLAP 66

6.1 IMPLEMENTASI DATABASE 66
6.1.1 Implementasi Database *User Management* dan OLAP Report 66
6.1.2 Implementasi Database OLAP Entri 67

6.2 PEMBUATAN MODUL 68
6.2.1 Pembuatan Modul User Management 68
6.2.2 Pembuatan Modul OLAP Entri 71
6.2.2.1 Pembuatan Form Struktur Proyek *74*
6.2.2.2 Pembuatan Form Entri Kelas Proyek *80*
6.2.2.3 Pembuatan Form Entri Balance Sheet *80*
6.2.2.4 Pembuatan Form Entri Income Statement *81*
6.2.2.5 Pembuatan Form Entri Benefit Cost *82*
6.2.2.6 Pembuatan Form Entri Cash Flow *82*
6.2.2.7 Pembuatan Form Report BCR *83*
6.2.2.8 Pembuatan Form Report IRR *84*
6.2.2.9 Pembuatan Form Report PBP *86*
6.2.2.10 Pembuatan Form Report ROA *88*
6.2.2.11 Pembuatan Form Report ROI *89*
6.2.3 Pembuatan Modul OLAP Report 91
6.2.3.1 Pembuatan Analysis Wizard *93*
6.2.3.2 Pembuatan Field Properties *98*
6.2.3.3 Pembuatan Range Properties *99*
6.3 JOIN-QUERY VIEW 99

BAB VII UJI COBA DAN ANALISIS HASIL 100

7. 1 LINGKUNGAN UJI COBA 100
7.2 SKENARIO UJI COBA 100
7.2.1 Uji Coba Login 101
7.2.2 Uji Coba *User Management* 102
7.2.3 Uji Coba OLAP Entri 103
7.2.4 Uji Coba OLAP Report 105
7.3 ANALISIS HASIL UJI COBA 113

BAB VIII KESIMPULAN DAN SARAN 114

8.1 KESIMPULAN 114
8.2 SARAN 114

DAFTAR PUSTAKA

LAMPIRAN

LAMPIRAN A DESAIN ALGORITMA PERHITUNGAN INDIKATOR KEUANGAN
LAMPIRAN B JOIN-QUERY PEMBUATAN CUBE
LAMPIRAN C DAFTAR ISIAN SEKTOR/SUBSEKTOR/PROGRAM PROYEK PEMBANGUNAN

DAFTAR GAMBAR

DAFTAR GAMBAR

GAMBAR 2.1 ILUSTRASI FEASIBILITY STUDY PROYEK PEMERINTAH 11
GAMBAR 2.2 MULTIDIMENSIONAL DATA VIEW 20
GAMBAR 2.3 ARSITEKTUR ROLAP 23
GAMBAR 2.4 ARSITEKTUR MOLAP 25
GAMBAR 2.5 ARSITEKTUR MQE 26
GAMBAR 2.6 SYSTEM DEVELOPMENT LIFE CYCLE 30
GAMBAR 3.1 STRUKTUR PROYEK 35
GAMBAR 5.1 DAD LEVEL 0 47
GAMBAR 5.2 DAD LEVEL 1 47
GAMBAR 5.3 DAD LEVEL 2 LOGIN 49
GAMBAR 5.4 DAD LEVEL 2 USER MANAGEMENT 50
GAMBAR 5.5 DAD LEVEL 2 OLAP ENTRI 52
GAMBAR 5.6 DAD LEVEL 2 OLAP REPORT 53
GAMBAR 5.7 DAD LEVEL 3 PENENTUAN CUBE DAN LAYOUT 55
GAMBAR 5.8 DAD LEVEL 3 DATA DISPLAY CUSTOMIZING 56
GAMBAR 5.9 ERD USER MANAGEMENT 58
GAMBAR 5.10 ERD OLAP ENTRI 60
GAMBAR 5.11 ERD STRUKTUR DATABASE OLAP REPORT 61
GAMBAR 5.12 ERD PENYIMPANAN DISPLAY ANALISIS OLAP REPORT 61
GAMBAR 5.13 DESAIN ANTARMUKA USER MANAGEMENT 63
GAMBAR 5.14 DESAIN ANTARMUKA OLAP ENTRI 64
GAMBAR 5.15 DESAIN ANTARMUKA OLAP REPORT 65
GAMBAR 6.1 FORM DATA USER 69
GAMBAR 6.2 FORM DEPARTEMEN 70
GAMBAR 6.3 FORM GROUP USER 70
GAMBAR 6.4 FORM PASSWORD USER 71
GAMBAR 6.5 MDI OLAP ENTRI 71
GAMBAR 6.6 TOOLBAR MDI OLAP ENTRI 73
GAMBAR 6.7 TOOLBAR ACTIVEFORM PADA OLAP ENTRI 74
GAMBAR 6.8 PENGGAMBARAN TREE-VIEW PADA STRUKTUR PROYEK 75
GAMBAR 6.9 FORM ENTRI SEKTOR 77
GAMBAR 6.10 FORM ENTRI SUBSEKTOR 77
GAMBAR 6.11 FORM ENTRI PROGRAM 78
GAMBAR 6.12 FORM ENTRI PROYEK 79
GAMBAR 6.13 FORM ENTRI KELAS PROYEK 80
GAMBAR 6.14 FORM ENTRI BALANCE SHEET 80
GAMBAR 6.15 FORM ENTRI INCOME STATEMENT 81

GAMBAR 6.16 FORM ENTRI BENEFIT COST 82
GAMBAR 6.17 FORM ENTRI CASH FLOW 82
GAMBAR 6.18 FORM REPORT BCR 83
GAMBAR 6.19 FORM REPORT IRR 86
GAMBAR 6.20 FORM REPORT PBP 86
GAMBAR 6.21 FORM REPORT ROA 88
GAMBAR 6.22 FORM REPORT ROI 89
GAMBAR 6.23 MDI OLAP REPORT 91
GAMBAR 6.24 TOOLBAR OLAP REPORT 92
GAMBAR 7.1 PERINGATAN USER GROUP TIDAK MEMILIK HAK AKSES TERHADAP SEBUAH MODUL 101
GAMBAR 7.2 PERINGATAN INPUTAN KOMBINASI USER NAME DAN PASSWORD YANG KELIRU 102
GAMBAR 7.3 PERINGATAN PENUTUPAN APLIKASI SEGERA OLEH SISTEM. 102
GAMBAR 7.4 MENU OLAP ENTRI YANG DIPERKENANKAN UNTUK USER GROUP DATA 103
GAMBAR 7.5 MENU OLAP ENTRI YANG DIPERKENANKAN UNTUK USER GROUP KEUANGAN 103
GAMBAR 7.6 STRUKTUR PROYEK HASIL UJI COBA PEMASUKAN SEKTOR,SUBSEKTOR, PROGRAM DAN PROYEK BARU 105
GAMBAR 7.7 HASIL UJI COBA PEMBUATAN DOKUMEN ANALISIS BARU 106
GAMBAR 7.8 HASIL UJI COBA MANIPULASI DENGAN CUSTOM WIZARD.... 108
GAMBAR 7.9 HASIL UJI COBA MANIPULASI DENGAN RANGE 109
GAMBAR 7.10 HASIL UJI COBA MANIPULASI DENGAN FIELDS PROPERTIES 110
GAMBAR 7.11 PEMILIHAN SEL DATA GRAFIK 110
GAMBAR 7.12 HASIL UJI COBA PENGGAMBARAN GRAFIK 111
GAMBAR 7.13 HASIL UJI COBA EXPORT DATA KE EXCELL WORKBOOK 111
GAMBAR 7.14 HASIL UJI COBA PENYIMPANAN DOKUMEN FAVORITE ANALYSIS 112
GAMBAR A.1 FLOWCHART ALGORITMA PERHITUNGAN ROA 117
GAMBAR A.2 FLOWCHART ALGORITMA PERHITUNGAN ROI 118
GAMBAR A.3 FLOWCHART ALGORITMA PERHITUNGAN IRR 120
GAMBAR A.4 FLOWCHART ALGORITMA PERHITUNGAN PBP 121
GAMBAR A.5 FLOWCHART ALGORITMA PERHITUNGAN SC 122
GAMBAR A.6 FLOWCHART ALGORITMA PERHITUNGAN BCR 124

DAFTAR TABEL

DAFTAR TABEL

TABEL 2.1 CONTOH BENEFIT, DISBENEFIT, COST 15
TABEL 6.1 MENU-MENU MDI OLAP ENTRI 72
TABEL 6.2 FUNGSI ICON-ICON PADA TOOLBAR MDI OLAP ENTRI 73
TABEL 6.3 FUNGSI ICON-ICON PADA TOOLBAR ACTIVEFORM OLAP ENTRI ... 74
TABEL 6.4 MENU-MENU PADA MDI OLAP REPORT 91
TABEL 6.5 FUNGSI ICON-ICON PADA TOOLBAR OLAP REPORT 92
TABEL C.1 DAFTAR ISIAN SEKTOR/SUBSEKTOR/PROGRAM PROYEK PEMBANGUNAN 128

BAB I

PENDAHULUAN

BAB I
PENDAHULUAN

Pada bab ini akan dijelaskan beberapa hal dasar yang meliputi latar belakang, tujuan, permasalahan, batasan permasalahan, metodologi, dan sistematika pembahasan. Diharapkan bab ini akan memberikan gambaran umum terhadap permasalahan dan pemecahannya yang dapat digunakan untuk menyelesaikan tugas akhir ini.

1.1 Latar Belakang

Memasuki era reformasi sejak pertengahan tahun 1998, sangat disadari pentingnya membangun kembali manajemen pemerintahan melalui paradigma baru menuju *good governance*. Menurut Bank Dunia (1992), *good governance* adalah sistem pemerintahan yang handal, pelayanan publik yang efisien, serta pemerintahan yang akuntabel terhadap publik. Pemakaian istilah *good governance* direkomendasikan oleh Bank Dunia sebagai pilihan dari *good government* atau *clean government*. *Good governance* berlaku terhadap keseluruhan lembaga negara dalam penyelenggaraan negara (ROSJIDI 2001: 141-142).

Dalam setiap penyelenggaraan, *good governance* harus berlandaskan pada tiga prinsip dasar, yaitu: transparansi, partisipasi, dan akuntabilitas. Ketiga prinsip tersebut merupakan bagian tak terpisahkan dalam setiap penentuan kebijakan publik, implementasi, dan pertanggungjawabannya dalam bingkai *good governance*.

Good governance berlaku untuk keseluruhan lembaga negara dalam penyelenggaraan negara. Salah satu hal penting dalam dimensi fungsi lembaga-

lembaga negara adalah jenis-jenis layanan fungsionalnya kepada *stakeholders* masing-masing lembaga. Layanan fungsional ini di beberapa negara lain sudah merupakan layanan baku *(standard services)* yang bersifat wajib diberikan oleh lembaga pemerintah, baik kepada masyarakat umum secara langsung maupun lewat lembaga-lembaga pemerintah lain yang mempunyai kegiatan yang searah. Layanan-layanan dasar ini antara lain adalah layanan di bidang kesehatan masyarakat, di bidang prasarana umum, di bidang pendidikan. Bahkan bidang-bidang baru seperti teknologi informasi juga sudah diberikan bobot yang cukup besar penanganannya.

Perlunya layanan fungsional sebagai *standard services* yang diberikan oleh pemerintah dan lembaga-lembaga pemerintah juga harus dijawab oleh daerah dalam pengembangan potensi-potensi yang dimiliki oleh daerah, sejalan dengan rencana pembangunan dan pengembangan potensi-potensi yang dimiliki oleh daerah. Untuk mewujudkan hal ini, pemerintah daerah memerlukan partisipasi aktif lembaga-lembaga yang ada dalam lingkup pemerintahan daerah, seperti Bappeda, Pemda, DPRD, Dinas-Dinas Daerah, dan lembaga-lembaga daerah terkait lainnya. Lembaga-lembaga daerah ini menjalankan tugas dan fungsinya berkaitan dengan pembangunan daerah secara terstruktur dan koordinatif untuk membantu kelancaran tata alir prosedur pembangunan infrastruktur daerah, sarana dan prasarana, serta pengembangan potensi-potensi yang dimiliki daerah.

Pemerintah daerah dalam hal ini perlu mengambil inisiatif untuk mengidentifikasi dan merencanakan program-program pembangunan dan pengembangan potensi-potensi yang mereka miliki dan segera menjabarkan kebutuhan layanan masyarakat untuk meningkatkan pendapatan asli daerah, seperti pelayanan administrasi (identitas penduduk dan perijinan), pelayanan infrastruktur

(jalan raya, jaringan irigasi, jembatan), pelayanan kebutuhan dasar (pangan, air minum, kesehatan, pendidikan), dan pelayanan penerimaan daerah. Selain itu, pemerintah daerah juga menjabarkan dengan sistematis indikator-indikator penilaian program pengembangan potensi daerah sebagai sebuah proyek pengembangan yang ditangani oleh swasta (private, proses investasi), pemerintah daerah sendiri lewat RAPBD, atau nasional (pemerintah pusat). Sehingga dalam proses pembangunan tersebut juga diperlukan tertib administrasi dan keuangan.

Sebagaimana yang telah dijabarkan di atas, wacana pengelolaan sistem pemerintahan yang transparan dalam lingkup sistem pemerintahan nasional dan regional yang telah berkembang saat ini, membuat pemerintah daerah perlu menjabarkan jenis-jenis layanan fungsional pemerintah kepada masyarakat umum dan kalangan dunia usaha secara luas sehubungan dengan pengembangan potensi-potensi yang dimiliki daerah untuk meningkatkan taraf hidup masyarakat di daerah. Jenis-jenis layanan fungsional pemerintah memiliki beberapa dimensi kajian fungsional, yaitu Finansial, Administrasi, Proses, dll.

Penjabaran informasi layanan secara transparan secara langsung maupun tidak langsung membuka peluang dalam proses pelaksanaan pembangunan. Informasi kinerja proyek-proyek pembangunan beserta perencanaan tujuan dari masing-masing proyek membuka peluang yang lebih besar untuk terjadinya investasi bagi sektor bersangkutan. Informasi kinerja finansial yang cukup lengkap akan banyak membantu dalam proses investasi.

Untuk menjawab tantangan diatas, maka perlu dibuat sebuah sistem yang mampu memberi kemudahan kepada Pemerintah Daerah khususnya Bappeda dalam melakukan presentasi (promosi) terhadap potensi-potensi daerahnya dan juga dalam

perencanaan pembangunan yang akan dilakukan. OLAP (*Online Analytical Processing*) sebagai *analysis tool* merupakan salah satu metode yang digunakan untuk menampilkan data dalam bentuk suatu rekapitulasi yang dapat digunakan dalam sebuah Analisis.

Oleh sebab itu, OLAP akan memudahkan pengguna untuk melihat dan menganalisis sekumpulan data dari berbagai sudut pandang, sehingga akan membantu pengguna dalam mengambil keputusan secara tepat, misalnya: dalam sebuah proses menentukan kelayakan apakah sebuah proyek dapat dilaksanakan, maka pengguna dapat menganalisa dari berbagai kondisi yang ada, misal dengan Analisis *Benefit Cost Ratio*, analisis *Return On Investment*, atau bahkan melakukan analisis kombiner terhadap data-data yang ada sampai pada kurun waktu tertentu secara langsung (*online*).

1.2 Tujuan

Tujuan pembuatan tugas akhir ini adalah menghasilkan suatu perangkat lunak yang akan menerapkan pemakaian teknologi OLAP sebagai sebuah *analysis tool*, sehingga pengguna dapat memanipulasi tampilan, memindahkan sudut pandang data, merubah urutan data, menambahkan field-field, dan melakukan perubahan parameter dengan menggunakan fungsi-fungsi *aggregate* untuk dapat menjawab kebutuhan analisis yang dikehendaki. Dengan kelebihan dari OLAP tersebut, maka diharapkan dapat membantu pemerintah (khususnya Bappeda Pemda Kabupaten Kediri) dalam menentukan kebijakan dalam menentukan kelayakan proyek-proyek pembangunan.

MILIK PERPUSTAKAAN ITS

1.3 Permasalahan

Permasalahan yang timbul sehingga diperlukan solusi pada tugas akhir ini adalah:

- Bagaimana membuat sebuah sistem yang dapat membantu pengguna untuk mengintegrasi, mengorganisasi, dan mengolah data sehingga akan membantu pengguna dalam mempercepat proses analisis pada nantinya.
- Bagaimana menampilkan data sehingga dapat dilakukan analisis dari berbagai sudut pandang, dimana data dapat di pindahkan atau dimanipulasi dalam berbagai bentuk tabulasi, rekapitulasi dan dapat digunakan dalam berbagai sudut pandang dalam sebuah kerangka OLAP.
- Bagaimana memberikan kemudahan kepada pengguna dalam memakai dan menangkap logika hasil dari aplikasi OLAP, dimana reporting yang dihasilkan OLAP mendekati reporting manual (*spreadsheet*) dan mendekati logika pengguna untuk melihat sebuah data.
- Bagaimana membuat sebuah aplikasi yang dapat menyimpan konfigurasi proses analisis yang telah dilakukan supaya dapat digunakan kembali untuk proses analisis serupa selanjutnya tanpa harus melakukan re-definisi sehingga akan menjadi sebuah standar analisis.

1.4 Batasan Permasalahan

Dari permasalahan diatas, maka batasan dalam tugas akhir ini adalah:

- Analisis studi kelayakan dari model fungsional ini hanya ditinjau dari *financial aspect*. Penyusunan model fungsional ini dimaksudkan sebagai kajian pengembangan sistem informasi pada tahap *planning* dan *analysis*.

- Data struktur proyek yang ada pada studi kasus merupakan data *sample* yang diambil dari contoh-contoh yang ada di lapangan, untuk mendukung fungsionalitas perangkat lunak. Pengambilan *sample* data dilakukan di lingkup Bappeda Pemda Kabupaten Kediri. Beberapa data kinerja keuangan yang tidak didapatkan dari Bappeda diambil dari beberapa kajian pada literatur yang digunakan
- Karena aplikasi ini dipakai oleh pengguna yang berposisi dibagian perencanaan pada Bappeda, maka aplikasi ini merupakan aplikasi berbasis *windows*, bukan merupakan aplikasi berbasis *web*.
- Aplikasi OLAP yang dibuat pada tugas akhir ini merupakan sebuah prototipe yang dibangun dengan menggunakan bahasa pemrograman ***Visual Basic 6.0***, dan karena data-data yang akan disimpan merupakan data-data yang frekuensi perubahannya rendah (bukan data transaksional), maka database yang digunakan adalah ***Microsoft Access Database (mdb)***.

1.5 Metodologi

Pembuatan tugas akhir ini dilakukan dengan mengikuti metodologi sebagai berikut:

- ***Studi literatur***

  Pada tahap ini akan dipelajari konsep-konsep tentang *Online Analitycal Processing* (OLAP) dan hal-hal lain yang berkaitan OLAP. Konsep ini didapat baik dari buku-buku literatur, paper maupun beberapa artikel di internet.

Pada tahap ini juga akan dipelajari metodologi dan algoritma yang akan digunakan dalam pembuatan perangkat lunak sehingga membantu pada tahap perancangan dan pembuatan perangkat lunak.

- ***Perancangan Sistem dan Aplikasi***

Perancangan Desain Database

Pada tahap ini akan ditentukan arsitektur database OLAP yang akan dipergunakan dalam aplikasi dan juga sebuah struktur database yang akan menyimpan aktivitas-aktivitas yang dilakukan pengguna terhadap sebuah dokumen analisis.

Perancangan Antarmuka

User interface yang bersifat *GUI (Graphical Pengguna Interface)* dan *user friendly* dirancang untuk memudahkan bagi pengguna dalam menggunakan aplikasi ini.

- ***Pembuatan Perangkat Lunak***

Dalam tahap ini, dilakukan implementasi berdasarkan rancangan yang telah dibuat pada tahap sebelumnya.

- ***Uji Coba dan Evaluasi***

Pada tahap ini aplikasi yang telah dibuat diuji dengan parameter sebagai berikut :

- Kemampuan dalam melakukan operasi-operasi standar OLAP terhadap data-data nilai yang ada.
- Kemampuan menyimpan dan menampilkan kembali konfigurasi proses analisis yang telah dilakukan.

- ***Penyusunan Buku Tugas Akhir***

  Pada tahap terakhir ini disusun buku sebagai dokumentasi dari penyelesaian Tugas Akhir.

1.6 Sistematika Pembahasan

Pembahasan yang akan disajikan dalam tugas akhir ini, akan dibagi dalam beberapa bab sebagai berikut:

Bab I, *Pendahuluan*, berisi latar belakang, tujuan, permasalahan, batasan permasalahan, metodologi dan sistematika pembahasan dari tugas akhir.

Bab II, *Teori Penunjang*, akan dibahas dasar-dasar ilmu yang mendukung pengerjaan tugas akhir ini, seperti pengertian dari sistem OLAP (*drill-down, slicing* dan *dicing, consolidation*), *Relational Database feasibility study* dalam proyek Pemda (termasuk perhitungan analisis keuangan) dan juga *System Development Life Cycle* sebagai metode dalam pengembangan perangkat lunak.

Bab III, *Pengenalan Proses Perencanaan Bappeda*, dalam bab ini akan dipaparkan permasalahan yang terjadi berkaitan dengan sistem yang ada di lingkungan Bappeda sekarang dan sistem yang diharapkan Bappeda di masa mendatang.

Bab IV, *Analisis Kebutuhan Sistem*, bab ini berisi identifikasi kebutuhan pada proses perencanaan di Bappeda dan obyektif sistem OLAP yang akan dibuat sesuai dengan hasil identifikasi tersebut, dan juga berkaitan dengan analisis aspek keuangan.

Bab V, *Perancangan Sistem*, bab ini akan membahas mengenai asumsi model sistem sesuai dengan *requirement* hasil survei lapangan.

Bab VI, ***Pembuatan Aplikasi Sistem Feasibility Study berbasis OLAP***, pada bab ini akan dilakukan pembuatan aplikasi berdasarkan model desain serta studi kasus analisis kelayakan proyek Pemda Kabupaten Kediri yang telah dibuat sebagai hasil survei lapangan.

Bab VII, ***Uji Coba dan Analisis Hasil***, akan dilakukan uji coba aplikasi pada studi kasus sistem yang ada dan kemudian dilakukan analisis terhadap hasil uji coba tersebut.

Bab VIII, ***Penutup***, berisi kesimpulan yang dapat diambil dari pengerjaan tugas akhir ini beserta saran untuk proses pengembangan selanjutnya.

BAB II

TEORI PENUNJANG

BAB II
TEORI PENUNJANG

Bab ini akan menguraikan hal-hal yang menunjang penyusunan tugas akhir ini. Dalam bab ini penulis akan menjelaskan tentang konsep dasar sebuah analisis studi kelayakan khususnya dalam perencanaan pembangunan di Bappeda, *Relational Database,* OLAP, dan Metodologi pembuatan *software* yang dipergunakan yaitu *System Development Life Cycle* (SDLC). Adapun perancangan dan pembuatan sistem dalam pembuatan tugas akhir ini merupakan implementasi dari hal-hal tersebut.

2.1 Feasibility Study

Studi Kelayakan *(Feasibility Study)* merupakan salah satu cara yang dipakai pada proses perencanaan untuk dapat menentukan apakah sebuah program/proyek yang direncanakan tersebut dapat dilaksanakan. Analisis studi kelayakan dilakukan terhadap permasalahan-permasalahan yang sering timbul di lapangan. Demikian halnya peranan sebuah studi kelayakan dalam perencanaan pembangunan-pembangunan yang harus dilakukan Pemerintah. Fokus utama studi kelayakan proyek terpusat pada 4 macam aspek, yaitu (1) Aspek pasar dan pemasaran barang atau jasa yang dihasilkan (2) Produk, teknis, dan teknologi (3) Manajemen dan Sumber Daya Alam dan (4) Keuangan dan ekonomi (SUTOJO 2000). Evaluasi aspek keuangan dan ekonomi mencakup penghitungan anggaran investasi yang dibutuhkan untuk membangun dan mengoperasikan proyek. Oleh sebab itu, pada dasarnya proses ini dapat dipakai juga oleh investor dalam melakukan analisis kelayakan investasi dari sisi finansial.

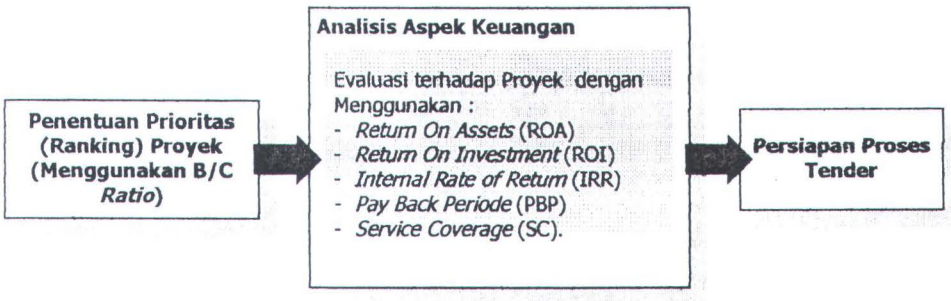
MILIK PERPUSTAKAAN

2.1.1 Analisis Keuangan dalam *Feasibility Study*

Proses pembangunan merupakan salah satu kegiatan utama bagi pemerintah daerah disamping pelayanan dan operasional birokrasi internal. Dalam melakukan pembangunan pemerintah sebagai wakil rakyat dituntut untuk melaksanakannya dengan efektif. Artinya, harus tepat sasaran dengan target yang jelas dan bisa diukur hasil dan dampaknya.

Analisis keuangan merupakan salah satu faktor penting dalam sebuah studi kelayakan proyek-proyek tersebut, karena pada dasarnya pada sebuah proses investasi calon investor lebih tertarik untuk mengevaluasi kelayakan suatu bisnis dengan melihat analisis keuangannya saja. Hal ini dapat dimaklumi mengingat bagaimanapun orientasi dari suatu bisnis adalah profit dan kinerjanya pun lebih "dihargai" dengan indikator keuangannya.

Sebuah Analisis Keuangan dalam studi kelayakan perencanaan pembangunan yang dilakukan oleh Bappeda dapat diilustrasikan seperti gambar berikut:



Gambar 2.1 Ilustrasi Feasibility Study Proyek Pemerintah.

2.1.2 Indikator-Indikator Keuangan

Pelaksanaan proyek-proyek pemerintah secara esensi memiliki karakteristik yang berbeda dengan proyek-proyek swasta. Proyek-proyek swasta senantiasa diukur berdasarkan nilai keuntungan yang dijanjikan, sedangkan pada proyek-proyek

pemerintah kriteria kelayakannya tidak selamanya bisa atau harus diukur berdasarkan nilai keuntungannya. Karena proyek-proyek pemerintah sering kali memang tidak mungkin diukur berdasarkan nilai-nilai rupiah yang jelas maka analisis kelayakannya dinyatakan dalam ukuran manfaat umum yang bisa ditimbulkannya.

Ada beberapa sumber pembiayaan yang bisa dipakai oleh pemerintah untuk membiayai proyek-proyeknya. Secara garis besar sumber-sumber tersebut bisa diklasifikasikan menjadi 4, yaitu:

1) Bagian sisa lebih perhitungan anggaran tahun yang lalu.
2) Bagian Pendapatan Asli Daerah (PAD) sendiri, yang terdiri atas: pajak daerah, retribusi daerah, bagian laba usaha daerah, lain-lain pendapatan.
3) Pendapatan yang berasal dari pemberian pemerintah dan atau instansi yang lebih tinggi, yang meliputi: bagi hasil pajak, bagi hasil bukan pajak, belanja rutin daerah, dana pembangunan daerah, dan penerimaan lainnya.
4) Bagian pinjaman pemerintah daerah, antara lain: pinjaman dari pemerintah pusat dan pinjaman dari keuangan dalam negeri.

Analisis aspek keuangan yang dilakukan pada proses studi kelayakan perencanaan proyek-proyek pemerintah, khusunya dalam sebuah investasi, dilakukan berdasarkan nilai-nilai sebagai berikut, yang secara rinci penggunaan rumusan-rumusan tersebut akan dijelaskan pada bab 5:

2.1.2.1 *Return On Total Assets* (ROA)

ROA merupakan perbandingan antara *net income* dengan *total assets*, dengan rumus:

$$ROA = \frac{NetIncome}{TotalAssets}$$

2.1.2.2 *Return On Investment* (ROI)

ROI merupakan perbandingan dari pemasukan (*income*) pertahun terhadap dana investasi. Secara umum banyak sekali rumusan untuk menentukan nilai ROI, namun akan digunakan rumusan dibawah ini, karena memperhitungkan semua investasi sepanjang tahun, bukan hanya investasi di saat awal proyek semata.

$$ROI = \frac{NetIncome}{InvestasiAwal}$$

2.1.2.3 *Internal Rate of Return* (IRR)

IRR merupakan sebuah tingkat pengembalian yang akan membuat *cash flow* selama proyek menjadi bernilai 0 (nol) untuk *present value*. IRR akan didapatkan jika NPV=0. Perhitungan matematis untuk mendapatkan IRR-nya akan dapat diturunkan dari persamaan NPV=0 ini. Perhitungan IRR adalah sebagai berikut:

$$IRR^* = IRR_j - \frac{NPVTotal_j(IRRj - IRR_{j-1})}{NPVTotal_j - NPVTotal_{j-1}}$$, dimana NPV dihitung dari:

$$NPVTotalj = \sum_{t=0}^{n} \frac{NF_t}{(1+IRR_j)t}$$

2.1.2.4 *Payback Periode* (PBP)

PBP adalah jangka waktu yang diperlukan untuk mengembalikan modal sebuah investasi, dihitung dari aliran kas bersih (*net*). Maka perumusannya adalah sebagai berikut:

$$PBP = (n-1) + \left[NF0 - \sum_{t=1}^{n-1} NF1 \right] \left(\frac{1}{NFn} \right)$$

2.1.2.5 *Service Coverage* (SC)

SC merupakan rasio penggunaan layanan pemerintah daerah oleh penduduk. Perumusannya adalah sebagai berikut:

$$SC = \frac{PgL}{PPL}$$

2.1.3 Penentuan Prioritas (*Ranking*) Proyek

Karena tiap sumber pendanaan proyek mempunyai kriteria (indikator) tersendiri, maka proses pemilihan prioritas proyek tidak bisa dilakukan langsung pada semua proyek. Proses prioritasisasi akan dilakukan persumber dana terlebih dahulu.

Berikut ini adalah beberapa tinjauan yang dipergunakan Bappeda dalam melakukan prioritasisasi pelaksanaan proyek.

2.1.3.1 *Benefit, Disbenefit, Cost*

Benefit atau manfaat adalah semua dampak positif yang akan dirasakan oleh masyarakat umum dengan terlaksananya suatu proyek. Disbenefit adalah ketidakmanfaatan atau dampak negatif yang menjadi konsekuensi bagi masyarakat umum dengan berdirinya atau berlangsungnya proyek tersebut. Sedangkan *cost* adalah biaya yang dikeluarkan untuk melakukan konstruksi, operasional dan perawatan proyek. Suatu proyek akan dipertimbangkan untuk dilakukan jika nilai keuntungan yang didapatkan lebih tinggi daripada nilai kerugian yang diakibatkan oleh proyek tersebut.

Untuk evaluasi terhadap sebuah proyek harus diingat bahwa pemilik proyek tersebut adalah masyarakat dan yang harus mengeluarkan *cost* adalah pemerintah. Namun perlu diingat tentang adanya kontrak sosial bahwa hakikatnya pemerintah

adalah sekumpulan orang yang diberikan tugas dan wewenang oleh masyarakat untuk menjalankan pekerjaan yang tidak dapat dikerjakan oleh masing-masing individu di masyarakat.

Sebagai contoh, misalkan pemerintah akan membangun jalan tol yang menghubungkan 2 kota yang akan melewati daerah pertanian yang cukup luas. Dengan proyek jalan tol ini, maka manfaat, dampak negatif, maupun ongkos-ongkos yang mungkin timbul sebagai berikut.

Tabel 2.1 Contoh Benefit, Disbenefit, Cost

Dampak Positif (benefit) bagi masyarakat umum
▪ Penurunan biaya operasional kendaraan, termasuk bahan bakar ▪ Waktu perjalanan menjadi lebih singkat dan lancar ▪ Peningkatan keamanan lalu lintas ▪ Kemudahan mengendarai kendaraan ▪ Peningkatan harga tanah di sekitar jalan tol
Dampak negatif (*disbenefit*) bagi masyarakat umum
▪ Pengurangan lahan pertanian ▪ Terganggunya saluran air untuk irigrasi ▪ Peningkatan polusi udara
Ongkos yang harus ditanggung oleh Pemerintah
▪ Ongkos konstruksi ▪ Ongkos perawatan ▪ Ongkos administratif
Pendapatan bagi Pemerintah
▪ Pendapatan dari iuran (toll) pemakai jalan ▪ Peningkatan pajak akibat meningkatnya nilai tanah di sekitar jalan tol

Berbagai manfaat yang bisa muncul dari suatu proyek memang tidak semuanya dapat dikuantifikasi apalagi dalam nilai mata uang. Prinsip yang harus dipegang dalam melakukan analisis manfaat-biaya bukan semata-mata mengukur

nilai-nilai manfaat tersebut dalam nilai mata uang, tetapi yang lebih penting adalah melakukan konversi terhadap nilai-nilai manfaat dan biaya tersebut.

2.1.3.2 Benefit Cost Analysis

Analisis manfaat biaya (*benefit cost analysis*) adalah analisis yang umum digunakan untuk mengevaluasi proyek-proyek pemerintah. Analisis ini adalah cara praktis untuk menaksir nilai manfaat sebuah proyek, dimana seharusnya untuk melakukan analisis sejenis ini pada dasarnya diperlukan tinjauan panjang dan luas.

Tinjauan yang panjang dalam hal ini berarti mengevaluasi proyek tersebut selama horizon perencanaan atau umurnya, yang mana biasanya akan jauh lebih panjang dibandingkan yang terjadi pada proyek-proyek swasta. Tinjauan yang luas berarti semua efek ongkos-ongkos maupun manfaat harus dilihat dan dianalisis. Ini perlu dilakukan karena biasanya proyek-proyek pemerintah secara langsung atau tidak langsung akan mempengaruhi orang banyak. Pengaruh ini bisa positif atau negatif.

Suatu proyek akan dikatakan layak atau bisa dilaksanakan *jika rasio antara manfaat terhadap biaya yang dibutuhkannya lebih besar dari satu*. Oleh karenanya, dalam melakukan analisis manfaat dan biaya kita harus berusaha melakukan kuantifikasi manfaat dari suatu usulan proyek, bila perlu dalam bentuk satuan mata uang. Namun, karena uang itu mempunyai sifat berubah nilai-nya terhadap waktu, maka selanjutnya elemen-elemen rupiah tersebut harus disamakan nilai-nya dengan cara membuat semua elemen rupiah tersebut ke dalam skala waktu atau masa yang sama, misalkan: masa sekarang (*present*), masa depan (*future*), atau masa sepanjang tahun (*annual*). Metode yang dapat digunakan adalah *present-worth*, *future-worth*, dan *annual-worth*. Perhitungan B/C Ratio adalah sebagai berikut:

$$B/C = \frac{(benefits - disbenefits)}{costs} \qquad \text{atau} \qquad B/C = \frac{(B-D)}{C}$$

2.1.3.3 Analisis *Benefit Cost Ratio* untuk Membandingkan Alternatif

Proses pengambilan keputusan pada proyek-proyek pemerintah juga akan melalui tahapan-tahapan yang sistemastis seperti halnya yang dilakukan oleh proyek-proyek swasta. Jadi, sebelum diputuskan untuk dilaksanakan, sebuah usulan proyek pemerintah harus dievaluasi mulai dari tahapan identifikasi kebutuhan, pemunculan alternatif, dan pemilihan alternatif yang terbaik. Hanya saja, pada proyek-proyek pemerintah, pemilihan alternatif yang terbaik tidak dilakukan pada besarnya profit yang bisa dihasilkan oleh proyek tersebut, namun lebih ditekankan pada manfaat atau kesejahteraan umum yang bisa diberikan kepada masyarakat.

Sebagai contoh, misalnya pemerintah sedang mempertimbangkan untuk membangun tempat olahraga atau perustakaan. Dana yang tesedia hanya cukup untuk membangun salah satu dari keduanya. Pemilihan alternatif-alternatif ini tidak bisa dilakukan atas dasar profit atau keuntungan karena memang proyek ini tidak dibangun untuk orientassi keuntungan, melainkan untuk kesejahteraan masyarakat. Oleh karenanya pemilihannya harus dirancang sedemikian rupa sehingga alternatif yang terpilih adalah alternatif yang menjanjikan manfaat atau kesejahteraan yang lebih besar.

Hal yang penting dipahami adalah alternatif-alternatif yang *dependent* (*mutually exclusive* atau saling meniadakan) dengan yang *independent* akan mempunyai prosedur perbandingan yang berbeda. Jika alternatif yang yang diperbandingkan tersebut bersifat *dependent*, maka dari masing-masing alternatif tersebut perlu diperbandingkan dengan menggunakan analisis manfaat dan biaya.

Namun jika ternyata alternatif yang ada tidak mempunyai ketergantungan, maka semua alternatif tersebut diperbandingkan dengan alternatif tidak melakukan apa-apa (*do nothing alternative*).

2.2 Online Analytical Processing

2.2.1 OLAP

OLAP (*Online Analytical Processing*) merupakan sebuah teknologi perangkat lunak yang memudahkan seorang analis, manajer, maupun kalangan eksekutif untuk mendapatkan wawasan akan sebuah data dengan cepat, konsisten, dengan representasi data yang interaktif dengan adanya kemungkinan berbagai penampakan informasi yang telah ditransformasikan dari sekumpulan data yang merefleksikan keadaaan sebenarnya dari sebuah organisasi.

Fungsi OLAP diperlihatkan dengan analisis multidimensional yang dinamis dari sekumpulan data untuk mendukung proses analisis dan aktivitas pencarian atau navigasi, antara lain:

a. kalkulasi dari data baik secara hirarki maupun non-hirarki dengan beberapa model penampakan.
b. analisis *trend* yang terjadi selama beberapa kurun waktu.
c. melakukan proses penjabaran data sampai ke level terbawah dari hirarki data untuk mendapatkan detail dari data.
d. mengubah-ubah konfigurasi dimensi data untuk proses pembandingan (*comparison*).

Beberapa operasi-operasi yang biasanya digunakan dalam aktivitas analisis seperti yang telah disebutkan sebelumnya, antara lain:

- ***Roll-up***, melakukan rekapitulasi terhadap beberapa level dalam sebuah hirarki dimensi.
- ***Drill-down***, merupakan kebalikan dari proses *roll-up* yaitu proses mendapatkan informasi detail dari sebuah hirarki dimensi .
- ***Pivoting***, pemindahan posisi dimensi untuk memudahkan analisis data dalam berbagai sudut pandang. Hasil dari sebuah proses *pivoting* disebut *cross-tabulation.*
- ***Slicing***, pemilihan himpunan data berdasarkan satu atau lebih dimensi.
- ***Dicing***, sebuah proses untuk menentukan *range* dari himpunan data.

2.2.2 Multidimensional Data Cube

Berikut ini akan dijelaskan alternatif lain untuk menampilkan sebuah data multidimensional. Sebagai contoh bagaimana cara terbaik untuk menampilkan sebuah query berikut. "Berapa jumlah total pendapatan yang dihasilkan berdasar penjualan properti di setiap kota pada setiap triwulan di tahun 1997?". Data pendapatan dapat ditampilkan dengan baik pada sebuah tabel yang terdiri dari 3 kolom, seperti pada gambar (2.2a)

Akan tetapi data ini akan lebih mudah ketika ditampilkan dalam sebuah matrik dua dimensi dengan data kota dan data waktu (triwulan) sebagai dimensinya. Seperti yang terlihat pada gambar (2.2b). Pemilihan model representasi bergantung kepada query yang diberikan oleh pengguna. Sebagai contoh, jika pengguna memberikan sebuah query sebagai berikut: "Berapa total pendapatan tahunan untuk setiap kota?" atau "Berapa pendapatan rata-rata untuk setiap kota?", maka hal ini akan mengakibatkan proses pengambilan banyak data dan terjadinya proses agregasi terhadap data bersangkutan. Jika berhadapan sebuah basisdata yang berukuran besar

dan berisi ribuan kota, maka waktu yang dibutuhkan oleh sebuah RDBMS untuk melakukan proses kalkulasi ini menjadi sangat signifikan. Sebuah RDBMS pada umumnya dapat melakukan scanning terhadap beberapa ratus baris dalam setiap detik. Sebuah multidimensional-DBMS pada umumnya dapat melakukan proses agregasi ini rata-rata 10000 baris /detik atau lebih.

City	Time	Total
Glasgow	Q1	29726
Glasgow	Q2	30443
Glasgow	Q3	30582
Glasgow	Q4	31390
London	Q1	43555
London	Q2	48244
London	Q3	56222
London	Q4	45632
Aberdeen	Q1	53210
Aberdeen	Q2	34567
Aberdeen	Q3	45677
Aberdeen	Q4	50056
.....	.....	.....
.....	.....	.....

(a)

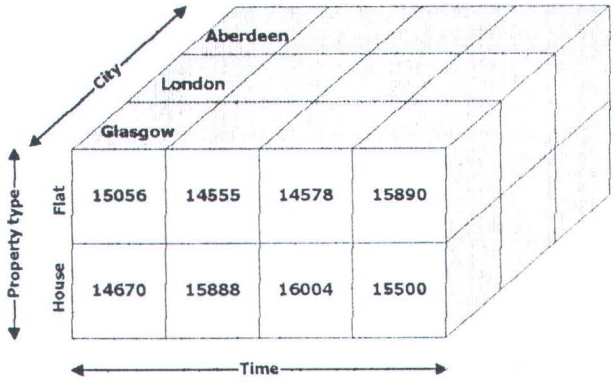
Property	City	Time	Total
Flat	Glasgow	Q1	15056
House	Glasgow	Q1	14670
Flat	Glasgow	Q2	14555
House	Glasgow	Q2	15888
Flat	Glasgow	Q3	14578
House	Glasgow	Q3	16004
Flat	Glasgow	Q4	15890
House	Glasgow	Q4	15500
Flat	London	Q1	19678
House	London	Q1	23877
Flat	London	Q2	19567
House	London	Q2	28677
.....	.....	.....	.....
.....	.....	.....	.....

(c)

City

Time \ City	Glasgow	London	Aberdeen	.....
Q1	29726	43555	53210	.....
Q2	30443	48244	34567	.....
Q3	30582	56222	45677	.....
Q4	31390	45632	50056	.....

(b)



(d)

Gambar 2.2 Multidimensional data view.

Misalnya data pendapatan dengan penambahan pada dimensinya, sebagai contoh penambahan tipe properti yang terjual. Dalam kasus ini, data yang merepresentasikan total pendapatan yang dihasilkan oleh penjualan pada setiap tipe

properti (sebagai contoh tipe Flat dan tipe House), berdasarkan kota, dan berdasarkan waktu tiap triwulan. Sekali lagi, data ini dapat ditampilkan kedalam sebuah tabel yang terdiri dari 4 kolom. Seperti yang terlihat pada gambar (2.2c).

Akan tetapi data ini akan lebih mudah dimengerti jika ditampilkan dalam sebuah *three-dimensional cube* (kubus tiga dimensi), seperti yang terlihat pada gambar (2.2d). Cube ini merepresentasikan data sebagai kumpulan dari sel-sel (*cells*) dalam bentuk sebuah array dengan menampilkan data total pendapatan sebagai hubungan dari dimensi tipe properti, kota dan waktu tiap triwulan. Akan tetapi tabel didalam RDMBS hanya dapat menyimpan data *multidimensional* dalam bentuk dua dimensi.

Basisdata OLAP menggunakan struktur multidimensi untuk menyimpan data dan hubungan-hubungan antar data. Struktur *multidimensional* paling bagus jika divisualisasikan dalam bentuk *cube* dan *cubes within cubes* (kubus dalam kubus). Setiap sisi dari sebuah *cube* menunjukkan sebuah dimensi.

Basisdata multidimensional adalah sebuah cara menampilkan dan memanipulasi elemen-elemen data yang memiliki banyak keterhubungan secara kompak dan mudah untuk dipahami. Cube dapat dikembangkan dengan menambahkan dimensinya, sebagai contoh data jumlah staf penjualan pada setiap kota. Cube mendukung proses-proses aritmatika matrik yang akan mengakibatkan cube ini dapat menampilkan data rata-rata pendapatan untuk setiap staf penjualan dengan cara melakukan sebuah opersi matrik pada setiap sel yang bersangkutan dalam cube *(Rata-rata total pendapatan untuk setiap staf penjualan = Total pendapatan/ Jumlah staf penjualan)*

Waktu respon dari sebuah query multidimensional bergantung pada seberapa banyak jumlah sel yang harus ditambahkan pada setiap proses yang berlangsung (*on the fly*). Seiring dengan bertambahnya dimensi, jumlah dari sel dari sebuah cube meningkat secara eksponensial. Keunggulan dari query multidimensional adalah data terkalkulasi atau data-data laporan. Oleh karena itu, solusi untuk membangun sebuah basisdata multidimensional yang efisien adalah untuk melakukan pre-agregasi atau konsolidasi terhadap semua data total atau sub total pada semua dimensi. Proses pre-agregasi ini akan sangat berarti, karena pada dasarnya dimensi-dimensi itu adalah sebuah hirarki. Sebagai contoh, dimensi waktu dapat terdiri dari hirarki tahun, triwulan, bulan, minggu dan hari, dan dimensi lokasi dapat terdiri dari kantor cabang, area, kota dan negara. Dengan diketahuinya definisi hirarki dari sebuah dimensi memberikan kemampuan untuk melakukan proses pre-agregasi dan sebaliknya. Proses *drill-down* (penjabaran) sebagai contoh dari data pendapatan tahunan, pendapatan triwulan sampai pendapatan bulanan. (CONNOLLY, 1998 : 950)

2.2.3 Kategorisasi OLAP

Alat-alat OLAP dikelompokkan berdasarkan arsitektur dari basisdata yang digunakan sebagai sumber dalam proses analisis. Ada beberapa jenis kelompok OLAP antara lain (CONNOLLY, 1998 : 955-957):

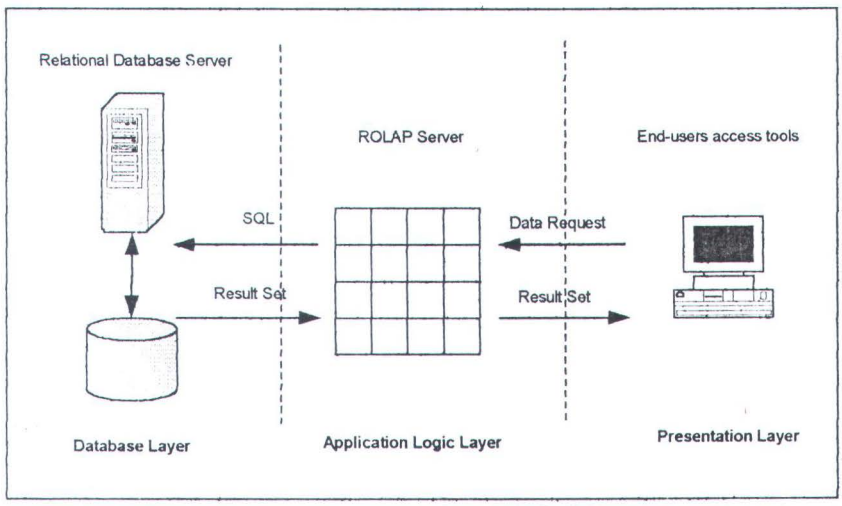
2.2.3.1 ROLAP

Relational OLAP (ROLAP) merupakan salah satu teknologi OLAP ytang berkembang paling cepat saat ini. Teknologi ini mendukung produk-produk RDBMS —basisdata relasional— melalui penggunaan dari sebuah metadata layer, mekanisme ini menghindarkan para *developer* terhadap sebuah kebutuhan yang menyebabkan

desain struktur data multidimensional yang statis. Konsep ini memberikan kemudahan kepada para developer untuk membuat banyak multidimensional *view* dari tabel-tabel dua dimensi.

Basisdata relasional adalah sebuah teknologi basisdata yang berdasarkan konsep relasional yang terstruktur dimana struktur tersebut terdiri dari elemen-elemen antara lain *relation*, *tuple*, *attribute*, *domain*, *degree* dan *cardinality*. Elemen yang seringkali berinteraksi dengan *developer* adalah relasi yang adalah sebuah tabel yang terdiri dari baris (*tuple*) dan kolom (*attribute*).

Untuk memperbaiki kinerjanya, beberapa produk ROLAP meningkatkan kemampuan SQL engine untuk mendukung kompleksitas proses analisis multidimensional. Sementara itu produk lain menggunakan pendekatan dengan menggunakan konsep denormalisasi desain basis data seperti penggunaan skema star. Arsitektur umum sebuah ROLAP terlihat pada gambar 2.3



Gambar 2.3 Arsitektur ROLAP

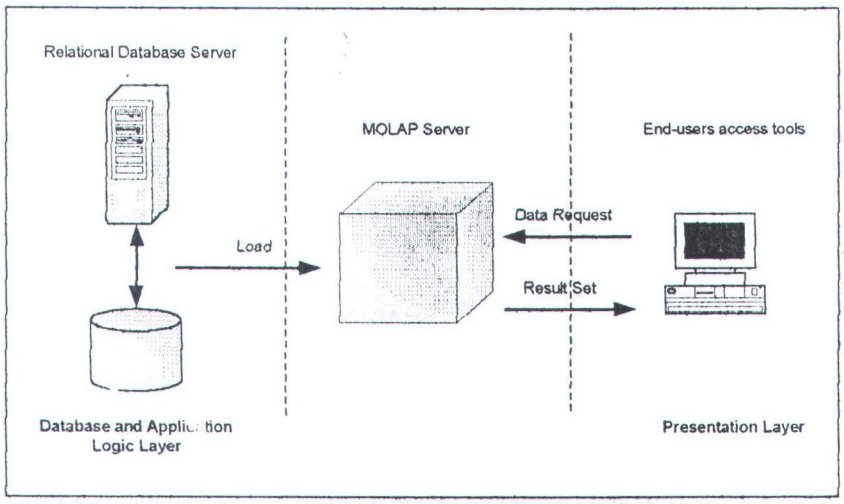
Permasalahan-permasalahan pada proses pengembangan ROLAP adalah sebagai berikut:

- Para developer perlu untuk melakukan pengembangan *middleware* atau *application layer* khusus untuk memberikan kemudahan pada proses pengembangan sebuah aplikasi *multi-dimensional*, yaitu sebuah perangkat lunak yang pada akan melakukan proses konversi dari tabel-tabel dua dimensi menjadi sebuah struktur multidimensional.
- Penambahan fitur untuk memberikan pilihan untuk melakukan penyimpanan terhadap struktur-struktur multidimensional yang dipergunakan.

2.2.3.2 MOLAP

MOLAP menggunakan struktur data dan Sistem Basisdata multi-dimensional yang khusus dan spesifik untuk melakukan pengorganisasian, navigasi, dan analisis data. Untuk meningkatkan kinerja dari proses *query*, pada umumnya data sudah teragregasi dan disimpan sesuai dengan perkiraan penggunaannya. Struktur data MOLAP menggunakan teknologi *array* dan menggunakan teknologi penyimpanan yang efisien untuk meminimalisasi ukuran penggunaan *space* data pada media penyimpanan.

Alat-alat MOLAP akan memberikan kinerja yang sangat bagus ketika digunakan sesuai dengan desain dari struktur data dan pada kasus pengambilan keputusan yang spesifik. Aplikasi MOLAP memiliki *application layer* dan *presentation layer* yang spesifik juga terhadap sebuah kasus. Arsitektur umum dari sebuah MOLAP seperti pada gambar 2.4.



Gambar 2.4 Arsitektur MOLAP

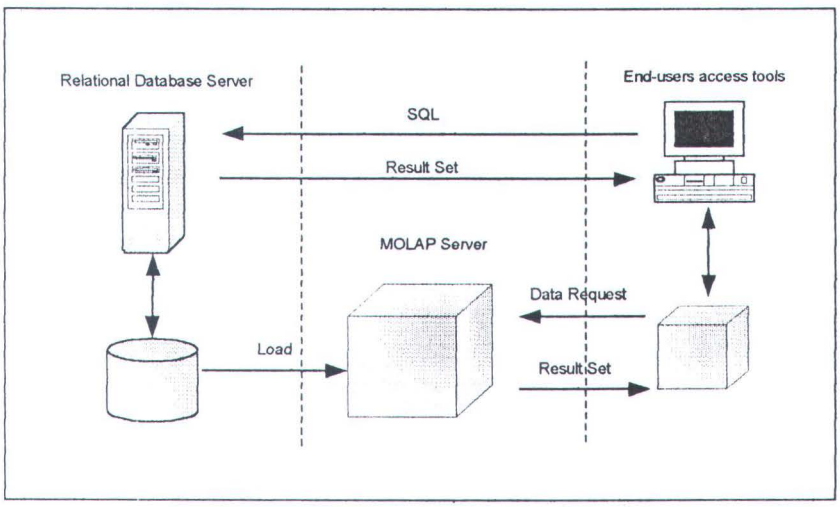
Permasalahan yang dihadapi dalam pengembangan sebuah alat MOLAP antara lain adalah:

- Struktur data sumber yang terbatas terhadap subyek dari proses analisis dan keterbatasan dalam memberikan akses kepada data-data detail. Permasalahan ini biasanya diselesaikan dengan menggunakan mekanisme yang memberikan kemampuan kepada alat MOLAP untuk mengkases data-data detail yang disimpan pada sebuah RDBMS. Solusi ini biasanya dipakai dalam konsep *Hybrid OLAP* (HOLAP) yang akan dijelaskan pada bagian selanjutnya.
- Keterbatasan navigasi dan analisis data. Hal ini disebabkan karena data telah didesain sesuai dengan kebutuhan yang didefinisikan sebelumnya. Oleh sebab itu perlu untuk melakukan kembali pengorganisasian data untuk secara optimal mendukung kebutuhan-kebutuhan baru.

- Aplikasi MOLAP membutuhkan keahlian-keahlian tertentu dan alat-alat khusus untuk membangun dan merawat basisdata dan hal ini menimbulkan meningkatnya biaya dan kompleksitas dari sumber daya pendukung.

2.2.3.3 MQE

Managed Query Environment (MQE) adalah sebuah teknologi OLAP yang relatif baru. Teknik ini memberikan kemampuan analisis yang terbatas, terhadap produk-produk RDBMS atau produk-produk MOLAP server. MQE mengirimkan data-data terpilih langsung dari sebuah sistem basisdata atau melalui sebuah MOLAP server ke pengguna atau ke server lainnya dalam bentuk sebuah data cube yang disimpan, dianalisis dan dirawat secara lokal pengguna atau lokal server bersangkutan.



Gambar 2.5 Arsitektur MQE

Para pengembang menawarkan teknologi ini sebagai sebuah teknologi yang relatif sederhana dan mudah untuk dipasang dan diatur dengan biaya dan perawatan yang lebih kecil. Arsitektur umum dari sebuah MQE terlihat pada gambar 2.5.

Permasalahan-permasalahan pada pengembangan sebuah MQE adalah sebagai berikut:

- Arsitektur ini menghasilkan *redundancy* data yang signifikan dan menyebabkan permasalahan pada beban jaringan untuk mendukung banyak pengguna.
- Kemampuan yang diberikan kepada setiap pengguna untuk mendefinisikan *data cube* masing-masing dapat menyebabkan tingkat inkonsistensi data yang cukup tinggi.
- Dengan teknologi ini jumlah data yang dapat di maintain menjadi sangat terbatas.

2.2.3.4 HOLAP

Ada lagi sumber yang menambahkan kategori OLAP yang bernama HOLAP (*Hybrid OLAP*). HOLAP merupakan kombinasi dari MOLAP dan ROLAP. Untuk penyimpanan *Cube* datanya. HOLAP menyimpan *cube* data dalam sebuah struktur relasional (ROLAP) dan data-data agregasi dalam sebuah multidimensional struktur (MOLAP). Jenis OLAP yang *hybrid* inilah yang nantinya akan dipakai dalam pengerjaan tugas akhir ini (AMO 2000 : 12).

2.3 Metodologi SDLC

Metodologi pengembangan sistem adalah kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, dan aturan-aturan yang digunakan untuk mengembangkan sebuah sistem (informasi). Dengan terdapatnya beberapa pendekatan metodologi didalam pengembangan sebuah sistem, maka dalam

pembuatan tugas akhir ini akan dipergunakan metodologi *System Development Life Cycle* (SDLC), dengan tahap-tahap sebagai berikut (DAVIS 1994: 11-16) :

1. ***Problem definition***

   Untuk dapat menyelesaikan dan mengatasi sebuah permasalahan yang sebelumnya belum kita mengerti, langkah pertama yang harus dilakukan dalam *SDLC* adalah proses *problem definition*. Hal ini dimaksudkan untuk mengidentifikasi masalah tersebut, menentukan sebab-sebab sehingga terjadi permasalahan tersebut, dan menentukan gambaran garis besar strategi yang akan dilakukan untuk mengatasi permasalahan tersebut. Tahap ini seharusnya dilakukan dengan cermat dan teliti, karena sebuah proses *problem definition* yang asal-asalan dan sekedarnya, akan memberikan jaminan bahwa sistem yang nantinya akan dibuat akan gagal untuk mengatasi permasalahan yang mungkin terjadi.

2. ***Analysis***

   Setelah permasalahan-permasalahan tersebut dapat terdefinisi dengan jelas dalam tahap *Problem Definition*, maka tahap selanjutnya adalah tahap Analysis. Tujuan dari tahap ini adalah untuk menentukan dengan pasti apa yang harus dilakukan utnuk mengatasi permasalah-permasalahan tersebut – *what must be done to solve the problem*–. Selama tahap ini, kita akan menentukan proses-proses penting, elemen-elemen data, obyek-obyek yang ada, dan entitas lain yang menjadi kunci penyelesaian permasalah tersebut tanpa mempedulikan bagaimana komponen-komponen tersebut akan diimplementasikan. Akhir dari tahap ini nantinya akan kita dapatkan sekumpulan *requirement specification*

PERPUSTAKAAN ITS

yang akan mendefinisikan dengan jelas apa yang nantinya harus dapat dilakukan oleh sistem yang hendak kita buat.

3. ***Design***

Jika tahap analisis telah selesai, dan kita telah mengetahui apa saja yang harus kita lakukan untuk mengatasi permasalahan tersebut, maka tahap selanjutnya adalah tahap *design*. Tahap desain ini bertujuan untuk menentukan bagaimana permasalahan tersebut harus kita atasi. Dalam tahap ini akan dibuat desain aplikasi dengan menggunakan standar notasi Data Alur Diagram (DAD), *Flowchart*, dan *Entity Relationship Diagram* (ERD).

4. ***Development***

Sebuah sistem akan dibuat selama fase *development*. Aplikasi akan dibuat, proses pengkodean dilakukan, *debug*, dan *testing*. Inisialisasi *database* dan komponen-komponen yang diperlukan.

5. ***Testing***

Setelah fase *development* dilalui, maka tahap selanjutnya adalah tahap *testing* atau uji coba. Tahap ini dilakukan untuk memastikan bahwa sistem atau aplikasi yang telah dibuat telah bekerja sesuai dengan fungsinya sesuai dengan desain yang telah dibuat. Uji coba ini akan dilakukan dengan mengambil beberapa studi kasus yang telah ditentukan.

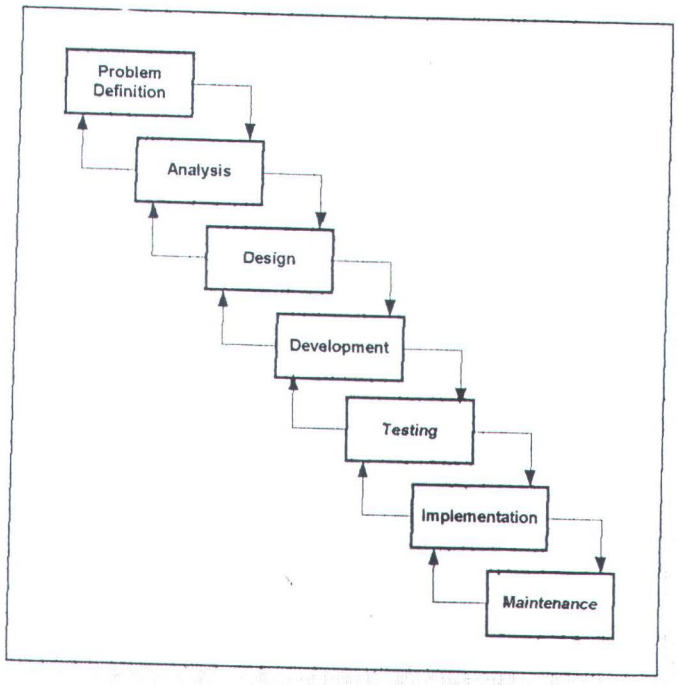
6. ***Implementation***

Setelah proses uji coba selesai dilakukan dan semua kesalahan telah diperbaiki, maka sistem/aplikasi tersebut siap untuk di-*release*.

7. *Maintenance*

Setelah sistem tersebut diimplementasikan sesuai dengan permintaan pengguna, maka tahap selanjutnya adalah proses *maintenance*. Dimana pada tahap ini bertujuan untuk menjaga bahwa sistem yang telah kita buat akan senantiasa dapat melakukan fungsinya untuk permasalahan-permasalahan yang telah didefinisikan. Hal ini dilakukan, karena terkadang pada saat uji coba masih ada *error* yang belum terdeteksi.

Metodologi pengembangan sistem dengan SDLC ini digambarkan sebagai berikut:



Gambar 2.6 System Development Life Cycle

BAB III

PENGENALAN PROSES PERENCANAAN BAPPEDA

BAB III
PENGENALAN PROSES PERENCANAAN BAPPEDA

Pada bab ini akan dipaparkan tentang sistem perencanaan pembangunan yang selama ini dilakukan oleh Bappeda Pemda Kabupaten Kediri beserta dengan harapan-harapan yang diinginkan untuk memperbaiki sistem yang selama ini ada. Pemaparan sistem yang telah ada di Bappeda Kabupaten Kediri ini merupakan langkah pertama —yaitu *problem definition*— dalam metodologi SDLC.

3.1 Sistem yang ada sekarang

Selama ini proses perencanaan pembangunan yang dilakukan oleh pihak Bappeda di Pemda Kabupaten Kediri masih konvensional. Segala proses yang berhubungan dengan perencanaan pembangunan tersebut harus dirapatkan dalam sebuah forum yang harus dihadiri seluruh elemen perencana dan disertai adanya sejumlah kertas-kertas (dokumen) yang berisi data-data untuk menganalisa usulan proyek yang direncanakan.

Proses perencanaan pembangunan yang dilakukan Bappeda dalam analisis kelayakan proyek sampai sekarang masih dilakukan secara manual dengan melakukan alat bantu aplikasi Microsoft Excel (Excel) dibantu dari bagian EDP (*Electronic Data Processing*). Analisis perencanaan dengan menggunakan Excel berarti pengguna harus mencari dan mengkombinasikan setiap data-data yang ada untuk dapat menunjang analisis yang dilakukannya. Pencarian data-data yang ingin ditampilkan itulah yang sering menjadi masalah, karena dengan banyaknya data,

berarti proses yang dibutuhkan akan semakin lama, dan akan menjadi kegiatan yang membosankan bagi pengguna.

Adakalanya, seorang pengguna yang berperan sebagai manajer perencanaan – yang memiliki sedikit literasi database– harus memanfaatkan jasa seorang staf EDP. Hal ini akan berarti pemborosan waktu dan *resource*, karena proses perantaraan data dari pengguna ke staf EDP sering menghambat proses perencanaan-perencanaan selanjutnya. Apalagi dalam proses perencanaan di Bappeda tersebut, melibatkan beberapa Kepala Sub Bagian (Kasubag), Kepala Sub Bidang (Kasubid), Kepala Bidang (Kabid), dan Kepala Dinas (Kadin) dari dinas-dinas yang bersangkutan. Permasalahan yang sering muncul adalah jika pada saat Rapat Koordinasi Pembangunan (Rakorbang) salah satu elemen perencana tersebut berhalangan hadir, maka akan menghambat proses perencanaan.

Jika setiap analisa dilakukan secara manual, maka dapat dibayangkan berapa banyak waktu yang harus dibutuhkan pengguna (staf perencanaan pembangunan) untuk mencari data-data yang dibutuhkan dalam proses analisa. Karena dengan semakin banyaknya data yang ada, kebutuhan akan informasi yang akan digunakan dalam proses analisis akan semakin banyak pula. Hal ini disebabkan oleh banyaknya kombinasi terhadap data-data yang ada untuk melakukan analisis dari berbagai sudut pandang, sehingga keputusan yang akan diambil selanjutnya akan benar-benar mendukung keputusan layak tidaknya suatu proyek yang direncanakan akan dilaksanakan.

Disebutkan diawal tulisan, bahwa semakin banyaknya proyek-proyek yang harus dilaksanakan Pemerintah Daerah, menuntut Pemerintah untuk melakukan penentuan prioritas terhadap proyek-proyek yang diusulkan. Sebuah sistem Studi

Kelayakan akan membantu Bappeda sebagai badan perencana pembangunan pemerintah dalam melakukan analisis terhadap situasi dan kondisi berkaitan dengan proyek yang akan dilaksanakan. Dalam Studi Kelayakan ini nantinya akan didapatkan suatu kesimpulan apakah suatu proyek tersebut **dapat** atau **layak** untuk dilaksanakan oleh pemerintah atau tidak.

3.1.1 Kebutuhan Data untuk Analisis

Data-data yang dibutuhkan untuk mendukung proses analisis kelayakan proyek mencakup data-data proyek ditinjau dari struktur proyek tersebut dan juga aspek keuangan berkaitan dengan pelaksanaan proyek tersebut.

3.1.1.1 Data Struktur Proyek

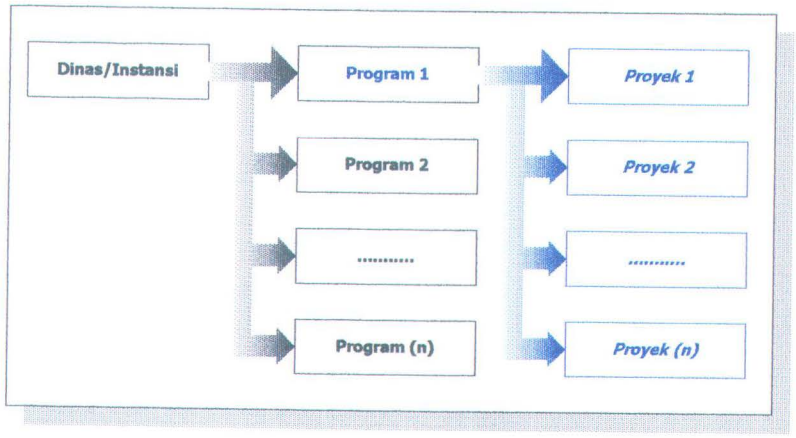
Data struktur proyek berisi kegiatan-kegiatan atau proyek-proyek pembangunan yang dilakukan oleh pemerintah daerah (Kabupaten Kediri) dalam setiap tahun perencanaan. Perencanaan proyek-proyek pembangunan tersebut merupakan kegiatan rutin dari pemerintah daerah dalam rangka melaksanakan aktivitas pembangunan daerahnya. Kegiatan tersebut berawal dari berbagai usulan dari desa dan dibawa sampai kabupaten dengan tahapan:

1) Dari desa, melalui rapat MUSBANG (Musyawarah Pembangunan) masyarakat memberikan usulan kebutuhan pembangunan yang terkait dengan desa yang bersangkutan.
2) Usulan dari desa ditampung di Kecamatan. Di Kecamatan usulan-usulan yang masuk diolah, dipilih dan dipilah berdasarkan faktor-faktor prioritas di dalam sebuah forum UDKP (Unit Daerah Kerja Kecamatan).
3) Hasil dari UDKP di Kecamatan, dibawa ke dinas atau instansi yang bersangkutan atau langsung dibawa ke tingkat Kabupaten.

4) Dinas atau instansi, selain berdasar masukan dari kecamatan, juga dapat mengajukan usulan proyek pembangunan ke tingkat kabupaten berdasarkan analisis kebutuhan yang telah diidentifikasikan oleh masing-masing dinas atau instansi.
5) Di tingkat kabupaten, usulan-usulan proyek pembangunan baik dari kecamatan maupun dinas atau instansi diolah dalam forum RAKORBANG (Rapat Koordinasi Pembangunan). Dari forum inilah beberapa proyek pembangunan dipilih dan mengalami proses penentuan prioritas berdasar tujuan pembangunan yang telah ditetapkan.

Sebenarnya proses perencanaan pada Bappeda Kediri biasanya merupakan penggabungan dari metode *bottom-up* dan *top-down*. Hal ini disebabkan karena seringkali dinas-dinas daerah tidak tahu kebutuhan daerahnya, sehingga dibutuhkan beberapa informasi dari pusat, disebabkan pula oleh adanya usulan yang memang telah direncanakan dari tingkat Pusat. Informasi dari pusat tersebut diberikan ke tiap-tiap propinsi di daerah yang bersangkutan kemudian dengan Rakorbang yang diadakan di tingkat propinsi (Rakorbang Tingkat I), akan didapatkan usulan yang akan diberikan ke tingkat Kabupaten. Selanjutnya adalah tugas Bappeda di Kabupaten untuk mensinkronkan usulan-usulan yang berasal dari pusat tersebut dengan usulan-usulan dari dinas-dinas yang ada di daerah.

Dari beberapa jenis data yang masuk, struktur proyek pembangunan yang digunakan adalah digambarkan seperti gambar 3.1. Dari Gambar 3.1 tersebut dapat diketahui bahwa sejumlah dinas atau instansi akan mempunyai beberapa program pembangunan. Masing-masing program pembangunan mempunyai sejumlah proyek.



Gambar 3.1 Struktur Proyek.

3.1.1 Elemen Perencana

Rakorbang yang diadakan setiap satu tahun sekali dipergunakan untuk membahas usulan-usulan proyek-proyek pembangunan yang nantinya akan dilaksanakan pada periode tahun berikutnya. Untuk mengadakan Rakorbang, setiap elemen perencana dari dinas-dinas atau instansi-instansi yang berkaitan harus hadir, hal ini sehubungan dengan kepentingan masing-masing elemen untuk mengajukan usulan dan dalam proses analisis. Syarat kehadiran bagi setiap elemen perencana inilah yang sering menjadi kendala dalam forum Rakorbang. Karena jika terdapat satu elemen saja yang tidak hadir maka kondisi ini akan memperlambat proses perencanaan.

Elemen-elemen perencana yang harus hadir pada setiap Rakorbang adalah pihak-pihak yang berada pada level *middle management*, antara lain Kepala Bidang (Kabid), Kepala Sub Bidang (Kasubid), Kepala Bagian (Kabag), Kepala Sub Bagian (Kasubag), dan Kepala Dinas (Kadin).

Selain itu elemen-elemen yang mendukung proses perencanaan pada Rakorbang tentu saja adalah pihak-pihak yang berkaitan dengan data-data yang

diperlukan untuk kelancaran proses perencanaan tersebut. Pihak-pihak pendukung tersebut adalah bagian Keuangan yang menyediakan laporan-laporan keuangan proyek dan bagian Data yang merekapitulasi usulan-usulan yang masuk ke Bappeda.

3.1.2 Sarana Pendukung

Sistem yang terkomputerisasi sebagai sarana pendukung proses perencanaan pembangunan telah terpasang di lingkungan Bappeda Pemda Kabupaten Kediri, walaupun sistem tersebut sampai saat ini belum terintegrasi. Komputer telah terpasang di setiap ruangan pihak-pihak *middle management* maupun dibeberapa tempat yang membutuhkan peranan komputer sebagai sarana pendukung kerja. Masing-masing komputer belum terhubung dalam sebuah LAN dengan mempergunakan *operating system* Windows (Windows 98).

3.2 Sistem yang diharapkan

Pemerintah Daerah dituntut untuk dapat melakukan perencanaan yang semakin tepat, cepat, akurat dan terintegrasi. Apalagi sejak diterapkannya sistem otonomi daerah, Pemerintah Daerah (Kabupaten Kediri) harus siap mengelola sendiri semua sumber daya dan potensi daerahnya, serta mengatur strategi berdasarkan tujuan yang diharapkan.

Sementara itu, kompleksitas perencanaan pembangunan di suatu daerah melahirkan tuntutan tentang sebuah standar pengendalian terhadap daftar usulan proyek. Standar pengendalian inilah yang berupa *template* atau *form feasibility study* (studi kelayakan) proyek pembangunan daerah yang diusulkan dinas-dinas sehingga dapat memudahkan Badan Perencanaan Pembangunan Daerah (Bappeda) dalam melakukan proses penyaringan, penentuan prioritas dan sinkronisasi perencanaan program-program pembangunan daerah.

Format Studi Kelayakan yang akan dibuat diharapkan dapat memenuhi semua masukan yang dibutuhkan Bappeda dalam perencanaan program-program pembangunan daerah melalui Rakorbang.

Dengan meningkatnya kompleksitas pembangunan yang harus dilaksanakan pemerintah, proses perencanaan proyek-proyek pembangunan tersebut diharapkan menjadi sebuah proses yang lebih cepat dan efektif. Sebuah proses perencanan yang cepat dan efektif tersebut akan dapat dilaksanakan jika:

- Seluruh elemen perencana yang berkepentingan hadir dalam forum Rakorbang. Masalah waktu merupakan kunci utama yang harus dipenuhi.
- Ditunjang dengan ketersediaan data-data yang akan membantu proses perencanaan. Ketersediaan data berkaitan dengan ada tidaknya data yang diperlukan pada saat dibutuhkan dan dapat tidaknya data yang dibutuhkan tersebut dapat dicapai. Dengan kata lain, sebuah sistem pengolahan data yang terintegrasi akan sangat mendukung proses perencanaan ini.
- Sistem dapat memberikan respon yang cepat terhadap sebuah proses analisis data.
- Adanya fungsi-fungsi standar yang dapat mempercepat pengguna untuk mendapatkan informasi-informasi yang dibutuhkan pada proses analisis. Fungsi-fungsi standar itu berkaitan dengan kebutuhan sebuah proses analisis terhadap data-data, baik itu data hasil rekapitulasi sampai data pada level detail. Hal ini akan memberikan fleksibilitas pada proses analisis data.
- Untuk beberapa proses analisis yang sering dan telah dilakukan dapat disimpan dan diambil kembali untuk proses analisis-analisis sejenis

selanjutnya, sehingga proses analisis selanjutnya akan lebih cepat dilakukan karena tidak perlu lagi didefinisikan dari awal.

- Didukung dengan visualisasi data yang bermacam-macam seperti dalam bentuk *spreadsheet*, grafik dan tampilan yang berbasis grafis (*Graphical User Interface*).

BAB IV

ANALISIS KEBUTUHAN SISTEM

BAB IV
ANALISIS KEBUTUHAN SISTEM

Dalam bab IV ini akan diberikan sejumlah analisis terhadap kebutuhan sistem yang diharapkan akan membantu meningkatkan kinerja proses perencanaan pembangunan di Kabupaten Kediri, khususnya di Bappeda. Analisis sistem yang akan dilakukan ini merupakan langkah kedua —yaitu *analysis*— dalam metodologi SDLC.

4.1 Analisis Kebutuhan

Sebuah perencanaan pembangunan memerlukan proses analisis yang lebih dalam lagi terhadap data-data yang tersedia, sehingga mampu menjadi sarana yang efektif untuk menetapkan kebijakan dan strategi pembangunan. Peranan data-data tersebut sebenarnya sangat penting, karena berfungsi antara lain untuk:

a. *Referensi*, sebagai rujukan dan bahan pertimbangan dalam menentukan kebijakan dan strategi pembangunan serta sasaran dari setiap kegiatan pembangunan;
b. *Monitoring*, sebagai bahan dalam melakukan pengendalian pelaksanaan pembangunan yang sedang diselesaikan;
c. *Evaluasi*, sebagai bahan untuk melihat tingkat perkembangan (kemajuan atau kemunduran) potensi dari suatu wilayah (kabupaten atau kecamatan) dalam suatu periode;

d. *Presentasi*, sebagai media penyampaian hasil-hasil pelaksanaan pembangunan dan potensi yang dimiliki oleh Kabupaten Kediri kepada masyarakat maupun kalangan industri atau investor.

Namun, dengan melihat kondisi-kondisi yang ada di Pemda Kediri dalam menanggapi sebuah usulan dari Dinas atau Instansi dalam rangka melaksanakan proses perencanaan pembangunan yang masih konvensional, maka akan menjadi suatu hal yang sangat sulit untuk dapat merealisasikan poin-poin tersebut di atas, walaupun dalam beberapa hal, telah dilakukan komputerisasi terhadap data-data, akan tetapi tetap akan menjadi suatu hal yang sulit jika data-data tersebut tidak terintegrasi. Terlebih lagi jika usulan-usulan tersebut semakin banyak, kompleks dan memiliki sifat kepentingan yang bermacam-macam.

Menganalisa harapan-harapan pihak Bappeda dalam memperbaiki proses perencanaan yang ada, maka seharusnya terdapat sebuah sistem yang nantinya akan dapat memenuhi kebutuhan-kebutuhan pihak Bappeda dalam melakukan sebuah proses perencanaan pembangunan yang cepat, efektif dan efisien. Sistem yang dimaksud beserta kebutuhan-kebutuhan yang ingin dipenuhi dengan adanya sistem bersangkutan adalah:

1) Mampu untuk mengatasi kesulitan proses perolehan data yang tersebar pada beberapa bagian atau departemen yang ada dengan bermacam-macam operator yang berbeda. Untuk mengatasi perbedaan fungsi dan perbedaan jenis data dari banyaknya bidang yang ada beserta staf-staf yang ada didalamnya, maka sebuah sistem yang mampu memanajemen pengguna

dari berbagai bidang maupun fungsi dan jenis data (*user management system*).

2) Mampu untuk menjamin ketersediaan data, berkaitan dengan ada tidaknya data yang diperlukan pada saat dibutuhkan dan dapat tidaknya data yang dibutuhkan tersebut dapat dicapai, maka sebuah sistem pengolahan data terkomputerisasi yang terintegrasi akan sangat mendukung proses perencanaan ini.

3) Mampu untuk memenuhi dan mendukung kebutuhan proses analisis data, maka diperlukan sebuah sistem yang mampu mempresentasikan data dalam berbagai sudut pandang. Penampakan data dalam berbagai sudut pandang ini akan dipenuhi dengan adanya kemampuan OLAP dalam melakukan *dimensioning* terhadap data secara "*on the fly*".

Untuk selanjutnya, dalam pengerjaan tugas akhir ini, penulis akan melakukan perancangan dan pembuatan *prototype* sistem yang memiliki kemampuan untuk melakukan manajemen *user*, entri data yang terintegrasi dan presentasi data secara multidimensi.

4.1.1 Analisis User Management

User Management merupakan administrasi *user* yang memiliki hak akses pada sistem yang akan dibuat. User yang dimaksud adalah orang-orang yang mempunyai kepentingan terhadap data entri (baik itu berupa data struktur proyek maupun data keuangan proyek) dan juga kepentingan analisis data (pengguna). Oleh sebab itu, dalam *user management* nanti akan dibuat suatu *group* terhadap pengguna yang memiliki kepentingan terhadap sistem tersebut. Data grup tersebut merupakan data statis yang terdiri atas: Admin yang akan mengatur hak akses pengguna,

Operator Data Proyek yang melakukan entri data struktur proyek, Operator Data Keuangan yang melakukan entri data keuangan proyek-proyek yang telah didefinisikan dalam struktur proyek, dan *User Analyst* yang mempunyai kepentingan untuk melakukan analisis terhadap data-data yang telah diinputkan.

Dengan adanya bermacam-macam posisi pengguna dari bermacam-macam departemen yang terlibat dalam sistem ini, maka *user management* juga akan mengkategorikan pengguna berkaitan dengan adanya beberapa (lebih dari satu) pengguna yang berkepentingan terhadap entri data, adanya beberapa pengguna yang hanya dapat menganalisa data yang telah ada tanpa berhak merubah data tersebut, ataupun pengguna yang akan memegang kendali *user management*.

Karena fungsi dari adanya *user management* ini adalah untuk mengatur hak akses pengguna, maka *user management* tentu saja akan menjadi media penyimpanan data-data yang berhubungan dengan kewenangan seorang pengguna terhadap sistem (data *user*). Data-data ini berkaitan dengan login, password dan juga posisi pengguna tersebut dalam sebuah departemen dan group *user*.

4.1.2 Analisis Kebutuhan Entri Data terhadap Sistem yang Akan Dibuat

Untuk melakukan sebuah analisis pada aspek keuangan, data-data yang dibutuhkan menyangkut informasi tentang Data Struktur Proyek dan Data-data Keuangan.

4.1.2.1 Kebutuhan Data Struktur Proyek

Kebutuhan data struktur proyek berkaitan dengan data-data tentang Sektor, Subsektor, Program, dan Proyek dalam sebuah hirarki, dan Data Kelas Proyek. Data struktur proyek dan kelas proyek ini diperoleh dari Dinas-dinas atau Instansi terkait dengan potensi-potensi masing-masing daerah. .

4.1.2.2 Kebutuhan Data Keuangan Proyek

Pada sistem ini, proses utama yang akan dibuat adalah sebuah proses yang akan memberikan fasilitas terhadap proses analisis keuangan. Proses analisis ini dapat berjalan dengan tersedianya data-data keuangan. Oleh sebab itu diperlukan sejumlah inputan (entri data) berupa data-data keuangan.

Data keuangan yang dibutuhkan adalah data *Balance Sheet, Income Statement, Benefit Cost*, dan *Cash Flow* yang diperoleh sebagai laporan keuangan dari bagian keuangan untuk kurun waktu tertentu. Kemudian dari data-data tersebut, akan dihasilkan nilai-nilai BCR, IRR, PBP, ROI, dan ROA yang merupakan hasil perhitungan yang dilakukan dengan memberikan formulasi-formulasi terhadap *Balance Sheet, Income Statement, Benefit Cost,* dan *Cash Flow.*

Data-data keuangan tersebut diatas, secara keseluruhan belum dapat disediakan oleh sistem yang ada di Bappeda, sehingga dalam pengerjaan tugas akhir ini, penulis akan menggunakan data-data laporan keuangan yang terdapat pada literatur yang berkaitan dalam melakukan ujicoba.

Dengan adanya keperluan data keuangan BCR, IRR, PBP, ROI, dan ROA, maka dalam proses pengambilan data akan digunakan sejumlah *view* untuk melakukan operasi *cross-join* terhadap tabel-tabel yang ada.

4.1.3 Analisis Fitur OLAP Report

Sistem aplikasi OLAP yang akan dibuat seharusnya memiliki kemampuan untuk:

- Memberikan respon yang cepat terhadap sebuah proses analisis data. Kemampuan respon yang cepat ini hanya akan didapatkan jika sistem tersebut merupakan sistem yang *online.*

- Melakukan sejumlah operasi-operasi kalkulasi yang mendukung proses analisis terhadap sejumlah data-data yang telah di-entri-kan.
- Menyediakan fungsi-fungsi standar navigasi yang dapat mempercepat pengguna untuk mengolah data-data yang dibutuhkan pada proses analisis, sehingga sistem ini akan memberikan fleksibilitas pada proses analisis data.
- Menyimpan dokumen-dokumen analisis yang telah dibuat oleh masing-masing pengguna, sehingga untuk proses-proses analisis selanjutnya yang sejenis, proses analisis ini akan lebih cepat dilakukan karena tidak perlu lagi didefinisikan dari awal. Ditambah lagi dengan adanya kemampuan untuk membedakan analisis-analisis yang sering dilakukan (*favorite analysis*).
- Pembuatan dokumen analisis dalam berbagai bentuk, model dan konfigurasi akan memberikan visualisasi yang berbeda terhadap hasil rekapitulasi data. Sebuah proses *cutomization* terhadap tampilan analisis yang telah dibuat akan sangat membantu pengguna dalam melihat data pada perspektif yang bermacam-macam sehingga akan memudahkan proses analisis.
- Memberikan visualisasi data dalam berbagai bentuk, sehingga akan memudahkan pengguna —yang walaupun ada pada level yang sama namun memiliki keahlian yang berbeda-beda— tetap dapat secara fleksibel melakukan analisis terhadap data-data yang komplek.

4.2 Analisis Sarana Pendukung

Untuk mendukung aplikasi OLAP penunjang analisis kelayakan proyek ini, maka dibutuhkan spesifikasi minimum untuk sistem yang telah ada, baik software maupun hardware. Adapun aplikasi OLAP untuk analisis kelayakan proyek ini dapat dijalankan dengan:

Spesifikasi minimum system

- Windows 9X / NT 4.0 / 2000 / XP
- Microsoft Excel 9X/ 2000

Spesifikasi minimum hardware

- Pentium II
- RAM 128 MB
- Resolusi monitor 800 x 600 (disarankan)
- 100 MB free space disk.

Spesifikasi minimal dari sistem tersebut harus dipenuhi dikarenakan adanya integrasi sistem OLAP yang akan dibuat dengan Microsoft Excel. Microsoft telah merekomendasikan spesifikasi minimum system untuk Microsoft Excel-nya. Spesifikasi minimal dari hardware tersebut sangat dianjurkan , karena dalam aplikasi OLAP yang akan dibuat akan banyak melakukan perhitungan-perhitungan matematis.

Kebutuhan sarana pendukung tersebut sangat diperlukan untuk menunjang kecepatan aplikasi OLAP ini dalam memproses data tergantung dari besarnya database yang digunakan. Oleh karena itu jika database aplikasi berukuran besar, maka lebih baik jika spesifikasi hardware diatas dipenuhi.

BAB V

PERANCANGAN SISTEM

BAB V
PERANCANGAN SISTEM

Setelah tahap analisis telah selesai, maka dari hasil analisis tersebut, akan dibuat suatu sistem yang nantinya merupakan pemecahan dari setiap permasalahan yang telah didefinisikan. Bab V akan memberikan desain dari sistem yang akan dibuat penulis dalam tugas akhir ini dan tahap desain ini merupakan tahapan kedua dalam metodologi *SDLC*.

5.1 Desain Proses

Desain proses menggambarkan alur proses yang terjadi dari mulai adanya sekumpulan data yang diberikan sebagai input pada sistem (data-data pendukung) sampai mencapai sebuah output yang merupakan implementasi sebuah OLAP. Desain Proses ini akan digambarkan dengan menggunakan DAD (Diagram Arus Data) menggunakan *modelling tool Power Designer 6.0* dengan metode *Gane and Sarson*. Penggambaran DAD ini dianggap sudah rinci menjelaskan sistem OLAP yang akan dibuat.

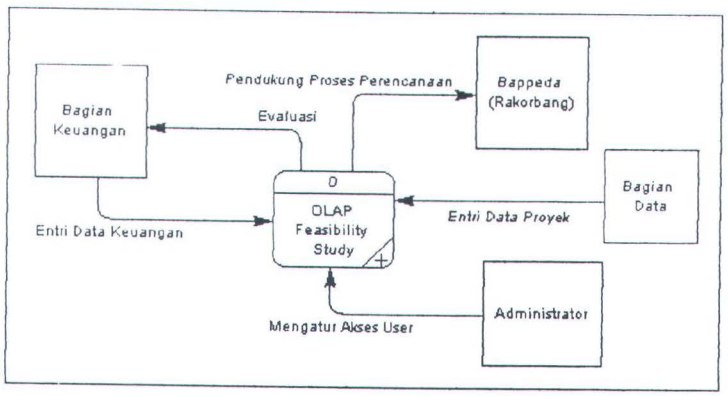
5.3.1 DAD Level 0

DAD Level 0 merupakan level teratas (*Top Level*) yang mewakili penggambaran proses dari keseluruhan sistem ***OLAP Feasibility Study*** yang akan dibuat pada pengerjaan tugas akhir ini.

Entitas *group user* yang terlibat dalam aliran data diatas adalah:

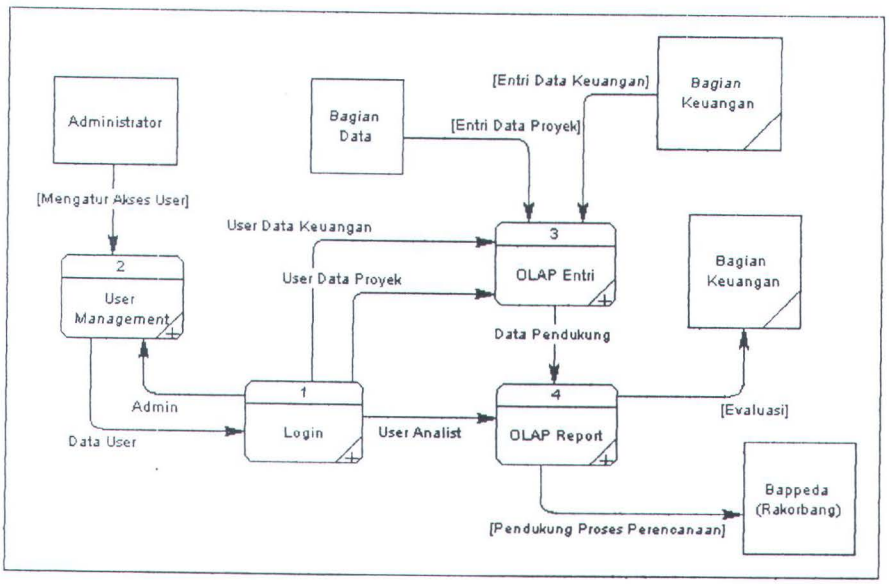
- Bagian Data, melakukan entri data-data proyek berkaitan dengan struktur dan kelas proyek.

- Bagian Keuangan, melakukan entri data-data keuangan proyek dari daftar proyek yang telah didefinisikan oleh bagian data. Bagian keuangan juga akan dapat melakukan evaluasi keuangan dari sistem analisis kelayakan proyek ini.
- Bappeda, sebagai perencana proyek yang akan menggunakan sistem analisis kelayakan proyek.
- Administrator, mengatur hak akses *user-user* yang terlibat pada sistem.



Gambar 5.1 DAD Level 0

5.3.2 DAD Level 1



Gambar 5.2 DAD Level 1

DAD Level 1 pada gambar 5.2 merupakan gambaran proses pada sistem yang akan dikerjakan pada tugas akhir berkaitan dengan studi kasus Analisis Studi Kelayakan dalam aspek keuangan. Secara garis besar, sistem akan terdiri dari 3 proses utama (modul) yaitu *User Management*, OLAP Entri dan OLAP Report. Proses Login akan dipergunakan oleh sistem untuk validasi *user* terhadap hak akses yang dimilikinya.

Proses-proses yang yang ada pada penggambaran DAD level 1 tersebut, dijelaskan sebagai berikut:

- Login, proses ini merupakan syarat utama yang dilakukan *user* untuk dapat mengakses proses-proses yang lain. Proses ini memberikan validasi kepada pengguna sesuai dengan hak akses yang diberikan pada *User Management*.
- *User Management*, digunakan oleh *Administrator* untuk mengatur hak akses *user* sehingga setiap *user* berdasarkan group *user* akan memiliki wewenang yang berbeda terhadap sistem. Misalnya, *user-user* yang tergabung dalam *user group Analist* tidak akan dapat melakukan entri data yang ada pada OLAP Entri, dan sebaliknya.
- OLAP Entri, merupakan proses entri data-data pendukung untuk membuat sebuah aplikasi OLAP. Entri data ini akan dilakukan oleh bagian data dan bagian keuangan.
- OLAP Report, proses analisis yang merupakan implementasi dari sebuah aplikasi OLAP yang akan melakukan pengolahan data-data dari input yang telah diberikan sistem melalui OLAP Entri untuk mendapatkan berbagai informasi yang diperlukan. OLAP Report ini akan membantu proses

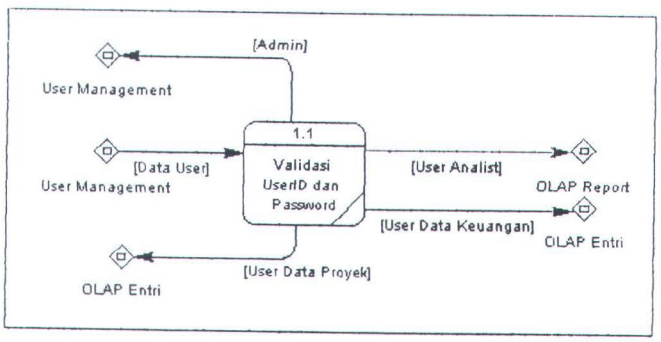
perencanan yang dilakukan Bappeda dalam Rakorbang dan juga digunakan Bagian Keuangan untuk melakukan evaluasi terhadap proyek.

5.3.3 DAD Level 2

Berikut akan lebih didetilkan penggambaran proses-proses yang ada di DAD level 1 kedalam DAD level 2 untuk masing-masing proses.

5.3.3.1 DAD Level 2 *Login*

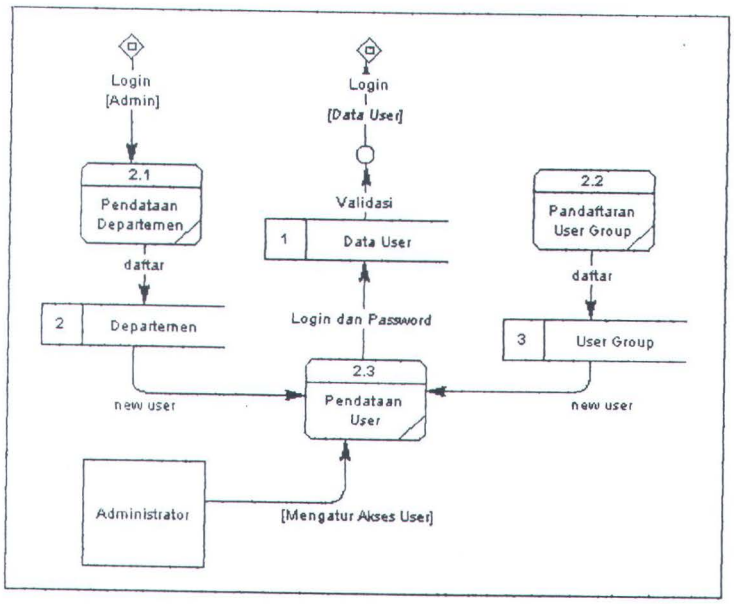
Dari data *user* yang terdapat pada *User Management*, maka proses Validasi kombinasi UserID dan Password akan menentukan kelompok *user group* bersangkutan. Jika dalam proses validasi dengan data *user* tersebut menyatakan kedudukan seorang pengguna dalam sebuah kelompok *user group* adalah sebagai Admin, maka pengguna ini akan memiliki akses terhadap modul *User Management*, jika dalam kelompok *user group* User Data Proyek memiliki akses terhadap modul OLAP Entri (untuk entri-entri data proyek), *user group* User Data Keuangan terhadap modul OLAP Entri (untuk entri-entri data keuangan), dan jika *user group* User Analist memiliki akses terhadap modul OLAP Report. Setiap pengguna hanya dapat tergolong dalam sebuah *user group*.



Gambar 5.3 DAD Level 2 Login

5.3.3.2 DAD Level 2 *User Management*

DAD Level 2 *User Management* ini menggambarkan proses pendaftaran *user* beserta pendefinisian group dan departemen untuk menentukan hak akses yang akan dimiliki oleh seorang pengguna.



Gambar 5.4 DAD Level 2 User Management

Dari Gambar 5.4, maka proses yang terjadi pada *user management* meliputi proses-proses:

- Pendataan Departemen, mendefinisikan departemen-departemen yang ada terlibat dalam sistem OLAP ini.
- Pendataan *User Group*, mendefinisikan grup-grup user yang berkaitan dengan sistem. Grup user ini telah didefinisikan menjadi 4, yaitu grup Administrator, Data Proyek, Data Keuangan dan Analisis. Data hasil proses ini merupakan data statis, karena hal ini berkaitan dengan modul-modul yang ada dalam sistem yang statis juga.

- Proses Pendataan User, merupakan proses mendaftarkan pengguna pada sistem, mendefinisikan login dan default password, melakukan kategorisasi departemen dan group *user* yang dimiliki pengguna tersebut. Karena proses ini membutuhkan keterangan tentang Departemen dan *Group User*, maka proses ini sebaiknya dilakukan setelah data-data pada proses Pendataan Departemen dan Pendataan *Group User* lengkap. Keseluruhan proses pada DAD level 2 *User Management* ini dilakukan oleh *Group User Administrator.*

5.3.3.3 DAD Level 2 OLAP Entri

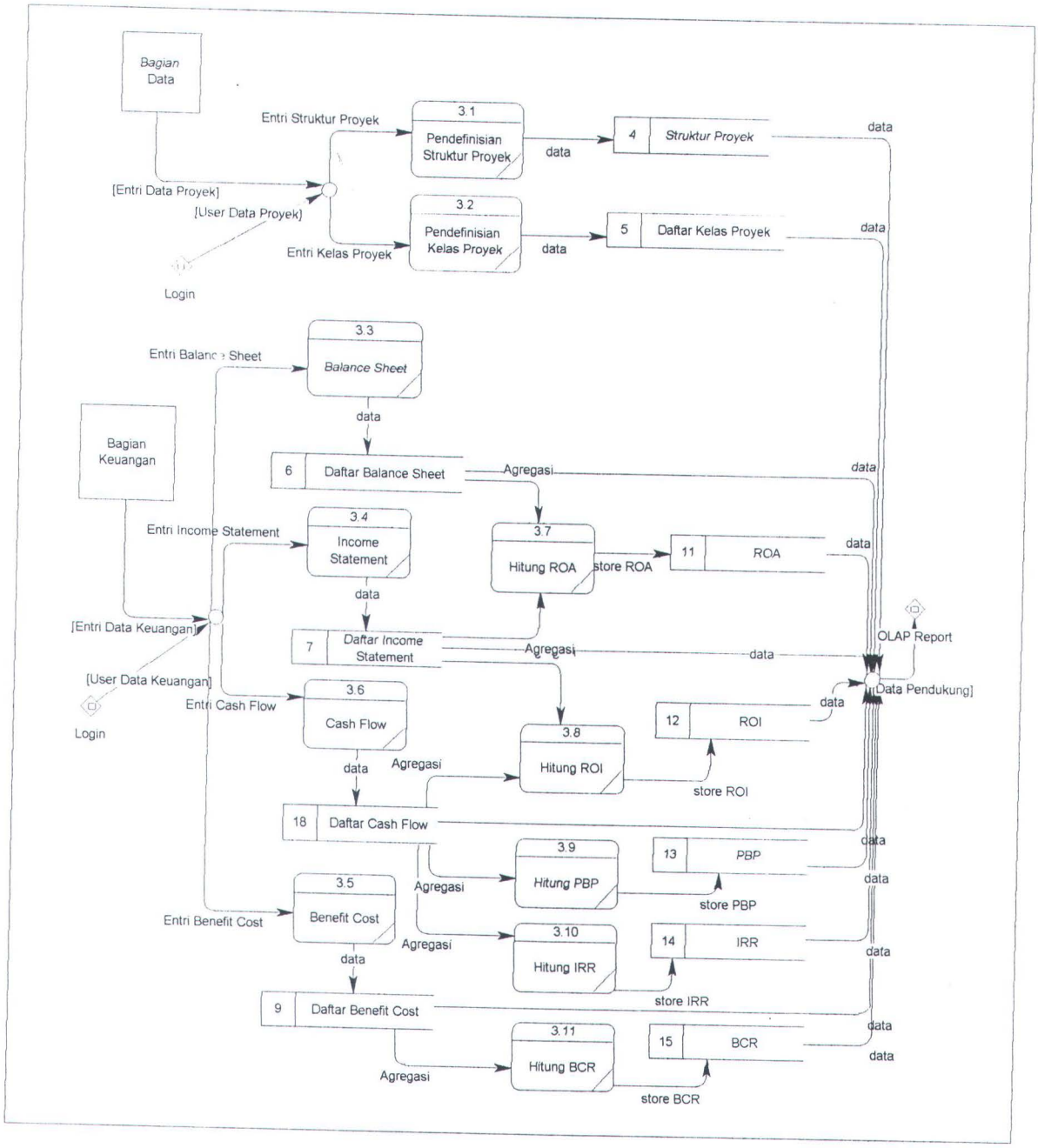
DAD level 2 OLAP Entri menggambarkan proses entri (input) data-data pendukung yang nantinya akan mendukung sistem OLAP yang akan dibuat. Data-data pendukung berasal dari bagian keuangan dan bagian data.

Dari Gambar 5.5 , maka proses yang terjadi pada OLAP Entri dibedakan berdasarkan jenis data yang diinputkan, terdiri atas:

- Bagian data, melakukan proses entri data struktur proyek beserta kelas proyek.
- Bagian keuangan, melakukan proses-proses entri data keuangan terhadap data struktur proyek yang telah didefiniskan. Proses-proses entri data keuangan ini meliputi entri data Balance Sheet, Income Statement, Cash Flow dan Benefit Cost. Kemudian, dari hasil entri data-data tersebut, maka sistem akan secara otomatis melakukan proses agregasi data, sehingga akan menghasilkan data-data keuangan baru yaitu data ROA, ROI, PBP, IRR dan BCR.

Proses entri data-data tersebut, baik data struktur proyek maupun data keuangan proyek merupakan proses yang menjadi syarat utama dari sistem OLAP

pada tugas akhir ini, karena data-data itulah yang nantinya akan diolah dalam pengimplementasian sebuah aplikasi OLAP.

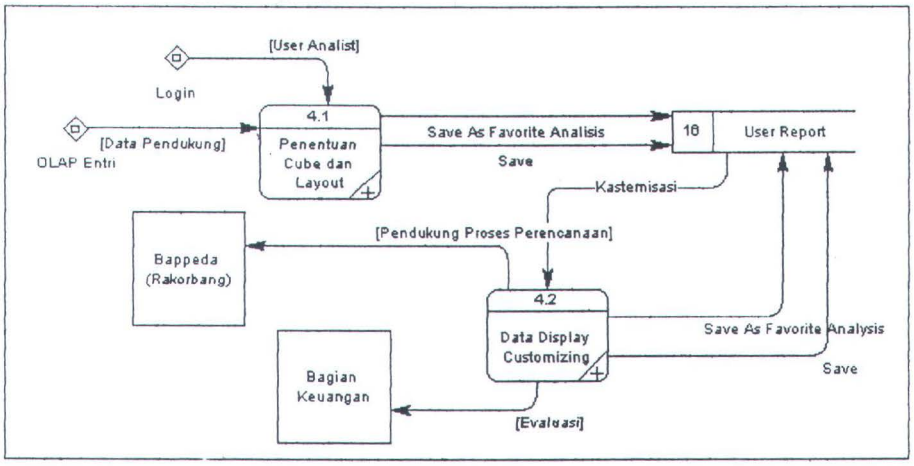


Gambar 5.5 DAD Level 2 OLAP Entri

5.3.3.4 DAD Level 2 OLAP Report

- DAD level 2 OLAP Report menggambarkan proses pembuatan dokumen OLAP Report yang hasilnya akan merupakan sebuah aplikasi OLAP yang

sesungguhnya, yang akan yang terdiri atas 2 (dua) proses utama yaitu Penentuan *Cube* dan *Layout* serta proses Data *Display Customizing*, yang digambarkan seperti Gambar 5.6.



Gambar 5.6 DAD Level 2 OLAP Report

Proses yang terjadi pada OLAP Report secara garis besar terdiri atas:

- Penentuan *Cube* dan *Layout*, merupakan proses pertama yang harus dilakukan pengguna terhadap sistem. Proses ini memerlukan data-data yang telah diinputkan pada proses sebelumnya yaitu pada OLAP Entri. Setelah melakukan aktivitas-aktivitas yang ada dalam proses ini, maka dokumen analisis yang telah dibuat *user*, dapat langsung disimpan (*save*) ataupun disimpan sebagai *favorite analysis* (*Save As Favorite Analysis*).

Data *Display Customizing*, proses *customization* terhadap hasil dokumen report yang telah disimpan, untuk semakin merubah penampakan dari data-data yang ada. Kemudian, setelah proses tersebut, analisis ini dapat disimpan lagi dengan nama yang berbeda, karena proses penyimpanan yang dilakukan oleh sistem dibuat secara incremental, hal ini akan berguna untuk tetap menjaga keberadaan dokumen analisis lama, untuk keperluan perbandingan dengan dokumen yang baru dibuat.

5.3.4 DAD Level 3

Penggambaran DAD level 3 sebagai penjelas dari masing-masing proses yang ada di level 2 yang masih bisa didetilkan.

5.3.4.1 DAD Level 3 Penentuan *Cube* dan *Layout*

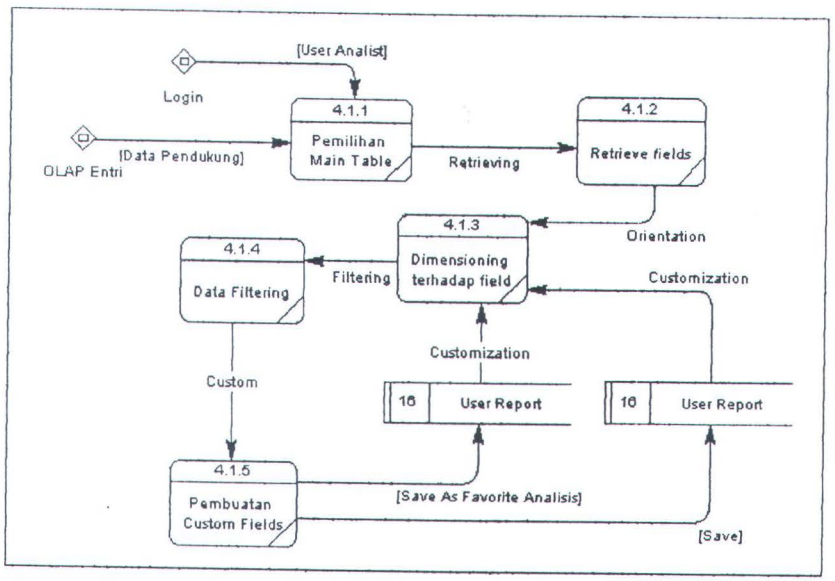
DAD level 3 Penentuan *Cube* dan *Layout* menggambarkan proses pemilihan terhadap field-field pada sebuah tabel utama (*main table*) untuk kemudian ditentukan orientasi field tersebut dalam penampakan (layout) pada sebuah dokumen analisis. DAD ini juga akan menggambarkan proses *filtering* data dan pembuatan *custom field* yang masing-masing akan secara optional didefinisikan *user*.

Dari Gambar 5.7 , proses-proses yang ada pada DAD level 3 Penentuan *Cube* dan *Layout* lebih detil dijelaskan sebagai berikut:

- Pemilihan Main Table, memilih sumber data yang akan dianalisa.
- Retrieve Fields, pengambilan semua field yang ada pada sumber data.
- *Dimensioning* terhadap Field, menentukan orientasi field pada dokumen report yang akan dibuat. Orientasi field dapat berupa *Page*, *Row*, *Column*, atau *Data*. Sebuah field yang telah di-*retrieve* hanya dapat berada sebuah orientasi saja. Pada orientasi *Data* hanya dapat diisi oleh field-field yang type datanya numerik.
- *Data Filtering*, melakukan filter terhadap data-data yang telah dipilih. Dalam implementasi akan dipergunakan semacam *SQL string builder* yang akan memudahkan pengguna untuk melakukan *filtering* data dalam bentuk visual.
- Pembuatan *Custom Field*, pembuatan field-field lain oleh pengguna untuk menambah informasi terhadap data-data yang ada. Pembuatan custom field ini dapat dilakukan dengan operasi-operasi terhadap field-field numerik sebagai

operand-nya. Untuk mempermudah dalam pembuatan custom field ini, maka dalam implementasi sistem ini akan dilengkapi sebuah *Formula Editor* yang mirip sebuah kalkulator.

Serangkaian proses-proses tersebut akan menghasilkan sebuah *analysis document* yang dapat disimpan sebagai *analysis document* biasa (*save*) atau dapat juga disimpan sebagai sebuah analisis favorit (*save as favorite analysis*). Namun, dengan hanya melalui tahapan proses Pemilihan *Main Table*, *Retrieve Fields* dan *Dimensioning* terhadap *Field*, maka dokumen report dapat langsung dilihat dan dapat langsung disimpan. Hasil dokumen analysis yang telah disimpan ini dapat pula dikastemisasi terhadap orientasi field-field yang ada, dengan mereferensi *main table* yang sama.



Gambar 5.7 DAD Level 3 Penentuan Cube dan Layout

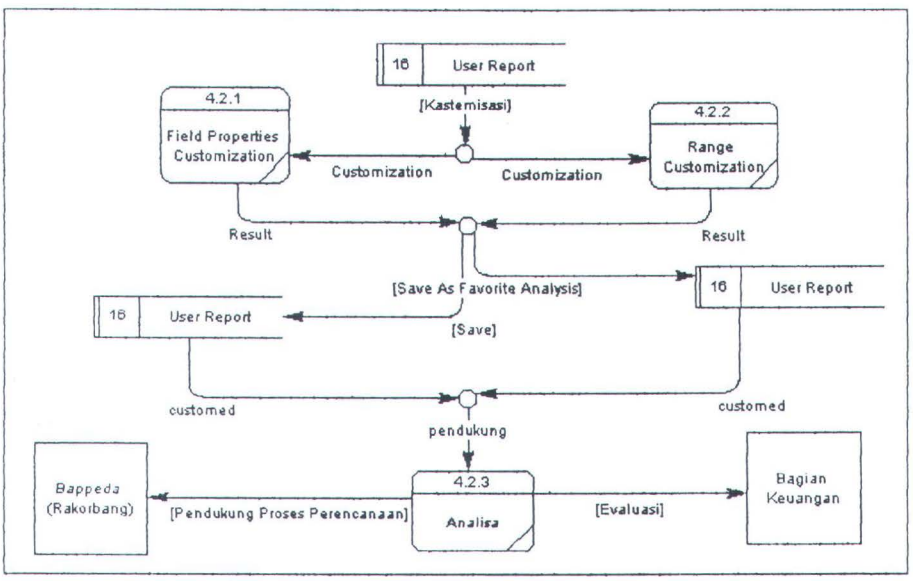
5.3.4.2 DAD Level 3 *Data Display Customizing*

DAD level 3 *Data Display Customizing* ini menggambarkan proses *customization* terhadap tampilan dari dokumen analisis yang telah dibuat/disimpan.

Kastemisasi ini berupa proses kastemisasi terhadap field yang ada pada dokumen maupun proses pewarnaan terhadap sebuah range data. Hasil proses ini akan menghasilkan sebuah dokumen analisis yang sesuai dengan perubahan konfigurasi yang dilakukan pengguna.

Proses-proses yang terlihat pada Gambar 5.8 dijelaskan dengan lebih rinci sebagai berikut:

- *Field Properties Customization*, kastemisasi terhadap field-field yang terlihat pada dokumen analisis. Proses kastemisasi ini dilakukan pada field *caption*, pemilihan tipe *group footer*, *rank style*, dan *rank on* untuk data-data yang memiliki orientasi selain *Data*, serta agregasi data, tipe data *display* dan penulisan format data untuk data-data yang berorientasi sebagai *Data*.
- *Range Customization*, merupakan proses kastemisasi terhadap sebuah interval data. Kastemisasi ini berupa pewarnaan (pemilihan warna) terhadap sebuah data terhadap beberapa interval yang didefinisikan pengguna.



Gambar 5.8 DAD Level 3 Data Display Customizing

MILIK PERPUSTAKAAN ITS

5.2 Desain Data

Desain data ini akan menggambarkan struktur database pada *user management*, OLAP Entri, dan OLAP Report. Proses Desain data akan dibuat dengan menggunakan *data modelling tool* **ERWin 4.0** berupa desain *Entity Relationship Diagram* (ERD).

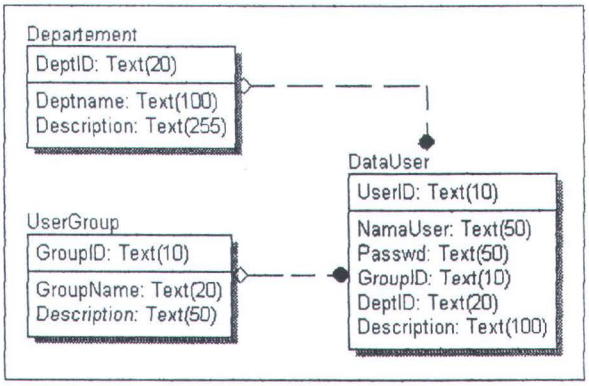
5.2.1 Desain Data *User Management*

Desain data *User Management* menggambarkan struktur tabel dan keterhubungan tabel-tabel yang berisi data-data tentang UserGroup, Departemen, dan DataUser. Tabel UserGroup secara detil menyimpan data-data tentang identitas group, nama group user dan keterangan. Pada tabel ini, identitas group (GroupID) dipakai sebagai *primary key*. *Primary Key* dari tabel ini direferensi sebagai sebuah *foreign key* pada tabel DataUser, informasi ini menunjukkan posisi pengguna terhadap sebuah group. Seorang pengguna hanya dapat berada pada sebuah group.

Tabel Departemen akan menyimpan data tentang departemen-departemen yang memiliki keterkaitan dengan sistem perencanaan di Bappeda. Pada tabel ini, identitas departemen (DeptID) dipakai sebagai *primary key*, yang direferensi sebagai sebuah *foreign key* pada tabel DataUser. Informasi ini menunjukkan posisi *user* terhadap sebuah departemen. Seorang *user* hanya berada pada sebuah departemen.

Tabel DataUser menyimpan data tentang pengguna-pengguna yang memiliki kepentingan terhadap sistem. Data-data ini berupa UserID yang merupakan *primary key* dari tabel DataUser, serta NamaUser, password (Passwd), serta posisi user tersebut terhadap sebuah group (GroupID) dan dalam sebuah departemen (DeptID).

Keterhubungan ketiga tabel tersebut dalam sebuah *user management* akan terlihat dalam sebuah Diagram ER seperti Gambar 5.9.



Gambar 5.9 ERD User Management

5.2.2 Desain Data OLAP Entri

Desain data OLAP Entri menggambarkan struktur tabel dan keterhubungan tabel-tabel yang berisi data-data struktur proyek dan data-data keuangan proyek. Desain basisdata pada OLAP Entri ini akan memakai konsep HOLAP, dimana data keuangan proyek akan disimpan dalam bentuk yang telah teragregasi.

Seperti telah dijelaskan pada bagian analisis (4.1.2), maka OLAP Entri ini akan menjembatani proses manual yang ada berkaitan dengan daftar usulan proyek yang diajukan dengan aplikasi OLAP Report sebagai tool yang membantu proses analisis dalam sebuah perencanaan.

Data-data struktur proyek yang harus diinputkan meliputi data-data pada TbSektor, TbSubSektor, TbProgram, dan TbProyek dususun berdasarkan sebuah hirarki. Sedangkan LtKelasProyek digunakan sebagai *look-up table* yang akan dipakai sebagai referensi untuk menentukan sebuah kelas proyek untuk setiap proyek yang didefinisikan.

Data-data keuangan proyek yang dipakai meliputi data-data inputan pada TbBalanceSheet, TbIncomeStatement, TbCashFlow dan TbBenefitCost. Namun,

untuk melakukan analisis terhadap sebuah proyek pada TbProyek, maka dibutuhkan pula data-data keuangan yang didefinisikan pada TbROA, TbROI, TbBCR, TbPBP dan TbIRR, yang mana data-data pada kelima tabel ini bukan merupakan inputan dari pengguna, melainkan hasil agregasi yang dilakukan oleh sistem yang akan dibuat.

Keterhubungan antara tabel-tabel dalam desain basisdata ini adalah sebuah struktur basisdata *snowflake*, karena dimensinya disusun dengan model hirarki TbSektor, TbSubSektor, TbProgram, dan TbProyek, terhadap *fact table* data-data keuangan. Desain data snowflake ditunjukkan seperti Gambar 5.10.

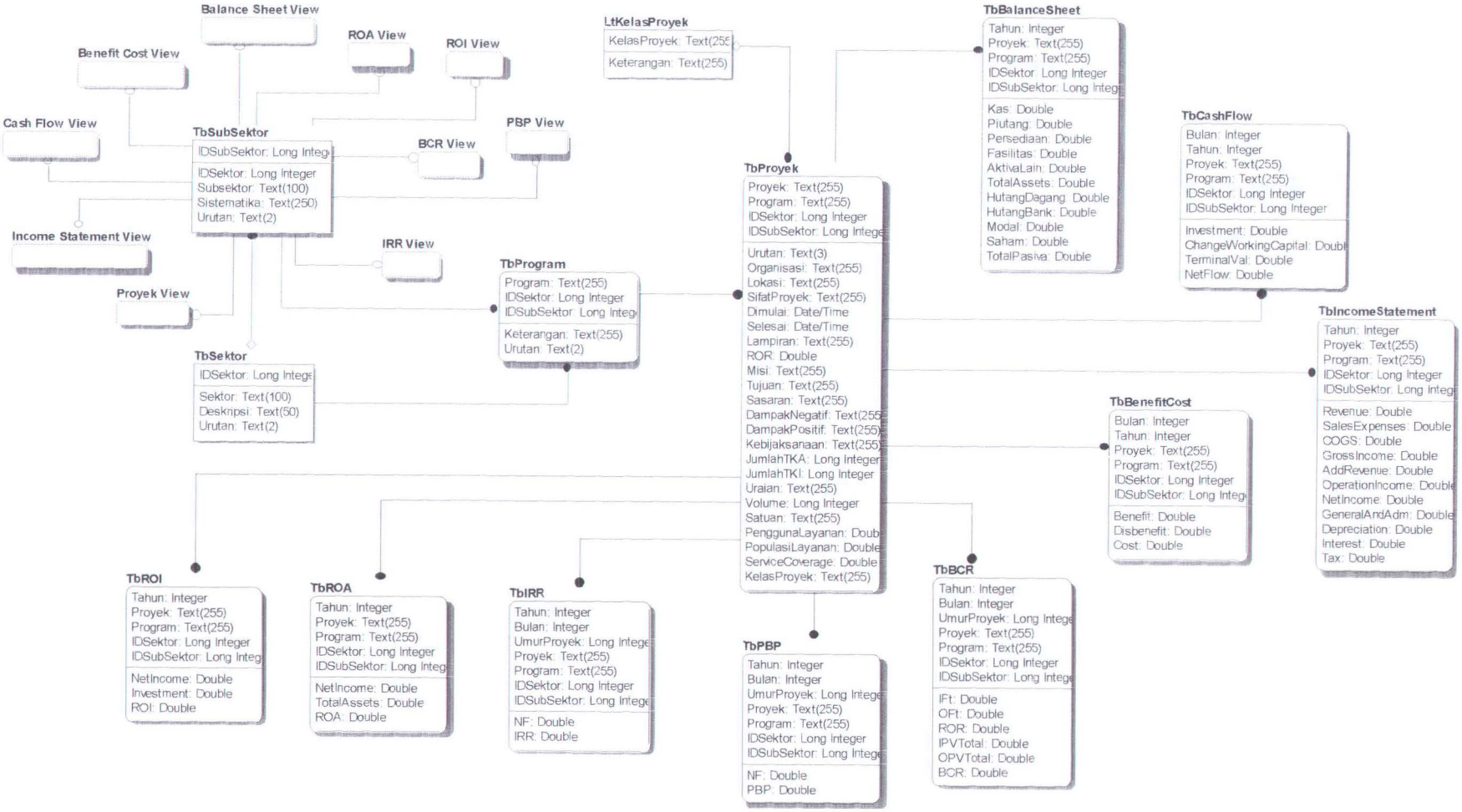
5.2.3 Desain Data OLAP Report

Desain data untuk OLAP Report terdiri atas dua bagian, yaitu desain data yang menggambarkan penyimpanan struktur database dari semua tabel (digunakan dalam OLAP Entri) serta penyimpanan terhadap display analysis report yang telah dibuat oleh seorang pengguna.

5.2.3.1 Desain Data Struktur Database

Desain data struktur database ini akan menangani masalah penyimpanan struktur database dari tabel yang digunakan dalam analisis yang terdiri atas LtTables dan LtColums ditunjukkan pada Gambar 5.11.

LtTables digunakan untuk menyimpan Tabel dan LtColumns digunakan untuk menyimpan Kolom yang dipakai untuk proses analisis pada OLAPReport.

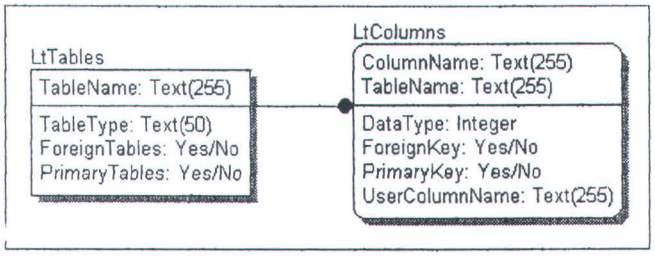


Gambar 5.1 ERD OLAP Entri

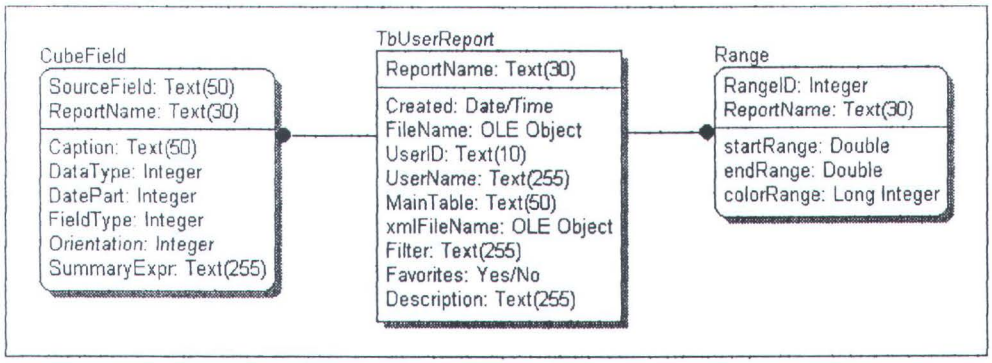
5.2.3.2 Desain Data Penyimpanan Display Analisis

Desain penyimpanan display analisis ini digunakan untuk menyimpan layout serta display dari analisis yang dilakukan pada OLAPReport. Terdiri atas TBUserReports, CubeField, dan Range ditunjukkan pada Gambar 5.12.

CubeField menyimpan informasi field yang menjadi dimensi pada proses analisis. Range menyimpan pewarnaan terhadap nilai-nilai data. TbUserReport menyimpan file analisis yang telah dilakukan pengguna.



Gambar 5.11 ERD Struktur Database OLAP Report



Gambar 5.12 ERD Penyimpanan Display Analisis OLAP Report

5.2.3.3 Desain *Query Data Cube*

Dalam sebuah proses analisis, tentu saja diperlukan proses pengambilan data. Karena struktur target sumber database-nya dalam bentuk *snowflake*, maka

diperlukan *join query*. Beberapa *query* (*view*) dapat dibuat berdasarkan *fact table* yang terdapat di database, antara lain:

- ***Proyek View***, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, dan TbProyek.
- ***BalanceSheet View***, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbBalanceSheet.
- ***IncomeStatement*** *View*, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbIncomeStatement.
- ***CashFlow*** *View*, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbCashFlow.
- ***BenefitCost*** *View*, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbBenefitCost.
- ***ROA*** *View*, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbROA.
- ***ROI View,*** merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbROI.
- ***IRR View***, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbIRR.
- ***BCR View***, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbBCR.
- ***PBP View***, merupakan *join query* antara tabel TbSektor, TbSubSektor, TbProgram, TbProyek, dan TbPBP.

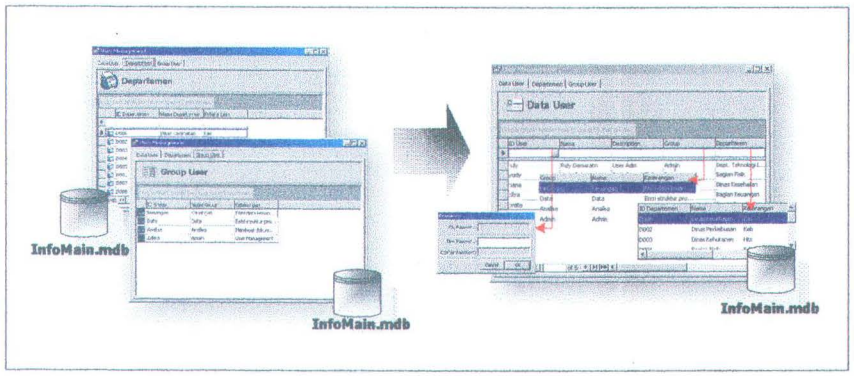
5.3 Desain Antarmuka

Desain antarmuka akan menggambarkan form-form yang mendukung pembuatan sistem OLAP beserta dengan hak akses terhadap pengguna yang terkait.

Terdapatnya 4 (empat) proses utama yang menjadi pembahasan dalam pembuatan sistem OLAP dalam pengerjaan tugas akhir ini. Keempat proses utama tersebut secara keseluruhan akan dibuat dalam 3 (tiga)modul yaitu Modul User Management, Modul OLAP Entri dan Modul OLAP Report. Oleh sebab itu, maka setiap modul akan memiliki desain antarmuka yang saling terpisah

5.3.1 Desain Antarmuka *User Management*

Desain antarmuka User Management (Gambar 5.13) akan terdiri dari 2 (dua) buah form sebagai media seorang Administrator untuk mendata pengguna-pengguna yang nantinya memiliki hak akses terhadap sistem OLAP secara keseluruhan baik terhadap User Management, OLAP Entri maupun OLAP Report.

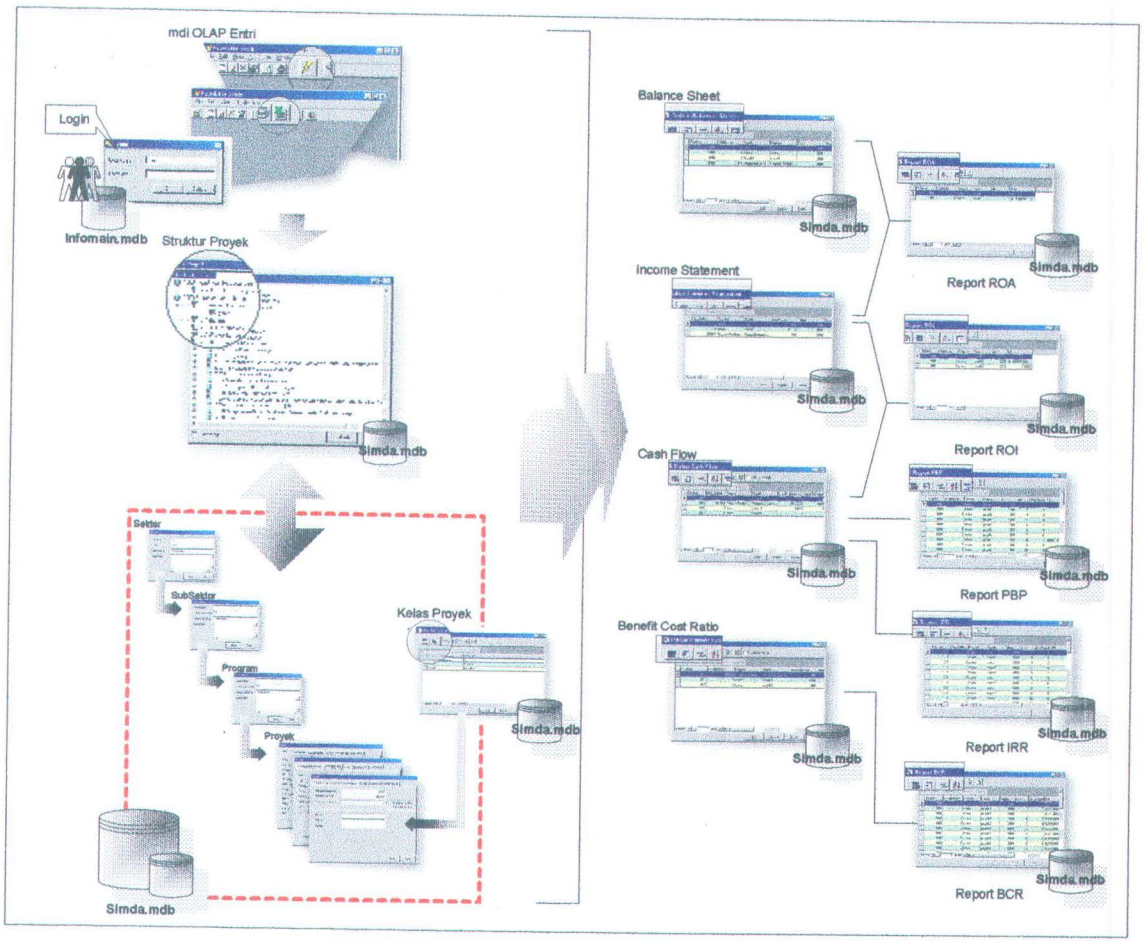


Gambar 5.13 Desain Antarmuka User Management

5.3.2 Desain Antarmuka OLAP Entri

Antarmuka OLAP Entri nantinya akan menjadi sebuah media yang akan menghubungkan sistem luar yang berbentuk manual dengan sistem OLAP yang akan dibuat. OLAP Entri akan berisi form-form yang akan menjadi media data input.

OLAP Entri dilengkapi pula dengan form-form yang berisi laporan-laporan dari data-data yang telah diinputkan, yang mana form-form ini hanya bersifat *Read-Only* seperti digambarkan pada gambar 5.14.

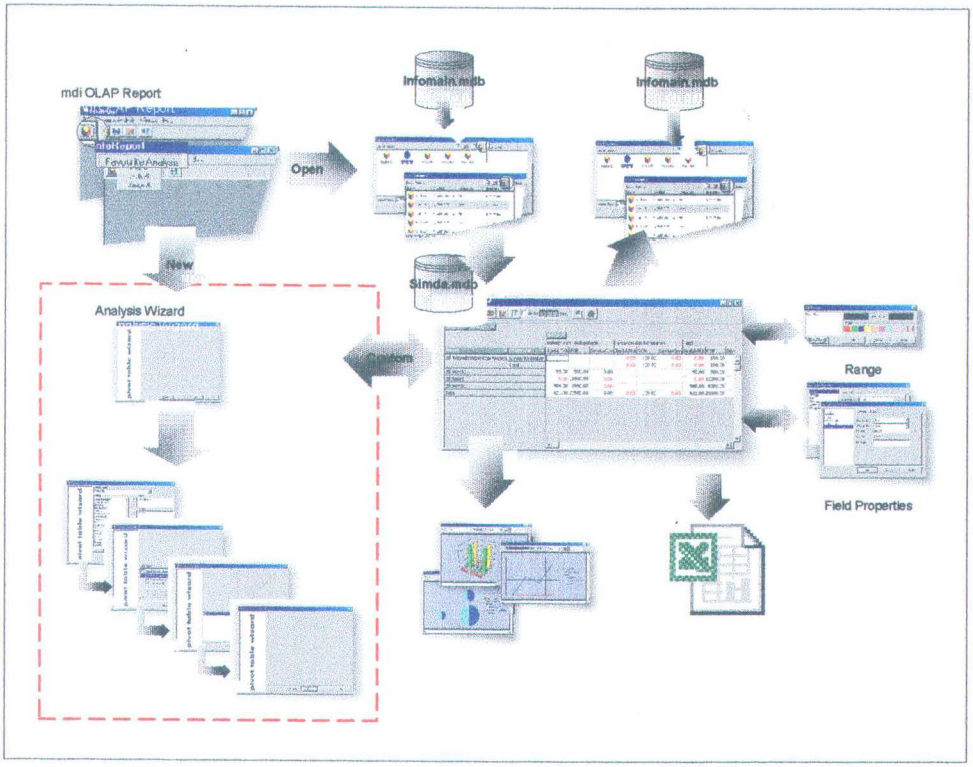


Gambar 5.14 Desain Antarmuka OLAP Entri

5.3.3 Desain Antarmuka OLAP Report

OLAP Report nantinya akan menjadi *analysis tool* yang dipergunakan Bappeda untuk mendukung proses analisis kelayakan proyek yang menjadi studi kasus pengerjaan tugas akhir ini. Oleh sebab itu, MDI OLAP Report akan berisi form-form yang akan membantu pengguna untuk melakukan pemilihan Data Cube yang dipakai dalam analisis dan menyesuaikan tampilan dokuman analisis..

Secara garis besar, penggambaran antarmuka untuk modul OLAP Report akan memiliki hubungan antar form-form sebagai berikut pada gambar 5.15.



Gambar 5.15 Desain Antarmuka OLAP Report

5.4 Desain Algoritma Perhitungan Indikator Keuangan

Setelah diketahui bahwa analisis keuangan proyek membutuhkan indikator-indikator keuangan yang sebagian data-datanya didapatkan dari agregasi nilai data-data yang diinputkan, maka berikut ini akan diberikan algoritma dari perhitungan-perhitungan tersebut, yang nantinya dalam proses implementasi akan menjadi function. Desain Algoritma ini akan diberikan sebagai lampiran (Lampiran A) pada akhir bagian pada buku Tugas Akhir ini.

BAB VI

PEMBUATAN APLIKASI SISTEM *FEASIBILITY STUDY* BERBASIS OLAP

BAB VI
PEMBUATAN APLIKASI SISTEM *FEASIBILITY STUDY* BERBASIS OLAP

Tahap selanjutnya dalam *SDLC* setelah tahap desain adalah tahap *development*, dimana pada tahap ini akan dilakukan pengkodean untuk membuat aplikasi yang sesuai dengan desain yang telah dibuat. Bab VI secara garis besar akan menjelaskan tahapan-tahapan yang dilakukan untuk melakukan konstruksi modul-modul (aplikasi-aplikasi) yang ada pada sistem *OLAP Feasibility Study* ini. Konstruksi aplikasi menggunakan ***Visual Basic 6.0*** dengan pemakaian beberapa komponen-komponen visual dan menggunakan database Access.

6.1 Implementasi Database

Jika pada subbab 5.2 tentang Desain Data yang telah dijelaskan sebelumnya, desain data dibuat untuk 3 (tiga) modul yang ada, maka dalam implementasi database hanya akan dibuat 2 database, yang mana database pertama akan mengatur struktur data *User Management* dan OLAP Report sedangkan database kedua akan mengatur struktur data OLAP Entri. Implementasi database dalam tugas akhir ini dipergunakan Access, dengan pertimbangan data-data yang menjadi target sistem merupakan data-data yang frekuensi aksesnya tidak begitu tinggi, dan dengan volume yang yang tidak begitu besar (sudah merupakan data-data hasil agregasi).

6.1.1 Implementasi Database *User Management* dan OLAP Report

Struktur data *User Management* dan OLAP Report akan diimplementasikan menjadi sebuah database Microsoft Access yang diberi nama **Infomain.mdb**, dengan

pertimbangan bahwa kedua struktur data ini memiliki kesamaan fungsi, yaitu untuk menyimpan aktivitas aplikasi (modul-modul) yang dibuat.

Database ini akan dibangun menggunakan *modelling tools* ERWin berdasarkan relationship yang telah dibuat. ERWin akan membangun (men-*generate*) sebuah DDL (*Data Definition Language*) yang akan merepresentasikan struktur database InfoMain.mdb, seperti yang terlihat dari potongan kode berikut ini:

```
'  Starting Access Basic DAO Session...

Dim ERwinWorkspace As Workspace
Dim ERwinDatabase As Database
Dim ERwinTableDef As TableDef
Dim ERwinQueryDef As QueryDef
Dim ERwinIndex As Index
Dim ERwinField As Field
Dim ERwinRelation As Relation

Set ERwinWorkspace = DBEngine.WorkSpaces(0)

Set ERwinDatabase = ERwinWorkspace.OpenDatabase(sERwinDatabase)

'  CREATE TABLE "CubeField"
Set ERwinTableDef = ERwinDatabase.CreateTableDef("CubeField")
Set ERwinField = _
       ERwinTableDef.CreateField("SourceField", DB_TEXT, 50)

ERwinField.Required = True

...

'  CREATE TABLE "DataUser"
Set ERwinTableDef = ERwinDatabase.CreateTableDef("DataUser")
Set ERwinField = ERwinTableDef.CreateField("UserID", DB_TEXT, 10)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Passwd", DB_TEXT, 50)
ERwinField.Required = True

...
```

6.1.2 Implementasi Database OLAP Entri

Struktur data OLAP Entri akan diimplementasikan menjadi sebuah database Microsoft Access yang dinamai **Simda.mdb**. Database Simda.mdb juga akan dibangun menggunakan *modelling tools* ERWin berdasarkan relationship yang telah dibuat. ERWin akan membangun (men-*generate*) sebuah DDL (*Data Definition*

Language) yang akan merepresentasikan struktur database Simda.mdb, seperti yang terlihat dari potongan kode berikut ini:

```
'  Starting Access Basic DAO Session...

Dim ERwinWorkspace As Workspace
Dim ERwinDatabase As Database
Dim ERwinTableDef As TableDef
Dim ERwinQueryDef As QueryDef
Dim ERwinIndex As Index
Dim ERwinField As Field
Dim ERwinRelation As Relation

Set ERwinWorkspace = DBEngine.WorkSpaces(0)

Set ERwinDatabase = ERwinWorkspace.OpenDatabase(sERwinDatabase)

'  CREATE TABLE "LtKelasProyek"
Set ERwinTableDef = ERwinDatabase.CreateTableDef("LtKelasProyek")
Set ERwinField = _
      ERwinTableDef.CreateField("Keterangan", DB_TEXT, 255)

ERwinTableDef.Fields.Append ERwinField
Set ERwinField = _
      ERwinTableDef.CreateField("KelasProyek", DB_TEXT, 255)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef

'  CREATE INDEX "PrimaryKey"

Set ERwinTableDef = ERwinDatabase.TableDefs("LtKelasProyek")
Set ERwinIndex = ERwinTableDef.CreateIndex("PrimaryKey")
Set ERwinField = ERwinIndex.CreateField("KelasProyek")
ERwinIndex.Fields.Append ErwinField

...
```

6.2 Pembuatan Modul

Bagian ini akan menjelaskan garis besar pembuatan modul-modul yang ada dengan menggunakan bahasa pemrograman Visual Basic 6.0 didukung dengan beberapa komponen-komponen yang sudah ada, sehingga tidak ada tahapan pembuatan komponen dalam tugas akhir ini.

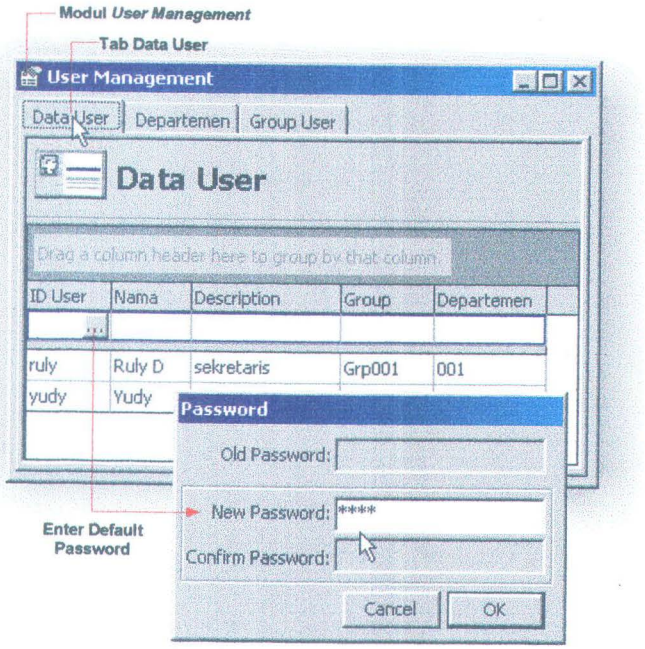
6.2.1 Pembuatan Modul User Management

Pembuatan modul *User Management* menggunakan komponen *Janus Systems Components* dalam implementasi antarmuka dan menggunakan *MS ADO 2.1* dalam

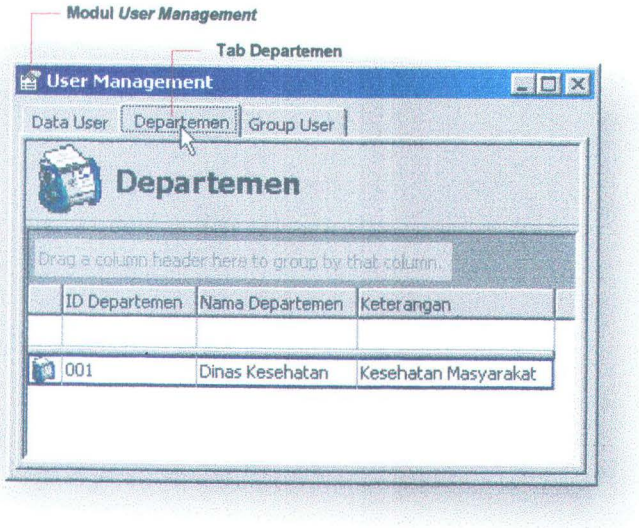
akses database dengan metode data akses yaitu *data binding* seperti layaknya *data control*.

Form pertama merupakan form utama yang dipakai Administrator untuk melengkapi data-data keperluan *User Management*. Form ini terdiri atas 3 tab, yaitu Data User, Departemen dan Group User. Ketiga tab tersebut masing-masing akan digunakan untuk memasukkan data-data *user*, departemen dan *user group*, seperti gambar 6.1.

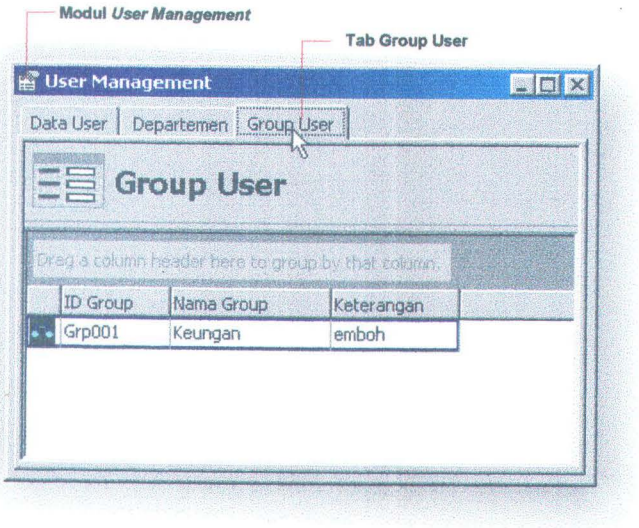
Pada Gambar 6.1, jika seorang Administrator mendaftarkan *user* baru, maka Admin tersebut juga akan mengisikan *default* password yang akan dipergunakan pengguna untuk *login* sistem pertama kali. *Password user* ini tidak akan diperlihatkan pada *form*, tapi secara langsung akan dilakukan *encryption* oleh sistem.



Gambar 6.1 Form Data User

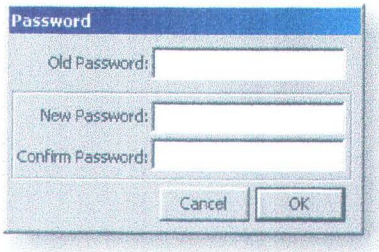


Gambar 6.2 Form Departemen



Gambar 6.3 Form Group User

Form kedua merupakan dialog yang digunakan oleh Administrator untuk mengisikan password awal seorang *user* saat mendaftarkannya pada sistem. Dalam pembuatan modul ini, informasi password tidak ditampilkan di tab Data User. Untuk merubah password seorang *user*, Admin harus melakukan *click* pada tombol UserID. Form ini juga akan digunakan oleh seorang *user* saat akan merubah *password*-nya pada OLAP Entri dan OLAP Report.

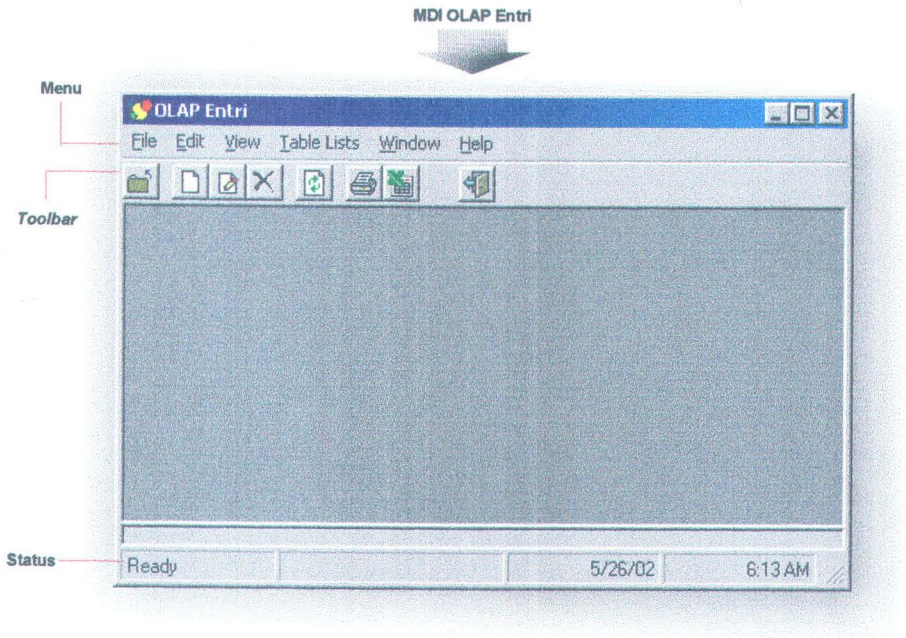


Gambar 6.4 Form Password User

6.2.2 Pembuatan Modul OLAP Entri

Modul OLAP Entri digunakan untuk menjembatani sistem manual yang sudah ada dengan aplikasi OLAP Report yang akan dibuat, yang berisi form-form entri dari data-data yang dibutuhkan dalam proses analisis. Form-form entri yang dibuat terdiri atas Entri Struktur Proyek yang telah ada dan Entri Laporan Keuangan yang mengikuti Standar Akuntansi Keuangan.

Form Utama dari Modul OLAP Entri berupa sebuah MDI (*Multiple Document Interface*) yang dijelaskan sebagai berikut:



Gambar 6.5 MDI OLAP Entri

MDI dari OLAP Entri terdiri dari 3 (tiga) bagian, yaitu: Menu, Toolbar dan Status.

- ***Menu***

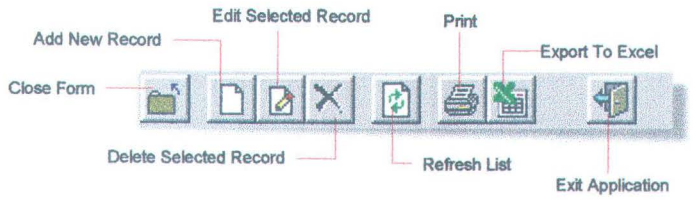
  MDI OLAP Entri memiliki beberapa menu yang akan dirinci dalam tabel berikut:

Tabel 6.1 Menu-menu MDI Olap Entri

Menu	*List*	*Keterangan*
File	Print	Print dokumen
	Print Setup	Setup printer
	Export To Excel	Eksport dokumen ke Excel
	Close	Menutup *form* yang aktif
	Change Password	Merubah *password user*
	Database Setting	Mendefinisikan lokasi database
	Exit	Keluar aplikasi OLAP Entri
Edit	Add	Menambah record
	Edit	Mengedit data
	Delete	Menghapus record
	Refresh	Refresh data
View	ToolBar	Menampilkan ToolBar *(Yes/No)*
	StatusBar	Menampilkan StatusBar *(Yes/No)*
Table List	Struktur Proyek	*Add/Edit/Delete* Struktur Proyek
	Kelas Proyek	*Add/Edit/Delete* Kelas Proyek
	Balance Sheet	*Add/Edit/Delete* Balance Sheet
	Income Statement	*Add/Edit/Delete* Income Statement
	Benefit Cost	*Add/Edit/Delete* Benefit Cost
	Cash Flow	*Add/Edit/Delete* Cash Flow
	Report BCR	Menampilkan Report BCR
	Report IRR	Menampilkan Report IRR
	Report PBP	Menampilkan Report PBP
	Report ROA	Menampilkan Report ROA
	Report ROI	Menampilkan Report ROI
Window	Tile Horisontally	Pengaturan window horisontal
	Tile Vertically	Pengaturan window vertikal

Menu	*List*	*Keterangan*
	Cascade	Pengaturan window bertumpuk
	Close All	Menutup semua window
	Minimize All	Meminimasi semua window
Help	About	Tentang OLAP Entri

- ***Toolbar***



Gambar 6.6 Toolbar MDI OLAP Entri

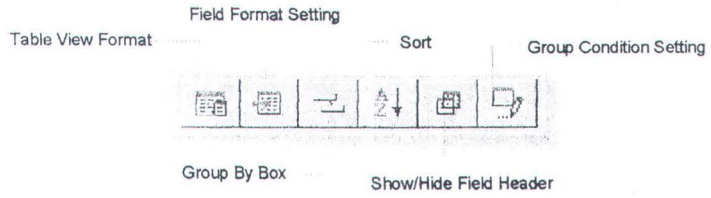
Gambar Icon pada MDI OLAP Entri akan digambarkan dan dijelaskan seperti pada Gambar 6.5 dan Tabel 6.2:

Tabel 6.2 Fungsi Icon-icon pada Toolbar MDI OLAP Entri

Nama Icon	*Fungsi*
Close Form	Menutup form yang aktif
Add New Record	Membuat Report baru
Edit Selected Record	Memodifikasi data yang diinginkan
Delete Selected Record	Menghapus data
Refresh List	*Refresh* data
Print	Mencetak data
Export To Excel	Mengirim data ke *Excel*
Exit Application	keluar dari aplikasi OLAP Entri

Menu Tabel List untuk *option* Struktur Proyek akan ditampilkan dalam bentuk *tree-view* sedangkan *form* selain Struktur Proyek pada pemilihan menu Tabel List akan diberikan tambahan *toolbar* pada *active form*

tersebut untuk fasilitas kastemisasi tampilan (*display*). *Toolbar* tersebut akan digambarkan dan dijelaskan pada Gambar 6.6 dan Tabel 6.2.



Gambar 6.7 Toolbar ActiveForm pada OLAP Entri

Tabel 6.3 Fungsi Icon-icon pada Toolbar ActiveForm OLAP Entri

Nama Icon	*Fungsi*
Table View Format	Digunakan untuk melakukan setting pada tabel, misalnya pemilihan font.
Field Format Setting	Digunakan untuk merubah tampilan type data dari field tanpa mempengaruhi database
Group By Box	Digunakan jika ingin melakukan grouping pada data berdasarkan field.
Sort	Digunakan untuk melakukan multisort dari field yang ada.
Show/Hide Field Header	Digunakan untuk menampilkan/ menyembunyikan field-field yang diinginkan.

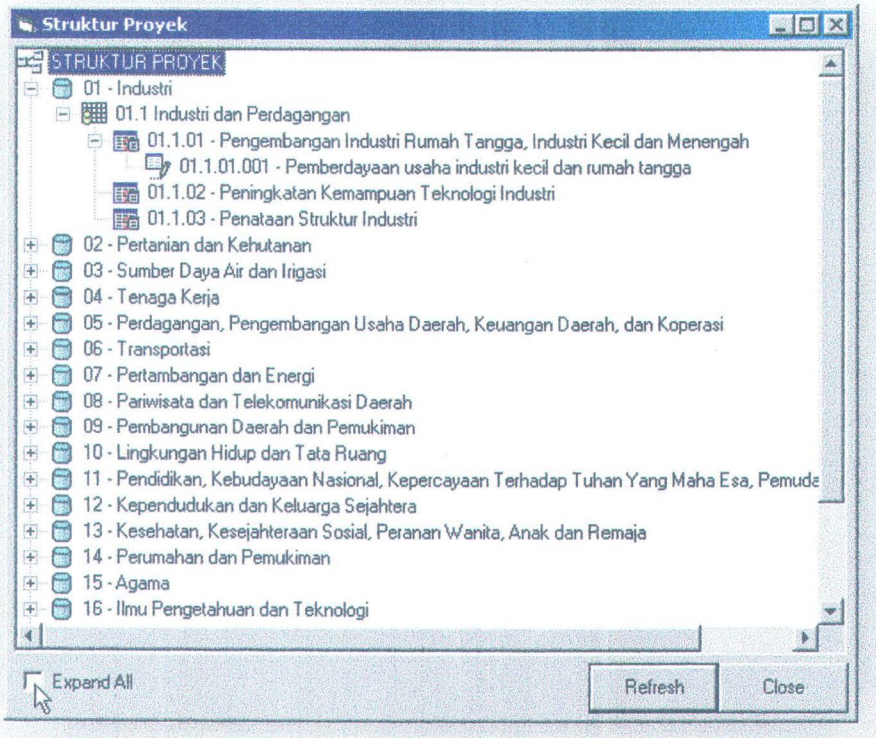
- ***Status***

  Menunjukkan status aplikasi OLAP Entri saat diakses user yang terdiri atas Informasi UserName, Status aplikasi, *Date* dan *Time*.

6.2.2.1 Pembuatan *Form* Struktur Proyek

Entri struktur proyek dibangun dari komponen *Tree View*, sehingga struktur proyek akan ditampilkan secara hirarki mulai dari Sektor → Sub Sektor → Program → Proyek. *Checkbox Expand all* digunakan untuk menampilkan semua *node-node* dalam struktur *tree*.

PERPUSTAKAAN



Gambar 6.8 Penggambaran tree-view pada Struktur Proyek

Pengaturan Struktur Proyek pada sebuah *tree-view* dilakukan dengan kode seperti dibawah ini:

```
Public Sub LoadDataTree()

...

TreeView1.Visible = False
TreeView1.Nodes.Clear
DoEvents

'//Insert Header
Set nodex = TreeView1.Nodes.Add(, , "Key 0", "STRUKTUR PROYEK", 5)
DoEvents

'//Loading Sektor
rs1.Open  "SELECT IDSektor, Sektor, Urutan FROM TbSektor ORDER BY _
          Urutan ASC", ConnMain, adOpenForwardOnly, adLockReadOnly
Do While Not rs1.EOF
       Set nodex = TreeView1.Nodes.Add("Key 0", tvwChild, "AFF-" & _
       rs1("IDSektor"), rs1("Urutan") & " - " & rs1("Sektor"), 1)

       '//Select Sub Sektor
       rs2.Open  "SELECT IDSubSektor,SubSektor,Urutan FROM _
                 TbSubSektor WHERE IDSektor=" & rs1("IDSektor") & _
                 "ORDER BY Urutan ASC", ConnMain, adOpenForwardOnly,
                 _ adLockReadOnly
       Do While Not rs2.EOF
```

```
                Set nodex = TreeView1.Nodes.Add("AFF-" &
                rs1("IDSektor"), tvwChild, rs1("IDSektor") & "#" &
                rs2("IDSubSektor"), rs1("Urutan") & "." &
                rs2("Urutan") & " " & rs2("SubSektor"), 2)

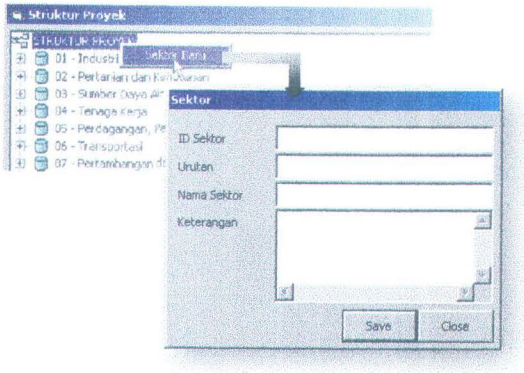
                '//Select Program
                rs3.Open  "SELECT Program,Urutan FROM TbProgram WHERE
                          IDSektor=" & rs1("IDSektor") & " AND
                          IDSubSektor=" & rs2("IDSubSektor") &
                          " ORDER BY Urutan ASC ", ConnMain,
                          adOpenForwardOnly, adLockReadOnly
                Do While Not rs3.EOF
                     Set nodex = TreeView1.Nodes.Add(rs1("IDSektor") &
                                 "#" & rs2("IDSubSektor"), tvwChild,
                                 rs1("IDSektor") & "#" &
                                 rs2("IDSubSektor") & "#" &
                                 rs3("Program"), rs1("Urutan") & "." &
                                 rs2("Urutan") & "." & rs3("Urutan") &
                                 " - " & rs3("Program"), 3)

                     '//SELECT Proyek
                     rs4.Open  "SELECT Proyek, Urutan FROM TbProyek
                               WHERE IDSektor=" & rs1("IDSektor") &
                               " AND IDSubSektor=" & rs2("IDSubSektor")
                               & " AND Program='" & rs3("Program") &
                               "'", ConnMain, adOpenForwardOnly,
                               adLockReadOnly
                     Do While Not rs4.EOF
                        Set nodex =
                        TreeView1.Nodes.Add(rs1("IDSektor") & "#" &
                        rs2("IDSubSektor") & "#" & rs3("Program"),
                        tvwChild, rs1("IDSektor") & "#" &
                        rs2("IDSubSektor") & "#" & rs3("Program") &
                        "#" & rs4("Proyek"), rs1("Urutan") & "." &
                        rs2("Urutan") & "." & rs3("Urutan") & "." &
                        rs4("Urutan") & " - " & rs4("Proyek"), 4)
                        rs4.MoveNext
                        DoEvents
                     Loop
                     rs4.Close
                     rs3.MoveNext
                     DoEvents
                Loop
                rs3.Close
                rs2.MoveNext
                DoEvents
           Loop
           rs2.Close
           rs1.MoveNext
           DoEvents
      Loop
      rs1.Close
      TreeView1.Visible = True
      TreeView1.Nodes("Key 0").Expanded = True

Exit Sub
errHandler:
TreeView1.Visible = True
Err.Raise Err.Number
End Sub
```

- ***Pembuatan form entri Sektor***

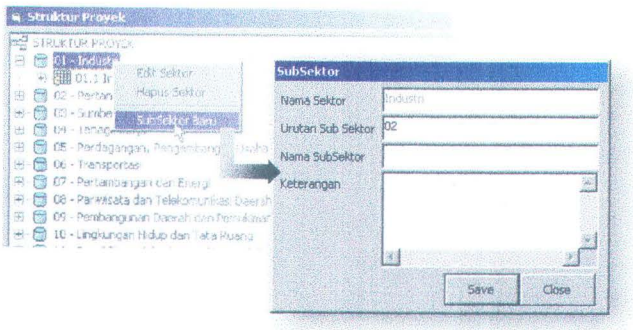
Add/Edit/Delete data Sektor dilakukan dengan *click* kanan pada *node* Struktur Proyek. *Event click* ini akan menampilkan form Sektor. Isian Urutan akan dipergunakan sebagai kode sebuah Sektor yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree*.



Gambar 6.9 Form Entri Sektor

- ***Pembuatan form entri Sub Sektor***

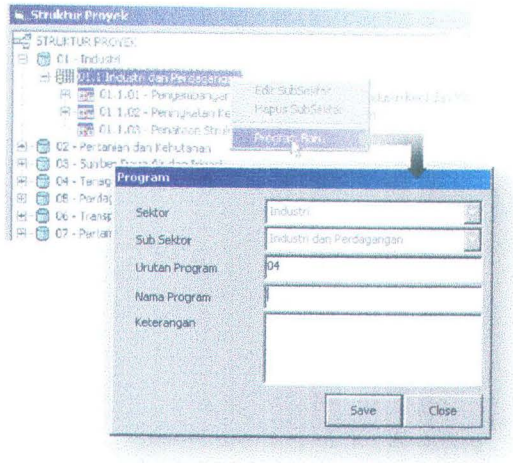
Add/Edit/Delete data SubSektor dilakukan dengan *click* kanan pada *node* tipe Sektor. *Event click* ini akan menampilkan *form* Sub Sektor. Isian Urutan akan dipergunakan sebagai kode sebuah Sub Sektor yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree* setelah sebelumnya didahului dengan penulisan urutan Sektor induknya.



Gambar 6.10 Form Entri SubSektor

- ***Pembuatan form entri Program***

Add/Edit/Delete data Program dilakukan dengan *click* kanan pada Sub Sektor. *Event click* ini akan menampilkan *form* Program. Isian Urutan akan dipergunakan sebagai kode sebuah Program yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree* setelah sebelumnya didahului dengan penulisan urutan Sektor dan Sub Sektor induknya.



Gambar 6.11 Form Entri Program

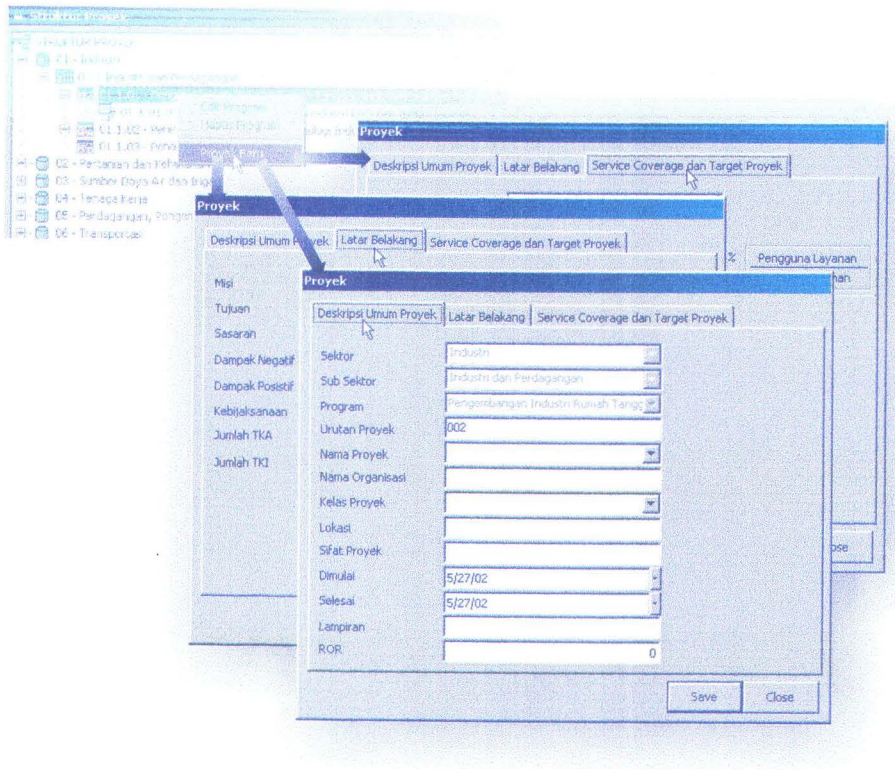
- ***Pembuatan form entri Proyek***

Add/Edit/Ddelete data Proyek dilakukan dengan *click* kanan pada Program. *Event click* ini akan menampilkan form Proyek yang terdiri atas 3 tab, yaitu Deskripsi Umum Proyek, Latar Belakang, dan tab Service Coverage dan Target Proyek.

Tab Deskripsi Umum

Digunakan untuk menginputkan data-data umum yang dimiliki oleh sebuah proyek. Isian Urutan akan dipergunakan sebagai kode sebuah Proyek yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree*

setelah sebelumnya didahului dengan penulisan urutan Sektor, Sub Sektor dan Program induknya.



Gambar 6.12 Form Entri Proyek

Tab Latar Belakang

Digunakan untuk menginputkan data-data latar belakang dari diadakannya sebuah proyek.

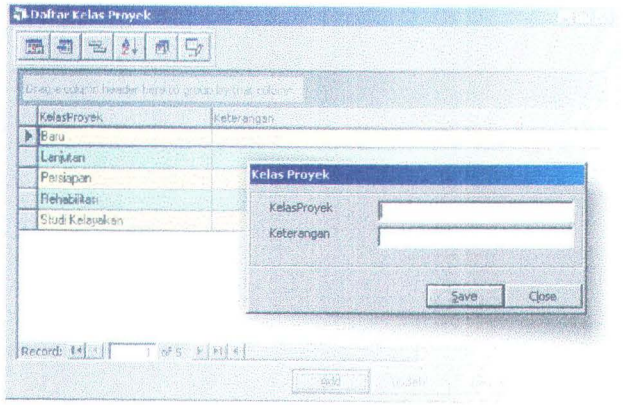
Tab Service Coverage dan Target Proyek

Digunakan untuk menginputkan data-data *service coverage* dan target dari diadakannya sebuah proyek.

Service Coverage akan mengindikasikan perbandingan jumlah populasi pengguna layanan hasil proyek tersebut terhadap jumlah keseluruhan populasi yang berada disekitar pembuatan proyek.

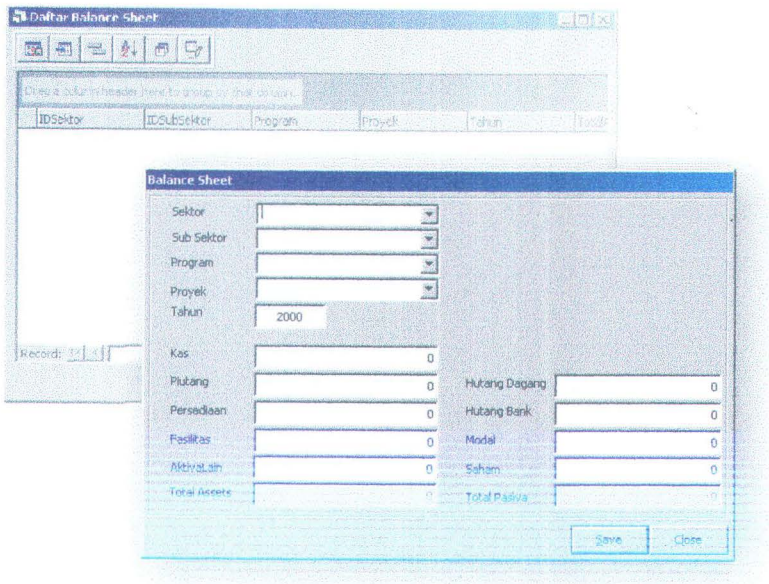
6.2.2.2 Pembuatan *Form* Entri Kelas Proyek

Entri Kelas Proyek akan digunakan sebagai referensi oleh Proyek untuk menentukan kategori dari suatu proyek. Pemasukan data Kelas Proyek melalui form berikut ini:



Gambar 6.13 Form Entri Kelas Proyek

6.2.2.3 Pembuatan *Form* Entri Balance Sheet



Gambar 6.14 Form Entri Balance Sheet

Balance Sheet digunakan untuk menunjukkan posisi keuangan sebuah proyek pada suatu periode tertentu. Entri di lajur sebelah kiri merupakan pencatatan Harta dan entri yang berada pada lajur sebelah kanan dipergunakan untuk mencatat Kewajiban dan Modal. Pencatatan tersebut disusun berdasarkan urutan kemampuan kepemilikan untuk digunakan sewaktu-waktu dibutuhkan.

TotalAssets = Kas + Piutang + Persediaan + Fasilitas + AktivaLain

TotalPassiva = HutangDagang + HutangBank + Modal + Saham

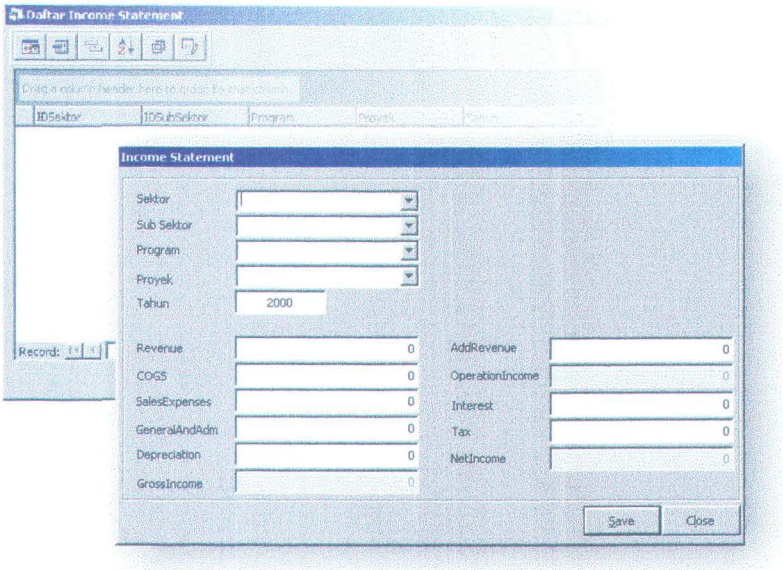
6.2.2.4 Pembuatan *Form* Entri Income Statement

Income Statement digunakan untuk mengisikan laporan yang sistematis tentang penghasilan, biaya, rugi-laba yang diperoleh selama periode tertentu.

GrossIncome = Revenue - COGS

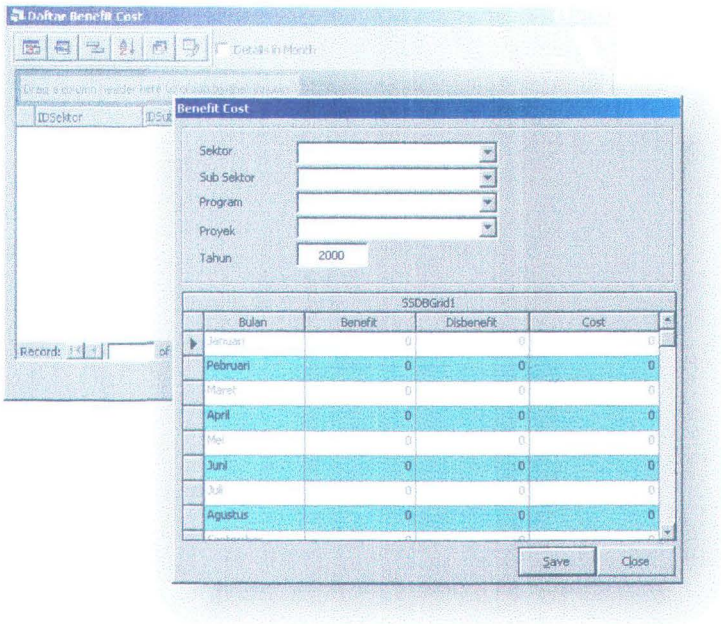
OperationIncome = GrossIncome – (SalesExpenses + GeneralAndAdm + Depreciation + Interest + Tax)

NetIncome = OperationIncome + AddRevenue



Gambar 6.15 Form Entri Income Statement

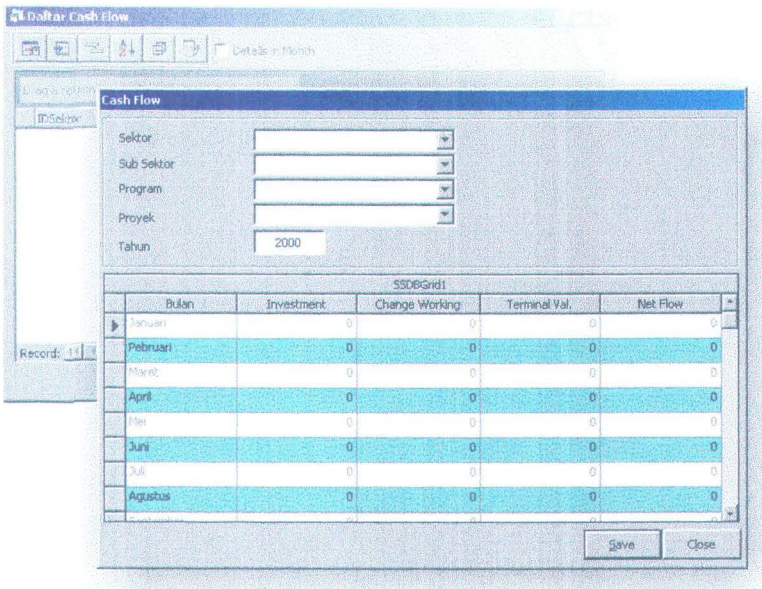
6.2.2.5 Pembuatan *Form* Entri Benefit Cost



Gambar 6.16 Form Entri Benefit Cost

Digunakan untuk mencatat nilai kemanfaatan sebuah proyek dibandingkan dengan biaya yang harus dikeluarkan untuk merelisasikan proyek tersebut.

6.2.2.6 Pembuatan *Form* Entri Cash Flow

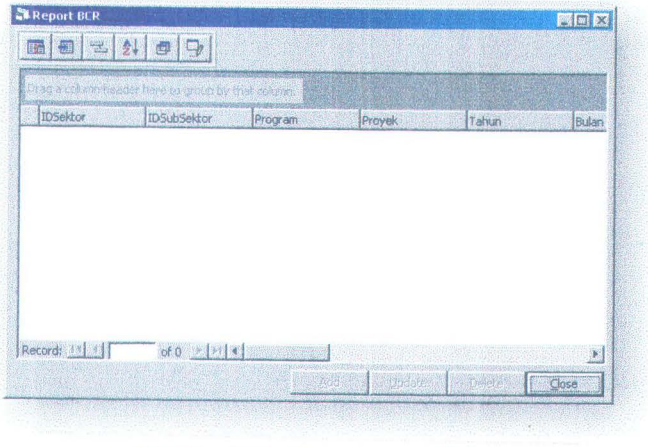


Gambar 6.17 Form Entri Cash Flow

Digunakan untuk memberikan informasi historis mengenai perubahan kas yang diklasifikasikan berdasarkan aktivitas operasi, investasi maupun pendanaan (*financing*) selama periode tertentu.

NetFlow = Investment – ChangeWorking + TerminalVal.

6.2.2.7 Pembuatan *Form* Report BCR



Gambar 6.18 Form Report BCR

Report BCR bersifat *read-only*. Report ini akan dipergunakan untuk menampilkan hasil agregasi data dari *Benefit Cost*. Penghitungan BCR menggunakan sebuah *public sub* seperti berikut:

```
Public Sub HitungBCR(pSektor As Long, pSubSektor As Long, _
      pProgram As  String, pProyek As String, pROR As Double)

On Error GoTo errDesc
Dim rs As New ADODB.Recordset
Dim IPVTotal As Double, OPVTotal As Double
Dim i As Integer, BCR As Double

ReDim ArrBCR(0)
rs.Open "SELECT Tahun,Bulan,IDSektor,IDSubSektor,Program, _
        Proyek,Benefit, Disbenefit,Cost FROM TbBenefitCost WHERE " _
        & "IDSektor=" & pSektor & " AND IDSubSektor=" & pSubSektor _
        & " AND Program='" & pProgram & "' AND Proyek='" & pProyek _
        & "' ORDER BY Tahun,Bulan", ConnMain, adOpenStatic, _
        adLockReadOnly

i = 0
With rs
      ConnMain.Execute "DELETE * FROM TbBCR WHERE IDSektor=" & _
      pSektor & " AND IDSubSektor=" & pSubSektor & " AND Program='"
      _ & pProgram & "' AND Proyek='" & pProyek & "'"
```

```
        Do While Not .EOF
            ReDim Preserve ArrBCR(i)
            ArrBCR(i).IF = !Benefit - !Disbenefit
            ArrBCR(i).OF = !Cost

            BCR = CalcBCR(i, pROR, IPVTotal, OPVTotal)

            ConnMain.Execute "INSERT INTO TbBCR  IDSektor, _
            IDSubSektor,Program,Proyek,Tahun," & "Bulan,UmurProyek, _
            IFt,OFt,IPVTotal,OPVTotal,ROR,BCR) VALUES (" & pSektor & _
            "," & pSubSektor & ",'" & pProgram & "','" & pProyek & _
            "'," & !Tahun & "," & !Bulan & "," & i & ", ( " & _
            ArrBCR(i).IF & " ), ( " & ArrBCR(i).OF & " ), ( " & _
            IPVTotal & " ), ( " & OPVTotal & " ), ( " & pROR & " ), _
            ( " & BCR & " ))"

            i = i + 1
            .MoveNext
        Loop
    End With
Set rs = Nothing
Exit Sub

errDesc:
    Set rs = Nothing
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

Calc BCR adalah sebuah fungsi sebagai berikut:

```
Private Function CalcBCR(pUmurProyek As Integer, pROR As Double,
pIPVTotal As Double, pOPVTotal As Double) As Double

Dim t As Integer, j As Integer
Dim IPVTot As Double, OPVTot As Double

    IPVTot = 0: OPVTot = 0

    For t = 0 To pUmurProyek
        IPVTot = IPVTot + (ArrBCR(t).IF / ((1 + pROR) ^ t))
        OPVTot = OPVTot + (ArrBCR(t).OF / ((1 + pROR) ^ t))
    Next
    pIPVTotal = IPVTot
    pOPVTotal = OPVTot
    CalcBCR = IPVTot / OPVTot
End Function
```

6.2.2.8 Pembuatan *Form* Report IRR

Report IRR bersifat *read-only*. Report ini akan dipergunakan untuk menampilkan hasil agregasi data dari *Cash Flow*. Penghitungan IRR menggunakan sebuah *public sub Hitung IRR* seperti berikut:

```
Public Sub HitungIRR(pSektor As Long, pSubSektor As Long, _
      pProgram As String, pProyek As String)

On Error GoTo errDesc
Dim rs As New ADODB.Recordset
```

```
Dim i As Integer

    ReDim ArrIRRNF(0)
    rs.Open "SELECT Tahun,Bulan,IDSektor,IDSubSektor, Program, _
            Proyek,NetFlow FROM TbCashFlow WHERE " & "IDSektor=" & _
            pSektor & " AND IDSubSektor=" & pSubSektor & " AND _
            Program='" & pProgram & "' AND Proyek='" & pProyek & _
            "' ORDER BY Tahun,Bulan", ConnMain, adOpenStatic, _
            adLockReadOnly

    i = 0
    With rs
          ConnMain.Execute "DELETE * FROM TbIRR WHERE IDSektor=" & _
          pSektor & " AND IDSubSektor=" & pSubSektor & " AND _
          Program='" & pProgram & "' AND Proyek='" & pProyek & "'" _

          Do While Not .EOF

              ReDim Preserve ArrIRRNF(i)
              ArrIRRNF(i) = CDbl(!NetFlow)

              ConnMain.Execute "INSERT INTO TbIRR (IDSektor, _
              IDSubSektor,Program,Proyek,Tahun," &"Bulan, _
              UmurProyek,NF,IRR) VALUES (" & pSektor & "," & _
              pSubSektor & ",'" & pProgram & "','" & pProyek & "'," _
              & !Tahun & "," & !Bulan & "," & i & "," & ArrIRRNF(i) _
              & "," & CalcIRR(i) & ")"

              i = i + 1
              .MoveNext
          Loop
    End With
Set rs = Nothing
Exit Sub

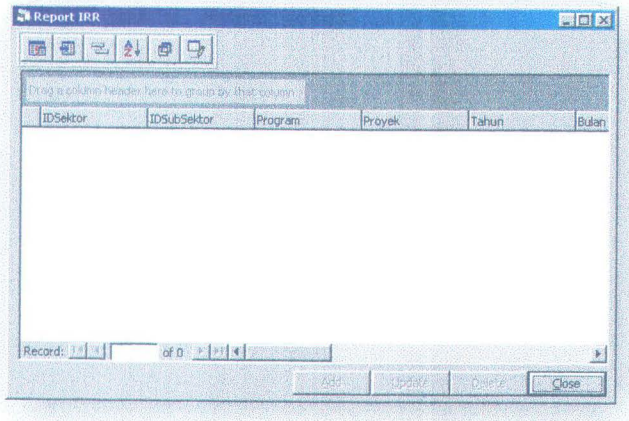
errDesc:
    Set rs = Nothing
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

Calc IRR adalah sebuah fungsi sebagai berikut:

```
Private Function CalcIRR(pUmurProyek As Integer) As Double
Dim t As Integer, j As Integer, Iterasi As Long
Dim NPVTotal As Double, IRR As Double
Dim NPVTotalx As Double, IRRx As Double

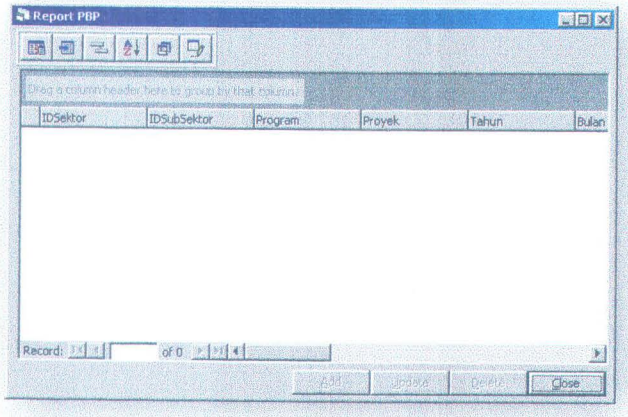
If pUmurProyek = 0 Then
    CalcIRR = 0
    Exit Function
End If
    IRR = 0: j = 1: IRRx = 0: NPVTotalx = 0
ULANG:
    NPVTotal = 0
    For t = 0 To pUmurProyek
        NPVTotal = NPVTotal + (ArrIRRNF(t) / ((1 + IRR) ^ t))
    Next
    If NPVTotal = 0 Then
        'Jika nol langsung ketemu
        CalcIRR = IRR
        Exit Function
    ElseIf NPVTotal > 0 Then
```

```
            CalcIRR = IRR - ((NPVTotal * (IRR - IRRx)) / _
            (NPVTotal - NPVTotalx))
            Exit Function
        ElseIf NPVTotal < 0 Then
            IRRx = IRR
            NPVTotalx = NPVTotal
            IRR = IRR + 1
            GoTo ULANG
        End If
    End Function
```



Gambar 6.19 Form Report IRR

6.2.2.9 Pembuatan *Form* Report PBP



Gambar 6.20 Form Report PBP

Report PBP bersifat *read-only*. Report ini akan dipergunakan untuk menampilkan hasil agregasi data dari *Cash Flow*. Penghitungan PBP menggunakan sebuah *public sub Hitung PBP* seperti berikut:

```
Public Sub HitungPBP(pSektor As Long, pSubSektor As Long, pProgram _
       As String, pProyek As String)

On Error GoTo errDesc
Dim rs As New ADODB.Recordset
Dim i As Integer

ReDim ArrIRRNF(0)

rs.Open "SELECT Tahun,Bulan,IDSektor,IDSubSektor, Program,Proyek, _
        NetFlow FROM TbCashFlow WHERE " & "IDSektor=" & pSektor & _
        " AND IDSubSektor=" & pSubSektor & " AND Program='" & _
        pProgram & "' AND Proyek='" & pProyek & "' ORDER BY Tahun, _
        Bulan", ConnMain, adOpenStatic, adLockReadOnly
    i = 0
    With rs
          ConnMain.Execute "DELETE * FROM TbPBP WHERE IDSektor=" & _
          pSektor & " AND IDSubSektor=" & pSubSektor & " AND _
          Program='" & pProgram & "' AND Proyek='" & pProyek & "'" _
          Do While Not .EOF

              ReDim Preserve ArrIRRNF(i)
              ArrIRRNF(i) = CDbl(!NetFlow)

              ConnMain.Execute "INSERT INTO TbPBP (IDSektor, _
              IDSubSektor,Program,Proyek,Tahun," & "Bulan, _
              UmurProyek, NF,PBP) VALUES (" & pSektor & "," & _
              pSubSektor & ",'" & pProgram & "','" & pProyek & "'," _
              & !Tahun & "," & !Bulan & "," & i & "," & ArrIRRNF(i) _
              & "," & CalcPBP(i) & ")"

              i = i + 1
             .MoveNext
        Loop
    End With
Set rs = Nothing
Exit Sub

errDesc:
    Set rs = Nothing
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

Calc PBP adalah sebuah fungsi sebagai berikut:

```
Private Function CalcPBP(pUmurProyek As Integer) As Double
Dim t As Integer
Dim TotNF As Double

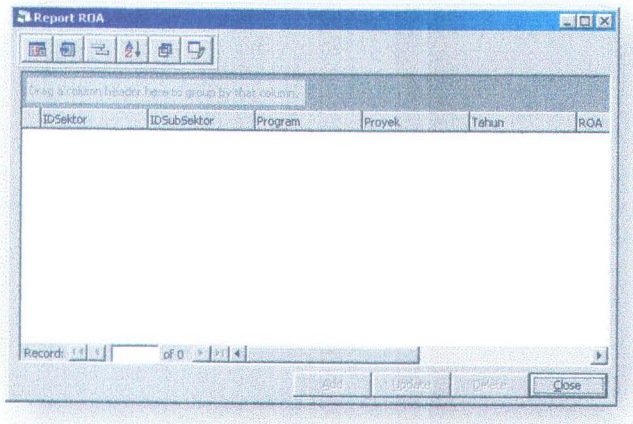
If pUmurProyek = 0 Or pUmurProyek = 1 Then
    CalcPBP = 0
    Exit Function
End If

TotNF = 0
For t = 1 To pUmurProyek - 1
    TotNF = TotNF + ArrIRRNF(t)
Next
If ArrIRRNF(pUmurProyek) = 0 Then
    CalcPBP = -1
    Exit Function
End If
```

```
CalcPBP = (pUmurProyek - 1) + (ArrIRRNF(0) - _
            (TotNF / ArrIRRNF(pUmurProyek)))

End Function
```

6.2.2.10 Pembuatan *Form* Report ROA



Gambar 6.21 Form Report ROA

Report ROA bersifat *read-only*. Report ini akan dipergunakan untuk menampilkan hasil agregasi data dari *Balance Sheet* dan *Income Statement*. Penghitungan ROA menggunakan sebuah *public sub Hitung ROA* seperti berikut:

```
Public Sub HitungROA(pSektor As Long, pSubSektor As Long, _
      pProgram As  String, pProyek As String)

On Error GoTo errDesc
Dim rs As New ADODB.Recordset

rs.Open "SELECT BS.IDSektor, BS.IDSubSektor, BS.Program, BS.Proyek, _
        BS.Tahun,NetIncome," & "TotalAssets,(NetIncome/TotalAssets) _
        AS ROA  FROM TbBalanceSheet AS BS," & "TbIncomeStatement AS _
        SS WHERE  BS.IDSektor=SS.IDSektor AND  " & _
        "BS.IDSubSektor=SS.IDSubSektor AND BS.Program=SS.Program _
        AND  " & "BS.Proyek=SS.Proyek AND BS.Tahun=SS.Tahun AND _
        BS.IDSektor=" & pSektor & " AND BS.IDSubSektor=" & _
        pSubSektor & " AND BS.Program='" & pProgram & "' AND _
        BS.Proyek='" &  pProyek & "'", ConnMain, adOpenStatic, _
        adLockReadOnly

With rs
      Debug.Print rs.RecordCount

      If rs.RecordCount > 0 Then rs.MoveFirst
      Do While Not .EOF
           ConnMain.Execute "DELETE * FROM TbROA WHERE IDSektor=" &
           _  !IDSektor & " AND " & "IDSubSektor=" & !IDSubSektor &
```

```
            " _ AND Program='" & !Program & "' AND " & "Proyek='" & _
            !Proyek & "' AND  Tahun=" & !Tahun

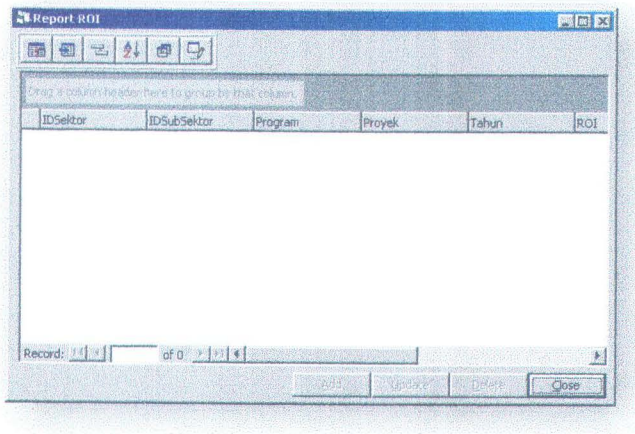
            ConnMain.Execute "INSERT INTO TbROA (IDSektor, _
            IDSubSektor,  Program, Proyek, Tahun," & "NetIncome, _
            TotalAssets, ROA) VALUES (" & !IDSektor & ", " & _
            !IDSubSektor & ", '" & !Program & "', '" & !Proyek & "', _
            " & !Tahun & ", " & !NetIncome & ", " & !TotalAssets & _
            ", " & !roa & ")"

        .MoveNext
    Loop
End With

Set rs = Nothing
Exit Sub

errDesc:
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

6.2.2.11 Pembuatan *Form* Report ROI



Gambar 6.22 Form Report ROI

Report ROI bersifat *read-only*. Report ini akan dipergunakan untuk menampilkan hasil agregasi data dari *Cash Flow* dan *Income Statement*. Penghitungan ROI menggunakan sebuah *public sub Hitung ROI* seperti berikut:

```
Public Sub HitungROI(pSektor As Long, pSubSektor As Long, _
      pProgram As String, pProyek As String)

On Error GoTo errDesc
Dim rs As New ADODB.Recordset
Dim cat As New ADOX.Catalog
Dim cmd As New ADODB.Command
Dim i As Double

cat.ActiveConnection = ConnMain
```

```
cmd.CommandText = "SELECT BS.IDSektor, BS.IDSubSektor, BS.Program,
                  BS.Proyek, BS.Tahun,NetIncome," & "Investment, _
                  Bulan  FROM TbIncomeStatement AS BS,TbCashFlow AS _
                  SS WHERE  " & "BS.IDSektor=SS.IDSektor AND _
                  BS.IDSubSektor=SS.IDSubSektor AND _
                  BS.Program=SS.Program " & "AND BS.Proyek=SS.Proyek_
                  AND BS.Tahun=SS.Tahun AND BS.IDSektor=" & pSektor _
                  & " AND BS.IDSubSektor=" & pSubSektor & " AND _
                  BS.Program='" & pProgram & "' AND BS.Proyek='" & _
                  pProyek & "'"

cat.Views.Append "HitungROI", cmd

rs.Open "SELECT IDSektor,IDSubSektor,Program,Proyek, Tahun, _
        NetIncome,SUM(Investment) AS SumInvestment " & "FROM _
        HitungROI Group BY Tahun,IDSektor,IDSubSektor,Program, _
        Proyek,NetIncome", ConnMain, adOpenStatic, adLockReadOnly

cat.Views.Delete "HitungROI"

With rs

    Do While Not .EOF
          ConnMain.Execute "DELETE * FROM TbROI WHERE IDSektor=" & _
          !IDSektor & " AND " & "IDSubSektor=" & !IDSubSektor & " _
          AND Program='" & !Program & "' AND " & "Proyek='" & _
          !Proyek & "' AND  Tahun=" & !Tahun

          If !suminvestment = 0 Then
               i = -1
          Else

               i = !NetIncome / !suminvestment
          End If

          ConnMain.Execute "INSERT INTO TbROI (IDSektor, _
          IDSubSektor,Program,Proyek,Tahun," & "NetIncome, _
          Investment,ROI) VALUES (" & !IDSektor & "," & _
          !IDSubSektor & ",'" & !Program & "','" & !Proyek & "'," & _
          !Tahun & ", (" & !NetIncome & "), (" & !suminvestment & _
          "), (" & i & "))"

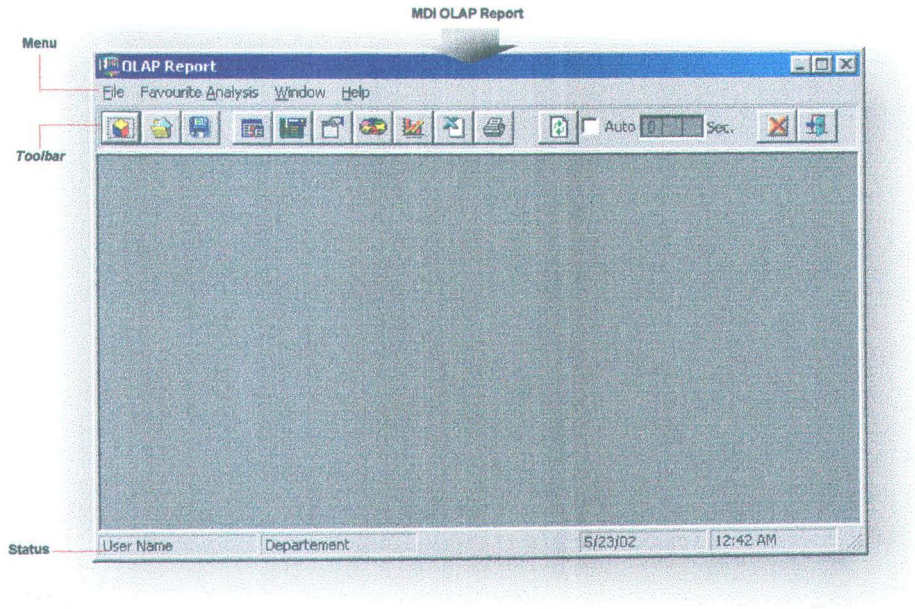
          .MoveNext
    Loop
End With

Set cat = Nothing
Set cmd = Nothing
Set rs = Nothing
Exit Sub

errDesc:
    Set cat = Nothing
    Set cmd = Nothing
    Set rs = Nothing
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

6.2.3 Pembuatan Modul OLAP Report

Seperti pada MDI OLAP Entri, maka MDI OLAP Report juga terdiri dari 3 (tiga) bagian, yaitu: Menu, Toolbar dan Status.



Gambar 6.23 MDI OLAP Report

- ***Menu***

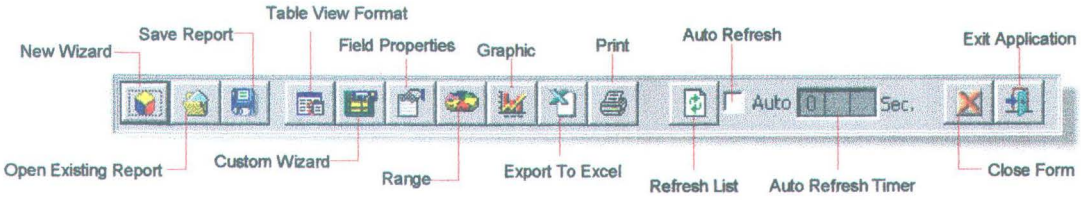
  MDI OLAP Report memiliki beberapa menu yang akan dirinci dalam tabel berikut:

Tabel 6.4 Menu-menu pada MDI OLAP Report

Menu	*List*	*Keterangan*
File	New Analysis	Membuat dokumen analisis baru
	Open	Membuka dokumen analisis lama
	Close	Menutup *form* yang aktif
	Save	Menyimpan dokumen analisis aktif
	Change Password	Mengganti *password user*
	Exit	Keluar aplikasi OLAP Report
Favorite Analysis	Default = *empty*	Berisi dokumen analisis yang telah di-*save* dengan status *Favorite Analysis* = **TRUE**
Window	Tile Horisontally	Pengaturan window horisontal

Menu	*List*	*Keterangan*
	Tile Vertically	Pengaturan window vertikal
	Cascade	Pengaturan window bertumpuk
Help	About	Tentang OLAP Report

- ***Toolbar***



Gambar 6.24 Toolbar OLAP Report

Tabel 6.5 Fungsi Icon-icon pada Toolbar OLAP Report

Nama Icon	*Fungsi*
New Wizard	Membuat *Report* baru dengan wizard
Open Existing Report	Membuka Report lama yang telah dibuat
Save Report	Menyimpan Report yang baru dibuat
Table View Format	Digunakan untuk melakukan setting pada tabel, misalnya pemilihan font
Custom Wizard	Digunakan untuk melakukan konfigurasi ulang terhadap dokumen analisis yang aktif.
Field Properties	Digunakan untuk meng-custom field-field pada Report
Range	Melakukan pewarnaan (colouring) pada field
Graphic	Memindahkan Report dalam bentuk grafik
Export To Excel	Mengirim data ke Excel
Print	Mencetak data
Refresh List	Refresh data
Auto Refresh	Refresh data otomatis berdasarkan setting *timer*
Auto Refresh Timer	*Timer* untuk melakukan Auto Refresh
Close Form	Menutup report yang aktif
Exit Application	Keluar dari aplikasi OLAP Report

- ***Status***

  Menunjukkan status aplikasi OLAP Report saat diakses user yang terdiri atas informasi UserName, Status aplikasi, *Date* dan *Time*.

6.2.3.1 Pembuatan *Analysis Wizard*

Dalam membuat sebuah dokumen analisis, *user* akan dibantu dengan adanya sebuah *wizard* yang akan menuntun tahap demi tahap untuk mendapatkan sebuah dokumen analisis yang dengan tampilan yang bermacam-macam (dinamik) sesuai yang diinginkan *user*. Proses-proses yang menjadi tahapan dalam *analysis wizard* adalah:

- Tahap 1

  Merupakan tahap yang berisi informasi tentang langkah-langkah dalam membangun konfigurasi sebuah dokumen analisis.

- Tahap 2

  Merupakan tahap pemilihan *cube* data, yaitu dengan memilih *source table* yang akan dipergunakan dalam analisis. Tahap kedua ini juga mengatur orientasi *field* untuk setiap *field table* yang dipilih.

  Untuk melakukan penyimpanan semua *main table* beserta *field*-nya dilakukan dengan menggunakan *clsCubes*. Untuk menyimpan semua informasi yang dimiliki oleh *field* yang berkaitan dengan dimensioning, maka dibuat sebuah *class* yaitu *clsPivot*

- Tahap 3

  Tahapan ketiga dalam analysis wizard adalah *data filtering*. Tahapan ini bersifat *optional* bagi user, artinya bahwa tahapan ini dapat langsung dilewati ke tahap selanjutnya.

- Tahap 4

  Merupakan tahap pembuatan *custom field*. Tahapan ini bersifat *optional* bagi pengguna. Pembuatan custom field oleh pengguna akan dibantu dengan pembuatan sebuah Formula editor yang bentuknya mirip kalkulator. Dengan Formula Editor ini, pengguna akan dapat membuat sebuah custom field, tanpa perlu mengingat-ingat field data apa saja yang *available* dalam pembuatan custom field tersebut.

- Tahap 5

  Merupakan proses *finishing*. Tahapan ini sebagai konfirmasi kepada pengguna terhadap setiap proses yang telah dilakukan pada tahap-tahap sebelumnya. Tahap ini dilengkapi dengan SQL-text yang merupakan SQL untuk membangun dokumen analysis berdasarkan proses-proses yang dilakukan pengguna pada tahap-tahap sebelumnya.

Setelah tahap 5 selesai, sistem memulai proses untuk mengimplementasikan konfigurasi yang telah dibangun oleh *pengguna*. Dimulai dari membangun obyek cubes yang merupakan tempat penyimpanan dari semua *field* yang telah dipilih pada step pertama, yang akan digunakan untuk membangun *query/SQL string* untuk mengambil data dari database seperti yang terlihat pada kode dibawah ini.

```
Private Sub SubmitProcess()
On Error GoTo errHandler
Dim cbFld As CubeField
Dim fld As DynamiCubeLibCtl.Field
Dim counter As Long, MaxCounter As Long

    MaxCounter = mCubes.Count
    For counter = 1 To MaxCounter
        mCubes.Remove 1
    Next

    mCubes.SourceTable = mTableName

    'Hidden - FieldList
```

```
    For counter = 1 To UBound(FieldList)
       mCubes.Add DBField, FieldList(counter).Name, _
FieldList(counter).Name, "", NonePart, Hidden, _
FieldList(counter).DataType
    Next

    'Row
    For counter = 1 To UBound(RowFld)
        mCubes.Add DBField, RowFld(counter).Name, _
       RowFld(counter).Name, "", NonePart, Row, _
       RowFld(counter).DataType

        If RowFld(counter).DataType = adDate Or _
       RowFld(counter).DataType = adDBDate Then

       'Buat dua DatePart
                 mCubes.Add DBField, RowFld(counter).Name, "Year_" & _
       RowFld(counter).Name,"", YearPart, Row, _
       RowFld(counter).DataType

            mCubes.Add DBField, RowFld(counter).Name, "Quarter_" & _
       RowFld(counter).Name,"", Quarter, Row, _
       RowFld(counter).DataType
        End If
   Next

   'Column
   For counter = 1 To UBound(ColFld)
       mCubes.Add DBField, ColFld(counter).Name, _
       ColFld(counter).Name,"", NonePart, Column, _
       ColFld(counter).DataType

      If ColFld(counter).DataType = adDate Or _
      ColFld(counter).DataType = adDBDate Then

      'Buat dua DatePart
                mCubes.Add DBField, ColFld(counter).Name, "Year_" & _
      ColFld(counter).Name,"", YearPart, Column, _
      ColFld(counter).DataType

           mCubes.Add DBField, ColFld(counter).Name, "Quarter_" & _
      ColFld(counter).Name,"", Quarter, Column, _
      ColFld(counter).DataType
       End If
  Next

  'Page
  For counter = 1 To UBound(PageFld)
      mCubes.Add DBField, PageFld(counter).Name, _
      PageFld(counter).Name,"", NonePart, Page, _
      PageFld(counter).DataType

      If PageFld(counter).DataType = adDate Or _
      PageFld(counter).DataType = adDBDate Then

      'Buat Dua DatePart
           mCubes.Add DBField, PageFld(counter).Name, "Year_" & _
      PageFld(counter).Name,"", YearPart, Page, _
      PageFld(counter).DataType

     mCubes.Add DBField, PageFld(counter).Name, "Quarter_" &_
      PageFld(counter).Name,"", Quarter, Page, _
      PageFld(counter).DataType
      End If
```

```
    Next

    'Data
    For counter = 1 To UBound(DataFld)
        mCubes.Add DBField, DataFld(counter).Name, _
       DataFld(counter).Name,"", NonePart, Data, _
       DataFld(counter).DataType
    Next

    'Custom
    For counter = 1 To UBound(Custom)
       mCubes.Add SummaryField, Custom(counter).Name, _
       Custom(counter).Name, _
Custom(counter).SummaryExpr, NonePart, _ Custom(counter).Orientation,
Custom(counter).DataType
    Next
       ...
```

Berikutnya adalah proses pengambilan filter data yang telah didefinisikan pada step ketiga, seperti pada potongan kode dibawah ini:

```
...

C1QueryFrame1.Item.MoveToRoot
If C1QueryFrame1.Item.HasChildren Then
       C1Query1.SQLInfo.TableQuote = c1qQuoteBrackets
       C1Query1.BuildSQL
       WhereClause = " FROM " & C1Query1.ClauseFrom & " WHERE " & _
C1Query1.ClauseWhere
Else
       WhereClause = " FROM [" & mTableName & "]"
End If
...
```

Proses dilanjukan dengan pembangunan konfigurasi field pada komponen visual tampilan OLAP (Dynamic Cube), seperti terlihat pada kode dibawah ini:

```
    ...

    With mCube
            .AutoDataRefresh = False
            .CursorStyle = DCLight
            .DCConnectType = DCCT_UNBOUND
            .DCConnect = strDbConnection
            .QueryByPass = False
           dcDeleteAllFrom .colFields
           dcDeleteAllFrom .PageFields
           dcDeleteAllFrom .RowFields
           dcDeleteAllFrom .DataFields
    End With

    'Row
    For counter = 1 To UBound(RowFld)
           mCube.Fields.Add RowFld(counter).Name, RowFld(counter).Name, _
           Row
```

```
        If RowFld(counter).DataType = adDate Or _
              RowFld(counter).DataType = adDBDate Then

              'Buat dua DatePart
              mCube.Fields.Add "Year_" & RowFld(counter).Name, "Year_"
              & RowFld(counter).Name, Row

              mCube.Fields.Add "Quarter_" & RowFld(counter).Name, _
              "Quarter_" & RowFld(counter).Name, Row
        End If
Next

'Custom
For counter = 1 To UBound(Custom)
        If Custom(counter).Orientation = DCRow Then
              Set fld = mCube.Fields.Add(Custom(counter).Name, _
              Custom(counter).Name, Custom(counter).Orientation)
              fld.AggregateFunc = DCSum
        End If
Next

'Column
For counter = 1 To UBound(ColFld)
       mCube.Fields.Add ColFld(counter).Name, ColFld(counter).Name, _
       Column

       If ColFld(counter).DataType = adDate Or _
             ColFld(counter).DataType = adDBDate Then

       'Buat dua DatePart
           mCube.Fields.Add "Year_" & ColFld(counter).Name, "Year_" &
           ColFld(counter).Name, Column

           mCube.Fields.Add "Quarter_" & ColFld(counter).Name, _
             "Quarter_" & ColFld(counter).Name, Column
        End If
    Next

'Custom
For counter = 1 To UBound(Custom)
       If Custom(counter).Orientation = DCColumn Then
             Set fld = mCube.Fields.Add(Custom(counter).Name, _
             Custom(counter).Name, Custom(counter).Orientation)
             fld.AggregateFunc = DCSum
       End If
Next

'Page
For counter = 1 To UBound(PageFld)
      mCube.Fields.Add PageFld(counter).Name,
      PageFld(counter).Name,_ Page

      If PageFld(counter).DataType = adDate Or _
      PageFld(counter).DataType = adDBDate Then

      'Buat Dua DatePart
      mCube.Fields.Add "Year_" & PageFld(counter).Name, "Year_" &
            PageFld(counter).Name, Page

      mCube.Fields.Add "Quarter_" & PageFld(counter).Name, _
            "Quarter_" & PageFld(counter).Name, Page
      End If
Next
```

```
'Custom
For counter = 1 To UBound(Custom)
      If Custom(counter).Orientation = DCPage Then
            Set fld = mCube.Fields.Add(Custom(counter).Name, _
            Custom(counter).Name, Custom(counter).Orientation)
            fld.AggregateFunc = DCSum
      End If
Next

'Data
For counter = 1 To UBound(DataFld)
      Set fld = mCube.Fields.Add(DataFld(counter).Name, _
      DataFld(counter).Name, Data)
      fld.AggregateFunc = DCSum
Next

'Custom
For counter = 1 To UBound(Custom)
     If Custom(counter).Orientation = DCData Then
           Set fld = mCube.Fields.Add(Custom(counter).Name, _
           Custom(counter).Name, Custom(counter).Orientation)
           fld.AggregateFunc = DCSum
     End If
Next
...
```

Langkah terakhir merupakan proses pembuatan SQL string terhadap konfigurasi yang telah dibangun dan mengambil data di database seperti terlihat pada potongan kode berikut ini:

```
...
mCube.DCRecordSource = mCubes.SelectString & " " & mCubes.Filter
mCube.RefreshData
...
```

6.2.3.2 Pembuatan *Field Properties*

Fungsi utama pada proses ini terjadi pada saat sistem mengimplementasikan perubahan konfigurasi properti pada sebuah field yang dilakukan pengguna, seperti terlihat pada potongan kode di bawah ini:

```
Private Sub ApplyChange(FieldIndex As Long)

On Error GoTo errAppChange

Dim fld As DynamiCubeLibCtl.Field

    On Error Resume Next
    Set fld = mObjCubeControl.Fields(FieldIndex)
    On Error GoTo errAppChange
    If fld Is Nothing Then
        Exit Sub
    End If

    'isi tab general
```

```
        fld.Caption = txt(0)
        If fld.Orientation <> DCData Then
            fld.RankStyle = Combo1(0).ListIndex
    ...

        mObjCubeControl.DCRecordSource = _
              getSQLSelectString(mObjCubeControl, parent.Cubes)
        mObjCubeControl.RefreshData
        parent.isDirty = True
Exit Sub
errAppChange:
    Err.Raise Err.Number, Err.Source, Err.Description
End Sub
```

6.2.3.3 Pembuatan *Range Properties*

Range Properties dibuat dengan sebuah *class modul cbRanges* yang merupakan collection dari *cbRange*. Proses utama dari range properties adalah mendefinisikan interval-interval nilai beserta pewarnaan terhadap interval tersebut.

Berikutnya adalah sistem yang akan mengimplementasikan pewarnaan interval terhadap data-data yang terkandung dalam kontrol visual DCube pada dokumen analisis bersangkutan, seperti terlihat pada potongan kode berikut:

```
...
For Each rng In Ranges
      If cellAttrib.value < rng.EndRange And cellAttrib.value >= _
      rng.StartRange Then
           cellAttrib.ForeColor = rng.Color
           Exit Sub
      End If
Next
...
```

6.3 *Join-Query View*

Penggunaan struktur snowflake pada basisdata OLAP entri menyebabkan adanya sebuah kebutuhan untuk membuat bebrapa *join-query* untuk mengambil data seperti yang telah dijelaskan pada poin 5.2.3.3.

Implementasi dari masing-masing *join-query* yang dimaksud akan diberikan sebagai lampiran, seperti terlihat pada lampiran B.

BAB VII

UJI COBA DAN ANALISA HASIL

BAB VII
UJI COBA DAN ANALISIS HASIL

Bab VII akan membahas tentang uji coba dan analisis hasil yang dilakukan untuk sistem yang telah dibuat. Uji coba dan analisis hasil ini akan menjadi bahan evaluasi yang nantinya akan dapat dipergunakan untuk perbaikan sistem dalam pengembangan selanjutnya. Uji coba dilakukan dengan melakukan beberapa *testing* terhadap fitur-fitur sistem OLAP dengan memberikan inputan variansi data.

7. 1 Lingkungan Uji Coba

Uji coba terhadap sistem yang telah dibuat dilakukan dengan menggunakan lingkungan dengan spesifikasi sebagai berikut:

Spesifikasi system

- Microsoft Windows 98
- Microsoft Office 2000

Spesifikasi hardware

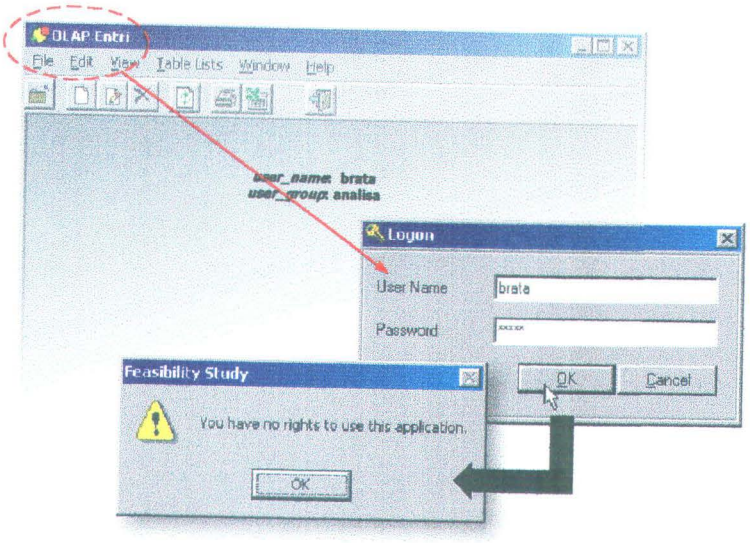
- AMD Duron 1 GHz
- RAM 128 MB
- 20 GB harddisk

7.2 Skenario Uji Coba

Skenario uji coba dilakukan untuk menguji kebenaran fungsionalitas dari proses-proses utama yang ada pada proses Login, *Pengguna Management*, OLAP Entri dan OLAP Report seperti yang dipaparkan pada desain sistem sebelumnya (bab 5).

7.2.1 Uji Coba Login

Uji coba proses login dilakukan dengan pengujian validitas beberapa *user* yang telah terdaftar pada *user management*. Uji coba validitas dilakukan untuk menguji proses login *user* terhadap modul-modul yang ada dengan pemetaan terhadap *user group* dimana *user* tersebut didaftarkan pada *user management*. Jika terdapat ketidaksesuaian antara user group yang dimiliki dan modul yang hendak diakses, maka akan ditampilkan pesan peringatan. Misalnya, jika seorang pengguna berada pada *user group* analisis atau admin, jika mereka masuk pada modul OLAP Entri, maka akan diberikan tanda peringatan. Bahwa OLAP Entri hanya dapat diakses oleh *user-user* pada *user group* data dan keuangan, telah didefinisikan admin pada modul *user management*. Hal serupa akan terjadi untuk modul-modul yang diakses oleh *user group* yang tidak sesuai (tidak memiliki hak akses terhadap modul yang bersangkutan).

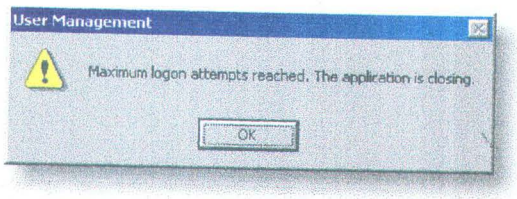


Gambar 7.1 Peringatan user group tidak memilik hak akses terhadap sebuah modul

Uji coba terhadap login juga digunakan untuk menguji keabsahan penginputan kombinasi antara password *user* dan *user name* pada saat proses *login*. Jika seorang pengguna login dengan memberikan inputan kombinasi *user name* dan *password* yang tidak sesuai, maka akan ditampilkan peringatan kesalahan. Jika kesalahan tersebut terjadi lebih dari 3 (tiga) kali, maka akan ditampilkan peringatan bahwa aplikasi tersebut akan ditutup secara otomatis oleh sistem.



Gambar 7.2 Peringatan inputan kombinasi user name dan password yang keliru



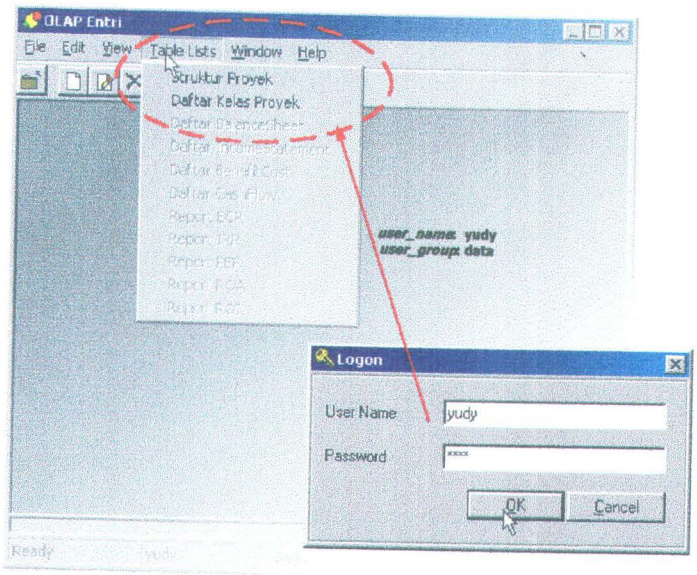
Gambar 7.3 Peringatan penutupan aplikasi segera oleh sistem

7.2.2 Uji Coba *User Management*

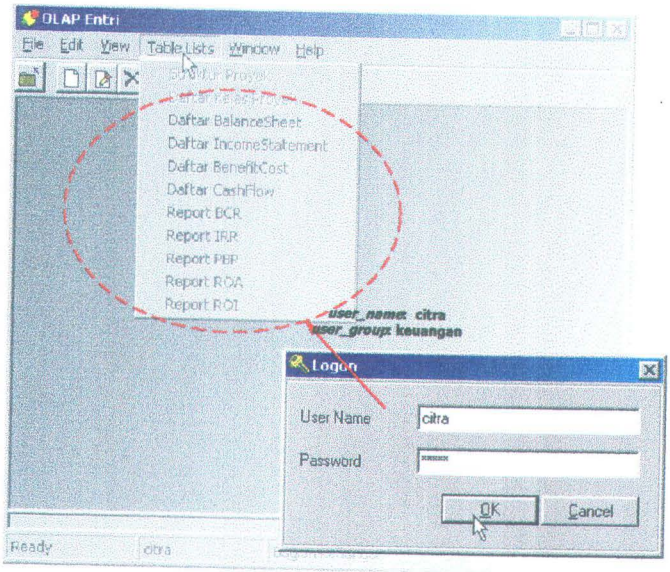
Uji coba ini dilakukan untuk menguji modul *User Management* yang telah dibuat. Modul *user management* ini hanya dapat diakses oleh *user group* Admin. Pengujian yang dilakukan meliputi penambahan *user* baru, *edit* data *user* dan penghapusan hak akses *user*. Seorang *user* bisa didaftarkan untuk memiliki hak akses terhadap beberapa modul yang berbeda dengan memberi *user id* yang berbeda.

7.2.3 Uji Coba OLAP Entri

Uji coba ini dilakukan untuk menguji modul *OLAP Entri* yang telah dibuat. Pengujian yang dilakukan meliputi penambahan data dan update serta penghapusan terhadap data-data yang telah diinputkan.



Gambar 7.4 Menu OLAP Entri yang diperkenankan untuk user group Data



Gambar 7.5 Menu OLAP Entri yang diperkenankan untuk user group Keuangan

Sesuai dengan fungsi dari *user management* yang mengatur hak akses *user*, jika seorang *user* terdaftar pada *user group* bagian Data proyek, maka *user* tersebut hanya dapat mengakses modul OLAP Entri untuk menu-menu Table List Struktur Proyek dan Kelas Proyek. Jika seorang *user* terdaftar pada *user group* bagian keuangan, maka *user* tersebut hanya dapat mengakses modul OLAP Entri untuk menu-menu Balance Sheet, Income Stetement, Benefit Cost dan Cash Flow, serta *report-report* keuangan yang ada.

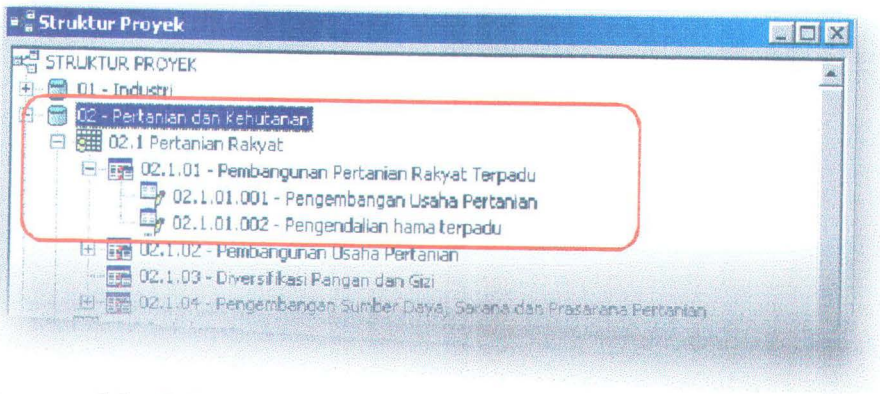
Penggambaran menu OLAP Entri yang dimanajemen berdasarkan hak akses group user, digambarkan pada gambar 7.4 dan 7.5.

Ujicoba Struktur Proyek

Uji coba struktur proyek dilakukan dengan melakukan penambahan sebuah Sektor, Sub Sektor, Program dan Proyek dalam sebuah hirarki sebagai berikut:

Nama Sektor	→ Pertanian dan Kehutanan
Urutan Sektor	→ 02
Nama Subsektor	→ Pertanian Rakyat
Urutan Sub Sektor	→ 1
Nama Program	→ Pembangunan Pertanian Rakyat Terpadu
Urutan Program	→ 01
Nama Proyek	→ Pengendalian Hama Terpadu
Urutan Proyek	→ 002

Dari hasil penginputan struktur proyek tersebut, maka terlihat seperti gambar 7.6. Pengisian secara lengkap terhadap struktur proyek tersebut dapat mereferensi dari lampiran C yang disertakan penulis diakhir pembuatan dokumentasi tugas akhir ini.



Gambar 7.6 Struktur proyek hasil uji coba pemasukan Sektor,SubSektor, Program dan Proyek Baru

7.2.4 Uji Coba OLAP Report

Jika seorang user terdaftar pada *user group* bagian analisis, maka user tersebut hanya dapat mengakses modul OLAP Report. Uji coba ini dilakukan untuk menguji modul *OLAP Report* yang telah dibuat. Pengujian yang dilakukan meliputi uji coba terhadap fitur-fitur OLAP dengan cara pembuatan beberapa *sample* dokumen analisis.

Uji Coba Pembuatan Dokumen Analisis Baru

Uji coba dilakukan dengan melakukan pembuatan dokumen analisis baru dengan menggunakan wizard analisis dengan konfigurasi sebagai berikut:

Source Table: Proyek View
Data Filter:
(
(Proyek_View.JumlahTKI>=0)
)
Hidden Field:
DampakNegatif
DampakPositif
Dimulai
JumlahTKA
Kebijaksanaan
Lampiran
Lokasi
Misi
Organisasi
PopulasiLayanan
Program
ROR

Sasaran
Selesai
ServiceCoverage
SifatProyek
Subsektor
Tujuan

Page Field:
Sektor

Column Field:
Proyek

Row Field:
KelasProyek

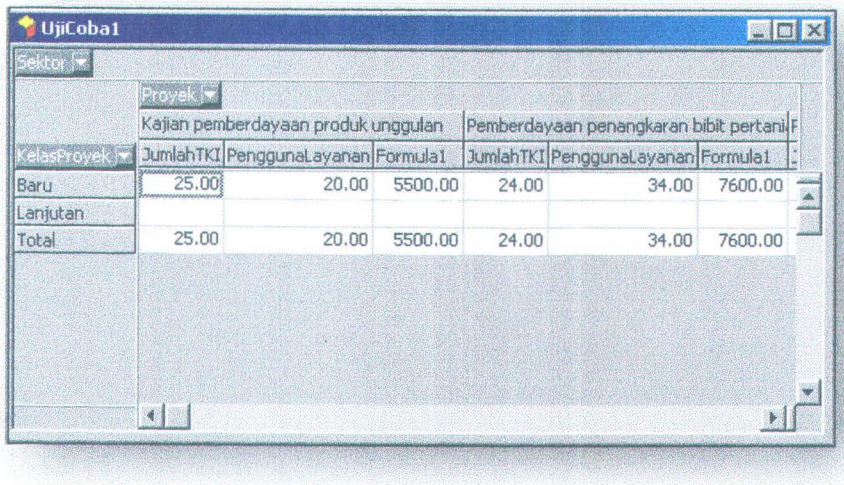
Data Field:
JumlahTKI
PenggunaLayanan

Custom Field:
Formula1

SQL Select String:

```
SELECT [KelasProyek], [Proyek], [Sektor], [JumlahTKI], [PenggunaLayanan], ([PopulasiLayanan] *  100
) as [Formula1]  FROM [Proyek View] Proyek_View WHERE
        (
          (Proyek_View.JumlahTKI>=0)
        )
```

Dari konfigurasi yang telah didefinisikan tersebut, maka akan didapatkan sebuah dokumen analisis seperti pada gambar 7.7.



Gambar 7.7 Hasil Uji Coba Pembuatan Dokumen Analisis Baru

Uji Coba Manipulasi Dokumen Analisis

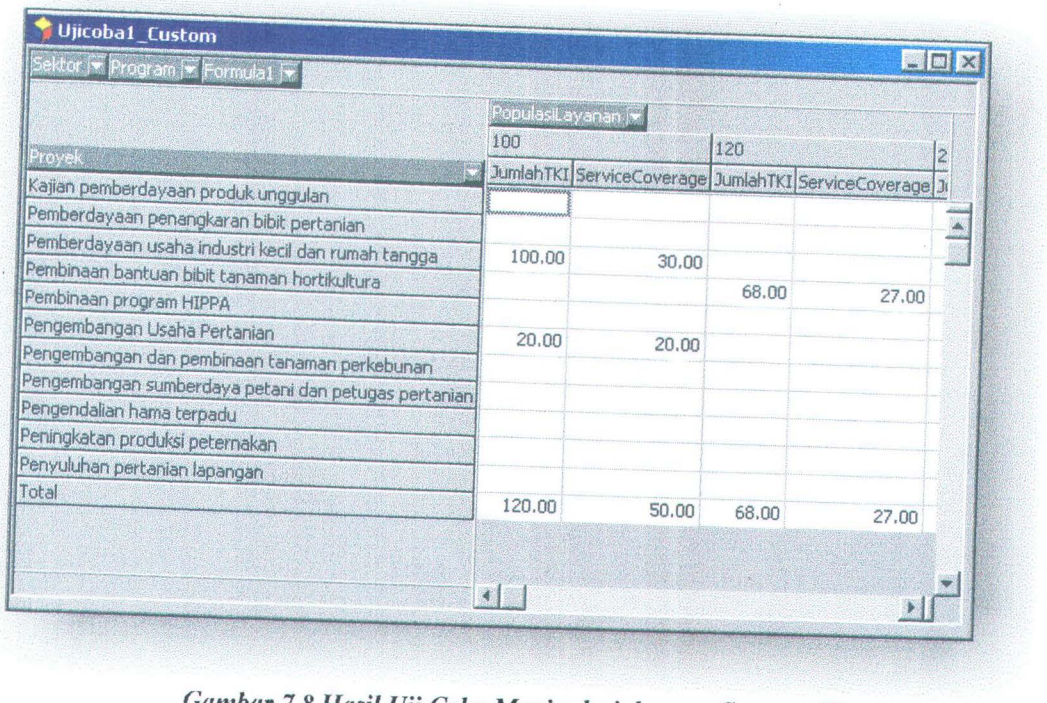
Uji coba manipulasi dokumen analisis dilakukan dengan melakukan perubahan konfigurasi terhadap sebuah dokumen analisis yang telah dibuat pada uji coba sebelumnya. Manipulasi ini dapat berupa satu atau beberapa fungsi, yaitu antara lain:

Custom Wizard

Uji coba ini akan dilakukan pada dokumen analisis **Ujicoba1**. Berdasarkan tahap desain sebelumnya, custom wizard hanya dapat digunakan untuk memanipulasi konfigurasi dari sebuah dokumen analisis tanpa merubah *source table*-nya. Hasil dari custom wizard yang dilakukan, menghasilkan konfigurasi baru sebagai berikut:

Source Table: Proyek View
Data Filter:
Hidden Field:
DampakNegatif
DampakPositif
Dimulai
JumlahTKA
Kebijaksanaan
Lampiran
Lokasi
Misi
Organisasi
ROR
Sasaran
Selesai
SifatProyek
Subsektor
Tujuan
KelasProyek
PenggunaLayanan
Page Field:
Sektor
Program
Column Field:
PopulasiLayanan
Row Field:
Proyek
Data Field:
JumlahTKI
ServiceCoverage
Custom Field:
Formula1
SQL Select String:
SELECT [Proyek], [PopulasiLayanan], [Sektor], [Program], ([JumlahTKA] + [PenggunaLayanan]) as [Formula1], [JumlahTKI], [ServiceCoverage] FROM [Proyek View]

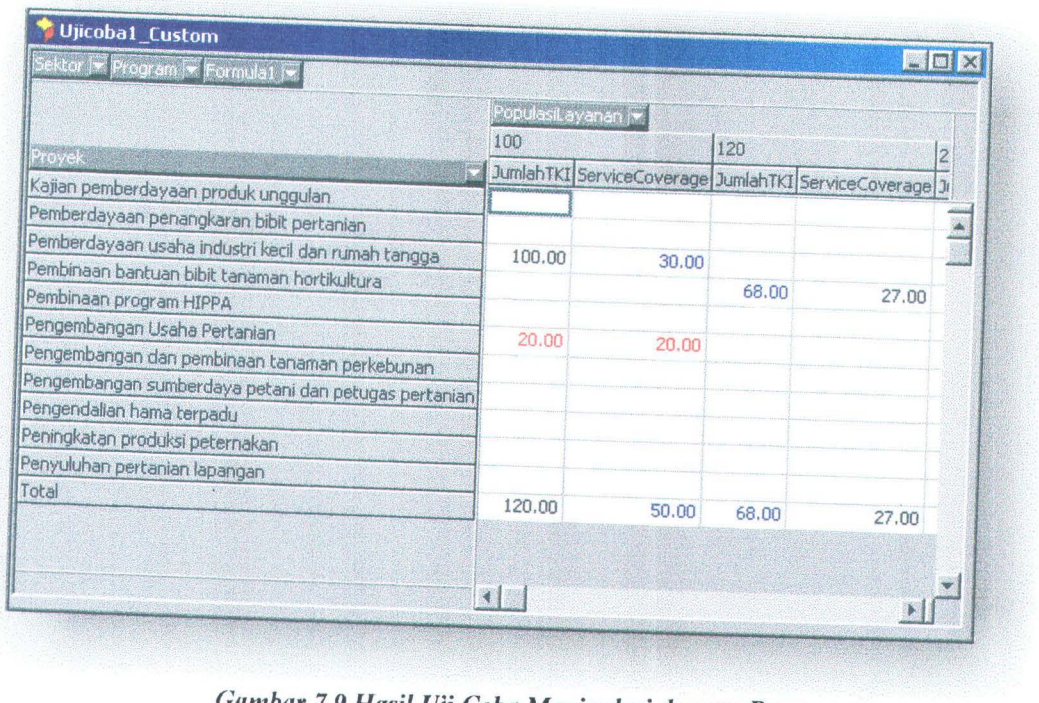
Konfigurasi baru hasil proses custom wizard tersebut menghasilkan perubahan tampilan bagi dokumen analisis **Ujicoba1,** yang selanjutnya akan disimpan dengan nama **Ujicoba1_Custom** pada gambar 7.8.



Gambar 7.8 Hasil Uji Coba Manipulasi dengan Custom Wizard

Range

Uji coba terhadap *range* akan dilakukan terhadap dokumen analisis **Ujicoba1_Custom**. Pemberian warna merah akan dilakukan untuk data numerik yang memenuhi konfigurasi *range* **0<=Range< 25**. Pemberian warna biru akan dilakukan untuk data numerik yang memenuhi konfigurasi *range* **30<=Range<70.** Dari hasil proses range ini, maka tampilan dokumen analisis Ujicoba1_Custom akan berubah menjadi seperti gambar 7.9:



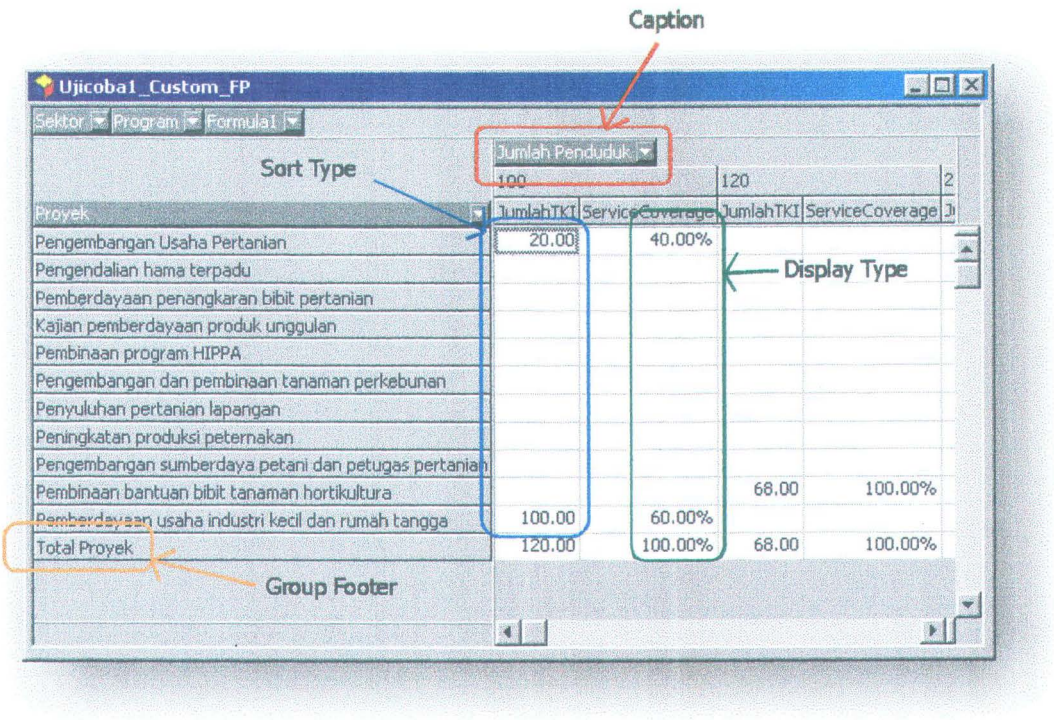
Gambar 7.9 Hasil Uji Coba Manipulasi dengan Range

Fields Properties

Uji coba fitur ini dilakukan terhadap dokumen analisis Ujicoba1_Custom yang telah dibuat. Pengujian dibuat berdasarkan perubahan konfigurasi field-field yang ada didalamnya. Pada uji coba ini, informasi yang dirubah antara lain:

- *Caption* pada field PopulasiLayanan menjadi Jumlah Penduduk.
- *Group Footer* dari field Proyek menjadi Total Proyek.
- Field Proyek diurutkan berdasarkan data (*Rank On*) JumlahTKI dengan *Sort Type Ascending*.
- Merubah tampilan (*Display As*) dari *field ServiceCoverage* menjadi nilai %Column.

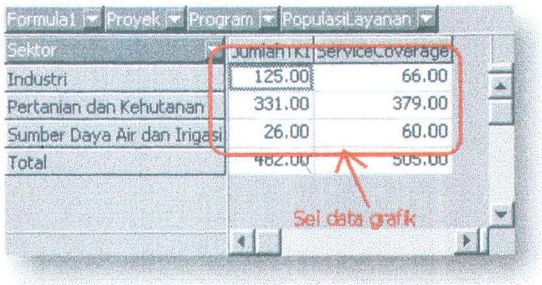
Dari manipulasi yang dilakukan, maka dihasilkan sebuah dokumen analisis yang akan disimpan dengan nama **Ujicoba1_Custom_FP** seperti pada gambar 7.10.



Gambar 7.10 Hasil Uji Coba Manipulasi dengan Fields Properties

Uji Coba Penggambaran Grafik

Uji coba penggambaran grafik dilakukan dengan melakukan pemilihan sekumpulan sel data pada dokumen analisis Ujicoba1_Custom. Pemilihan ini dilakukan dengan mengadakan perubahan orientasi dari beberapa field yang ada. Dari hasil pemilihan ini sel pada gambar 7.11, maka didapatkan gambar grafik seperti pada gambar 7.12.

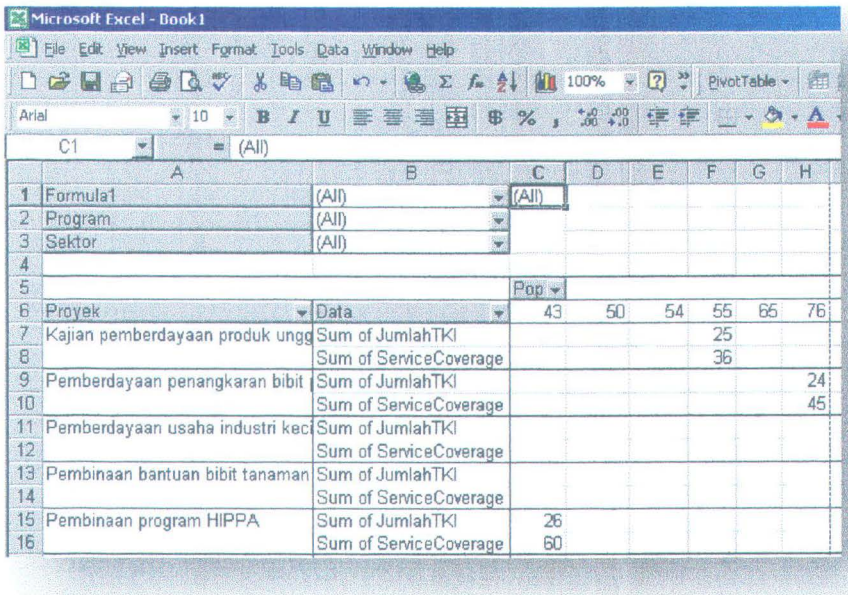


Gambar 7.11 Pemilihan Sel Data Grafik



Gambar 7.12 Hasil Uji Coba Penggambaran Grafik

Uji Coba Export Data ke Excel *Workbook*



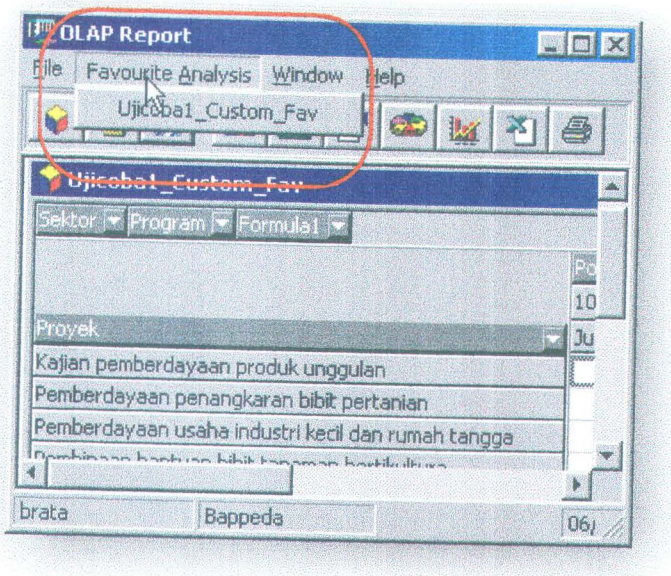
	A	B	C	D	E	F	G	H
1	Formula1	(All)	(All)					
2	Program	(All)						
3	Sektor	(All)						
4								
5			Pop					
6	Proyek	Data	43	50	54	55	66	76
7	Kajian pemberdayaan produk ungg	Sum of JumlahTKI				25		
8		Sum of ServiceCoverage				36		
9	Pemberdayaan penangkaran bibit	Sum of JumlahTKI						24
10		Sum of ServiceCoverage						45
11	Pemberdayaan usaha industri kec	Sum of JumlahTKI						
12		Sum of ServiceCoverage						
13	Pembinaan bantuan bibit tanaman	Sum of JumlahTKI						
14		Sum of ServiceCoverage						
15	Pembinaan program HIPPA	Sum of JumlahTKI	26					
16		Sum of ServiceCoverage	60					

Gambar 7.13 Hasil Uji Coba Export Data ke Excell Workbook

Uji coba ini dilakukan pada dokumen analisis Ujicoba1_Custom. Hasil dari proses pengiriman data ke Excell terlihat pada potongan *workbook* Excell pada gambar 7.13.

Uji Coba Penyimpanan Dokumen Analisis

Uji coba proses penyimpanan dokumen analisis dilakukan terhadap 2 (dua) jenis dokumen analisis yang dapat dikelola oleh sistem. Dokumen analisis biasa dan dokumen analisis yang dijadikan favorite analisis. Dokumen analisis yang disimpan sebagai favorite analysis akan ditambahkan sebagai sub menu pada *Menu Favorite Analisis* pada form utama. Pengujian ini akan dilakukan pada dokumen analisis **Ujicoba1_Custom** yang akan disimpan dengan membuat atribut *Save As Favorite Analysis* menjadi *checked.*



Gambar 7.14 Hasil Uji Coba Penyimpanan Dokumen Favorite Analysis

Uji Coba Membuka Dokumen Analisis Tersimpan

Proses *open* dokumen dapat dilakukan dengan 2 (dua) cara, dengan menggunakan *window open* atau melalui sub menu pada Menu Favorite Analysis (hanya untuk dokumen analisis favorite).

7.3 Analisis Hasil Uji Coba

Dari hasil uji coba yang telah dilakukan, semua proses yang telah didesain pada bagian/bab sebelumnya telah dapat berfungsi sebagaimana seharusnya.

BAB VIII

KESIMPULAN DAN SARAN

BAB VIII
KESIMPULAN DAN SARAN

8.1 Kesimpulan

Teknologi OLAP yang diimplementasikan dalam sebuah perangkat lunak sebagai sebuah *analysis tool* akan dapat membantu pengguna untuk memanipulasi tampilan, memindahkan sudut pandang data, merubah urutan data, menambahkan field-field, dan melakukan perubahan parameter dengan menggunakan fungsi-fungsi *aggregate* sehingga akan dapat menjawab kebutuhan analisis dalam menentukan kebijakan dalam menentukan kelayakan proyek-proyek pembangunan terutama ditinjau dari aspek keuangan.

OLAP Feasibility System yang dibuat pada pengerjaan Tugas Akhir ini terdiri atas tiga modul utama, yaitu: User Management, OLAP Entri dan OLAP Report. Modul User Management akan mengatur hak akses user dalam melakukan entri data dan analisa. Modul OLAP Entri merupakan sebuah aplikasi yang dapat membantu *user* dalam mengorganisasikan dan mengintegrasikan, serta mengolah data yang akan dipergunakan untuk proses analisis selanjutnya. Modul OLAP Report membantu *user* untuk merepresentasikan data dalam berbagai sudut pandang. Modul OLAP Report ini dapat menyimpan setiap dokumen analisis yang dibuat *user* untuk kemudian dapat dipergunakan lagi untuk proses analisis selanjutnya.

8.2 Saran

Saran bagi pengembangan aplikasi *Sistem Feasibility Study* yang menerapkan teknologi OLAP untuk proses investasi yang dikerjakan pada pembuatan tugas akhir ini antara lain adalah:

PERPUSTAKAAN

- Sistem *OLAP Feasibility Study* dapat dikembangkan dengan target struktur basisdata yang lain selain yang dipakai dalam pengerjaan tugas akhir ini.
- Dapat ditambahkan fitur pembuatan cube data secara dinamis, sehingga akan memungkinkan pengguna untuk membuat kombinasi dimensi dan *fact table* yang bervariasi.
- Dapat dikembangkan menjadi sebuah aplikasi yang berbasis web.
- Dapat dikembangkan menjadi sebuah aplikasi yang meggunakan basisdata yang terdistribusi untuk masing-masing proyek, dinas dan badan-badan yang terkait.

Saran diberikan juga untuk pihak Bappeda, dimana saat ini walaupun sudah banyak terpasang komputer-komputer di tiap departemen, tapi keberadaan komputer tersebut belum ditunjang dengan keberadaan sumber daya manusia yang mencukupi; komputer-komputer tersebut belum terintegrasi, sehingga belum dapat dimanfaatkan secara efektif; komputer-komputer tersebut kebanyakan masih memiliki spesifikasi yang sudah ketinggalan teknologi.

Oleh sebab itu, hal utama yang disarankan bagi pihak Bappeda adalah pengintegrasian komputer-komputer tersebut dalam sebuah sistem yang terintegrasi dengan baik, sehingga diharapkan Bappeda akan benar-benar siap dalam mengejar ketinggalan teknologi khususnya dibidang Teknologi Informasi. Dengan adanya integrasi sistem tersebut dan ditunjang dengan sejumlah hardware yang memadai, maka pihak Bappeda akan benar-benar dapat mengimplementasikan aplikasi OLAP yang dibuat pada pengerjaan Tugas Akhir ini dengan lebih baik lagi.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

Amo, William C. , 2000, *Microsoft SQL Server™ OLAP Developer's Guide*, M&T Books

Connolly, Thomas dkk, 1998, *Database Systems – A Practical Approach to Design, Implementation, and Management*, Addison-Wesley

Davis, William S. , 1994, *Bussiness Systems Analysis and Design*, Wadsworth Publishing Company

Halvorson, Michael, 2000, *Step By Step Microsoft Visual Basic 6.0 Professional*, Microsoft Press

IAI, 1995, *Standar Akuntansi Keuangan*, Salemba Empat, Jakarta

Jogiyanto, HM, 1995, *Analisis dan Disain Sistem Informasi*, ANDI Yogyakarta.

Mardiasmo, 2002, *Otonomi dan Manajemen Keuangan Daerah*, ANDI Yogyakarta

Pendse, Nigel, 2001, *The OLAP Report – What is OLAP?*, http:// www. olapreport.com/fasmi.htm

Ramakrishnan, Raghu, 1997, *Database Management System*, McGRAWHILL

Rosjidi, 2001, *Akuntansi Sektor Publik Pemerintah – Kerangka, Standar dan Metode*, Aksara Satu – Surabaya

Sutojo, Siswanto, 2000, *Studi Kelayakan Proyek – Konsep, Teknik dan Kasus*, Damar Mulia Perkasa

Thacker, Ronald, 1979, *Accounting Principles*, Prentice Hall, New Jersey

LAMPIRAN

LAMPIRAN

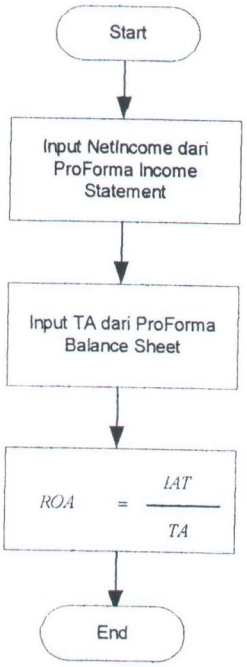
LAMPIRAN A

Desain Algoritma Perhitungan Indikator Keuangan

A.1 Algoritma Perhitungan ROA

Algoritma Perhitungan ROA :

1) Masukkan nilai Net Income dari Pro Forma Income Statement untuk tahun yang diinginkan (NetIncome).
2) Masukkan Total Assets yang terdapat pada tahun tersebut (TA) pada Pro Forma Balance Sheet.
3) Hitung Return On Total Assets (ROA) pada tahun yang diinginkan tersebut dengan rumus: $ROA = \frac{NetIncome}{TA}$



Gambar A.1 Flowchart Algoritma Perhitungan ROA

A.2 Algoritma Perhitungan ROI

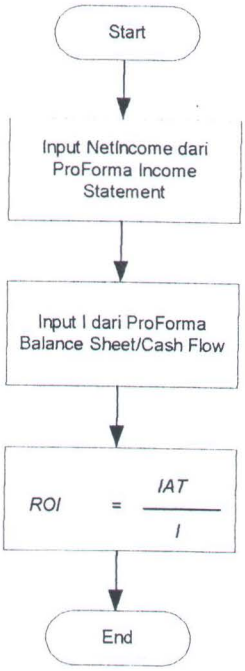
Algoritma Perhitungan ROI

1) Masukkan nilai Net Income dari Pro Forma Income Statement untuk tahun tersebut (NetIncome).

2) Masukkan Total Investment yang terdapat pada tahun ke-nol (I) dari Pro Forma Balance Sheet atau Pro Forma Cash Flows.

3) Hitung Return On Investment (ROI) pada tahun tersebut dengan rumus:

$$ROI = \frac{NetIncome}{I}$$



Gambar A.2 Flowchart Algoritma Perhitungan ROI

A.3 Algoritma Perhitungan IRR

Algoritma Perhitungan IRR

1) Masukkan umur proyek (n) dalam satuan bulan.
2) Masukkan Inflow (IF) dan Outflow (OF) untuk setiap bulannya sepanjang umur proyek tersebut .
3) Hitung Netflow (NF) untuk setiap bulannya sepanjang umur proyek tersebut. Mulai dari t = 0 sampai dengan n, hitung:

$$NF_t = IF_t - OF_t$$

4) Untuk j=1, maka set IRR_j=0
5) Lakukan perhitungan Total Nilai Sekarang atau Net Present Value Total (NPVTotal) untuk IRR_j tertentu. Perhitungan $NPVTotal_j$ adalah:

$$NPVTotal_j = \sum_{t=0}^{n} \frac{NF_t}{(1+IRR_j)^t}$$

Jika nilai $NPVTotal_j < 0$ maka set nilai $IRR_j = IRR_j + 1$ dan ulangi langkah nomor 5.

Jika nilai $NPVTotal_j > 0$ maka lanjutkan langkah nomor 6.

Jika nilai $NPVTotal_j = 0$ maka $IRR^* = IRR_j$.

6) Lakukan perhitungan IRR^* dengan Metoda Interpolasi menggunakan data $NPVTotal_j$, IRR_j, $NPVTotal_{j-1}$, dan IRR_{j-1}. Perhitungannya adalah sebagai berikut:

$$IRR^* = IRR_j - \frac{NPVTotal_j(IRR_j - IRR_{j-1})}{NPVTotal_j - NPVTotal_{j-1}}$$

A.4 Algoritma Perhitungan PBP

Algoritma Perhitungan PBP :

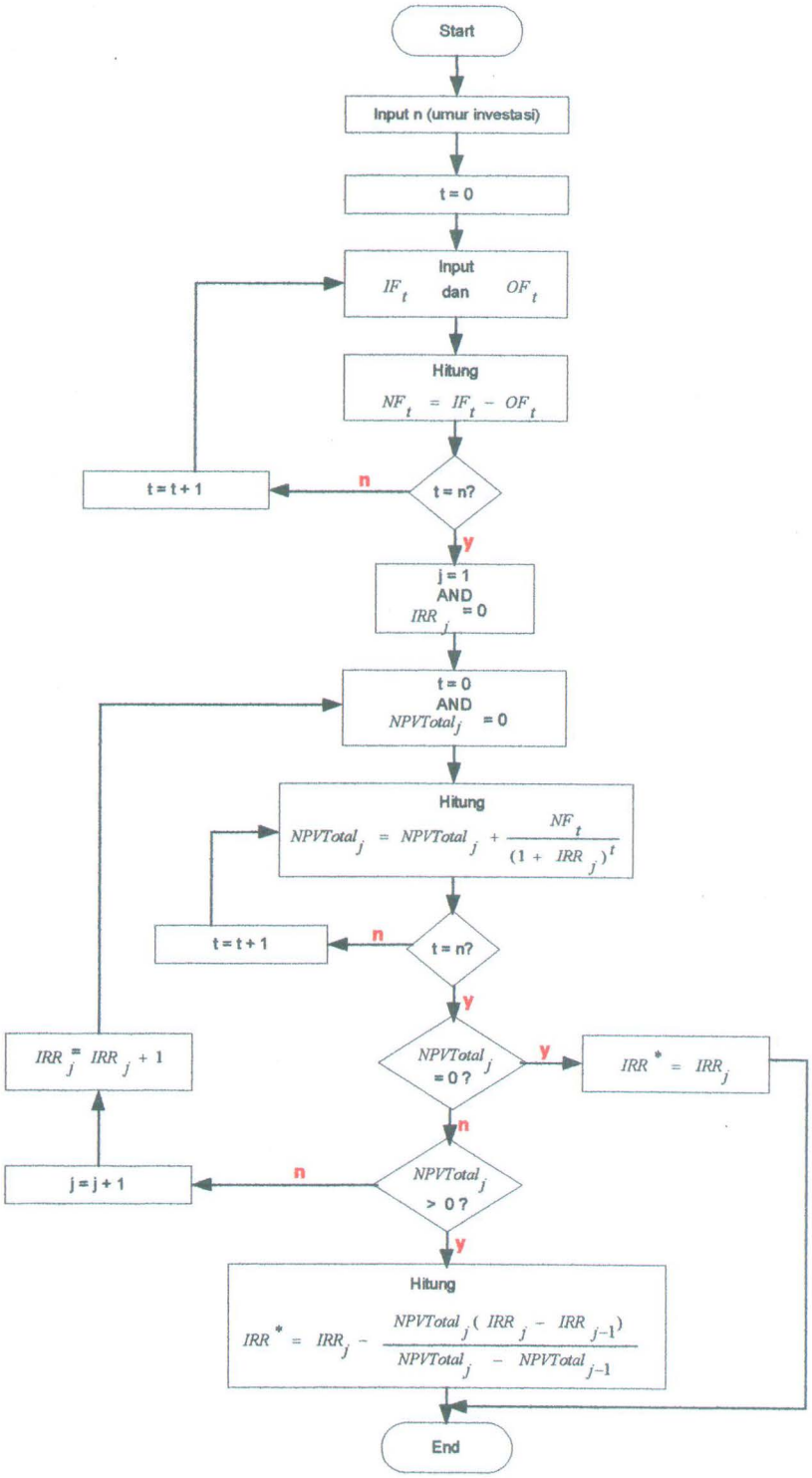
1) Masukkan umur proyek (n) dalam satuan bulan.
2) Masukkan Inflow (IF) dan Outflow (OF) untuk setiap bulannya sepanjang umur proyek tersebut .
3) Hitung Netflow (NF_t) untuk setiap bulannya sepanjang umur proyek tersebut. Mulai dari t = 0 sampai dengan n, hitung

$$NF_t = IF_t - OF_t$$

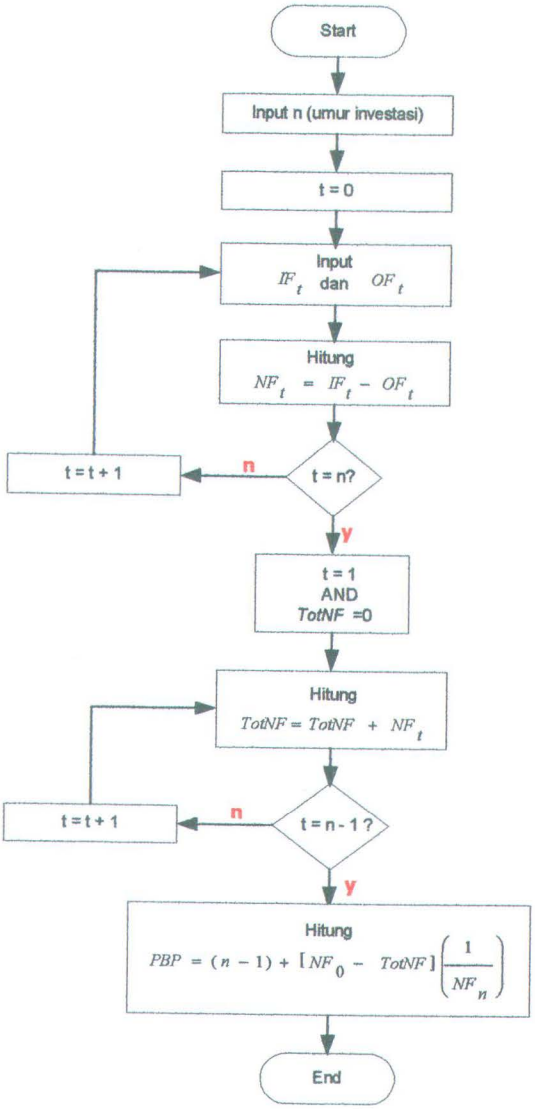
Biaya investasi yang pertama adalah NF_0.

4) Periode Pengembalian (Pay Back Periode) dapat dihitung sebagai berikut :

$$PBP = (n-1) + \left[NF_0 - \sum_{t=1}^{n-1} NF_t \right] \left(\frac{1}{NF_n} \right)$$



Gambar A.3 Flowchart Algoritma Perhitungan IRR



Gambar A.4 Flowchart Algoritma Perhitungan PBP

A.5 Algoritma Perhitungan SC

Algoritma Perhitungan Service Coverage (SC) :

1) Masukkan jumlah penduduk yang berhak untuk mendapatkan layanan pemerintah tersebut (atau "populasi layanan"), PpL.

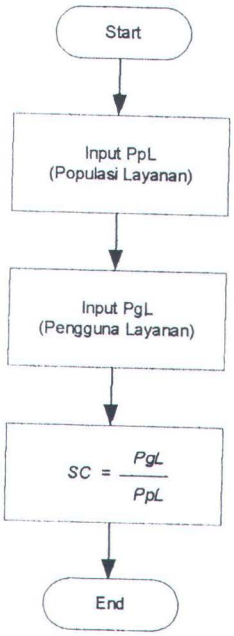
   Nilai PL ini akan sangat tergantung pada : jenis layanan, dan wilayahnya. Misalkan: untuk investasi bangunan SD Inpres di kecamatan A, maka yang disebut populasi layanan (PpL) adalah jumlah anak yang berumur dari 7 – 12 tahun di kecamatan A.

2) Masukkan prediksi jumlah penduduk yang akan mendapatkan layanan pemerintah tersebut, PgL.

Misalkan: daya tampung siswa SD pada bangunan SD Inpres di kecamatan A tersebut.

3) Hitung Service Coverage, dengan perhitungan sebagai berikut:

$$ServiceCoverage = \frac{PgL}{PpL}$$



Gambar A.5 Flowchart Algoritma Perhitungan SC

A.6 Algoritma Perhitungan BCR

Algoritma Perhitungan BCR Untuk Ranking :

1) Masukkan nilai m yang menyatakan jumlah proyek yang akan diranking.
2) Masukkan Rate of Return (ROR) yang menjadi standar aliran pengembalian kas bagi pemerintah. Bisa jadi didasarkan pada SBI (Sertifikat Bank Indonesia). Set nilai $j = 1$ yaitu untuk proyek pertama
3) Masukkan umur proyek (n_j) untuk proyek P_j, dalam satuan bulan.
4) Masukkan Inflow (IF_{jt}) dan Outflow (OF_{jt}) untuk t dari mulai 0 sampai dengan n_j.

Nilai IF_{jt} dan OF_{jt} merupakan bentuk konversi dari benefit, disbenefit, dan cost untuk proyek P_j tersebut dalam satuan rupiah. IF_{jt} didapatkan dari benefit dikurangi disbenefits, sedangkan OF_{jt} didapatkan dari cost.

Konversi ini sesuai dengan panduan dari masing-masing dinas yang ada. Misalkan: panduan dari Departemen Pemukiman dan Prasarana Wilayah menyatakan bahwa benefit pembangunan jalan adalah Rp 300/m-orang. Sehingga jika dibangun jalan sepanjang 100 m dan terdapat 2000 penduduk yang menggunakan jalan tersebut selama bulan Mei, maka didapatkan *cash flow* benefit (atau Inflow) sejumlah Rp 6.000.000 untuk bulan tersebut.

5) Lakukan perhitungan Total Nilai Sekarang atau Inflow Present Value Total (IPVTotal) untuk Inflow proyek j. Perhitungan $IPVTotal_j$ adalah:

$$IPVTotal_j = \sum_{t=0}^{n} \frac{IF_t}{(1+ROR)^t}$$

6) Lakukan perhitungan juga Total Nilai Sekarang atau Outflow Present Value Total (OPVTotal) untuk Outflow proyek j. Perhitungan $OPVTotal_j$ adalah:

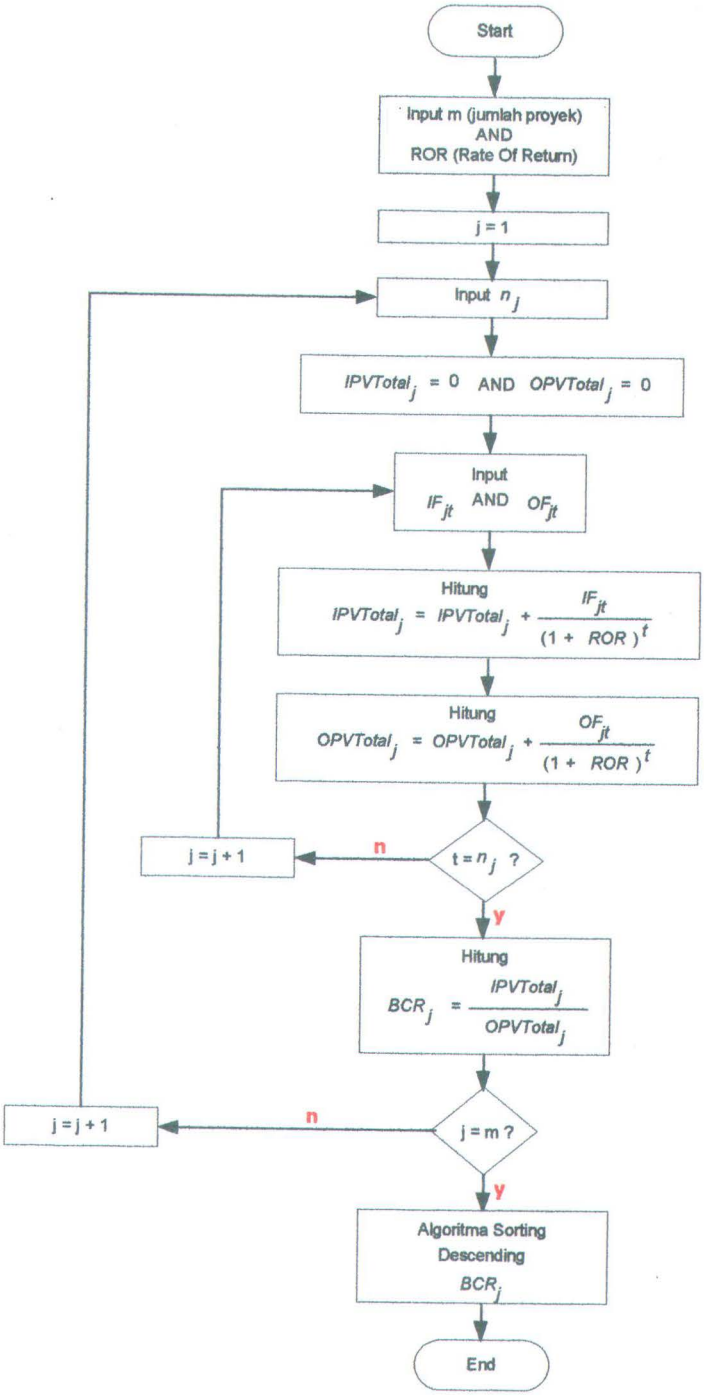
$$OPVTotal_j = \sum_{t=0}^{n} \frac{OF_{jt}}{(1+ROR)^t}$$

7) Lakukan perhitungan Benefit Cost Ratio untuk proyek j, dengan perhitungan BCRatio adalah sebagai berikut:

$$BCRatio_j = \frac{IPVTotal_j}{OPVTotal_j}$$

8) Ulangi mulai dari langkah 3 sampai dengan 7 untuk proyek berikutnya ($j = j+1$). Jika nilai j adalah m maka lanjutkan pada langkah 9.

9) Penentuan ranking dari proyek diurutkan atas nilai terbesar dari BCRatio tersebut sampai yang terkecil (*metode sorting descending*).



Gambar A.6 Flowchart Algoritma Perhitungan BCR

LAMPIRAN B

Join-Query Pembuatan Cube

- *Proyek View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage]

FROM

(TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])

- *BalanceSheet View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbBalanceSheet].[Tahun], [TbBalanceSheet].[Kas], [TbBalanceSheet].[Piutang], [TbBalanceSheet].[Persediaan], [TbBalanceSheet].[Fasilitas], [TbBalanceSheet].[AktivaLain], [TbBalanceSheet].[TotalAssets], [TbBalanceSheet].[HutangDagang], [TbBalanceSheet].[HutangBank], [TbBalanceSheet].[Modal], [TbBalanceSheet].[Saham], [TbBalanceSheet].[TotalPasiva]

FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbBalanceSheet **ON** ((([TbProyek].[IDSektor] =[TbBalanceSheet].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbBalanceSheet].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbBalanceSheet].[Program])) **AND** ([TbProyek].[Proyek] =[TbBalanceSheet].[Proyek])

- *IncomeStatement View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbIncomeStatement].[Tahun], [TbIncomeStatement].[Revenue], [TbIncomeStatement].[SalesExpenses], [TbIncomeStatement].[COGS],

[TbIncomeStatement].[GrossIncome], [TbIncomeStatement].[AddRevenue],
[TbIncomeStatement].[OperationIncome], [TbIncomeStatement].[NetIncome],
[TbIncomeStatement].[GeneralAndAdm], [TbIncomeStatement].[Depreciation],
[TbIncomeStatement].[Interest], [TbIncomeStatement].[Tax]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor]
=[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER**
JOIN TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND**
([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program]
=[TbProyek].[Program])) **INNER JOIN** TbIncomeStatement **ON** ((([TbProyek].[IDSektor]
=[TbIncomeStatement].[IDSektor]) **AND** ([TbProyek].[IDSubSektor]
=[TbIncomeStatement].[IDSubSektor])) **AND** ([TbProyek].[Program]
=[TbIncomeStatement].[Program])) **AND** ([TbProyek].[Proyek] =[TbIncomeStatement].[Proyek])

- *CashFlow View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek],
[TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek],
[TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi],
[TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif],
[TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian],
[TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan],
[TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbCashFlow].[Tahun],
[TbCashFlow].[Bulan], [TbCashFlow].[Investment], [TbCashFlow].[ChangeWorkingCapital],
[TbCashFlow].[TerminalVal], [TbCashFlow].[NetFlow]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor]
=[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER**
JOIN TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND**
([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program]
=[TbProyek].[Program])) **INNER JOIN** TbCashFlow **ON** ((([TbProyek].[IDSektor]
=[TbCashFlow].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbCashFlow].[IDSubSektor])) **AND**
([TbProyek].[Program] =[TbCashFlow].[Program])) **AND** ([TbProyek].[Proyek] =[TbCashFlow].[Proyek])

- *BenefitCost View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek],
[TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek],
[TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi],
[TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif],
[TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian],
[TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan],
[TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbBenefitCost].[Tahun],
[TbBenefitCost].[Bulan], [TbBenefitCost].[Benefit], [TbBenefitCost].[Disbenefit], [TbBenefitCost].[Cost]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor]
=[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER**
JOIN TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND**
([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program]
=[TbProyek].[Program])) **INNER JOIN** TbBenefitCost **ON** ((([TbProyek].[IDSektor]
=[TbBenefitCost].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbBenefitCost].[IDSubSektor])) **AND**
([TbProyek].[Program] =[TbBenefitCost].[Program])) **AND** ([TbProyek].[Proyek]
=[TbBenefitCost].[Proyek])

- *ROA View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai] [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbROA].[Tahun], [TbROA].[NetIncome], [TbROA].[TotalAssets], [TbROA].[ROA]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbROA **ON** ((([TbProyek].[IDSektor] =[TbROA].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbROA].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbROA].[Program])) **AND** ([TbProyek].[Proyek] =[TbROA].[Proyek])

- *ROI View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbROI].[Tahun], [TbROI].[NetIncome], [TbROI].[Investment], [TbROI].[ROI]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram ON [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbROI **ON** ((([TbProyek].[IDSektor] =[TbROI].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbROI].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbROI].[Program])) **AND** ([TbProyek].[Proyek] =[TbROI].[Proyek])

- *IRR View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbIRR].[Tahun], [TbIRR].[Bulan], [TbIRR].[UmurProyek], [TbIRR].[NF], [TbIRR].[IRR]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbIRR **ON** ((([TbProyek].[IDSektor] =[TbIRR].[IDSektor]) **AND**

([TbProyek].[IDSubSektor] =[TbIRR].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbIRR].[Program])) **AND** ([TbProyek].[Proyek] =[TbIRR].[Proyek])

- *BCR View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbBCR].[Tahun], [TbBCR].[Bulan], [TbBCR].[UmurProyek], [TbBCR].[IFt], [TbBCR].[OFt], [TbBCR].[IPVTotal], [TbBCR].[OPVTotal], [TbBCR].[BCR]
FROM
((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbBCR **ON** ((([TbProyek].[IDSektor] =[TbBCR].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbBCR].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbBCR].[Program])) **AND** ([TbProyek].[Proyek] =[TbBCR].[Proyek])

- *PBP View*

SELECT

[TbSektor].[Sektor], [TbSubSektor].[Subsektor], [TbProgram].[Program], [TbProyek].[Proyek], [TbProyek].[KelasProyek], [TbProyek].[Organisasi], [TbProyek].[Lokasi], [TbProyek].[SifatProyek], [TbProyek].[Dimulai], [TbProyek].[Selesai], [TbProyek].[Lampiran], [TbProyek].[ROR], [TbProyek].[Misi], [TbProyek].[Tujuan], [TbProyek].[Sasaran], [TbProyek].[DampakNegatif], [TbProyek].[DampakPositif], [TbProyek].[Kebijaksanaan], [TbProyek].[JumlahTKA], [TbProyek].[JumlahTKI], [TbProyek].[Uraian], [TbProyek].[Volume], [TbProyek].[Satuan], [TbProyek].[PenggunaLayanan], [TbProyek].[PopulasiLayanan], [TbProyek].[ServiceCoverage], [TbPBP].[Tahun], [TbPBP].[Bulan], [TbPBP].[UmurProyek], [TbPBP].[NF], [TbPBP].[PBP]
FROM

((TbSubSektor **INNER JOIN** (TbSektor **INNER JOIN** TbProgram **ON** [TbSektor].[IDSektor] =[TbProgram].[IDSektor]) **ON** [TbSubSektor].[IDSubSektor] =[TbProgram].[IDSubSektor]) **INNER JOIN** TbProyek **ON** (([TbProgram].[IDSektor] =[TbProyek].[IDSektor]) **AND** ([TbProgram].[IDSubSektor] =[TbProyek].[IDSubSektor])) **AND** ([TbProgram].[Program] =[TbProyek].[Program])) **INNER JOIN** TbPBP **ON** ((([TbProyek].[IDSektor] =[sTbPBP].[IDSektor]) **AND** ([TbProyek].[IDSubSektor] =[TbPBP].[IDSubSektor])) **AND** ([TbProyek].[Program] =[TbPBP].[Program])) **AND** ([TbProyek].[Proyek] =[TbPBP].[Proyek])

LAMPIRAN C

Daftar Isian Sektor/Subsektor/Program Proyek Pembangunan

Tabel C.1 Daftar Isian Sektor/Subsektor/Program Proyek Pembangunan

Kode	*Sektor/Subsektor/Program*
01	**Industri**
01.1	*Industri*
01.1.01	Pengembangan industri rumah tangga industri kecil dan menengah

01.1.02	Peningkatan kemampuan teknologi
01.1.03	Penataan struktur industri
02	**Pertanian dan Kehutanan**
02.1	*Pertanian rakyat*
02.1.01	Pembangunan pertanian rakyat terpadu
02.1.02	Pengembangan usaha tani
02.1.03	Diversifikasi pangan dan gizi
02.1.04	Pengembangan sumber daya, sarana dan prasarana pertanian
02.2	*Perkebunan*
02.2.01	Pembangunan perkebunan rakyat terpadu
02.2.02	Pembangunan usaha perkebunan
02.2.03	Diversifikasi pangan dan gizi
02.2.04	Pengembangan sumber daya, sarana dan prasarana perkebunan
02.3	*Peternakan*
02.3.01	Pembangunan Peternakan rakyat terpadu
02.3.02	Pembangunan usaha Peternakan
02.3.03	Diversifikasi pangan dan gizi
02.3.04	Pengembangan sumber daya, sarana dan prasarana Peternakan
02.4	*Perikanan*
02.4.01	Pembangunan Perikanan rakyat terpadu
02.4.02	Pembangunan usaha Perikanan
02.4.03	Diversifikasi pangan dan gizi
02.4.04	Pengembangan sumber daya, sarana dan prasarana Perikanan
02.5	*Kehutanan*
02.5.01	Pembangunan dan pembinaan kehutanan
02.5.02	Pengembangan usaha hutan rakyat
03	**Sumberdaya air dan irigrasi**
03.1	*Pengembangan sumber daya air*
03.1.01	Pengembangan dan konservasi Sumber daya air
03.1.02	Penyediaan dan pengelolaan air baku
03.1.03	Pengelolaan sungai, danau, dan sumber air lainnya
03.2	*Irigrasi*
03.2.01	Pengembangan dan pengelolaan jaringan irigrasi
04	**Tenaga Kerja**
04.1	*Tenaga kerja*
04.1.01	Pelatihan dan peningkatan Ketrampilan tenaga kerja
04.1.02	Penyebaran dan pendayagunaan tenaga kerja
04.1.03	Pembinaan dan pengembangan produktivitas dan kesempatan kerja
04.1.04	Pembinaan hubungan industrial dan perlindungan tenaga kerja
05	**Perdagangan, Pengembangan usaha daerah, keuangan daerah dan koperasi**

05.1	*Perdagangan*
05.1.01	Pengembangan perdagangan dan sistem distribusi
05.1.02	Pengembangan Usaha dan Lembaga Perdagangan
05.1.03	Pengembangan Kerjasama Perdagangan Internasional
05.1.04	Pengembangan Ekspor
05.2	*Pengembangan Usaha Daerah*
05.2.01	Pengembangan usaha golongan ekonomi lemah
05.2.02	Pengembangan penanaman modal
05.2.03	Penyertaan modal pemerintah
05.3	*Keuangan daerah*
05.3.01	Penerimaan pendapatan daerah
05.3.02	Pengembangan lembaga keuangan
05.3.03	Pembinaan kekayaan daerah
05.3.04	Pembinaan perusahaan-perusahaan daerah
05.4	*Koperasi dan Pengusaha Kecil*
05.4.01	Pembinaan dan Pengembangan Koperasi
05.4.02	Pembinaan dan Pengembangan Usaha Kecil
06	**Transportasi**
06.1	*Prasarana Jalan*
06.1.01	Rehabilitasi dan Pemeliharaan Jalan dan Jembatan
06.1.02	Peningkatan jalan dan penggantian jembatan
06.1.03	Pembangunan jalan dan jembatan
06.2	*Transportasi Darat*
06.2.01	Pengembangan fasilitas lalu lintas jalan
06.2.02	Peningkatan angkutan sungai, danau, dan penyeberangan
06.2.03	Pengembangan sistem transportasi
07	**Pertambangan dan Energi**
07.1	*Pertambangan rakyat*
07.1.01	Pembangunan pertambangan
07.1.02	Pembangunan Sistem Informasi Geologi/Geografi
07.2	*Energi/Listrik Pedesaan*
07.2.01	Pengembangan Tenaga Listrik
07.2.02	Pengembangan Listrik Pedesaan
07.2.03	Pengembangan Tenaga Migas, batubara dan energi lainnya
08	**Pariwisata dan Telekomunikasi Daerah**
08.1	*Pariwisata*
08.1.01	Pengembangan Pariwisata
08.1.02	Pengembangan dan Pengelolaan produk/obyek wisata
08.2	*Telekomunikasi Daerah*
08.2.01	Pengembangan telekomunikasi daerah

09	**Pembangunan daerah dan pemukiman**
09.1	*Pembangunan daerah bawahan*
09.1.01	Pembangunan daerah tingkat II
09.1.02	Pembangunan kawasan khusus
09.2	*Pembangunan kota*
09.2.01	Pembangunan prasarana kota
09.3	*Pembangunan desa*
09.3.01	Pembangunan desa
09.4	*Pemukiman*
09.4.01	Pemukiman dan Lingkungan
09.4.01	Pengerahan dan Pembinaan Transmigran
10	**Lingkungan Hidup dan Tata ruang**
10.1	*Lingkungan Hidup*
10.1.01	Pembinaan daerah pantai
10.1.02	Pembinaan dan pengelolaan lingkungan hidup
10.1.03	Penyelamatan hutan, tanah dan air
10.1.04	Pengendalian pencemaran lingkungan
10.1.05	Inventarisasi dan evaluasi sumber daya alam
10.1.06	Rehabilitasi lahan kritis
10.2	*Tata Ruang*
10.2.01	Penataan ruang
10.2.02	Pertanahan
11	**Pendidikan kebudayaan nasional, Kepercayaan terhadap Tuhan YME, Pemuda dan Olahraga**
11.1	*Pendidikan*
11.1.01	Pembinaan pendidikan dasar
11.1.02	Pembinaan pendidikan menengah
11.1.03	Pembinaan pendidikan tinggi
11.1.04	Pembinaan tenaga kependidikan dan kebudayaan
11.1.05	Operasi dan perawatan fasilitas pendidikan dan kebudayaan
11.2	*Pendidikan luar sekolah dan kedinasan*
11.2.01	Luar sekolah
11.3	*Kebudayaan daerah dan kepercayaan terhadap Tuhan Yang Maha Esa*
11.3.01	Inventarisasi dan pembinaan nilai-nilai budaya
11.3.02	Pembinaan kebahasaan, kesusateraan dan kepustakaan
11.3.03	Pembinaan kesenian daerah
11.3.04	Pembinaan tradisi, peninggalan sejarah dan museum
11.3.05	Pembinaan penganut kepercayaan terhadap Tuhan Yang Maha Esa
11.4	*Pemuda dan Olah Raga*
11.4.01	Pembinaan dan pengembangan kepemudaan
11.4.02	Pembinaan Keolahragaan
12	**Kependudukan dan keluarga sejahtera**

12.1	*Kependudukan dan keluarga berencana*
12.1.01	Kependudukan
12.1.02	Keluarga berencana
13	**Kesehatan, kesejahteraan sosial, peranan wanita, anak dan remaja**
13.1	*Kesehatan*
13.1.01	Penyuluhan kesehatan dan perbaikan gizi masyarakat
13.1.02	Pelayanan kesehatan rujukan dan rumah sakit
13.1.03	Pelayanan kesehatan masyarakat
13.1.04	Pencegahan dan pemberantasan penyakit
13.1.05	Pengawasan obat dan makanan
13.1.06	Pembinaan pengobatan tradisional
13.2	*Kesejahteraan sosial*
13.2.01	Pembinaan kesejahteraan sosisal
13.2.02	Pelayanan dan rehabilitasi sosial
13.2.03	Pembinaan partisipasi sosial masyarakat
13.2.04	Penanggulangan bencana alam
13.3	*Peranan wanita, anak dan remaja*
13.3.01	Peningkatan peranan wanita dan PKK
13.3.02	Pembinaan anak dan remaja
14	**Perumahan dan pemukiman**
14.1	*Perumahan dan pemukiman*
14.1.01	Penyediaan Perumahan dan pemukiman
14.1.02	Perbaikan Perumahan dan pemukiman
14.1.03	Penyehatan lingkungan pemukiman
14.1.04	Penyediaan dan pengelolaan air bersih
14.2	*Penataan kota dan bangunan*
14.2.01	Penataan Kota
14.2.02	Penataan Bangunan
15	**Agama**
15.1	*Pelayanan kehidupan beragama*
15.1.01	Peningkatan sarana kehidupan beragama
15.1.02	Penerangan bimbingan dan Kerukunan hidup beragama
15.1.03	Peningkatan pelayanan ibadah haji
15.2	*Pembinaan pendidikan agama*
15.2.01	Pembinaan pendidikan agama
16	**Ilmu pengetahuan dan teknologi**
16.1	*Ilmu pengetahuan dan teknologi*
16.1.01	Pengkajian dan penelitian ilmu pengetahuan dasar
16.1.02	Pengkajian dan penelitian ilmu pengetahuan terapan
16.1.03	Pengembangan prasarana dan sarana ilmu pengetahuan dan teknologi

16.2	*Tenik produksi dan teknologi*
16.2.01	Penerapan dan pengembangan teknik produksi
16.2.02	Penguasaan alih teknologi
16.2.03	Penerapan dan pengembangan model-model pembnagunan di daerah
16.3	*Penelitian*
16.3.01	Penelitian pendukung perencanaan pembangunan
16.3.02	Perancangan pembangunan daerah
16.3.03	Pembinaan kelembagaan ilmu pengetahuan dan teknologi
16.4	*Kelautan*
16.4.01	Inventarisasi dan Evaluasi Potensi kelautan
16.4.02	Pemanfaatan sumber daya kelautan
16.5	*Sistem Informasi dan statistik*
16.5.01	Pengembangan sistem informasi
16.5.02	Penyempurnaan dan pengembangan statistik
17	**Hukum**
17.1	*Hukum*
17.1.01	Perencanaan dan pembentukan Hukum
17.1.02	Pengembangan jaringan informasi dan dokumentasi Hukum di daerah
17.1.03	Penerapan dan penegakan Hukum
17.1.04	Penyuluhan Hukum
17.1.05	Pembinaan sarana dan prasarana Hukum
18	**Aparatur pemerintah dan pengawasan**
18.1	*Aparatur pemerintah*
18.1.01	Peningkatan sarana dan prasarana aparatur pemerintah
18.1.02	Peningkatan efisiensi dan pendayagunaan aparatur pemerintah
18.1.03	Pendidikan dan pelatihan aparatur pemerintahan
18.1.04	Penelitian dan pengkajian kebijaksanaan
18.2	*Pendayagunaan sistem dan pelaksanaan pengawasan*
18.2.01	Pendayagunaan sistem pengawasan
18.2.02	Peningkatan efektivitas pelaksanaan pengawasan
19	**Politik, penerangan, komunikasi dan media massa**
19.1	*Politik*
19.1.01	Pembinaan politik di daerah
19.1.02	Penyelenggaraan dan peningkatan kualitas otonomi daerah
19.1.03	Pembinaan dan pengembangan kerjasama luar negeri
19.2	*Penerangan, komunikasi dan media massa*
19.2.01	Pengembangan operasi penerangan
19.2.02	Pembinaan dan pengembangan radio, televisi dan film di daerah
19.2.03	Pembinaan dan pengembangan pers di daerah
20	**Keamanan dan ketertiban hukum**

20.1	*Keamanan dan ketertiban hukum*
20.1.01	Pembinaan kekuatan rakyat dan bela negara
20.1.02	Pembinaan ketentraman dan ketertiban
A	**Bantuan pembangunan kepada daerah bawahan**
A.1	*Bantuan pembangunan Dati I*
A.1.01	Subsidi Bantuan pembangunan Dati I

31000000 11390

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH NOPEMBER

TUGAS AKHIR
OE1701

STUDI EKSPERIMENTAL PENINGKATAN KEKUATAN FATIGUE BUTT JOINT PADA MATERIAL ASTM A516 GRADE 70 DENGAN TIG DRESSING

RSke
G20.1126
Jay
S-1
1998

PERPUSTAKAAN
I T S
Tgl. Terima 3-3-99
No. Agenda Prp. 0374

Disusun Oleh :

DANI IVAN JAYA
4392 100 027

JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1998

STUDI EKSPERIMENTAL PENINGKATAN KEKUATAN FATIGUE BUTT JOINT PADA MATERIAL ASTM A516 GRADE 70 DENGAN TIG DRESSING

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Kelautan
Pada
Jurusan Teknik Kelautan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
(Ir. Jusuf Sutomo MSc.)	(Ir. Soeweify MEng.)

SURABAYA
1998

Untuk seluruh Keluargaku

Ayah, Ibu, Eyang Putri dan Adik-adikku Eric, Pitha

Yang telah mengikatku erat dengan kasih sayang.

" We make a living by what we get
but we make a life by what we give."

Winston Curchill

ABSTRAK

Fatigue merupakan proses pengurangan kekuatan pada komponen struktur karena pembebanan berulang, bila pengulangan beban cukup besar maka dapat menimbulkan keretakan dan akan terus menjalar hingga terjadi kegagalan pada komponen tersebut. Pada struktur yang penyambungannya menggunakan teknologi pengelasan, daerah las merupakan daerah potensial terjadinya retak awal karena pembebanan berulang, sambungan las menimbulkan konsentrasi tegangan sehingga terjadi tegangan lokal yang harganya jauh lebih tinggi dari tegangan rata-rata yang diterapkan pada struktur. Konsentrasi tegangan ini yang memicu keretakan awal hingga menimbulkan kegagalan fatigue pada struktur. Studi ini mempelajari cara untuk mengurangi konsentrasi tegangan yang terjadi pada sambungan las tumpul sehingga kekuatan fatigue-nya dapat ditingkatkan. Pengurangan konsentrasi tegangan dilakukan dengan memodifikasi geometri manik las dengan tujuan untuk menurunkan faktor konsentrasi tegangan yang terjadi. Pada spesimen dilakukan perlakuan TIG Dressing single heating pass dan double heating pass yang kemudian diuji dengan mesin uji fatigue untuk mengetahui umur fatigue dan umur keretakan awalnya sehingga dapat dibandingkan dengan hasil pengujian pada spesimen as welded. Hasil pengujian menunjukkan bahwa perlakuan TIG Dressing dapat menurunkan faktor konsentrasi tegangan dan meningkatkan kekuatan fatigue spesimen, teknik single heating pass dapat meningkatkan kekuatan fatigue 32,85 % dari kondisi as welded sedangkan teknik double heating pass hingga 102,86 %. Kedua teknik perlakuan TIG ini terbukti dapat meningkatkan kekuatan fatigue spesimen, teknik double heating pass dapat meningkatkan kekuatan fatigue lebih besar akan tetapi dari segi proses pelaksanaan lebih lama karena dibutuhkan dua kali proses untuk satu sambungan sedangkan teknik single heating pass hanya membutuhkan satu kali proses untuk setiap sambungan, sehingga lebih singkat dan lebih ekonomis.

KATA PENGANTAR

Alhamdulillahirrobil'alamin. Segala puji hanya kepada Allah SWT, Tuhan sesembahan seluruh alam semesta. Atas karunia-Nya penulisan Laporan Tugas Akhir dengan judul ***"Studi Eksperimental Peningkatan Kekuatan Fatigue Butt Joint Pada Material ASTM A 516 Grade 70 Dengan TIG Dessing"***, ini dapat penulis selesaikan. Semua itu juga tidak lepas dari bantuan dan bimbingan Bapak Ir. Jusuf Sutomo MSc. dan Bapak Ir. Soeweify MEng. selaku dosen pembimbing. Ucapan terima kasih juga penulis sampaikan kepada Bapak Dr. Ir. Eko Budi Djatmiko bersama seluruh staff pengajar di Jurusan Teknik Kelautan atas bantuannya kepada penulis.

Laporan tugas akhir ini disusun untuk melengkapi pengerjaan Mata Kuliah Tugas Akhir (OE 1701) di Jurusan Teknik Kelautan, Fakultas Teknologi Kelautan, ITS. Dalam laporan ini dijelaskan eksperimen untuk meningkatkan kekuatan fatigue pada sambungan las tumpul dengan menggunakan teknik perlakuan TIG Dressing, efisiensi dari teknik yang digunakan dan perbandingan antara hasil eksperimen dengan hasil perhitungan teoritis.

Pada akhirnya penulis hanya bisa berharap bahwa hasil studi ini dapat bermanfaat dan berguna, baik bagi penulis pribadi maupun pembaca sekalian.

Surabaya, 16 Maret 1997

Penulis

UCAPAN TERIMA KASIH

Segala usaha, bantuan dan peran serta banyak pihak dalam penyelesaian Tugas Akhir ini adalah sesuatu yang berharga bagi penulis disamping karunia, rahmat dan pertolongan Allah SWT. Hasil karya ini terselesaikan atas bantuan dan peran serta yang tak terhingga. Untuk itu dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih dan penghargaan yang setulusnya kepada :

1. Ayahanda Budi Supomo dan Ibunda Sriningsih Ekosari atas doa dan limpahan kasih yang tak terhingga kepada penulis.
2. Bapak Ir. Jusuf Sutomo MSc. selaku dosen pembimbing Tugas Akhir ini atas bimbingan dan petunjuk yang telah beliau berikan.
3. Bapak Ir. Soeweify MEng. selaku dosen pembimbing Tugas Akhir ini atas bantuan dan bimbingan beliau. Banyak hal berguna yang dapat penulis pelajari dari saran dan petunjuk yang beliau sampaikan.
4. Bapak Ir. Agus Suprayitno beserta seluruh staff Departemen Konstruksi, General Engineering PT. PAL Surabaya, yang telah banyak memberikan bantuan kepada penulis.
5. Bapak E.N. Kristanto dan seluruh staff PT. Schneider Indonesia atas bantuan fasilitas dan petunjuk-petunjuknya.
6. Bapak Dr. Ir. Eko Budi Djatmiko selaku Ketua Jurusan Teknik Kelautan.
7. Bapak Dr. Ir. W.A. Pratikto MSc. selaku dosen wali selama penulis menempuh studi di Jurusan Teknik Kelautan, atas pengarahan dan bimbingan beliau selama ini.
8. Seluruh dosen pengajar Jurusan Teknik Kelautan atas seluruh ilmu dan pengetahuan yang dapat penulis pelajari.
9. Seluruh karyawan Jurusan Teknik Kelautan atas bantuan dan segala fasilitas yang telah diberikan.

10. Pak Hardiman, Mas Yanto, Mas Didik dan seluruh karyawan Lab. Produksi, Lab. Teknologi Mekanik dan Lab. Konstruksi FTK-ITS.
11. Rekan-rekan Jurusan Teknik Kelautan Angkatan '92, Arifin, Hasbi, Rifai, Arik, Taufik, Adi, Fauzi, Zaini, Joe, Satya, Dwik dan semuanya atas segala solidaritas dan persahabatan kita selama ini.
12. Kakak dan adik kelasku, Mas Edi, PakDhe Heri, Budi Gimbun, Iwan, Ayik, Wiwid, Micky, Momo, Heni dan Carchi atas bantuan dan dorongan kepada penulis untuk tetap bersemangat menghadapi segala kendala yang ada.
13. Anas dan Deni 'You are my best friends', juga seluruh penghuni Hard Rock Café, Ipunk, Bayu, Feri, Sam, Sahab, Tofik-wendys, Ipok, Zoel dan semuanya atas kekompakan dan keceriaan yang kalian tawarkan.
14. Untuk 'yang terindah' Endyah Pratiwi N. atas spirit dan inspirasi yang selalu kau sediakan bagiku.
15. Dan kepada pihak-pihak yang dengan tidak mengurangi rasa hormat tidak dapat penulis sebutkan satu-persatu dalam lembar ini.

Tiada yang pantas kami ucapkan, sekali lagi adalah beribu terima kasih atas segala yang telah terbantukan kepada penulis selama ini.

DAFTAR ISI

Lembar pengesahan	
Abstrak	
Kata pengantar	
Daftar Isi	i
Daftar Tabel	iv
Daftar Gambar	v
Daftar Notasi	vi
BAB I PENDAHULUAN	
1.1. Latar Belakang	I-1
1.2. Batasan Masalah	I-6
1.3. Tujuan Penulisan	I-7
1.4. Manfaat Penulisan	I-7
1.5. Metodologi Pemecahan Masalah	I-7
BAB II PENGELASAN DAN SIFAT KELELAHAN SAMBUNGAN LAS	
2.1. Baja Dalam Pengelasan	II-1
2.1.1. Struktur Mikro Dan Sifat Mekanis	II-1
2.1.2. Siklus Thermal Daerah Las	II-4
2.1.3. Baja ASTM A 516 grade 70	II-10
2.2. Butt Joint Pada Pengelasan Baja	II-11
2.3. Kelelahan Pada Sambungan Las	II-13
2.3.1. Pengaruh Bentuk Pada Kekuatan Fatigue	II-13
2.3.2. Peningkatan Kekuatan Fatigue	II-15

BAB III KELELAHAN DAN KEPATAHAN

3.1. Konsentrasi Tegangan III-1

3.2. Aturan Neuber Tentang Konsentrasi Tegangan III-5

3.2.1. Konsentrasi Tegangan III-5

3.2.2. Hubungan Tegangan-Regangan III-8

3.2.3. Kurva D-Nc III-10

3.3. Kegagalan Fatigue III-11

3.3.1. Mekanisme Kelelahan III-14

3.3.2. Kharakteristik Kepatahan III-21

BAB IV PROSEDUR DAN PELAKSANAAN PENGUJIAN

4.1. Persiapan Material Uji IV-1

4.1.1. Material IV-1

4.1.2. Pengelasan GMAW IV-2

4.1.3. Pengujian Radiografi IV-6

4.1.4. Perlakuan TIG Dressing IV-7

4.1.5. Persiapan Akhir Material Uji IV-10

4.2. Pengukuran Flank Angle Dan Jari-Jari Toe IV-12

4.3. Pengujian Fatigue IV-15

4.3.1. Kalibrasi Beban IV-15

4.3.2. Penentuan Beban IV-17

BAB V ANALISA DATA HASIL PENGUJIAN

5.1. Perhitungan Faktor Konsentrasi Tegangan V-1

5.2. Perhitungan Umur Keretakan Awal V-6

5.3. Kurva S-N V-19

5.4. Pembahasan Hasil Pengujian V-21

BAB VI PENUTUP

6.1. Kesimpulan VI-1

6.2. Saran VI-2

DAFTAR PUSTAKA

LAMPIRAN-A

LAMPIRAN-B

DAFTAR TABEL

Tabel 3.1. Konstanta material, definisi dan perkiraan

Tabel 4.1. Sifat mekanis dan komposisi kimia baja ASTM A 516 grade 70

Tabel 4.2. Komposisi kimia elektoda AWS A5 18ER70S-6

Tabel 4.3. Welding parameter

Tabel 4.4. Welding parameter TIG Dressing

Tabel 4.5. Data pengukuran flank angle

Tabel 4.6. Data pengukuran jari-jari toe

Tabel 4.7. Kalibrasi beban mesin uji dinamis

Tabel 5.1. Faktor konsentrasi tegangan pada spesimen as welded

Tabel 5.2. Faktor konsentrasi tegangan pada spesimen double heating pass

Tabel 5.3. Faktor konsentrasi tegangan pada spesimen single heating pass

Tabel 5.4. Perhitungan fungsi tegangan-regangan

Tabel 5.5. Posisi keretakan awal dan faktor konsentrasi tegangan spesimen

Tabel 5.6. Perhitungan kurva S-Nc

Tabel 5.7. Umur fatigue spesimen

Tabel 5.8. Perbandingan umur keretakan awal prediksi dengan hasil pengujian

Tabel 5.9. Perbandingan kekuatan fatigue spesimen

Tabel 5.10. Perbandingan teknik Double Heating Pass dan Single Heating Pass

DAFTAR GAMBAR

Gambar 1.1. Diagram alur pelaksanaan eksperimen.

Gambar 2.1. Diagram CCT baja ASTM 4340

Gambar 2.2. Siklus Thermal las pada beberapa jarak dari batas las.

Gambar 2.3. Daerah dan besar butir logam hasil pengelasan.

Gambar 2.4. Hubungan antara kekerasan maksimum pada daerah HAZ dengan kadar karbon dalam baja.

Gambar 2.5. Hubungan jenis sambungan dengan aliran tegangan.

Gambar 2.6. Hubungan antara sudut kemiringan sisi manik las dengan kekuatan fatigue sambungan tumpul.

Gambar 2.7. Klasifikasi metode peningkatan kekuatan fatigue.

Gambar 2.8. TIG Dressing pada titik toe

Gambar 3.1. Aliran gaya-gaya pada pelat yang terbebani.

Gambar 3.2. Konsentrasi tegangan pada pelat berlubang.

Gambar 3.3. Geometri sambungan las.

Gambar 3.4. Slip yang terjadi pada pembebanan statis (material ductile).

Gambar 3.5. Slip yang terjadi pada pembebanan siklis (material ductile).

Gambar 3.6. Skema tahap I dan II penjalaran retak transkristalin.

Gambar 3.7. Penajaman dan penumpulan ujung retak.

Gambar 3.8. Skema proses kelelahan.

Gambar 3.9. Dua pembagian pada permukaan kepatahan fatigue.

Gambar 4.1. Permukaan hasil las dengan berbagai gas pelindung.

Gambar 4.2. Persiapan sisi dan urutan pada pengelasan.

Gambar 4.3. Perbedaan teknik perlakuan TIG.

Gambar 4.4. Bentuk manik las hasil perlakuan TIG.
Gambar 4.5. Dimensi specimen uji fatigue.
Gambar 4.6. Proses untuk melokalisasi daerah keretakan awal.
Gambar 4.7. Geometri sambungan las.
Gambar 4.8. Grafik kalibrasi beban mesin uji dinamis.
Gambar 5.1. Kurva D-Nc baja ASTM A 516 grade 70
Gambar 5.2. Kurva S-Nc spesimen as welded, single heating pass dan double heating pass
Gambar 5.3. Kurva S-Nc spesimen as welded.
Gambar 5.4. Kurva S-Nc spesimen double heating pass
Gambar 5.5. Kurva S-Nc spesimen single heating pass
Gambar 5.6. Kurva S-N Untuk Spesimen as welded, single heating pass dan double heating pass

DAFTAR NOTASI

a = eksponen kekuatan fatigue

A_r = luas penampang rata-rata

b = eksponen regangan fatigue

B = lebar manik las

d = diameter lubang

D = fungsi tegangan-regangan

E = modulus elastisitas (Young)

$F_{app.}$ = gaya yang diterapkan pada spesimen

H = tinggi manik las

K = faktor konsentrasi tegangan

K_e = faktor konsentrasi tegangan efektif

K_t = faktor konsentrasi tegangan elastis (teoritis)

K_ε = faktor konsentrasi regangan plastis

K_σ = faktor konsentrasi tegangan plastis

n = cyclic strain hardening exponent

N_c = umur keretakan awal

N_f = umur fatigue

P = gaya yang diterapkan pada material

R = jari-jari toe

T = tebal pelat

W = lebar pelat

Δ_e = rentang regangan nominal

Δ_ε = rentang regangan lokal yang sebenarnya

Δ_s = rentang tegangan nominal

Δ_σ = rentang tegangan lokal yang sebenarnya

ε_a = regangan total

ε_a^e = regangan elastis

ε_a^p = regangan plastis

ε_f = regangan fracture statis

ε'_f = koefisien regangan fatigue

σ = tegangan normal

σ_f = kekuatan fracture statis

σ'_f = koefisien tegangan fatigue

σ_{mak} = tegangan maksimal (lokal)

σ_r = tegangan normal rata-rata

$\sigma_{ult.}$ = kekuatan ultimate statis

θ = flank angle

BAB 1

PENDAHULUAN

BAB I
PENDAHULUAN

1.1. LATAR BELAKANG

Pada masa sekarang ini penggunaan logam khususnya baja dalam pembuatan struktur sudah sedemikian luas, sehingga hampir tidak ada bangunan yang tidak ditopang atau menggunakan komponen baja. Hal ini dapat kita lihat pada berbagai struktur yang ada pada saat sekarang ini seperti jembatan, mobil, kereta api, kapal, bangunan lepas pantai, gedung bertingkat dan sebagainya. Salah satu keunggulan penggunaan baja adalah kekuatannya sangat tinggi bila dibandingkan dengan material yang lain dengan dimensi yang sama.

Dengan semakin luasnya penggunan baja maka makin banyak pula ditemukan permasalahan yang berkaitan dengan penggunaannya dalam struktur. Banyak studi dilakukan untuk mempelajari karakteristik logam ini dengan tujuan untuk memperluas penggunaannya dalam dunia teknik dan industri. Keunggulan dan kelemahan baja dipelajari sehingga dapat digunakan sebagi bahan pertimbangan dalam perancangan struktur.

Salah satu sifat baja yang penting adalah perilakunya tehadap beban yang diterapkan. Hubungan antara kekuatan material baja dengan beban merupakan faktor penting dalam perancangan struktur. Konsep utamanya adalah kekuatan material harus mampu menopang beban yang diterapkan sesuai dengan umur struktur yang direncanakan. Pada umumnya kekuatan material diukur dengan parameter kekuatan berdasarkan beban statis, sedangkan dalam pengoperasiannya material struktur akan lebih cenderung untuk menerima beban dinamis, yaitu beban yang bekerja secara berulang-ulang.

Kondisi seperti ini harus benar-benar diperhitungkan dalam perancangan struktur. Penerapan beban secara berulang-ulang akan menurunkan kekuatan bahkan dalam jumlah pengulangan yang besar akan menimbulkan kepatahan pada material, walaupun beban yang diterapkan lebih kecil dari kekuatan statis material tersebut. Fenomena seperti ini yang disebut dengan *fatigue* atau kelelahan pada komponen struktur karena pembebanan berulang.

Geometri material ternyata juga mempunyai kontribusi yang besar pada kekuatan material terhadap beban berulang. Perubahan bentuk pada material secara tiba-tiba akan menimbulkan konsentrasi tegangan, sehingga pada area tersebut akan timbul tegangan lokal yang harganya jauh lebih besar dari pada tegangan

rata-rata yang bekerja. Tegangan lokal inilah yang akan memicu timbulnya keretakan pada komponen, dengan adanya beban berulang keretakan tersebut akan menjalar dengan cepat hingga terjadi kepecahan pada komponen struktur tersebut.

Dalam proses pembuatan struktur baja, penyambungan antar komponen biasanya dilakukan dengan pengelasan. Teknik penyambungan ini dipilih karena beberapa keuntungan, diantaranya adalah karena mempunyai kekuatan yang hampir sama dengan material yang disambung, pelaksanaannya relatif mudah dan ekonomis. Namun demikian pengelasan juga mempunyai berbagai kelemahan, diantaranya yang paling utama adalah timbulnya cacat dan menyebabkan konsentrasi tegangan pada daerah sambungan tersebut.

Daerah sambungan las merupakan daerah potensial timbulnya konsentrasi tegangan, disamping perubahan bentuk karena adanya manik las juga karena cacat las yang timbul seperti *undercut, slag inclusion* dan sebagainya. Secara umum pada daerah pertemuan antara logam las dan logam induk, yang dinamakan titik *toe*, merupakan daerah timbulnya konsentrasi tegangan. Pada titik ini beban nominal yang diterapkan akan berlipat oleh adanya faktor konsentrasi tegangan hingga mencapai suatu harga tegangan maksimum yang dapat memicu timbulnya keretakan awal.

Apabila struktur tersebut menerima beban dinamis atau beban berulang maka keretakan awal yang telah timbul akan terus menjalar sejalan dengan waktu. Semakin besar penjalaran retak semakin kecil kekuatan yang tersisa, proses ini akan terus berlanjut hingga mencapai suatu kondisi dimana kekuatan material tidak sanggup lagi menahan beban yang diterapkan, maka struktur akan pecah. Kegagalan seperti ini yang disebut dengan kegagalan fatigue.

Berbagai penelitian telah banyak dilakukan untuk mengantisipasi kegagalan fatigue pada daerah lasan. Karena konsentrasi tegangan merupakan salah satu penyebab kegagalan ini maka terus dicari cara untuk mengurangi konsentrasi tegangan, diantara yang telah ditemukan adalah sebagai berikut, [Zaczek, 1984] :

1. Menghilangkan seluruh manik las sehingga rata dengan permukaan logam induk.
2. Pemilihan kondisi pengelasan dan *filler* untuk meningkatkan *flank angle* (sudut kemiringan manik las) dan jari-jari toe.
3. Peningkatan flank angle dan jari-jari toe dengan penggerindaan.

Metoda-metoda tersebut diterapkan dengan tujuan untuk mengurangi konsentrasi tegangan pada titik toe, karena semakin

besar flank angle dan jari-jari toe maka konsentrasi tegangan akan semakin kecil.

Masih ada satu metoda lagi untuk mengurangi konsentrasi tegangan pada sambungan las yaitu dengan proses TIG *Dressing*. Metoda ini mempunyai tujuan yang hampir sama yaitu meningkatkan flank angle dan jari-jari toe, dengan menggunakan proses pengelasan TIG (GTAW) tanpa filler. Panas yang timbul digunakan untuk mencairkan manik las dan membentuknya untuk memperbesar flank angle dan jari-jari toe. Ada dua teknik dalam perlakuan TIG ini, pertama teknik konvensional dengan melakukan pencairan pada kedua sisi manik las secara bertahap *(double heating pass)* sedangkan teknik yang kedua dilakukan dengan satu kali tahap pencairan *(single heating pass)* pada seluruh permukaan manik las.

Penelitian ini akan membahas perlakuan TIG dengan kedua teknik tersebut untuk membandingkan efisiensi dan pengaruhnya terhadap kekuatan fatigue, untuk itu akan dilakukan serangkaian percobaan pada sambungan las tumpul *(butt joint)* baja ASTM A 516 grade 70 yang diberikan perlakuan TIG dan kemudian diuji dengan mesin uji fatigue untuk mengetahui seberapa besar peningkatan kekuatannya terhadap beban dinamis.

Baja karbon ASTM A 516 grade 70 yang digunakan dalam percobaan ini merupakan baja yang mempunyai kekuatan tarik tinggi dan keras akan tetapi bersifat getas *(brittle)* sehingga bersifat sensitif terhadap retak. Pengelasan dilakukan dengan proses GMAW semiotomatis menggunakan gas pelindung campuran Argon dan CO_2 dan dilakukan dalam kondisi yang terkontrol sehingga diharapkan diperoleh hasil pengelasan yang tanpa cacat. Hal ini diutamakan karena tinjauan bahwa keretakan yang timbul adalah karena konsentrasi tegangan dan bukan karena cacat las.

1.2. BATASAN MASALAH

Untuk memfokuskan pada pembahasan masalah yang dipilih maka diterapkan beberapa pembatasan sebagai berikut :

- Pembahasan dilakukan pada sambungan las tumpul.
- Material yang digunakan adalah baja ASTM A 516 grade 70 dengan ketebalan 8 mm.
- Keretakan yang timbul adalah karena konsentrasi tegangan dan diasumsikan tidak ada cacat pengelasan.
- Pengelasan dilakukan dengan GMAW semiotomatis dan perlakuan TIG dilakukan secara manual.
- Pembebanan dinamis dilakukan secara konstan.

1.3. TUJUAN PENULISAN

- Untuk mengetahui pengaruh perlakuan TIG Dressing terhadap kekuatan fatigue sambungan las tumpul.
- Membandingkan efisiensi perlakuan TIG Dressing, teknik konvensional dengan teknik single heating pass.
- Mengetahui kharakteristik fatigue pada material ASTM A 516 grade 70.

1.4. MANFAAT PENULISAN

Dapat diketahui efektifitas metoda TIG Dressing untuk meningkatkan kekuatan fatigue sambungan las tumpul pada material ASTM A 516 grade 70.

1.5. METODOLOGI PEMECAHAN MASALAH

Untuk menyelesaikan permasalahan dalam penulisan ini digunakan metoda analisis dan eksperimen. Analisis didasarkan pada studi literatur digunakan untuk :

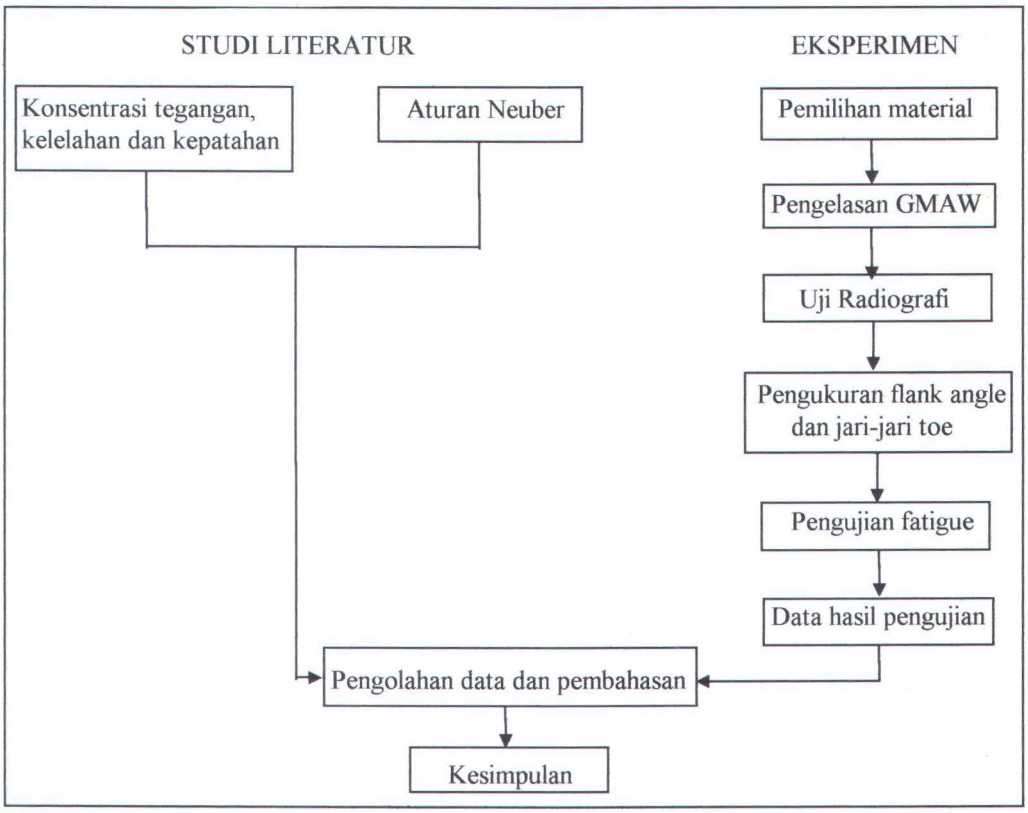
1. Memahami konsep kelelahan dan konsentrasi tegangan serta pengaruh yang ditimbulkannya pada material baja.
2. Mengetahui pengaruh pengelasan terhadap perubahan struktur mikro dan sifat mekanis baja.

3. Mengetahui pengaruh bentuk sambungan las terhadap kekuatan fatigue komponen struktur.

Sedangkan metode eksperimen dilakukan untuk :

1. Mendeteksi keberadaan cacat pengelasan.
2. Mengetahui faktor konsentrasi tegangan pada titik toe.
3. Mengetahui umur keretakan awal dan lokasi terjadinya keretakan awal pada spesimen.
4. Mengetahui umur fatigue spesimen dan perubahannya karena perlakuan TIG Dressing.

Secara umum pelaksanaan studi eksperimen ini dapat diringkas dalam diagram berikut :



Gambar 1.1 Diagram alur pelaksaan studi eksperimen

BAB II
PENGELASAN
DAN SIFAT KELELAHAN SAMBUNGAN LAS

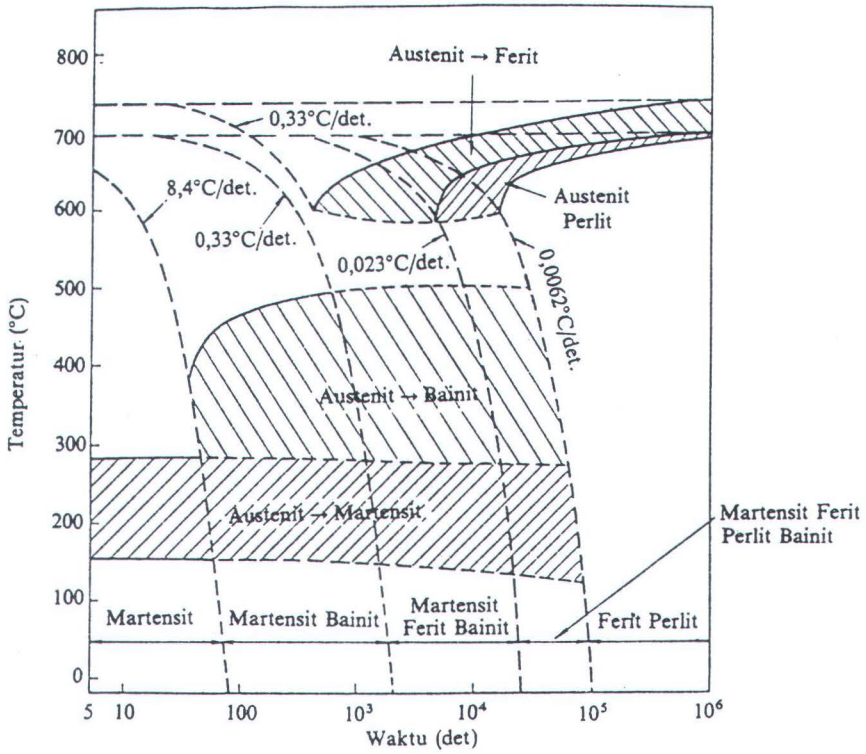
BAB II
PENGELASAN
DAN SIFAT KELELAHAN SAMBUNGAN LAS

2.1. BAJA DALAM PENGELASAN

2.1.1. Struktur Mikro dan Sifat Mekanis

Pada umumnya struktur mikro dari baja tergantung dari kecepatan pendinginannya dari suhu daerah *austenit* sampai ke suhu kamar. Karena perubahan struktur ini maka dengan sendirinya sifat-sifat mekanis yang dimilikinya juga berubah. Hubungan antara sifat pendinginan dengan struktur mikro yang terbentuk, digambarkan dengan suatu diagram yang menghubungkan waktu dan suhu transformasi yang biasanya disebut dengan diagram CCT *(Continuous Cooling Transformation).* Pada gambar (2.1) ditunjukkan contoh diagram CCT untuk baja ASTM 4340.

Dari gambar tersebut diatas dapat dilihat bahwa bila kecepatan pendinginan naik yang berarti waktu pendinginan dari suhu austenit turun, struktur akhir yang terjadi berubah dari campuran *ferit-perlit-bainit-martensit*, menjadi *bainit-martensit* dan akhirnya pada kecepatan yang tinggi sekali terbentuk struktur *martensit.*



Gambar 2.1 Diagram CCT baja ASTM 4340

Kekuatan baja *ferit-perlit* terutama batas luluhnya tergantung pada ukuran butir *ferit,* disamping kekuatan ternyata besar butir juga mempengaruhi energi patah dan sifat perambatan retaknya. Penelitian-penelitian menunjukkan bahwa makin halus butir-butir kristal makin rendah suhu ulet-getasnya, karena itu tindakan memperhalus butir adalah tindakan yang tepat untuk memperbaiki kekuatan dan ketangguhan baja ferit-perlit.

Kekuatan, ketangguhan dan kekerasan baja juga ditentukan oleh komposisi kimia yang dikandung dalam baja. Komposisi kimia juga menentukan sifat mampu las baja, untuk menentukan sifat

mampu las suatu baja biasanya digunakan kadar karbon dimana unsur-unsur lain dinyatakan dalam karbon ekuivalen.

$$C_{ek} = C + 1/6\ Mn + 1/24\ Si + 1/40\ Ni + 1/5\ Cr + 1/4\ Mo + 1/14\ V$$

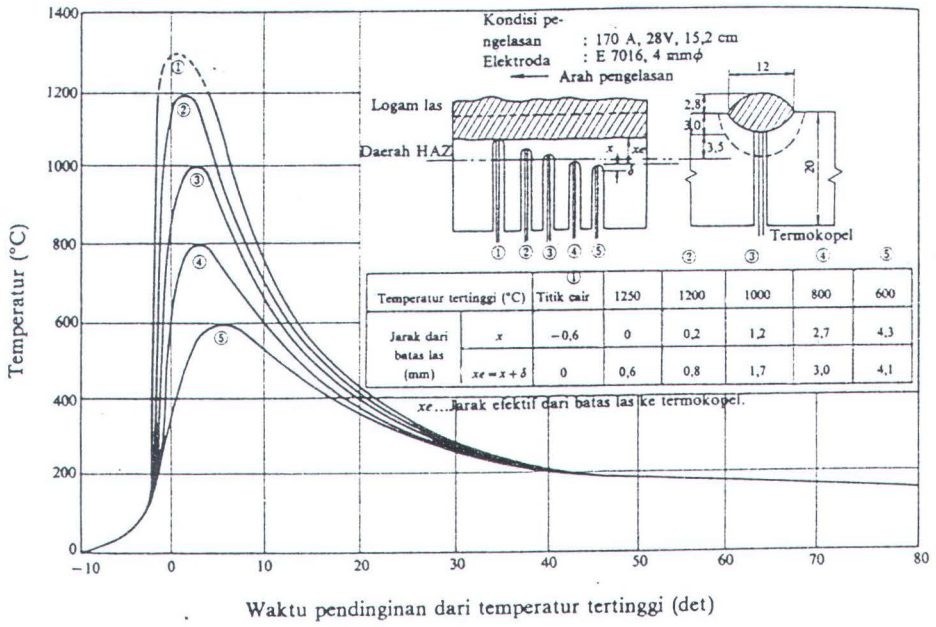
Untuk memperbaiki sifat baja kadang-kadang perlu ditambahkan unsur V dan Nb yang akan mempertinggi kekuatan baja dengan membentuk endapan halus VN dan NbC pada waktu terjadi transformasi austenit ke ferit.

Baja berkekuatan tinggi *(high tensile steel)* biasanya mengandung beberapa unsur campuran seperti Cu, Ni, Cr, Mo dan V. Hal terpenting dalam menghasilkan baja berkekuatan tinggi adalah usaha untuk menurunkan karbon ekuivalen serendah mungkin sehingga sifat mampu-lasnya tetap tinggi. Selain itu adalah pengaturan struktur mikro pengeras, yaitu struktur mikro ganda dari *bainit halus* dan martensit, dapat terbentuk sehingga dapat memperbaiki kekuatan baja.

Struktur mikro ganda merupakan struktur bainit yang terbentuk lebih dahulu yang dalam pertumbuhannya akan membagi butir-butir austenit menjadi belahan-belahan yang lebih kecil daripada butir martensit tunggal. Dengan pembagian yang lebih halus ini maka jarak rambat retak menjadi lebih pendek dan akibatnya rambatan retak dapat lebih terhalangi.

2.1.2. Siklus Thermal Daerah Lasan

Siklus thermal adalah proses pemanasan dan pendinginan didaerah lasan. Pada gambar (2.2) ditunjukkan siklus thermal daerah lasan dari las busur listrik dengan elektroda terbungkus *(Shield Metal Arc Welding/SMAW)*, untuk beberapa tempat dalam daerah HAZ dengan kondisi pengelasan tetap. Lamanya pendinginan dari suatu area temperatur tertentu dari suatu siklus thermal las, sangat mempengaruhi kualitas sambungan. Karena itu banyak sekali usaha-usaha pendekatan untuk menentukan lamanya waktu pendinginan tersebut.

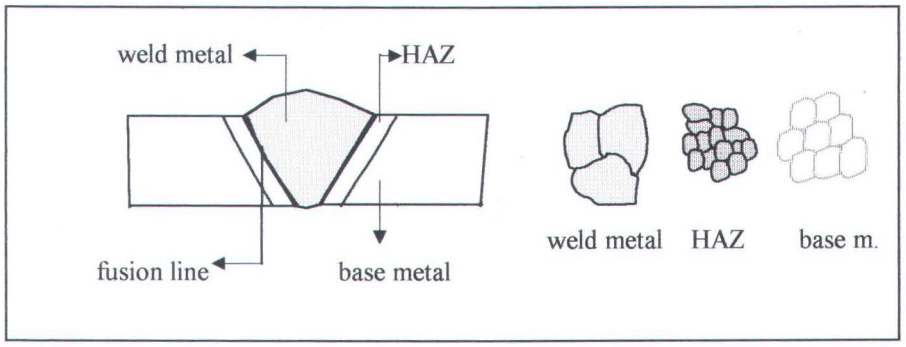


Temperatur tertinggi (°C)		Titik cair	1250	1200	1000	800	600
Jarak dari batas las (mm)	x	−0,6	0	0,2	1,2	2,7	4,3
	$xe = x + \delta$	0	0,6	0,8	1,7	3,0	4,1

Gambar 2.2 Siklus thermal las pada beberapa jarak dari batas las

Pengelasan menghasilkan tiga bentuk logam yang mempunyai kharakteristik berbeda yaitu logam las *(weld metal)*, daerah yang

dipengaruhi panas *(Heat Affected Zone/HAZ)* dan logam induk *(base metal)*. Logam las adalah bagian dari logam yang pada waktu pengelasan mencair dan kemudian membeku. Daerah pengaruh panas atau HAZ adalah logam dasar yang bersebelahan dengan logam las yang selama proses pengelasan mengalami siklus thermal pemanasan dan pendinginan yang cepat. Logam induk merupakan bagian dari logam dasar yang tidak terpengaruh panas secara langsung, dimana suhu pengelasan tidak menyebabkan perubahan-perubahan struktur dan sifat. Disamping ketiga pembagian tersebut masih ada satu daerah khusus yang membatasi antara logam las dan daerah HAZ yang disebut daerah batas las atau *fusion line*.



Gambar 2.3 Daerah dan besar butir logam hasil pengelasan.

Struktur logam pada weld metal terbentuk besar-besar sedangkan struktur logam induk lebih kecil dan halus. Struktur logam pada daerah pengaruh panas (HAZ) berubah secara berangsur dari struktur logam induk ke struktur logam las. Pada daerah HAZ yang dekat dengan garis lebur, kristalnya tumbuh dengan cepat dan

membentuk butir-butir kasar, daerah tersebut yang dinamakan daerah batas las [Wiryosumarto, 1994].

Struktur mikro dan sifat mekanis dari daerah HAZ sebagian besar tergantung pada lamanya pendinginan dari temperatur 800° sampai 500°. Struktur yang terbentuk, kekerasan dan proses berlangsungnya transformasi dari daerah HAZ dapat terbaca dari diagram transformasi pendinginan (diagram CCT), dengan diagram ini dapat digunakan untuk membahas pengaruh struktur terhadap retak las, keuletan dan sebagainya.

Pada daerah batas las dimana butir-butirnya sangat kasar, logam menjadi sangat getas sehingga dinamakan penggetasan batas las. Pada batas las inilah sering terjadi konsentrasi tegangan yang disebabkan diskontinuitas pada kaki manik las, takik las, retak las dan sebagainya. Penggetasan batas las disebabkan oleh tumbuhnya kristal menjadi butir-butir kasar atau karena terbentuknya struktur *bainit atas*. Ketangguhan batas las ini selain ditentukan oleh kecepatan pendinginan juga ditentukan oleh komposisi kimia logam induk.

Pada baja kekuatan tinggi, tingkat kegetasan yang terjadi akibat masukan panas lebih besar bila dibandingkan dengan baja kekuatan rendah. Bila terbentuk struktur bainit atas, maka akan terbentuk butir-butir martensit karbon tinggi yang sangat getas diantara butir bainit tersebut. Bila hal ini terjadi maka ketangguhan

batas las akan lebih rendah dibandingkan bila terbentuk struktur ferit-perlit. Pada baja kekuatan tinggi karena banyak mengandung unsur paduan dengan kadar tinggi maka kemungkinan terbentuknya bainit atas sangat besar.

Jelas sekali bahwa pengaruh komposisi kimia logam induk sangat besar terhadap ketangguhan batas las dan dapat berubah secara rumit, selain karena macam dan besarnya kadar dari masing-masing unsur juga karena kecepatan pendinginan yang terjadi selama pengelasan. Unsur paduan yang mempunyai hubungan erat dengan penggetasan batas las adalah karbon (C) dan nikel (Ni). Dalam hal ini penurunan kadar karbon dan penambahan unsur nikel akan meningkatkan ketangguhan batas las.

2.1.3. Baja ASTM A 516 grade 70

Baja *(steel)* adalah suatu produk besi yang mengandung kadar karbon sekitar 1,7 %. Besi murni *(ferrit)* tidak mengandung karbon sehingga bersifat relatif lunak, liat dan mampu tempa akan tetapi tidak kuat. Hampir semua besi murni mempunyai kekuatan tarik batas sekitar 40.000 psi. Penambahan karbon kedalam besi murni dalam jumlah yang berkisar antara 0,05 % sampai 1,7 % akan menghasilkan baja yang secara teknis disebut dengan baja karbon.

Selain besi sebagai komponen utama dan karbon sebagai unsur kedua, didalam baja karbon terdapat unsur Mn, P, S dan Cu.

Namun demikian untuk memperbaiki sifat baja yang terbentuk seringkali dimasukkan beberapa unsur tambahan yang tidak dimiliki oleh baja karbon seperti Ni, Cr, Mo dan sebagainya. Produk baja jenis ini disebut baja paduan *(alloy steel).*

Sifat baja karbon sangat tergantung pada kadar karbon didalamnya, secara umum semakin tinggi kandungan karbon akan menimbulkan :

1. Penambahan kekuatan dan kekerasan.
2. Penurunan elastisitas *(ductility).*
3. Penurunan sifat mampu las.
4. Penurunan konduktivitas dan ketahanan terhadap korosi.

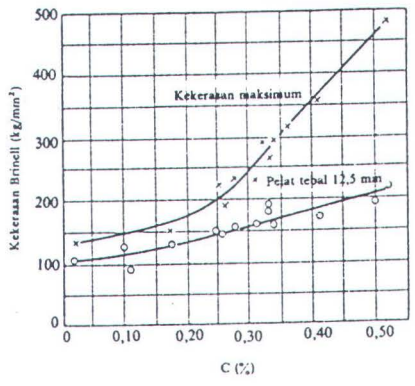
Oleh karena itu baja karbon dibagi dalam tiga kelompok berdasarkan pada kandungan karbonnya :

1. Baja karbon rendah *(low carbon steel)* mengandung maksimal 0,3 % C.
2. Baja karbon menengah *(medium carbon steel)* mengandung 0,3 % sampai 0,45 % C.
3. Baja karbon tinggi *(high carbon steel)* mengandung 0,45 % sampai 1,7 % C.

Baja ASTM A 516 grade 70 tergolong baja karbon menengah *(medium carbon steel).* Baja ini mempunyai sifat yang kuat dan keras, tetapi mempunyai sifat mampu las yang rendah sehingga pengelasan cukup sulit untuk dilakukan. Pengelasan akan

membentuk hasil yang baik jika menggunakan prosedur dan elektroda yang khusus, sebab jika tidak akan mudah terjadi keretakan.

Secara umum baja karbon menengah dan baja karbon tinggi mengandung banyak karbon dan unsur lain yang dapat memperkeras baja. Oleh karena itu daerah pengaruh panas atau HAZ pada baja ini mudah menjadi keras bila dibandingkan dengan baja dengan kadar karbon rendah. Hubungan antara kekerasan maksimum yang dapat dicapai dengan kadar karbon dapat dilihat pada gambar (2.4). Sifat yang mudah keras ini ditambah dengan adanya difusi hidrogen akan menyebabkan baja ini sangat peka terhadap retak las. Disamping itu pengelasan dengan menggunakan elektroda yang mempunyai kekuatan yang sama dengan logam las akan mempunyai sifat perpanjangan yang rendah.



Gambar 2.4. Hubungan antara kekerasan maksimum pada daerah HAZ dengan kadar karbon dalam baja

Sesuai dengan spesifikasi ASME, material ini dinamakan *SA 516 grade 70 carbon steel* dan digolongkan sebagai material pressure vessel dengan kekuatan tarik 70-90 Ksi (485-620 Mpa) dan regangan minimum 17 %, sehingga temasuk dalam kelompok material dengan kekuatan tarik tinggi *(high tensile steel).*

Pemilihan baja ASTM A 516 grade 70 sebagai material pada eksperimen ini didasarkan pada kebutuhan untuk mendapatkan kharakteristik material yang elastisitasnya rendah atau *brittle.* Sifat ini dibutuhkan untuk menganalisa penurunan kekuatan terhadap beban berulang pada suatu daerah yang mengalami diskontinuitas dan terjadi konsentrasi tegangan.

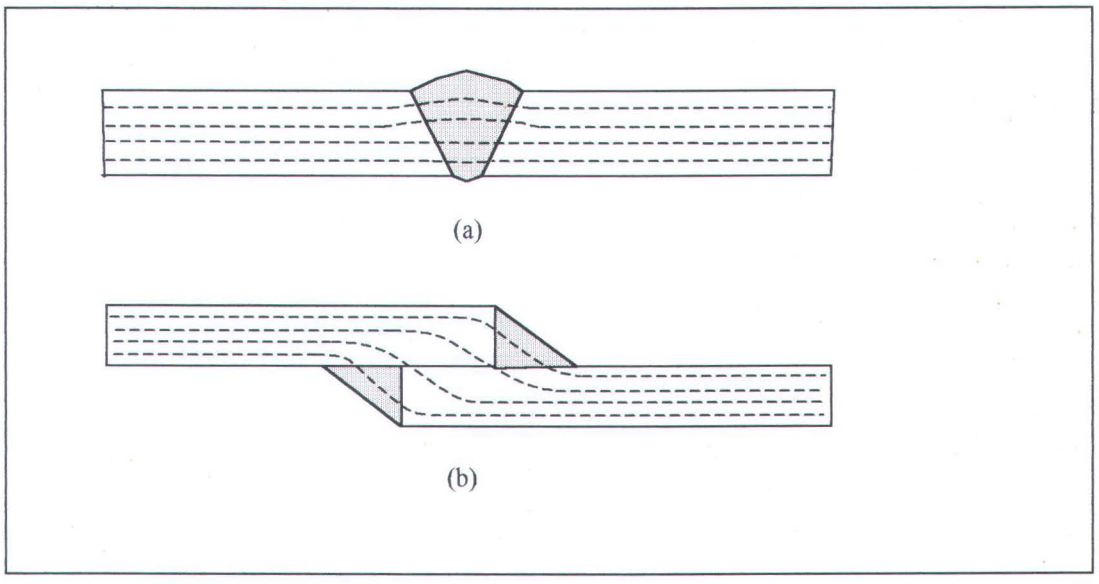
Pada material yang ulet *(ductile),* apabila terjadi suatu diskontinuitas yang menimbulkan konsentrasi tegangan pada daerah tersebut, maka puncak distribusi tegangan yang terjadi akan direduksi oleh adanya pemuluran *(yielding).* Pemuluran ini akan dimulai pada titik dimana terjadi tegangan maksimum, jika material mempunyai rentang ketidak-elastisan yang tinggi seperti pada material struktur pada umumnya *(mild steel)* maka material akan mulur pada daerah konsentrasi tegangan tanpa menimbulkan bahaya kepecahan [Byars & Snyder, 1968].

Hal ini merupakan kelebihan material ductile sehingga dalam desain, faktor konsentrasi tegangan kurang begitu diperhitungkan kecuali pada pembebanan dinamis. Sebaliknya pada material yang

brittle sifat tersebut tidak dimiliki, sehingga konsentrasi tegangan merupakan salah satu faktor utama yang harus dipertimbangkan dalam desain baik untuk pembebanan statis terlebih lagi pada pembebanan dinamis.

2.2. BUTT JOINT PADA PENGELASAN BAJA

Dibandingkan dengan jenis sambungan las yang lain, maka pada sambungan tumpul *(butt joint)* gangguan yang terjadi pada aliran tegangan *(stress flow)* adalah yang paling kecil, sehingga diharapkan pada jenis sambungan ini mempunyai kekuatan kelelahan yang lebih baik dibandingkan dengan jenis sambungan lain yang mengalami perubahan aliran tegangan secara tiba-tiba.



Gambar 2.5. Hubungan jenis sambungan dengan aliran tegangan

Dalam ketidakhadiran cacat pada pengelasan, konsentrasi tegangan merupakan faktor yang paling berpengaruh terhadap

kekuatan kelelahan. Untuk spesimen yang dibiarkan pada kondisi normal *(as welded)*, konsentrasi tegangan yang paling besar adalah pada titik toe. Oleh karenanya kegagalan fatigue pada suatu spesimen biasanya terjadi pada titik toe, baik pada bagian atas (top) maupun bagian bawah (backing run).

Namun demikian, pada suatu spesimen yang merupakan hasil pengelasan berkualitas yang berarti tanpa timbul cacat, ada sejumlah faktor yang dapat mempengaruhi kekuatannya terhadap pembebanan berulang yaitu :

1. Komposisi kimia dan sifat mekanis material induk.
2. Bentuk lasan.
3. Proses pengelasan dan jenis elektroda.
4. Jenis dan metode persiapan las.
5. Posisi pengelasan.
6. Proses machining yang dilakukan setelah pengelasan.
7. Perlakuan panas setelah pengelasan (PWHT).

Faktor-faktor tersebut saling mempengaruhi, namun kenyataan yang timbul menunjukkan bahwa bentuk lasan adalah faktor yang paling berpengaruh dalam menentukan kekuatan kelelahan sambungan las tumpul. Sedangkan pengaruh dari faktor-faktor yang lain ditentukan oleh pengaruh mereka terhadap bentuk lasan.

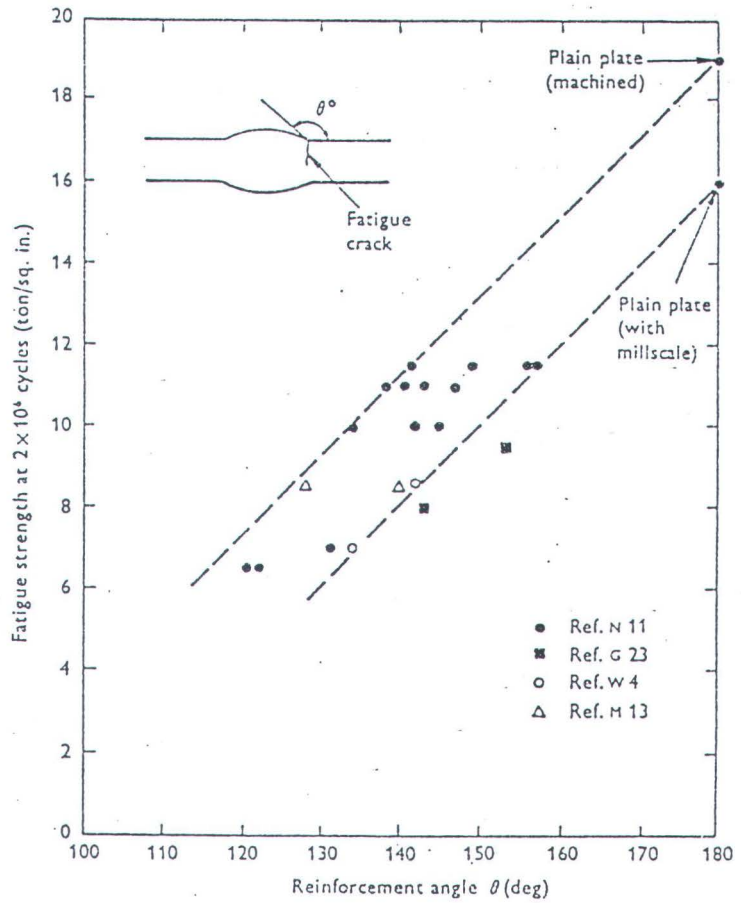
2.3. KELELAHAN PADA SAMBUNGAN LAS

2.3.1. Pengaruh Bentuk pada Kekuatan Kelelahan

Wilson melakukan penelitian sejauh mana pengaruh bentuk lasan terhadap kekuatan kelelahan pada sambungan tumpul. Dalam eksperimen tersebut digunakan pelat mild steel 22 mm dengan persiapan sisi U, kemudian hasil pengelasan dikelompokkan secara kualitatif berdasarkan tinggi manik las dan ketajaman sudut pada sisi-sisinya. Spesimen dengan permukaan manik las halus dan agak landai digolongkan pada kelompok A, sedangkan spesimen yang permukaan manik lasnya berbentuk tinggi dan kasar digolongkan dalam kelompok D, sementara kelompok B dan C adalah spesimen dengan mutu manik las diantara keduanya. Setelah dilakukan pengujian dapat disimpulkan bahwa walaupun dapat dilakukan pengelompokan bentuk hasil las yang baik dan buruk, namun definisi kualitatif ini kurang praktis untuk mengetahui pengaruh bentuk hasil las terhadap kekuatan fatigue.

Pada tahap berikutnya Newman dan Gurney melakukan penelitian terhadap beberapa jenis sambungan tumpul yang dibuat dengan proses manual maupun secara otomatis. Bentuk hasil las didefinisikan secara kuantitatif yaitu dengan mengukur sudut kemiringan sisi manik las terhadap permukaan material induk *(θ)*. Hasil pengujian menunjukkan bahwa besarnya sudut kemiringan sisi manik las bervariasi sepanjang pengelasan, terutama pada

proses pengelasan manual, akan tetapi kegagalan biasanya diawali pada titik dengan sudut kemiringan yang kecil. Kemudian hasil pengujian yang berupa kekuatan fatigue dari masing-masing spesimen diplot terhadap sudut kemiringan sisi manik las, seperti tampak pada gambar (2.6) berikut.



Gambar 2.6 Hubungan antara sudut kemiringan sisi manik las dengan kekuatan fatigue sambungan tumpul.

Perbedaan bentuk manik las ternyata juga mempengaruhi daerah timbulnya keretakan awal pada sambungan las [Newman & Gurney, 1974]. Hal ini terbukti pada pengujian yang menggunakan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

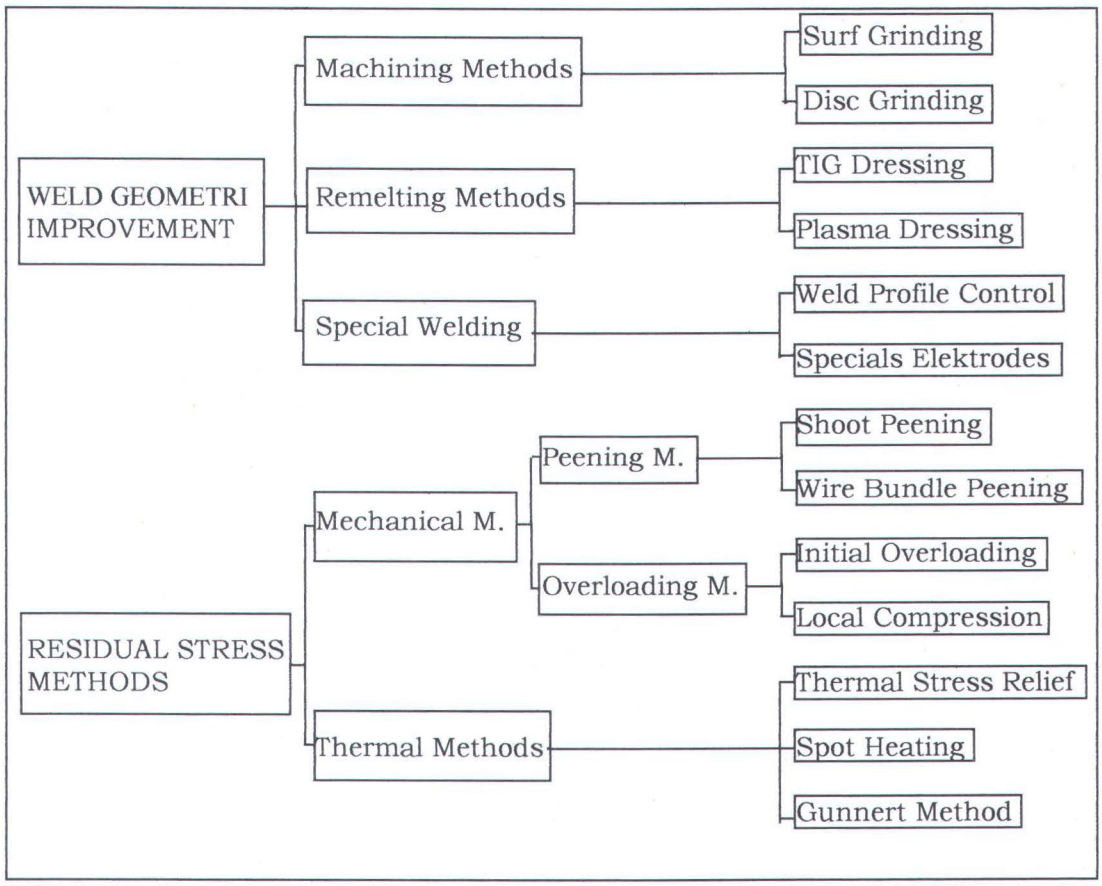
teknik pengelasan yang berbeda, pada pengelasan manual yang menghasilkan permukaan manik las yang bervariasi ketinggian maupun sudutnya, keretakan justru terlokalisasi pada titik toe. Sedangkan teknik pengelasan otomatis yang menghasilkan manik las rata dan halus kegagalan kurang dapat diperkirakan tempat kejadiannya karena banyak faktor lain yang mempengaruhi.

2.3.2. Peningkatan Kekuatan Fatigue

Ada tiga faktor utama yang mempengaruhi kekuatan fatigue sambungan las yaitu : 1. Konsentrasi tegangan karena sambungan dan bentuk manik las; 2. Cacat, bentuk dan distribusinya; 3. Tegangan sisa. Berdasarkan hal tersebut maka usaha untuk meningkatkan kekuatan fatigue sambungan las dapat digolongkan dalam dua kelompok [Almar-Naess, 1985]:

1. Modifikasi bentuk manik las dan penghilangan cacat.
2. Metode tegangan sisa.

Pembahasan akan dilakukan pada metoda peningkatan kekuatan fatigue dengan modifikasi bentuk manik las. Metoda penggerindaan merupakan metoda yang banyak dipakai secara luas karena pelaksanaannya cukup mudah dan praktis. Penggerindaan dapat dilakukan dengan *rotary burr* atau dengan *disc grinding*, penggerindaan dilakukan dapat pada seluruh permukaan manik las ataupun hanya pada titik toe saja.



Gambar 2.7 Klasifikasi metoda peningkatan kekuatan fatigue

Tujuan daripada penggerindaan adalah untuk mendapatkan permukaan manik las yang landai bahkan seluruh permukaan manik las dapat dihilangkan agar aliran tegangan pada sambungan tidak terganggu atau terpatahkan oleh adanya manik las. Manik las merupakan gangguan pada aliran tegangan sehingga menimbulkan konsentrasi tegangan pada daerah tersebut, mekanisme konsentrasi tegangan inilah yang dapat menurunkan kekuatan kelelahan.

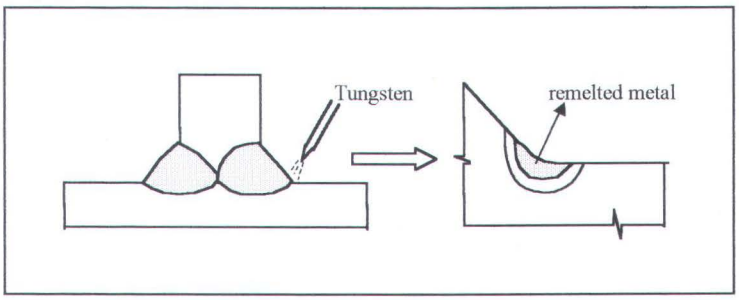
Laporan dari berbagai hasil penelitian menunjukkan bahwa kegagalan seringkali terjadi dimulai dari adanya cacat pada hasil

pengelasan, namun demikian dari pengujian spesimen yang tidak mengandung cacat didapatkan bahwa konsentrasi tegangan pada toe merupakan penyebab utama kegagalan fatigue. Oleh karena itu dengan menghilangkan manik las maka konsentrasi tegangan pada toe akan hilang, sehingga apabila tidak terjadi cacat pada pengelasan kekuatan sambungan terhadap beban berulang diharapkan sama dengan kekuatan logam induk.

Penggerindaan dapat dilakukan pada hampir semua sambungan dalam bagian struktur termasuk pada bagian yang tetutup dan sulit. Akan tetapi resiko yang mungkin terjadi pada penggerindaan adalah timbulnya goresan *(scratcing)* pada permukaan yang digerinda. Goresan ini sebaliknya dapat berperan sebagai konsentrasi tegangan baru yang memicu keretakan awal. Oleh karena itu penggerindaan harus dilakukan dengan prosedur yang benar.

Selain dengan penggerindaan, proses untuk membentuk permukaan manik las dapat dilakukan dengan teknik pencairan *(remelting)*. Proses ini dilakukan dengan menggunakan panas untuk mencairkan permukaan logam las dan membentuknya sehingga permukaannya halus dan sudut kemiringannya landai. Pemanasan dapat dilakukan dengan pengelasan TIG maupun dengan pengelasan plasma.

Teknik pencairan ini juga telah banyak digunakan, terutama dengan pengelasan TIG *(TIG Dressing)*. Proses ini menggunakan teknik pengelasan elektroda *non-consumable* atau TIG *(Tungsten Inert Gas) welding* tanpa menggunakan filler. Panas yang ditimbulkan oleh elektoda tungsten digunakan untuk mencairkan



Gambar 2.8. TIG Dressing pada titik toe.

kembali manik las yang telah terbentuk. Pencairan dilakukan pada titik toe sehingga sudut kemiringan sisi manik las menjadi lebih besar.

Metoda pembentukan manik las yang lain adalah dengan mengatur dan mengontrol pembentukan manik las pada saat proses pengelasan. Penggunaan elektroda yang khusus sehingga manik las yang dihasilkan bagus dan halus, juga merupakan alternatif yang dapat dilakukan. Dari semua cara perlakuan untuk membentuk manik las ini mempunyai maksud yang sama, yaitu membentuk permukaan manik las yang halus dan rata serta membentuk sudut kemiringan yang landai. Sedangkan tujuan akhir dari semua

perlakuan itu adalah untuk mengurangi konsentrasi tegangan pada daerah sambungan las.

Pada pembebanan berulang kekuatan sambungan las sangat ditentukan oleh besarnya konsentrasi tegangan. Semakin rendah tingkat konsentrasi tegangan yang terjadi pada sambungan las, semakin kecil kemungkinan timbulnya keretakan awal. Dengan demikian semakin panjang umur fatigue sambungan tersebut.

BAB III
KELELAHAN DAN KEPATAHAN

BAB III
KELELAHAN DAN KEPATAHAN

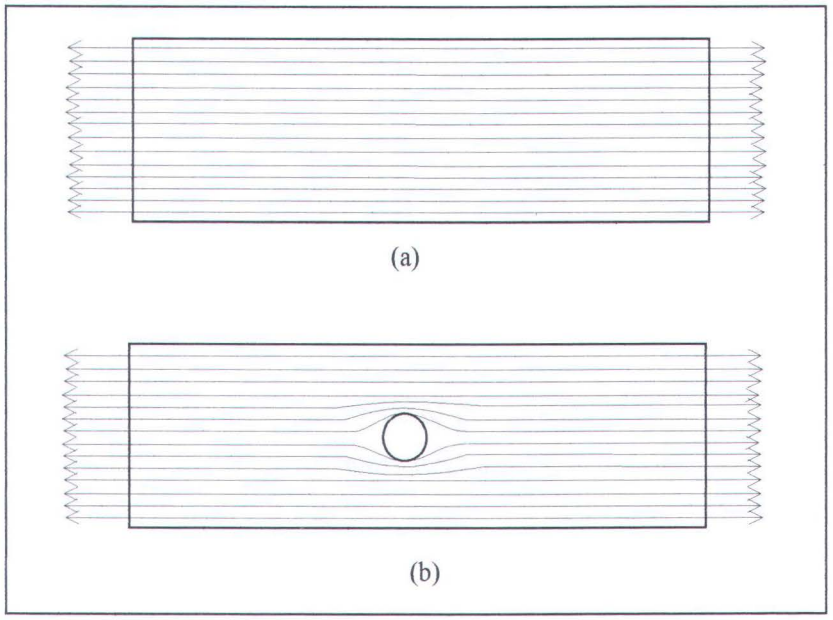
3.1. KONSENTRASI TEGANGAN

Fatigue dan konsentrasi tegangan merupakan dua hal yang saling berkaitan, sehingga pembahasan mengenai salah satu topik tersebut akan selalu melibatkan yang lain. Kehadiran kedua hal tersebut akan menjadi faktor pengontrol dalam desain struktur, karena kombinasi fatigue dan konsentrasi tegangan banyak menimbulkan kegagalan pada struktur sebagi akibat beban berulang.

Konsep mengenai konsentrasi tegangan dapat dijelaskan dengan uraian sebagai berikut. Pada suatu material yang menerima gaya, aliran gaya yang melaluinya dapat terganggu atau terpatahkan oleh diskontinuitas atau adanya perubahan bentuk material secara tiba-tiba sehingga konsentrasi dari garis-garis gaya akan terbentuk.

Dalam gambar (3.1-a) ditunjukkan garis-garis gaya pada suatu pelat yang dibebani secara uniaxial. Apabila pada pelat tersebut timbul suatu diskontinuitas, misalnya ditunjukkan dengan adanya suatu lubang, maka pada daerah disekitar lubang tersebut akan terbentuk konsentrasi aliran gaya seperti tampak pada gambar (3.1-b).

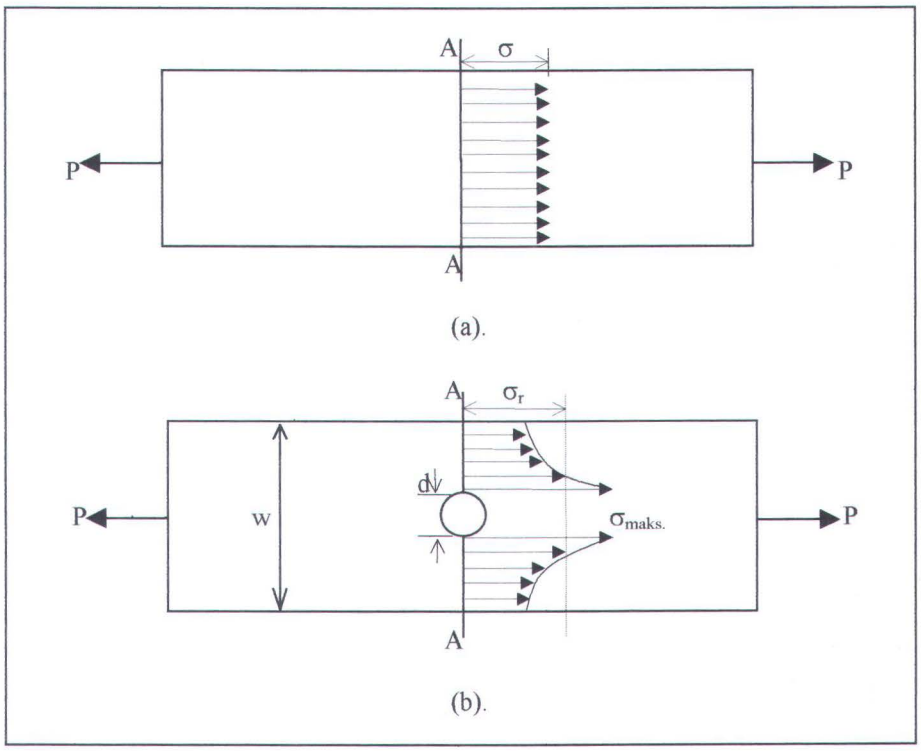
MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 3.1 Aliran gaya-gaya pada pelat yang terbebani

Panjang vektor pada gambar (3.2-a) menunjukkan besarnya tegangan yang terjadi pada beberapa titik sepanjang bidang pemotongan A-A, tegangan tersebut merata sepanjang bidang. Dengan hadirnya lubang pada pelat maka regangan dan tegangan akan terkonsentrasi disekitar lubang, gambar (3.2-b). Puncak tegangan pada bidang A-A terjadi pada daerah yang berbatasan dengan lubang, puncak tegangan ini mempunyai harga yang jauh lebih besar dari tegangan rata-rata yang terjadi pada bidang tanpa lubang. Puncak tegangan yang terjadi pada struktur yaitu pada daerah dimana terjadi diskontinuitas atau perubahan bentuk pada material, merupakan faktor yang sangat penting dalam desain struktur yang menerima beban berulang. Selain itu fenomena

ini juga harus menjadi pertimbangan utama dalam desain struktur yang menggunakan material getas (*brittle*).



Gambar 3.2 Konsentrasi tegangan pada pelat berlubang.

Faktor konsentrasi tegangan

Dalam pelaksanaannya sulit untuk menentukan besarnya tegangan maksimum pada daerah konsentrasi tegangan berdasarkan analisa teoritis. Oleh karena itu biasanya untuk menentukan besarnya tegangan maksimum dilakukan dengan mengalikan suatu faktor pada tegangan rata-rata. Faktor ini disebut dengan *faktor konsentasi tegangan (K)* yang diperoleh secara

eksperimental pada beberapa perubahan bentuk yang umum seperti lubang, celah, takikan dan sebagainya.

Faktor konsentrasi tegangan ini dinyatakan sebagai perbandingan antara tegangan maksimum ($\sigma_{maks.}$) dengan tegangan rata-rata (σ_r).

$$K = \frac{\sigma_{maks.}}{\sigma_r} \qquad (3.1)$$

Tegangan rata-rata biasanya dinyatakan dengan menggunakan penampang rata-rata (A_r) pada bidang yang mengalami perubahan bentuk, sehingga pada gambar (3.2-b) diatas, $A_r = (w-d)t$.

$$\sigma_{maks.} = K.\frac{P}{A_r} \qquad (3.2)$$

Faktor konsentrasi tegangan selalu berharga lebih dari satu, kecuali pada kasus yang sangat khusus. Pada umumnya faktor konsentrasi tegangan ditentukan secara eksperimental dengan menggunakan model (misalnya dengan menggunakan model *photoelastic*). Hasilnya merupakan suatu faktor yang hanya berdasarkan pada pertimbangan bentuk tanpa memperhitungkan pengaruh dari material. Faktor konsentrasi tegangan ini disebut *faktor konsentrasi tegangan teotitis* (K_t), yang merupakan perbandingan distribusi tegangan yang ditentukan dari teori elastisitas secara matematis, dengan harga tegangan rata-rata. Sedangkan faktor konsentrasi yang ditentukan dengan

perbandingan distribusi tegangan nyata (termasuk pengaruh sifat material pada distribusi tegangan) dengan tegangan rata-rata yang diperoleh dari rumus dasar, disebut dengan *faktor konsentrasi tegangan efektif (K_e).*

3.2. ATURAN NEUBER TENTANG KONSENTRASI TEGANGAN

3.2.1. Konsentrasi Tegangan

Umur fatigue dari suatu spesimen yang mengalami diskontinuitas (bertakik) dapat diperkirakan berdasarkan pada hasil pengujian spesimen material tanpa takik. Untuk memperkirakan umur fatigue spesimen bertakik ini, dibutuhkan konsentrasi tegangan dan regangan pada ujung takikan dan faktor konsentrasi tegangan elastis yang dinyatakan dengan simbol K_t [Neuber, 1961]. Akan tetapi apabila pada material dikenakan beban berulang (siklus), ujung takikan akan bersifat plastis sehingga konsep faktor konsentrasi tegangan tidak dapat diberlakukan.

Neuber melakukan serangkaian analisa pada bentuk takik dan pembebanan yang berbeda untuk memecahkan permasalahan apabila deformasi pada ujung-ujung takikan bersifat tidak linier. Berdasarkan pada asumsi bahwa faktor konsentrasi tegangan elastis sebanding dengan rata-rata geometris dari faktor konsentrasi tegangan dan faktor konsentrasi regangan yang sesungguhnya.

$$K_t = (K\sigma . K\varepsilon)^{1/2} \qquad (3.3)$$

dengan , K_t = Faktor konsentrasi tegangan elastis.

$K\sigma$ = Faktor konsentrasi tegangan plastis.

$K\varepsilon$ = Faktor konsentrasi regangan plastis.

Apabila terjadi pemuluran *(yielding)* harga $K\sigma$ dan $K\varepsilon$ tidak lagi konstan, harga $K\sigma$ akan meningkat sebaliknya harga $K\varepsilon$ akan turun sehingga beralasan apabila perkalian antara $K\sigma$ dan $K\varepsilon$ akan mendekati konstan. Jika pada persamaan (3.3) dilakukan substitusi dengan suatu perbandingan tegangan $\frac{\Delta\sigma}{\Delta\varepsilon}$ dan perbandingan regangan $\frac{\Delta\varepsilon}{\Delta e}$ maka dapat dituliskan sebagai berikut :

$$K_t = \left(\frac{\Delta\sigma}{\Delta s}.\frac{\Delta\varepsilon}{\Delta e}\right)^{1/2} \qquad (3.4)$$

dimana, $\Delta\sigma$ = rentang tegangan lokal yang sebenarnya.

$\Delta\varepsilon$ = rentang regangan lokal yang sebenarnya.

Δs = rentang tegangan nominal.

Δe = rentang regangan nominal.

Kedua sisi persamaan dikalikan dengan modulus Young *(E)* dan disusun kembali sebagai berikut :

$$K_t = \left(\frac{\Delta\sigma}{\Delta s}.\frac{\Delta\varepsilon}{\Delta e}\right)^{1/2}$$

$$K_t^2.E = \frac{\Delta\sigma}{\Delta s}.\frac{\Delta\varepsilon}{\Delta e}.E$$

$$K_t^2.\Delta s.\Delta e.E = \Delta\sigma.\Delta\varepsilon.E$$

$$K_t.(\Delta s.\Delta e.E)^{1/2} = (\Delta\sigma.\Delta\varepsilon.E)^{1/2} \qquad (3.5)$$

Untuk kasus pada material yang bersifat elastis sehingga berlaku hukum *Hooke,* maka persamaan (3.5) dapat dituliskan sebagai berikut :

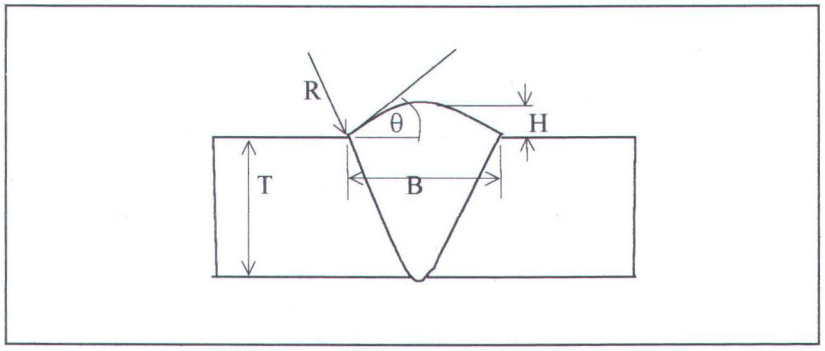
$$K_t.(\Delta s.(\Delta e.E))^{1/2} = (\Delta\sigma.\Delta\varepsilon.E)^{1/2}$$

$$K_t.(\Delta s.\Delta s)^{1/2} = (\Delta\sigma.\Delta\varepsilon.E)^{1/2}$$

$$K_t.\Delta s = (\Delta\sigma.\Delta\varepsilon.E)^{1/2} \qquad (3.6)$$

Dari eksperimen dengan kondisi amplitudo beban konstan dan bersifat bolak-balik penuh, dimana perbandingan antara tegangan minimum dengan tegangan maksimum sama dengan –1, didapatkan bahwa pada spesimen yang bertakik $K_t.(\Delta s.\Delta e.E)^{1/2}$ dan yang tak bertakik $(\Delta\sigma.\Delta\varepsilon.E)^{1/2}$ akan menghasilkan keretakan pada umur fatigue yang sama. Hal ini membuktikan bahwa hasil percobaan pada spesimen tak bertakik dapat digunakan untuk membuat *master plot* guna memperkirakan umur kretakan awal, N_c pada spesimen bertakik.

Untuk menghitung besarnya faktor konsentrasi tegangan karena perubahan bentuk pada sambungan las dapat digunakan persamaan (3.7) berikut. Faktor konsentrasi tegangan ini merupakan fungsi dari flank angle *(θ)*, jari-jari toe *(R)*, tinggi manik las *(H)*, lebar manik las *(B)* dan tebal material *(T)* [Itoh, 1987].



Gambar 3.3 Geometri sambungan las

$$K_t = 1 + f(\theta).\left[\gamma.\left(\frac{1}{(1.4).\Delta i - 2}.\frac{H}{R}\right)^{\beta}\right] \qquad (3.7)$$

dimana, $f(\theta) = \dfrac{1-\exp\left((-0.90).\theta.\sqrt{\Delta i}\right)}{1-\exp\left((-0.45).\pi.\sqrt{\Delta i}\right)}$

$$\beta = 0.65 - 0.1\exp\left(-0.63\frac{B}{T}\right)$$

$$\gamma = 1 - 0.48\exp\left(-0.74\frac{B}{T}\right)$$

$$\Delta i = \frac{2H + T}{2T}$$

3.2.2. Hubungan Tegangan–Regangan

Regangan total *(ε_a)* dapat dibagi menjadi dua bagian yaitu regangan elastis *($\varepsilon_a{}^e$)* dan regangan plastis *($\varepsilon_a{}^p$)* [Manson, 1965]. Regangan elastis dan regangan plastis yang diperoleh dari data hasil percobaan ternyata mempunyai hubungan dengan umur keretakan awal material uji *(N_c)*.

$$\varepsilon_a = \varepsilon_a^e + \varepsilon_a^p$$

$$\varepsilon_a = A.(N_c)^a + B.(N_c)^b \qquad (3.8)$$

dimana : *A, B, a* dan *b* = konstanta yang tergantung pada jenis material.

Sehingga berdasarkan pada persamaan (3.8) diatas dapat dihitung harga tegangan elastis *(σ_a)* sebagai berikut :

$$\sigma_a = \varepsilon_a^e.E$$

$$\sigma_a = A.(N_c)^a.E \qquad (3.9)$$

Untuk mendapatkan konstanta *A, B, a* dan *b* dapat diperoleh dengan melakukan pengujian pada spesimen tanpa takik (material dasar). Selain dengan metoda eksperimen, konstanta material ini juga dapat diperoleh dengan menggunakan suatu rumus perkiraan [Osgood, 1979]. Rumus perkiraan ini merupakan suatu pendekatan yang diperoleh dari berbagai pengujian yang telah dilakukan, sehingga dapat digunakan untuk menggantikan konstanta dari hasil pengujian.

Konstanta material yang digunakan dalam studi ini merupakan konstanta yang diperoleh dari rumus perkiraan. Dalam *Fatigue Design* dijelaskan cara mendapatkan konstanta material berdasarkan pada suatu rumus perkiraan. Konstanta $A = \frac{\sigma'_f}{E}$, $B = \varepsilon'_f$ sedangkan *a* dan *b* merupakan eksponen kekuatan dan

eksponen regangan fatigue. Definisi dan rumus perkiran dari masing-masing konstanta selengkapnya dapat dilihat dalam tabel (3.1) berikut :

Tabel 3.1 Konstanta material, definisi dan perkiraan

Konstanta	Definisi	Rumus Perkiraan
Koefisien kekuatan fatigue (σ'_f)	Perpotongan kurva log $\Delta\sigma$ - log N_f dengan garis vertikal yang di plot pada $N_f = 1$	$\sigma'_f \approx$ kekuatan fracture statis yang sebenarnya (σ_f)
Koefisien regangan fatigue (ε'_f)	Perpotongan kurva log $\Delta\varepsilon^p$ – log N_f dengan garis vertikal yang diplot pada $N_f = 1$	$\varepsilon'_f \approx$ regangan fracture statis (ε_f)
Eksponen kekuatan fatigue (*a*)	Kemiringan kurva log $\Delta\sigma$ - log N_f	$a \approx -1/6 \log 2\sigma_f/\sigma_{ult.}$
Eksponen regangan fatigue (*b*)	Kemiringan kurva log $\Delta\varepsilon^p$ – log N_f	$b \approx -1/(1+ 5n)$

3.2.3. Kurva D-N_c

Dengan menggunakan rumus perkiraan dapat dihitung persamaan fungsi tegangan-regangan *(D)* dan diplot dalam bentuk suatu kurva. Kurva ini dapat diperoleh dengan substitusi persamaan (3.8) dan persamaan (3.9) kedalam persamaan (3.6) yang dinyatakan dalam bentuk amplitudo dari masing-masing besaran. Amplitudo tegangan lokal, $\sigma_a = \frac{\Delta\sigma}{2}$ sedangkan amplitudo regangan lokal, $\varepsilon_a = \frac{\Delta\varepsilon}{2}$.

$$D = (\Delta\sigma.\Delta\varepsilon.E)^{1/2}$$

$$D = 2(\sigma_a.\varepsilon_a.E)^{1/2} \qquad (3.10)$$

Substitusi persamaan (3.8) dan persamaan (3.9) kedalam persamaan (3.10) sebagai berikut :

$$D = 2\left[\left(A.(N_c)^a.E\right)\left(A.(N_c)^a + B.(N_c)^b\right)E\right]^{1/2}$$

$$D = 2\left[F.(N_c)^f + G.(N_c)^g\right]^{1/2} \qquad (3.11)$$

dengan : $F = A^2.E^2$ dan $f = 2.a$

$G = A.B.E^2$ dan $g = a + b$

Menggunakan persamaan (3.11) dapat digambarkan suatu kurva hubungan fungsi tegangan-regangan dan jumlah siklus tegangan yang dibutuhkan agar terjadi keretakan awal, yang disebut dengan kurva D-N_c. Kurva ini dapat digunakan untuk memperkirakan umur keretakan awal dari material uji apabila diketahui besarnya rentang beban yang diterapkan dan faktor konsentrasi tegangannya, dengan menggunakan persamaan $D = K_t.\Delta s$.

3.3. KEGAGALAN FATIGUE

Fatigue dalam dunia teknik digunakan untuk menjelaskan fenomena yang berkaitan dengan beban berulang, yaitu proses pengurangan kekuatan pada komponen karena pembebanan berulang [Vassalos, 1991]. Apabila pengulangan beban cukup besar untuk menyebabkan keretakan, maka keretakan tersebut akan dimulai dari titik dimana terjadi tegangan maksimum dan

konsentrasi tegangan merupakan pemicu timbulnya tegangan maksimum ini.

Perubahan struktural yang disebabkan oleh fatigue merupakan perubahan yang bersifat permanen, terlokalisasi dan progresif. Perubahan ini dapat terjadi pada material yang mendapat regangan secara berulang atau berfluktuasi akibat suatu tegangan, meskipun tegangan tersebut mempunyai nilai maksimum dibawah kekuatan tarik dari material tersebut. Fluktuasi tegangan dan regangan pada selang waktu tertentu akan mengakibatkan terjadinya retak, bahkan sampai pada tingkat tertentu akan mengakibatkan patah *(fracture)* pada material.

Kepatahan fatigue disebabkan oleh adanya : tegangan yang berulang, tegangan tarik dan regangan plastis yang terjadi secara simultan. Tanpa kehadiran salah satu dari ketiga komponen tersebut, retak tidak akan terjadi dan menjalar. Dalam hal ini tegangan berulang akan menimbulkan retak awal kemudian tegangan tarik akan menjalarkan retak tersebut. Secara umum proses fatigue (kelelahan) terbagi dalam tiga tahap, yaitu :

1. Terbentuknya retak awal
2. Penjalaran retak
3. Kepatahan

Permukaan dari suatu kepatahan karena kelelahan dapat dikarakterisasikan untuk menentukan tiga zona atau tahapan perkembangan retak sebagai berikut :

- Tahap I

  Awal terbentuknya retak dan penjalarannya oleh bidang slip yang masuk dari permukaan material dengan sudut 45 derajat terhadap sumbu tegangan. Tahap ini terjadi hanya pada sekitar dua sampai lima partikel dari sumber *(origin)* retak dan ditandai dengan adanya *striation*, bahkan pada beberapa kasus tidak dapat diamati dengan jelas.

- Tahap II

  Tahap ini merupakan kelanjutan dari tahap I dengan transisi yang berupa orientasi bidang kepatahan utama pada tiap-tiap partikel dari satu atau dua bidang geser menjadi sejumlah lapisan paralel yang dipisahkan oleh suatu puncak memanjang. Lapisan-lapisan ini biasanya mempunyai arah yang normal terhadap arah tegangan tarik maksimum.

- Tahap III

  Tahap ini terjadi pada siklus tegangan yang terakhir, saat penampang material yang tersisa tidak mampu lagi menahan tegangan yang bekerja. Kepatahan akhir ini dapat terjadi secara brittle atau ductile atau kombinasi dari keduanya.

3.3.1 Mekanisme Kelelahan

Secara lebih terperinci tahap-tahap kelelahan pada material dapat dijelaskan dalam pendekatan mekanisme kelelahan. Mekanisme tersebut terbagi dalam dua tahapan utama, yaitu tahap pembentukan retak mikro dan tahap pertumbuhan retak kelelahan yang berlangsung sebelum fracture terjadi pada material. Penjalaran akhir tidak dimasukkan dalam mekanisme kelelahan karena keadaan tersebut pada siklus tegangan yang terakhir yang serupa dengan kondisi pada pembebanan statis (monotonik).

a. Pembentukan Retak Mikro

Secara alamiah logam tersusun dari kristal yang terdiri dari atom-atom yang tersusun dalam bentuk tertentu. Struktur logam kebanyakan berupa *polycrystalline,* sehingga terdiri dari sejumlah besar kristal atau partikel dalam rangkaian tertentu. Tiap partikel tersebut mempunyai sifat mekanik tersendiri, demikian pula dengan arah susunan kristalnya. Pada beberapa partikel tertentu susunannya adalah sedemikian rupa sehingga bidang yang mudah bergeser berada pada arah yang sama dengan tegangan geser maksimum yang bekerja.

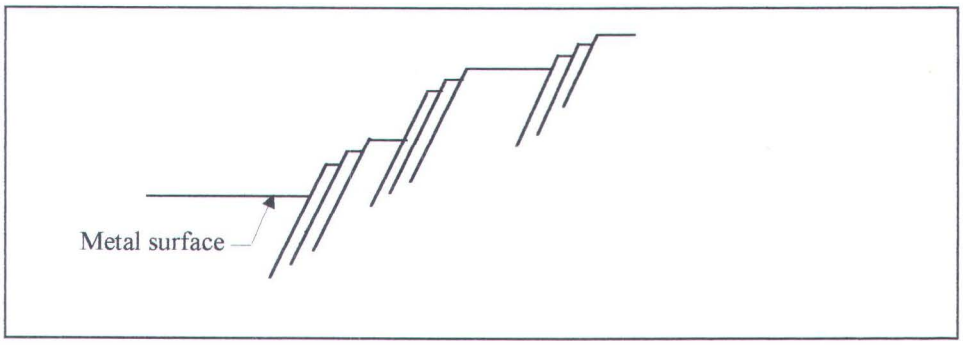
Pada material yang liat (ductile), pergeseran ini terjadi di dalam tiap-tiap partikel berupa suatu dislokasi yang bergerak sepanjang bidang kristalografik. Satu atau lebih bidang tersebut

saling bergeser satu sama lain. Pergeseran ini dapat merupakan akibat dari pembebanan monotonik maupun pembebanan berulang. Pembebanan monotonik atau pembebanan berulang dengan tingkat tegangan yang tinggi akan menghasilkan pergeseran dengan bentuk permukaan yang kasar. Sebaliknya permukaan pergeseran yang lebih halus dihasilkan oleh pembebanan berulang dengan tingkat tegangan yang lebih rendah. Permukaan halus dan kasar yang terjadi akibat pergeseran *(slip)* ini dapat diamati dengan menggunakan mikroskop optikal maupun elektron pada spesimen yang telah dietsa.

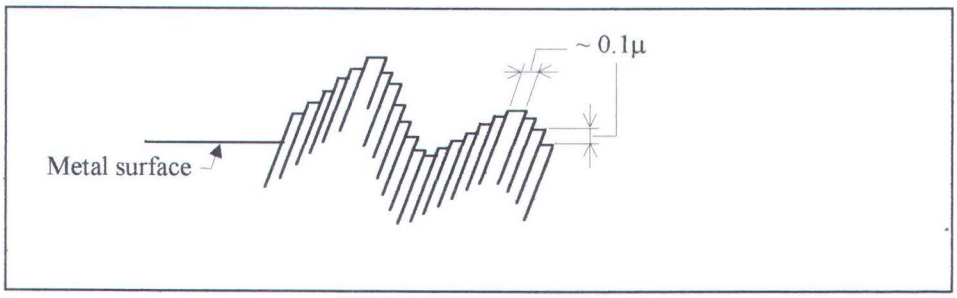
Melalui mikroskop elektron dapat diamati bahwa pembebanan berulang mengakibatkan adanya pita-pita pergeseran berupa *intrusi* maupun *ekstrusi* pada permukaan metal. Intrusi pada pita pergeseran ini dapat menyebabkan terjadinya konsentrasi tegangan yang memungkinkan tejadinya penjalaran retak. Pergeseran pada permukaan ini lebih ditentukan oleh tegangan geser dan tidak terlalu dipengaruhi oleh tegangan normal. Semakin besar tegangan geser yang terjadi semakin besar juga tingkat pergeseran yang terbentuk.

Pada foto mikro dari spesimen yang mengalami pembebanan berulang dapat diamati adanya intensifikasi garis-garis pergeseran sebagai fungsi dari siklus pembebanan yang diberikan. Pergeseran dalam tingkat yang rendah akan mulai tampak pada siklus

pembebanan awal, pada siklus yang semakin tinggi jumlah pergeseran ini akan semakin besar dan pada garis-garis pergeseran tertentu tampak adanya peningkatan ketebalan. Pita-pita pergeseran yang terintensifikasi ini merupakan sumber dari timbulnya retak fatigue. Pergeseran ini terjadi pada daerah-daerah dimana terjadi tegangan dan regangan yang tinggi.



Gambar 3.4 *Slip* yang terjadi pada pembebanan statis (material ductile)



Gambar 3.5 *Slip* yang terjadi pada pembebanan siklis (material ductile)

Garis-garis pergeseran yang tampak pada foto mikro merupakan suatu bentuk slip secara tiga dimensional. Garis-garis tersebut menunjukkan pergeseran permukaan spesimen dengan kedalaman pergeseran yang berbeda-beda. Umumnya pita-pita

pergeseran itu dapat dihilangkan atau diratakan kembali dengan menggunakan *electropolishing* sampai dengan beberapa mikron. Namun demikian beberapa pita slip tertentu akan menjadi semakin jelas dan dinamakan *persistent slip band.* Dari pita pergeseran yang persisten ini kemudian akan tumbuh retak akibat kelelahan.

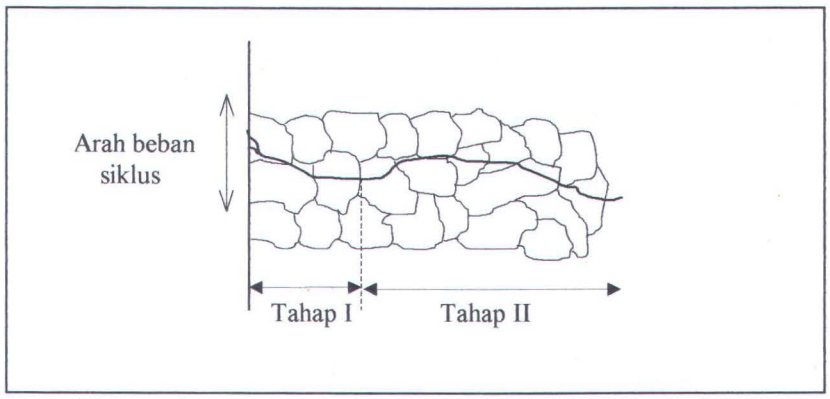
Berdasarkan kenyataan ini perbaikan atau peningkatan umur fatigue dapat dilakukan dengan cara menghilangkan pita pergeseran persisten ini. Pembebanan yang berselang-seling *(intermiten)* dan *electropilishing* terbukti dapat meningkatkan umur kelelahan sehingga kegagalan dapat dihindari. Aspek yang harus diperhatikan disini adalah bahwa tiap awal dari kelelahan adalah merupakan fenomena permukaan, sehingga umur kelelahan ditentukan secara dominan oleh kondisi permukaan spesimen.

Retak kelelahan berawal dari pita pergeseran lokal dan kemudian berkembang melalui bidang dengan interval tegangan geser maksimum. Pertumbuhan ini sangat kecil dan umumnya hanya mencapai besaran beberapa partikel. Selama pembebanan siklus berlanjut, akan terjadi penggabungan antara retak-retak tersebut dan akan terjadi pertumbuhan retak pada bidang dimana terjadi tegangan geser maksimum.

Pada pembesaran mikroskopis terhadap spesimen, dapat diamati pertumbuhan retak fatigue dimana retak awal akan dimulai pada permukaan dan akan berlanjut sampai beberapa besaran

partikel dengan dikendalikan oleh tegangan geser utama. Retak tersebut akan merambat dengan pola zig-zag secara tegak lurus terhadap interval tegangan tarik utama.

Pada tahap ini tegangan tarik berperan sebagai pengendali pertumbuhan retak tersebut. Umumnya retak merambat melalui batas partikel *(transcrystalline)*. Meskipun demikian retak dapat juga merambat melalui batas partikel *(intercrystalline)*, namun dengan tingkat penjalaran yang lebih kecil. Sepanjang penjalaran retak, fenomena pergeseran seperti yang terjadi pada permukaan akan terus berlangsung dan merupakan kejadian mendahului didepan penjalaran retak.

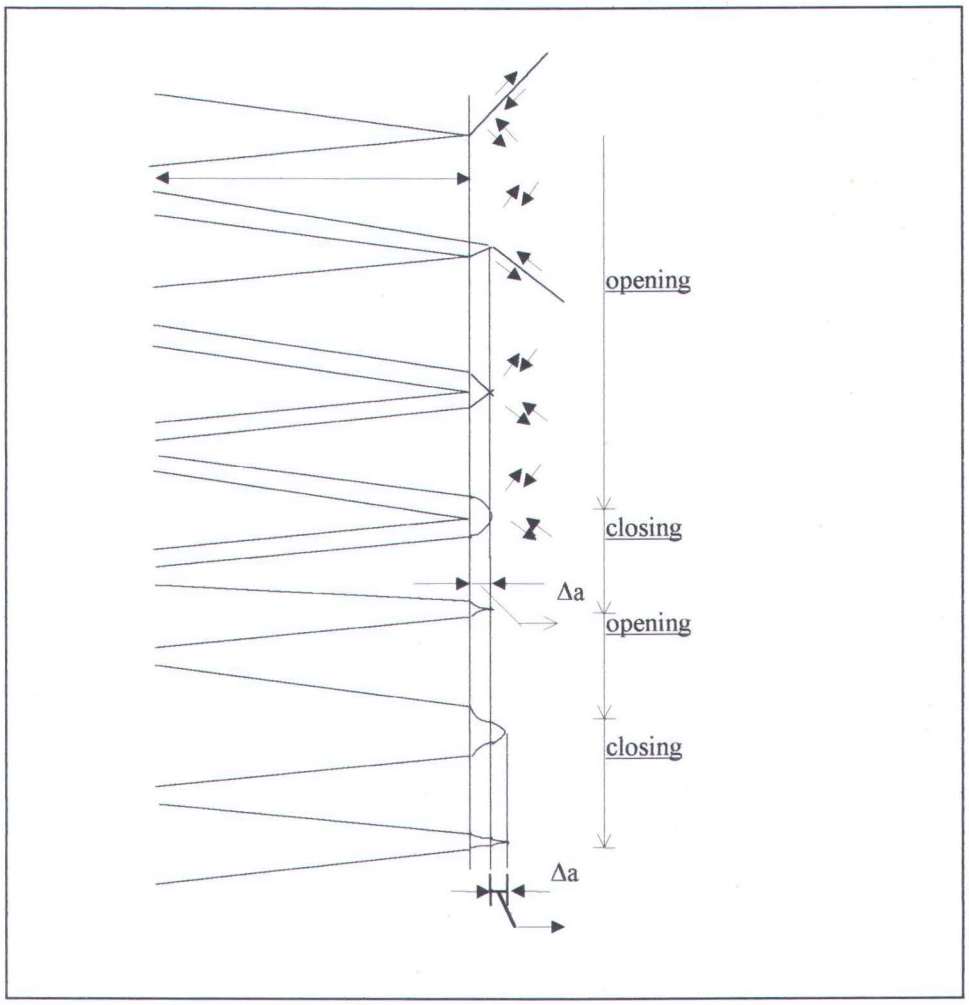


Gambar 3.6 Skema tahap I dan II penjalaran retak transkristalin

b. Pertumbuhan Retak Fatigue

Retak fatigue yang telah berjalan juga dapat berkembang dengan mekanisme pembalikan pergeseran. Retak berujung tajam

yang terjadi pada saat diberikan pembebanan tarik akan menyebabkan terjadinya konsentrasi tegangan diujung retak tersebut sehingga akan mudah terjadi pergeseran. Material yang berada diatas retak akan mengalami pergeseran menurut bidang yang mengalami tegangan geser yang maksimum.



Gambar 3.7 Penajaman dan penumpulan ujung retak

Akibat pergeseran tersebut, retak akan membuka dan menjalar dengan jarak tertentu *(Δa)*. Pergeseran juga dapat terjadi pada bidang geser yang lain seperti ditunjukkan pada periode

penjalaran yang berikutnya. Pengerasan *(hardening)* dan tegangan yang bertambah kemudian juga akan mengaktifkan bidang-bidang pergeseran paralel lainnya sehingga terjadi penumpulan pada ujung retak. Kemudian pada saat dikenakan pembebanan maksimum retak akan kembali menjalar sebesar Δa.

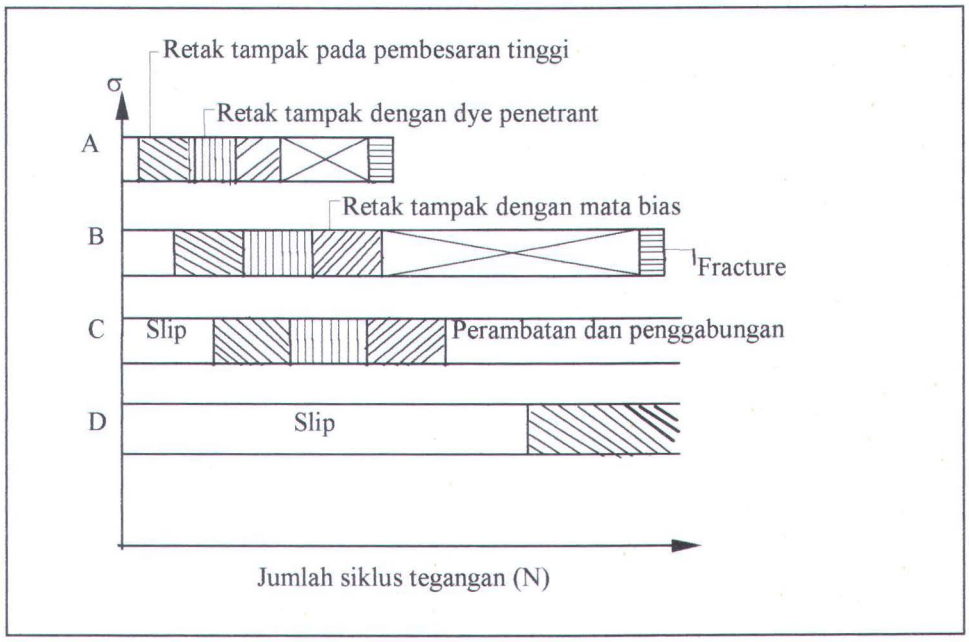
Secara skematis mekanisme fatigue dapat diringkas seperti pada gambar (3.7). Pertama-tama, pergeseran akan terjadi dan diikuti oleh retak halus yang dapat dilihat dengan tingkat pembesaran yang tinggi. Setelah itu retak akan merambat akibat pembebanan yang berulang hingga dapat diamati dengan mata biasa dan akan berkembang sedemikian hingga hanya beberapa retak utama saja yang akan terus merambat.

Deformasi plastis yang terjadi pada daerah ujung retak berada pada posisi ditengah-tengah material yang masih bersifat elastis. Pada saat pembebanan dilepaskan, bagian elastis ini akan mengalami penyusutan dan bagian plastis yang telah berdeformasi tersebut akan tidak berada dalam kondisi yang bersesuaian lagi dengan daerah di sekelilingnya.

Hal ini menyebabkan timbulnya tegangan kompresif oleh bagian elastis dari material terhadap bagian plastis pada saat pembebanan yang terjadi menuju minimum. Tegangan yang terjadi diujung retak ini akan mencapai tingkat diatas titik luluh sehingga

deformasi plastis pada arah yang sebaliknya akan terjadi dan ujung retak tersebut akan kembali menutup dan mengalami penajaman.

Pertumbuhan retak yang terus berlanjut akan berakhir dengan timbulnya fracture yang mengakibatkan kegagalan pada spesimen. Proses yang sama akan berlangsung lebih cepat dengan semakin besarnya tegangan yang diberikan. Pertumbuhan retak juga memiliki kemungkinan untuk terhenti bila terjadi swa-tegangan *(self-stress)* yang bersifat kompresif.



Gambar 3.8 Skema proses kelelahan

3.3.2. Karakteristik Kepatahan

a. Karakteristik Kepatahan secara Makroskopis

Pengamatan makroskopis pada permukaan kepatahan fatigue umumnya dilakukan secara visual baik langsung maupun dengan

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

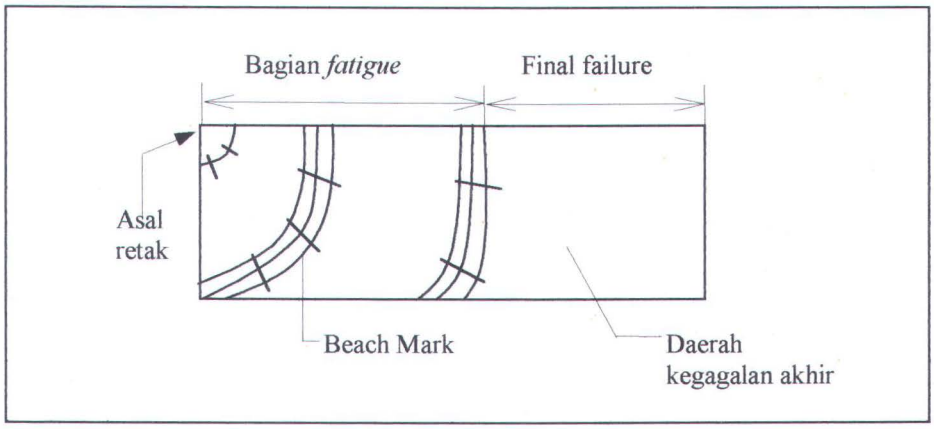
alat bantu. Pengamatan ini dapat langsung dilakukan di tempat kejadian dimana kegagalan terjadi dan tidak memerlukan persiapan khusus serta tidak merusak spesimen atau permukaan retak. Pengamatan visual dengan pembesaran sekitar 25 sampai 100 diameter biasanya telah cukup untuk memperlihatkan tempat asal retak *(crack origin)*. Beberapa hal yang dapat diperoleh dari pengamatan karakteristik kepatahan antara lain :

1. Informasi yang dapat mendukung penafsiran hasil tes yang diperoleh dari pengamatan pemukaan kepatahan.
2. Data yang diperlukan analisa grafis kepatahan *(fractographic)* yang merupakan hal yang sangat penting untuk menemukan kegagalan suatu komponen selama masa operasinya.

Beach Mark

Karakteristik yang paling jelas pada permukaan kepatahan kelelahan adalah adanya garis pantai *(beach mark)* yang umumnya mengelilingi tempat asal retak. Tanda ini merupakan karakteristik yang paling jelas untuk diamati dan paling penting dalam mengidentifikasi kegagalan komponen. Garis pantai terdapat pada permukaan kepatahan yang biasanya sangat datar, dimana kedua sisi komponen yang cacat disambung kembali, jika bagian akhir kegagalan tidak ada yang menghalangi. Beach mark ini terjadi akibat adanya perubahan pada pembebanan, perubahan frekuensi atau akibat timbulnya oksidasi pada permukaan kepatahan selama

periode penahanan retak. Pada kasus dimana kepatahan terjadi tanpa adanya penghentian pada penjalaran retak dan kondisi tanpa variasi pembebanan pengamatan tidak menunjukkan adanya beach mark.



Gambar 2.8 Dua pembagian pada permukaan kepatahan fatigue

Daerah Kepatahan Akhir

Pada kepatahan fatigue, daerah kepatahan akhir umumnya tampak berserat, hampir menyerupai pada kasus *impact* atau *fracture toughness* untuk material yang sama. Ukuran dari daerah ini tergantung pada besarnya pembebanan dan bentuknya ditentukan oleh bentuk, ukuran dan arah dari pembebanan pada daerah yang patah tersebut. Pada material yang cukup tangguh, daerah kepatahan akhir akan terdiri dari dua modus yang berbeda, yaitu daerah *plain strain* dimana kepatahan merambat dari daerah fatigue dan tetap berada pada bidang yang sama dan daerah *plain stress* dimana permukaannya membentuk sudut 45 derajat.

Pada material dengan ketangguhan cukup, bentuk kepatahan yang terjadi sedikit berbeda. Ketika retak merambat dari zona kelelahan, bidang kepatahan akan berputar menurut sumbu searah dengan penjalaran retak hingga membentuk sudut sekitar 45 derajat terhadap arah pembebanan. Gejala ini dapat terjadi baik pada bidang geser tunggal maupun bidang geser ganda.

Arah Penjalaran

Retak kelelahan menjalar dengan arah tegak lurus tegangan utama bila kecepatan rambat retak tidak terlalu tinggi. Untuk beban tarik berulang, arah penjalaran akan tegak lurus arah pembebanan. Pada pembebanan torsi berulang, arah penjalaran akan membentuk sudut 45 derajat dengan arah pembebanan sehingga menyebabkan arah penjalaran retak spiral. Jika retak kelelahan menjalar sangat cepat permukaan pergeseran terbentuk pada permukaan bebas seperti pada penjalaran retak statis *(static crack extension).* Lebar permukaan pergeseran akan bertambah selama penjalaran yang semakin cepat sampai kemudian meliputi tebal keseluruhannya.

Jejak Radial searah Penjalaran

Beberapa permukaan kepatahan menunjukkan jejak radial searah dengan penjalaran. Jejak seperti ini dapat merupakan akibat dari guncangan selama penjalaran retak. Secara umum jejak radial dan pita-pita penjalaran akan memudahkan dalam penentuan titik dimana retak dimulai. Jejak yang sama dalam arah yang tidak radial

juga dapat ditemukan dalam kejadian dimana terdapat beberapa retak yang terjadi *overlapping*.

Jumlah dan Ukuran Retak Makro

Jika beban fatigue rendah, retak inti akan terjadi pada beberapa titik terlemah saja, namun bila beban cukup tinggi retak inti bisa terjadi pada lebih banyak titik yang berpotensi untuk menimbulkan retak awal. Hal ini yang berhubungan dengan tingkat pembebanan adalah luasan dari bagian kelelahan. Untuk amplitudo tegangan yang lebih tinggi luasan tersebut akan lebih kecil karena σ maksimum menjadi lebih tinggi, demikian pula sebaliknya. Jadi luasan retak fatigue berbanding terbalik dengan dengan tegangan maksimum yang terjadi pada akhir umur kelelahan pada suatu komponen.

b. Karakteristik Kepatahan secara Mikroskopis

Pengamatan pada kepatahan fatigue dengan menggunakan mikroskop sampai dengan tingkat pembesaran tertentu menunjukkan adanya banyak kesulitan. Hal ini diakibatkan oleh ketinggian dari kontur permukaan patahan yang melebihi kemampuan kedalaman penglihatan pada mikroskop. Kesulitan ini terjadi terutama pada tingkat pembesaran yang tinggi.

Striation

Dengan menggunakan mikroskop elektron, karakteristik yang jelas dapat diamati adalah adanya garis-garis halus yang paralel pada permukaan patahan. Garis ini dinamakan *fatigue striation* dan mempunyai arah tegak lurus terhadap arah rambatan retak secara mikroskopis. Striation ini pada pembebanan yang tetap, mempunyai kecenderungan untuk memperbesar jarak spasinya selama retak merambat dari titik asal terjadinya kelelahan. Tiap-tiap garis yang terdapat pada striation menunjukkan pertumbuhan retak pada tiap pembebanan tunggal, jarak antar tiap-tiap garis tersebut merupakan laju penjalaran retak untuk setiap siklusnya. Garis striation tersebut juga menggambarkan posisi suksesif dari retak selama penjalarannya. Walaupun tiap garis striation dihasilkan dari sebuah siklus tegangan tunggal, namun tidak setiap siklus tegangan selalu menghasilkan garis-garis striation.

Striation tergantung pada besarnya tegangan yang bekerja sedangkan tingkat duktilitas material akan mempengaruhi jelas tidaknya tanda striation yang terjadi. Pada kondisi material yang ductile dan tegangan yang lebih tinggi dari fatigue limit, striation akan tampak lebih jelas. Striation akan sulit ditemukan pada material berkekuatan tinggi. Pada laju penjalaran retak yang tinggi striation akan tampak bergelombang dan menimbulkan permukaan yang kasar. Pada baja dengan laju penjalaran retak normal, striation

tidak selalu tampak jelas, sebaliknya pada aluminum paduan dengan duktilitas lebih tinggi striation tampak lebih jelas.

Mekanisme kelelahan pada material dengan kekuatan yang tinggi atau pada material yang brittle kemungkinan tidak akan menunjukkan adanya pembentukan pita-pita pergeseran. Beberapa karakteristik yang dapat diamati pada permukaan retak selain terbentuknya striation antara lain adalah penggabungan kekosongan mikro *(microvoid)* dan *microcleavage*.

Dalam hal ini retak mikro dapat terjadi secara langsung pada diskontinuitas yang terdapat pada material, seperti inklusi atau kekosongan *(void)*. Retak tersebut akan berkembang melalui bidang tegangan tarik yang maksimum. Seperti pada striation penggabungan microvoid yang cukup jelas dapat diamati pada material-material yang ductile.

BAB IV
PROSEDUR DAN PELAKSANAAN PENGUJIAN

BAB IV
PROSEDUR DAN PELAKSANAAN PENGUJIAN

4.1. PERSIAPAN MATERIAL UJI

4.1.1. Material

Material yang digunakan adalah *rolled steel for welded sructure* atau dalam klasifikasi material P1 group 2. Type spesifikasi material yang digunakan adalah baja karbon ASTM A 516 (SA 516) grade 70 dengan ketebalan 8 mm. Dalam aplikasinya material ini digunakan sebagai material dasar pressure vessel dan shell heat exchanger yang menentukan syarat kekuatan yang tinggi, terutama untuk daerah-daerah sambungan las. Sifat kimia dan sifat mekanis (chemical and mechanical properties) dari material ASTM A 516 grade 70 adalah sebagai berikut :

Tabel 4.1 Sifat Mekanis dan Komposisi Kimia Baja ASTM A 516 grade 70

Sifat Mekanis				Komposisi Kimia (%)						
TS Mpa	YS Mpa	A %	ε %	C	Mn	Si	P	S	Cr	Ni
509	366.7	23.5	17	0.23	1.10	0.29	0.015	0.026	0.07	0.06

Baja ini dikelompokkan dalam *Medium Carbon Steel* sesuai dengan kandungan karbonnya dan masuk dalam kelompok *High Tensile Steel* berdasarkan kekuatan tariknya. Mempunyai kekuatan

dan kekerasan yang tinggi akan tetapi kemampuan elastisitasnya rendah sehingga bersifat getas *(brittle)*. Baja ini juga mempunyai sifat yang sensitif terhadap retak, sekecil apapun retak yang timbul akan dapat menjalar dengan cepat dan menyebabkan kegagalan yang permanen pada komponen struktur. Sifat tersebut diperlukan dalam pengujian ini untuk mengetahui respon spesimen uji yang mengalami perubahan geometri, terhadap kekuatan menahan beban berulang.

Sifat mampu las baja ini termasuk dalam kategori sulit dilakukan sehingga untuk mendapatkan hasil pengelasan yan baik dibutuhkan persiapan dan prosedur yang khusus. Kandungan karbon yang tinggi menyebabkan daerah pengaruh panas (HAZ) sangat mudah menjadi keras dibandingkan pada baja dengan kadar karbon rendah. Sifat mudah menjadi keras dan ditambah dengan adanya difusi hidrogen menyebabkan baja ini sangat sensitif terhadap retak.

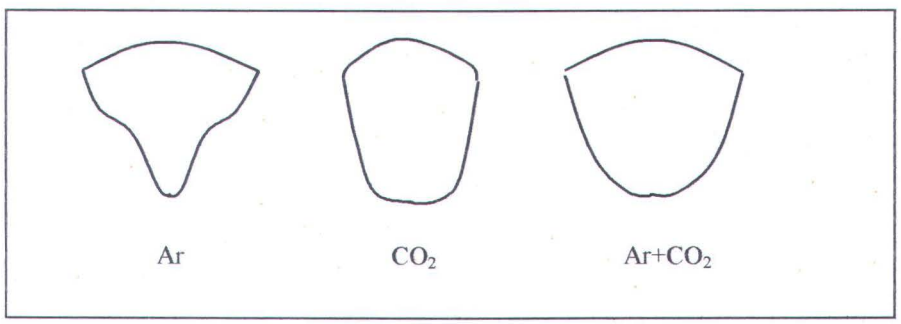
4.1.2. Pengelasan GMAW

Teknik pengelasan GMAW *(Gas Metal Arc Welding)* digunakan secara luas dalam dunia industri karena mempunyai sifat efisien dalam prosesnya, deposit weld metal tinggi dan bentuk manik lasnya bagus. Pengelasan GMAW mempunyai banyak variasi dalam pelaksanaannya, namun pada prinsipnya adalah merupakan

pengelasan busur listrik *(arc welding)* dengan menggunakan gas untuk melindungi cairan logam terhadap pengaruh udara luar.

Pada awal penggunaanya teknik pengelasan ini hanya digunakan pada pengelasan aluminium dan menggunakan Argon *(inert gas)* sebagai gas pelindungnya sehingga disebut MIG *(Metal Inert Gas welding)*. Kemudian penggunaannya diperluas untuk pengelasan baja dan digunakan gas CO_2 karena pemakaian Argon dinilai terlalu mahal. Namun pemakaian gas CO_2 mempunyai beberapa kelemahan diantaranya adalah timbul panas yang tinggi, kecepatan pengerjaan yang rendah dan manik las yang terbentuk kurang bagus.

Untuk memperoleh hasil yang lebih baik maka dilakukan perbaikan dengan menggunakan elektroda berdiameter 0.9 mm dan 1.1 mm yang kemudian dikenal dengan *Fine Wire Welding*. Agar diperoleh permukaan yang baik dan halus, penggunaan gas CO_2 diganti dengan campuran gas Argon dan CO_2. Penggunaan gas pelindung campuran ini terbukti memberikan hasil permukan las yang bagus.



Gambar 4.1 Permukaan hasil las dengan berbagai gas pelindung

Untuk pengujian ini pengelasan pada baja dilakukan dengan teknik GMAW dengan menggunakan gas pelindung campuran 80% Argon dan 20% CO_2. Elektroda yang digunakan dalam pengelasan adalah type AWS A5.18 ER70S-6, yang sesuai untuk pengelasan GMAW sambungan tumpul dan fillet pada baja berkekuatan tinggi. Berikut ini adalah komposisi kimia elektroda berdasarkan Inspection Certifacate [Kobe MIG Wire Co.Ltd].

Tabel 4.2 Komposisi Kimia Elektroda AWS A5.18 ER70S-6

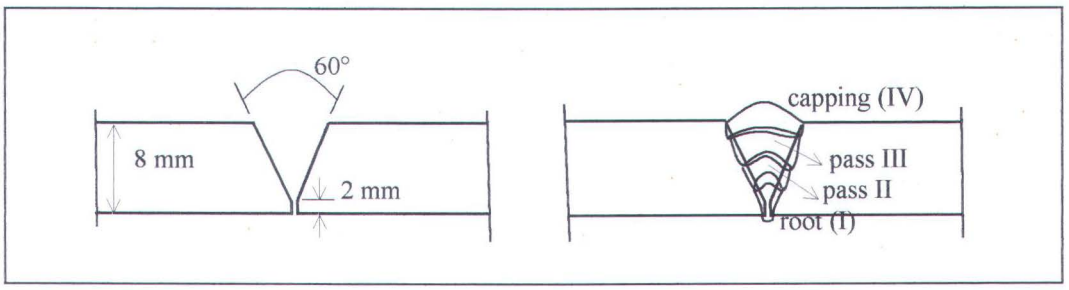
C (%)	Si (%)	Mn (%)	P (%)	S (%)	Cu (%)
0.11	0.84	1.56	0.011	0.016	0.12

Pengelasan dilakukan pada dua lembar pelat baja ASTM A 516 grade 70 berukuran 300 x 300 mm^2 dengan ketebalan 8 mm. Masing-masing pelat dipotong menjadi dua, melintang terhadap arah pengerolan material. Pemotongan ini dimaksudkan agar arah pengelasan dapat tegak lurus dengan arah pengerolan, sehingga pada pengujian fatigue pembebanan dapat searah dengan arah pengerolan material.

Pada sisi pelat yang akan dilakukan pengelasan, dilakukan persiapan sisi berupa *single-V-groove* dengan sudut sebesar 60 derajat agar penetrasi dapat berlangsung dengan sempurna.

Permukaan yang akan dilas dibersihkan dan dijaga dari karat, debu, minyak dan sebagainya karena dapat mempengaruhi mutu hasil pengelasan. Debu dan kotoran dapat terjebak dalam weld metal, minyak dan air mengandung partikel dan gas yang dapat terserap pada weld metal sehingga menimbulkan cacat difusi hidrogen, inklusi dan sebagainya.

Pengelasan yang dilakukan adalah untuk membentuk sambungan tumpul *(butt joint).* Pengelasan dilakukan dengan posisi mendatar *(flat)* secara semiotomatis, dimana selama proses pengelasan kawat elektroda diumpankan kebusur las terus-menerus secara otomatis kemudian busur las tersebut berjalan sepanjang arah pengelasan secara manual.



Gambar 4.2. Persiapan sisi dan urutan pass pengelasan

Untuk setiap sambungan dilakukan dengan 4 lapis pengelasan *(multi pass),* lapisan pertama merupakan *rooting* untuk membentuk akar las, kemudian dilanjutkan lapis kedua dan lapis ketiga. Sedangkan lapis keempat merupakan *capping* yaitu untuk membentuk permukaan manik las.

Tabel 4.3 Welding Parameter

Weld Layer	Proses	Filler		Arus (A)	Tegangan (V)	Speed (mm/menit)
		Type	Diam. (mm)			
1	GMAW	ER 705-S	1.2	130	20	200
2	GMAW	ER 705-S	1.2	150	20	200
3	GMAW	ER 705-S	1.2	150	20	170
4	GMAW	ER 705-S	1.2	150	20	150

Dalam eksperimen ini, walaupun bisa didapatkan hasil permukaan manik las yang halus namun pada saat capping dibuat agar ketinggian manik las cukup besar sehingga dapat dibedakan dengan manik yang mendapatkan perlakuan TIG Dressing.

4.1.3. Pengujian Radiografi

Keretakan yang terjadi pada sambungan las pada umumnya disebabkan karena cacat pengelasan dan karena konsentrasi tegangan. Dalam eksperimen ini diambil suatu asumsi bahwa keretakan fatigue yang terjadi pada sambungan las merupakan akibat konsentrasi tegangan, bukan karena adanya cacat pengelasan. Oleh karena itu dalam pelaksanaan pengelasan dijaga dan dihindari agar tidak menimbulkan cacat pada hasil pengelasan.

Untuk mendukung asumsi tersebut maka dilakukan pengujian tidak merusak untuk mengetahui mutu hasil pengelasan. Pengujian radiografi sinar X dilakukan pada sambungan las material 1 (M1) dan material 2 (M2), untuk mengetahui apakah timbul cacat

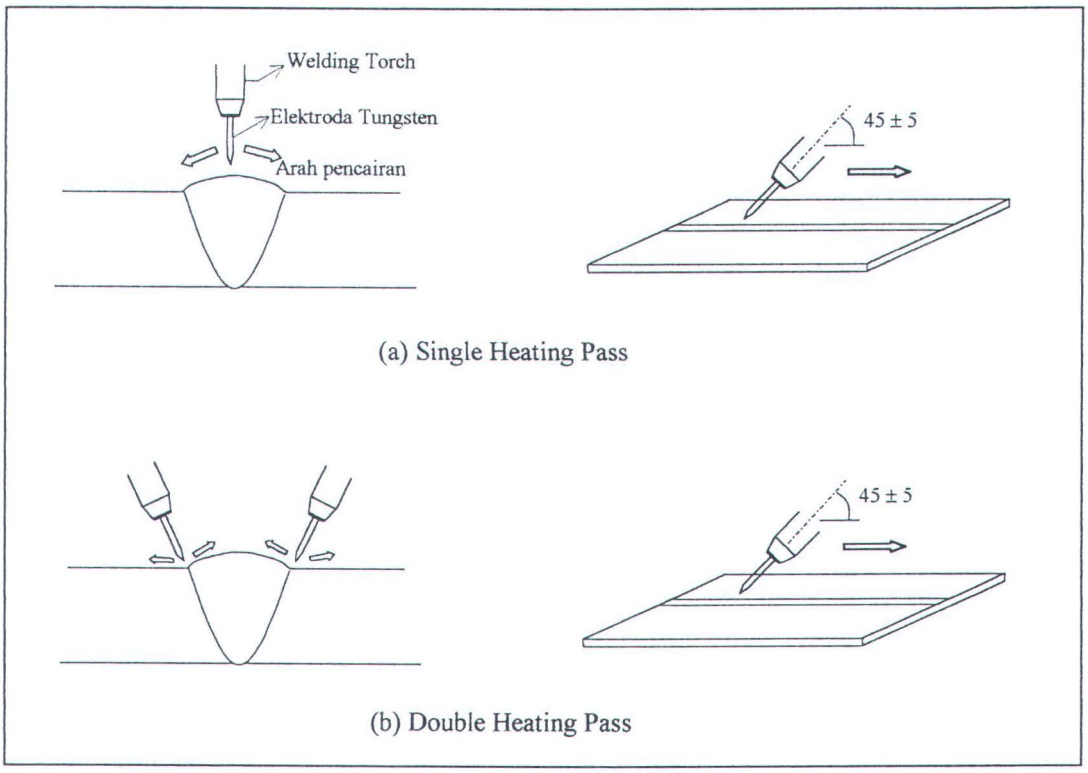
pada sambungan tersebut. Berdasarkan pengamatan film hasil pengujian dapat diketahui bahwa pengelasan tidak menimbulkan cacat yang berarti.

4.1.4. Perlakuan TIG

Perlakuan TIG (TIG Dressing) dilaksanakan dengan tujuan untuk membentuk manik las terutama pada titik toe agar mempunyai jari-jari toe dan flank angle yang besar. Peningkatan jari-jari toe dan flank angle diharapkan dapat mengurangi konsentrasi tegangan pada sambungan las, karena sambungan las yang menimbulkan perubahan geometri pada material merupakan daerah konsentrasi tegangan. Dengan TIG Dressing, konsentrasi tegangan tersebut akan diturunkan sehingga kekuatan fatigue sambungan las dapat ditingkatkan.

TIG Dressing dilakukan dengan teknik pengelasan TIG *(Tungsten Inert Gas)* atau dikenal juga dengan pengelasan GTAW, yaitu teknik pengelasan dengan elektroda tidak ikut mencair *(non consumable)*. Untuk melindungi cairan logam dari kontaminasi udara luar digunakan gas mulia (Argon atau Helium) sebagai gas pelindung. Panas yang ditimbulkan oleh elektroda tungsten digunakan untuk mencairkan kembali *(remelting)* manik las dan membentuknya agar jari-jari toe dan flank agle-nya meningkat.

Ada dua teknik dalam perlakuan TIG yaitu *double heating pass* dan *singgle heating pass*. Double heating pass merupakan teknik konvensional yang telah banyak dilakukan untuk meningkatkan kekuatan fatigue sambungan las terutama pada material yang tebal. Perlakuan TIG dilaksanakan pada salah satu sisi manik las hingga terbentuk jari-jari toe (R) dan flank angle (θ) yang lebih besar, kemudian baru dilanjutkan pada sisi yang lainnya.



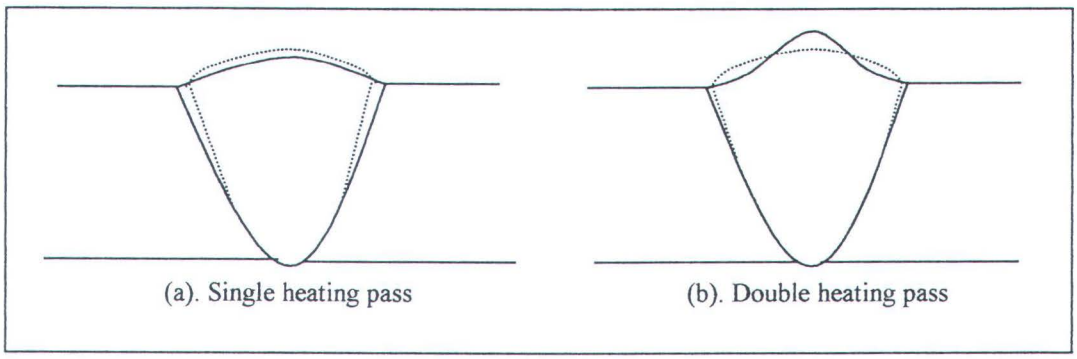
Gambar 4.3 Perbedaan teknik perlakuan TIG

Single heating pass merupakan modifikasi dari teknik yang pertama, untuk tujuan yang sama teknik ini dilakukan dengan satu kali proses pencairan pada seluruh manik las. Panas yang dibutuhkan lebih tinggi dan kecepatan pencairan lebih rendah bila

dibandingkan dengan teknik double heating pass. Akan tetapi hanya dibutuhkan satu kali proses untuk satu sambungan sedangkan pada teknik konvensional harus dua kali proses untuk satu sambungan, sehingga secara umum teknik ini prosesnya lebih cepat.

Tabel 4.4 Welding Parameter TIG Dressing

Parameter	Double Heating Pass	Single Heating Pass
Arus (A)	150	200
Tegangan (V)	14	14
Kecepatan (mm/menit)	144	106
Diameter Elektroda (mm)	2	4



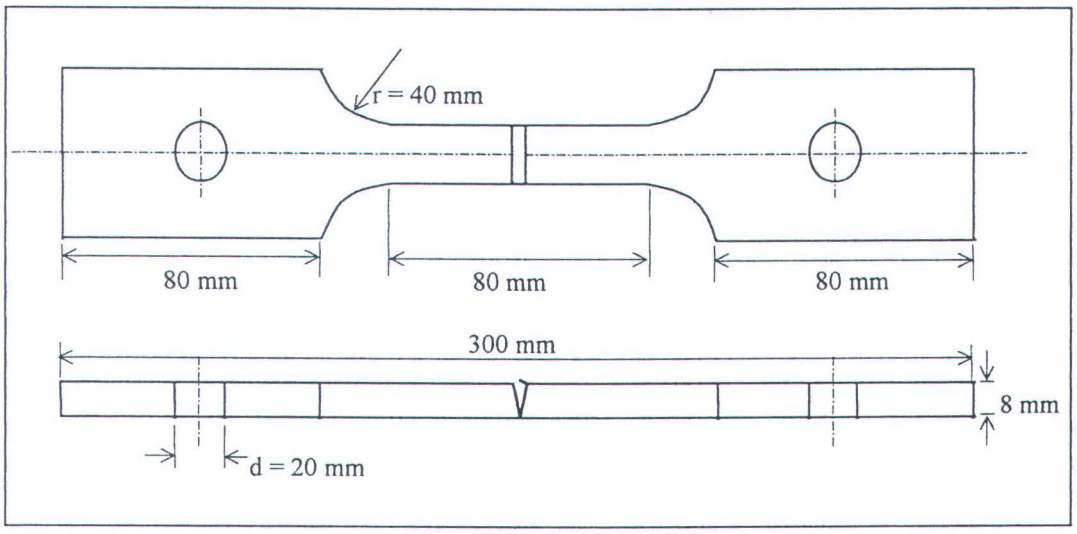
Gambar 4.4 Bentuk manik las hasil perlakuan TIG

Bentuk manik las yang dihasilkan dari kedua teknik perlakuan tersebut dapat dilihat pada gambar (4.4). Teknik single heating pass akan membentuk manik menjadi lebih lebar sehingga ketinggiannya berkurang sedangkan teknik double heating pass akan mempertinggi manik las pada bagian tengahnya. Akan tetapi perbedaan bentuk manik hasil perlakuan tersebut tidak mengurangi

tujuan utama dari perlakuan TIG yaitu memperbesar jari-jari toe dan flank angle.

4.1.4. Persiapan Akhir Material Uji

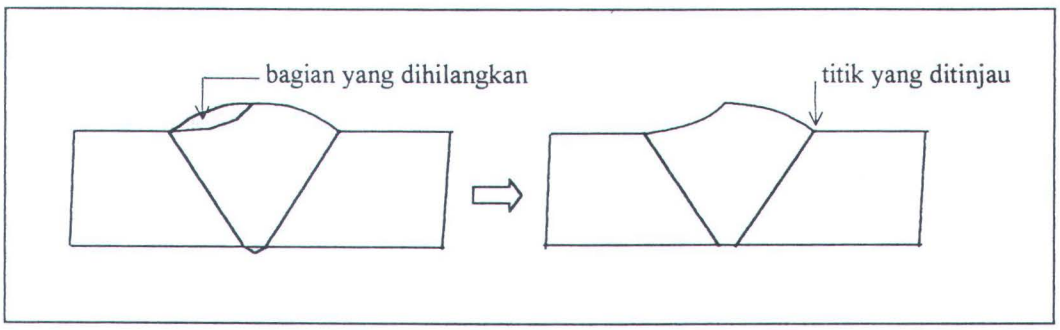
Setelah proses pengelasan selesai maka material harus dibentuk menjadi spesimen uji fatigue. Material yang telah mengalami proses pengelasan, baik yang mendapat perlakuan TIG maupun yang tidak (as welded), dipotong melintang terhadap arah pengelasan dengan lebar 60 mm. Potongan-potongan tersebut kemudian disekrap menjadi bentuk spesimen uji fatigue sesuai standart ASTM E 606 *(Standard Practice for Strain-Controlled Fatigue Testing).*



Gambar 4.5 Dimensi spesimen uji fatigue

Untuk memfokuskan pengamatan dan melokalisasi daerah terjadinya keretakan fatigue hanya pada salah satu titik toe maka

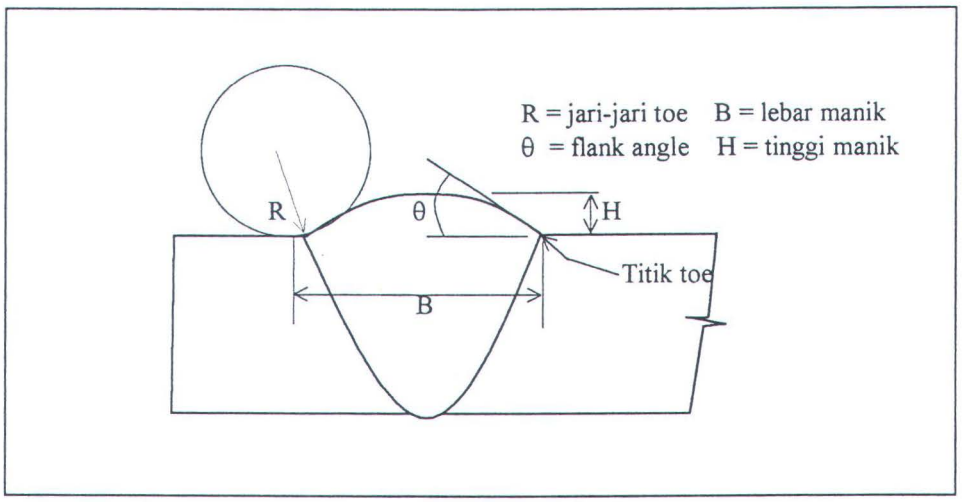
sisi lain dari manik las dihilangkan, demikian pula pada akar manik las. Untuk menghilangkan sisi manik tersebut digunakan kikir dan dihaluskan dengan kertas gosok, proses ini harus dilakukan dengan cermat dan hati-hati agar jangan sampai timbul goresan yang justru dapat menyebabkan konsentrasi tegangan. Sedangkan pada sisi manik las yang ditinjau, tidak dilakukan proses apapun agar kondisi yang sesungguhnya tetap terjaga.



Gambar 4.6 Proses untuk melokalisasi daerah keretakan awal

4.2. PENGUKURAN JARI-JARI TOE DAN FLANK ANGLE

Konsentrasi tegangan yang terjadi pada sambungan las sangat ditentukan oleh geometrinya. Pada sambungan las tumpul, konsentrasi tegangan terjadi karena kehadiran manik las yang menyebabkan perubahan bentuk pada permukaan material induk. Pusat konsentrasi tegangan terletak pada titik toe, yaitu titik pada permukaan yang terletak pada garis batas antara logam induk dan logam las.

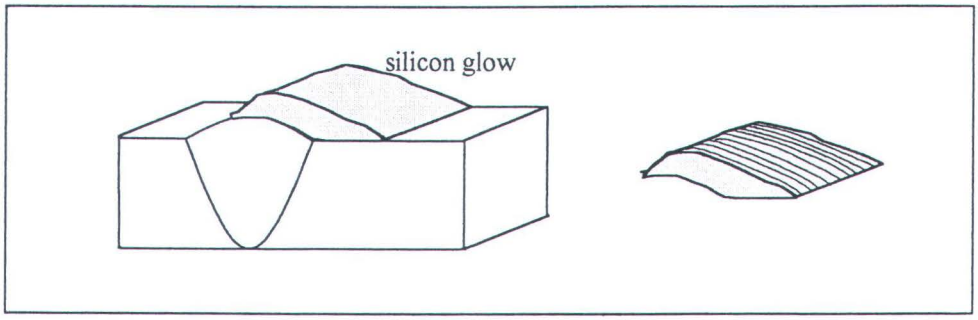


Gambar 4.7 Geometri sambungan las

Untuk mengukur jari-jari toe dan flank angle yang dimensinya cukup kecil, dibutuhkan pembesaran agar diperoleh data yang akurat. Pembesaran dilakukan dengan menggunakan *Profile Projector,* dengan tingkat pembesaran 10 kali untuk pengukuran jari-jari toe dan 20 kali untuk flank angle. Prosedur pengukuran adalah sebagai berikut :

1. Pada sisi manik las spesimen yang akan diukur dibuat cetakan profilnya dengan menggunakan *silicon glow.*
2. Setelah cetakan mengeras, cetakan dilepas dan dipotong menjadi 10 irisan dengan lebar masing-masing 2 mm.
3. Masing-masing irisan ditandai sesuai dengan posisinya pada manik las, kemudian diukur dengan pembesaran menggunakan profile projector.
4. Pengukuran sudut kemiringan manik (flank angle) dilakukan dengan pembesaran 10 kali. Flank angle diperoleh dengan

cara menggambar garis singgung kemiringan profil pada kertas transparan dan diukur besarnya sudut yang terbentuk.



Gambar 4. Pembuatan cetakan profil manik las

5. Pengukuran jari-jari toe menggunakan tingkat pembesaran 20 kali. Jari-jari diukur dengan menggunakan lingkaran-lingkaran yang dibuat pada kertas transparan dan dihimpitkan pada kelengkungan yang terjadi pada titik toe. Besarnya jari-jari toe dapat kita ketahui berdasarkan jari-jari lingkaran yang paling sesuai dengan kelengkungan pada titik toe tersebut.

Data hasil pengukuran flank angle dan jari-jari toe untuk masing-masing spesimen dapat dilihat pada tabel (4.5). Untuk spesimen yang tidak diberikan perlakuan TIG (as welded) diberi tanda A, spesimen double heating pass diberi tanda D dan spesimen single heating pass ditandai dengan huruf S.

Tabel 4.5 Data Pengukuran Flank Angle (derajat)

No.	TYPE SPESIMEN								
	A1	A2	A3	D1	D2	D3	S1	S2	S3
1	32.32	22.16	17.95	13.33	11.20	9.02	12.66	24.16	12.05
2	30.75	21.58	18.95	14.77	11.64	9.78	16.06	20.77	12.14
3	21.37	21.00	19.94	16.20	12.08	10.53	19.46	17.38	12.22
4	25.5	21.05	21.35	14.13	10.24	11.48	16.46	17.34	12.39
5	18.95	21.10	22.75	12.06	8.40	12.43	13.45	17.30	12.55
6	27.97	22.05	19.38	12.67	10.80	12.37	18.94	14.38	13.48
7	28.93	23.00	16.00	13.27	13.20	12.31	24.43	11.45	14.41
8	25.19	25.60	18.43	12.68	12.78	12.21	20.98	15.28	13.30
9	19.45	28.20	20.85	12.09	12.35	12.10	17.52	19.10	12.18
10	29.81	25.18	19.40	12.71	11.78	10.56	15.09	21.63	12.12

Tabel 4.6 Data Pengukuran Jari-Jari Toe (milimeter)

No.	TYPE SPESIMEN								
	A1	A2	A3	D1	D2	D3	S1	S2	S3
1	0.800	0.864	1.255	3.533	3.133	3.400	1.302	0.875	1.120
2	0.875	0.932	1.055	3.200	3.293	3.360	1.140	1.031	1.160
3	0.950	1.000	0.851	2.867	3.467	3.333	0.977	1.188	1.200
4	0.865	1.059	0.979	3.107	3.933	3.467	0.977	1.050	1.100
5	0.775	1.114	1.106	3.333	4.400	3.600	0.977	0.906	0.965
6	0.940	1.091	0.987	3.733	3.907	3.573	0.767	0.844	1.080
7	1.100	1.068	0.872	4.133	3.400	3.533	0.814	0.781	1.160
8	0.975	0.923	0.894	3.707	3.773	3.400	0.733	1.081	1.080
9	0.850	0.773	0.915	3.267	4.133	3.267	0.907	1.375	0.880
10	0.825	0.818	1.085	3.400	3.640	3.333	1.105	1.125	1.060

4.3. PENGUJIAN FATIGUE

Pengujian fatigue akan dilakukan pada masing-masing spesimen untuk mengetahui umur keretakan awal dan umur fatigue spesimen. Pengujian dilakukan dengan menggunakan *axial load fatigue testing machine.*

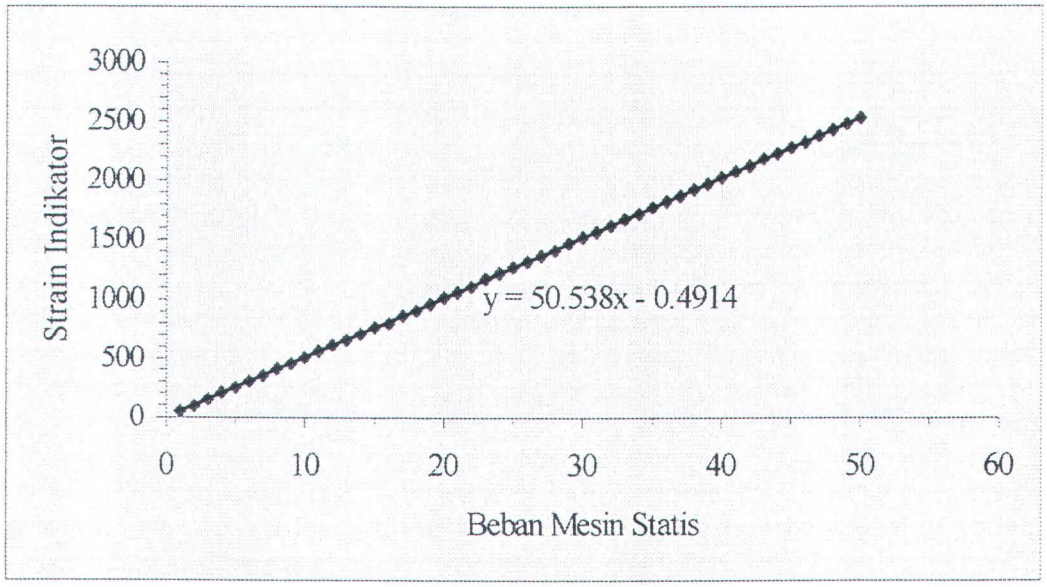
4.3.1. Kalibrasi Beban

Sebelum pengujian maka perlu dilakukan kalibrasi beban dari mesin uji statis-dinamis untuk mendapatkan kesetaraan load cell dari mesin uji statis dengan indikator pada mesin uji dinamis, sehingga diketahui besar strain indikator mesin sesuai dengan beban yang diterapkan. Adapun kalibrasi beban adalah sebagai berikut :

Tabel 4.7 Kalibrasi beban mesin uji dinamis

No.	Beban Mesin Statis (kN)	Strain Indikator
1	1	48
2	2	96
3	3	153
4	4	207
5	5	249
6	6	301
7	7	354
8	8	407
9	9	454
10	10	504
11	11	557
12	12	608
13	13	654
14	14	705
15	15	758
16	16	796
17	17	861
18	18	906
19	19	961
20	20	1016
21	21	1059
22	22	1105
23	23	1173
24	24	1216
25	25	1270

No.	Beban Mesin Statis (kN)	Strain Indikator
26	26	1311
27	27	1365
28	28	1410
29	29	1470
30	30	1518
31	31	1566
32	32	1615
33	33	1671
34	34	1717
35	35	1769
36	36	1820
37	37	1868
38	38	1920
39	39	1969
40	40	2020
41	41	2074
42	42	2117
43	43	2178
44	44	2222
45	45	2272
46	46	2320
47	47	2376
48	48	2427
49	49	2475
50	50	2524



Gambar 4.8 Grafik kalibrasi beban mesin uji dinamis

4.3.2. Penentuan Beban

Beban yang diterapkan pada spesimen adalah beban axial dengan rentang beban dibedakan dalam tiga tingkatan. Pada spesimen kelompok 1 (A1, D1 dan S1) diberikan beban sebesar 0.5 σ_y, kelompok 2 (A2, D2 dan D3) diberikan beban 0.75 σ_y dan untuk spesimen kelompok 3 (A3, D3 dan S3) dibebani sebesar σ_y. Dengan pola pembebanan seperti ini diharapkan dapat diperoleh data yang menggambarkan kekuatan fatigue pada tiap jenis spesimen.

Beban untuk Spesimen Kelompok 1

- Beban yang diterapkan :

  (0.5). σ_y = (0.5). 366.7

  = 183.35 N/mm^2

- Besarnya gaya yang diterapkan :

$F_{app.}$ = 183.35 (20 x 8)

= 29336 N

= 29.336 kN

- Beban pada mesin uji fatigue :

y = 50.538 x – 0.4914

= 50.538 (29.336) – 0.4914

= 1482.09

Beban untuk Spesimen Kelompok 2

- Beban yang diterapkan :

(0.75). σ_y = (0.75). 366.7

= 275.025 N/mm^2

- Gaya yang diterapkan :

$F_{app.}$ = 275.025 (20 x 8)

= 44.004 kN

- Beban pada mesin uji fatigue :

y = 50.538 (44.004) – 0.4914

= 2223.38

Beban untuk Spesimen Kelompok 3

- Beban yang diterapkan :

σ_y = 366.7

- Gaya yang diterapkan :

$F_{app.}$ = 366.7 (20 x 8)

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
PEPULUH

= 58.672 kN

- Beban pada mesin uji fatigue :

y = 50.538 (58.672) – 0.4914

= 2964.67

Rasio pembebanan yaitu perbandingan antara beban minimum dan beban maksimum ditetapkan sama dengan nol (R=0). Dengan rasio pembebanan ini berarti pembebanan dari nol, kemudian beban tarik sebesar rentang beban setelah itu beban dilepaskan ke nol lagi, jenis pembebanan ini dikenal dengan *pulsating tension*. Sedangkan frekuensi atau kecepatan pembebanan sesuai dengan kemampuan dan kondisi mesin uji, maka hanya dapat ditetapkan sebesar 2.5 Hz (siklus per detik).

Pencatatan yang dilakukan pada pengujian fatigue ini adalah jumlah siklus mulai terjadinya keretakan dan jumlah siklus hingga terjadi kepatahan pada spesimen. Selain itu posisi timbulnya keretakan juga dicatat agar dapat dianalisa sebab-sebab timbulnya keretakan tersebut.

BAB V
ANALISA DATA HASIL PENGUJIAN

BAB V
ANALISA DATA HASIL PENGUJIAN

5.1. PERHITUNGAN FAKTOR KONSENTRASI TEGANGAN

Pengujian fatigue yang dilakukan pada spesimen bertujuan untuk mengetahui respon material uji terhadap pembebanan berulang. Faktor yang dibedakan pada eksperimen ini adalah konsentrasi tegangan pada masing-masing spesimen, dengan cara memberikan perlakuan TIG Dressing pada manik las sehingga konsentrasi tegangan yang terjadi pada sambungan las tersebut menjadi lebih rendah.

Perlakuan TIG Dressing ini merubah harga faktor konsentrasi tegangan sambungan, faktor konsentrasi tegangan ini merupakan faktor yang meningkatkan tegangan nominal menjadi tegangan lokal pada sambungan las. Faktor konsentrasi tegangan pada sambungan las ditentukan oleh parameter geometri sambungan itu sendiri yang meliputi jari-jari toe *(R)*, flank angle *(θ)*, ketinggian manik las *(H)*, lebar manik las *(B)* dan ketebalan material induk *(T)*.

Geometri sambungan las bisa diketahui melalui pengukuran pada manik las secara langsung maupun dengan bantuan pembesaran, kemudian parameter geometri ini digunakan untuk menghitung besarnya faktor konsentrasi tegangan yang terjadi pada

sambungan las tersebut. Hasil pengukuran tinggi manik las, lebar manik las, flank angle dan jari-jari toe yang terdapat pada tabel (4.5), (4.6) dan (4.7) akan digunakan untuk mengukur besarnya faktor konsentrasi tegangan dengan menggunakan persamaan sebagai berikut :

$$K_t = 1 + f(\theta).\left[\gamma.\left(\frac{1}{(1.4).\Delta i - 2}.\frac{H}{R}\right)^{\beta}\right]$$

dimana, $f(\theta) = \dfrac{1-\exp\left((-0.90).\theta.\sqrt{\Delta i}\right)}{1-\exp\left((-0.45).\pi.\sqrt{\Delta i}\right)}$ $\qquad \gamma = 1 - 0.48\exp\left(-0.74\dfrac{B}{T}\right)$

$$\beta = 0.65 - 0.1\exp\left(-0.63\frac{B}{T}\right) \qquad \Delta i = \frac{2H+T}{2T}$$

Faktor konsentrasi tegangan setiap spesimen dihitung sesuai dengan data yang diperoleh pada pengukuran flank angle dan jari-jari toe, sesuai dengan posisi dimana cetakan dibuat. Dengan mengetahui faktor konsentrasi tegangan pada masing-masing posisi untuk setiap spesimen, dapat digunakan untuk menganalisa daerah terjadinya keretakan awal, apakah sesuai dengan teori bahwa keretakan awal mulai terjadi pada titik yang mempunyai faktor konsentrasi tegangan yang tertinggi atau akan terjadi fenomena yang berbeda.

Tabel 5.1 Faktor konsentrasi tegangan pada spesimen as welded

Spesimen A1
T=8 B=15.5 H=1.4

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	32.32	0.800	0.675	0.858378	0.620495	1.456038	2.710911
2	30.75	0.875	0.675	0.858378	0.620495	1.456038	2.618374
3	21.37	0.950	0.675	0.858378	0.620495	1.456038	2.537862
4	25.50	0.865	0.675	0.858378	0.620495	1.456038	2.629958
5	18.95	0.775	0.675	0.858378	0.620495	1.456037	2.744948
6	27.97	0.940	0.675	0.858378	0.620495	1.456038	2.547994
7	28.93	1.100	0.675	0.858378	0.620495	1.456038	2.404142
8	25.19	0.975	0.675	0.858378	0.620495	1.456038	2.513274
9	19.45	0.850	0.675	0.858378	0.620495	1.456037	2.647745
10	29.81	0.825	0.675	0.858378	0.620495	1.456038	2.678553

Spesimen A2
T=8 B=15.5 H=1.4

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	22.16	0.864	0.675	0.858378	0.620495	1.456038	2.631554
2	21.58	0.932	0.675	0.858378	0.620495	1.456038	2.556413
3	21.00	1.000	0.675	0.858378	0.620495	1.456038	2.489687
4	21.05	1.059	0.675	0.858378	0.620495	1.456038	2.437554
5	21.10	1.114	0.675	0.858378	0.620495	1.456038	2.393449
6	22.05	1.091	0.675	0.858378	0.620495	1.456038	2.411391
7	23.00	1.068	0.675	0.858378	0.620495	1.456038	2.429950
8	25.60	0.923	0.675	0.858378	0.620495	1.456038	2.565910
9	28.20	0.773	0.675	0.858378	0.620495	1.456038	2.748132
10	25.18	0.818	0.675	0.858378	0.620495	1.456038	2.687219

Spesimen A3
T=8 B=15.5 H=1.4

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	17.95	1.255	0.675	0.858378	0.620495	1.456036	2.293653
2	18.95	1.055	0.675	0.858378	0.620495	1.456037	2.440739
3	19.94	0.851	0.675	0.858378	0.620495	1.456038	2.646467
4	21.35	0.979	0.675	0.858378	0.620495	1.456038	2.509699
5	22.75	1.106	0.675	0.858378	0.620495	1.456038	2.399110
6	19.38	0.987	0.675	0.858378	0.620495	1.456037	2.501610
7	16.00	0.872	0.675	0.858378	0.620495	1.456028	2.621422
8	18.43	0.894	0.675	0.858378	0.620495	1.456036	2.597368
9	20.85	0.915	0.675	0.858378	0.620495	1.456038	2.574216
10	19.40	1.085	0.675	0.858378	0.620495	1.456037	2.416069

Tabel 5.2 Faktor konsentrasi tegangan pada spesimen double heating pass

Spesimen D1
T=8 B=18 H=1.65 H=1.65

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	13.33	3.533	0.70625	0.883687	0.625768	1.438774	1.783987
2	14.77	3.200	0.70625	0.883687	0.625768	1.438814	1.834162
3	16.20	2.867	0.70625	0.883687	0.625768	1.438827	1.893613
4	14.13	3.107	0.70625	0.883687	0.625768	1.438801	1.849750
5	12.06	3.333	0.70625	0.883687	0.625768	1.438677	1.813046
6	12.67	3.733	0.70625	0.883687	0.625768	1.438735	1.757414
7	13.27	4.133	0.70625	0.883687	0.625768	1.438771	1.710695
8	12.68	3.707	0.70625	0.883687	0.625768	1.438736	1.760820
9	12.09	3.267	0.70625	0.883687	0.625768	1.438681	1.823392
10	12.71	3.400	0.70625	0.883687	0.625768	1.438738	1.803067

Spesimen D2
T=8 B=18 H=1.65

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	11.20	3.133	0.70625	0.883687	0.625768	1.438533	1.845060
2	11.64	3.293	0.70625	0.883687	0.625768	1.438618	1.819178
3	12.08	3.467	0.70625	0.883687	0.625768	1.438679	1.793335
4	10.24	3.933	0.70625	0.883687	0.625768	1.438211	1.732813
5	8.40	4.400	0.70625	0.883687	0.625768	1.436329	1.682267
6	10.80	3.907	0.70625	0.883687	0.625768	1.438427	1.736049
7	13.20	3.400	0.70625	0.883687	0.625768	1.438768	1.803084
8	12.78	3.773	0.70625	0.883687	0.625768	1.438743	1.752384
9	12.35	4.133	0.70625	0.883687	0.625768	1.438708	1.710664
10	11.78	3.640	0.70625	0.883687	0.625768	1.438640	1.769459

Spesimen D3
T=8 B=18 H=1.65

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	9.02	3.400	0.70625	0.883687	0.625768	1.437267	1.802246
2	9.78	3.360	0.70625	0.883687	0.625768	1.437952	1.808594
3	10.53	3.333	0.70625	0.883687	0.625768	1.438334	1.812852
4	11.48	3.467	0.70625	0.883687	0.625768	1.438591	1.793286
5	12.43	3.600	0.70625	0.883687	0.625768	1.438715	1.774838
6	12.37	3.573	0.70625	0.883687	0.625768	1.438710	1.778449
7	12.31	3.533	0.70625	0.883687	0.625768	1.438704	1.783949
8	12.21	3.400	0.70625	0.883687	0.625768	1.438694	1.803042
9	12.10	3.267	0.70625	0.883687	0.625768	1.438682	1.823393
10	10.56	3.333	0.70625	0.883687	0.625768	1.438345	1.812858

Tabel 5.3 Faktor konsentrasi tegangan pada spesimen single heating pass

Spesimen S1

T=8 B=23.5 H=0.95

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	12.66	1.302	0.61875	0.924573	0.634286	1.490285	2.041685
2	16.06	1.140	0.61875	0.924573	0.634286	1.490458	2.133889
3	19.46	0.977	0.61875	0.924573	0.634286	1.490474	2.250370
4	16.46	0.977	0.61875	0.924573	0.634286	1.490463	2.250360
5	13.45	0.977	0.61875	0.924573	0.634286	1.490366	2.250279
6	18.94	0.767	0.61875	0.924573	0.634286	1.490473	2.457036
7	24.43	0.814	0.61875	0.924573	0.634286	1.490475	2.403662
8	20.98	0.733	0.61875	0.924573	0.634286	1.490475	2.500672
9	17.52	0.907	0.61875	0.924573	0.634286	1.490469	2.310543
10	15.09	1.105	0.61875	0.924573	0.634286	1.490441	2.156458

Spesimen S2

T=8 B=23.5 H=0.95

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	24.16	0.875	0.61875	0.924573	0.634286	1.490475	2.340727
2	20.77	1.031	0.61875	0.924573	0.634286	1.490475	2.208037
3	17.38	1.188	0.61875	0.924573	0.634286	1.490469	2.104628
4	17.34	1.050	0.61875	0.924573	0.634286	1.490469	2.194304
5	17.30	0.906	0.61875	0.924573	0.634286	1.490468	2.311209
6	14.38	0.844	0.61875	0.924573	0.634286	1.490419	2.371962
7	11.45	0.781	0.61875	0.924573	0.634286	1.490026	2.440216
8	15.28	1.081	0.61875	0.924573	0.634286	1.490446	2.172275
9	19.10	1.375	0.61875	0.924573	0.634286	1.490473	2.006544
10	21.63	1.125	0.61875	0.924573	0.634286	1.490475	2.143172

Spesimen S3

T=8 B=23.5 H=0.95

Posisi	etha	r	delta	gamma	betha	f(etha)	Kt
1	12.05	1.120	0.61875	0.924573	0.634286	1.490181	2.146180
2	12.14	1.160	0.61875	0.924573	0.634286	1.490200	2.120964
3	12.22	1.200	0.61875	0.924573	0.634286	1.490215	2.097128
4	12.39	1.100	0.61875	0.924573	0.634286	1.490244	2.159404
5	12.55	0.965	0.61875	0.924573	0.634286	1.490269	2.259827
6	13.48	1.080	0.61875	0.924573	0.634286	1.490369	2.173075
7	14.41	1.160	0.61875	0.924573	0.634286	1.490420	2.121130
8	13.30	1.080	0.61875	0.924573	0.634286	1.490354	2.173063
9	12.18	0.880	0.61875	0.924573	0.634286	1.490207	2.335650
10	12.12	1.060	0.61875	0.924573	0.634286	1.490196	2.186928

5.2. PERHITUNGAN UMUR KERETAKAN AWAL

Umur keretakan awal didefinisikan sebagai jumlah siklus pembebanan hingga mulai terjadi keretakan pada material atau spesimen. Pada kondisi yang sebenarnya keretakan awal ini merupakan gejala mikroskopis pada material yang diberikan beban berulang. Oleh karena itu proses terjadinya keretakan awal hanya dapat diamati dengan menggunakan pembesaran mikroskop dan sukar untuk ditentukan kapan terjadinya keretakan awal ini. Akan tetapi dalam ekperimen ditetapkan bahwa keretakan awal terjadi apabila panjang retak yang timbul berukuran 0.2 – 0.5 mm (0.008 - 0.02 inchi) [Itoh, 1987].

Dengan menggunakan aturan Neuber dan hubungan tegangan-regangan material, didapatkan bahwa umur keretakan awal *(N_c)* pada suatu spesimen yang bertakik atau mempunyai diskontinuitas *($K_t > 1$)*, dapat diprediksi dengan menggunakan fungsi tegangan-regangan *(D)* material tanpa takik atau material induk *($K_t = 1$)*. Hubungan antara fungsi tegangan-regangan dengan umur keretakan awal material tanpa takik ini dapat digambarkan pada kurva *D-N_c*.

Untuk memperoleh persamaan fungsi tegangan-regangan material induk, dapat dilakukan dengan percobaan laboratorium. Selain dengan cara percobaan, konstanta-konstanta yang menyusun persamaan fungsi ini bisa didapatkan dari rumus prediksi

berdasarkan parameter-parameter material induk [Osgood, 1979]. Rumus prediksi ini digunakan untuk memperoleh konstanta-konstanta dalam persamaan berikut :

$$D = 2\left[F.(N_c)^f + G.(N_c)^g\right]^{1/2}$$

dengan : $F = A^2.E^2$ dan $f = 2.a$

$G = A.B.E^2$ dan $g = a + b$

Dalam tabel (3.1) dijelaskan definisi dan rumus untuk memprediksi parameter material pada pembebanan siklus dengan menggunakan parameter material hasil pengujian beban statis. Dengan menggunakan rumus prediksi tersebut akan dicari konstanta *F, G, f* dan *g* berdasarkan pada parameter material baja ASTM A 516 grade 70 sebagai berikut :

- $A = \frac{\sigma'_f}{E}$; σ'_f dapat diprediksi dengan menggunakan kekuatan fracture material pada pembebanan statis (σ_f). Dari kurva tegangan-regangan pengujian statis material diketahui kekuatan fracture baja ASTM A 516 grade 70 adalah sebesar 372,54 MPa dan modulus elastisitas (Young) material adalah 206,7 x 10^9 N/mm atau 206.700 MPa.
- $B = \varepsilon'_f$; ε'_f diprediksi sama dengan regangan fracture material pada pembebanan statis (ε_f).

ε_f = ln (A_o/A_f) = ln $\{100/(100 - \%\ RA)\}$; dengan A_o adalah penampang spesimen asli, A_f adalah penampang minimum setelah terjadi fracture dan *RA* adalah reduction of area pada spesimen yang diukur setelah terjadi kepatahan. Persentase *RA* untuk material ini diketahui sebesar 23,5 %.

- a = eksponen kekuatan fatigue

  $a = -\ 1/6 \log 2\ (\sigma_f/\sigma_u)$; dengan σ_f kekuatan fracture statis dan σ_u adalah kekuatan ultimate statis baja ASTM A 516 grade 70 yang diketahui sebesar 509 MPa.

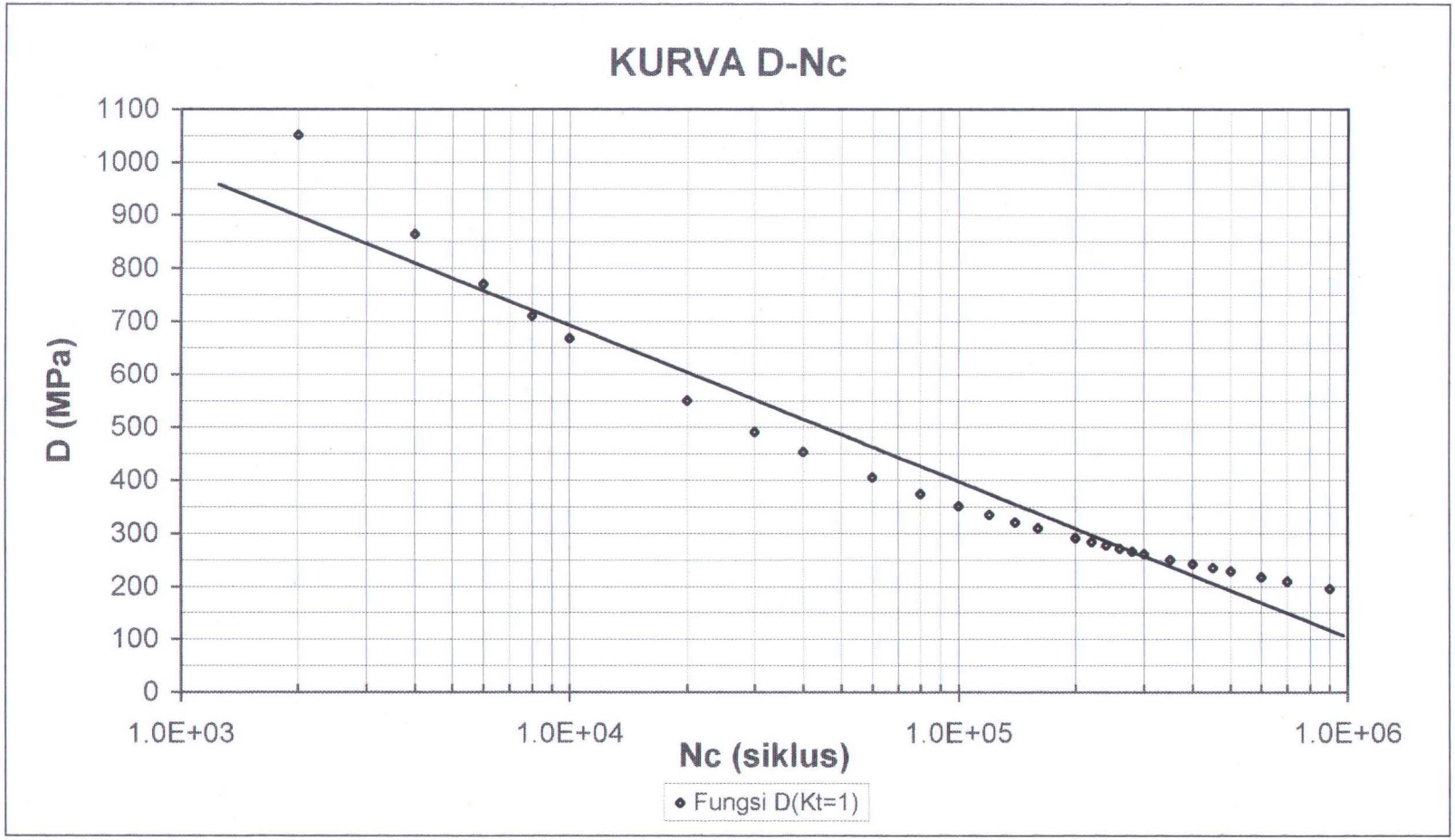
- b = eksponen regangan fatigue

  $b = -1/(1+5n)$; dengan n adalah cyclic strain hardening exponent, yang juga merupakan kemiringan kurva tegangan-regangan siklus. Untuk mendapatkan kurva ini dilakukan dengan pengujian pada round bar spesimen material induk, kemudian diukur kemiringan kurva yang terbentuk. Dari pengukuran yang telah dilakukan diperoleh harga n sebesar 0,17.

Perhitungan untuk mendapatkan persamaan fungsi tegangan regangan material induk dilakukan dengan memasukkan parameter-parameter prediksi tersebut. Secara ringkas perhitungan fungsi *D* dapat dilihat pada tabel (5.4) berikut :

Tabel 5.5 Perhitungan persamaan fungsi tegangan-regangan pada $K_{t=1}$

Nc	A	B	a	b	F	G	f	g	D
2000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	1051.38853
4000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	864.49540
6000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	771.14405
8000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	711.17737
10000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	667.94880
20000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	550.03138
30000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	491.19641
40000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	453.43432
60000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	405.27592
80000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	374.38582
100000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	352.14592
120000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	335.02050
140000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	321.23778
160000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	309.79257
200000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	291.64267
220000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	284.24741
240000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	277.67579
260000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	271.77833
280000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	266.44169
300000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	261.57799
350000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	251.06565
400000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	242.34270
450000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	234.93302
500000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	228.52354
600000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	217.90233
700000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	209.36869
900000	0.00180	0.26788	-0.02758	-0.54054	2315.727	20627793.617	-0.05516	-0.56812	196.28744



Gambar 5. kurva D-Nc baja ASTM 516 grade 70

Data hasil perhitungan fungsi tegangan-regangan *(D)* dan umur keretakan awal *(N_c)* tersebut dilinierisasi, sehingga diperoleh suatu kurva *D-N_c* yang linier, dengan persamaan garis :

$$y = -127{,}97 \ln (X) + 1871{,}4$$

Kurva ini merupakan master plot hubungan fungsi tegangan-regangan dan umur keretakan awal pada material induk yang rata *(smooth)* sehingga faktor konsentrasi tegangannya sama dengan satu *(K_t = 1)*. Karena $K_t = 1$ maka fungsi *D* mempunyai harga sama dengan rentang beban yang diterapkan *(Δs)*, sesuai dengan hubungan $D = K_t.\Delta s$.

Kurva D-Nc pada gambar (5.1) dapat digunakan untuk memperkirakan umur keretakan awal pada spesimen bertakik atau yang mengandung diskontinuitas *($K_t > 1$)*. Dengan berdasarkan pada persamaan (3.6) dapat digambarkan kurva tegangan-umur keretakan awal *(S-N_c)* untuk spesimen yang mempunyai faktor konsentrasi tegangan lebih dari satu.

$$K_t.\Delta s = (\Delta\sigma.\Delta\varepsilon.E)^{1/2} = D$$

$$\Delta s = \frac{(\Delta\sigma.\Delta\varepsilon.E)^{1/2}}{K_t} = \frac{D}{K_t} \qquad (5.1)$$

Dengan menggunakan persamaan (5.1) dapat dihitung dan digambarkan kurva S-N_c pada spesimen sambungan las, baik spesimen as welded maupun yang mendapatkan perlakuan TIG Dressing. Faktor konsentrasi tegangan yang digunakan dalam

pembuatan kurva ini adalah faktor konsentrasi tegangan paling tinggi untuk satu jenis perlakuan.

Pengujian fatigue yang dilakukan adalah untuk memperoleh data umur keretakan awal *(N_c)* dan data umur fatigue *(N_f)* spesimen yang diuji. Pada saat pengamatan terjadinya keretakan awal selain mencatat jumlah siklus pembebanan saat terjadinya keretakan awal, juga dicatat dan ditandai posisi dimana keretakan awal tersebut terjadi. Pengamatan posisi ini penting untuk menentukan faktor konsentrasi tegangan spesimen yang sesungguhnya, yaitu faktor konsentrasi tegangan pada posisi dimana keretakan awal tersebut terjadi.

Hasil pengamatan menunjukkan bahwa sebagian besar keretakan awal timbul pada posisi yang mempunyai faktor konsentrasi tegangan tertinggi. Posisi timbulnya keretakan awal dan faktor konsentrasi tegangan untuk masing-masing spesimen tampak pada tabel (5.5). Namun dari semua kejadian keretakan awal yang diamati tidak seluruhnya terjadi pada titik dengan faktor konsentrasi tegangan tertinggi, penyimpangan ini terjadi pada spesimen A1 dan S1. Keretakan awal pada kedua spesimen ini terjadi ditepi spesimen sedangkan faktor konsentrasi tegangan tertinggi terjadi pada posisi yang terletak agak ditengah spesimen.

Tabel 5.5 Posisi keretakan awal dan faktor konsentrasi tegangan spesimen

Spesimen	Posisi Keretakan	K_t pada posisi keretakan	K_t maksimal pada spesimen
A1	1	2.710911	2.744948
A2	9	2.748132	2.748132
A3	3	2.646467	2.646467
D1	3	1.893613	1.893613
D2	1	1.845060	1.845060
D3	9	1.823393	1.823393
S1	10	2.156458	2.500672
S2	7	2.440216	2.440216
S3	9	2.335650	2.335650

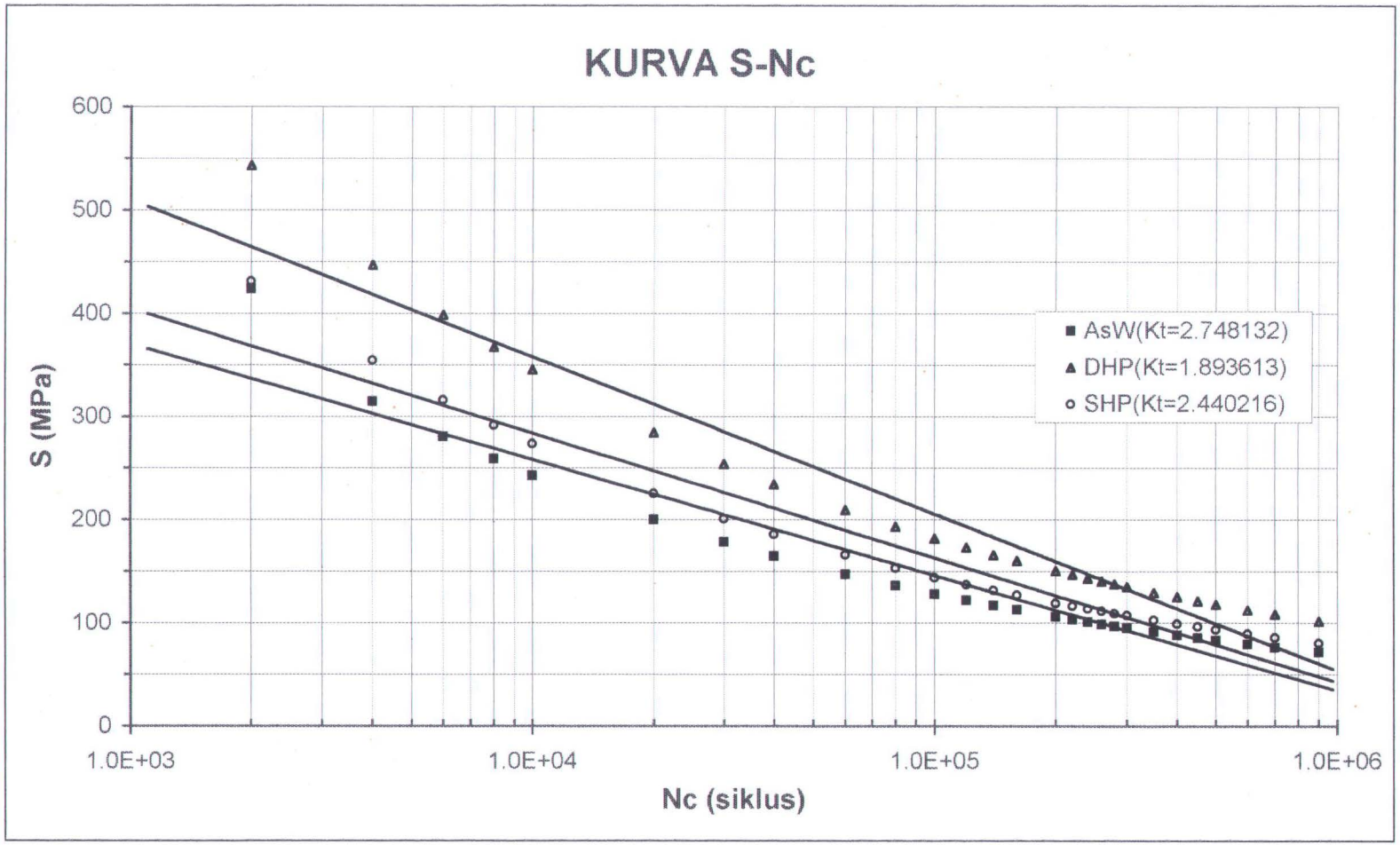
Faktor konsentrasi tegangan spesimen berbeda satu dengan yang lain, akan tetapi untuk satu kondisi perlakuan yang sama perbedaan tersebut tidak begitu besar. Oleh karena itu untuk kepentingan pembuatan kurva S-N_c, faktor konsentrasi tegangan pada satu kondisi perlakuan dipilih salah satu yaitu pada harga yang paling tinggi dari ketiga K_t yang ada. Untuk spesimen kondisi as welded diambil harga K_t sebesar 2.748132 (spesimen A2), untuk spesimen double heating pass diambil harga K_t sebesar 1.893613 (spesimen D1) dan untuk spesimen single heating pass diambil K_t sama dengan 2.440216 (spesimen S2).

Perhitungan kurva S-N_c pada spesimen as welded, double heating pass dan single heating pass dapat dilihat pada tabel (5.6) berikut :

Tabel 5. Perhitungan kurva S-Nc

Nc	D	S (Kt=2.748132)	S (Kt=1.893613)	S (Kt=2.440216)
2000	1051.38853	423.72146	543.03612	430.85880
4000	864.49540	314.57565	446.50690	354.27003
6000	771.14405	280.60663	398.29146	316.01467
8000	711.17737	258.78574	367.31902	291.44034
10000	667.94880	243.05557	344.99171	273.72528
20000	550.03138	200.14737	284.08804	225.40274
30000	491.19641	178.73829	253.70012	201.29218
40000	453.43432	164.99729	234.19622	185.81729
60000	405.27592	147.47324	209.32268	166.08199
80000	374.38582	136.23284	193.36812	153.42323
100000	352.14592	128.14011	181.88134	144.30933
120000	335.02050	121.90845	173.03616	137.29133
140000	321.23778	116.89314	165.91747	131.64317
160000	309.79257	112.72842	160.00608	126.95293
200000	291.64267	106.12397	150.63176	119.51511
220000	284.24741	103.43295	146.81215	116.48453
240000	277.67579	101.04165	143.41795	113.79148
260000	271.77833	98.89566	140.37195	111.37470
280000	266.44169	96.95375	137.61560	109.18775
300000	261.57799	95.18393	135.10353	107.19461
350000	251.06565	91.35866	129.67396	102.88665
400000	242.34270	88.18452	125.16861	99.31199
450000	234.93302	85.48826	121.34155	96.27550
500000	228.52354	83.15595	118.03109	93.64890
600000	217.90233	79.29107	112.54530	89.29633
700000	209.36869	76.18582	108.13773	85.79925
900000	196.28744	71.42577	101.38133	80.43855

Data hasil perhitungan ini kemudian diplot terhadap umur keretakan awal, sehingga menghasilkan suatu kurva yang disebut dengan kurva *S-N_c*. Kurva ini menggambarkan hubungan antara besar rentang tegangan yang diterapkan pada spesimen, dengan jumlah siklus pembebanan hingga mulai timbul keretakan awal. Dengan menggunakan kurva *S-N_c* ini dapat diperkirakan umur keretakan awal spesimen *(N_c)* bila diberikan beban siklus sebesar *S*.



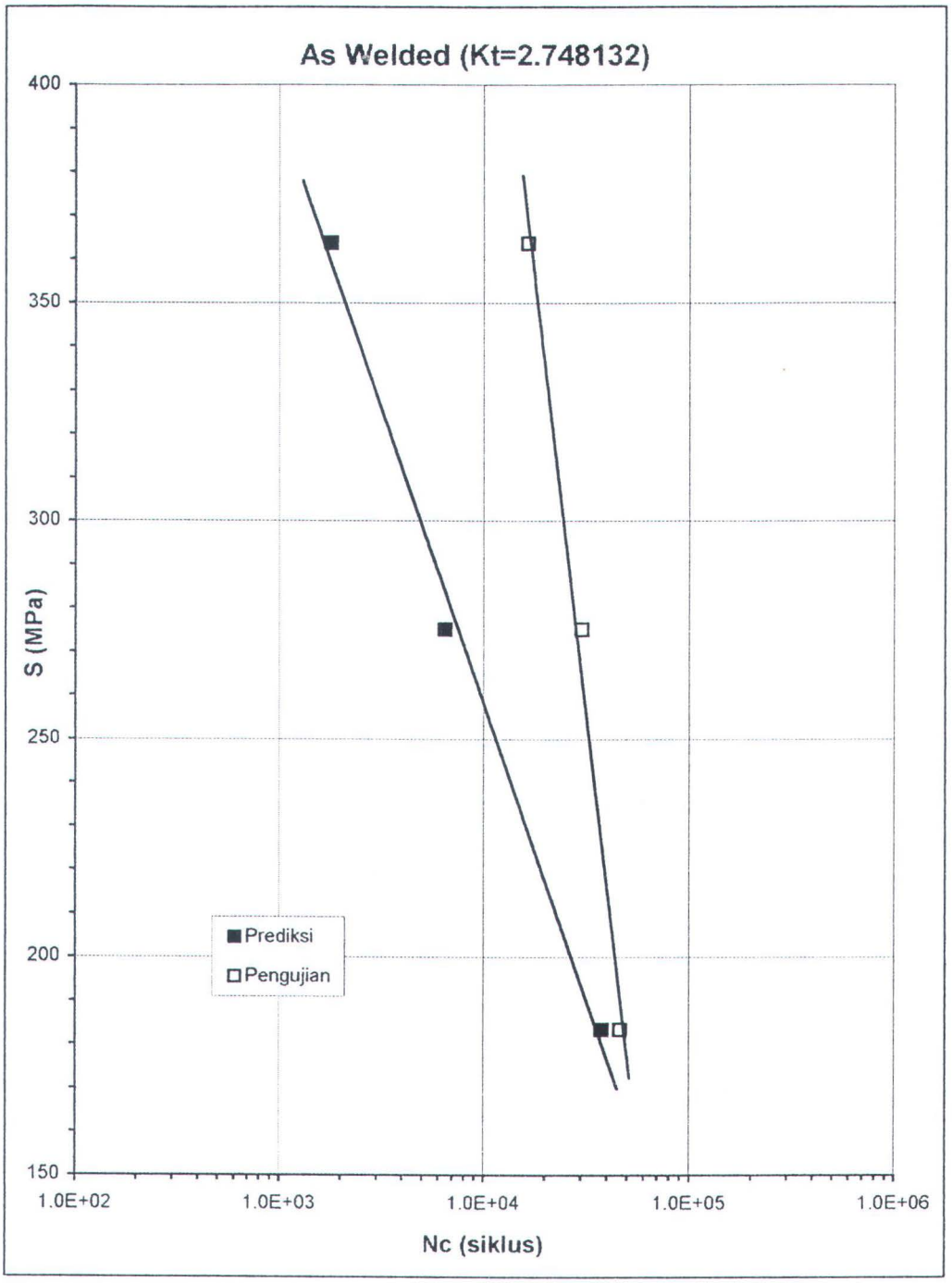
Gambar 5. Kurva D-Nc spesimen as welded, single heating pass dan double heating pass

As Welded (Prediksi)

Beban	Nc
363.698	1800
275.024	6500
183.349	38000

As Welded (Pengujian)

Beban	Nc
363.698	16552
275.024	30373
183.349	46493



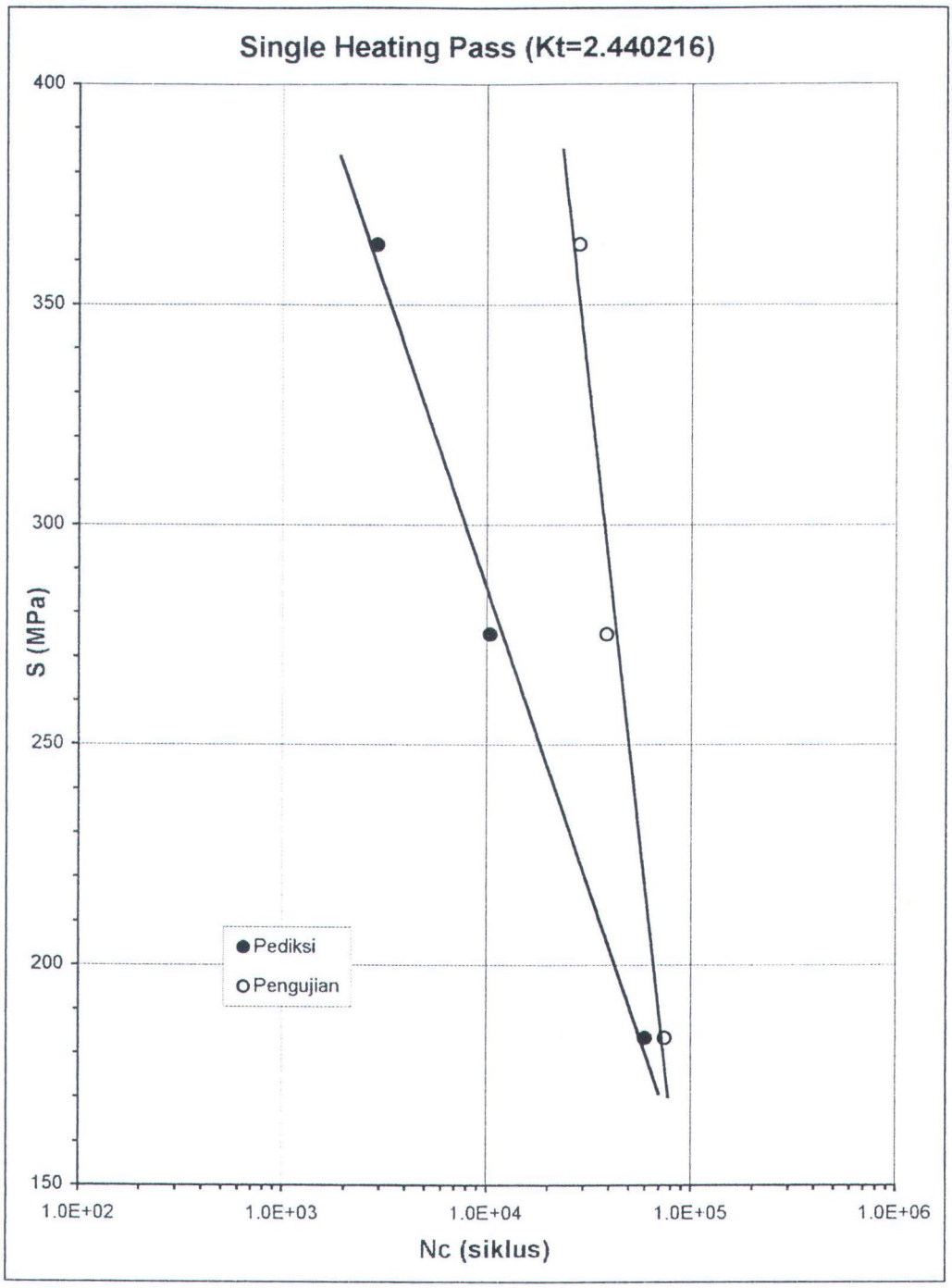
Gambar 5. Kurva S-Nc spesimen as welded

Single Heating Pass (Prediksi)

Beban	Nc
363.698	2900
275.024	10400
183.349	60000

Single Heating Pass (Pengujian)

Beban	Nc
363.698	28470
275.024	38878
183.349	74746



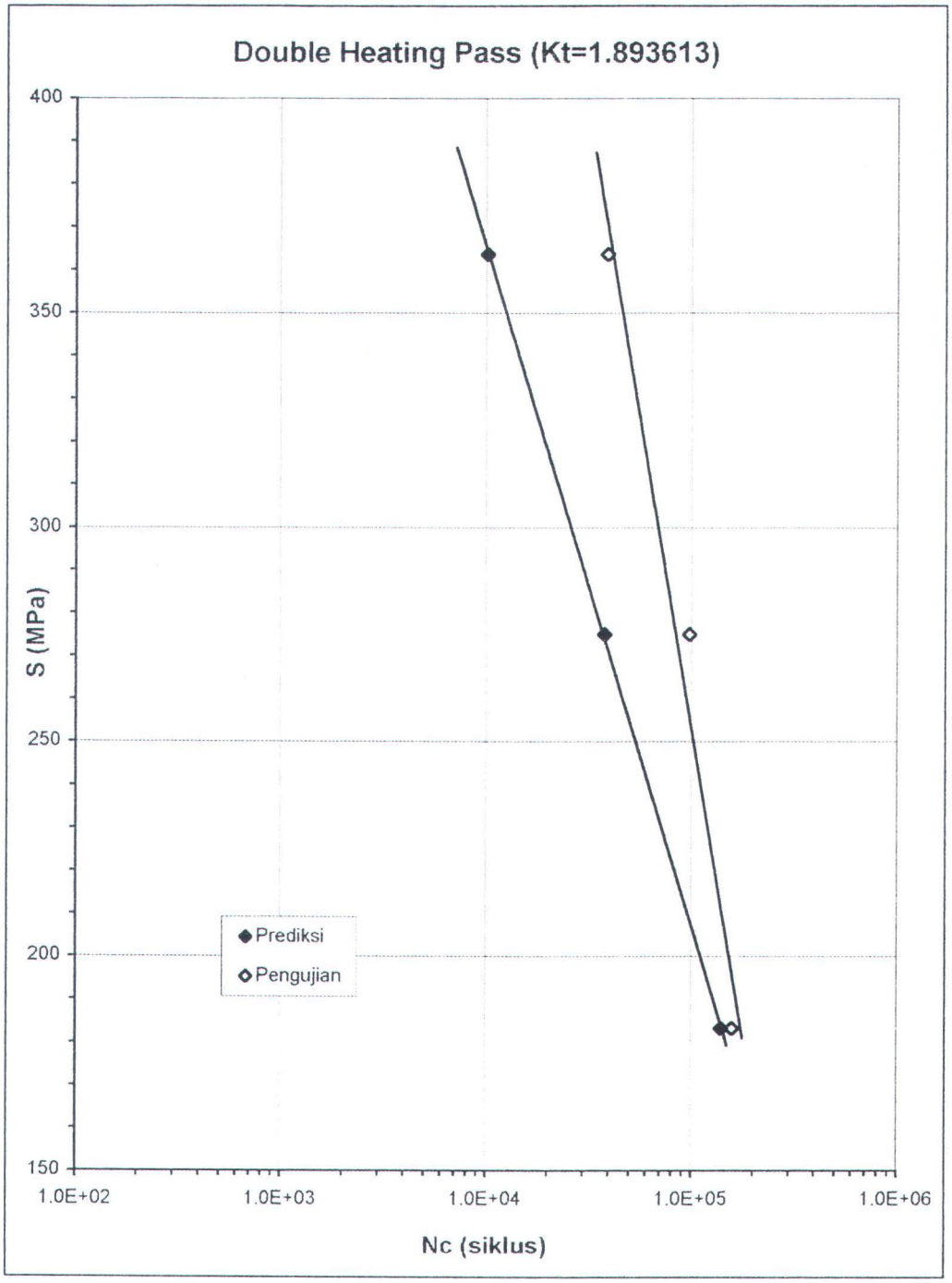
Gambar 5. Kurva S-Nc spesimen single heating pass

Double Heating Pass (Prediksi)

Beban	Nc
363.698	10150
275.024	38000
183.349	140000

Double Heating Pass (Pengujian)

Beban	Nc
363.698	39105
275.024	97650
183.349	158066



Gambar 5. kurva S-Nc spesimen double heating pass

5.3. KURVA S-N

Data utama yang ingin didapatkan dari pengujian fatigue adalah jumlah siklus pembebanan hingga terjadi kepatahan pada material, apabila diterapkan beban berulang sebesar S. Jumlah siklus pembebanan tersebut dinyatakan sebagai umur fatigue material *(N_f)*. Untuk tingkat pembebanan yang sama, material dengan umur fatigue yang lebih panjang dikatakan mempunyai kekuatan fatigue yang lebih tinggi. Kharakteristik kekuatan fatigue material biasanya digambarkan pada suatu kurva yang disebut dengan kurva stress-number (kurva *S-N*).

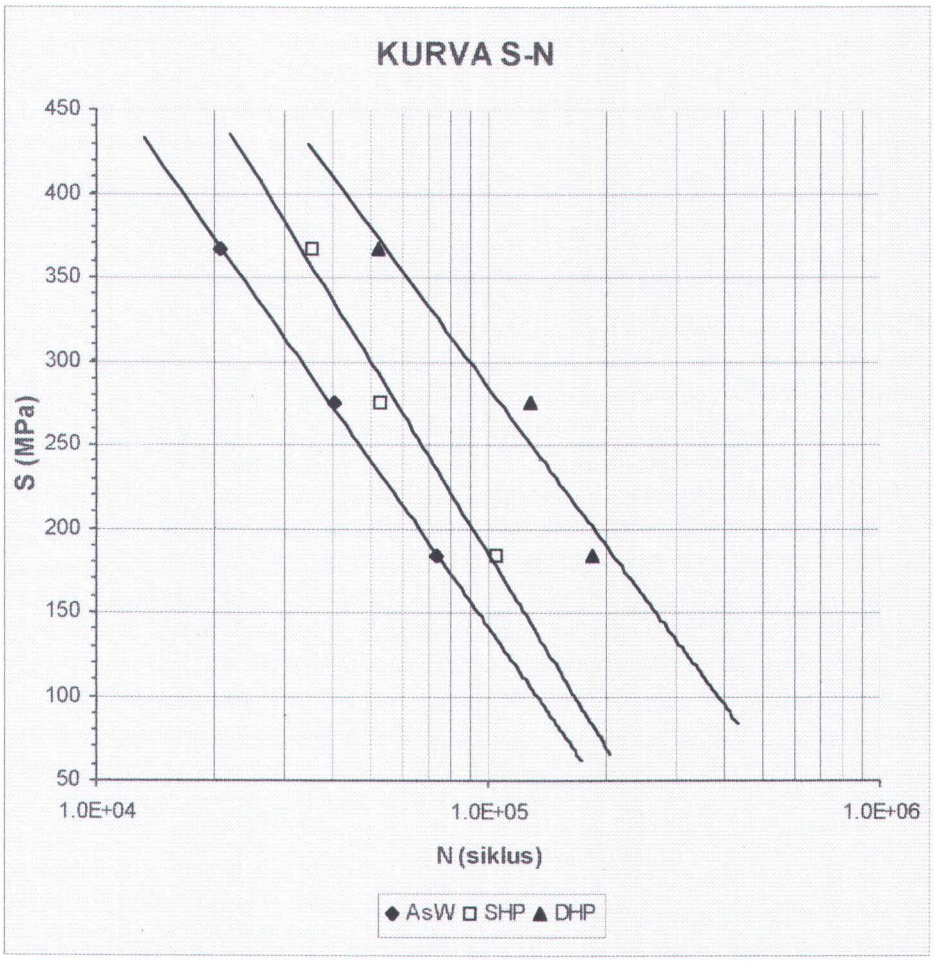
Dalam pengujian fatigue yang dilakukan pada spesimen sambungan las as welded, double heating pass dan single heating pass, diperoleh data jumlah siklus pembebanan hingga terjadi kepatahan (fracture) pada spesimen tersebut. Beban yang diterapkan pada satu jenis perlakuan dibedakan yaitu pada 0,5 σ_y (tegangan luluh material), 0,75 σ_y dan sebesar σ_y. Data umur fatigue masing-masing spesimen dan rentang beban yang diterapkan dapat dilihat pada tabel (5.7).

Umur fatigue dan rentang beban spesimen akan diplot pada suatu kurva stress-number, agar diperoleh kharakteristik kekuatan fatigue masing-masing spesimen yang mendapatkan perlakuan yang berbeda. Spesimen yang mendapatkan perlakuan TIG Dressing maupun spesimen as welded akan membentuk tren tersendiri,

sehingga dapat dibandingkan kharakteristik kekuatan fatigue masing-masing perlakuan.

Tabel 5.7 Umur fatigue spesimen

No.	BEBAN	AsW	SHP	DHP
1	366.698	20691	35588	52141
2	275.024	40498	52838	128608
3	183.349	73786	104655	183896



Gambar 5.6 Kurva S-N untuk spesimen as welded, spesimen dengan perlakuan single heating pass dan double heating pass

Dari kurva S-N pada gambar (5.6) didapatkan bahwa perlakuan TIG yang diterapkan pada spesimen dapat meningkatkan kekuatan fatigue-nya. Pada siklus pembebanan 100.000 kali,

kekuatan fatigue spesimen as welded sebesar 140 MPa, pada spesimen yang mendapat perlakuan single heating pass sebesar 186 MPa dan pada spesimen double heating pass kekuatan fatigue-nya sebesar 284 MPa.

5.4. PEMBAHASAN HASIL PENGUJIAN

Setelah melalui serangkaian proses pengujian pada spesimen sehingga diperoleh data yang kemudian diolah dan disajikan dalam bentuk tabel dan kurva, dapat dilakukan pembahasan berdasarkan pada fakta hasil pengujian yang diperoleh.

- Konsentrasi tegangan merupakan faktor penentu kekuatan fatigue sambungan.

  Konsentrasi tegangan pada material diekspresikan dengan faktor konsentrasi tegangan, semakin tinggi faktor konsentrasi tegangan semakin rendah kekuatan fatigue material tersebut. Hal ini dapat dibuktikan dengan mekanisme timbulnya keretakan awal pada pengujian fatigue spesimen yang mempunyai faktor konsentrasi tegangan yang berbeda.

  Dalam satu spesimen, faktor konsentrasi tegangannya berbeda dari satu titik ke titik yang lain. Dari sembilan spesimen yang diuji, didapatkan tujuh dari sembilan spesimen (77.8 %) keretakannya dimulai dari titik yang mempunyai faktor konsentrasi tegangan tertinggi, sedangkan dua dari sembilan

spesimen (22.2 %) keretakannya terjadi pada titik yang lain. Faktor yang kemungkinan besar menyebabkan terjadinya penyimpangan ini adalah pengaruh *edge effect* yaitu tingkat ketajaman sudut pada pinggir spesimen. Edge effect ini disebabkan karena kurang terbentuknya round pada pinggir spesimen pada saat pembentukan spesimen, sehingga memungkinkan timbulnya scratching. Hal ini terbukti dari kecenderungan penyimpangan daerah timbulnya retak selalu dimulai dari pinggir spesimen.

Berdasarkan pada kenyataan ini dapat disimpulkan bahwa faktor konsentrasi tegangan merupakan faktor dominan yang mengendalikan timbulnya keretakan pada spesimen, sehingga secara umum dapat dikatakan sebagai faktor penentu kekuatan fatigue spesimen.

- Umur keretakan awal yang didapatkan dari pengujian lebih lama dibandingkan dengan umur keretakan awal perkiraan.

Dari kurva prediksi keretakan awal spesimen, dapat diketahui bahwa ada kecenderungan pada pembebanan tinggi (mendekati σ_y) perbedaan umur keretakan awal *(N_c)* antara hasil prediksi dengan hasil pengujian cukup besar, sedangkan pada pembebanan rendah perbedaan yang terjadi tidak begitu besar. Pada tabel (5.8) ditunjukkan perbandingan antara N_c prediksi (P)

dan N_c pengujian/riil (R), bila P/R = 100 % berarti N_c prediksi sama dengan N_c riil (pengujian).

Tabel 5.8 Perbandingan umur keretakan awal hasil prediksi dan hasil pengujian

Jenis spesimen	Pembebanan	Prediksi (P) (siklus)	Pengujian (R) (siklus)	R / P	$\frac{P}{R}$ x100%
As Welded	0,5 σ_y	38.000	46.493	1,224 : 1	81,73
	0,75 σ_y	6.500	30.373	4,672 : 1	21,40
	σ_y	1.800	16.552	9,196 : 1	10,87
Double Heating Pass	0,5 σ_y	140.000	158.066	1,129 : 1	88,57
	0,75 σ_y	38.000	97.650	2,570 : 1	38,91
	σ_y	10.150	39.105	3,853 : 1	25,96
Single Heating Pass	0,5 σ_y	60.000	74.746	1,245 : 1	80,27
	0,75 σ_y	10.400	38.878	3,738 : 1	26,75
	σ_y	2.900	28.470	9,817 : 1	10,19

Faktor konsentrasi tegangan (K_t) dapat diterapkan apabila material dalam kondisi elastis, sehingga pembebanan harus dibawah tegangan yield (σ_y). Oleh karena itu pada tingkat pembebanan tinggi (mendekati σ_y), keretakan awal yang diperoleh dari prediksi fungsi tegangan-regangan hasilnya jauh berbeda dengan hasil pengujian. Sedangkan pada tingkat pembebanan dibawah σ_y, tren keretakan awal prediksi maupun pengujian menjadi lebih dekat.

Selain itu ada beberapa hal yang mungkin menyebabkan perbedaan umur keretakan awal ini : Pertama adalah keakuratan kontroler beban mesin uji dan kedua keterbatasan waktu pelaksanaan pengujian, pengujian tidak dapat dilakukan secara

bagi penelitian yang mungkin dilakukan pada masa yang akan datang adalah :

1. Dalam penelitian ini belum memasukkan pembahasan mengenai perubahan metalurgi material akibat perlakuan TIG Dressing. Proses TIG Dressing dengan masukan panas yang cukup tinggi, jelas merubah struktur logam baik pada logam induk maupun logam las. Perubahan ini sangat mungkin mempengaruhi kekuatan fatigue sambungan las, sehingga hal ini merupakan topik yang dapat dikembangkan dan diteliti.
2. Mesin uji fatigue yang digunakan dalam eksperimen kondisinya kurang memadai untuk mewakili kondisi pembebanan berulang yang dirancang. Ketidakstabilan kontroler beban dan juga keterbatasan waktu operasi merupakan faktor yang mempengaruhi keakuratan hasil pengujian. Diharapkan untuk penelitian lebih lanjut dapat menggunakan mesin uji fatigue yang lebih baik kondisinya dan bisa dioperasikan secara terus menerus sesuai dengan kondisi ideal pengujian fatigue.

DAFTAR PUSTAKA

Almar-Naess, A., [1985], *"Fatigue Handbook"*, Norges Tekniske Hogskole.

ASTM A 606, *"Standard Practice for Strain-Controlled Fatigue Testing"*.

Byars, E.F., & Snyder, R.D., [1986], *"Engineering Mechanics of Deformable Bodies"*, Departement of Theoritical and Applied Mechanics, West Virginia University.

Cary, H.B., [1994], *"Modern Welding Technology"*, Prentice Hall, New Jersey.

Davies, A.C., [1945], *"The Science and Practice of Welding"*, Cambridge University Press.

Itoh, Y., [1987], *"Fatigue Life Prediction for GMA Welded Butt Joint in a C-Mn-Si Steel"*, Welding Research Supplement.

Kariwibowo, P., [1996], *"Pengaruh Prosedur Pengelasan Terhadap Perambatan Retak Pada Butt Weld Joint Material SA 516 Carbon Steel"*, Fakultas Teknologi Kelautan-ITS, Surabaya.

Munse, W.H., & Grover, L., [1964], *"Fatigue of Welded Steel Structure"*, Welding Research Council.

Neuber, H., [1961], *"Theory of Stress Concentration for Shear Strained Prismatical Bodies with Arbitrary Non Linear Stress Strain Law"*, Journal of Applied Mechanics, pp 544-550.

Nishida, H., [1967], *"Stress Concentration"*, Mori Kita Publ. Co.

Osgood, C.C., [1982], *"Fatigue Design"*, Mechanical and Metallurgical Engineeer, Pergamon Press, New Jersey.

Soeweify, *"Studies on Bending Fatigue Strength of Cruciform Fillet Welded Joint"*, Departement of Ship Structure, The Faculty of Engineering, University of Hiroshima.

PERPUSTAKAAN
INSTITUT TEKNOLOGI

Vassalos, D., [1991], *"Fatigue Design of Offshore Structure"*, Marine Technology Centre, University of Strathclyde.

Wiryosumarto, H., & Okumura, T., [1994], *"Teknologi Pengelasan Logam"*, Pradnya Paramitha, Jakarta.

Zaczek, Z., [1984], *"Improvement in The Fatigue Strength of Butt Welded Joints by TIG Remelting of Weld Reinforcement"*, Metal Construction.

SPECIFICATION FOR PRESSURE VESSEL PLATES, CARBON STEEL, FOR MODERATE- AND LOWER-TEMPERATURE SERVICE

SA-516/SA-516M

(Identical with ASTM Specification A 516/A 516M-86)

1. Scope

1.1 This specification covers carbon steel plates intended primarily for service in welded pressure vessels where improved notch toughness is important.

1.2 Plates under this specification are available in four grades having different strength levels as follows:

Grade U.S. [SI]	Tensile Strength, ksi [MPa]
55 [380]	55–75 [380–515]
60 [415]	60–80 [415–550]
65 [450]	65–85 [450–585]
70 [485]	70–90 [485–620]

1.3 The maximum thickness of plates is limited only by the capacity of the composition to meet the specified mechanical property requirements; however, current practice normally limits the maximum thickness of plates furnished under this specification as follows:

Grade U.S. [SI]	Maximum Thickness, in. [mm]
55 [380]	12 [305]
60 [415]	8 [205]
65 [450]	8 [205]
70 [485]	8 [205]

1.4 The values stated in either inch-pound units or SI units are to be regarded separately as standard. Within the text, the SI units are shown in brackets. The values stated in each system are not exact equivalents; therefore, each system must be used independently of the other. Combining values from the two systems may result in nonconformance with the specification.

2. Referenced Documents

2.1 *ASTM Standards:*

A 20/A 20M Specification for General Requirements for Steel Plates for Pressure Vessels

A 435/A 435M Specification for Straight-Beam Ultrasonic Examination of Steel Plates

A 577/A 577M Specification for Ultrasonic Angle-Beam Examination of Steel Plates

A 578/A 578M Specification for Straight-Beam Ultrasonic Examination of Plain and Clad Steel Plates for Special Applications

3. General Requirements and Ordering Information

3.1 Material supplied to this material specification shall conform to Specification A 20/A 20M. These requirements outline the testing and retesting methods and procedures, permissible variations in dimensions, and mass, quality and repair of defects, marking, loading, etc.

3.2 Specification A 20/A 20M also establishes the rules for the ordering information that should be complied with when purchasing material to this specification.

3.3 In addition to the basic requirements of this specification, certain supplementary requirements are available when additional control, testing, or examination is required to meet end use requirements. These include:

3.3.1 Vacuum treatment,

3.3.2 Additional or special tension testing,

3.3.3 Impact testing, and

3.3.4 Nondestructive examination.

3.4 The purchaser is referred to the listed supplementary requirements in this specification and to the detailed requirements in Specification A 20/A 20M.

3.5 If the requirements of this specification are in conflict with the requirements of Specification A 20/A 20M, the requirements of this specification shall prevail.

4. Manufacture

4.1 *Steelmaking Practice*—The steel shall be killed and shall conform to the fine austenitic grain size requirement of Specification A 20/A 20M.

5. Heat Treatment

5.1 Plates 1.50 in. [40 mm] and under in thickness are normally supplied in the as-rolled condition. The plates may be ordered normalized or stress relieved, or both.

5.2 Plates over 1.50 in. [40 mm] in thickness shall be normalized.

5.3 When notch-toughness tests are required on plates 1½ in. [40 mm] and under in thickness, the plates shall be normalized unless otherwise specified by the purchaser.

5.4 If approved by the purchaser, cooling rates faster than those obtained by cooling in air are permissible for improvement of the toughness, provided the plates are subsequently tempered in the temperature range 1100 to 1300°F [595 to 705°C].

6. Chemical Requirements

6.1 The steel shall conform to the chemical requirements shown in Table 1 unless otherwise modified in accordance with Supplementary Requirement S17, Vacuum Carbon-Deoxidized Steel, in Specification A 20/A 20M.

7. Mechanical Requirements

7.1 *Tension Test Requirements*—The material as represented by the tension-test specimens shall conform to the requirements shown in Table 2.

TABLE 1 Chemical Requirements

Elements	Composition, %			
	Grade 55 [Grade 380]	Grade 60 [Grade 415]	Grade 65 [Grade 450]	Grade 70 [Grade 485]
Carbon, max[A]:				
½ in. [12.5 mm] and under	0.18	0.21	0.24	0.27
Over ½ in. to 2 in. [12.5 to 50 mm], incl	0.20	0.23	0.26	0.28
Over 2 in. to 4 in. [50 to 100 mm], incl	0.22	0.25	0.28	0.30
Over 4 to 8 in. [100 to 200 mm], incl	0.24	0.27	0.29	0.31
Over 8 in. [200 mm]	0.26	0.27	0.29	0.31
Manganese:				
½ in. [12.5] and under:				
Heat analysis[B]	0.60–0.90	0.60–0.90	0.85–1.20	0.85–1.20
Product analysis[B]	0.55–0.98	0.55–0.98	0.79–1.30	0.79–1.30
Over ½ in [12.5]:				
Heat analysis	0.60–1.20	0.85–1.20	0.85–1.20	0.85–1.20
Product analysis	0.55–1.30	0.79–1.30	0.79–1.30	0.79–1.30
Phosphorus, max[A]	0.035	0.035	0.035	0.035
Sulfur, max[A]	0.04	0.04	0.04	0.04
Silicon:				
Heat analysis	0.15–0.40	0.15–0.40	0.15–0.40	0.15–0.40
Product analysis	0.13–0.45	0.13–0.45	0.13–0.45	0.13–0.45

[A] Applies to both heat and product analyses.

[B] Grade 60 plates ½ in. [12.5 mm] and under in thickness may have 0.85–1.20 % manganese on heat analysis, and 0.79–1.30 % manganese on product analysis.

TABLE 2 Tensile Requirements

	Grade			
	55 [380]	60 [415]	65 [450]	70 [485]
Tensile strength, ksi [MPa]	55–75 [380–515]	60–80 [415–550]	65–85 [450–585]	70–90 [485–620]
Yield strength, min,[A] ksi [MPa]	30 [205]	32 [220]	35 [240]	38 [260]
Elongation in 8 in. [200 mm], min, %	23[B]	21[B]	19[B]	17[B]
Elongation in 2 in. [50 mm], min, %	27[B]	25[B]	23[B]	21[B]

[A] See Specification A 20/A 20M.

[A] Determined by either the 0.2 % offset method or the 0.5 % extension-under-load method.

INSPECTION CERTIFICATE

KOBE MIG WIRE (THAILAND) CO., LTD.
491, SOI 1, BANGPOO INDUSTRIAL ESTATE, SUKHUMVIT ROAD, PRAEKSA
SAMUTPRAKARN 10280 THAILAND. TEL: 324-0588-91 FAX: 324-0797

LAMPIRAN A

Purchaser: ---

Certificate No: 94 - 044

Date of Issue : 26 NOVEMBER.1994

Brand Name	Size	Mfg. No.	Applicable Specification and Classification
MG - 51T	1.2	FP - 3598	AWS A5.18 ER70S - 6

Chemical Composition of Wire (%)

C	Si	Mn	P	S	Cu			
0.11	0.84	1.56	0.011	0.016	0.12			

Tensile Test of Deposited Metal				Impact Test of Deposited Metal			Radiographic Examination
Yield Point (N/mm^2)	Yield Strength at 0.2% Offset (N/mm^2)	Tensile Strength (N/mm^2)	Elongation (%)	Test Temp. (°C)	Absorbed Energy (J) Avg.		
------	-------	------	--------	--------	-----		

Welding Condition			
Type of Current	----------	Arc Voltage (V)	----------
Amperage (A)	------- -----	Shielding Gas	---

Remarks

We hereby certify that the test results of the above welding materials are as described herein and satisfy

KOBE MIG WIRE (THAILAND) CO., LTD.

Approved by T. Nishimura

WELDING PROCEDURE SPECIFICATIONS (WPS)
(See QW-200.1, Section IX, ASME Boiler and Pressure Vessel Code)

Company Name PTT . PAL INDONESIA(PERSERO) By: Ir. IBP. JELANTIK
Welding Procedure Specification No. SL 34 - 08 A Date JULY, 14, 1986 Supporting PQR No.(s) SL 34 - 08
Revision No. R - 3 Date 28-2-1994
Welding Process(es) GMAW Type(s) SEMI AUT
(Automatic, Manual, Machine, or Semi-Auto.)

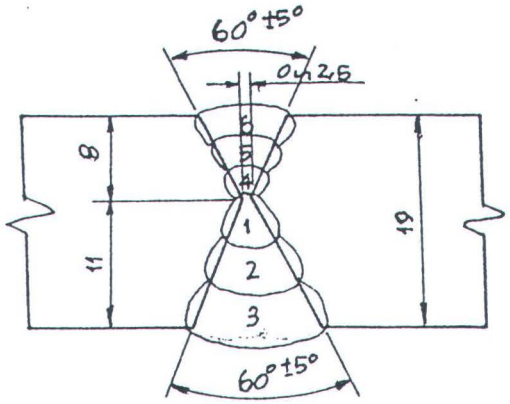
JOINTS (QW-402)
Joint Design X , V , U GROOVE
Backing (Yes) _____ (No) NO
Backing Material (Type) _____
(Refer to both backing and retainers.)

☐ Metal ☐ Nonfusing Metal
☐ Nonmetallic ☐ Other

Sketches, Production Drawings, Weld Symbols or Written Description should show the general arrangement of the parts to be welded. Where applicable, the root spacing and the details of weld groove may be specified.

(At the option of the Mfgr., sketches may be attached to illustrate joint design, weld layers and bead sequence, e.g. for notch toughness procedures, for multiple process procedures, etc.)

Details



*BASE METALS (QW-403)
P-No. 1 Group No. 1+2 to P-No. 1 Group No. 1+2
OR
Specification type and grade ALL P 1 GROUPS 1 & 2
to Specification type and grade ALL P 1 GROUPS 1 & 2
OR
Chem. Analysis and Mech. Prop. _____
to Chem. Analysis and Mech. Prop. _____
Thickness Range:
Base Metal: Groove 4,7 mm - 38 mm Fillet ALLL
Pipe Dia. Range: Groove ALL Fillet ALL
Other _____

NO SINGLE WELD SHALL EXCEED 1/2" IN THICKNESS

*FILLER METALS (QW-404)
Spec. No. (SFA) 5.18
AWS No. (Class) ER 70 S - 4
F-No. 6
A-No. 1 (Mn=1,60)
Size of Filler Metals Ø 1,2 mm
Weld Metal
Thickness Range:
Groove MAX = 38 mm
Fillet _____
Electrode-Flux (Class) _____
Flux Trade Name _____
Consumable Insert _____
Other TRADE NAME MG 50 T (KOBE STEEL)

*Each base metal-filler metal combination should be recorded individually.

This form (E00006) may be obtained from the Order Dept., ASME, 22 Law Drive, Box 2300, Fairfield, NJ 07007-230

QW-482 (Back)

WPS No SL 34-08A Rev. R - 3

POSITIONS (QW-405)
Position(s) of Groove: FLAT OR HARIZONTAL
Welding Progression: Up — Down —
Position(s) of Fillet: —

POSTWELD HEAT TREATMENT (QW-407)
Temperature Range: NONE
Time Range: —

PREHEAT (QW-406)
Preheat Temp. Min.: 10°C t > 1 1/4 = 200°F
Interpass Temp. Max.: 600°F
Preheat Maintenance: —
(Continuous or special heating where applicable should be recorded)

GAS (QW-408)

	Gas(es)	Percent Composition (Mixture)	Flow Rate
Shielding	Ar + CO_2	80%+20%	15-25 l/m
Trailing	—	—	—
Backing	—	—	—

ELECTRICAL CHARACTERISTICS (QW-409)
Current AC or DC: DC Polarity: RP
Amps (Range): 180 - 350 Volts (Range): 20-35
(Amps and volts range should be recorded for each electrode size, position, and thickness, etc. This information may be listed in a tabular form similar to that shown below.)

Tungsten Electrode Size and Type: — (Pure Tungsten, 2% Thoriated, etc.)
Mode of Metal Transfer for GMAW: SPRAY ARC (Spray arc, short circuiting arc, etc.)
Electrode Wire feed speed range: —

TECHNIQUE (QW-410)
String or Weave Bead: STRING & WEAVE
Orifice or Gas Cup Size: 15 - 20 mm
Initial and Interpass Cleaning (Brushing, Grinding, etc.): BRUSHING, CHIPPING & GRINDING
Method of Back Gouging: ARC AIR GOUGING, GRINDING, PENETRATION TEST
Oscillation: —
Contact Tube to Work Distance: 1- - 20 mm
Multiple or Single Pass (per side): MULTIPLE
Multiple or Single Electrodes: SINGLE
Travel Speed (Range): 32 - 36 mm/min
Peening: PROHIBITED
Other: NO SINGLE PASS > 1 M/2

Weld Layer(s)	Process	Filler Metal Class	Filler Metal Dia.	Current Type Polar.	Current Amp. Range	Volt Range	Travel Speed Range	Other (e.g., Remarks, Comments, Hot Wire Addition, Technique, Torch Angle, Etc.)
1	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	34-36Cm/min	
2	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	34-36Cm/min	
3	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	32-35Cm/min	
4	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	32-35Cm/min	
5	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	32-35Cm/min	
6	GMAW	E 70 S-4	1,2	DCRP	180-350	20-35V	32-35Cm/min	

EXPORT

Huta "CZĘSTOCHOWA"

ATEST HUTNICZY Nº 819/12/95

CERTIFICAT DE L'USINE — WORKS-CERTIFICATE — WERKS-ATTEST
СЕРТИФИКАТ

Zamawiający
Le client — Ordered by — Besteller — Заказчик **HZ Huta "CZĘSTOCHOWA"**

Adres wysyłkowy
Adresse — Address — Versandadresse — Адрес получателя **STENCOR UK LIMITED -SINGAPORE**

Nr i data zamówienia klienta No et date de la commande Order No and date Nr und Datum der Bestellung №№ и дата заказа	Nr zlecenia Order No Manuf. Order No Auftrag Nr № заказа	Nr awizu Avis No Advice No Versandanzeige No № извещения	Nr wagonu Wagon No Car No Wagen Nr № вагона
PL/000029463/95-5054	551-4573	817,829/VII	

Wyszczególnienie zamówienia:
Spécification de la commande — Order Specification — Spezifikation der Bestellung — Спецификация заказа

Poz. Pos. Pos. №№	Przedmiot i wykonanie (stan obr. termicz., mech. itp.) L'objet et l'exécution (traitement thermique et l'usinage) Item and specification (final and mechanical treatement etc.) Gegenstand und Ausführung (therm. und mechan. Bearbeitung usw.) Предмет и исполнение (состояние терм. и механобраб. и пр.)	Wymiar lub rysunek Dimension ou dessin Dimensions or drawing Abmessung oder Zeichnung Размер или чертеж	Marka Marque Steel type Marke Марка	Wytop Coulée Heat Abstich Плавка	Sztuk Pièces Pieces Stück Штук	mb c. prz. c. intr. l. M. пог. м	
	Boiler Quality carbon steel plate to ASTM A 516 ASTM A 20 DIN 17049/3.16 tested by LRS	15,9x1829x6096	"70"	190841 474442	4 14 -- 18		25

REVIEW BY QA
DATE: 6/12/96
ACCEPTED, FOR DEMO VESSEL ONLY.

REVIEW BY QA
DATE: 06/01/96
NOT ACCEPTED
SEE NCR No. DU-04/96
CERTIFIED COPY OF ORIGINAL
ACCEPTED
CONFORMS TO ASME SECT II A - 1995 & ADD. 1995
DEMO VESSELS PROJECT U2 STAMP (NO. TSO4NOI)

PT. HEAT EXCHANGERS INDONESIA QC ACCEPTED

Kontrole techniczne powyższego zamówienia przeprowadził Oddział Technicznej Kontroli. Wyniki badań podano niżej

Le contrôle technique de la commande a été exécuté par le Service de Contrôle. Les résultats des essais sont indiqués ci-après. The technical investigation of this order has been executed by the Works Control. Results of tests are as follows. Die technische Prüfung dieser Bestellung wurde von der Fabrikkontrollstelle durchgeführt. Die Ergebnisse der Prüfungen sind untenstehend angeführt.
Техническая проверка данного заказа проведена отделом Технического Контроля. Результат испытаний приведен ниже

1. SKŁAD CHEMICZNY — ANALYSE CHIMIQUE — CHEMICAL COMPOSITION — CHEMISCHE ZUSAMMENSETZUNG — ХИМИЧЕСКИЙ СОСТАВ

Marka Marque Steel type Marke Марка	Wytop Coulée Heat Abstich Плавка	C	Mn	Si	P	S	Cr	Ni	Cu	Mo
"70"	190841	0.19	1.10	0.30	0.019	0.033	0.08	0.05	0.12	0.01
	474442	0.23	1.10	0.28	0.015	0.028	0.07	0.06	0.12	0.01

6/12/96
G. BANATE

CERTIFIED TRUE COPY

2. BADANIA MECHANICZNE — ESSAIS MECANIQUES — MECHANICAL TESTS — MECHANISCHE UNTERSUCHUNGEN — МЕХАНИЧЕСКИЕ ИСПЫТАНИЯ

SHT-2-OF-2

Nr wytopu lub próby / No de la coulée ou de l'eprouvette / Heat No or. tests No / Abstich oder Probe No / № плавки или пробы	Stan obróbki termicznej / Traitement thermique / Heat treatement Therm. Bearbeitung / Термич. обработка	Re MPa / Бс МПа	Rm MPa / Бв МПа	A % / δ %	Z % / ψ %	U J/cm² / A Джж/см²	Twardość / Dureté / Hardness / Harte / Твердость		Złom / Riblon / Fracture / Bruch / Излом
							HB		Test No/P
136841		370	531	23.0	29.0	163,1	163/163	68341	
474442		382	539	21.5	30.0	156,1	156/156	68342	
		365	534	23.5	34.0	159,1	159/159	68343	
		386	548	22.5	34.0	158,1	158/158	68344	
		387	541	24.0	36.0	158,1	158/158	68345	

3. BADANIA TECHNOLOGICZNE

Controlled rolled

4. BADANIA METALOGRAFICZNE — ESSAIS METALLOGRAPHIQUES — METALLOGRAPHIC TESTS — METALLOGRAPHISCHE UNTERSUCHUNGEN — МЕТАЛЛОГРАФИЧЕСКИЕ ИСПЫТАНИЯ

Nr wytopu lub próby / No de la coulée ou d'essais / Heat No or. tests No / Abstich oder Probe No / № плавки или пробы	Zanieczyszczenie niemetaliczne / Impureté nonmetalliques / Nonmetallic impurities / Nichtmetallische Verunreinigungen / Неметаллические включения	Struktura / Structure / Structure / Struktur / Структура
	REVIEW BY QA DATE: 6/12/96 ACCEPTED. CONFORMS TO ~~ASME SECT IIA 1995~~ & ADD. 1995 ~~DEMO VESSELS~~ PROJECT U2 STAMP (FS0470H1). CERTIFIED COPY OF ORIGINAL. NOT ACCEPTED SEE MCR NO DV-04/96	CERTIFIED TRUE COPY

5. INNE BADANIA — AUTRES ESSAIS — OTHER TESTS — ANDERE UNTERSUCHUNGEN — ДРУГИЕ ИСПЫТАНИЯ

Powierzchnię i wymiary zbadano / Surface et dimensions ont été controlés / Surface and dimensions tested / Oberfläche und Abmessungen geprüft / Наружный осмотр и проверка измерений произведены

przez dz. prod. w / par Dept. et / by Dept. at 100% / von Prod.-Abt. zu / отд. произв. в

przez T. K. / par Sc. / by W. K. % / von F. K. / т. П. К.

REVIEW BY QA DATE: 6/12/96 ACCEPTED FOR DEMO VESSEL ONLY

Materiał oznaczono / Le material est marqué / Material marked / Das Material wurde bezeichnet

HEAT: C&M/HOSOWA & M... CUAN LEONG METALINDO ... dimension/BOILER QUA... SINGAPORE/A.936/1...

Na podstawie wyżej przeprowadzonych prób materiał zwolniono / Sur base des essais ci-dessus le material est délivré. — According to the carried ont tests the material relased — Auf Grund der oben durchgeführten Untersuchungen wurde das Material freigegeben — На основании вышеперечисленных испытаний, признан годным.

Kontrola Techniczna / Contrôle de Fabrikation / Control of Manufacture / Fabrikationskontrolle / Технический контроль	Dyrekcja Huty / Direction de l'Usine / Works Management / Hütten-Direktion / Дирекция Завода	
Huta „Częstochowa" Wydziałowy ... Jakości Walcownia ... Grubych	A. I. 6/14/93	LLOYD'S REGISTER SERVICE OF SHIPPING Wydano ... dnia ... 19... r.

terus menerus sehingga timbul kemungkinan terjadi proses strain hardening pada saat pengujian dihentikan.

- Perlakuan TIG Dressing terbukti dapat meningkatkan kekuatan fatigue sambungan las tumpul.

Pengujian yang dilakukan pada spesimen dengan kondisi sambungan yang berbeda menunjukkan bahwa pada spesimen yang mendapatkan perlakuan pada manik lasnya mempunyai kekuatan fatigue yang lebih tinggi.

Tabel 5.9 Perbandingan kekuatan fatigue spesimen

Kondisi sambungan	Kekuatan fatigue pada 10^5 siklus (MPa)	Peningkatan kekuatan (%)
As welded	140	-
Single heating passs	186	32,85
Double heating pass	284	102,86

Perlakuan TIG Dressing dengan single heating pass dapat meningkatkan kekuatan fatigue 32,85 % dari kondisi as welded, sedangkan teknik perlakuan teknik double heating pass meningkatkan kekuatan fatigue hingga 102,86 %.

Teknik double heating pass membutuhkan waktu pelaksanaan yang lebih lama karena perlu dua kali proses untuk satu sambungan sedangkan teknik single heating pass cukup satu kali proses. Perbandingan antara teknik perlakuan double heating pass dan single heating pass dapat dilihat pada tabel

(5.10). Pemilihan teknik perlakuan yang akan digunakan harus mempertimbangkan beberapa hal seperti efisiensi waktu dan proses pelaksanaan, kemampuan mesin las, kekuatan yang dihasilkan dan sebagainya. Secara umum kedua teknik tersebut produktivitasnya sama, sehingga pemilihan harus didasarkan pada situasi dan kondisi dimana teknik perlakuan TIG dilaksanakan.

Tabel 5.10 Perbandingan teknik double heating pass dan single heating pass

Jenis	DHP	SHP
Peningkatan kekuatan	102,86 %	32,85 %
Waktu proses pencairan (per 100 mm)	6,94 menit	4,72 menit
Rata-rata peningkatan jari-jari toe	11,39 %	8,59 %
Rata-rata peningkatan flank angle	201 %	60,6 %
Panas yang dibutuhkan	sedang	tinggi

BAB VI
KESIMPULAN DAN SARAN

BAB VI
KESIMPULAN DAN SARAN

6.1. KESIMPULAN

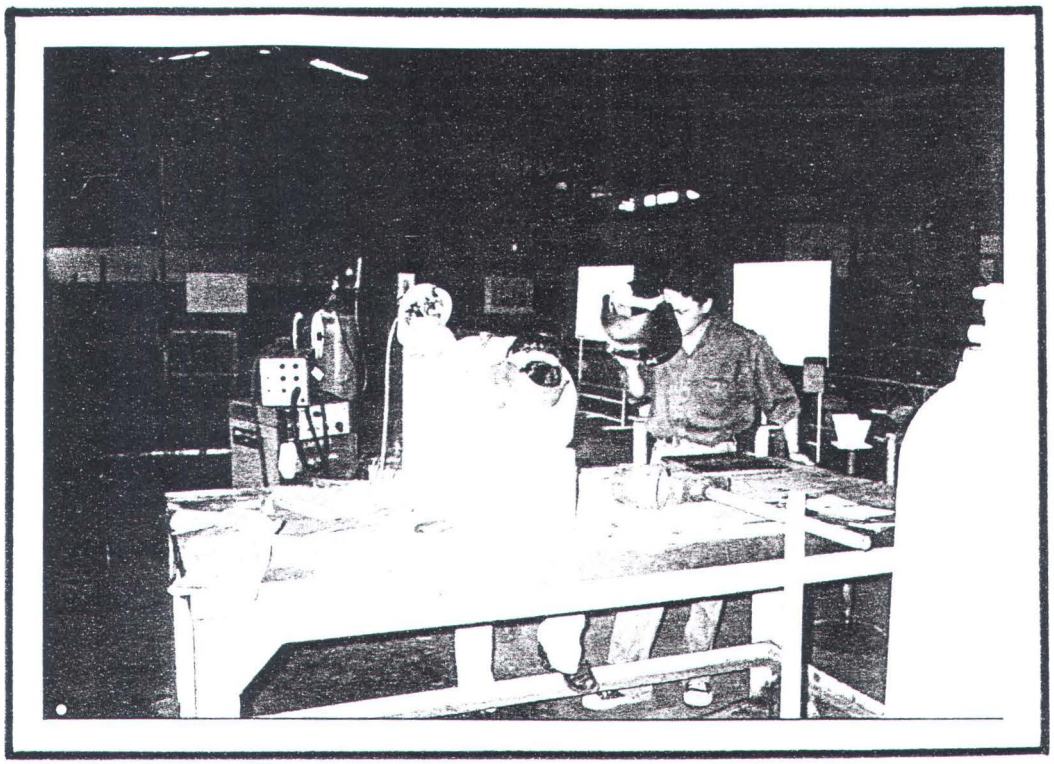
Pembahasan masalah yang dilakukan melalui serangkaian studi literatur dan ekperimen, telah dituangkan dalam bab-bab sebelumnya. Dari hasil pengujian yang telah dicapai, terdapat beberapa hal utama yang dapat diringkas dan disimpulkan sebagai berikut :

1. Faktor konsentrasi tegangan merupakan penentu kekuatan fatigue pada sambungan las tumpul yang tidak mengandung cacat pengelasan. Hal ini terbukti dari posisi timbulnya keretakan awal selalu dimulai pada titik yang mempunyai faktor konsentrasi tegangan tertinggi.
2. Perlakuan TIG Dressing pada permukaan manik las dapat menurunkan konsentrasi tegangan pada sambungan las. Berdasarkan perhitungan faktor konsentrasi tegangan, terbukti bahwa pada spesimen yang mendapatkan perlakuan TIG mempunyai faktor konsentrasi tegangan yang lebih rendah, sehingga konsentrasi tegangannya juga lebih rendah dibandingkan pada kondisi as welded.

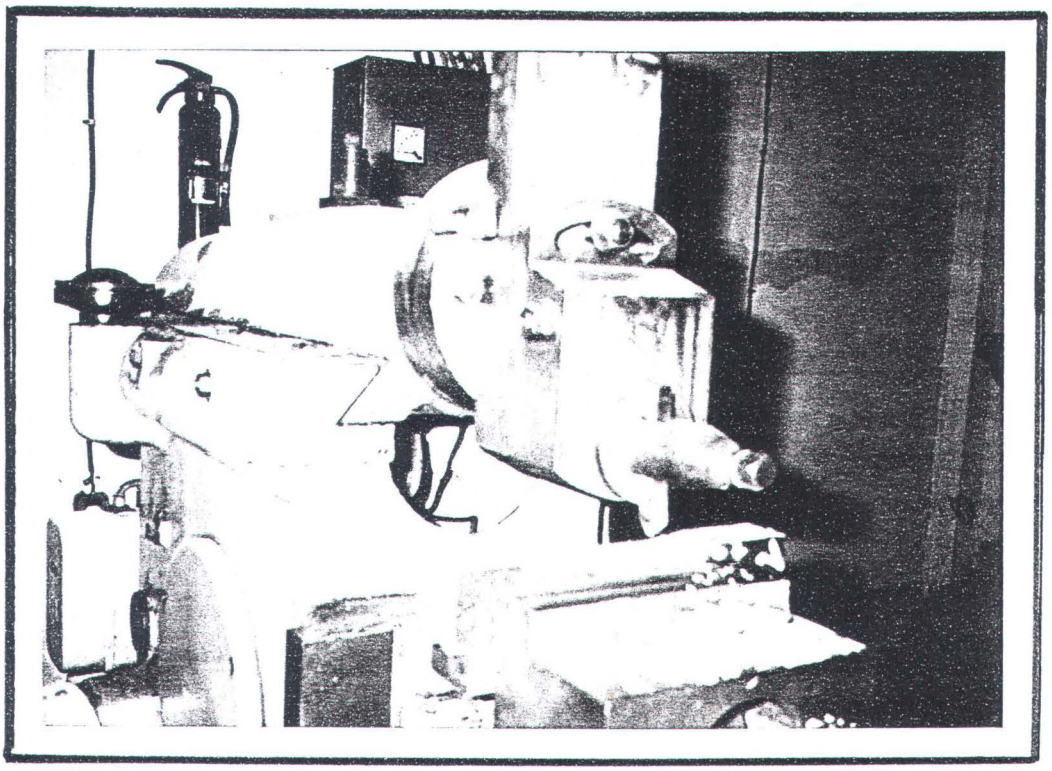
3. Fungsi tegangan-regangan *(D)* yang digunakan untuk memperkirakan umur keretakan awal hanya berlaku pada zona elastis material sehingga beban yang diterapkan harus dibawah tegangan yield (σ_y) material.
4. Teknik perlakuan TIG Dressing single heating pass dan double heating pas terbukti dapat meningkatkan kekuatan fatigue sambungan las tumpul. Pada 100.000 siklus pembebanan, teknik single heating pass dapat memperbesar kekuatan fatigue 32.85 %, sedangkan dengan teknik double heating pass hingga 102,86 % dari kekuatan fatigue pada kondisi as welded.
5. Material baja ASTM A 516 grade 70 mempunyai kharakteristik penjalaran retak yang cepat. Sesuai dengan sifat kegetasan pada material penjalaran retak berlangsung dengan kecepatan yang tinggi, terlebih lagi pada beban yang besar. Dari kondisi retak awal, tidak membutuhkan waktu yang lama hingga terjadi kepatahan menyeluruh pada spesimen.

6.2. SARAN

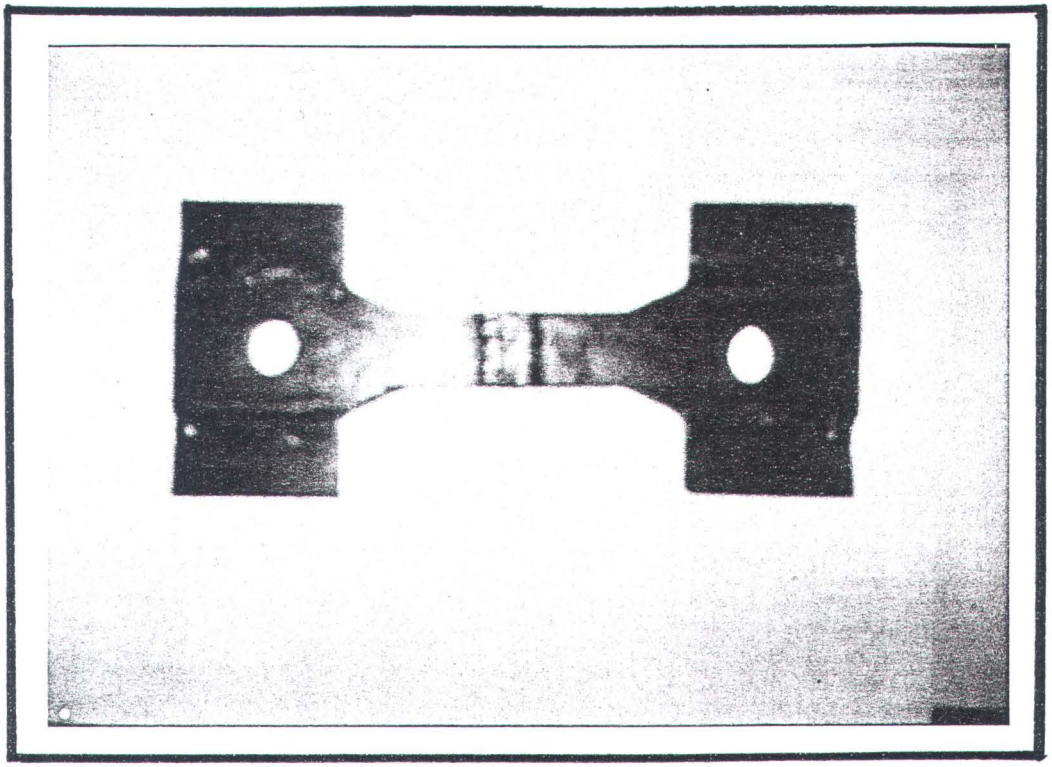
Proses pengujian yang dilakukan dalam studi ini dilakukan dalam beberapa keterbatasan, baik kondisi maupun sarana yang ada, sehingga hasil yang diperoleh belum bisa dikatakan mewakili kondisi yang sesungguhnya. Saran yang dapat penulis sampaikan



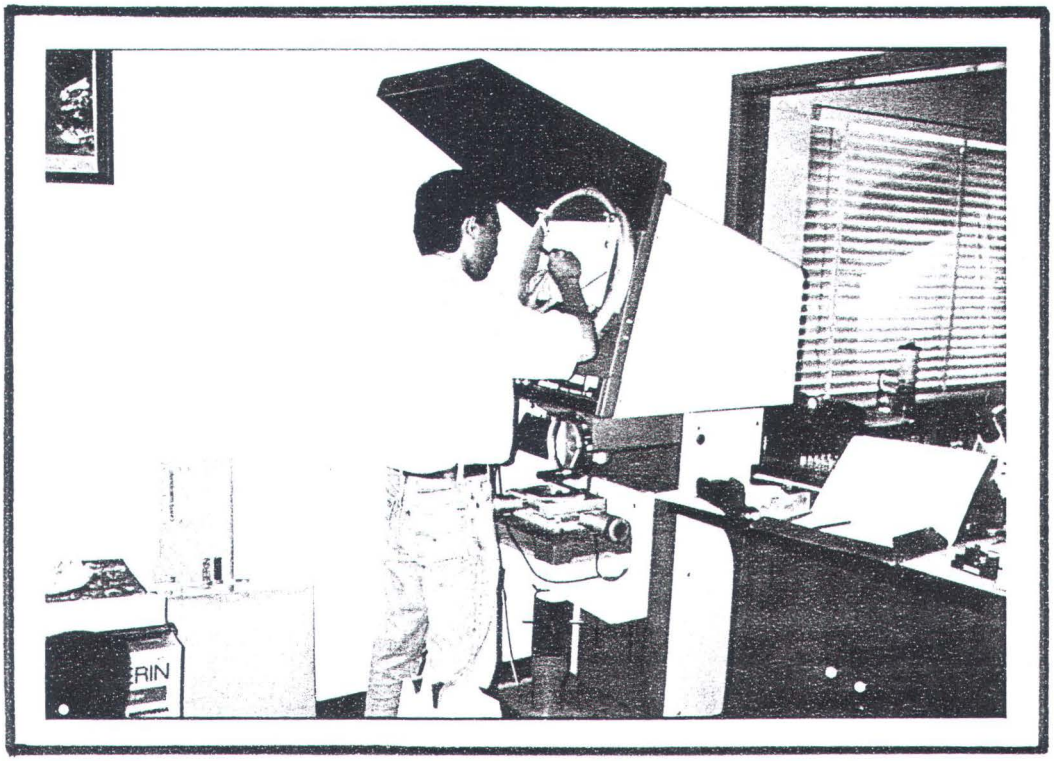
Gambar B.1 Perlakuan TIG Dressing



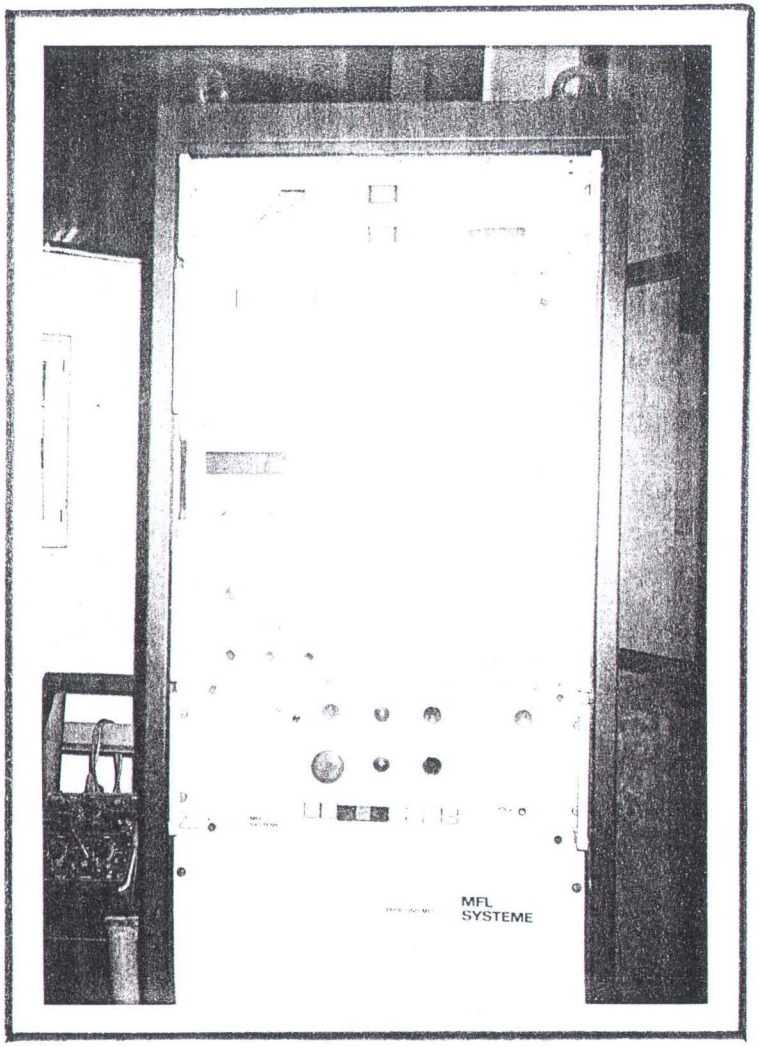
Gambar B.2 Proses machining pembentukan spesimen



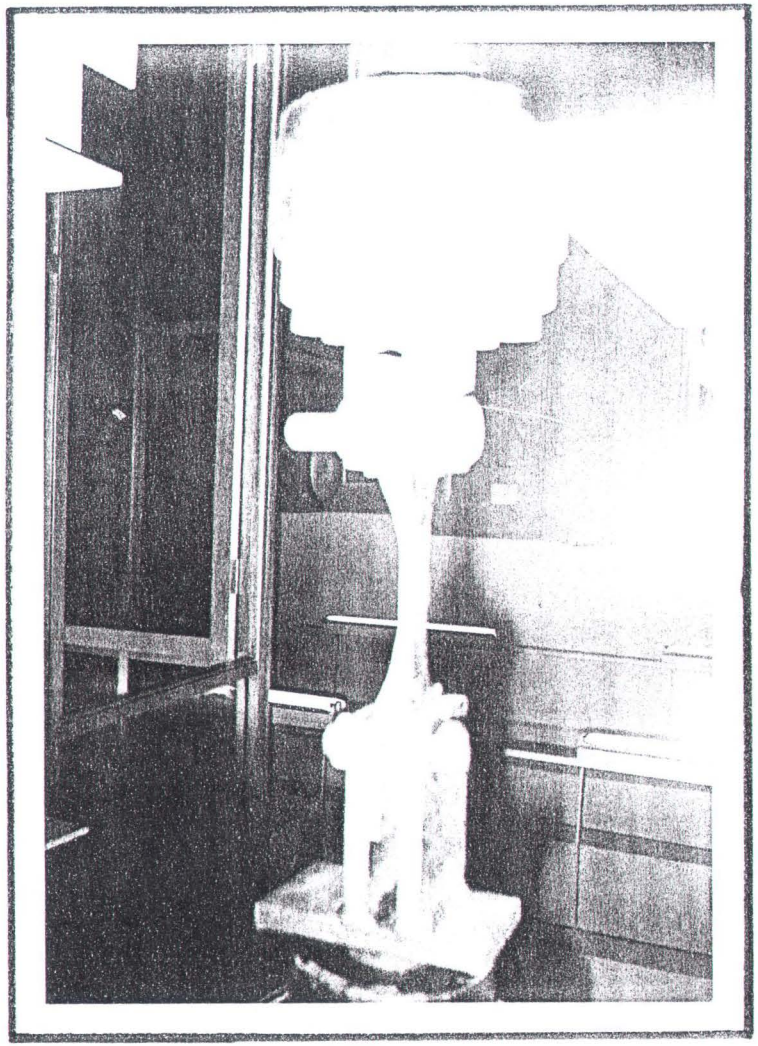
Gambar B.3 Spesimen single heating pass



Gambar B.4 Pengukuran flank angle dan jari-jari toe



Gambar B.6 Kontroler mesin uji fatigue



Gambar B.5 Spesimen double heating pass dalam pengujian fatigue

DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN
Kampus ITS - Sukolilo, Surabaya 60111 Telp. 5928105, 5994251 - 5 psw. 1104 Telex 34224 Fax 5947254

TUGAS AKHIR (OE.1702)

NO. : 101/PT.12.FTK4.H8/Q/96

MATA KULIAH	:	TUGAS AKHIR
NOMOR MATA KULIAH	:	OE.1702
NAMA MAHASWA	:	Dani Ivan Jaya
NOMOR POKOK	:	439200026
TGL.DIBERIKAN TUGAS	:	28 April 1997
TGL.SELESAINYA TUGAS	:	28 Desember 1997
DOSEN PEMBIMBING	:	1. Ir. Jusuf Sutomo, M.Sc
		2. Ir. Soeweify, M.Eng

THEMA / URAIAN / DATA-DATA YANG DIBERIKAN / JUDUL TUGAS AKHIR :

STUDI EKSPERIMENTAL PENINGKATAN KEKUATAN FATIGUE BUTT JOINT PADA MATERIAL ASTM A 516 GRADE 70 DENGAN TIG DRESSING

==========

Surabaya, 28 April 1997
Ketua Jurusan ;

DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS TEKNOLOGI KELAUTAN JURUSAN TEKNIK KELAUTAN

Ir.Eko Budi Djatmiko,MSc.
NIP.131 407 592

DIBUAT RANGKAP 4 :

1.Mahasiswa ybs.
2.Dekan FTK-ITS
3.Dosen Pembimbing
4.Arsip Jurusan .

DEPARTERMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN
Kampus ITS Sukolilo 60111 Telp 5947254, 5948757 & 5947274 Pes 261-262 Telex34224 Fax 5947845

LEMBAR PRESENSI KONSULTASI TUGAS AKHIR

NAMA MAHASISWA : DANI IVAN JAYA
NOMOR POKOK : 4392100027
NAMA DOSEN PEMBIMBING : 1. Ir. JUSUF SUTOMO, M.Sc.
2. Ir. SOEWEIFY, M.Eng.
TUGAS DIMULAI : 28 APRIL 1997
TUGAS DISELESAIKAN :
JUDUL TUGAS AKHIR : STUDI EKSPERIMENTAL PENINGKATAN KEKUATAN FATIGUE BUTT JOINT PADA MATERIAL ASTM A 516 GRADE 70 DENGAN TIG DRESSING.

No.	Tanggal	Konsultasi Mengenai	Tanda Tangan
1	/02 -97	Asistensi Bab I dan Bab II.	
	02/02 - 97	Konsultasi bab III, IV & V	
	03/03 - 97	Acc.	

Catatan : Lembar Presensi ini harus ditunjukkan pada saat Ujian Tugas Akhir.

DEPARTERMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN
Kampus ITS Sukolilo 60111 Telp 5947254, 5948757 & 5947274 Pes 261-262 Telex34224 Fax 5947845

LEMBAR PRESENSI KONSULTASI TUGAS AKHIR

NAMA MAHASISWA : DANI IVAN JAYA
NOMOR POKOK : 4392100027
NAMA DOSEN PEMBIMBING : 1. Ir. JUSUF SUTOMO, M.Sc.
2. Ir. SOEWEIFY, M.Eng.
TUGAS DIMULAI : 28 APRIL 1997
TUGAS DISELESAIKAN :
JUDUL TUGAS AKHIR : STUDI EKSPERIMENTAL PENINGKATAN KEKUATAN FATIGUE BUTT JOINT PADA MATERIAL ASTM A 516 GRADE 70 DENGAN TIG DRESSING.

No.	Tanggal	Konsultasi Mengenai	Tanda Tangan
1	3/6 -97	Laporan pelaksanaan.	
2	14/8 -97	Pengelasan Spesimen	
3.	22/08 - 97	Asistensi Pengefuran jari 2 toe	
4	16/10 -97	Pembentukan Material uji	
5.	20/10 -97	Finishing spesimen	
6.	16/1 -98	Laporan perkembangan T.A.	
7.	18/1 -98	Asistensi bab I & II	
8.	29/1 -98	Pengujian Fatigue, bab III & IV	
9.	03/2 -98	Perkembangan pengujian.	
10.	4/2 -98	Asistensi bab V	

Catatan : Lembar Presensi ini harus ditunjukkan pada saat Ujian Tugas Akhir.