

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

Oleh :

ABRAHAM AS'AD
NRP : 5196.100.045

PERPUSTAKAAN ITS	
Tgl. Terima	16-7-2003
Terima Dari	HI
No. Agenda Prp.	768249

**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



Dr. Ir. Joko Lianto Buliali, MSc

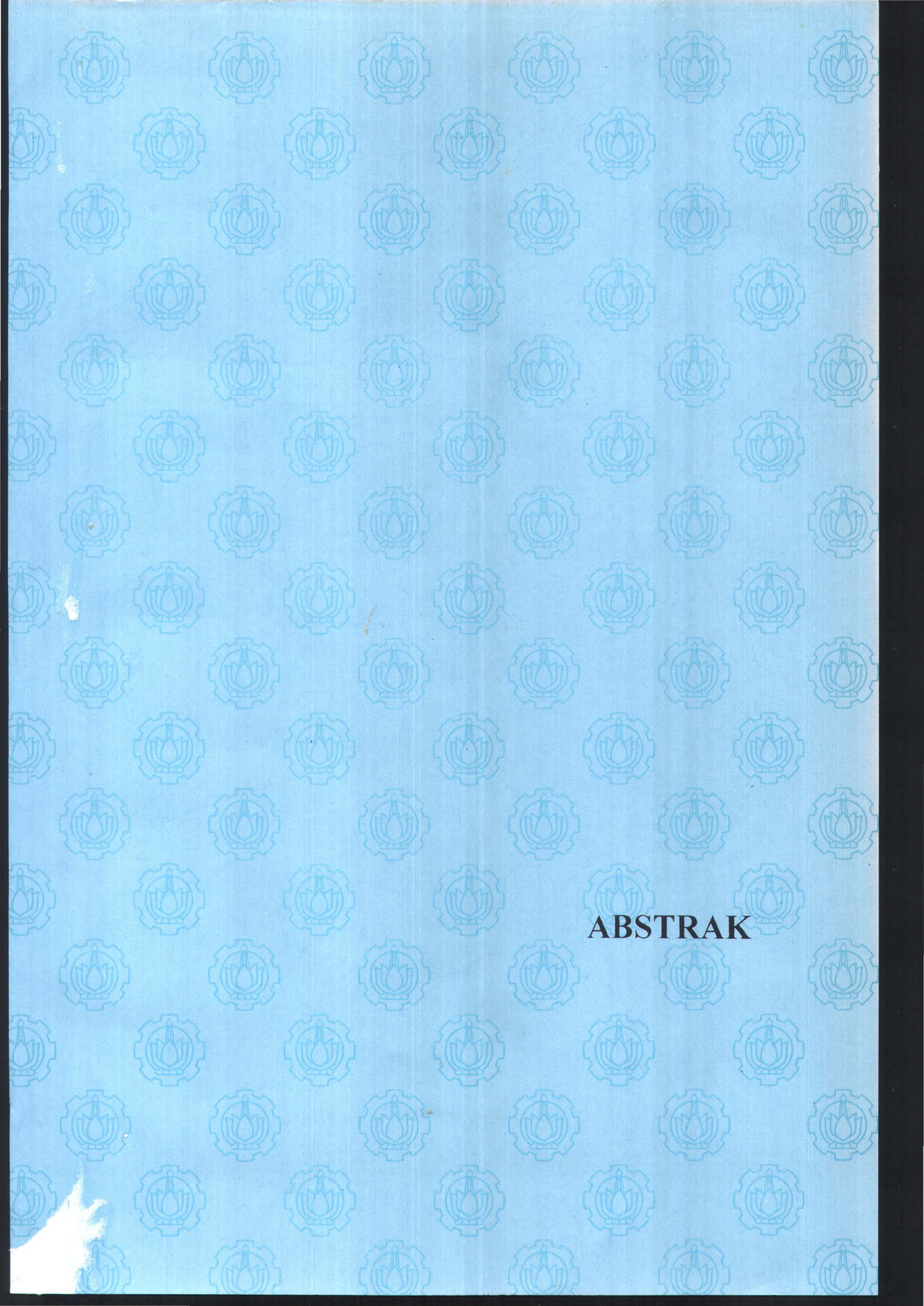
Dosen Pembimbing II



Dwi Sunaryono, S. Kom

SURABAYA

FEBRUARI 2002



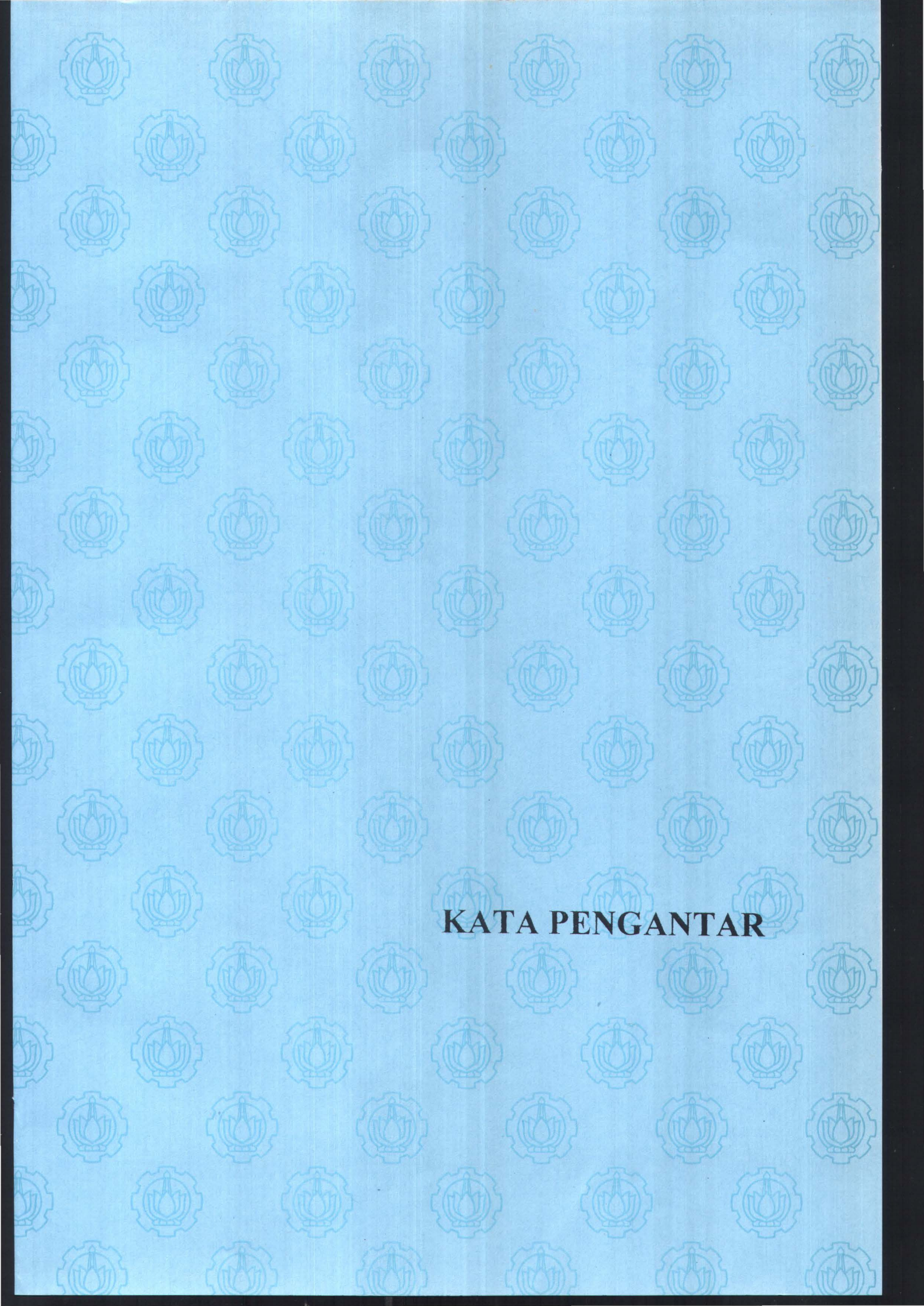
ABSTRAK

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 dimiliki 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, Dialah 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



ada kritik dan saran dari pembaca, Insya Allah akan diterima penulis dengan lapang dada.

Akhir kata penulis berdo'a semoga 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 <i>E-Business & E-commerce</i>	8
2.2 LOTUS NOTES / DOMINO	10
2.2.1 <i>Komponen Notes dan Domino</i>	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 <i>Notes</i>	17
2.2.2.1 Item-Item Notes	19
2.2.2.2 Type-Type 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 <i>Bahasa Pemrograman</i>	24
• Agent dan Action	25
2.3 NOTES DAN WEB	26
2.3.1 <i>Domino Web Server</i>	27
2.3.2 <i>Arsitektur Domino Web Server</i>	31
Komponen-Komponen HTTP Server Task	34
2.4 KEAMANAN LOTUS NOTES / DOMINO	36
2.5 UNIFIED MODELING LANGUAGE (UML)	40
2.5.1 <i>Diagram UML [CRA - 98]</i>	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

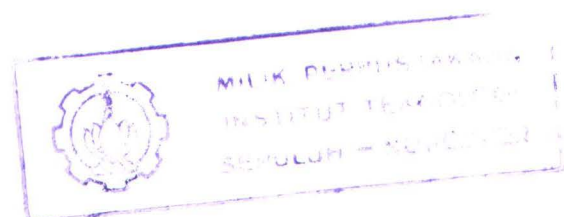
• View BukuTerbaru	90
• View KatalogHasilPencarian	90
• View KatalogperKategori	90
• View KatalogBuku	90
• View (CartDetailByID)	90
4.4 AGENT	90
4.4.1 CekStokBuku	90
4.4.2 AddtoCart	92
4.4.3 UpdateCart	93
4.4.4 ClearCart	95
4.4.5 ProsesKartuKredit	96
4.4.6 DaftarAnggota	98
4.5 SCRIPT LIBRARIES	101
BAB 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 SEQUENCE LOGIN NOTES	61
GAMBAR 3.19. DIAGRAM SEQUENCE LOGIN 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 DISPLAY KATALOG	74
GAMBAR 4.1. FORM REGISTRASI	76
GAMBAR 4.3. FORM ENTRI KARTU KREDIT	77
GAMBAR 4.4. FORM LUPA PASSWORD	78
GAMBAR 4.5. FORM KERANJANG BELANJA	79
GAMBAR 4.6. FORM HALAMAN UTAMA	80
GAMBAR 4.7. FORM KONFIRMASI	81
GAMBAR 4.8. FORM ORDER	81
GAMBAR 4.9. FORM ORDER STATUS CONFIRM	82
GAMBAR 4.10. FORM ORDER THANKS	83
GAMBAR 4.11. FORM CARA PEMBAYARAN	84
GAMBAR 4.12. FORM CARA PEMESANAN	84
GAMBAR 4.13. FORM CARA PENGIRIMAN	85
GAMBAR 4.14. FORM REG CHOICE	86
GAMBAR 4.15. FORM STATUS PESANAN	87

GAMBAR 4.16 FORM VIEW KATALOG PER KATEGORI	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 MANAJEMEN DATA 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/IIOP 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



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 tahun 1994, 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 detail transaksi.

- ***Business-to-Consumer (B2C)*** : Menyediakan akses kepada konsumen untuk menambah, menghapus, dan mengedit suatu data di basisdata perusahaan. 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 detail 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 mengizinkan 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.

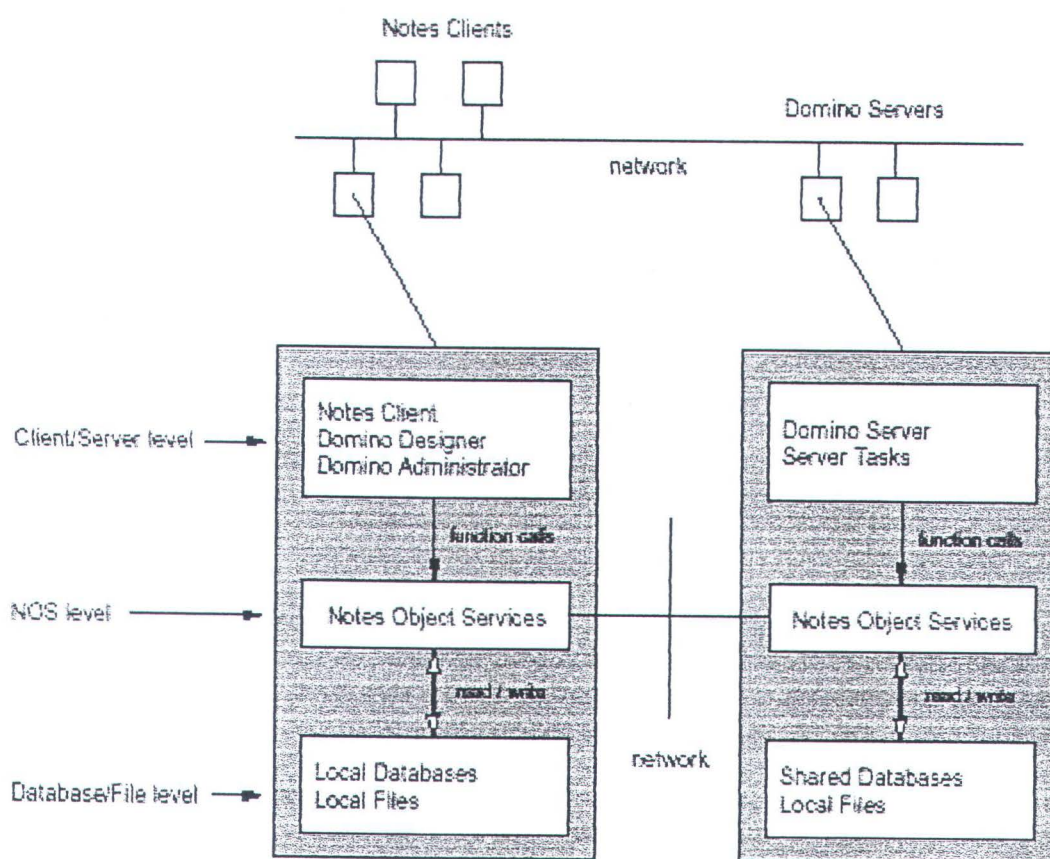
2.2 Lotus Notes / Domino

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*.

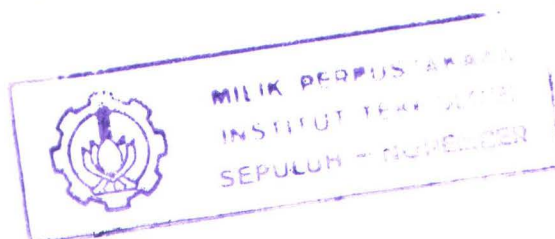
Pada saat fasilitas HTTP ditambahkan ke dalam sistem, Notes dan Domino merupakan *true application server* untuk internet yang pertama kali. Web server yang dibangun dengan Domino server ini, memiliki keistimewaan dapat sekaligus menjadi *firewall* bagi keluar masuknya data dari dan ke perusahaan yang 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



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 menggunakan 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 :

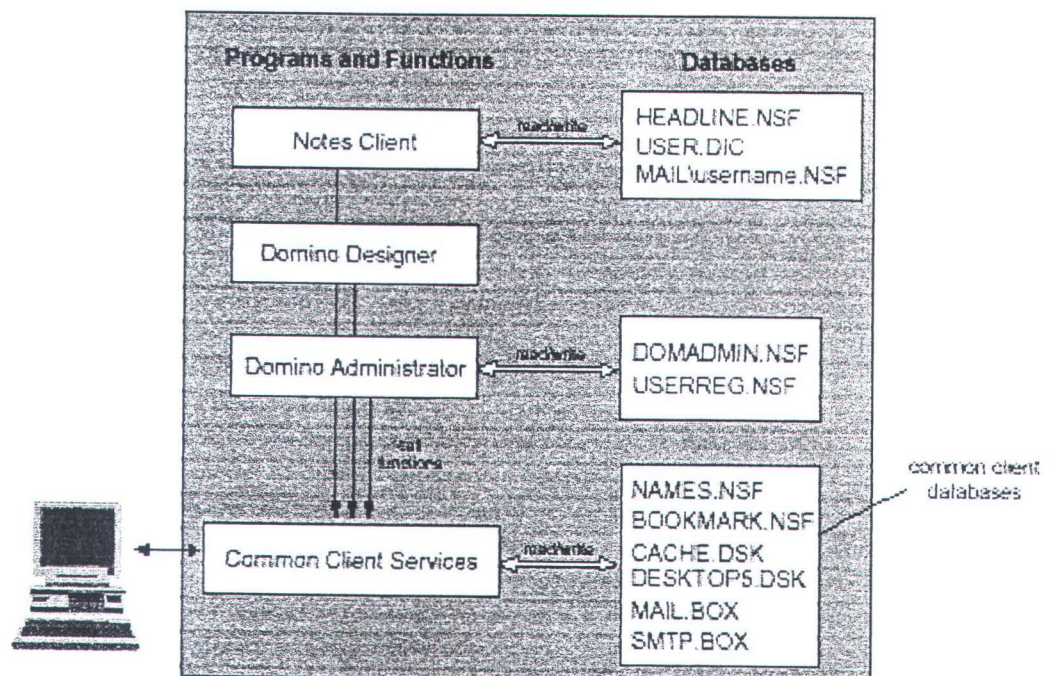
- Elemen-elemen perancangan Notes, seperti form dan view.
- Elemen-elemen perancangan Web, seperti frameset dan page.
- Bahasa Formula Notes, yang meliputi @command dan @function LotusScript, JavaScript dan Java.

- **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 keandalannya 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.

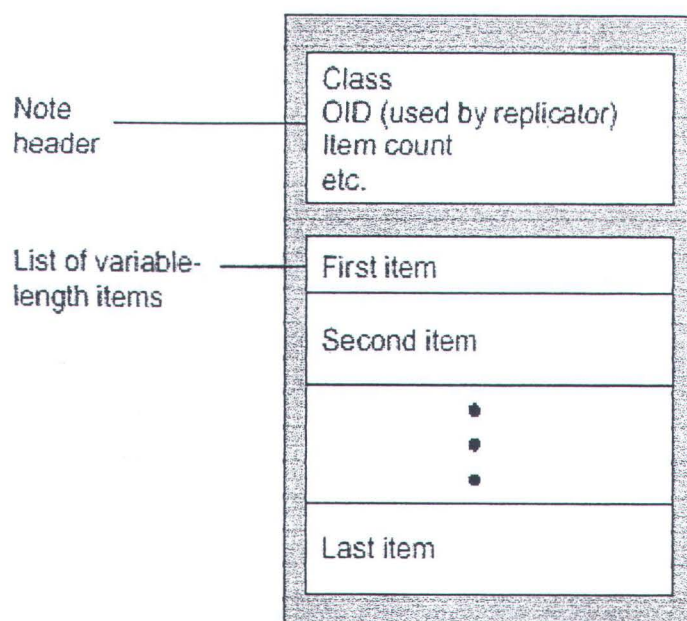
Sebuah database ataupun file adalah lokal, jika hanya terakses oleh program yang berjalan dalam komputer yang sama. Database dalam komputer *client* adalah lokal sebab program-program *client* tidak mempunyai pelaksanaan logis dalam program Domino Server, untuk merespon datangnya permintaan database. 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 tersebut 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*.



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

Design-Element	FILTER	Agent, database script, outline, script library
	FORM	Form, frameset, page, subform
	VIEW	Folder, navigator, view
	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 memodifikasinya.

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 designer 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 individual 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-tugas 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, yaitu :

- 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.

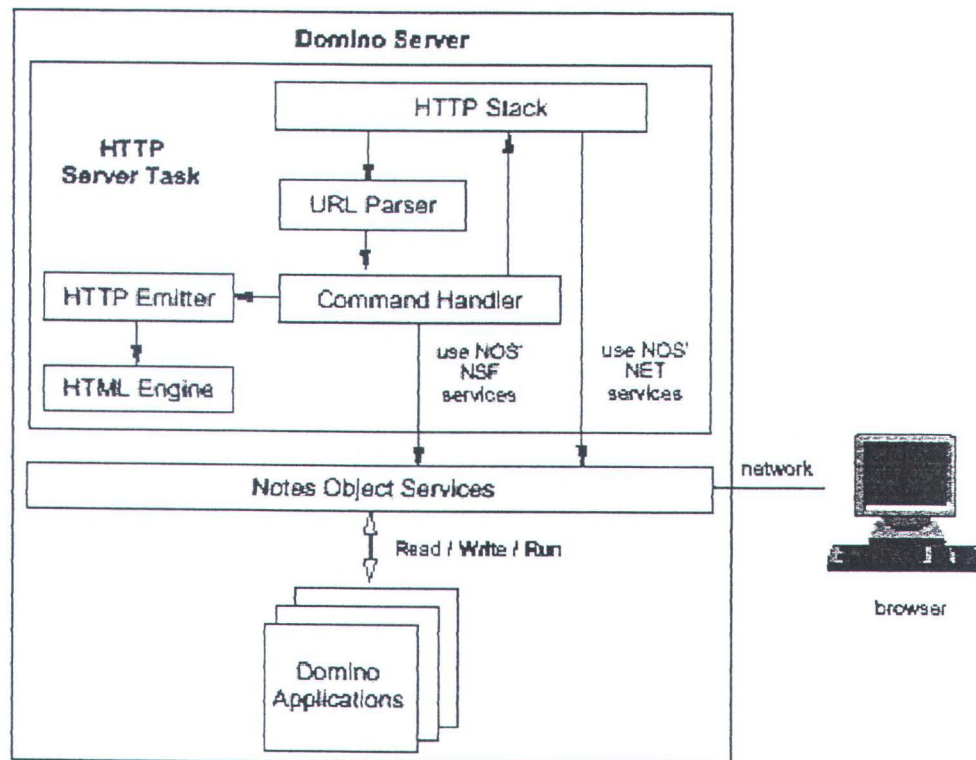
Generasi pertama *web server* adalah berdasarkan *file server* yang diterima *file request* melalui koneksi jaringan dan direspon dengan mengambil file dari hard disk serta meng-*copy*-nya ke jaringan.

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.



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, menggambarkan 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*.

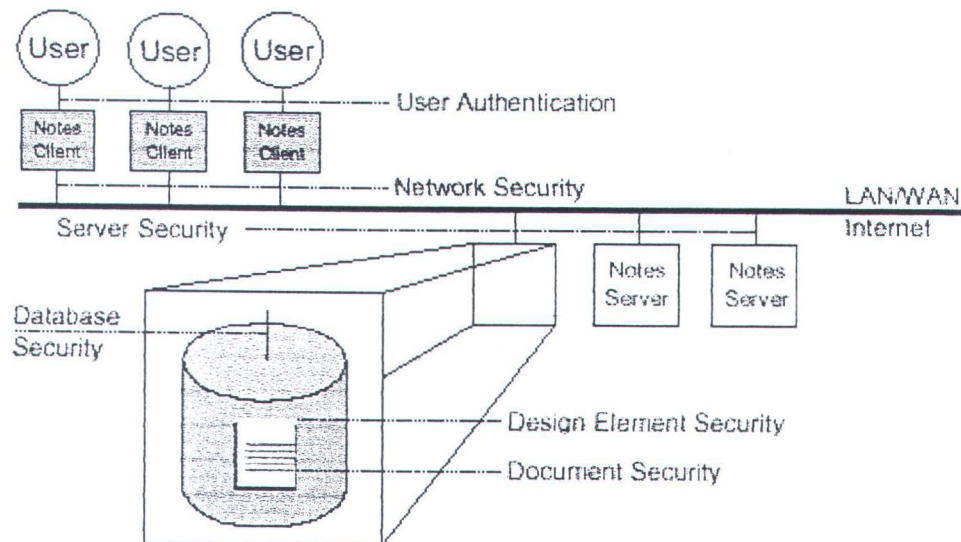
- **HTML Emitter dan Engine**

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 keamanan 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 password (*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 sendiri 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 menggunakan 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	Depo sitor	No Acce ss
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	Optio n
Write Public Document	√	√	√	Option	Option	Opti on	Optio n
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 informasi tetapi juga membiarkan pengguna untuk berinteraksi dengan aplikasi. Dengan menggunakan sintaks URL, pengguna *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 berpartisipasi 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 *parallel*. 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 *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 :

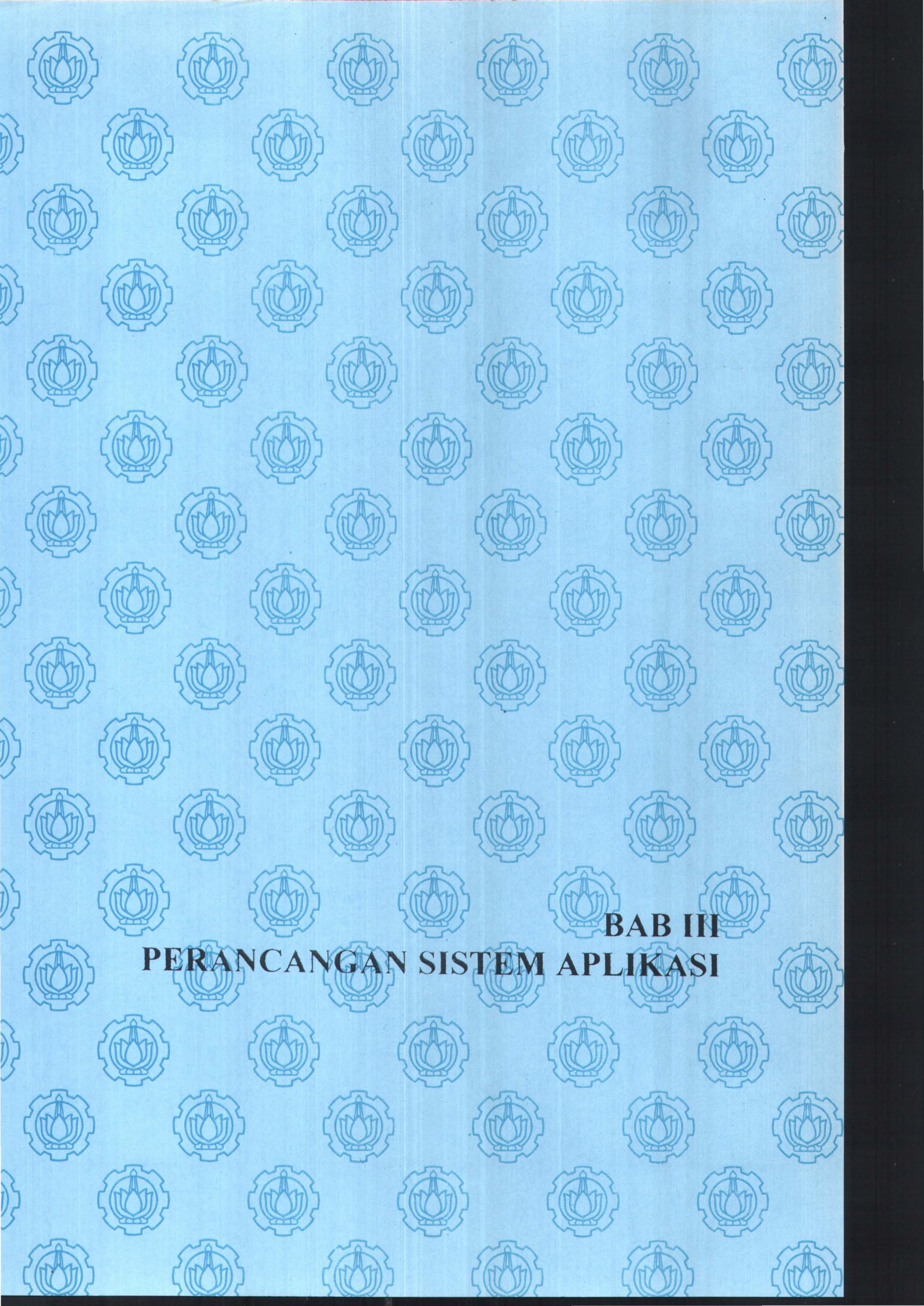
- Aktifitas / Activity
- Class
- Collaboration
- Component
- Deployment
- Object
- Use case
- Sequence
- State

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

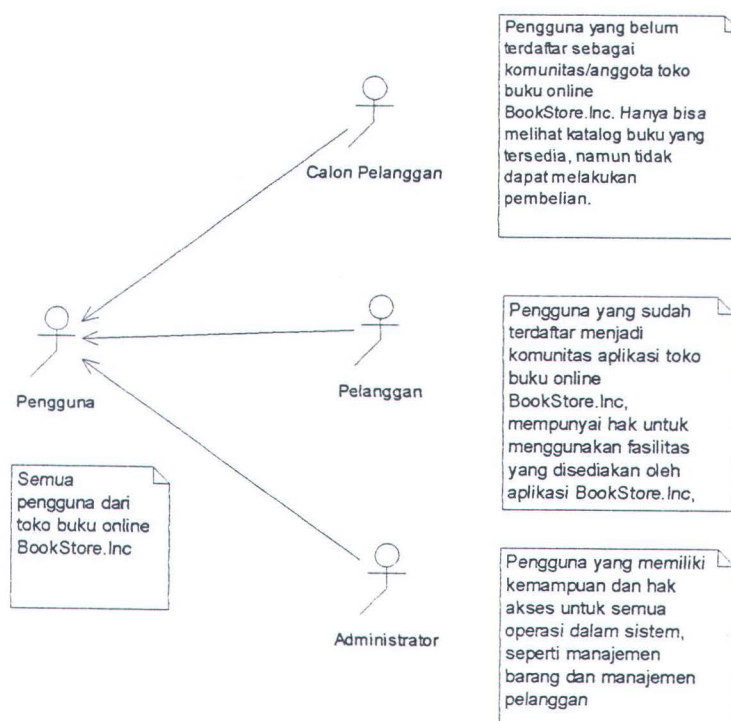


3.1 USE CASE VIEW

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.

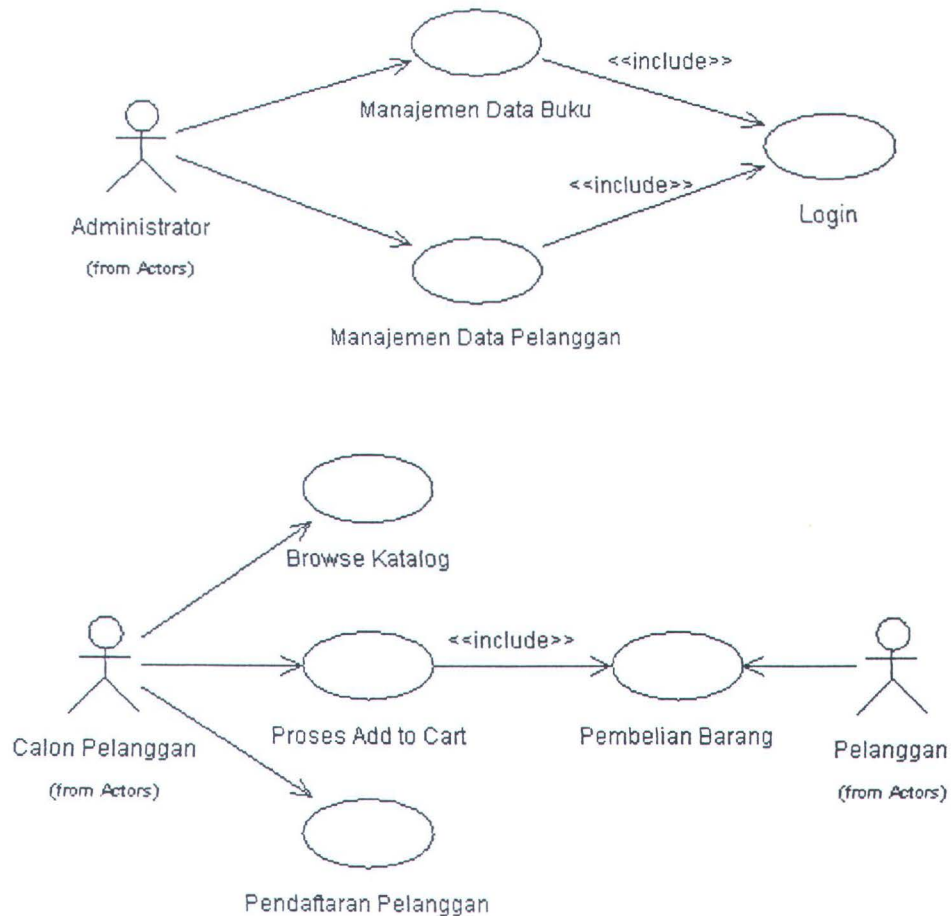
2.2.3 Diagram Use Case

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

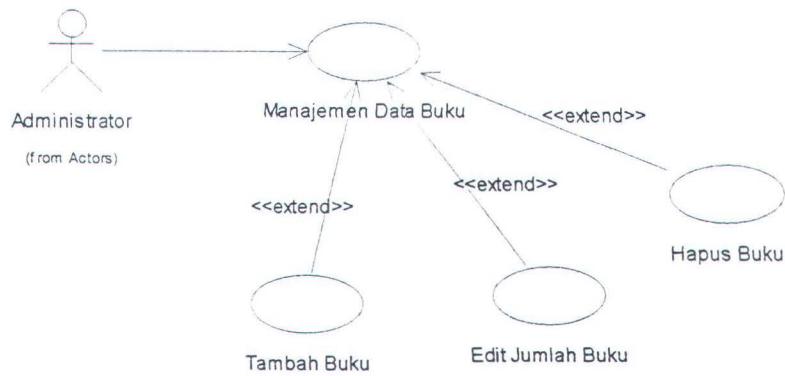
Dari gambar diatas dapat dilihat bahwa ada dua proses utama yang dapat dilakukan oleh administrator yaitu proses manajemen data buku dan manajemen data pelanggan.

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

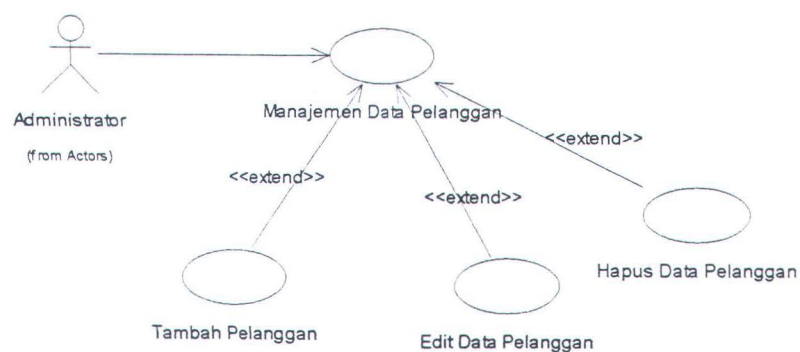
Manajemen terhadap data buku hanya bisa dilakukan oleh administrator. Manajemen yang dimaksud meliputi aktifitas-aktifitas menambah, menghapus dan memodifikasi data dari buku. Proses ini dapat dilihat pada gambar 3.3 di bawah ini.



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 .

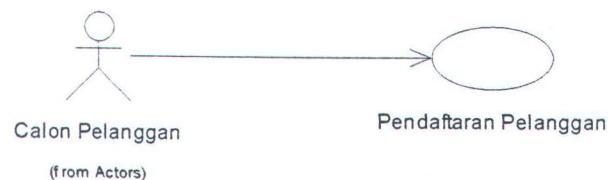


Gambar 3.4. Diagram Use Case Manajemen Pelanggan



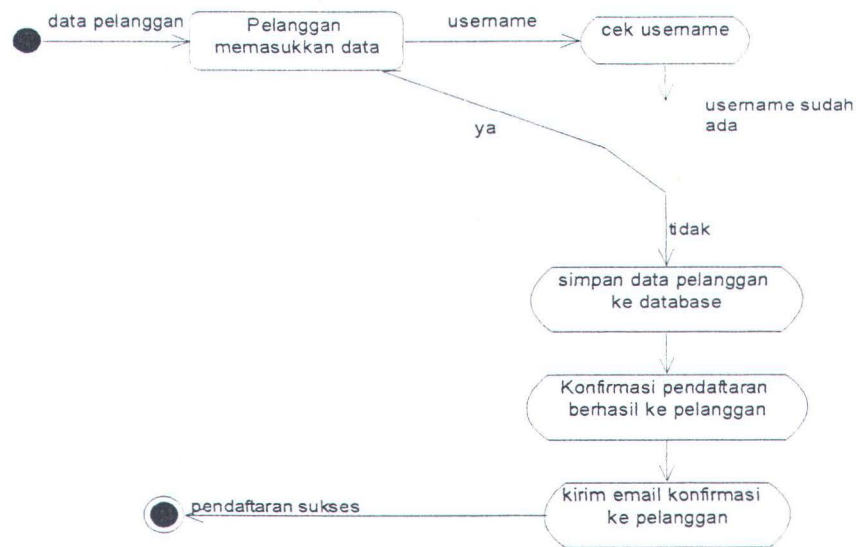
3.1.1.3 Diagram Use Case Pendaftaran Pelanggan

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

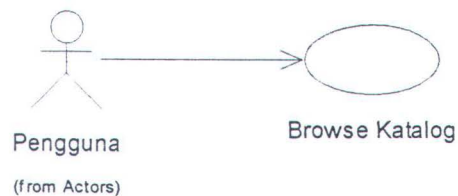
Selain diagram di atas, dibuat juga activity diagram untuk menjelaskan proses pendaftaran pelanggan seperti terlihat pada gambar 3.6. berikut ini.



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.

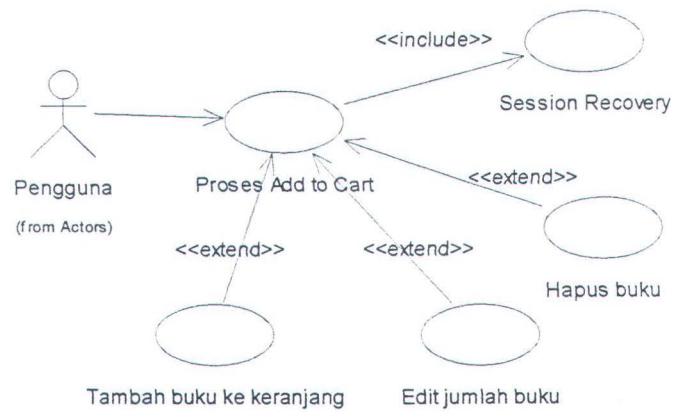


Gambar 3.7. Diagram Use Case Browse Katalog

3.1.1.5 Diagram Use Case Add to Cart

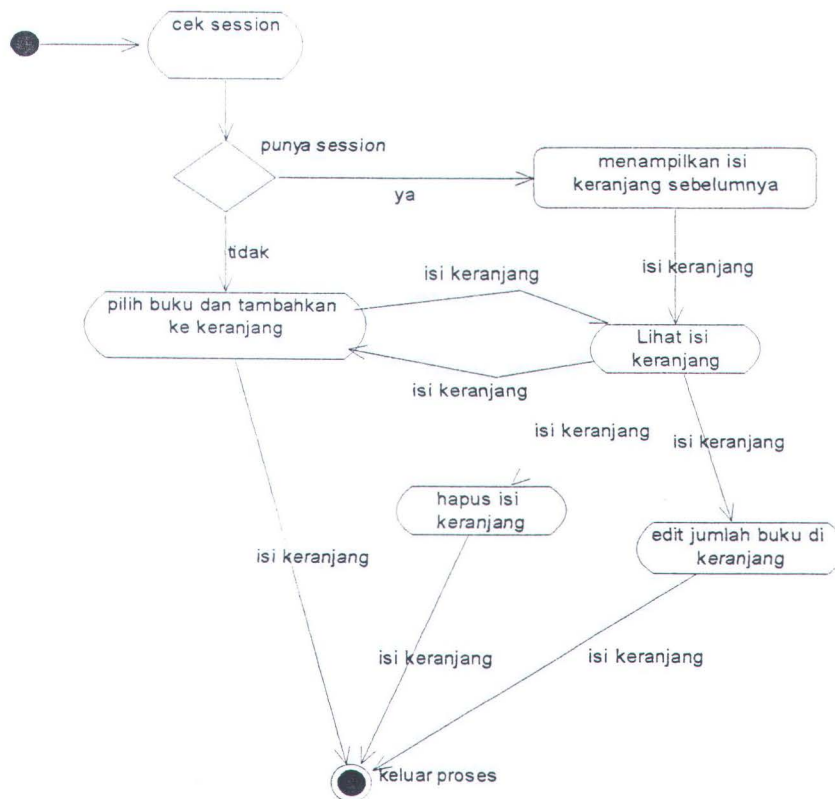
Didalam use case ini terdapat subproses-subproses menambah, menghapus, dan mengedit jumlah buku di dalam keranjang belanja, serta session recovery. Session recovery mengecek apakah pengguna pernah melakukan proses

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

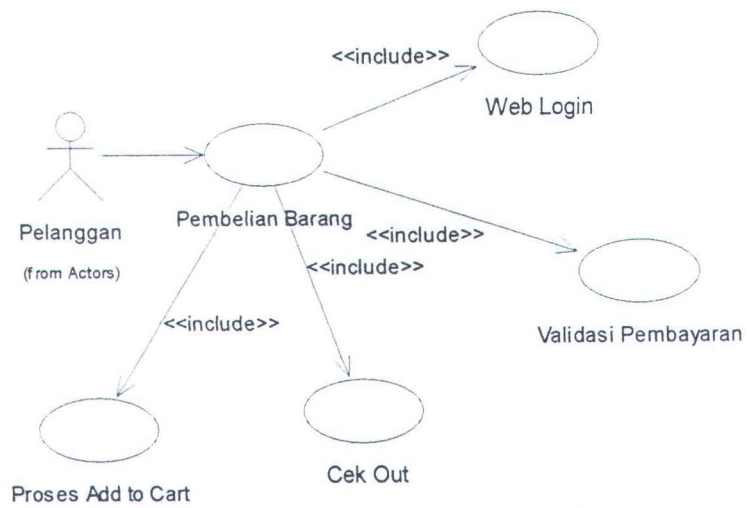
Aliran kejadian dari use case ini dapat digambarkan melalui activity diagram pada gambar3.9.



Gambar 3.9. Diagram Activity Add to Cart

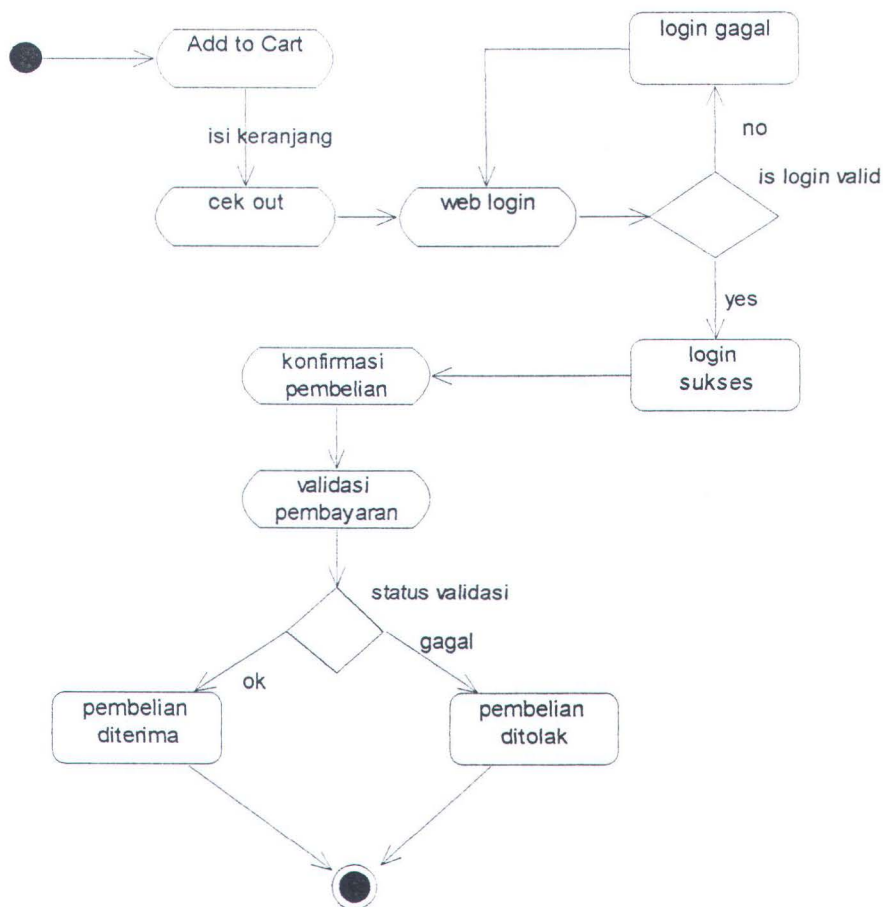
3.1.1.6 Diagram Use Case Pembelian Buku

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, berada pada data service.

3.1.3.1 Diagram Class Manajemen Data Buku

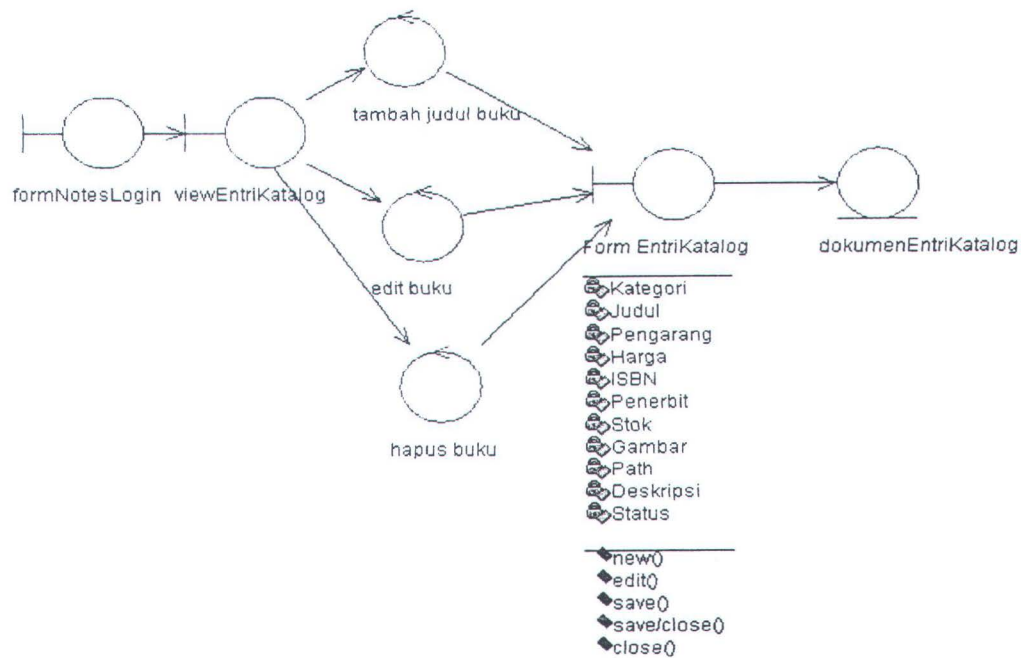
Berdasarkan diagram use case manajemen buku, dibuat rancangan class seperti tampak pada tabel di bawah ini.

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.





Gambar 3.12. Diagram Class pada Use Case Manajemen Data Buku

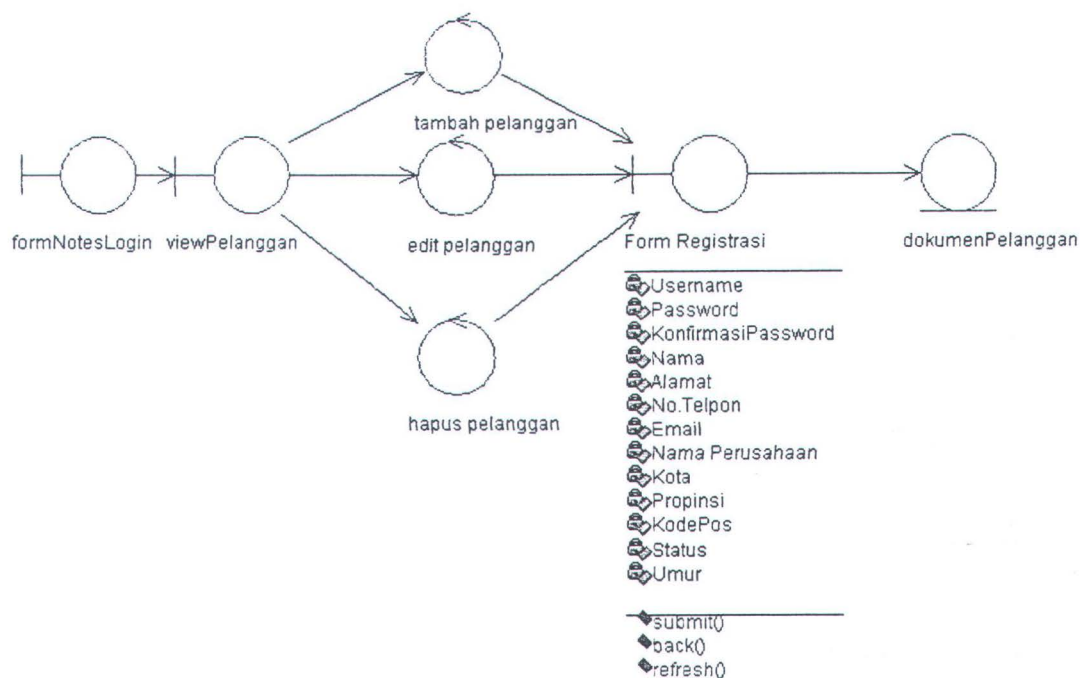
3.1.3.2 Diagram Class Manajemen Pelanggan

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

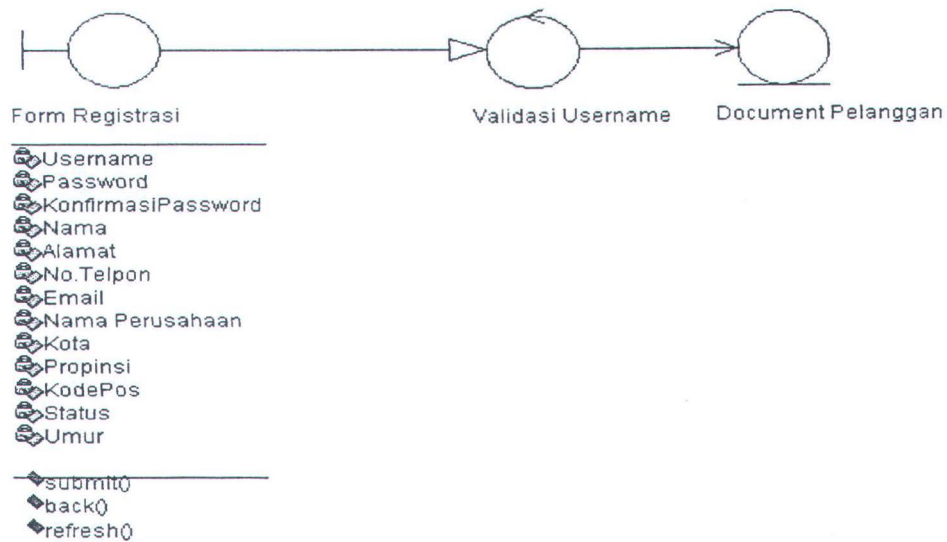
3.1.3.3 Diagram Class Pendaftaran Pelanggan

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

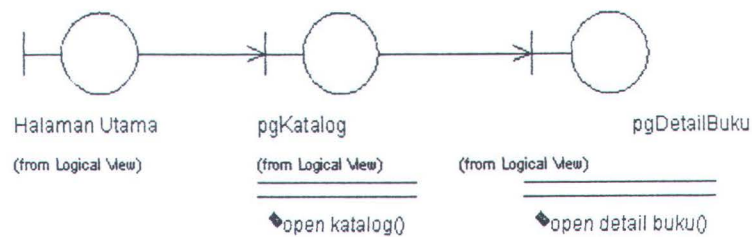
3.1.3.4 Diagram Class Browse Katalog

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

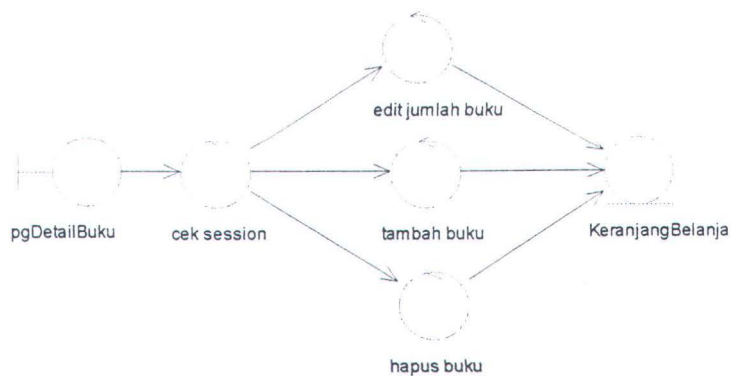
3.1.3.5 Diagram Class Add to Cart

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

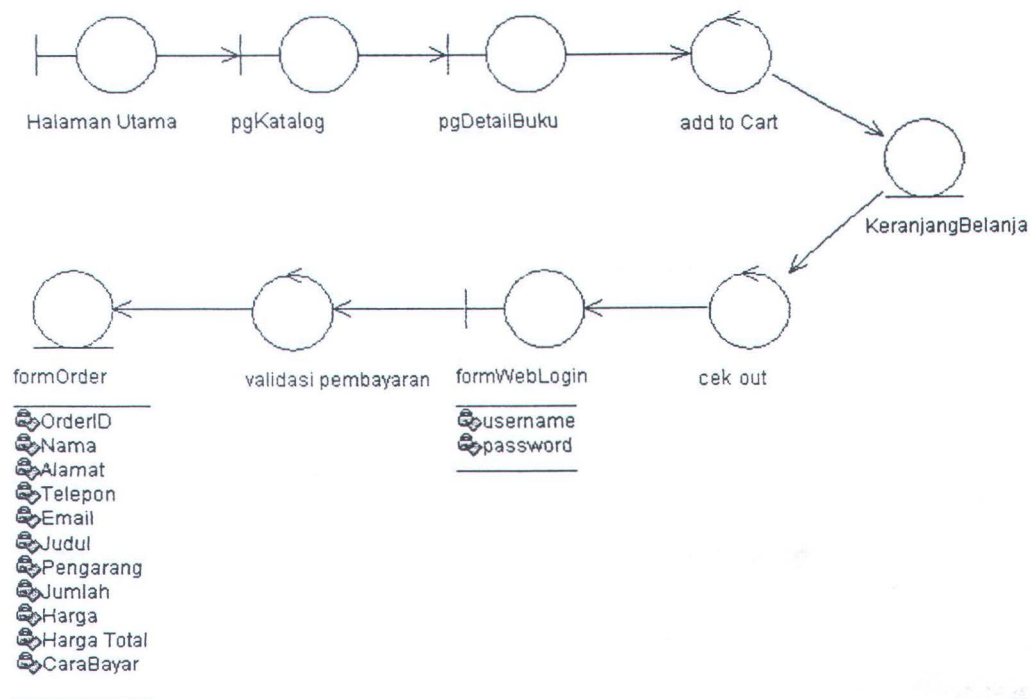
3.1.3.6 Diagram Class Pembelian Barang

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

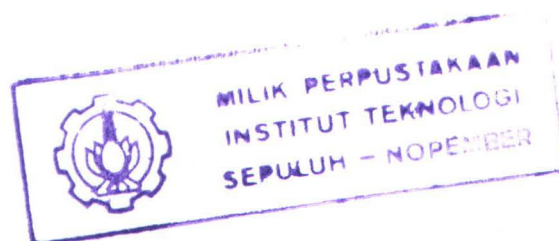
Bentuk diagram class dari use case ini adalah seperti pada gambar 3.17. berikut ini.



Gambar 3.17. Diagram Class pada Use Case Pembelian Barang

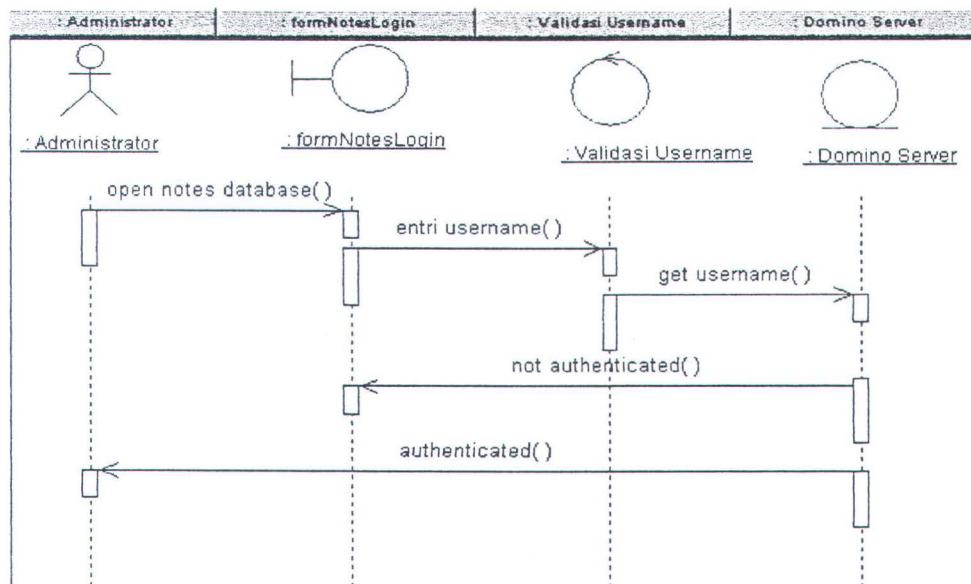
3.1.4 MODEL OBJEK DINAMIS

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.



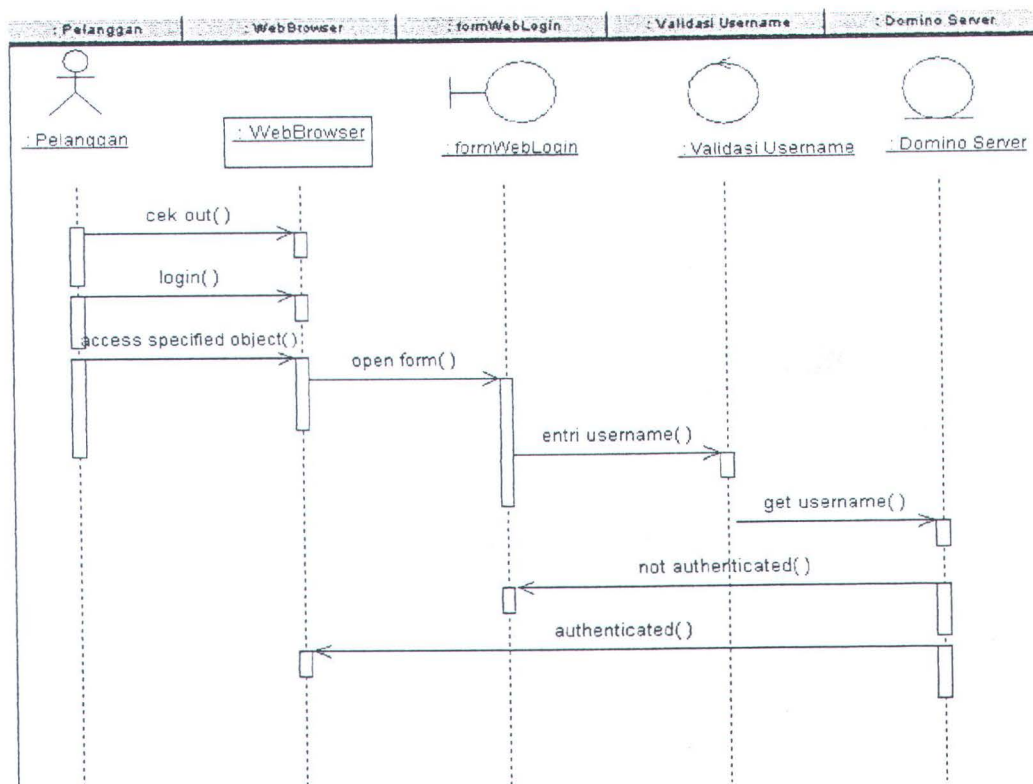
3.1.4.1 Diagram Sequence Login

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

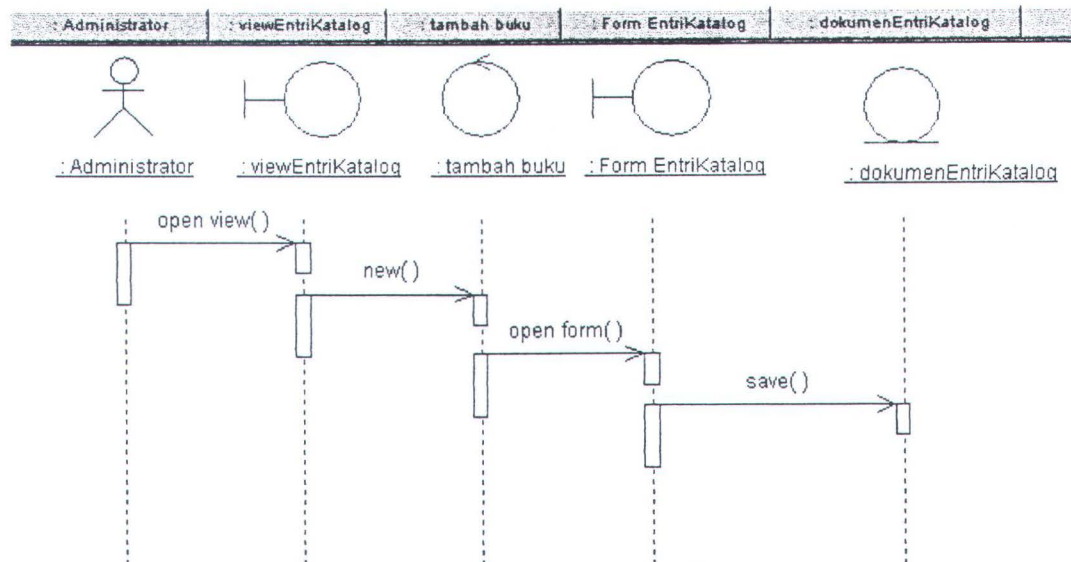
Sedangkan interaksi yang terjadi antar obyek/class pada saat login web dapat dilihat pada diagram sequence pada gambar 3.19.



Gambar 3.19. Diagram SequenceLogin Web

3.1.4.2 Diagram Sequence Manajemen Data Buku

Interaksi yang terjadi antar objek/class pada use case manajemen data buku dapat dilihat pada sequence diagram pada gambar 3.20.

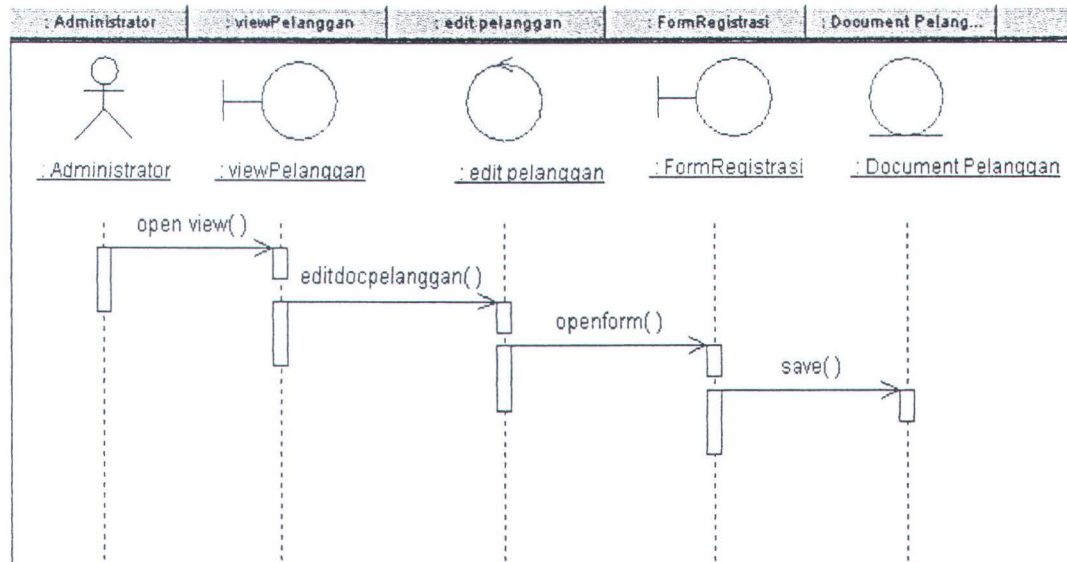


Gambar 3.20. Diagram Sequence Manajemen Data Buku

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 objek/class pada use case manajemen pelanggan dapat dilihat pada sequence diagram berikut ini.

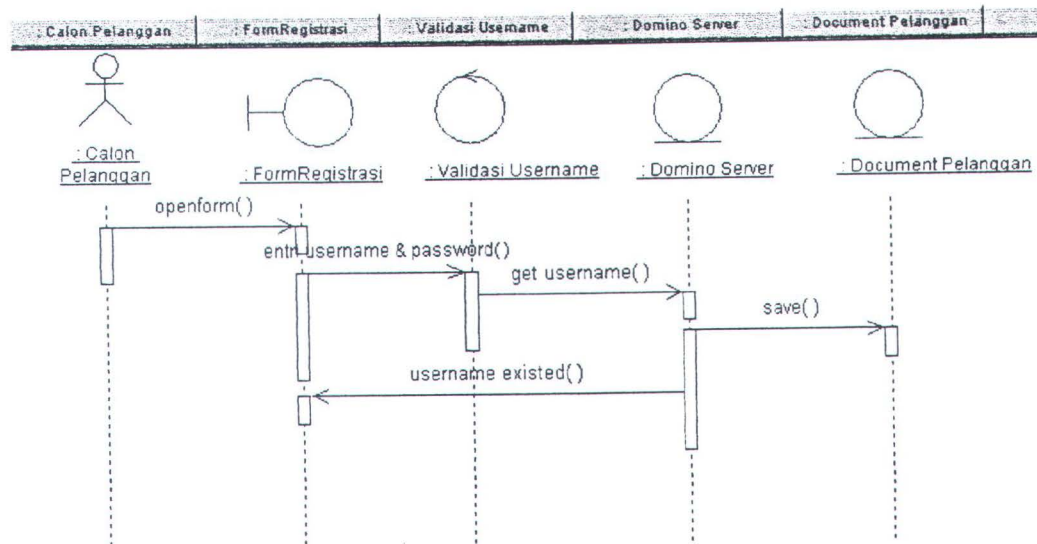


Gambar 3.21. Diagram Sequence Manajemen Pelanggan

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.

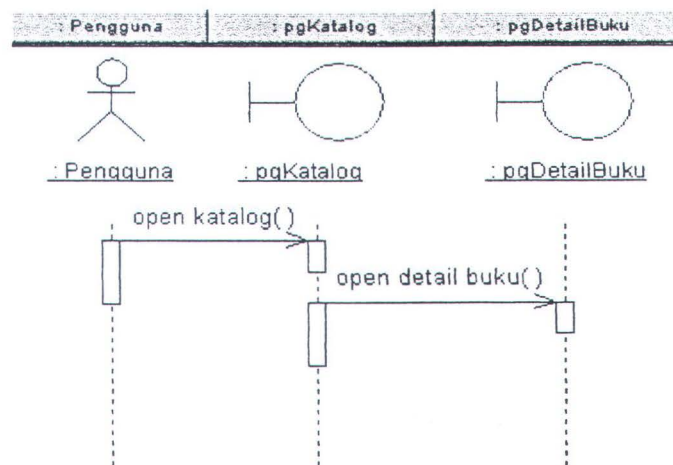


Gambar 3.22. Diagram Sequence Pendaftaran Pelanggan

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 objek/class pada use case browse katalog dapat dilihat pada diagram sequence berikut ini.



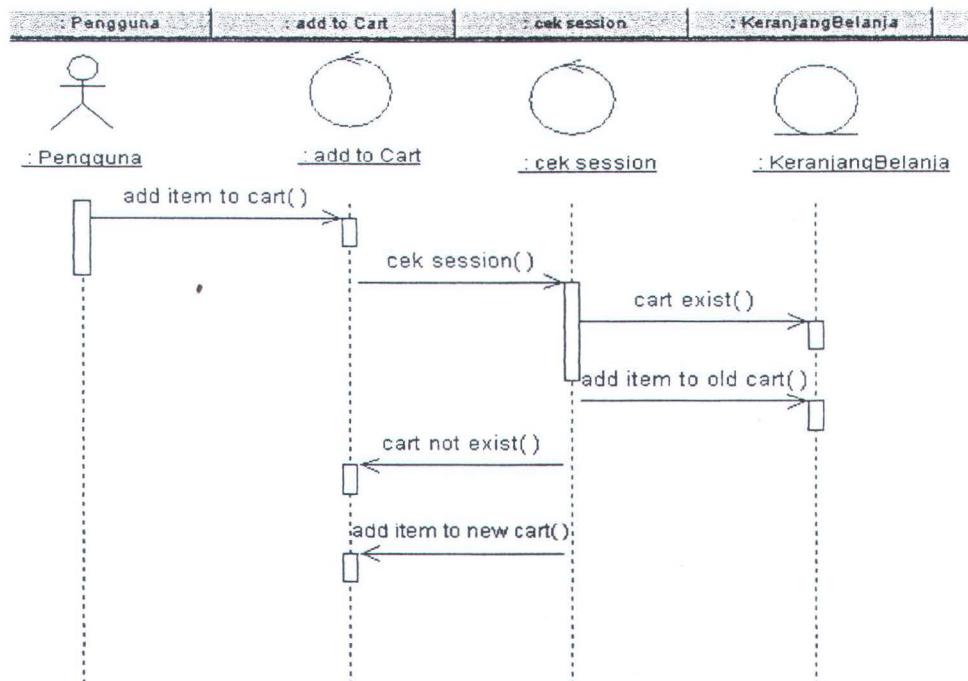
Gambar 3.23. Diagram Sequence Browse Katalog

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.

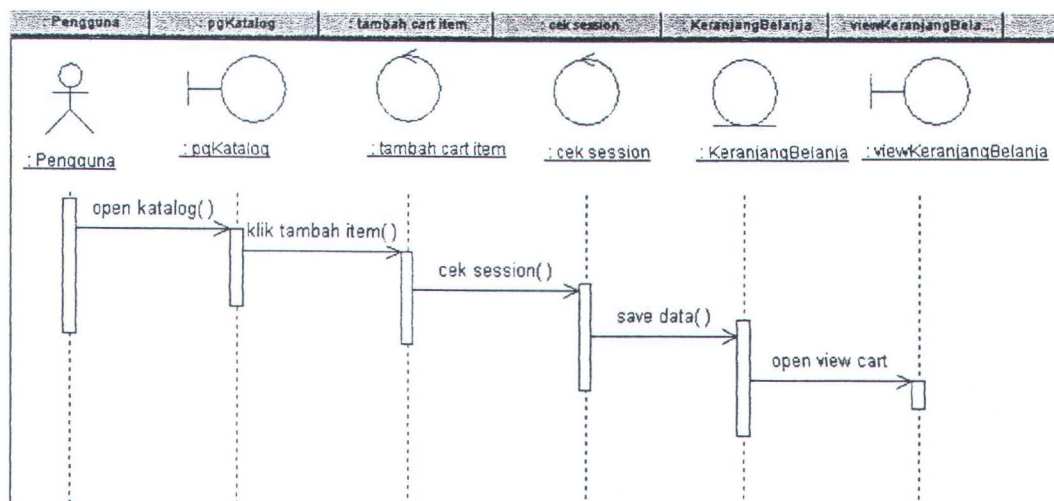
Diagram sequence pada gambar 3.24 berikut ini menggambarkan sub proses cek session.





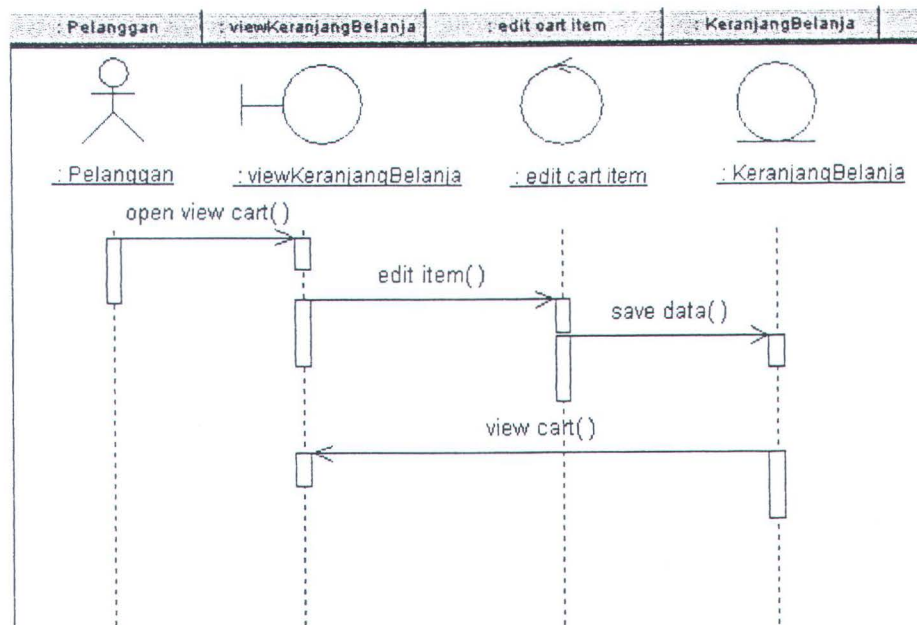
Gambar 3.24. Diagram Sequence Add to Cart sub proses Cek Session

Dibawah ini adalah diagram sequence untuk menambah item ke keranjang belanja.



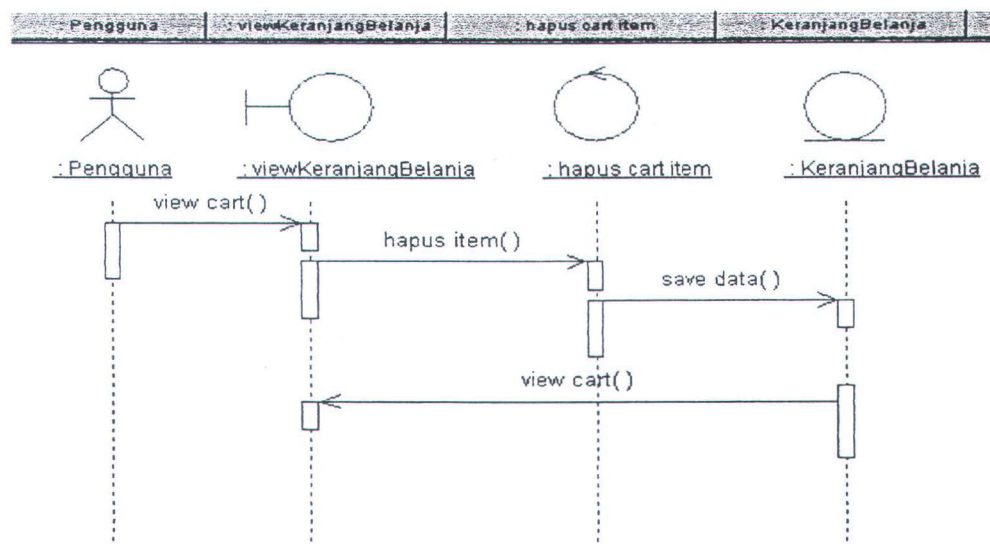
Gambar 3.25. Diagram Sequence Add to Cart sub proses Tambah Item

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 objek/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.

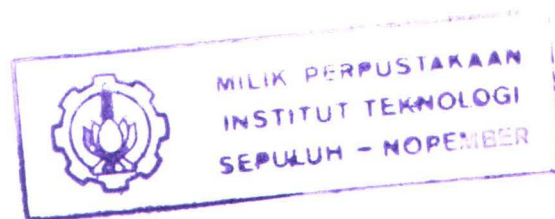
Sub proses pertama yang harus dijalankan adalah add to cart, kemudian dilanjutkan dengan sub proses cek out, yang akan menjalankan sub proses web login dan kemudian diakhiri dengan sub proses validasi pembayaran.

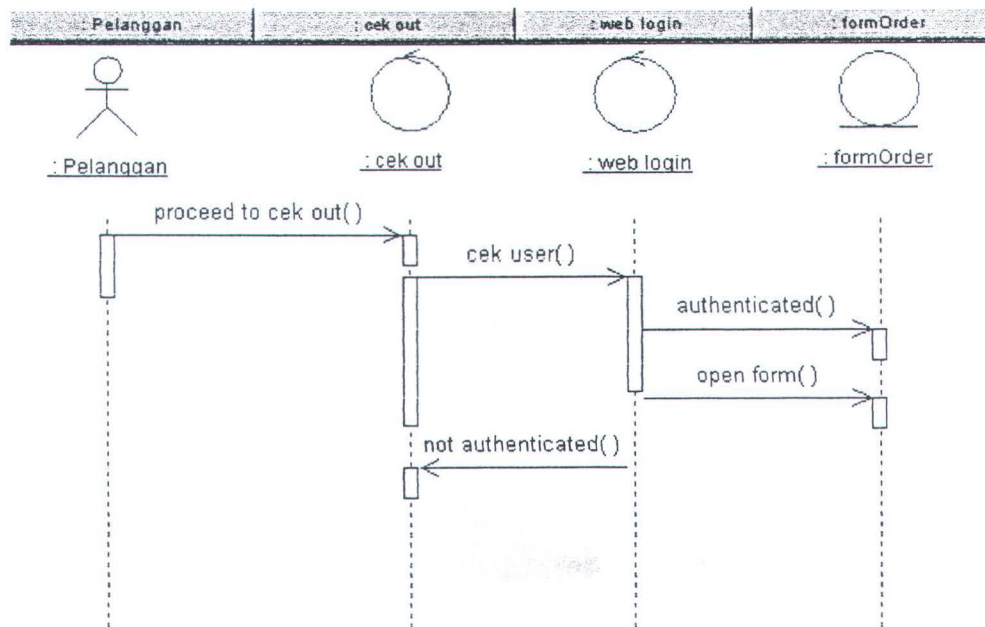
Sub proses Add to Cart dapat dilihat pada gambar 3.24, gambar 3.25, dan gambar 3.26 dan 3.27. Sedangkan sub proses web login dapat dilihat pada gambar 3.19.

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.

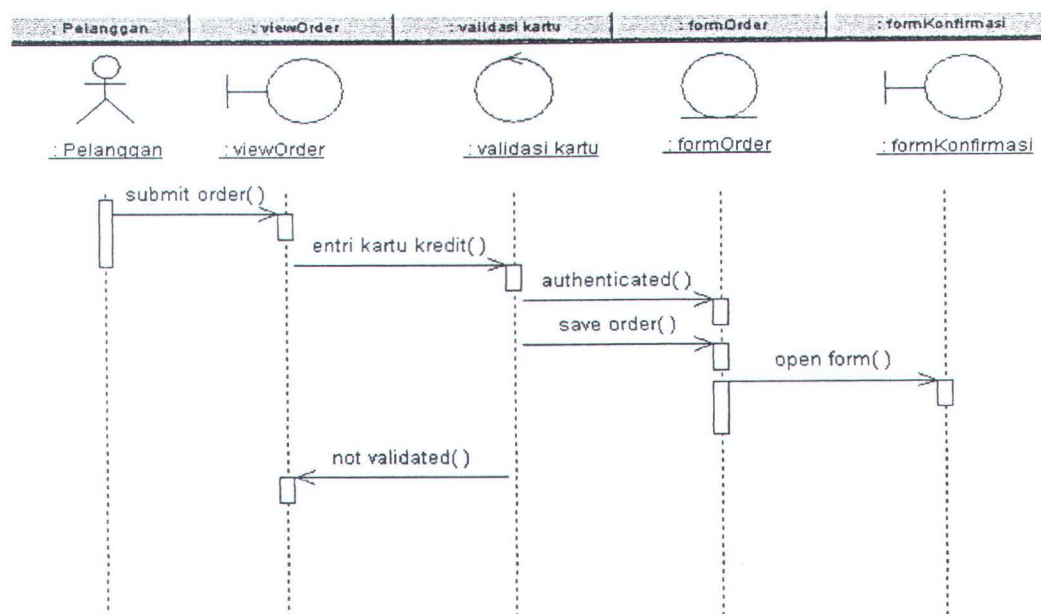




Gambar 3.28. Diagram Sequence Pembelian Buku sub proses Cek Out

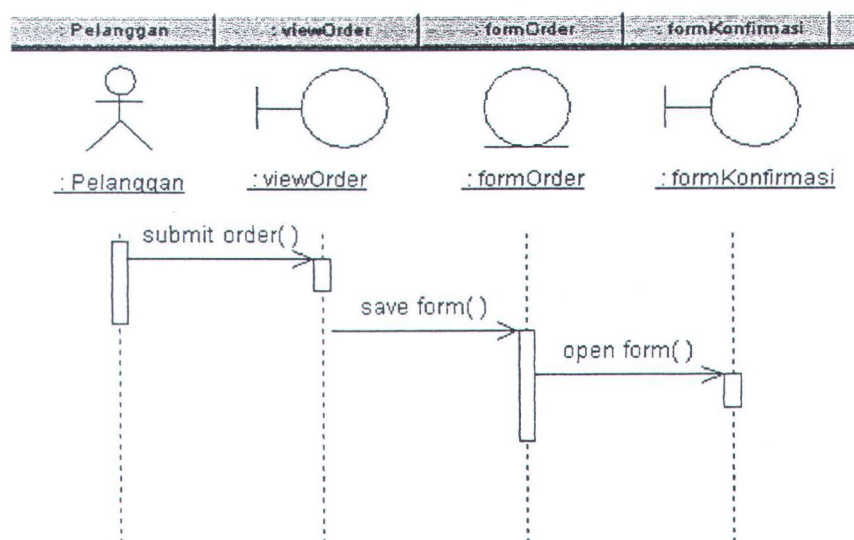
- **Diagram Sequence Sub Proses Validasi Pembayaran**

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 (Access 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 digunakan 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.

Informasi untuk Mendaftarkan Barang

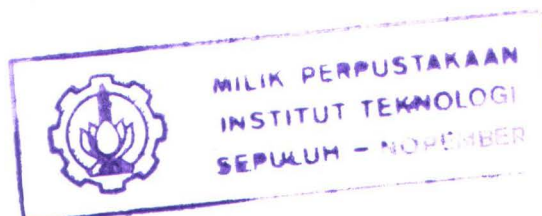
Form ini memiliki dua aksi, yaitu submit untuk memproses data pengguna dan reset untuk mengosongkan field-field yang diisi oleh pengguna.

Form ini mempunyai field-field Username untuk pengisian username, 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.

Gambar 4.1. Form Registrasi

Form ini mempunyai field-field Username untuk pengisian username, 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.



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.

Common Action Buttons subform

Entri Informasi Kartu Kredit :

Nomor Kartu Kredit

CCNo

Password

CCPassword

Saldo

CCSaldo

DispSaldo

\$\$Return SaveOptions DBpath

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.

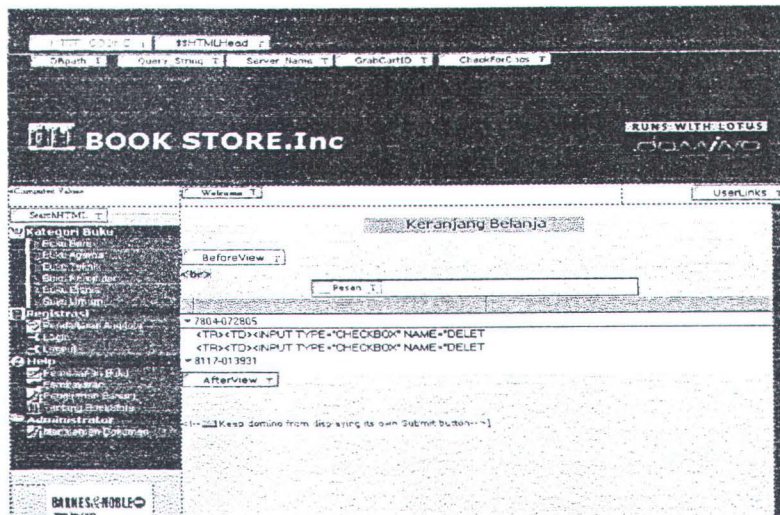
Gambar 4.4. Form LupaPassword

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.

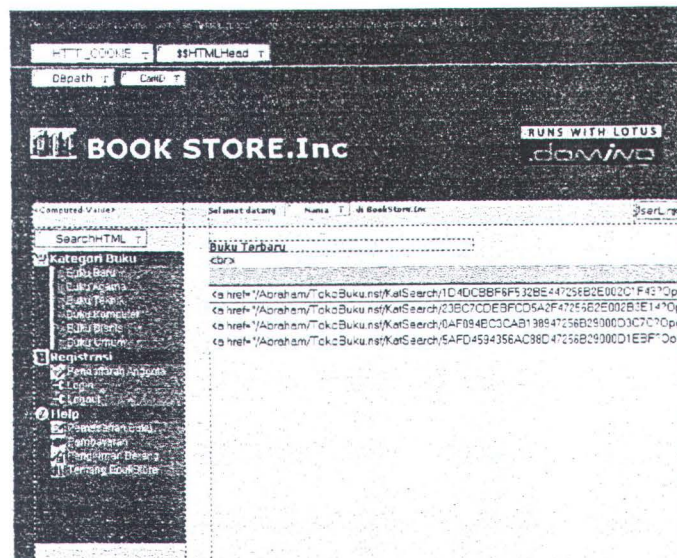


Gambar 4.5. Form Keranjang Belanja

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 dimiliki 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.



Gambar 4.7. Form Konfirmasi

4.2.9 Form Order

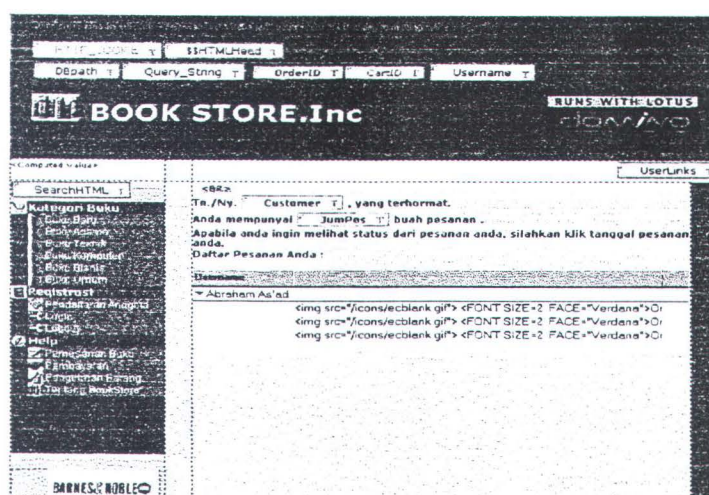
Form ini digunakan untuk menyimpan informasi pesanan(*order*) yang dilakukan oleh pelanggan. Gambar 4.8 dibawah ini adalah sebagian tampilan dari rancangan form Order.

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 dimiliki pelanggan. Gambar 4.9 dibawah ini adalah sebagian tampilan dari rancangan form OrderStatusConfirm.



Gambar 4.9. Form OrderStatusConfirm

BOOK STORE.Inc RUNS WITH LOTUS domino

Computed Value welcome UserLinks

SearchHTML

Kategori Buku

- Buku Baru
- Buku Ajaib
- Buku Teka-teki
- Buku Komputer
- Buku Bioskop
- Buku Ilmiah

Registrasi

- Peraturan Anggota
- Daftar
- Logout

Help

- Peraturan Buku
- Peraturan Bioskop
- Peraturan Anggota
- Peraturan Buku
- Tentang BookStore

Metode Pembayaran:

1. Transfer Bank

Untuk pembayaran melalui transfer bank, pembayaran bisa ditujukan kepada:

A/C: 122212121

Atas nama: BookStore.Inc / Abraham As'ad

Bank BCA Cabang Darmo - SURABAYA.

Kemudian kirimkan bukti pembayaran anda kepada kami dengan cara:

- Fax

- Email, dengan alamat: AbrahamAs.ad@sifodominio.edu

Sarung anda akan kami kirimkan setelah bukti pembayaran kami terima.

2. Cash On Delivery

Cash On Delivery ini hanya berlaku untuk pesanan dan Surabaya saja. Pembayaran secara tunai dilakukan di tempat saat pengambilan buku.

3. Kartu Kredit secara Online

Untuk pembelian kartu kredit, bisa...

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.

BOOK STORE.Inc RUNS WITH LOTUS domino

Computed Value welcome UserLinks

SearchHTML

Kategori Buku

- Buku Baru
- Buku Ajaib
- Buku Teka-teki
- Buku Komputer
- Buku Bioskop
- Buku Ilmiah

Registrasi

- Peraturan Anggota
- Daftar
- Logout

Help

- Peraturan Buku
- Peraturan Bioskop
- Peraturan Anggota
- Peraturan Buku
- Tentang BookStore

Metode Pemesanan Buku secara online:

1. Daftarkan diri anda.

Untuk bisa memproses pesanan Anda secara online, BookStore.Inc memerlukan Anda untuk mempunyai account di BookStore.Inc yang dapat diperoleh setelah anda mendaftarkan. Anda akan memerlukan username dan sebuah password untuk bisa mengakses Account Anda. Anda juga bisa memeriksa status order Anda secara online dengan menggunakan order ID yang akan diberikan kepada anda setelah anda melakukan transaksi.

2. Pilih buku yang akan dibeli.

Pencarian buku adalah sangat mudah. Anda bisa menggunakan fungsi 'Cari' untuk mencari buku yang Anda inginkan atau bisa menggunakan menu 'Kategori Buku' di samping untuk melihat buku yang ada pada satu kategori tertentu.

3. Masukkan belanjaan Anda ke Shopping Cart

Dengan mengklik tombol 'Add To Cart', buku tersebut akan otomatis masuk ke dalam keranjang belanja Anda. 'Keranjang Belanja' ini akan menyimpan informasi tentang buku yang ingin Anda beli. Untuk membatalkan sebuah buku, Anda bisa mengklik pilihan dibawah header 'hapus' kemudian klik tombol 'Update Cart'. Demikian juga untuk mengubah jumlah buku cukup isikan jumlah yang anda inginkan di field 'Jumlah' kemudian klik 'Update Cart'.

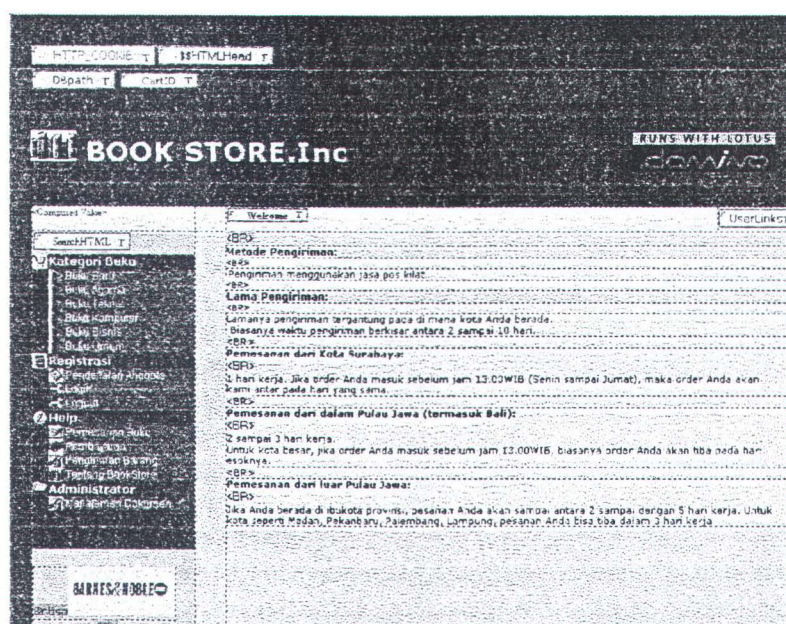
4. Pilih metode pembayaran Anda

Anda bisa membayar dengan menggunakan transfer bank dan kartu kredit. Untuk order Surabaya Anda mempunyai alternatif pembayaran lain, yaitu Cash On Delivery, bayar pada saat pesanan Anda datang.

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

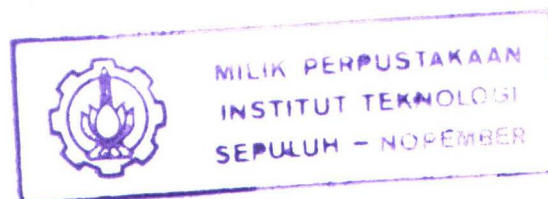
4.2.15 Form RegChoice

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.



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 mengisi 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.

- **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 :

```

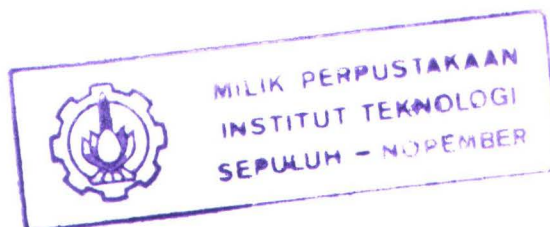
Sub Initialize
Dim s As New NotesSession
Dim db As NotesDatabase
Dim doc As NotesDocument, oiDoc As NotesDocument, cDoc As
NotesDocument
Set db = s.CurrentDatabase

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

```




```

'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 diisi 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 mengisi nomor kartu dan password dan menekan tombol submit pada form Order maka agent ini akan dijalankan. Agent ini akan mengecek nomor kartu dan password pada database dengan nomor kartu dan password yang diisi 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 = docKartu.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

```




```
'Set doc equal to submitted web form
Set doc = s.DOCUMENTCONTEXT
```

```
'initialize 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("(SLDAPCN)")
```

```
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

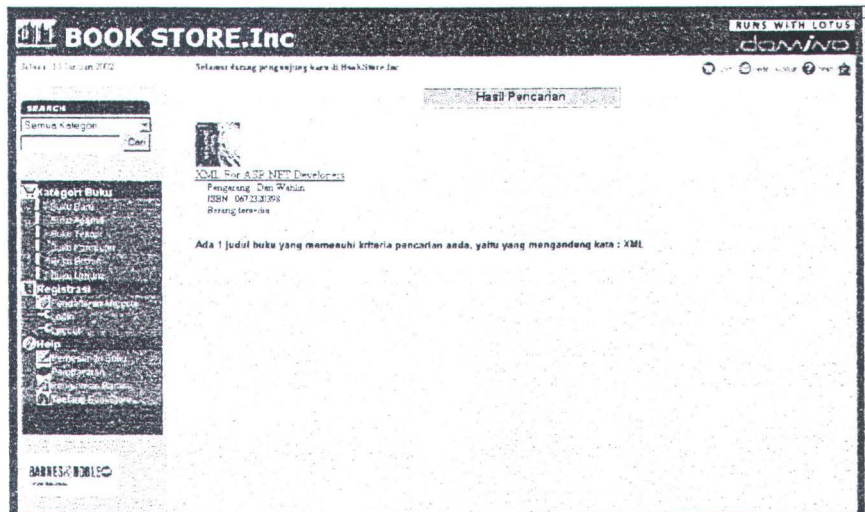
Perangkat Lunak : Lotus Domino Server 5, Lotus Domino
Administrator 5

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 hampir semua halaman BookStore.Inc.

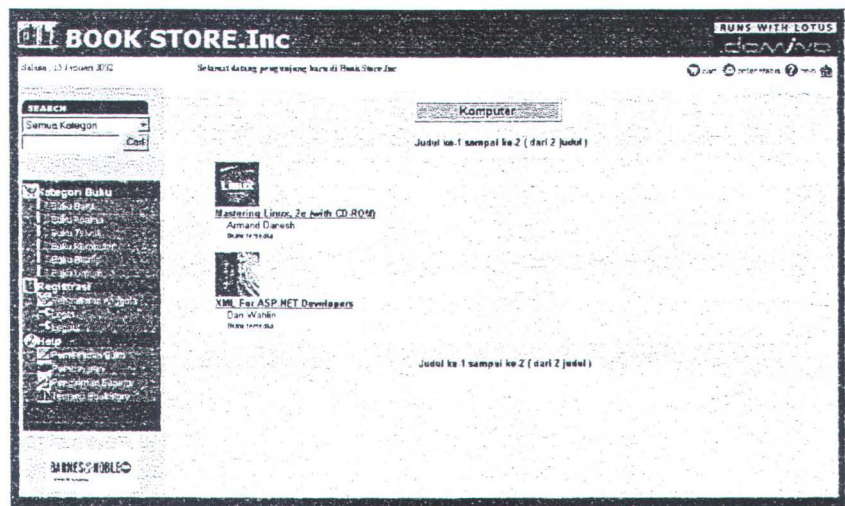
Gambar 6.1 berikut ini adalah tampilan halaman utama dari BookStore.Inc.



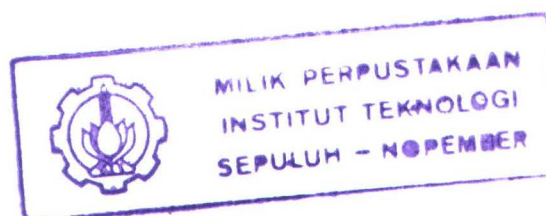
Gambar 5.2. Tampilan Hasil Pencarian judul “XML”

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.

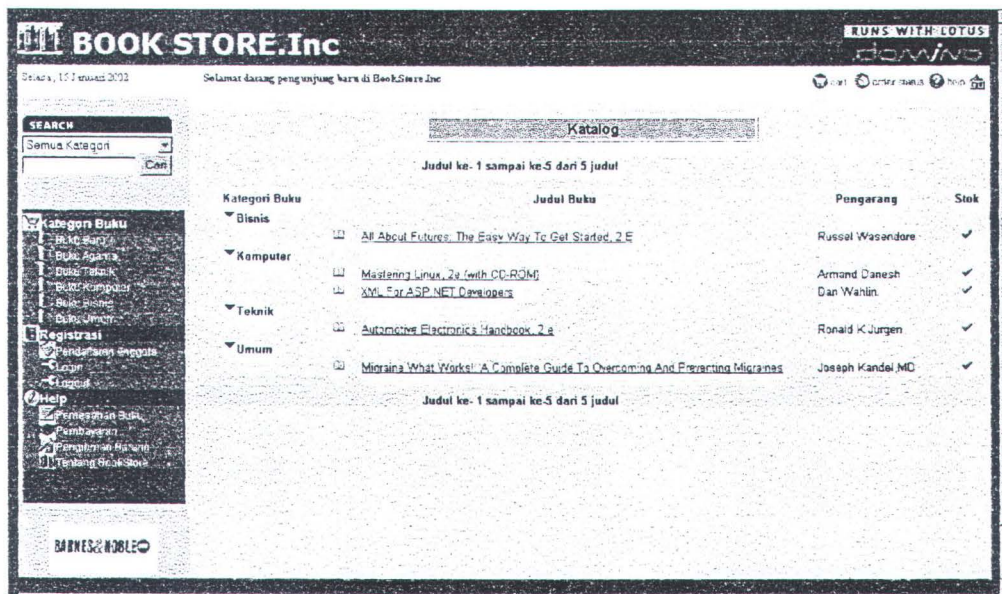
Berikut ini adalah tampilan yang muncul setelah penulis meng-klik kategori Komputer pada navigator sebelah kiri.



Gambar 5.3 Tampilan Katalog Kategori Komputer



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.

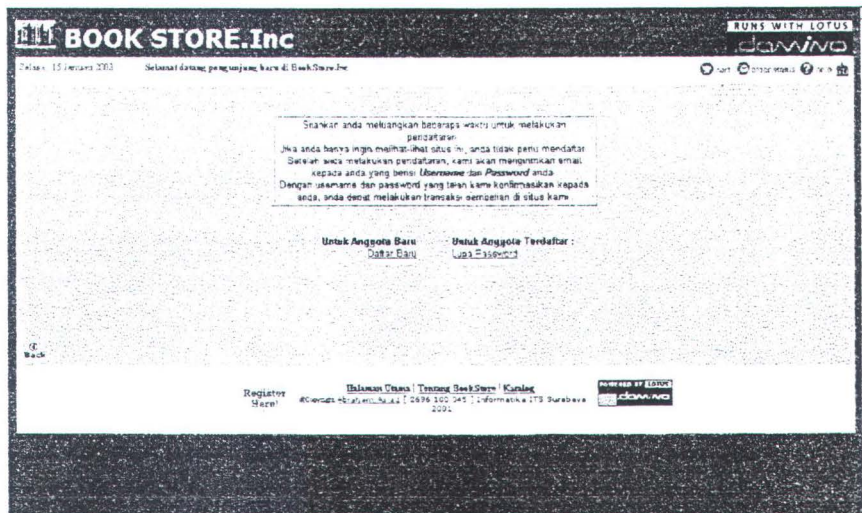


Gambar 5.4. Halaman Katalog Buku

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 muncul setelah penulis meng-klik menu Daftar Baru.

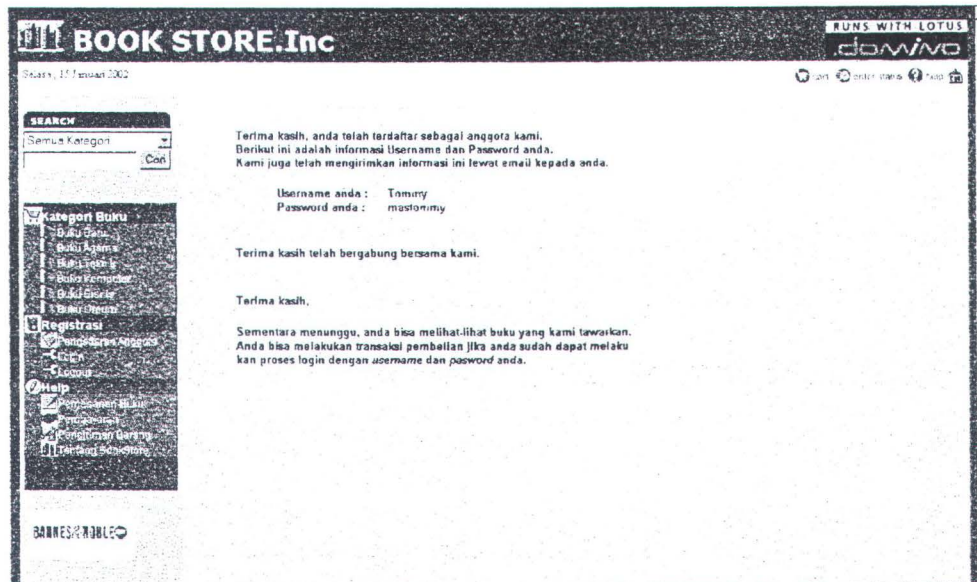
 The screenshot shows a registration form titled 'Selamat datang di situs ini. Yang harus anda lakukan adalah mengisi formulir pendaftaran ini. Setelah anda mengisi formulir pendaftaran ini, kami akan mengirimkan email kepada anda yang berisi Username dan Password anda. Dengan Username dan Password yang telah kami kirimkan kepada anda, anda dapat melakukan transaksi pembelian di situs kami.' The form is divided into two sections: 'Security' and 'Informasi untuk Pengiriman Barang'. The 'Security' section includes fields for 'User Name', 'Email (alamat email disarankan)', 'Password (min. 6 karakter)', and 'Konfirmasi Password'. The 'Informasi untuk Pengiriman Barang' section includes fields for 'Nama', 'Nama Perusahaan (optional)', 'Penerimaan (optional)', 'Alamat', 'Kota', 'Provinsi' (with a dropdown menu showing 'Jawa Timur'), 'Negara (optional)' (with a dropdown menu showing 'Indonesia'), 'Kode Pos (optional)', 'Telepon (optional)', and 'Email'. At the bottom, there are 'Submit' and 'Reset' buttons, and a 'Back' link.

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".



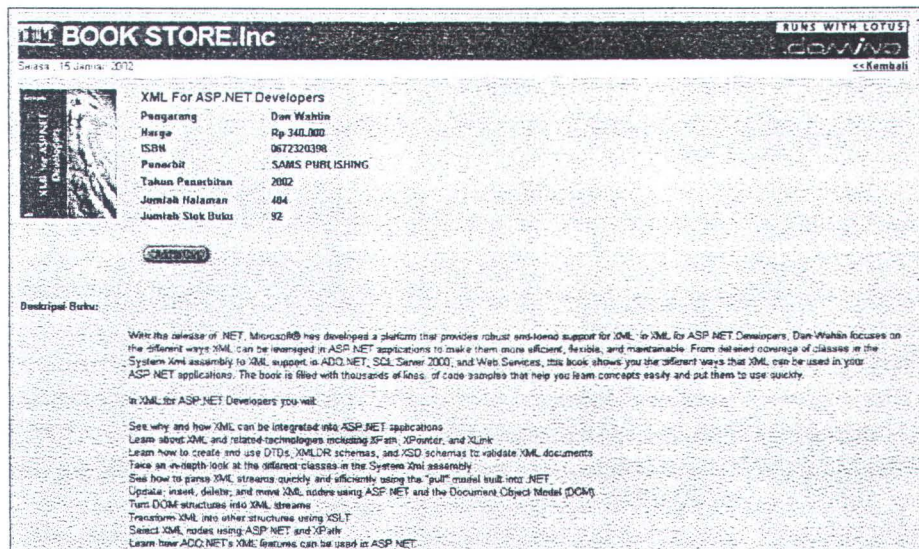
Gambar 5.7. Tampilan Halaman Konfirmasi Pendaftaran Berhasil

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.

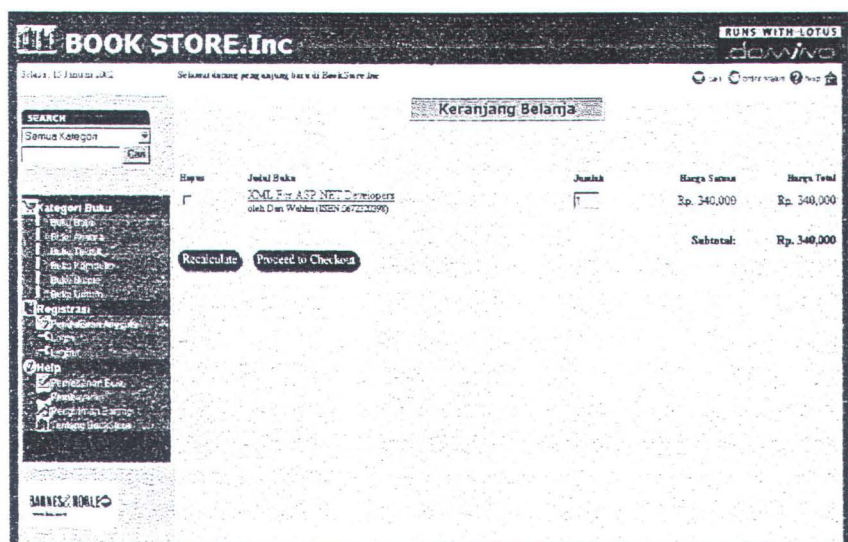
Untuk memasukkan buku kedalam keranjang belanja pengguna harus memilih buku, kemudian menekan tombol AddtoCart yang terdapat pada deskripsi buku. Khusus untuk buku baru yang ada di halaman utama, pengguna dapat langsung menekan tombol AddtoCart yang ada di bawah deskripsi singkat masing-masing buku.

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”

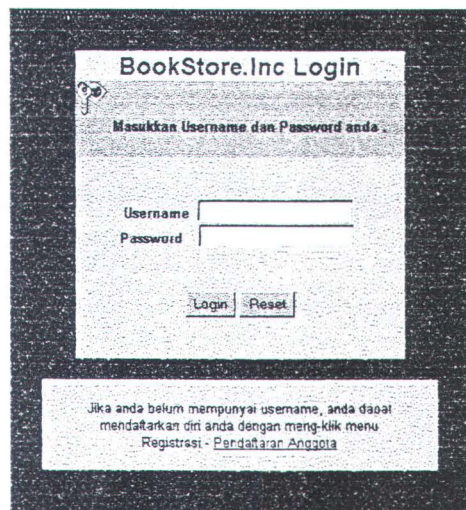
Setelah penulis menekan tombol AddtoCart maka halaman keranjang belanja akan ditampilkan. Berikut ini adalah gambar halaman keranjang belanja penulis.



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.



BookStore.Inc Login

Masukkan Username dan Password anda.

Username

Password

Login Reset

Jika anda belum mempunyai username, anda dapat mendaftarkan diri anda dengan meng-klik menu Registrasi - Pendaftaran Anggota

Gambar 5.10 Form Login

Pada uji coba ini penulis mengisi username “tommy” dan password “mastommy” yang telah didaftarkan pada proses pendaftaran anggota diatas. Setelah penulis menekan tombol Login maka kemudian muncul tampilan form Order. Apabila pengguna salah dalam mengisi 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.

BOOK STORE Inc RUNS WITH LOTUS DOMINO

Halaman : 15 / Senin, 04/12/2002

Konfirmasi Belanja Anda

User Profile:

Nama : Tommy J. Pisa
 Alamat : Jl. Moleo 3
 Kota : Jakarta
 Propinsi : Dki Jakarta
 Kode Pos : 56008
 Telepon : 821-32221376
 Email : abrahamaad@sisfodomino.edu

Belanja Anda:

Judul	Pengarang	Jumlah	Harga Satuan	Harga Total
XML For ASP.NET Developers	Daso Wahlin	1	Rp 340.000	Rp 340.000
Total :				Rp 340.000

Pilih Cara Pembayaran:

☒ Kartu Kredit
☐ Transfer Bank

No. Kartu Kredit :
 Password :

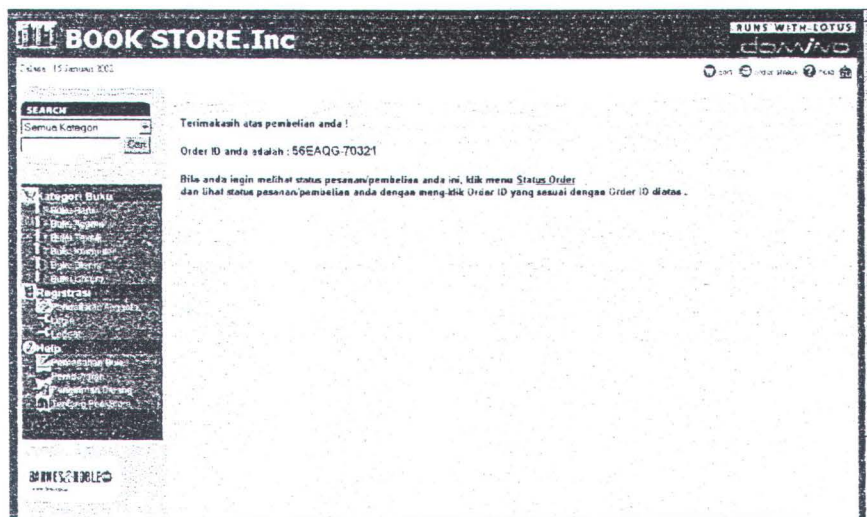
Gambar 5.11. Halaman Order

Pada form diatas informasi pelanggan seperti nama dan alamat ditampilkan secara otomatis. Demikian juga dengan buku yang akan dibeli beserta jumlah totalnya. Informasi pelanggan diperoleh dengan cara look up ke database names.nsf yang digunakan untuk menyimpan data pelanggan.

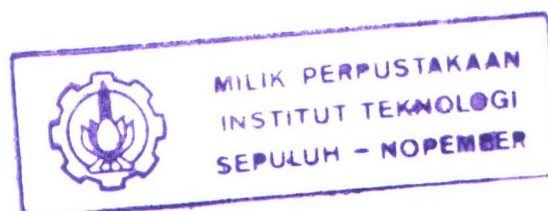
Proses selanjutnya adalah validasi pembayaran. Aplikasi ini memiliki dua macam cara pembayaran, yaitu dengan cara transfer ke rekening bank dan dengan menggunakan kartu kredit. Untuk validasi kartu kredit, pada aplikasi ini dibuat

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.



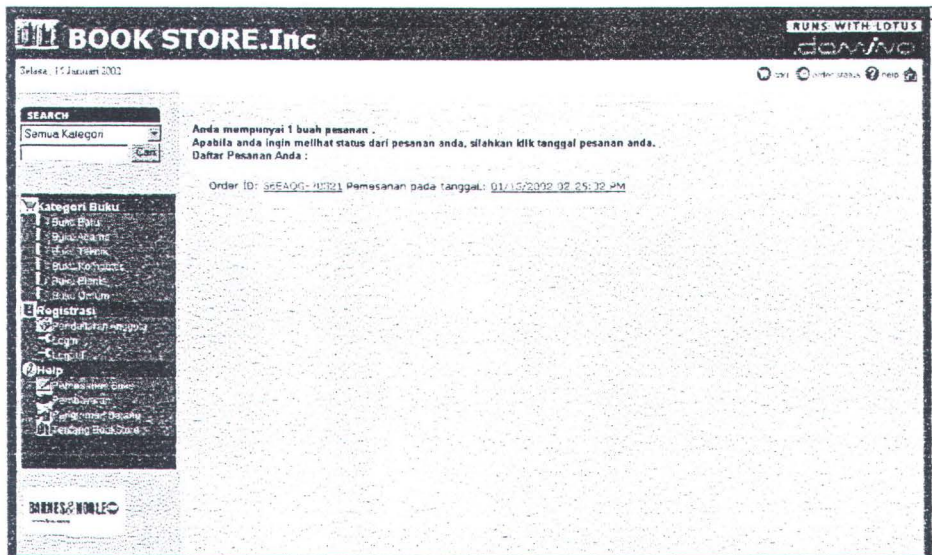
Gambar 5.12. Halaman Konfirmasi Pembelian Berhasil



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.

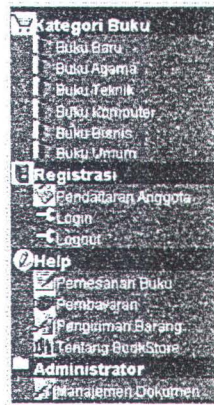
Apabila pelanggan salah dalam mengisi nomor kartu kredit dan password, maka pelanggan tidak dapat meneruskan pembelian yang berarti jumlah buku yang telah dipilih akan tetap seperti semula. Apabila pelanggan tidak memiliki kartu kredit, pelanggan dapat memilih alternatif kedua, yaitu dengan transfer 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 pesannya menjadi "Dibatalkan". Status Order sendiri memiliki empat nilai, yaitu MenungguPembayaran, 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.



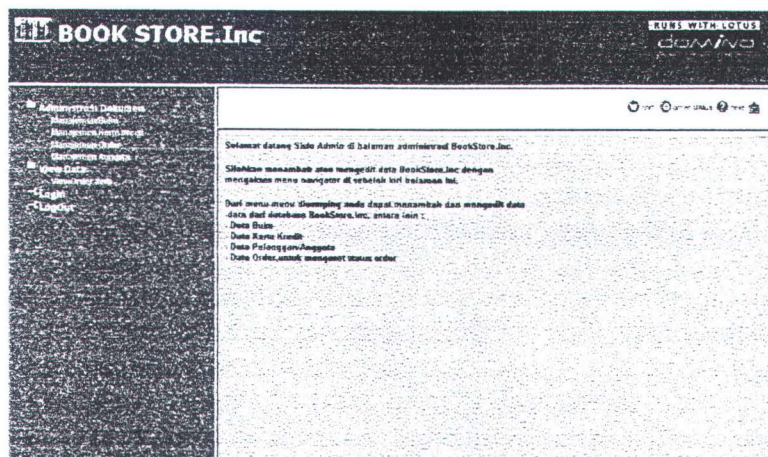
Gambar 5.13. Halaman Daftar Order Pengguna

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 5.15. Menu Manajemen Dokumen pada navigator kiri

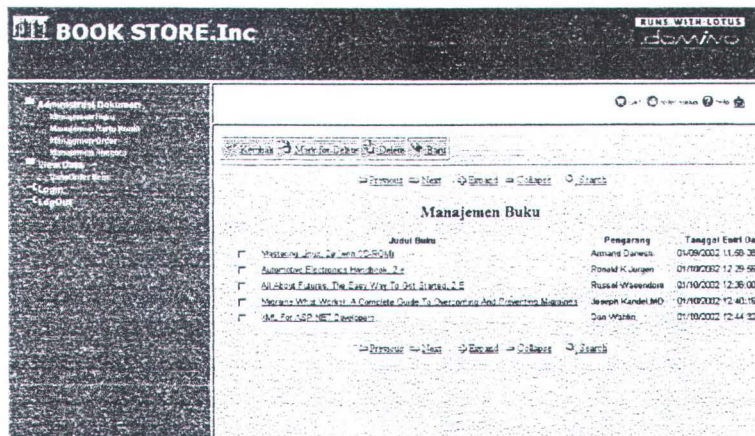
Setelah penulis meng-klik menu Manajemen Dokumen maka muncul halaman web baru yang berisi halaman administrator. Pada halaman ini ada menu Administrasi Dokumen yang terdiri dari menu manajemen buku, manajemen kartu kredit, manajemen order, dan manajemen pelanggan, dan juga menu View Order Item yang akan 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

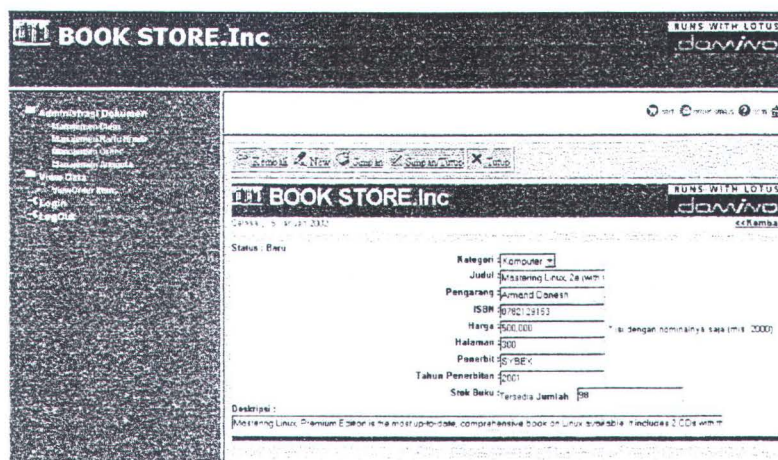
5.2.2.1 Manajemen Buku

Berikut ini adalah tampilan utama manajemen buku. Halaman ini berisi semua buku yang ada di database BookStore.Inc



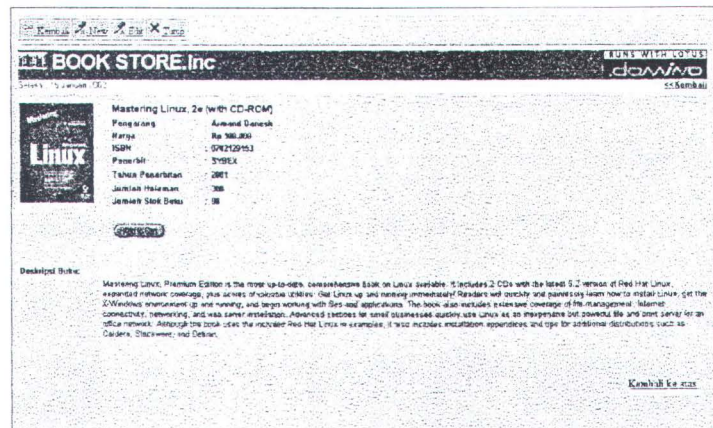
Gambar 5.17. Halaman Manajemen Buku

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

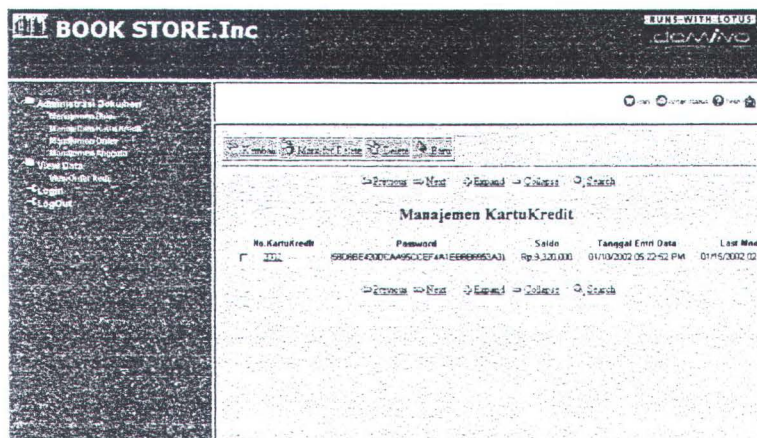
Perubahan yang terjadi dapat dilihat pada deskripsi buku Mastering Linux,2 e, seperti pada gambar dibawah ini.



Gambar 5.19. Halaman Deskripsi buku Mastering Linux,2e

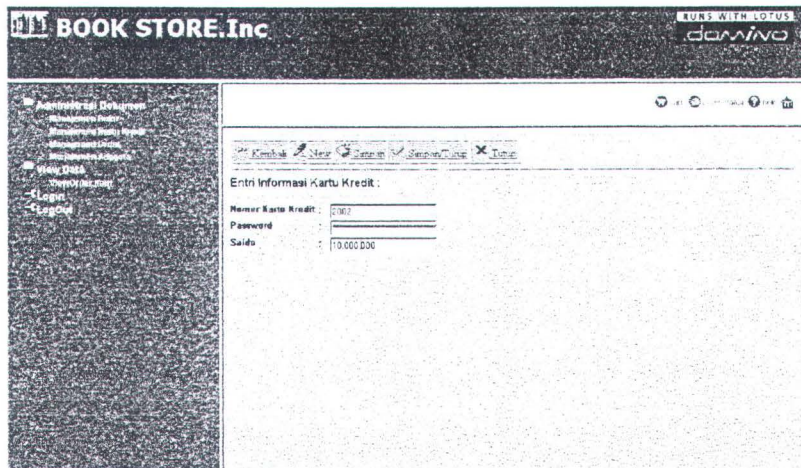
5.2.2.2 Manajemen Kartu Kredit

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 mengubah 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.

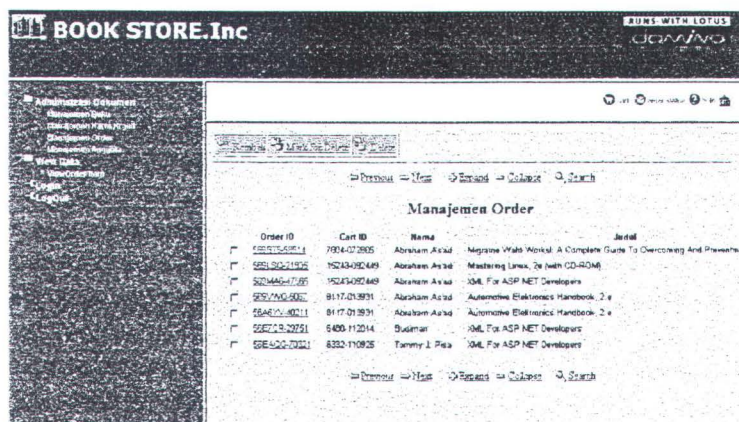


Gambar 5.21. Halaman Edit Kartu Kredit

Perubahan yang terjadi dapat dilihat pada halaman utama kartu kredit diatas.

5.2.2.3 Manajemen Order

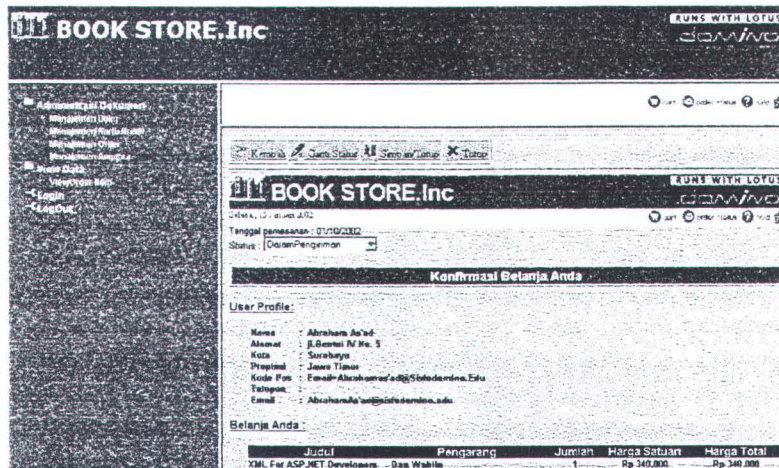
Berikut ini adalah tampilan utama manajemen order. Halaman ini berisi semua order yang telah dipesan atau dibeli oleh pelanggan.



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.

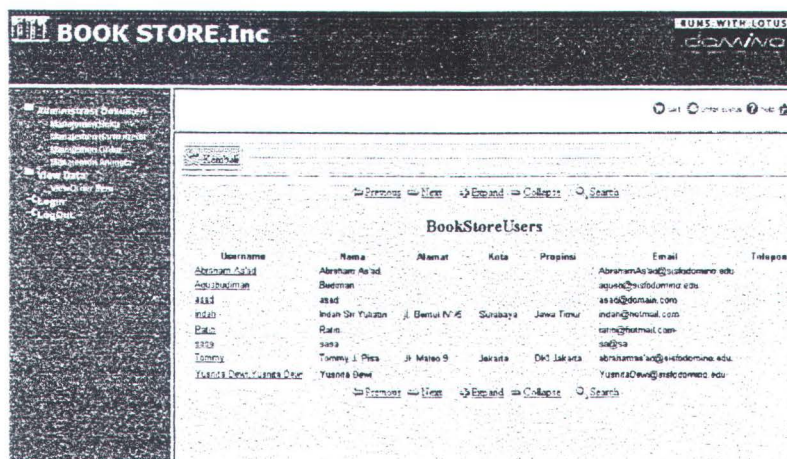


Gambar 5.23. Halaman Edit Order

Untuk melihat hasil perubahan yang terjadi dapat dilihat pada halaman utama manajemen order diatas.

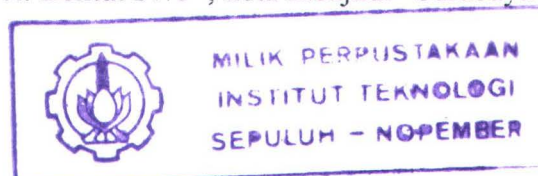
5.2.2.4 Manajemen Pelanggan

Berikut ini adalah tampilan utama manajemen pelanggan. Halaman ini berisi semua pelanggan yang terdaftar di BookStore.Inc.



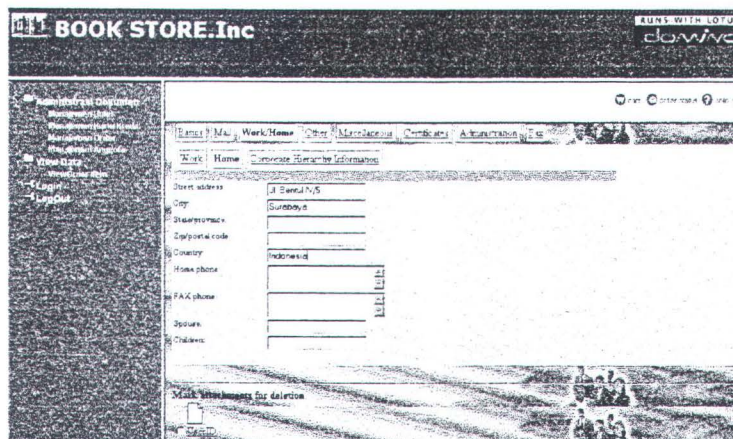
Gambar 5.24. Halaman Utama Manajemen Pelanggan

Pada ujicoba ini penulis mengubah alamat dari username Abraham As'ad yang sebelumnya kosong menjadi "Jl. Bentul IV/5", kota menjadi "Surabaya" dan



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 Administrator 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

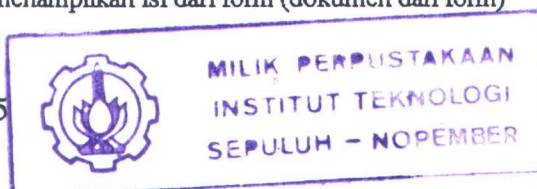
Bab ini menjelaskan kesimpulan dan kemungkinan pengembangan lebih lanjut dari aplikasi toko buku online BookStore.Inc.

6.1 Kesimpulan

Kesimpulan yang didapat dari tugas akhir ini adalah :

- 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 mempunyai Lotus untuk menampilkan isi dari form (dokumen dari form)



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 aplikasi 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 Sistem Lotus 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.

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



Dr. Ir. Arif Djunaidy, M.Sc.

NIP. 131 633 403

Dosen Pembimbing II



Rully Soelaiman, S.Kom

NIP. 132 085 802

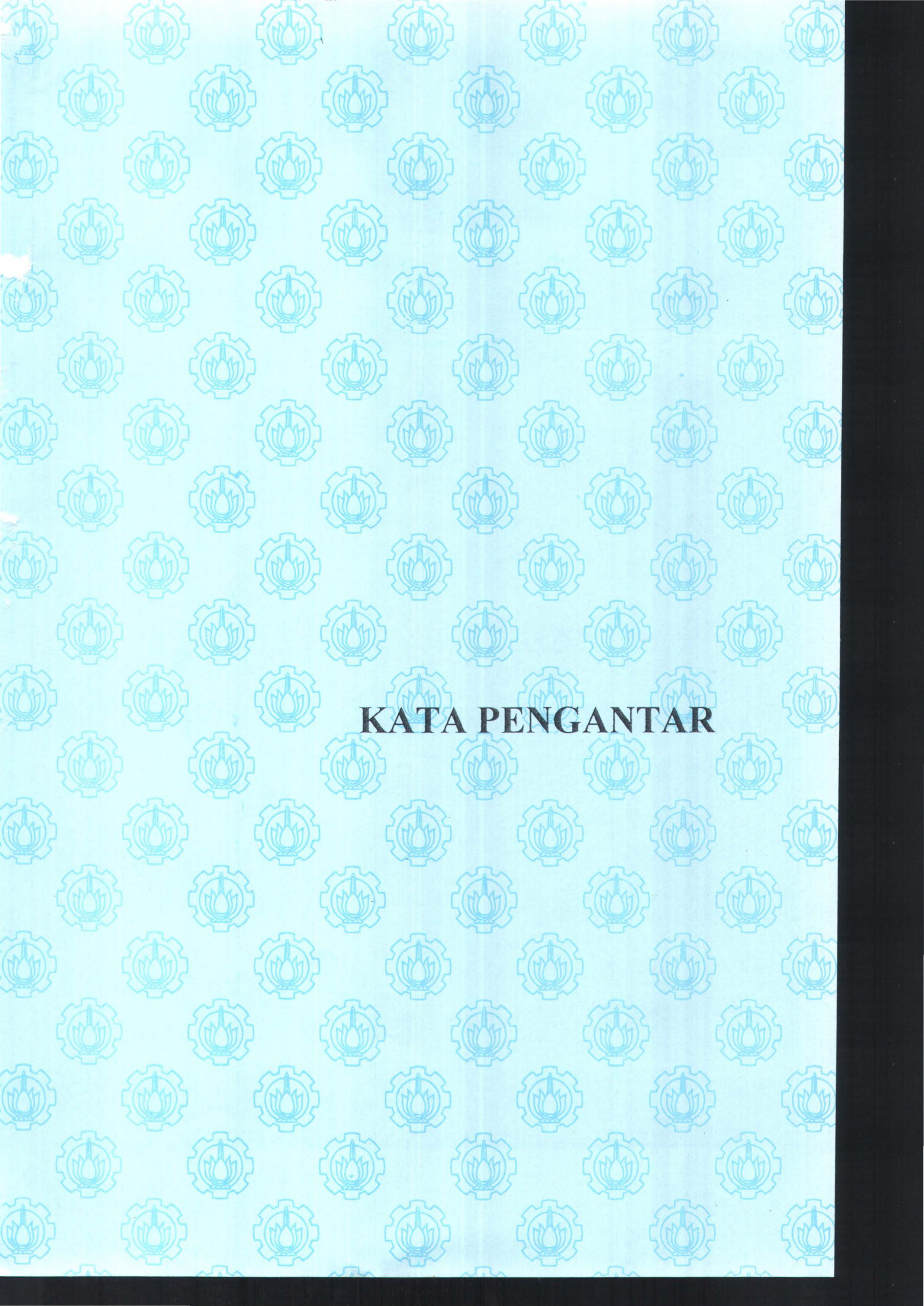
**Surabaya
Agustus, 2001**

Bismillahirrahmanirrahim

"Hai orang-orang yang beriman, bertakwalah kepada Allah sebenarnya 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)"

*Supersembahkan Untuk
Ayahanda, Ibunda dan Suami Tercinta*



KATA PENGANTAR

KATA PENGANTAR

Alhamdulillah 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 keluhan-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. Amini.

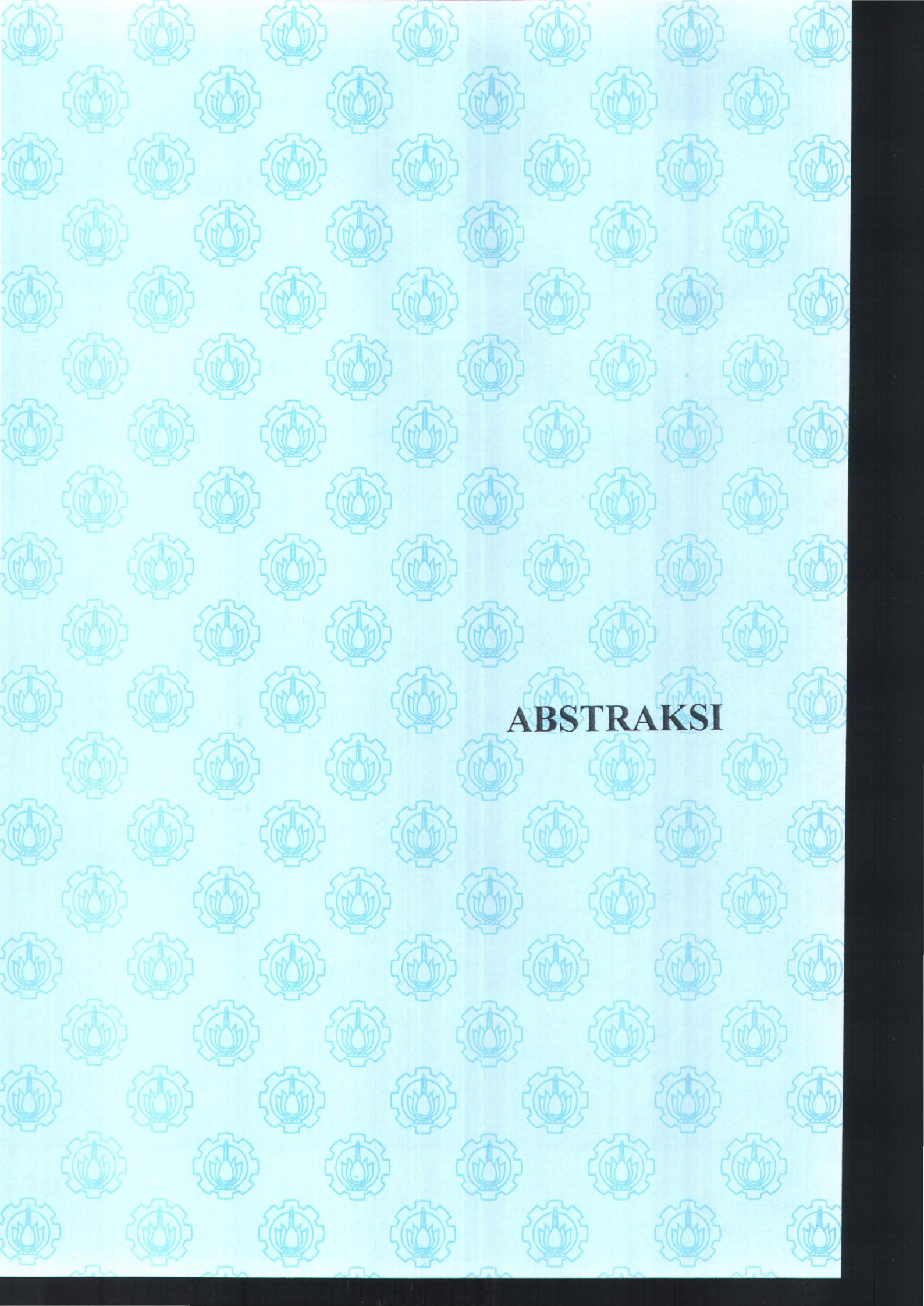
- Naim dan Yuli atas bantuan dan motivasinya. Cepetan Tugas Akhirnya dikerjakan. Semoga cepet nyusul (lulus dan mendapat jodoh).
- Anak-anakku Lyha dan Ita yang setia mendengarkan keluh-kesahku dan selalu memberi motivasi, semoga cepet lulus.
- Mbah Imam atas ilmu yang diberikan, serta bantuan dan motivasinya.
- 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



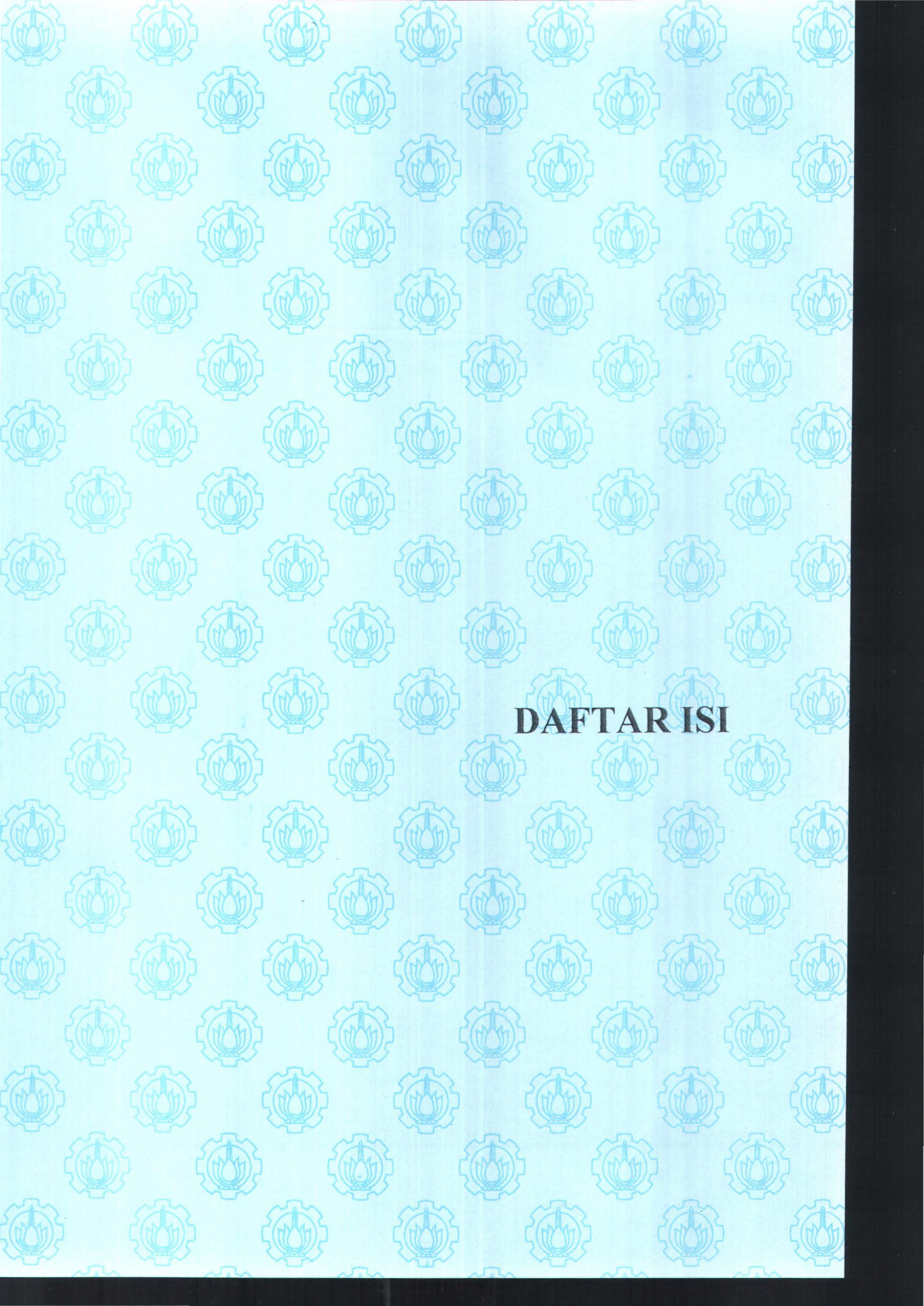
ABSTRAKSI

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.



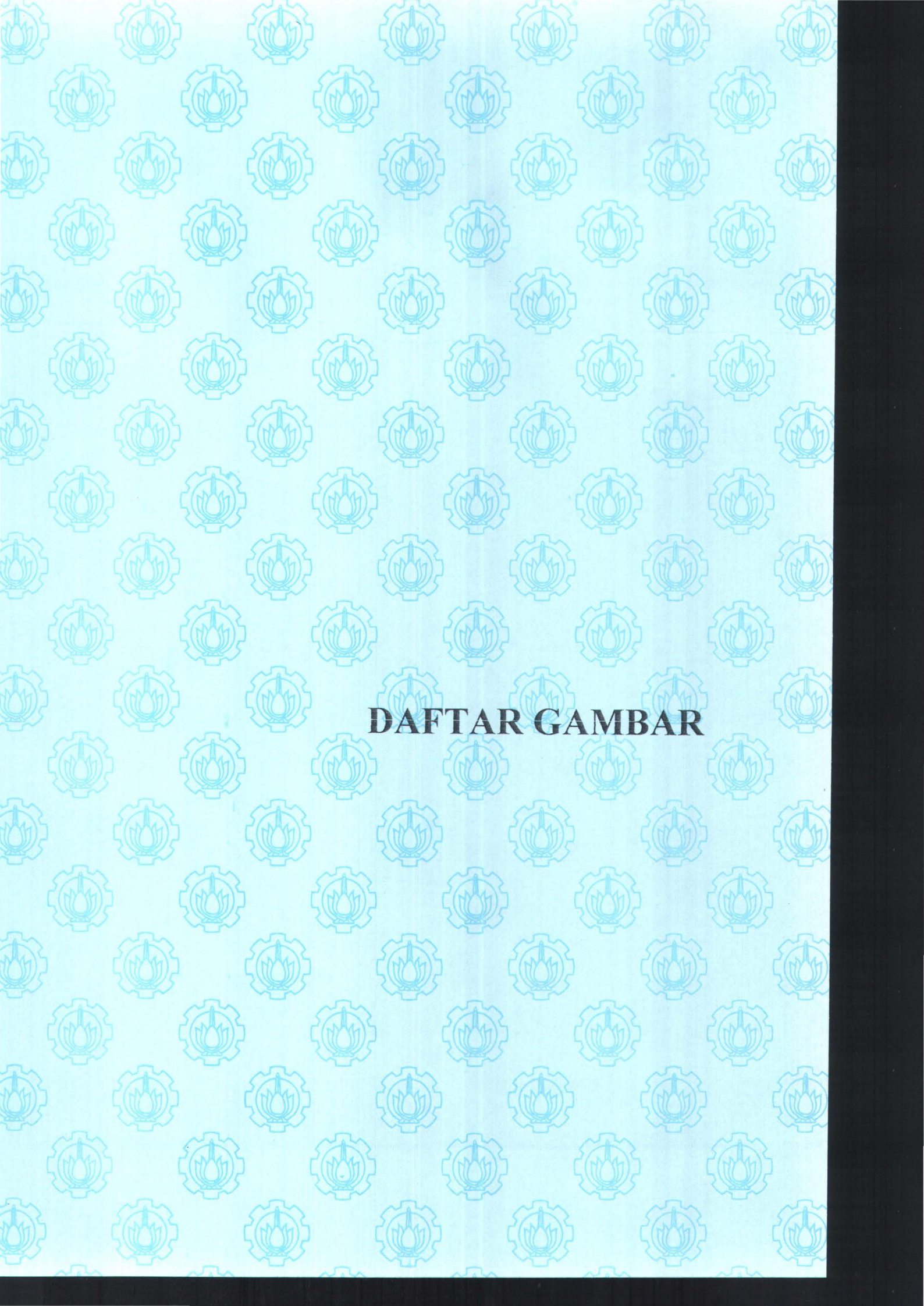
DAFTAR ISI

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

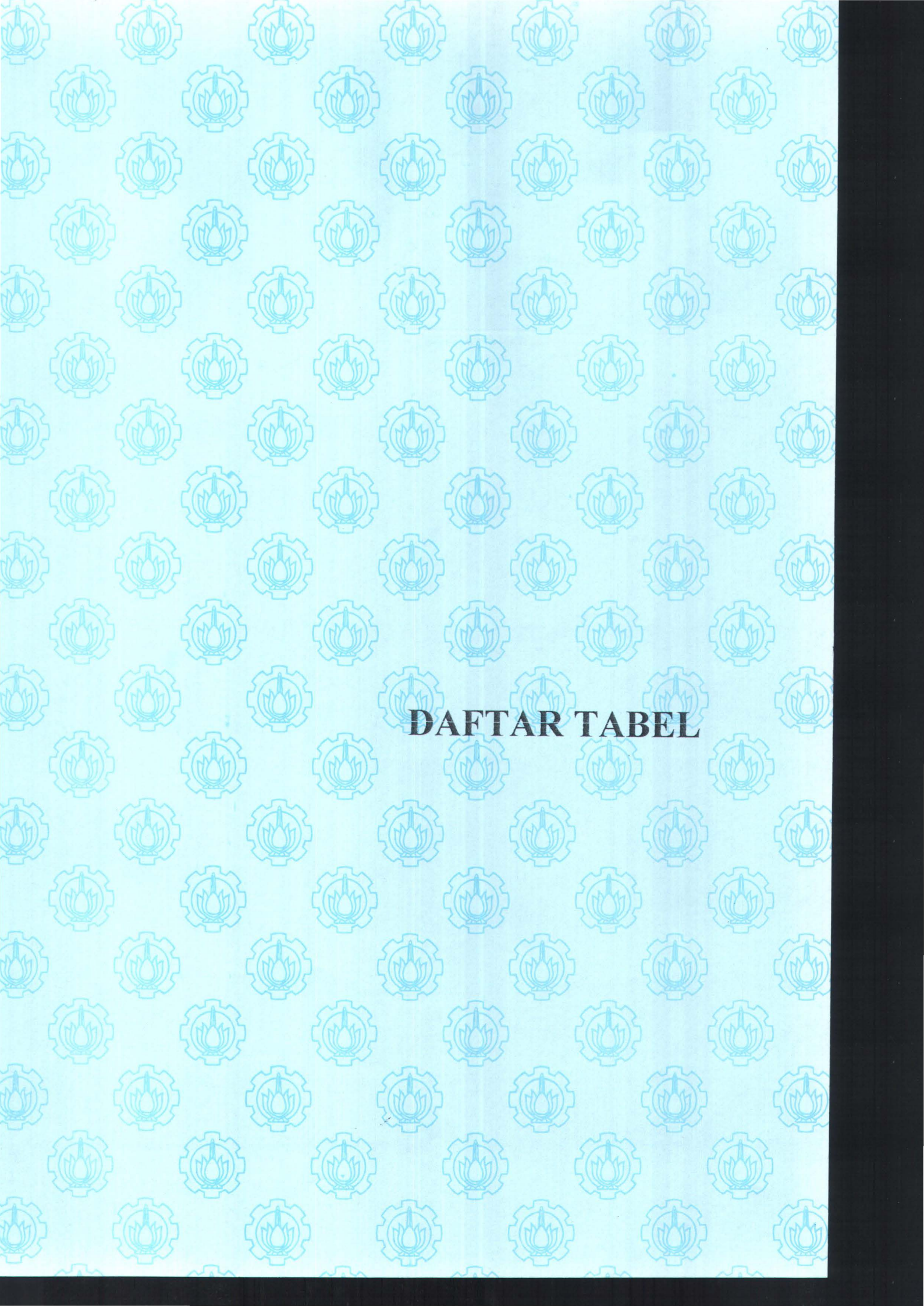
DAFTAR PUSTAKA	54
LAMPIRAN A : DATA PELATIHAN	A-1
LAMPIRAN B : PETUNJUK PEMAKAIAN PERANGKAT LUNAK	B-1



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

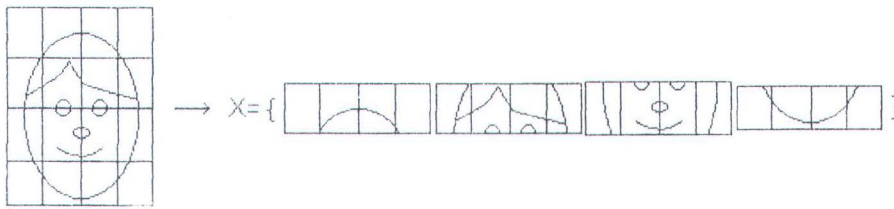
Dalam bab ini akan dibahas konsep-konsep dan teori-teori dasar tentang pengenalan wajah dengan menggunakan metode Fisherface yang berhubungan erat dengan pembuatan sistem perangkat lunak dan implementasi program. Dalam bab ini juga dibahas metode serta algoritma dari metode Fisherface

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 $w \cdot h$. 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

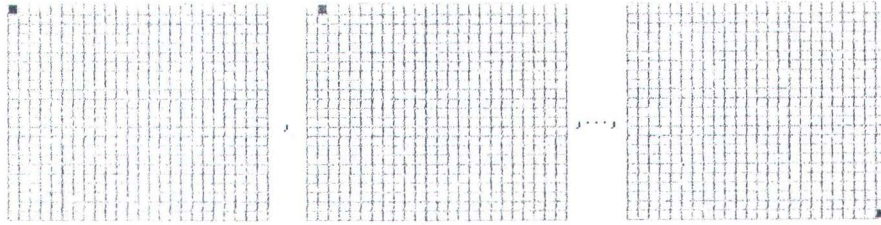
Vektor wajah yang telah dideskripsikan sebelumnya merupakan bagian dari sebuah ruang. Ruang ini adalah sebuah ruang lingkup gambar (*image space*), yaitu ruang dari semua gambar keseluruhan yang mempunyai dimensi $w \times h$ piksel. Basis dari ruang lingkup gambar dikomposisikan oleh vektor-vektor berikut, 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 $w \times h$. 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 $N \times N$ 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 $(N \times N) \times 1$.

Misalkan setiap gambar diberi notasi $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_K$ yang merupakan vektor kolom $N^2 \times 1$. Bila sebuah gambar dalam wajah pelatihan dinotasikan dalam bentuk matriks, maka akan diperoleh matriks yang berordo $N^2 \times K$.

$\Gamma_n = (\Gamma_1 \ \Gamma_2 \ \Gamma_3 \ \dots \ \Gamma_K)^T$, dimana T adalah matriks transpose. Sedang isi dari tiap vektor kolom adalah

$$\Gamma_n = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \dots & \gamma_{1,K} \\ \gamma_{2,1} & \gamma_{2,2} & \dots & \gamma_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N^*N,1} & \gamma_{N^*N,2} & \dots & \gamma_{N^*N,K} \end{bmatrix}$$

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 $N^2 \times K$ 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 $N^2 \times N^2$ dan merupakan sebuah pengukuran yang tidak hanya pada sebuah variansi, tetapi *kovariansi* 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 $A = (\Phi_1 \ \Phi_2 \ \dots \ \Phi_K)$ adalah matriks wajah ternormalisasi, maka matriks kovariansi di atas dapat ditulis menjadi :

$C_x = AA^T$ adalah matriks *outer product* berordo $N^2 \times N^2$, atau

$L = A^T A$ adalah matriks *inner product* berordo $K \times K$, $K \ll N^2$.

- 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 $n \times n$ 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.



2.3.1 Definisi Nilai Karakteristik dan Vektor Karakteristik

Definisi 1 : Misalkan A adalah sebuah matriks $n \times n$. Sebuah matriks bukan nol, P , yang berukuran $n \times 1$ 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 :

$$AP = \lambda P \quad \text{ekivalen dengan}$$

$$\lambda IP = AP \quad \text{atau} \quad (\lambda I - A)P = 0.$$

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 $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$. 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 $n \times n$ dikatakan *similar* bila $B = P^{-1}AP$.

Bukti bahwa nilai karakteristik dari matriks diagonal B adalah nilai karakteristik dari A dapat diilustrasikan sebagai berikut :

$$\text{Bukti : } B = P^{-1}AP = \begin{bmatrix} \lambda_1 & 0 & 0 & \Lambda & 0 \\ 0 & \lambda_2 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & & M \\ M & M & & 0 & M \\ 0 & 0 & \Lambda & \Lambda & \lambda_n \end{bmatrix}$$

Misalkan $P_1, P_2, \dots, P_n$ adalah kolom dari P, atau $P = [P_1 \ P_2 \ \dots \ P_n]$ adalah vektor karakteristik yang bersesuaian dengan nilai karakteristik, maka

$$A[P_1 P_2 P_3 \dots P_n] = [P_1 P_2 P_3 \dots P_n] \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Dengan menulis kembali dalam istilah kolom-kolom, maka

$$[AP_1 \ AP_2 \ AP_3 \ \dots \ AP_n] = [\lambda_1 P_1 \ \lambda_2 P_2 \ \lambda_3 P_3 \ \dots \ \lambda_n P_n]$$

Jadi diperoleh $AP_i = \lambda_i P_i$, $i = 1, 2, 3, \dots, n$.

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 $n \times n$ adalah matriks *simetris*, maka matriks $A^T = A$.

Dari uraian di atas, maka terbukti bahwa P adalah matriks yang dapat didiagonalkan secara ortogonal. Oleh karena P adalah *matriks ortogonal*, maka matriks $P^{-1} = P^T$. Akibatnya, matriks diagonal B pada uraian sebelumnya yang semula ditulis :

$$B = P^{-1}AP \text{ ekuivalen dengan } B = P^TAP$$

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 R_{pq} , yang mana elemen-elemen matriksnya adalah :

$$\begin{aligned} r_{pp} &= \cos \theta & r_{pq} &= \sin \theta \\ r_{qp} &= \sin \theta & r_{qq} &= -\cos \theta \end{aligned}$$

$$R_{pq} = \begin{bmatrix} 1 & & & & \\ & \Lambda & & & \\ & & \cos \theta & \Lambda & \sin \theta \\ & & & \Lambda & \\ & & \sin \theta & \Lambda & -\cos \theta \\ & & & & & \Lambda \\ & & & & & & 1 \end{bmatrix}$$

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 specific, 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:

$$W_{opt} = \arg \max_w \frac{|WS_B W^T|}{|WS_W W^T|}$$

$$= [w_1; w_2; \dots; w_m]$$

dimana $[w_1; w_2; \dots; w_m]$ 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 $i : 1 \dots m$ dan $d_1 > d_2 > \dots > d_m$.

Jika x_i , $i = 1 \dots N$, adalah vektor citra dimensi- n dan masing-masing vektor citra merupakan anggota salah satu dari C kelas citra wajah $\{X_1, X_2, \dots, X_C\}$ 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 , $i = 1 \dots C$.

Suatu citra wajah dengan lebar dan tinggi masing-masing l dan t piksel memiliki jumlah piksel sebanyak $l \times t$. 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 $i = 1 \dots N$ (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 $j = 1 \dots C$. Untuk setiap kelas wajah X_j terdapat N_j buah citra wajah, dimana $j = 1 \dots C$. Dengan demikian jumlah semua citra wajah adalah $N_1 + N_2 + \dots + N_C = N$.

Dari vektor citra wajah di atas dapat dibentuk suatu vektor baris citra wajah, yaitu: $x_i = [a_{i1} \ a_{i2} \ \dots \ a_{in}]$ $(1 * n)$

Dengan demikian vektor citra dikatakan berada dalam ruang citra dimensi- n , dimana $i = 1 \dots N$. Selanjutnya adalah membentuk matriks input berdimensi $N * n$ yang berisi kumpulan vektor baris citra yang akan digunakan dalam pelatihan dan pengujian.

$$\text{input} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad (N * n)$$

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

$$X_i = \frac{X_i}{\sqrt{\sum_{i=1}^n X_i^2}}$$

Dimana n adalah dimensi dari ruang citra.

2. Mencari rata-rata vektor citra

$$\mu_{PCA} = \frac{1}{N} \sum_{i=1}^N X_i \quad (1*n)$$

Dimana N adalah banyaknya data wajah yang akan mengalami pelatihan dan n adalah dimensi dari ruang citra.

3. Untuk $i = 1 \dots N$, mencari selisih vektor citra dengan rata-rata vektor citra,

μ_{PCA}

$$\Phi_i = X_i - \mu_{PCA} \quad (1*n)$$

Menyimpan Φ dalam matriks A, sehingga

$$A = [\Phi_1; \Phi_2; \dots; \Phi_N] \quad (N*n)$$

4. Mencari matriks total scatter, S_T

$$S_T = A A^T \quad (N*n)*(n*N) = (N*N)$$

5. Mencari vektor eigen dan nilai eigen dari matriks S_T , kemudian mengurutkan berdasarkan nilai eigen, dari yang terbesar.

$$[v, d] = \text{eig}(S_T)$$

$$\text{Nilai eigen dari } S_T : d = [d_1 \ d_2 \ \dots \ d_n] \quad (1*N)$$

Dimana $d_1 > d_2 > \dots > d_n$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$V = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix} \quad (N \times N)$$

6. Mereduksi vektor eigen dari S_T sesuai dengan nilai $(N-C)$

$$V' = X_i^T V D$$

$$(n \times N) \times (N \times N) \times (N \times (N-C)) = (n \times (N-C))$$

dimana :

- X_i adalah matriks wajah yang telah dinormalisasi dengan ordo $(N \times n)$
- V adalah matriks yang berisi vektor karakteristik dengan ordo $(N \times N)$
- D adalah matriks diagonal yang elemen diagonalnya adalah $(\lambda_i)^{-1/2}$ dengan ordo $(N \times (N-C))$

Matriks transformasi W_{PCA} adalah :

$$W_{PCA} = V'^T \quad ((N-C) \times n)$$

Transformasi PCA

7. Transformasi vektor citra dimensi- n menjadi vektor input dimensi- $(N-C)$, untuk input masukan metode FLD

$$\text{InputFLD} = A W_{PCA}^T \quad (N*n)*(n*(N-C))=(N*(N-C))$$

Atau untuk $i = 1 \dots N$:

$$X_i = \Phi_i W_{PCA}^T \quad (1*n)*(n*(N-C)) = (1*(N-C))$$

Jadi :

$$\text{input} = \begin{bmatrix} x'_1 \\ x'_2 \\ \dots \\ x'_N \end{bmatrix}$$

Metode FLD

8. Mencari rata-rata input FLD

$$\mu_{FLD} = \frac{1}{N} \sum_{i=1}^N X'_i \quad (1*(N-C))$$

9. Untuk $i = 1..C$, mencari rata-rata kelas X_i

$$\mu_i = \frac{1}{N_i} \sum_{i=1}^{N_i} X'_i \quad (1*(N-C))$$

dimana N_i adalah jumlah anggota kelas X_i

menyimpan μ_i , $i = 1..N$, dalam matriks μ_k , sehingga :

$$\mu_k = \begin{bmatrix} x'_1 \\ x'_2 \\ \dots \\ x'_C \end{bmatrix} \quad (C*(N-C))$$

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

$$\text{Rasio} = \frac{S_B}{S_W} \quad ((N-C)*(N-C))$$

13. Mencari vektor eigen dan nilai eigen dari matriks rasio, kemudian

mengurutkan berdasarkan nilai eigen, dari yang terbesar.

$$[v, d] = \text{eig}(\text{rasio})$$

$$\text{Nilai eigen dari rasio : } d = [d_1 \ d_2 \ \dots \ d_{(N-C)}] \quad (1*(N-C))$$

$$\text{Dimana } d_1 > d_2 > \dots > d_{(N-C)}$$

Vektor eigen yang bersesuaian dengan nilai eigen (dalam bentuk vektor kolom):

$$v = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1(N-C)} \\ w_{21} & w_{22} & \dots & w_{2(N-C)} \\ \dots & \dots & \dots & \dots \\ w_{(N-C)1} & w_{(N-C)2} & \dots & w_{(N-C)(N-C)} \end{bmatrix} \quad ((N-C)*(N-C))$$

14. Mereduksi vektor eigen dari rasio sesuai dengan nilai $m = (C - 1)$

$$v' = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{(N-C)1} & w_{(N-C)2} & \dots & w_{(N-C)m} \end{bmatrix} \quad ((N-C)*m)$$

Matriks transformasi W_{FLD} adalah :

$$W_{FLD} = v'^T \quad (m*(N-C))$$

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 :

$$\Omega = X W_{opt}$$

$$(N \times m) = (N \times n) \times (n \times m)$$

dimana :

- Ω adalah matriks bobot wajah pelatihan dengan ordo $(N \times m)$.
- X adalah matriks wajah pelatihan yang telah dinormalisasi dengan ordo $(N \times n)$.
- 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 :

$$\Omega_t = X_t W_{opt}$$

$$(1 \times m) = (1 \times n) \times (n \times m)$$

dimana :

- Ω_t adalah matriks bobot wajah ujicoba dengan ordo $(1 \times m)$.
- X_t adalah matriks dari wajah uji coba yang telah dinormalisasi dengan ordo $(1 \times n)$.
- 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 struktur 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 menghitung 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_{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 $|x_{1i} - x_{2i}| = (\sum (x_{1i} - x_{2i})^2)^{1/2}$.

3.2.1.3 Data Keluaran

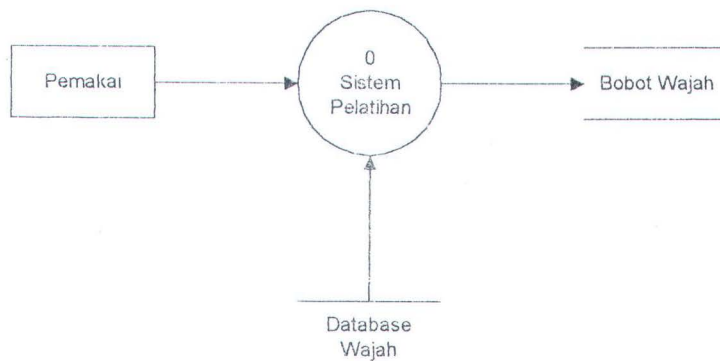
Data keluaran diperoleh dari pengurutan data hasil perhitungan jarak euclidean. Dari jarak terpendek akhirnya didapat foto dari data pelatihan yang mirip dengan foto dari data uji coba yang dimasukkan.

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.

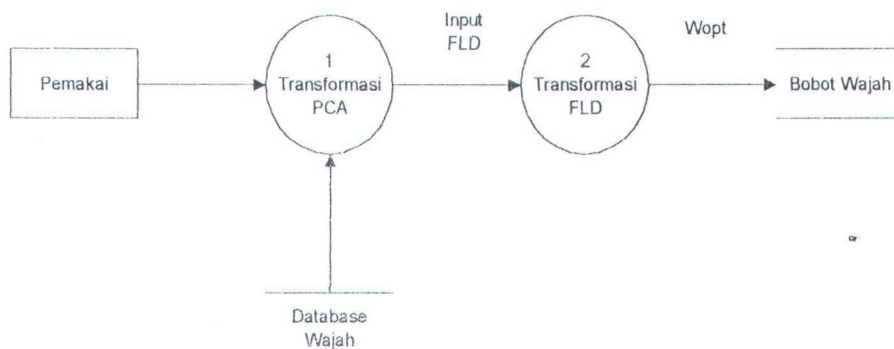
Bagian ini juga akan membahas perancangan sistem pengenalan wajah yang akan direpresentasikan dalam gambar, yaitu menggunakan Diagram Aliran Data (DAD) atau Data Flow Diagram (DFD). Dengan menggunakan diagram aliran data, dapat memperjelas semua proses dan aliran data yang terjadi dalam sistem pengenalan 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.

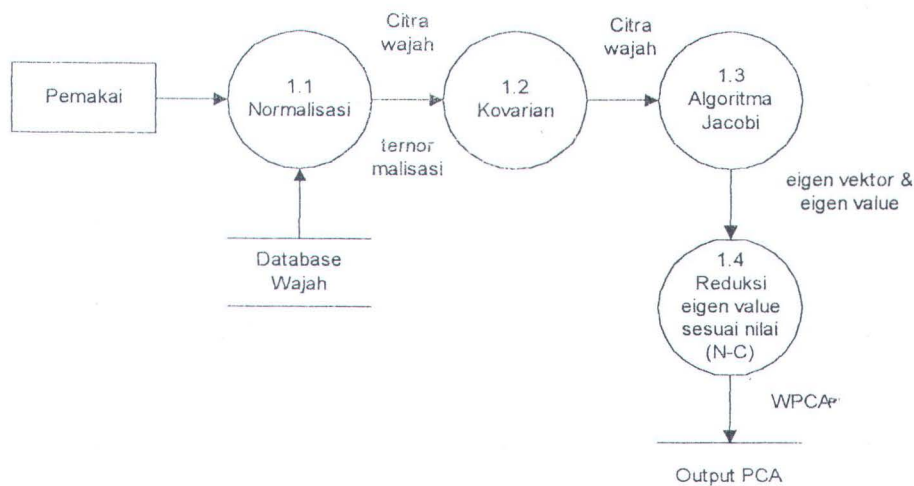


Gambar 3.2 DFD level 1 dari subsistem pelatihan

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.

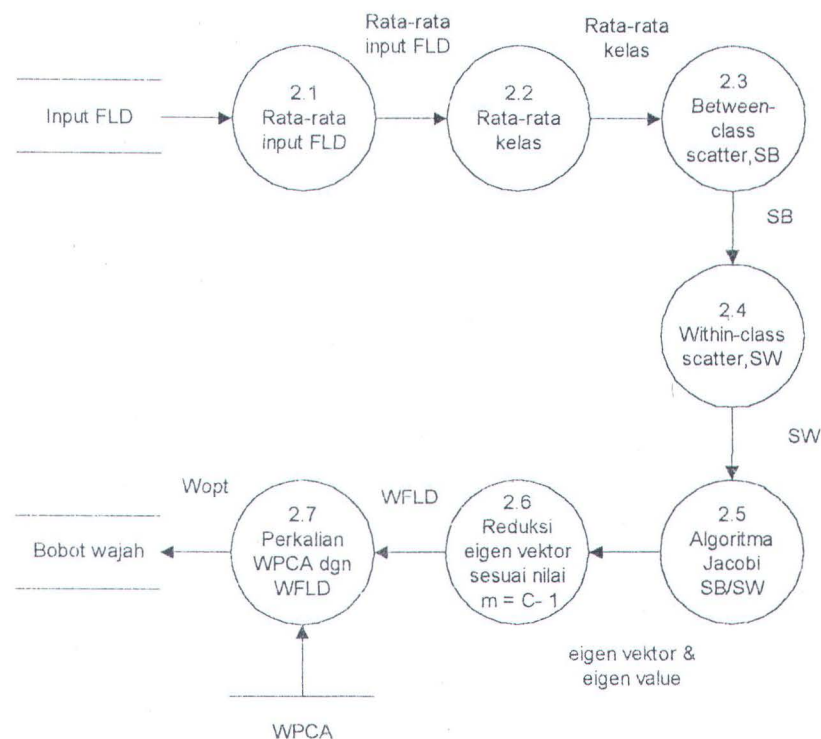
Proses dalam tranformasi PCA terdiri dari proses-proses sebagai berikut :

- 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.

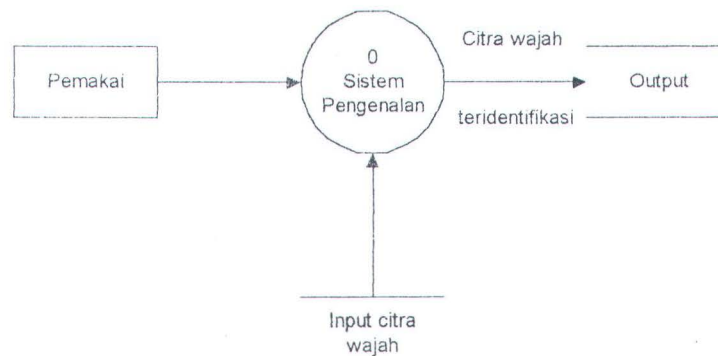


Gambar 3.4 DFD level 2 dari subsistem pelatihan Metode FLD

Proses dalam transformasi FLD terdiri dari proses-proses sebagai berikut :

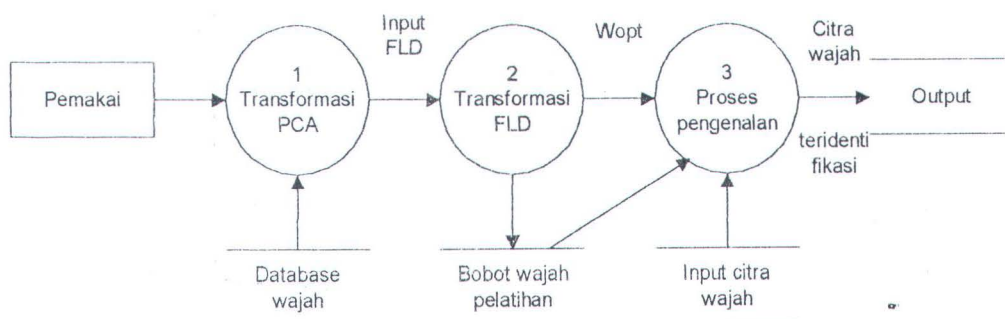
- Mencari rata-rata inputFLD.
- Mencari rata-rata kelas dari inputFLD.
- Mencari matriks *between-class scatter*, S_B .
- Mencari matriks *within-class scatter*, S_W .
- Mencari rasio antara S_B dengan S_W

- 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 menu 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 TFMMain.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 TFMMain.btnDeleteClick(Sender: TObject);
var idx: integer;
begin
    tblData.Delete;
    LoadData;
    ShowData;
end;

procedure TFMMain.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 TFMMain.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 TFMMain.Inisialisasi;
var i,j,k :integer;
begin
  setlength(imgdata,tblldata.recordcount);
  for k:=0 to tblldata.recordcount-1 do
  begin
    for i:=1 to 112 do
      for j:=1 to 92 do
        D[k+1,(i-1)*92+j]:=
          gettrvalue(imgdata[k].picture.bitmap.
            canvas.pixels[i-1,j-1]);
      end;
    end;
  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

[illegible]


```

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
end

```

Prosedur normalisasi matriks

Normalisasi matrik menggunakan rumus :
$$\frac{X_i}{\sqrt{\sum_{i=1}^n X_i^2}}$$

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 $\mu_i = \frac{1}{N_i} \sum_{i=1}^{N_i} X'_i$ }

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;

```

$$\{\text{Mencari } S_w = \sum_{i=1}^C \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i)\}$$

Mencari matriks jarak intra kelas (within-class scetter), diperoleh dari matriks hasil dari transformasi 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;
        end;
    end;

```

{Jarak Euclidean menggunakan rumus $|x_{1i} - x_{2i}| = (\sum (x_{1i} - x_{2i})^2)^{1/2}$ }

Proses ini digunakan untuk menghitung perbandingan bobot wajah ujicoba dengan wajah pelatihan.

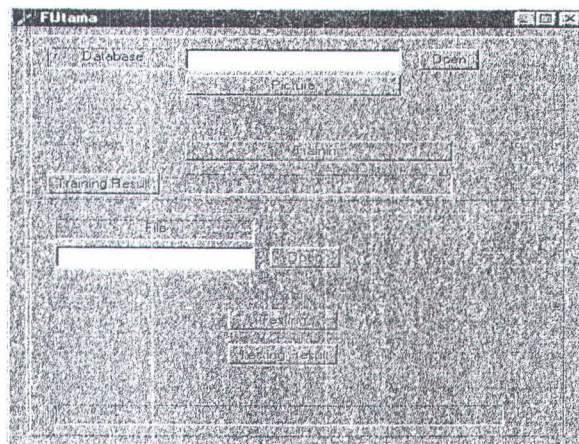
```

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;

```

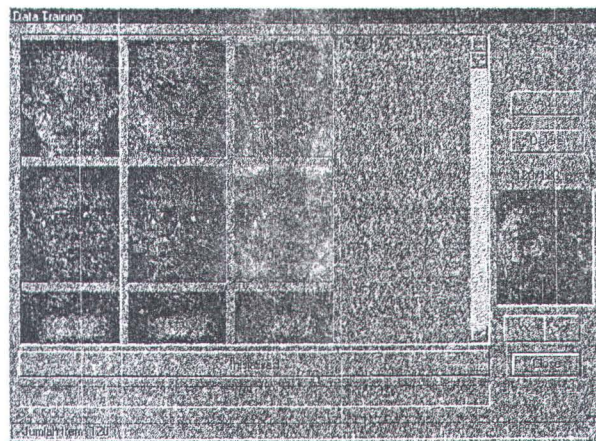
3.3.3 Implementasi Antarmuka

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 sistem 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 dilakukan 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

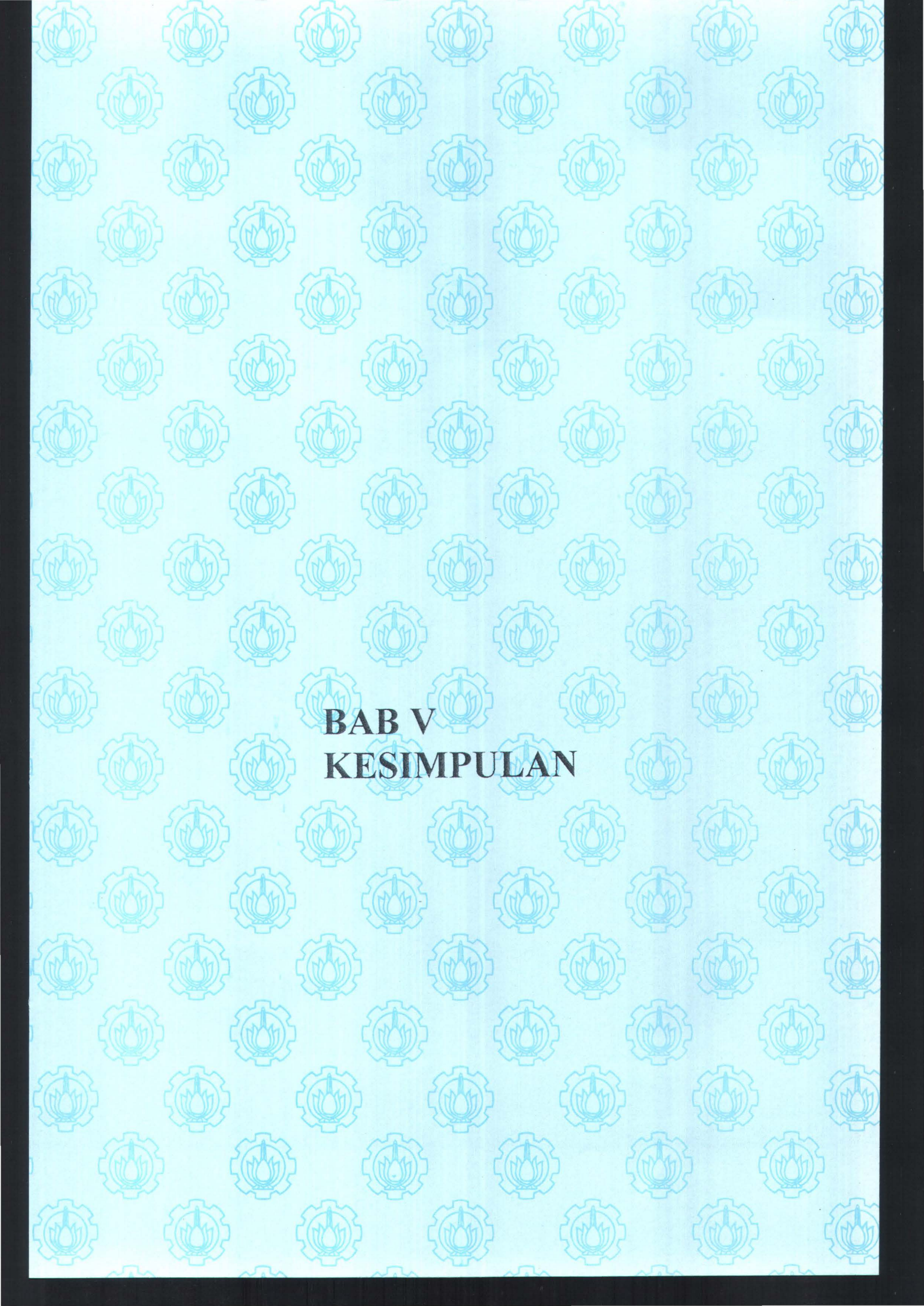
Berdasarkan hasil studi literature terhadap berbagai tugas akhir sarjana di Jurusan Teknik Informatika, terdapat dua tugas akhir mengenai pengenalan wajah yang dapat dijadikan sebagai bahan perbandingan dengan sistem pengenalan wajah yang dibuat dalam tugas akhir ini. Kedua tugas akhir yang relevan tersebut 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 pemilihan 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 digunakan untuk menemukan area wajah yang memiliki nilai log-likelihood 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

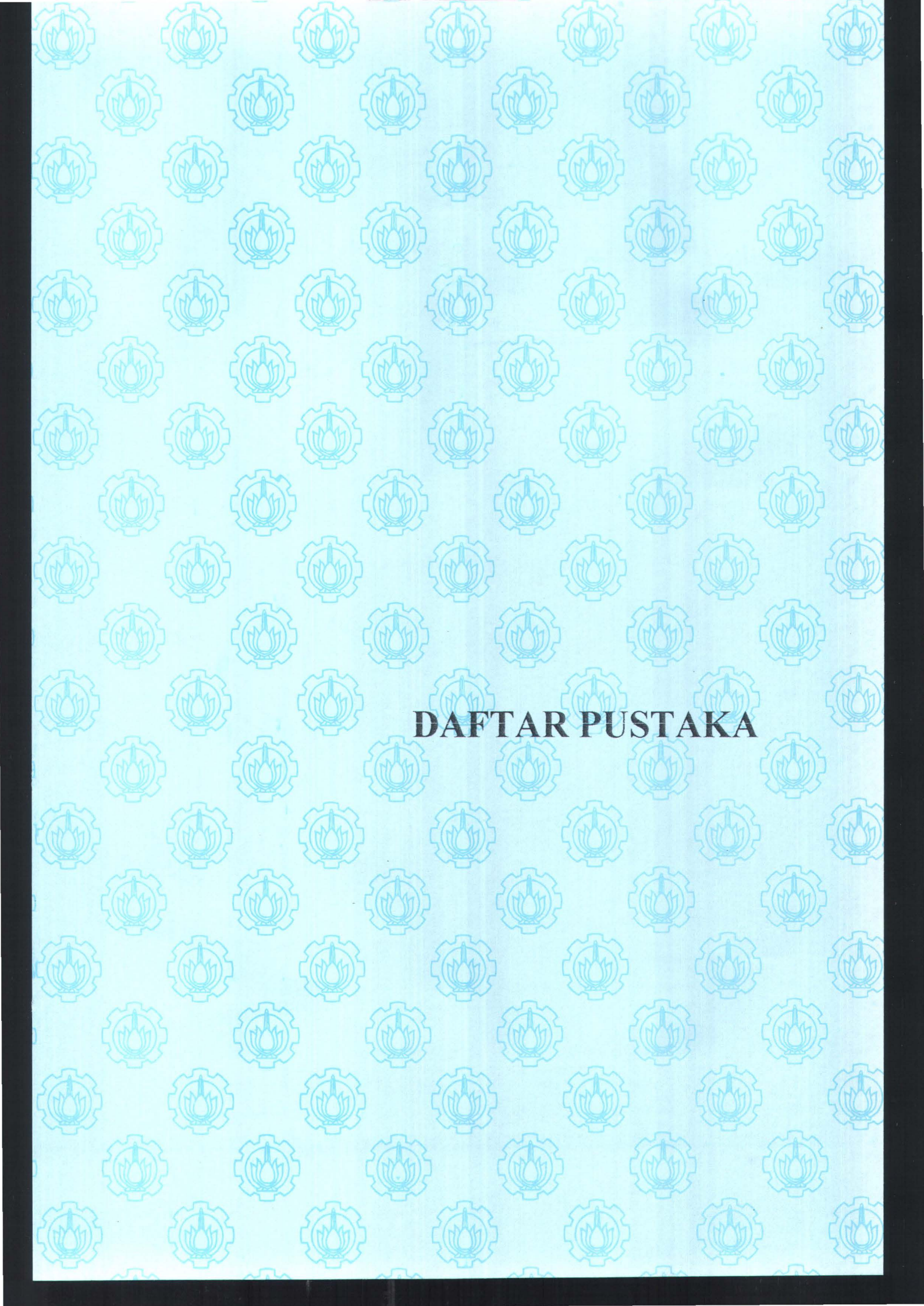
BAB V

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 :

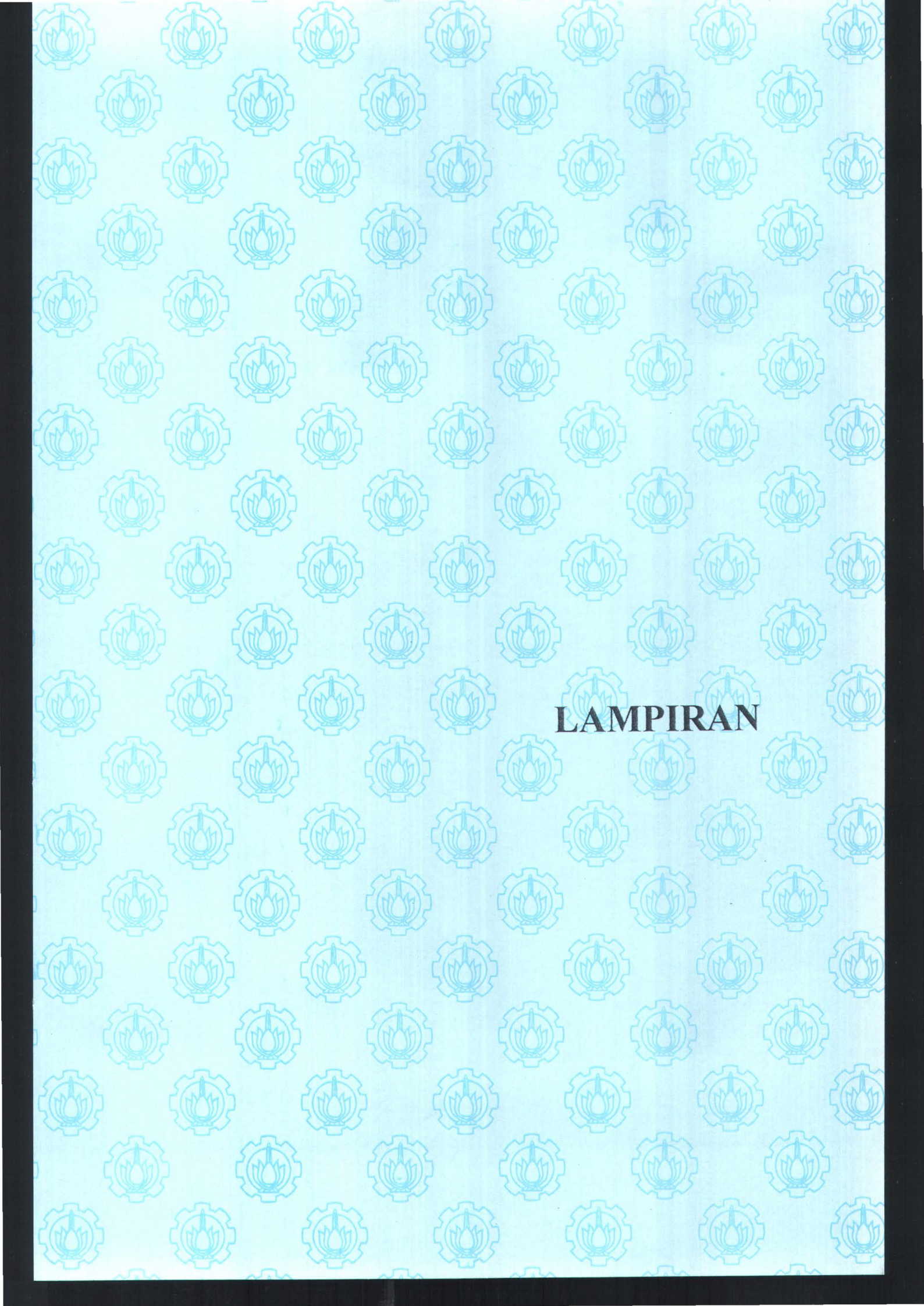
1. Hasil yang diperoleh dalam proses pengenalan wajah dengan menggunakan Metode *Fisherface* untuk data pelatihan terurut 80% sedangkan untuk data pelatihan terkendali 88%. Hasil ini menunjukkan bahwa pengenalan wajah dengan menggunakan Metode *Fisherface* cukup baik untuk diterapkan dalam sistem pengenalan 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

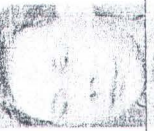
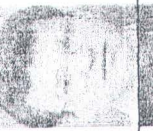
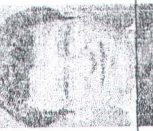
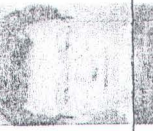
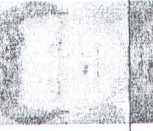
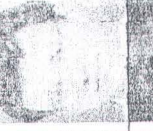
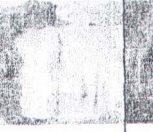
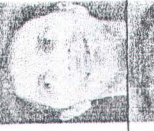
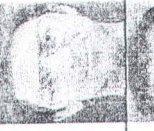


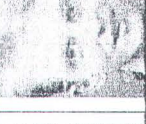
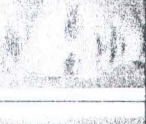
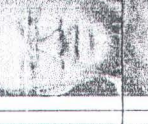
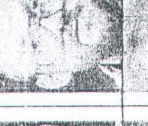
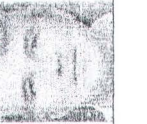
LAMPIRAN

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 indeksinya.

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				2,5 menit				3,5 menit			
	Pengenalan : 10 detik				20 detik				30 detik			

4.3.3 Evaluasi Hasil Uji Coba

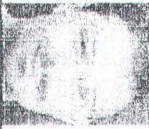
Prosentase keberhasilan pengenalan wajah seperti yang terlihat pada tabel 4.1 dan 4.2 sangat tergantung pada pemilihan data pelatihan yang digunakan. Pemilihan data pelatihan yang mewakili semua variasi atau mendekati semua variasi dalam data uji coba memberikan hasil pengenalan yang meningkat jika dibandingkan 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.

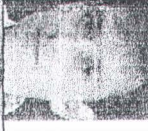
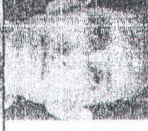
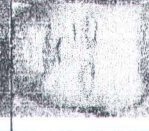
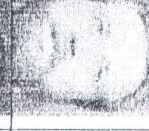
Tabel 4.3
Perbandingan keberhasilan pengenalan berdasarkan metode pemilihan data pelatihan yang digunakan.

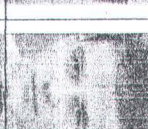
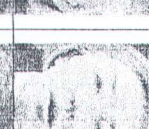
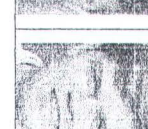
Tipe Data Uji Coba	Uji Coba Terurut			Uji Coba Terkendali		
	Presentase Keberhasilan	Waktu Komputasi		Presentase Keberhasilan	Waktu Komputasi	
		Pelatihan	Pengenalan		Pelatihan	Pengenalan
Jumlah Data 3	67%	1,5 menit	10 detik	76%	1,5 menit	10 detik
Jumlah Data 4	72%	2,5 menit	20 detik	83%	2,5 menit	20 detik
Jumlah Data 5	80%	3,5 menit	30 detik	88%	3,5 menit	30 detik

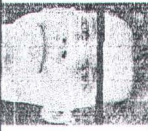
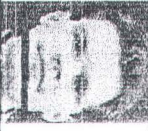
Waktu yang diperlukan untuk melakukan proses pelatihan berbanding lurus dengan banyaknya data yang ditraining. Semakin banyak jumlah data yang ditraining, maka semakin banyak pula waktu yang dibutuhkan. Dari pengamatan hasil uji coba baik dengan data pelatihan terurut maupun data pelatihan terkendali, 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 yang ada.

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										

Polak	1	2	3	4	5	6	7	8	9	10
34										
35										
36										
37										
38										
39										



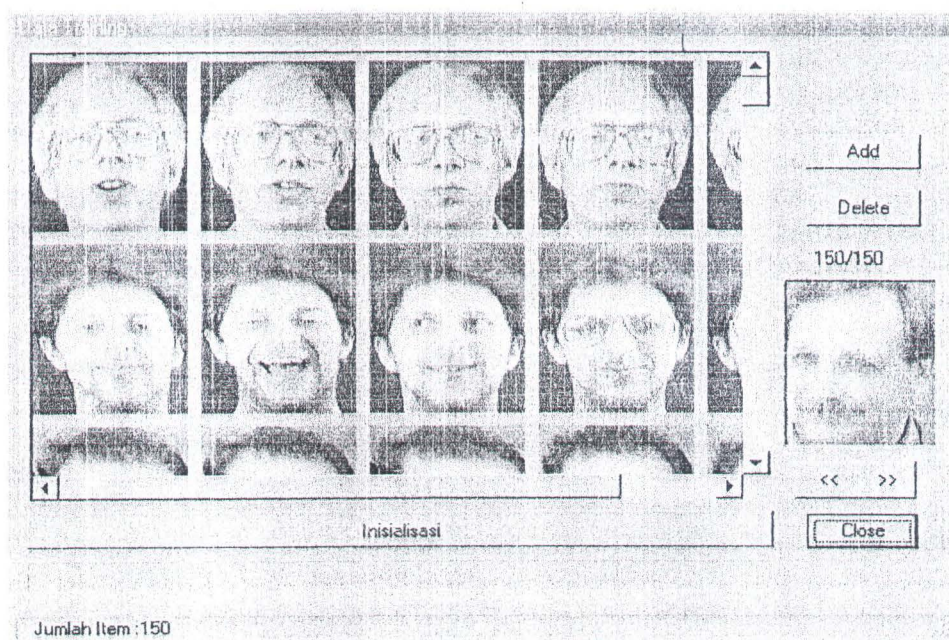
LAMPIRAN B

PETUNJUK PEMAKAIAAN 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*.

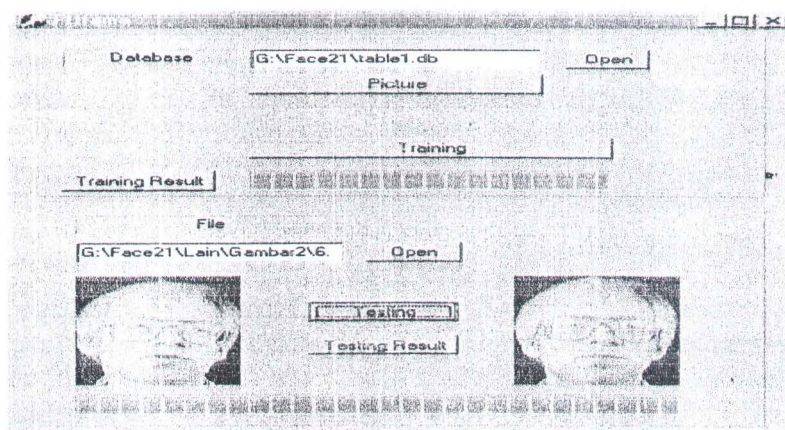
	1	2	3	4	5	6	7	8	9	10
1	0.01171492	0.00006565	-0.01369784	0.01210447	0.02169935	0.00109111	-0.02409829	0.01037314	0.01800332	0.00589024
2	0.01037314	0.01800332	0.00589024	-0.01723666	0.00432944	-0.00177895	0.00068854	-0.00775574	0.01002904	-0.01592338
3	-0.01002904	-0.01592338	0.00531281	-0.01696295	0.00133929	0.00373673	-0.00225634	-0.00582339	0.01025365	-0.01762612
4	0.01025365	-0.01762612	0.00425941	-0.01804291	0.00382479	-0.00483987	0.00256227	-0.00392725	0.01396186	0.01795281
5	0.01396186	0.01795281	0.00527013	-0.01719109	0.00106055	-0.00637404	-0.00077291	-0.00545689	-0.01234256	-0.01734838
6	-0.01234256	-0.01734838	0.00270587	-0.01880883	-0.00231825	0.00697301	-0.00101475	-0.00294918	0.00823159	-0.01948548
7	0.00823159	-0.01948548	0.00420268	-0.01846832	-0.00002780	0.00808657	0.00069007	-0.00451015	-0.00728855	0.01863453
8	-0.00728855	0.01863453	0.00442371	-0.01866194	0.00276340	-0.00673841	-0.00190066	-0.00450492	0.00903551	-0.01889763
9	0.00903551	-0.01889763	0.00574526	-0.01734636	0.00118390	0.00747378	-0.00199146	-0.00519384	-0.00518875	-0.01495896
10	-0.00518875	-0.01495896	0.00437282	-0.01485503	-0.00084990	-0.00982137	-0.00075790	-0.00335315	-0.00555196	-0.01547944

Gambar B.2 Tampilan matriks hasil dari proses training

Langkah-langkah dalam tahap testing:

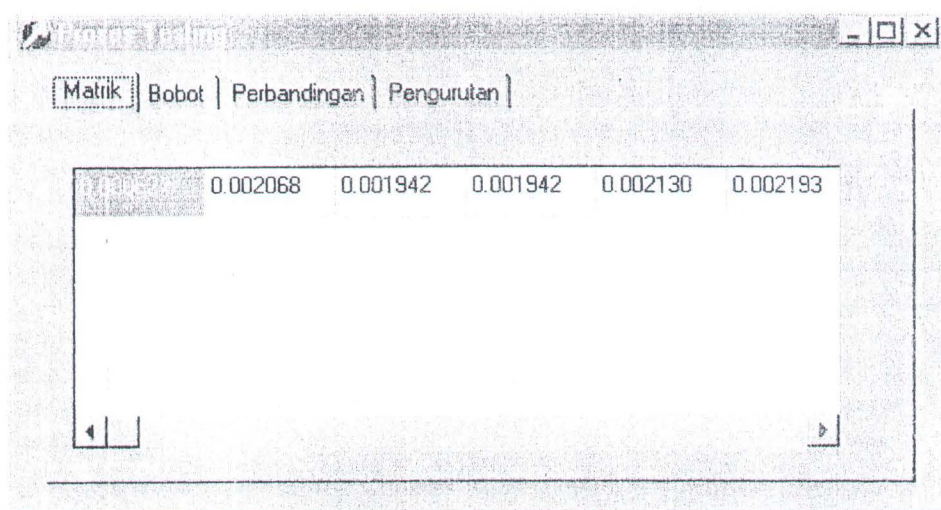
Langkah 1: Click tombol *Open*, tombol ini digunakan untuk menginputkan gambar yang akan diuji coba.

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*.



0.002068	0.001942	0.001942	0.002130	0.002193
----------	----------	----------	----------	----------

Gambar B.4 Tampilan matriks dari hasil proses Testing

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
Dosen

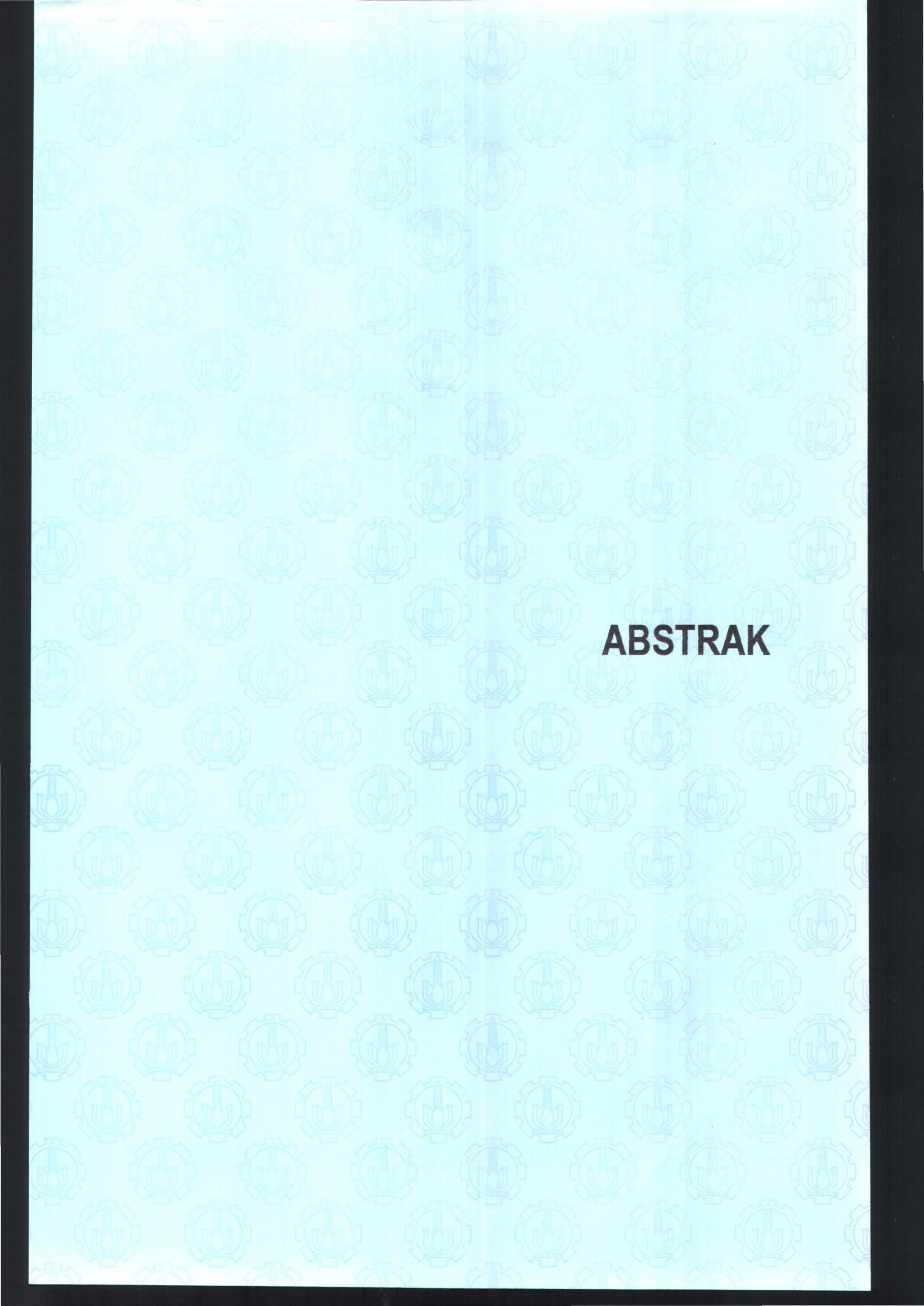
Pembimbing II
Kabid. Data Bappeda Kab. Kediri



Dr. Ir. Joko Lianto Buliali, M.Sc
NIP. 131996151

Drs. Sugeng Sundoro, M.M
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 detail** 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

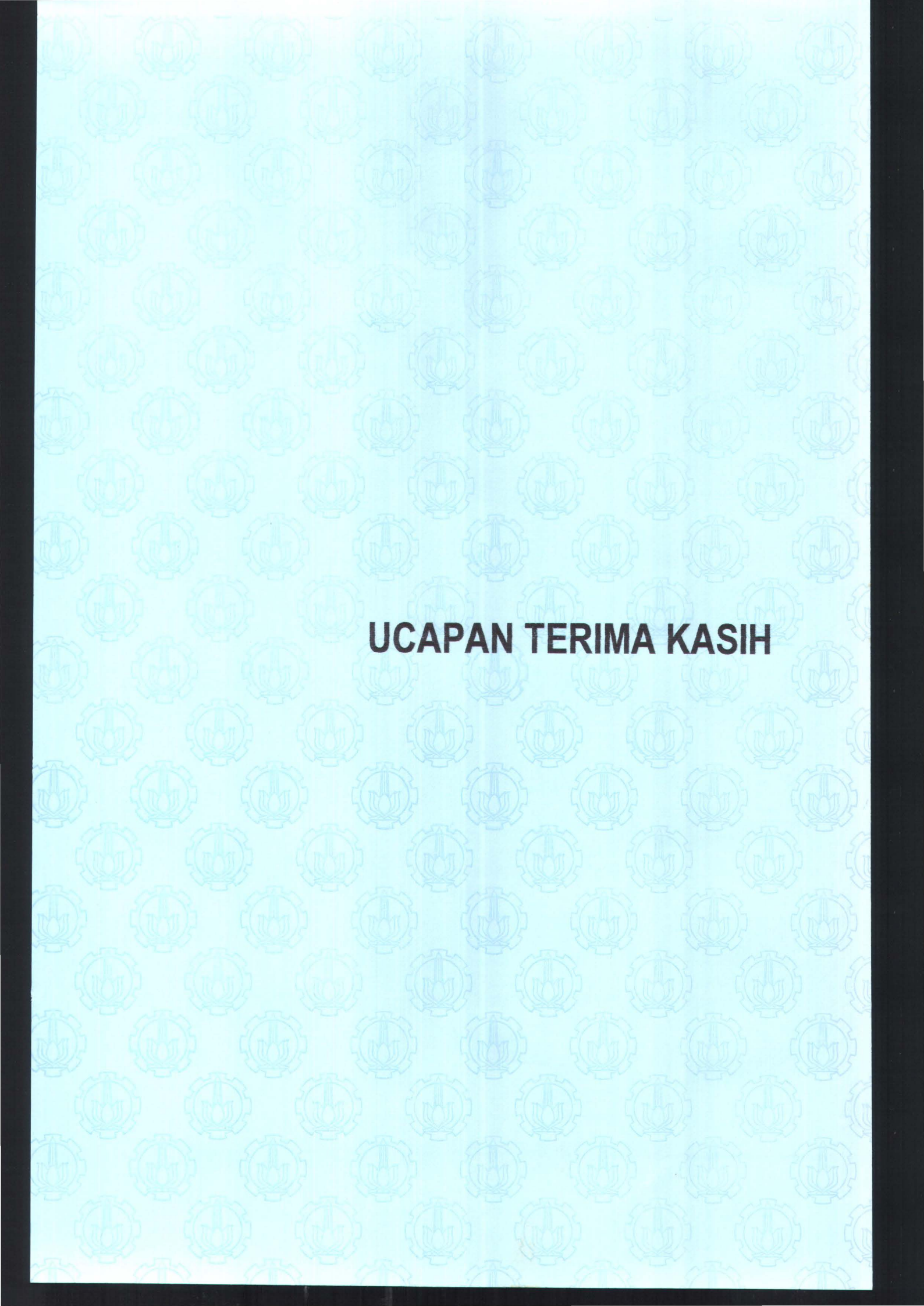
Segala puji syukur ke hadirat Tuhan Yang Maha Esa, Maha Kasih dan Penyayang yang telah memberikan berkat dan rahmat, serta kasih setiaNya, sehingga penulis telah dapat menyelesaikan penyusunan tugas akhir ini, dengan judul ***“Perancangan dan Pembuatan Sistem OLAP untuk Analisa Studi Kelayakan Proyek 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, inspirasi 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 khawatir. *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 izin, 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.



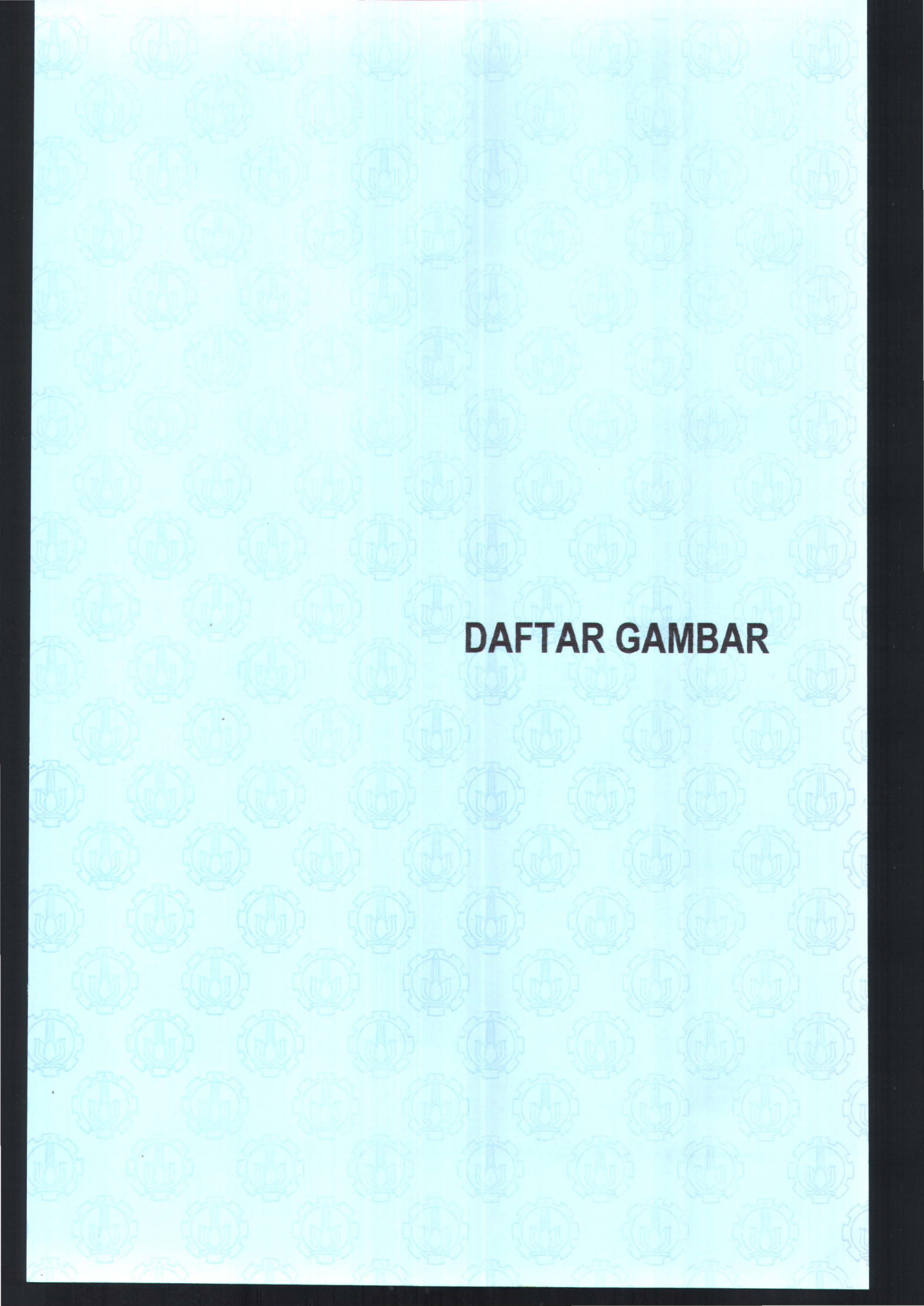
DAFTAR ISI

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 <i>Feasibility Study</i>	11
2.1.2 Indikator-Indikator Keuangan	11
2.1.2.1 <i>Return On Total Assets (ROA)</i>	12
2.1.2.2 <i>Return On Investment (ROI)</i>	13
2.1.2.3 <i>Internal Rate of Return (IRR)</i>	13
2.1.2.4 <i>Payback Periode (PBP)</i>	13
2.1.2.5 <i>Service Coverage (SC)</i>	14
2.1.3 Penentuan Prioritas (<i>Ranking</i>) Proyek.....	14
2.1.3.1 <i>Benefit, Disbenefit, Cost</i>	14
2.1.3.2 <i>Benefit Cost Analysis</i>	16
2.1.3.3 <i>Analisis Benefit Cost Ratio untuk Membandingkan Alternatif</i>	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 <i>ROLAP</i>	22
2.2.3.2 <i>MOLAP</i>	24
2.2.3.3 <i>MOE</i>	26
2.2.3.4 <i>HOLAP</i>	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	

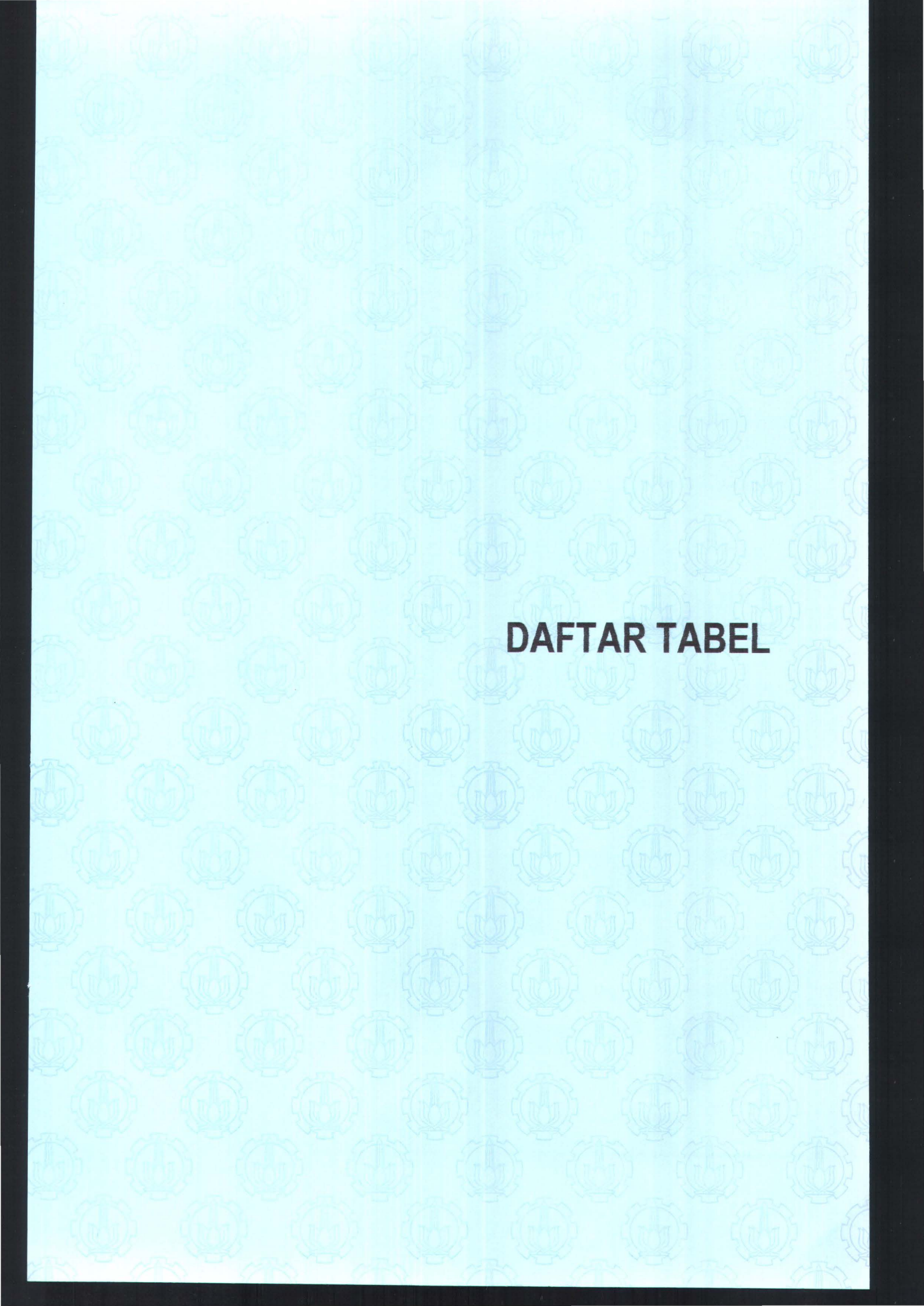


DAFTAR GAMBAR

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	PENG GAMBARAN 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 PENGGMABARAN 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.

1.3 Permasalahan

Permasalahan yang timbul sehingga diperlukan solusi pada tugas akhir ini adalah:

- Bagaimana membuat sebuah sistem yang dapat membantu pengguna untuk mengintegrasikan, 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 Analytical 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.

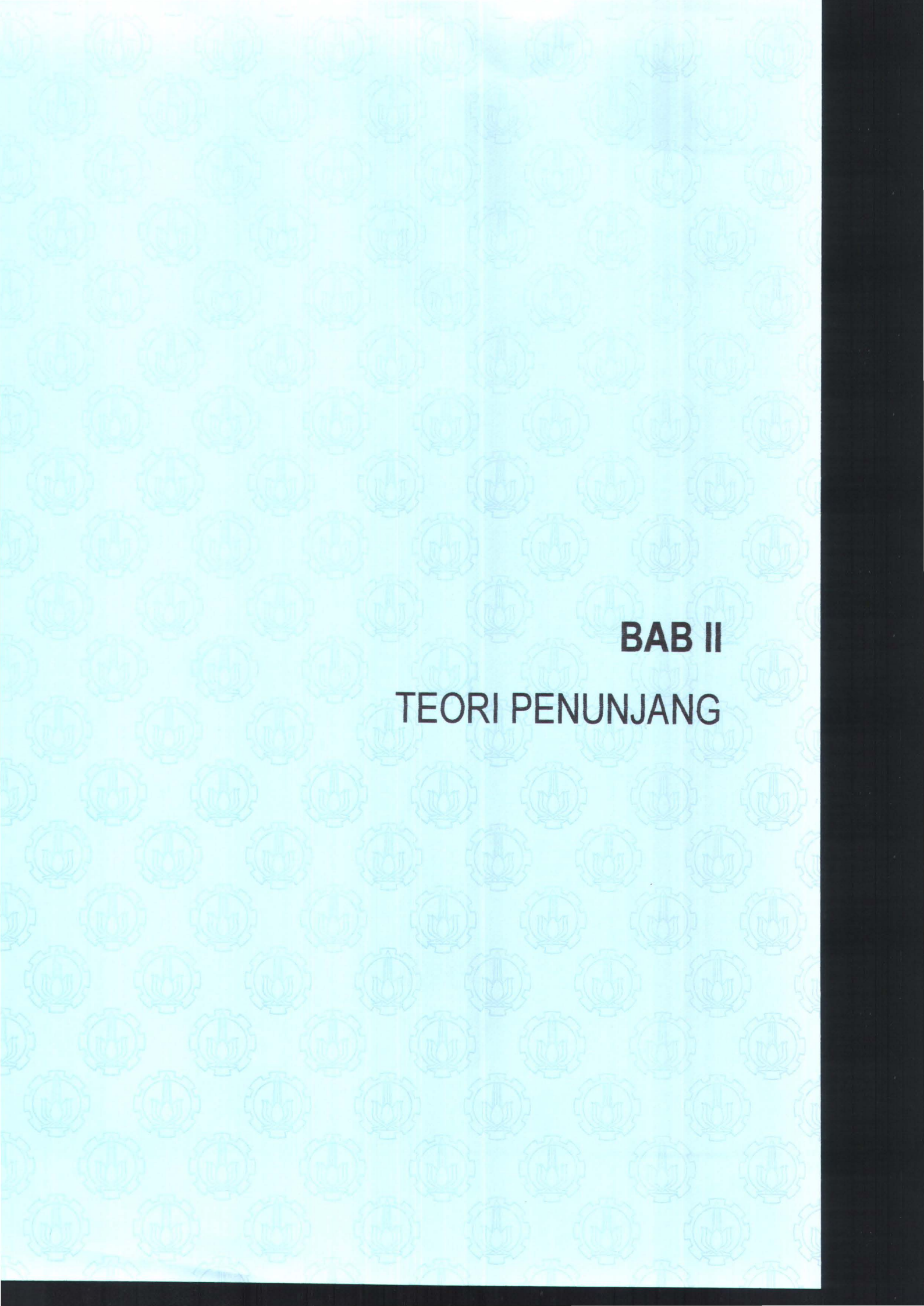
Bab IV, *Analisis Kebutuhan Sistem*, bab ini berisi identifikasi kebutuhan pada proses perencanaan di Bappeda dan obyektif sistem OLAP yang akan dibuat sesuai dengan hasil identifikasi tersebut, dan juga berkaitan dengan analisis aspek keuangan.

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.

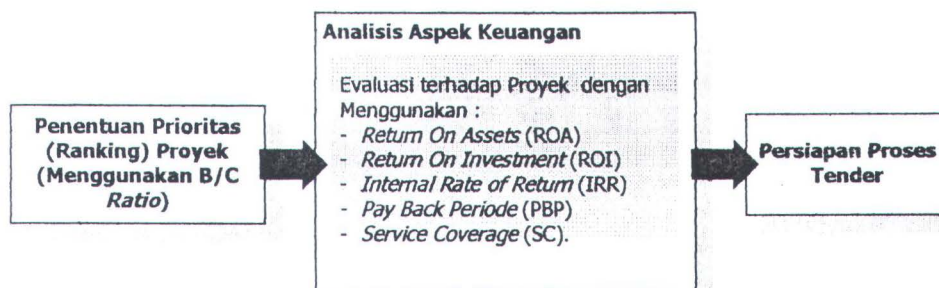


2.1.1 Analisis Keuangan dalam *Feasibility Study*

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, khususnya 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:

$$IRR^* = IRR_j - \frac{NPVTotal_j(IRR_j - IRR_{j-1})}{NPVTotal_j - NPVTotal_{j-1}}, \text{ dimana NPV dihitung dari:}$$

$$NPVTotal_j = \sum_{t=0}^n \frac{NF_t}{(1 + IRR_j)^t}$$

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:

$$PBP = (n-1) + \left[NF_0 - \sum_{t=1}^{n-1} NF_t \right] \left(\frac{1}{NF_n} \right)$$

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 prioritasasi akan dilakukan persumber dana terlebih dahulu.

Berikut ini adalah beberapa tinjauan yang dipergunakan Bappeda dalam melakukan prioritasasi 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 (<i>disbenefit</i>) 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:

$$B/C = \frac{(\text{benefits} - \text{disbenefits})}{\text{costs}} \quad \text{atau} \quad B/C = \frac{(B - D)}{C}$$

2.1.3.3 Analisis *Benefit Cost Ratio* untuk Membandingkan Alternatif

Proses pengambilan keputusan pada proyek-proyek pemerintah juga akan melalui tahapan-tahapan yang sistematis 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 tersedia 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 orientasi 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 dibandingkan tersebut bersifat *dependent*, maka dari masing-masing alternatif tersebut perlu dibandingkan 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 keadaan 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 perbandingan (*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 Quarter	Glasgow	London	Aberdeen	
	Q1	29726	43555	53210	
	Q2	30443	48244	34567	
	Q3	30582	56222	45677	
	Q4	31390	45632	50056	

(b)

	Property type	City			
		Glasgow	London	Aberdeen	
Time	Flat	15056	14555	14578	15890
	House	14670	15888	16004	15500

(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 operasi matrik pada setiap sel yang bersangkutan dalam cube ($\text{Rata-rata total pendapatan untuk setiap staf penjualan} = \frac{\text{Total pendapatan}}{\text{Jumlah staf penjualan}}$)

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):

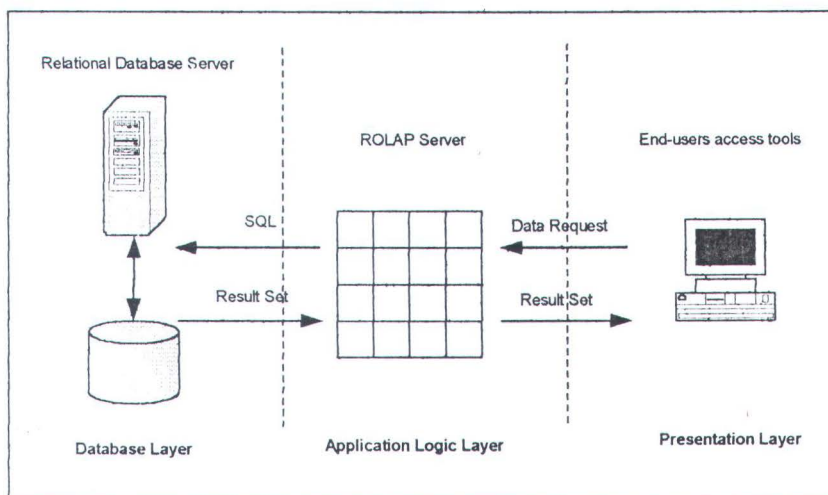
2.2.3.1 ROLAP

Relational OLAP (ROLAP) merupakan salah satu teknologi OLAP yang 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

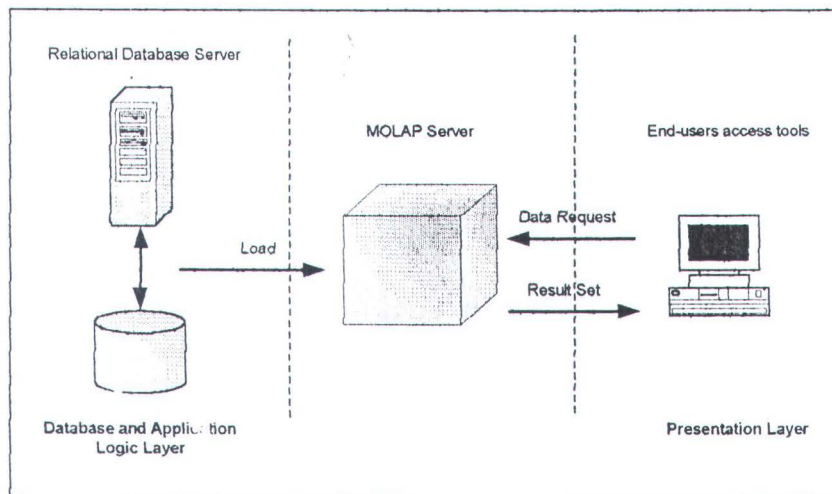
Permasalahan-permasalahan pada proses pengembangan ROLAP adalah sebagai berikut:

- 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

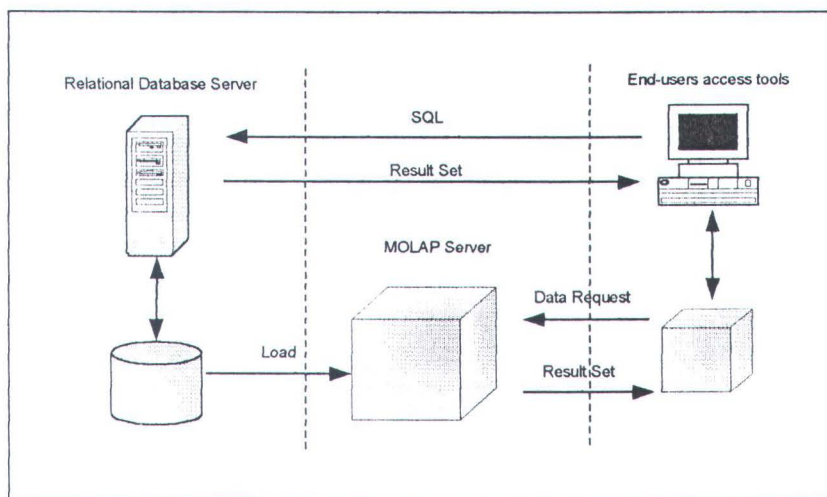
Permasalahan yang dihadapi dalam pengembangan sebuah alat MOLAP antara lain adalah:

- 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 untuk mengatasi permasalahan-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 permasalahan tersebut tanpa mempedulikan bagaimana komponen-komponen tersebut akan diimplementasikan. Akhir dari tahap ini nantinya akan kita dapatkan sekumpulan *requirement specification*

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.

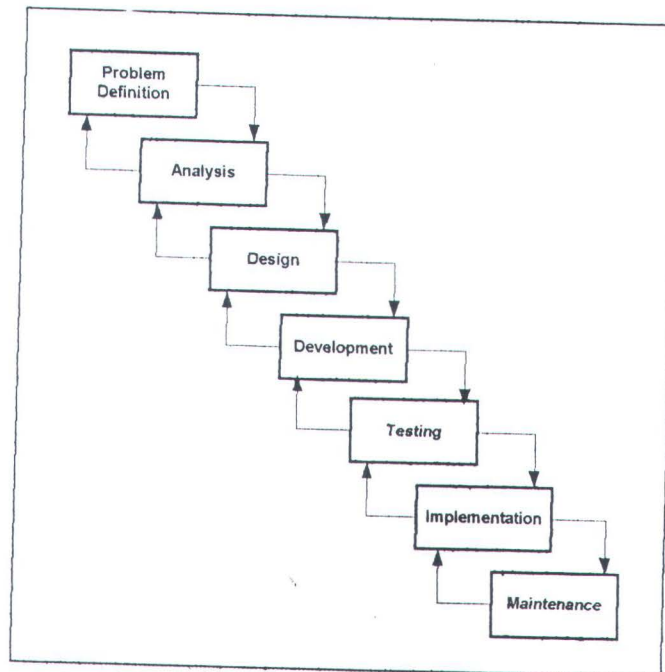
6. *Implementation*

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

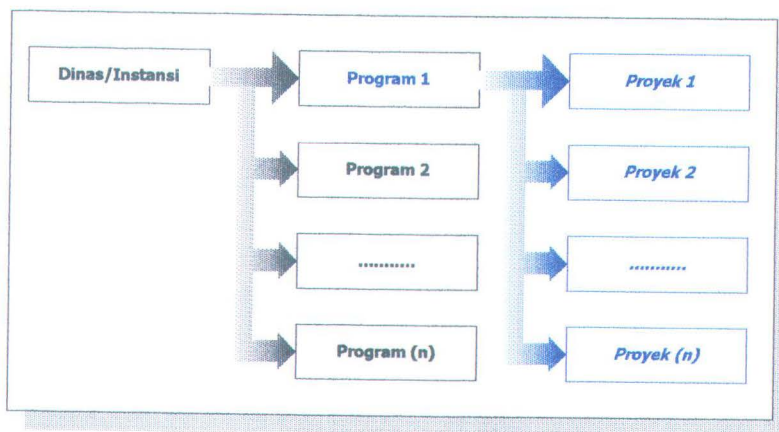
Data struktur proyek berisi kegiatan-kegiatan atau proyek-proyek pembangunan yang dilakukan oleh pemerintah daerah (Kabupaten Kediri) dalam setiap tahun perencanaan. Perencanaan proyek-proyek pembangunan tersebut merupakan kegiatan rutin dari pemerintah daerah dalam rangka melaksanakan aktivitas pembangunan 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 diidentifikasi 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 di beberapa tempat yang membutuhkan peranan komputer sebagai sarana pendukung kerja. Masing-masing komputer belum terhubung dalam sebuah LAN dengan menggunakan *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 perencanaan yang cepat dan efektif tersebut akan dapat dilaksanakan jika:

- Seluruh elemen rencana 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 manajemen 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 kompleks.

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.

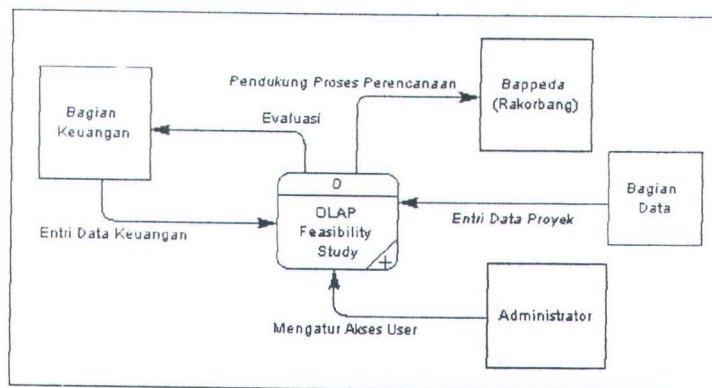
5.3.1 DAD Level 0

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:

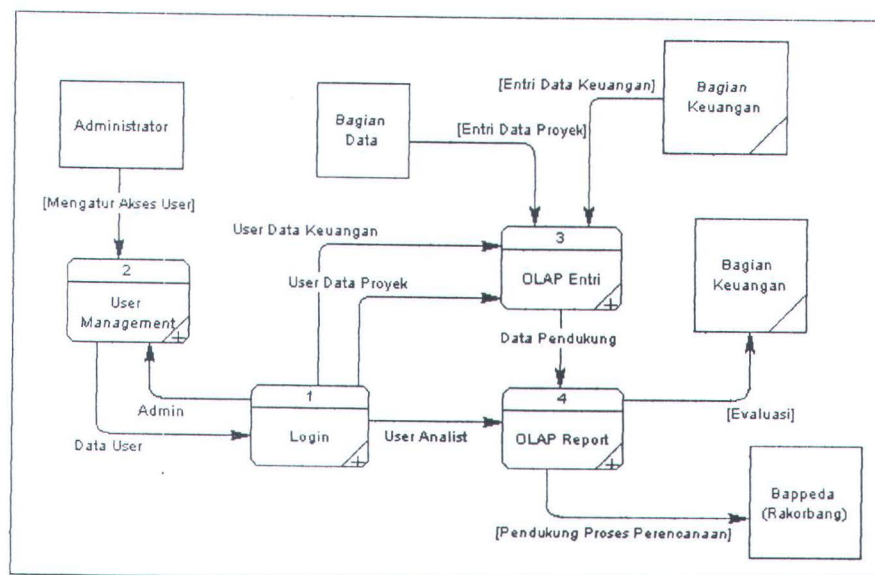
- Bagian Data, melakukan entri data-data proyek berkaitan dengan struktur dan kelas proyek.

- 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 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 Analyst* 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

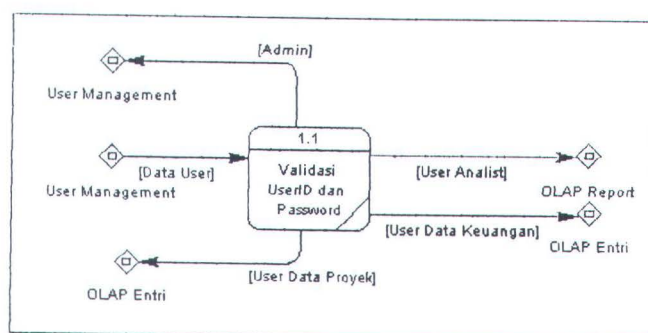
perencanaan yang dilakukan Bappeda dalam Rakorbang dan juga digunakan Bagian Keuangan untuk melakukan evaluasi terhadap proyek.

5.3.3 DAD Level 2

Berikut akan lebih didetilkkan 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*

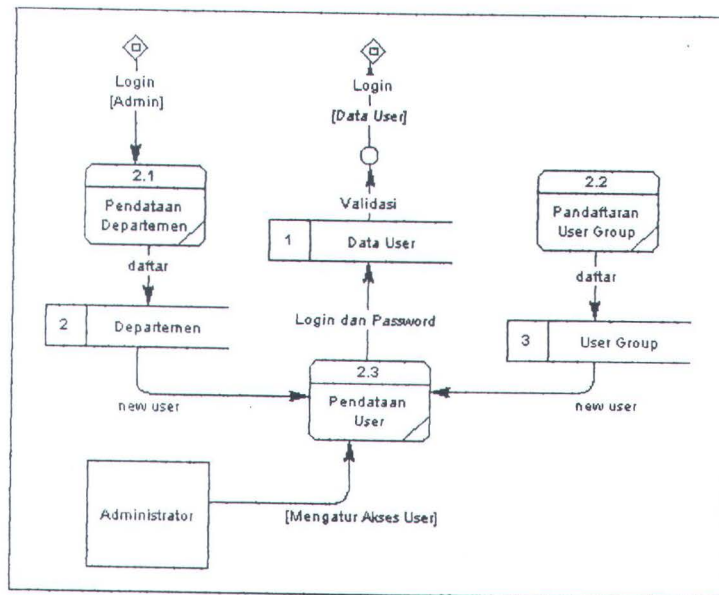
Dari data *user* yang terdapat pada *User Management*, maka proses Validasi kombinasi UserID dan Password akan menentukan kelompok *user group* bersangkutan. Jika dalam proses validasi dengan data *user* tersebut menyatakan kedudukan seorang pengguna dalam sebuah kelompok *user group* adalah sebagai 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

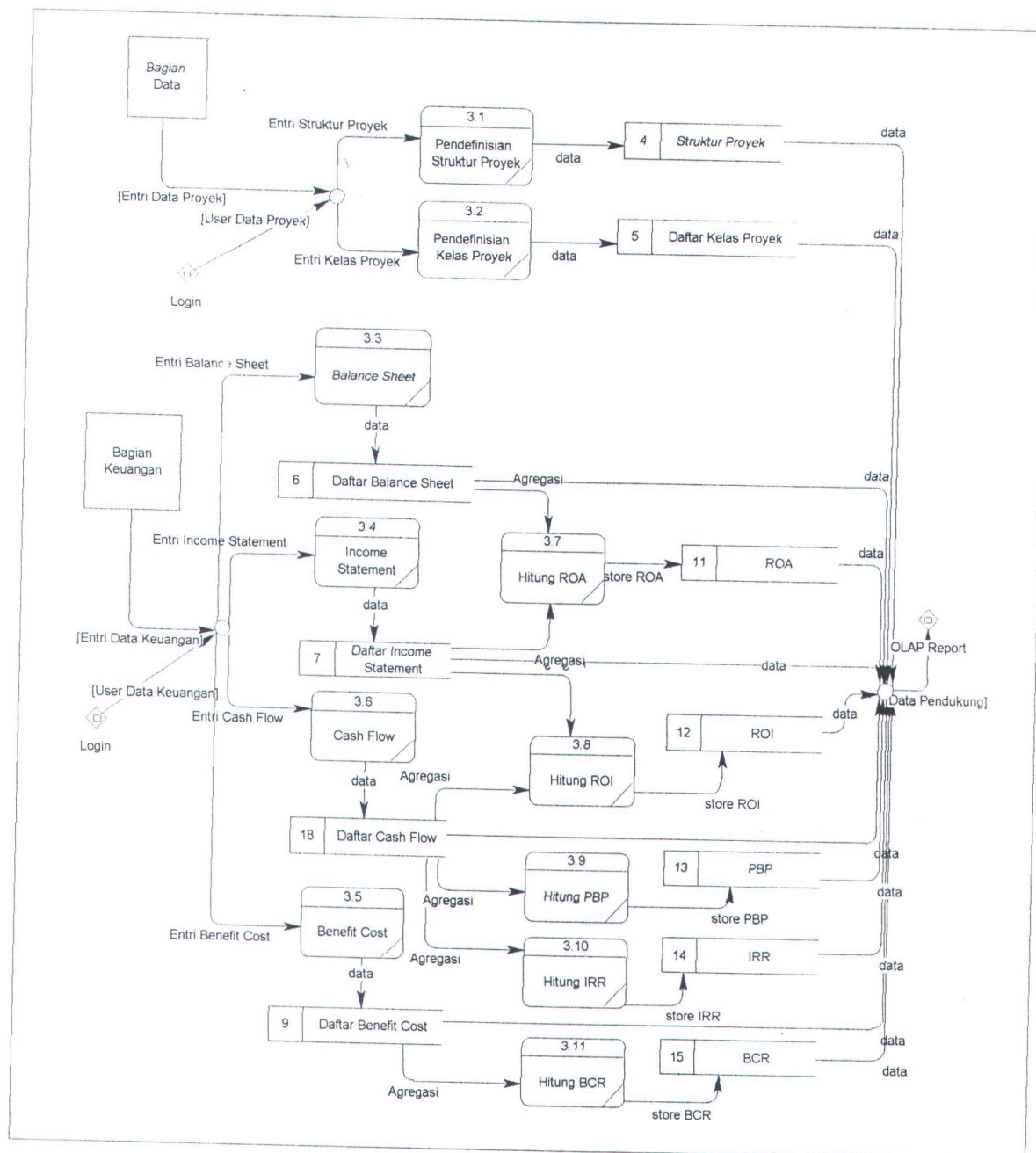
DAD level 2 OLAP Entri menggambarkan proses entri (input) data-data pendukung yang nantinya akan mendukung sistem OLAP yang akan dibuat. Data-data pendukung berasal dari bagian keuangan dan bagian data.

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 didefinisikan. 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.

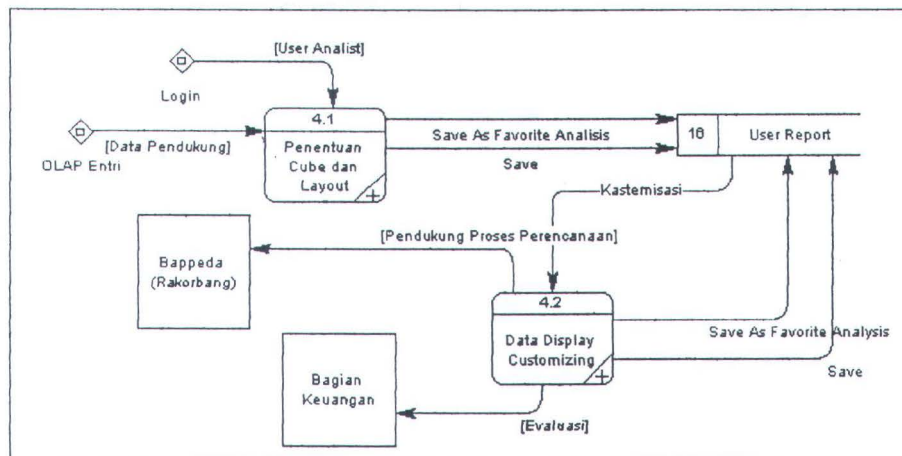


Gambar 5.5 DAD Level 2 OLAP Entry

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 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 penjabar dari masing-masing proses yang ada di level 2 yang masih bisa didetilkan.

5.3.4.1 DAD Level 3 Penentuan *Cube* dan *Layout*

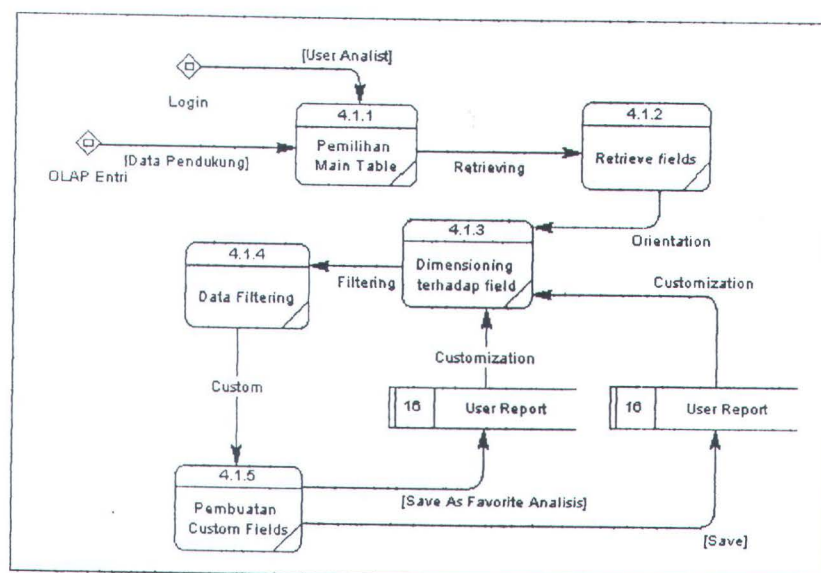
DAD level 3 Penentuan *Cube* dan *Layout* menggambarkan proses pemilihan terhadap field-field pada sebuah tabel utama (*main table*) untuk kemudian ditentukan orientasi field tersebut dalam penampakan (*layout*) pada sebuah dokumen analisis. DAD ini juga akan menggambarkan proses *filtering* data dan 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 analisis 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

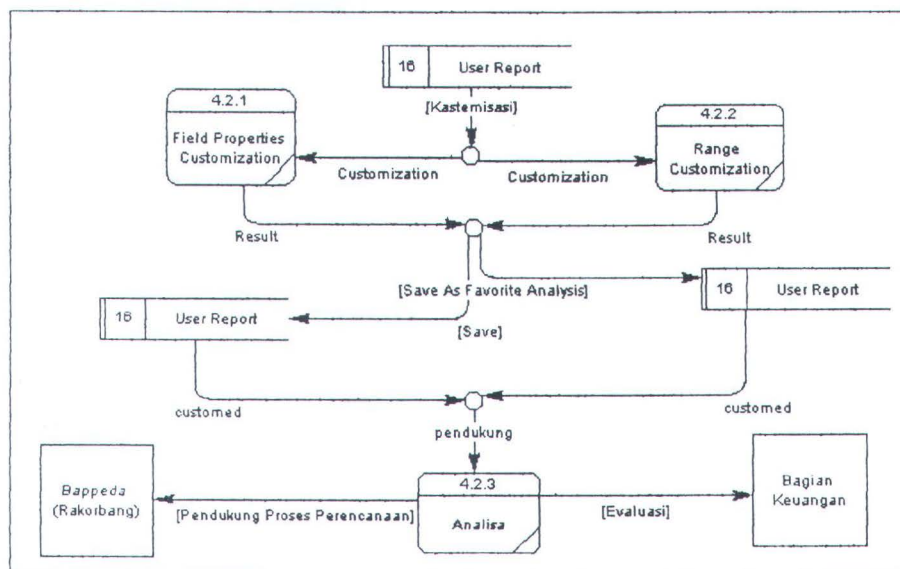
5.3.4.2 DAD Level 3 Data Display Customizing

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

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).

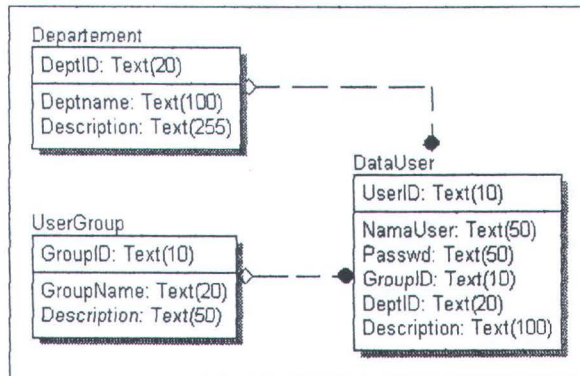
5.2.1 Desain Data *User Management*

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 disusun 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.

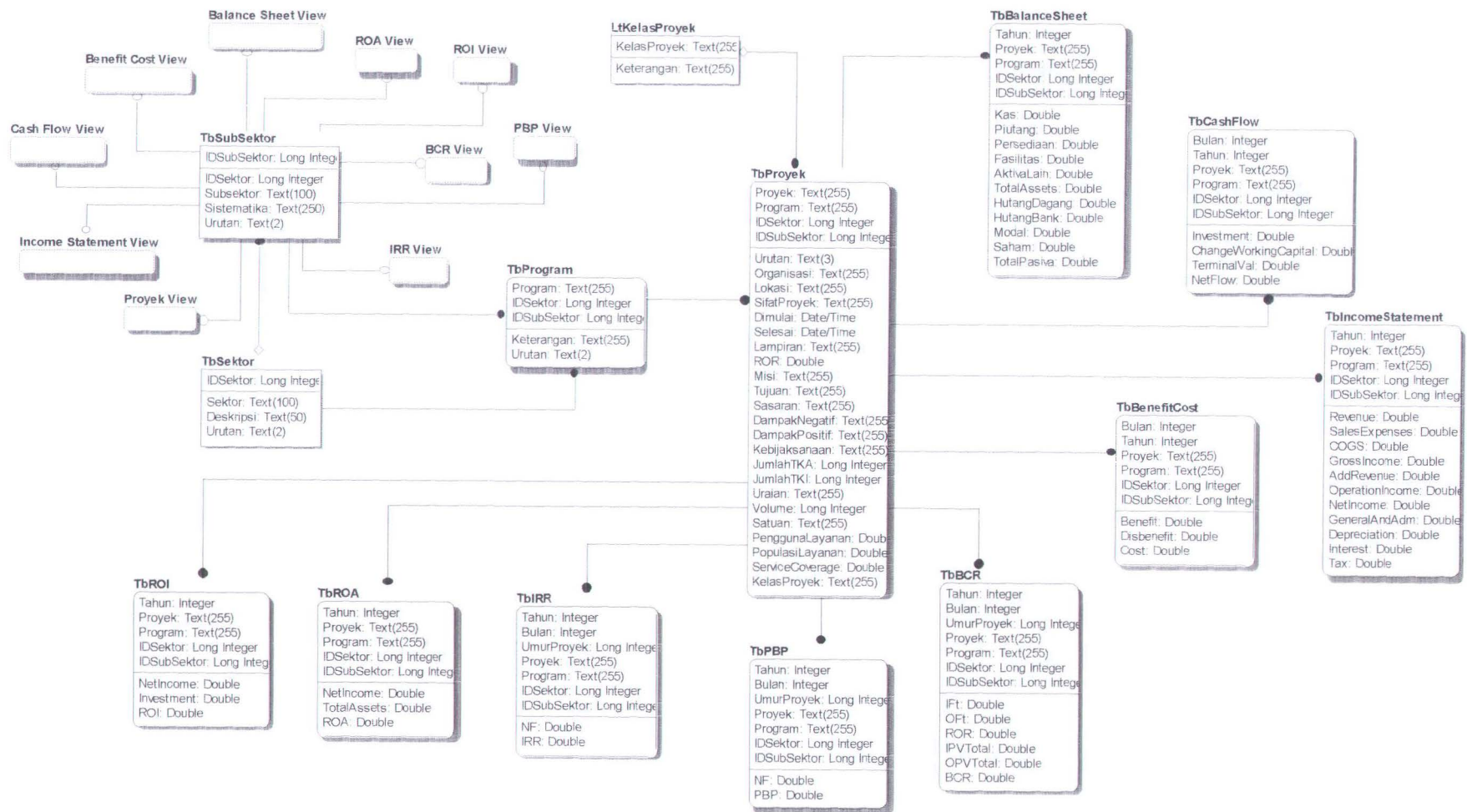
5.2.3 Desain Data OLAP Report

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 LtColumns ditunjukkan pada Gambar 5.11.

LtTables digunakan untuk menyimpan Tabel dan LtColumns digunakan untuk menyimpan Kolom yang dipakai untuk proses analisis pada OLAPReport.

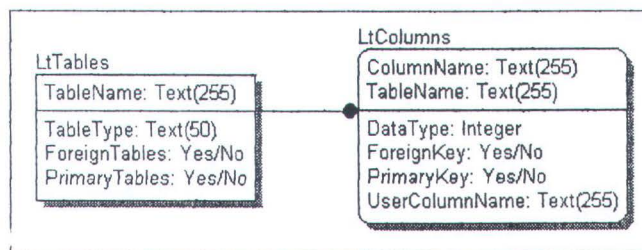


Gambar 5.1 ERD OLAP Entri

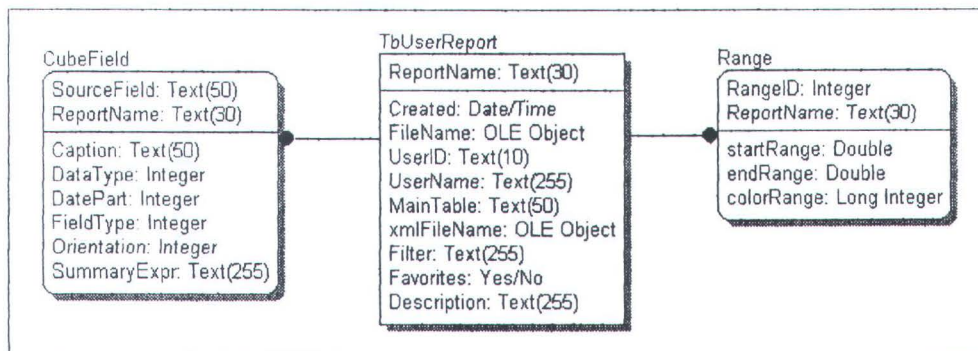
5.2.3.2 Desain Data Penyimpanan Display Analisis

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.

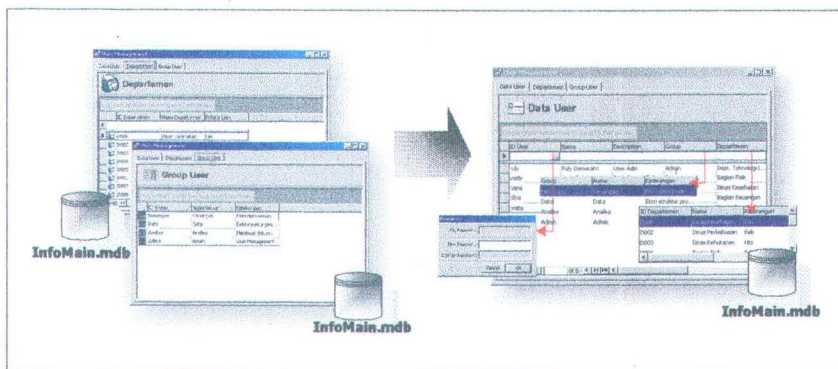
5.3 Desain Antarmuka

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.

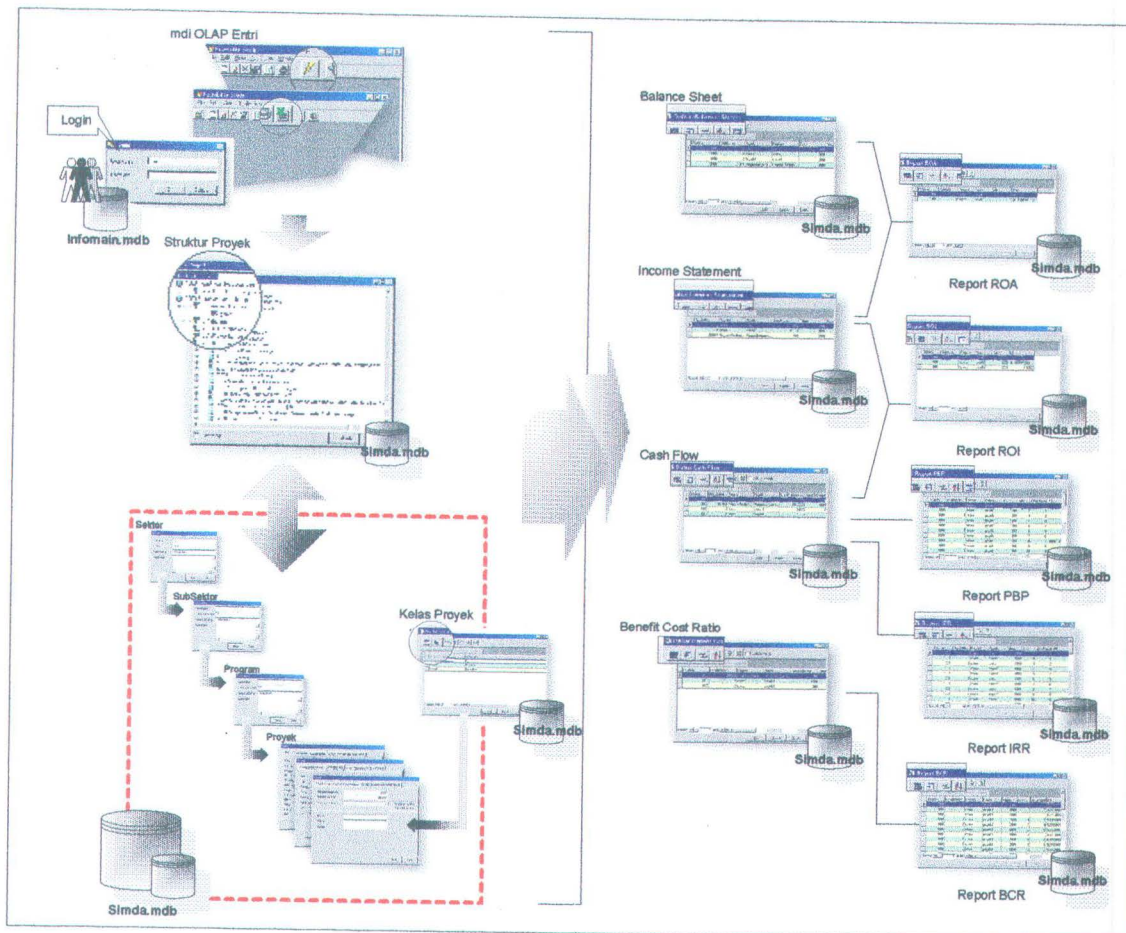


Gambar 5.13 Desain Antarmuka User Management

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.

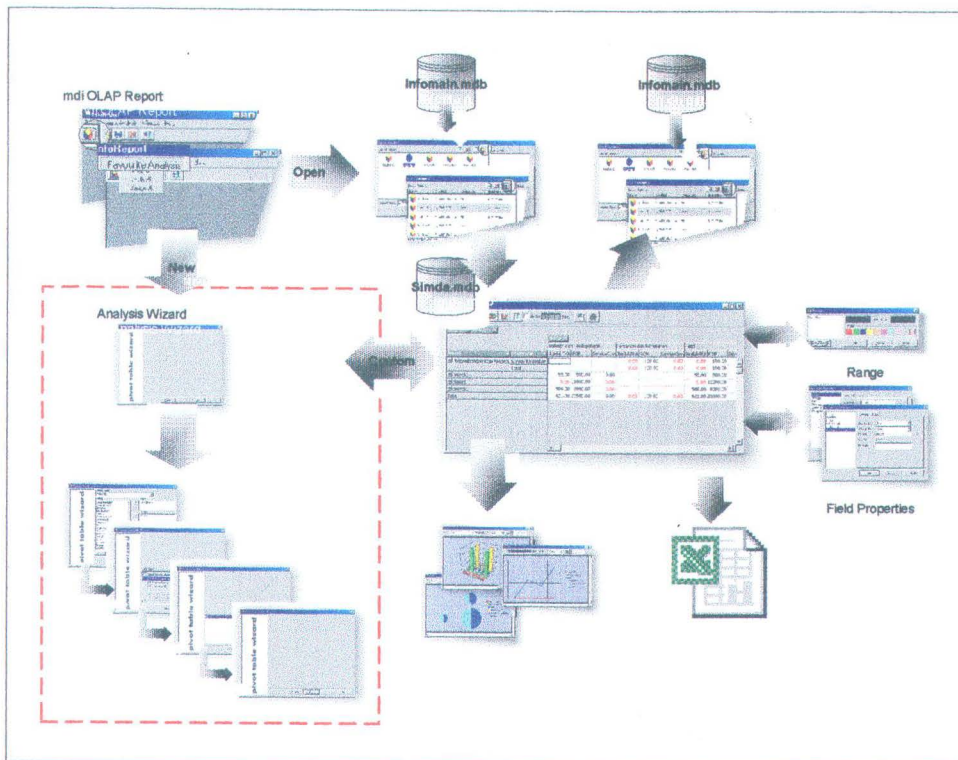


Gambar 5.14 Desain Antarmuka OLAP Entri

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 dokumen 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 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 mengisi *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.

The screenshot shows a Windows-style application window titled 'User Management'. It has three tabs: 'Data User' (selected), 'Departemen', and 'Group User'. The 'Data User' tab contains a table with columns: ID User, Nama, Description, Group, and Departemen. The table has two rows of data: 'ruly' (Ruly D, sekretaris, Grp001, 001) and 'yudy' (Yudy). A 'Password' dialog box is open in front of the table, with fields for 'Old Password', 'New Password' (masked with asterisks), and 'Confirm Password'. A red arrow points to the 'New Password' field with the label 'Enter Default Password'. The dialog box has 'Cancel' and 'OK' buttons at the bottom.

ID User	Nama	Description	Group	Departemen
ruly	Ruly D	sekretaris	Grp001	001
yudy	Yudy			

Gambar 6.1 Form Data User

Modul User Management

Tab Departemen

User Management

Data User | Departemen | Group User

Departemen

Drag a column header here to group by that column.

ID Departemen	Nama Departemen	Keterangan
001	Dinas Kesehatan	Kesehatan Masyarakat

Gambar 6.2 Form Departemen

Modul User Management

Tab Group User

User Management

Data User | Departemen | Group User

Group User

Drag a column header here to group by that column.

ID Group	Nama Group	Keterangan
Grp001	Keungan	emboh

Gambar 6.3 Form Group User

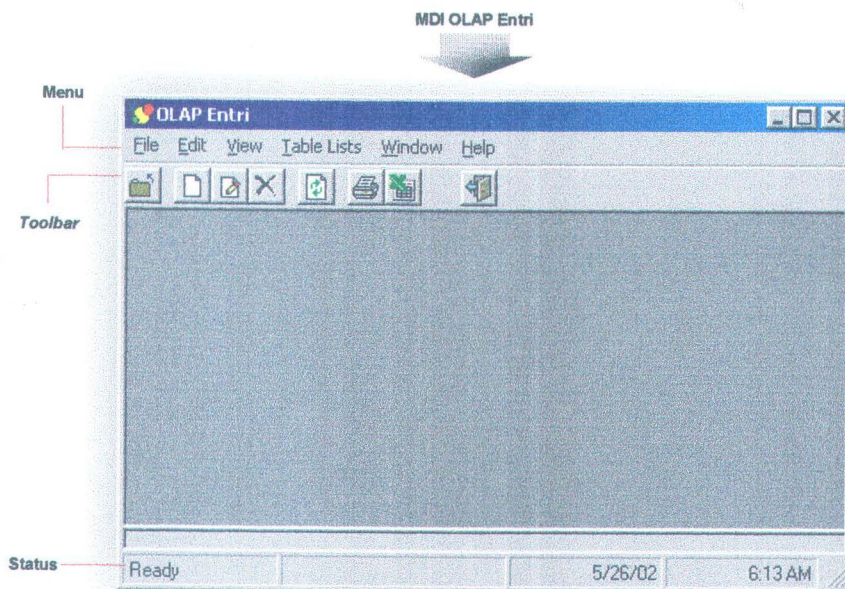
Form kedua merupakan dialog yang digunakan oleh Administrator untuk mengisi password awal seorang *user* saat mendaftarkannya pada sistem. Dalam pembuatan modul ini, informasi password tidak ditampilkan di tab Data User. Untuk merubah password seorang *user*, Admin harus melakukan *click* pada tombol 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**

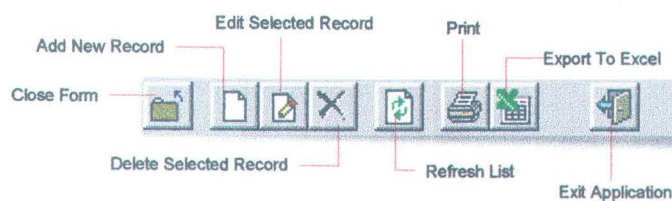
MDI OLAP Entri memiliki beberapa menu yang akan dirinci dalam tabel berikut:

Tabel 6.1 Menu-menu MDI Olap Entri

<i>Menu</i>	<i>List</i>	<i>Keterangan</i>
<i>File</i>	Print	Print dokumen
	Print Setup	Setup printer
	Export To Excel	Eksport dokumen ke Excel
	Close	Menutup <i>form</i> yang aktif
	Change Password	Merubah <i>password user</i>
	Database Setting	Mendefinisikan lokasi database
	Exit	Keluar aplikasi OLAP Entri
<i>Edit</i>	Add	Menambah record
	Edit	Mengedit data
	Delete	Menghapus record
	Refresh	Refresh data
<i>View</i>	ToolBar	Menampilkan ToolBar (<i>Yes/No</i>)
	StatusBar	Menampilkan StatusBar (<i>Yes/No</i>)
<i>Table List</i>	Struktur Proyek	<i>Add/Edit/Delete</i> Struktur Proyek
	Kelas Proyek	<i>Add/Edit/Delete</i> Kelas Proyek
	Balance Sheet	<i>Add/Edit/Delete</i> Balance Sheet
	Income Statement	<i>Add/Edit/Delete</i> Income Statement
	Benefit Cost	<i>Add/Edit/Delete</i> Benefit Cost
	Cash Flow	<i>Add/Edit/Delete</i> 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
<i>Window</i>	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

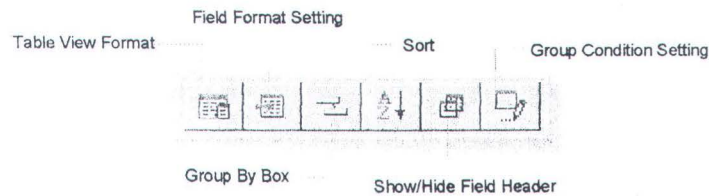
Gambar Icon pada MDI OLAP Entri akan digambarkan dan dijelaskan seperti pada Gambar 6.5 dan Tabel 6.2:

Tabel 6.2 Fungsi Icon-icon pada Toolbar MDI OLAP Entri

Nama Icon	Fungsi
<i>Close Form</i>	Menutup form yang aktif
<i>Add New Record</i>	Membuat Report baru
<i>Edit Selected Record</i>	Memodifikasi data yang diinginkan
<i>Delete Selected Record</i>	Menghapus data
<i>Refresh List</i>	Refresh data
<i>Print</i>	Mencetak data
<i>Export To Excel</i>	Mengirim data ke Excel
<i>Exit Application</i>	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.

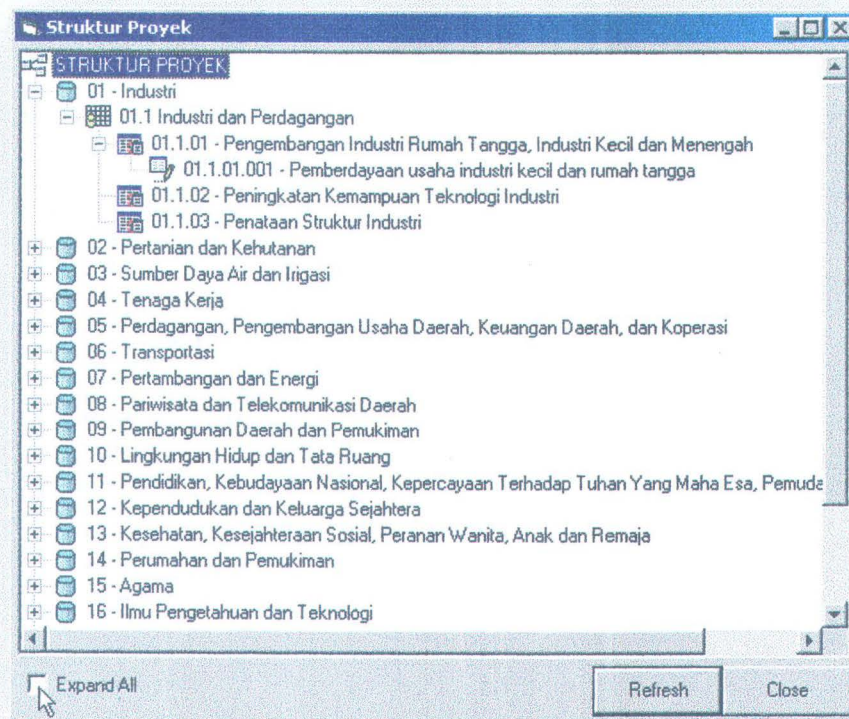
▪ Status

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*.





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)

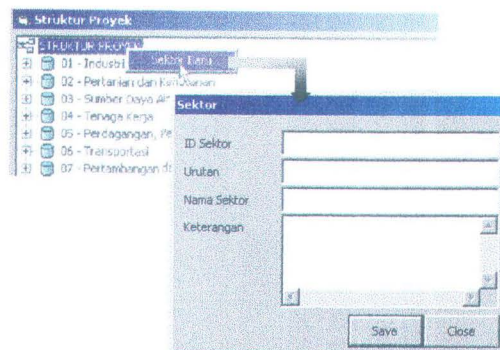
'//SELECT Proyek
rs4.Open "SELECT Proyek, Urutan FROM TbProyek _
WHERE IDSektor=" & rs1("IDSektor") & _
" AND IDSubSektor=" & rs2("IDSubSektor") _
& " AND Program=" & rs3("Program") & _
"", ConnMain, adOpenForwardOnly, _
adLockReadOnly
Do While Not rs4.EOF
    Set nodeX = _
TreeView1.Nodes.Add(rs1("IDSektor") & "#" & _
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

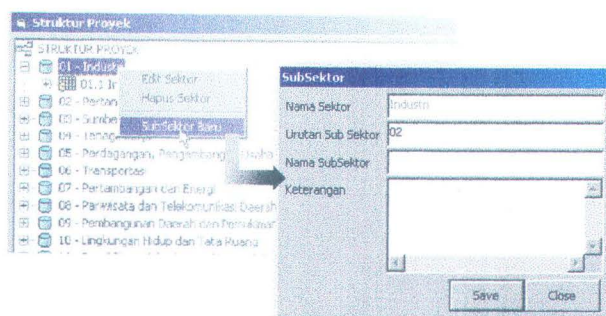
Add/Edit/Delete data Sektor dilakukan dengan *click* kanan pada *node* Struktur Proyek. Event *click* ini akan menampilkan form Sektor. Isian Urutan akan dipergunakan sebagai kode sebuah Sektor yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree*.



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.

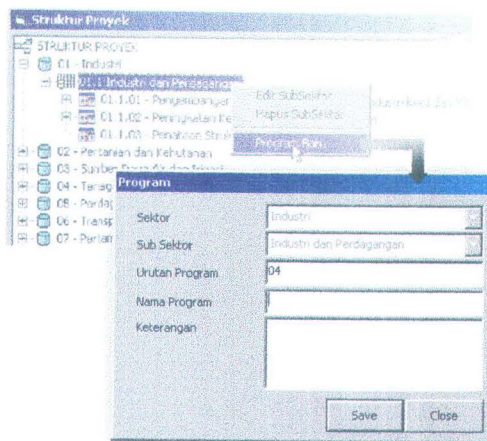


Gambar 6.10 Form Entri SubSektor

▪ 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/Delete 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:

The screenshot shows a window titled "Daftar Kelas Proyek". It contains a table with two columns: "KelasProyek" and "Keterangan". The table has five rows: "Baru", "Lanjutan", "Pensiapan", "Rehabilitasi", and "Studi Kelayakan". Overlaid on this window is a smaller dialog box titled "Kelas Proyek". This dialog box has two input fields: "KelasProyek" and "Keterangan". At the bottom of the dialog box are two buttons: "Save" and "Close".

Gambar 6.13 *Form* Entri Kelas Proyek

6.2.2.3 Pembuatan *Form* Entri Balance Sheet

The screenshot shows a window titled "Daftar Balance Sheet". It contains a table with six columns: "ID Sektor", "ID Sub Sektor", "Program", "Proyek", "Tahun", and "Tas". Overlaid on this window is a larger dialog box titled "Balance Sheet". This dialog box has several input fields: "Sektor", "Sub Sektor", "Program", "Proyek", and "Tahun" (with the value "2000" entered). Below these are two columns of financial items, each with an input field and a "0" value: "Kas", "Piutang", "Persediaan", "Fasilitas", "Aktivitas", "Total Assets" on the left, and "Utang Dagang", "Utang Bank", "Modal", "Saham", "Total Pasiva" on the right. At the bottom of the dialog box are two buttons: "Save" and "Close".

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.

$$\text{TotalAssets} = \text{Kas} + \text{Piutang} + \text{Persediaan} + \text{Fasilitas} + \text{AktivaLain}$$

$$\text{TotalPassiva} = \text{HutangDagang} + \text{HutangBank} + \text{Modal} + \text{Saham}$$

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.

$$\text{GrossIncome} = \text{Revenue} - \text{COGS}$$

$$\text{OperationIncome} = \text{GrossIncome} - (\text{SalesExpenses} + \text{GeneralAndAdm} + \text{Depreciation} + \text{Interest} + \text{Tax})$$

$$\text{NetIncome} = \text{OperationIncome} + \text{AddRevenue}$$

Gambar 6.15 *Form* Entri *Income Statement*

6.2.2.5 Pembuatan *Form* Entri Benefit Cost

Benefit Cost

Sektor:

Sub Sektor:

Program:

Proyek:

Tahun: 2000

Bulan	Benefit	Disbenefit	Cost
Januari	0	0	0
Pebruari	0	0	0
Maret	0	0	0
April	0	0	0
Mei	0	0	0
Juni	0	0	0
Juli	0	0	0
Agustus	0	0	0

Save Close

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

Cash Flow

Sektor:

Sub Sektor:

Program:

Proyek:

Tahun: 2000

Bulan	Investment	Change Working	Terminal Val.	Net Flow
Januari	0	0	0	0
Pebruari	0	0	0	0
Maret	0	0	0	0
April	0	0	0	0
Mei	0	0	0	0
Juni	0	0	0	0
Juli	0	0	0	0
Agustus	0	0	0	0

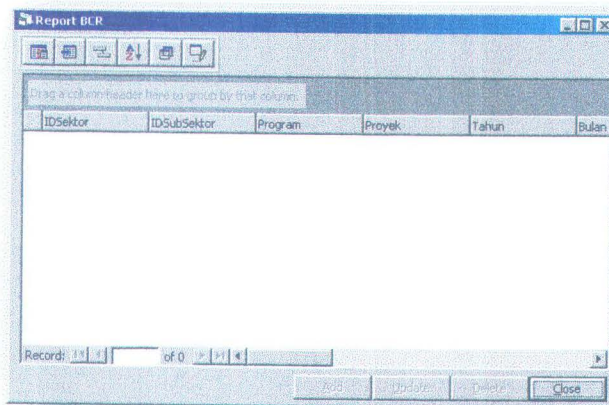
Save Close

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.

$$\text{NetFlow} = \text{Investment} - \text{ChangeWorking} + \text{TerminalVal.}$$

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
```

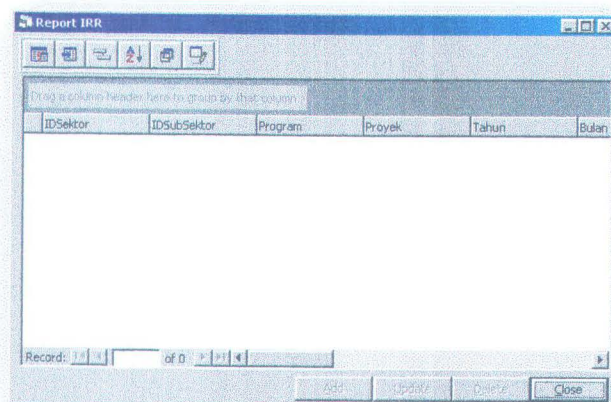
```
Exit Function
```

```
ElseIf NPVTotal > 0 Then
```

```

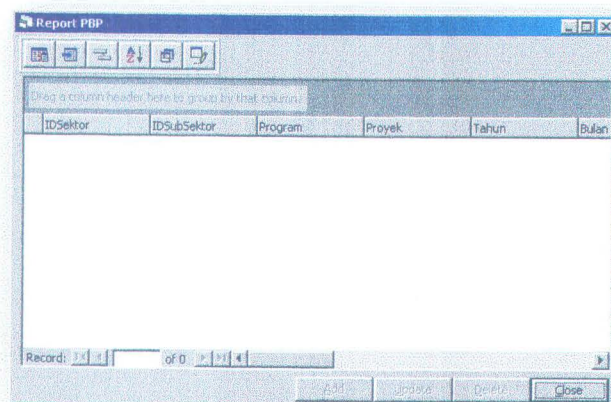
CalcIRR = IRR - ((NPVTotal * (IRR - IRRx)) / _
(NPVTotal - NPVTotalx))
Exit Function
ElseIf NPVTotal < 0 Then
    IRRx = IRR
    NPVTotalx = NPVTotal
    IRR = IRR + 1
    GoTo ULANG
End If
End Function

```



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

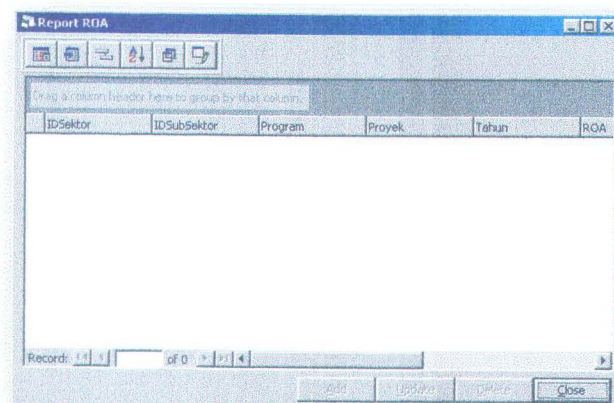
```

```

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 & ")"

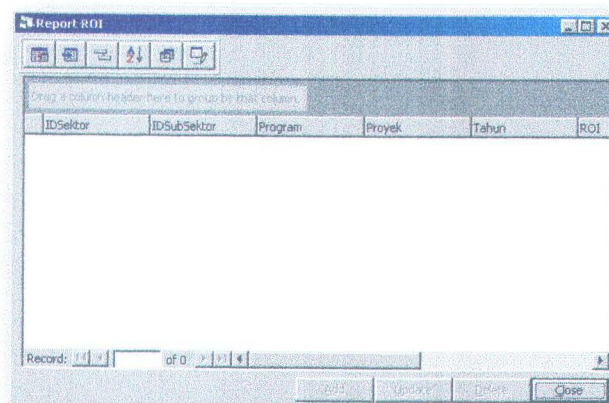
.MoveNext
Loop
End With

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
```

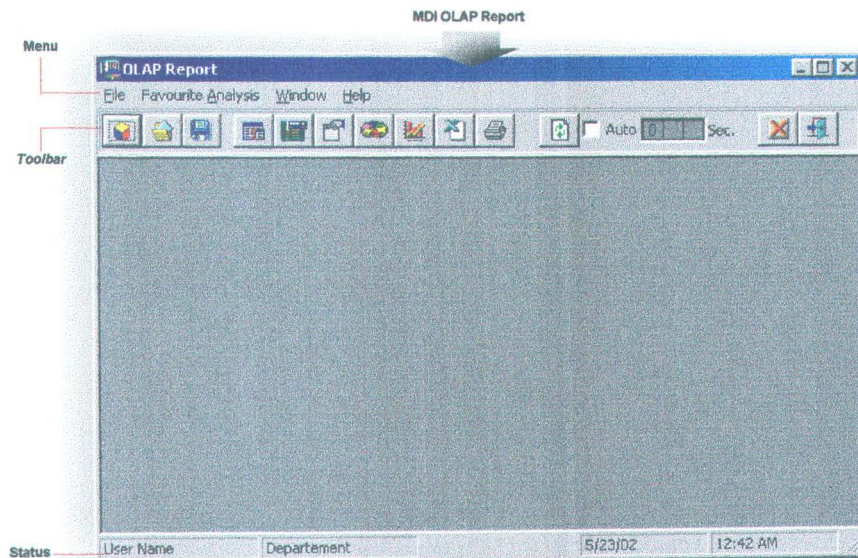
```
Set rs = Nothing
```

```
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

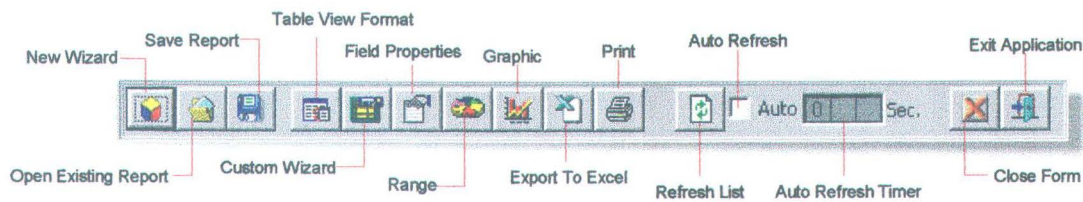
MDI OLAP Report memiliki beberapa menu yang akan dirinci dalam tabel berikut:

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 <i>form</i> yang aktif
	Save	Menyimpan dokumen analisis aktif
	Change Password	Mengganti <i>password user</i>
	Exit	Keluar aplikasi OLAP Report
Favourite Analysis	Default = <i>empty</i>	Berisi dokumen analisis yang telah di-save dengan status <i>Favourite Analysis</i> = 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:

```

...
ClQueryFramel.Item.MoveToRoot
If ClQueryFramel.Item.HasChildren Then
    ClQuery1.SQLInfo.TableQuote = clqQuoteBrackets
    ClQuery1.BuildSQL
    WhereClause = " FROM " & ClQuery1.ClauseFrom & " WHERE " & _
    ClQuery1.ClauseWhere
Else
    WhereClause = " FROM [" & mTableName & "]"
End If
...

```

Proses dilanjutkan dengan pembangunan konfigurasi field pada komponen visual tampilan OLAP (Dynamic Cube), seperti terlihat pada kode dibawah ini:

```

...
With mCube
    .AutoDataRefresh = False
    .CursorStyle = DCLight
    .DCCConnectType = DCCT_UNBOUND
    .DCCConnect = 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 DynamicCubeLibCtl.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:

```

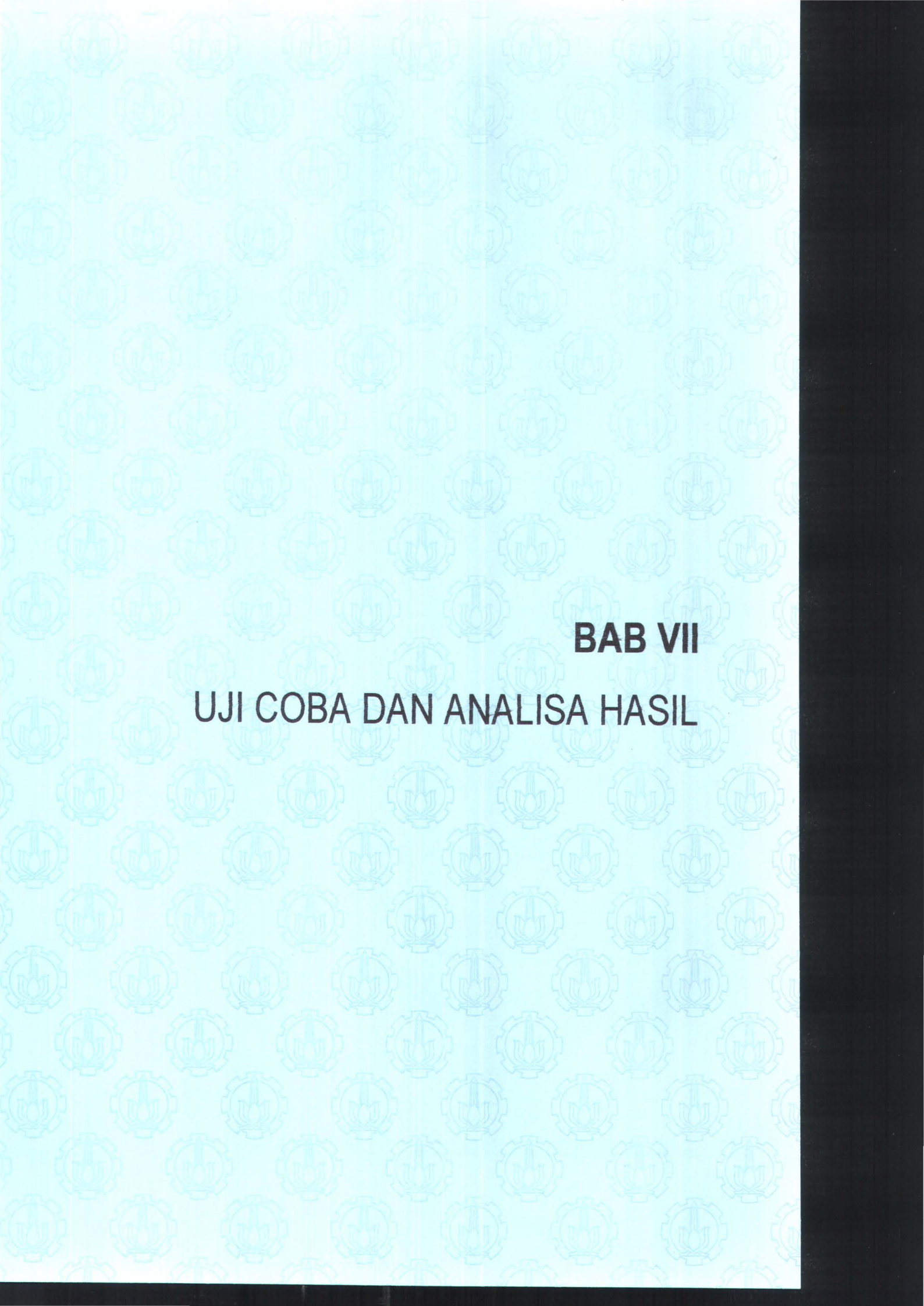
...
For Each rng In Ranges
    If cellAttrib.value < rng.EndRange And cellAttrib.value >= _
        rng.StartRange Then
        cellAttrib.ForeColor = rng.Color
        Exit Sub
    End If
Next
...

```

6.3 Join-Query View

Penggunaan struktur snowflake pada basisdata OLAP entri menyebabkan adanya sebuah kebutuhan untuk membuat beberapa *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 variasi 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

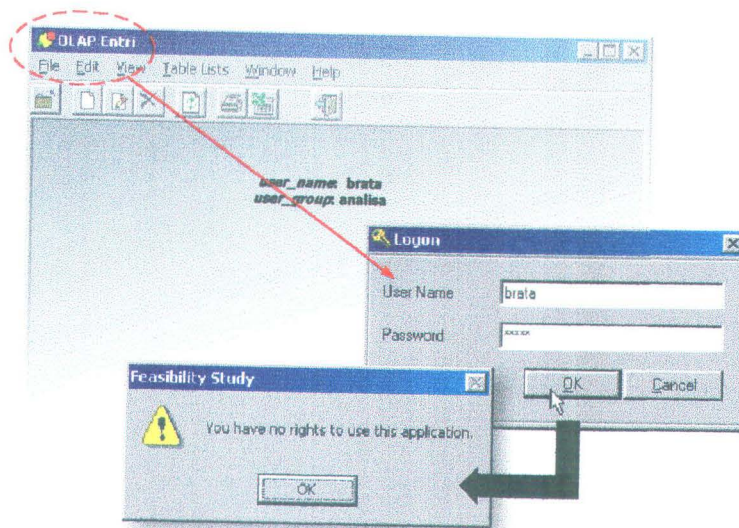
- AMD Duron 1 GHz
- RAM 128 MB
- 20 GB harddisk

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).

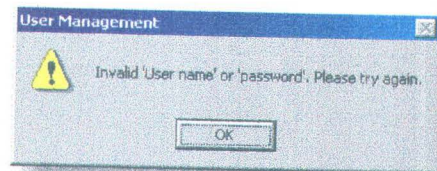
7.2.1 Uji Coba Login

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).

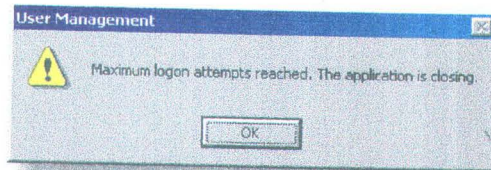


Gambar 7.1 Peringatan *user group* tidak memiliki hak akses terhadap sebuah modul

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



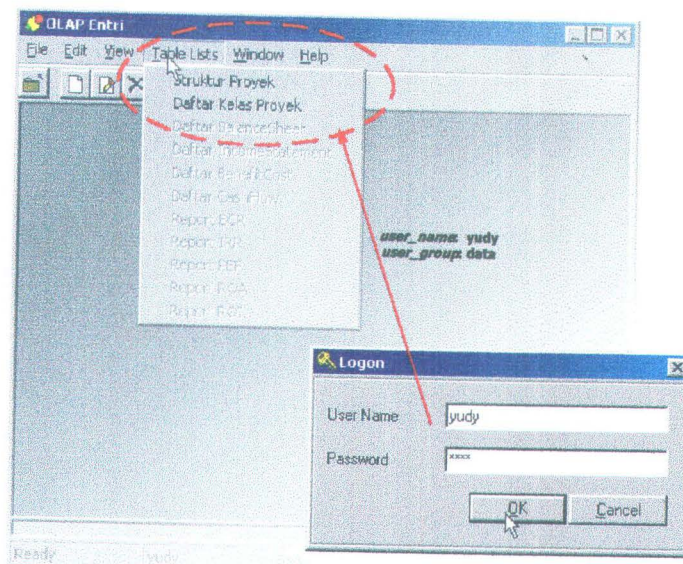
Gambar 7.3 Peringatan penutupan aplikasi segera oleh sistem

7.2.2 Uji Coba *User Management*

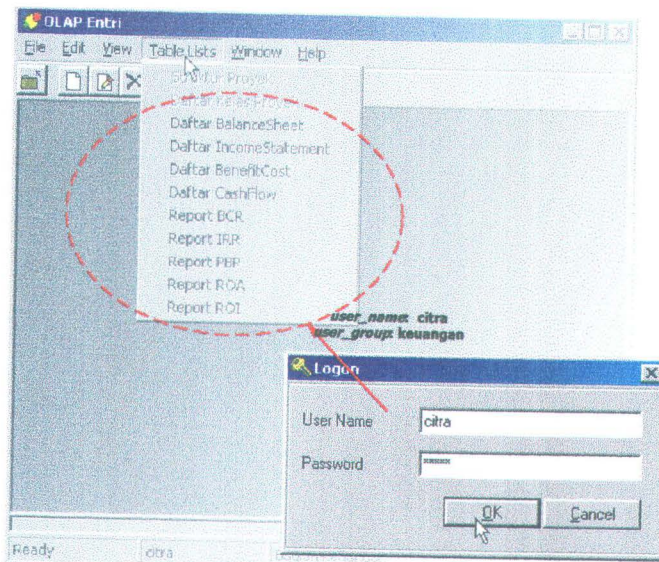
Uji coba ini dilakukan untuk menguji modul *User Management* yang telah dibuat. Modul *user management* ini hanya dapat diakses oleh *user group* Admin. Pengujian yang dilakukan meliputi penambahan *user* baru, *edit data user* dan penghapusan hak akses *user*. Seorang *user* bisa didaftarkan untuk memiliki 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.

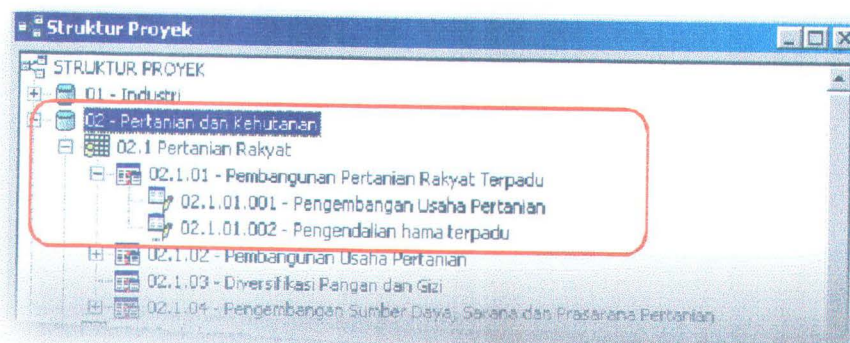
Penggambaran menu OLAP Entri yang dimanajemen berdasarkan hak akses group user, digambarkan pada gambar 7.4 dan 7.5.

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

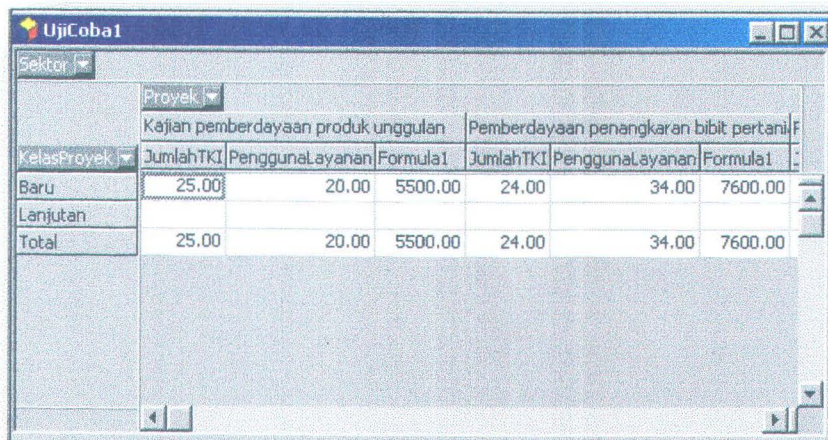
Custom Field:

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.



KelasProyek	JumlahTKI	PenggunaLayanan	Formula1	JumlahTKI	PenggunaLayanan	Formula1
Baru	25.00	20.00	5500.00	24.00	34.00	7600.00
Lanjutan						
Total	25.00	20.00	5500.00	24.00	34.00	7600.00

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

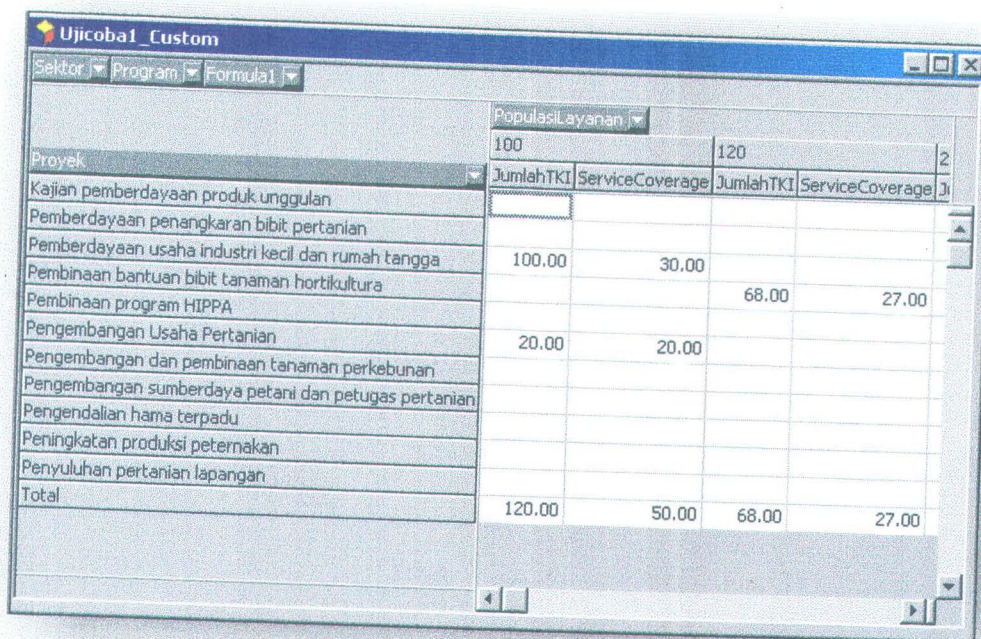
Custom Field:

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.



The screenshot shows a window titled "Ujicoba1_Custom" with a menu bar (Sektor, Program, Formula1) and a toolbar. Below the menu bar is a table with the following data:

Proyek	PopulasiLayanan		JumlahTKI		ServiceCoverage
	100	120	100	120	
Kajian pemberdayaan produk unggulan					
Pemberdayaan penangkaran bibit pertanian					
Pemberdayaan usaha industri kecil dan rumah tangga	100.00		30.00		
Pembinaan bantuan bibit tanaman hortikultura				68.00	27.00
Pembinaan program HIPPA					
Pengembangan Usaha Pertanian	20.00		20.00		
Pengembangan dan pembinaan tanaman perkebunan					
Pengembangan sumberdaya petani dan petugas pertanian					
Pengendalian hama terpadu					
Peningkatan produksi peternakan					
Penyuluhan pertanian lapangan					
Total	120.00		50.00	68.00	27.00

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* $0 \leq \text{Range} < 25$. Pemberian warna biru akan dilakukan untuk data numerik yang memenuhi konfigurasi *range* $30 \leq \text{Range} < 70$. Dari hasil proses range ini, maka tampilan dokumen analisis **Ujicoba1_Custom** akan berubah menjadi seperti gambar 7.9:

The screenshot shows a window titled "Ujicoba1_Custom" with a menu bar containing "Sektor", "Program", and "Formula1". Below the menu bar is a table with the following data:

Proyek	PopulasiLayanan		JumlahTKI		ServiceCoverage	
	100	120	100	120	100	120
Kajian pemberdayaan produk unggulan						
Pemberdayaan penangkaran bibit pertanian						
Pemberdayaan usaha industri kecil dan rumah tangga	100.00		30.00			
Pembinaan bantuan bibit tanaman hortikultura				68.00		27.00
Pembinaan program HIPPA						
Pengembangan Usaha Pertanian	20.00		20.00			
Pengembangan dan pembinaan tanaman perkebunan						
Pengembangan sumberdaya petani dan petugas pertanian						
Pengendalian hama terpadu						
Peningkatan produksi peternakan						
Penyuluhan pertanian lapangan						
Total	120.00		50.00	68.00		27.00

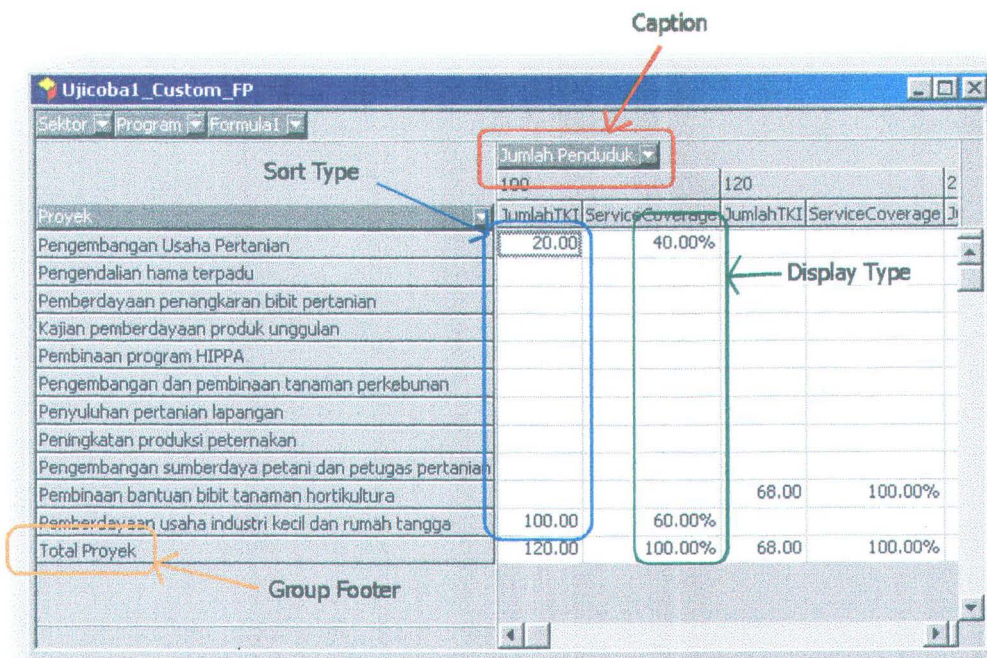
Gambar 7.9 Hasil Uji Coba Manipulasi dengan Range

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.

Sektor	JumlahTKI	ServiceCoverage
Industri	125.00	66.00
Pertanian dan Kehutanan	331.00	379.00
Sumber Daya Air dan Irigasi	26.00	60.00
Total	482.00	505.00

Gambar 7.11 Pemilihan Sel Data Grafik



Gambar 7.12 Hasil Uji Coba Penggambaran Grafik

Uji Coba Export Data ke Excel Workbook

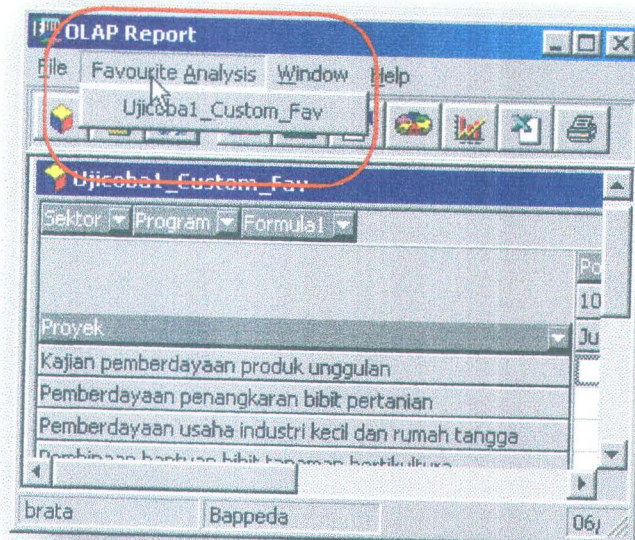
Microsoft Excel - Book1							
File Edit View Insert Format Tools Data Window Help							
Arial 10 B I U [Formatting Icons] 100% [PivotTable]							
	A	B	C	D	E	F	G
1	Formula1	(All)	(All)				
2	Program	(All)					
3	Sektor	(All)					
4							
5			Pop				
6	Proyek	Data	43	50	54	55	66
7	Kajian pemberdayaan 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*.



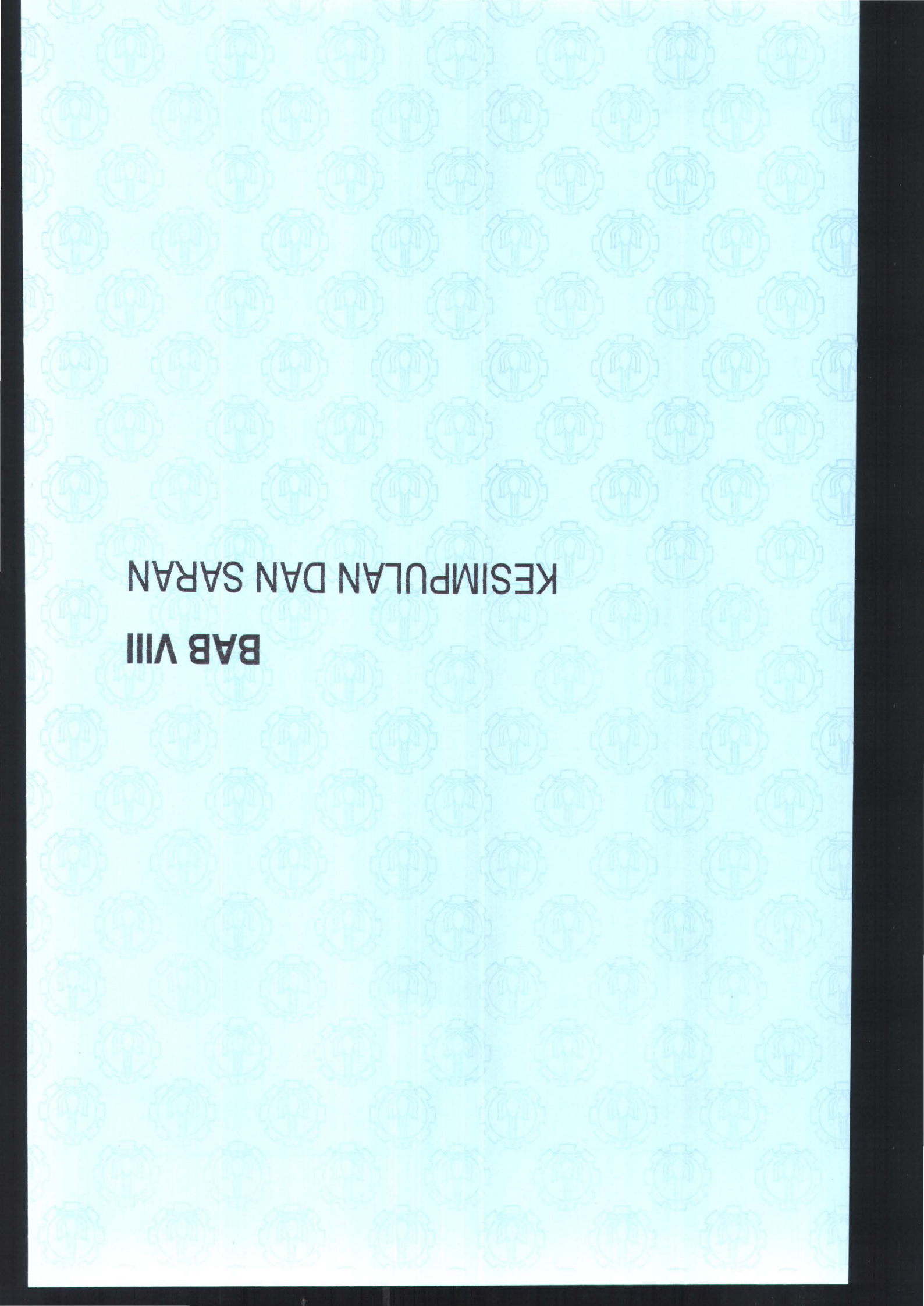
Gambar 7.14 Hasil Uji Coba Penyimpanan Dokumen Favorite Analysis

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.



KESIMPULAN DAN SARAN

BAB VIII

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

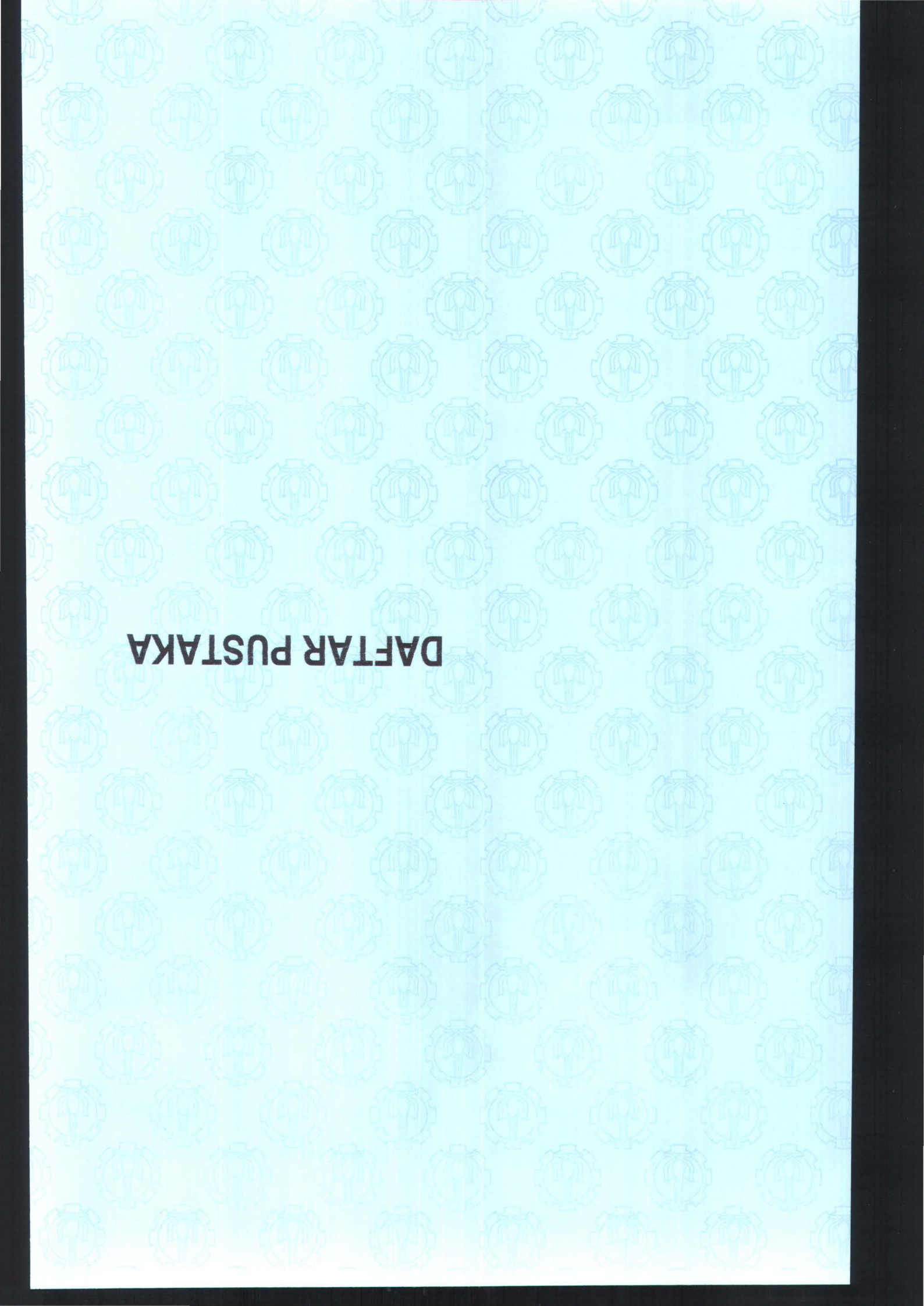
Saran bagi pengembangan aplikasi *Sistem Feasibility Study* yang menerapkan teknologi OLAP untuk proses investasi yang dikerjakan pada pembuatan tugas akhir ini antara lain adalah:



- 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 menggunakan 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](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

LAMP IRAN

LAMPIRAN

LAMPIRAN A

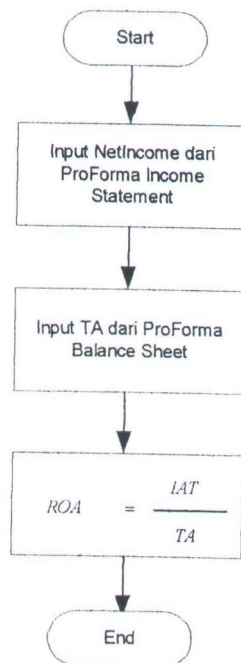
Desain Algoritma Perhitungan Indikator Keuangan

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

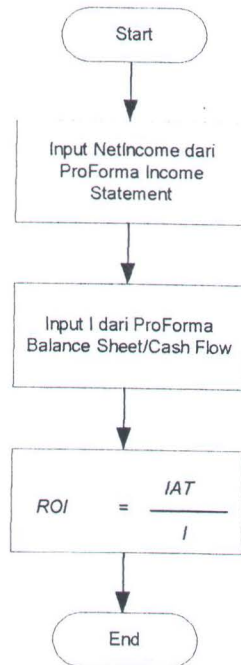
A.2 Algoritma Perhitungan ROI

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 :

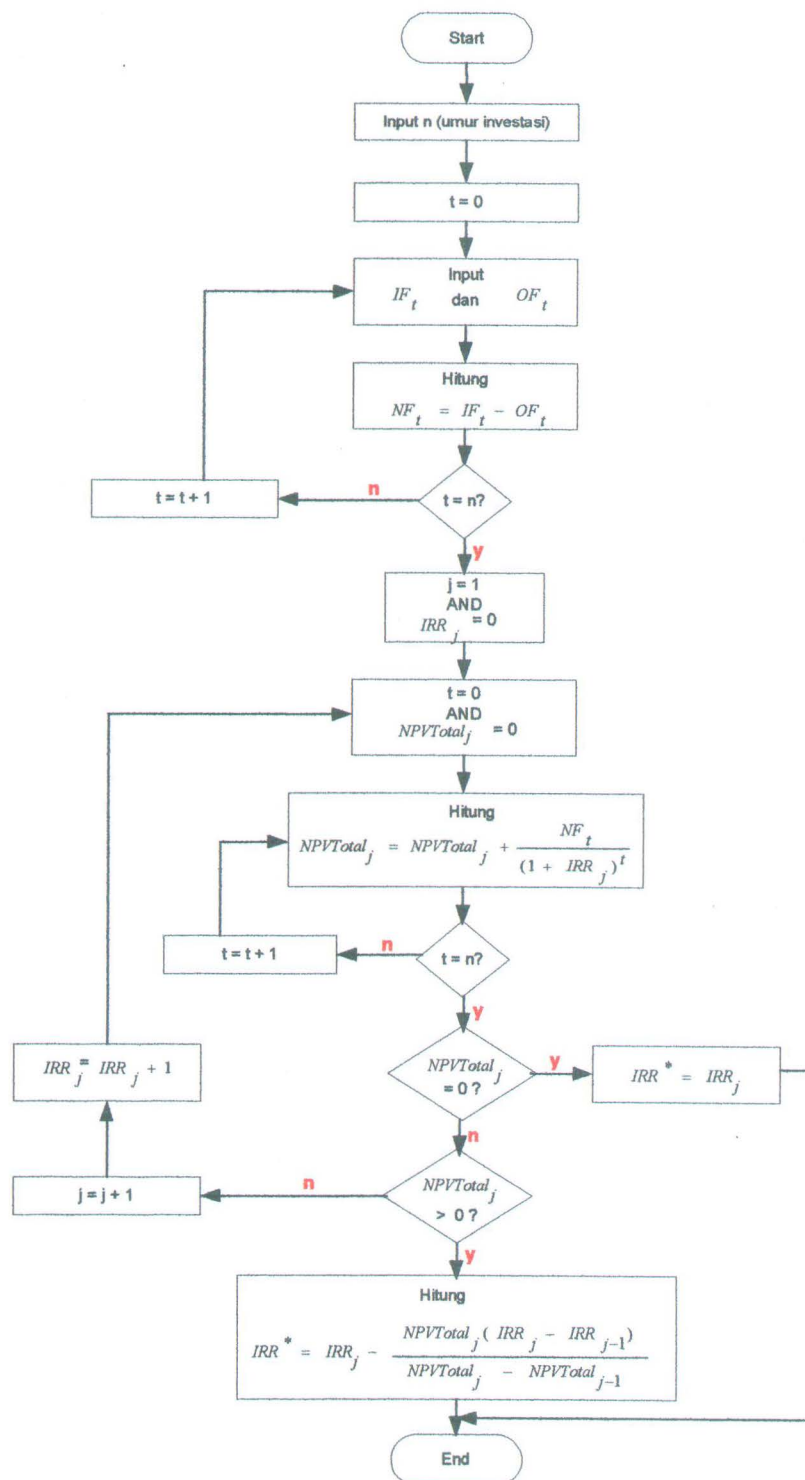
- 1) Masukkan umur proyek (n) dalam satuan bulan.
- 2) Masukkan Inflow (IF) dan Outflow (OF) untuk setiap bulannya sepanjang umur proyek tersebut .
- 3) Hitung Netflow (NF_t) untuk setiap bulannya sepanjang umur proyek tersebut.
Mulai dari $t = 0$ sampai dengan n, hitung

$$NF_t = IF_t - OF_t$$

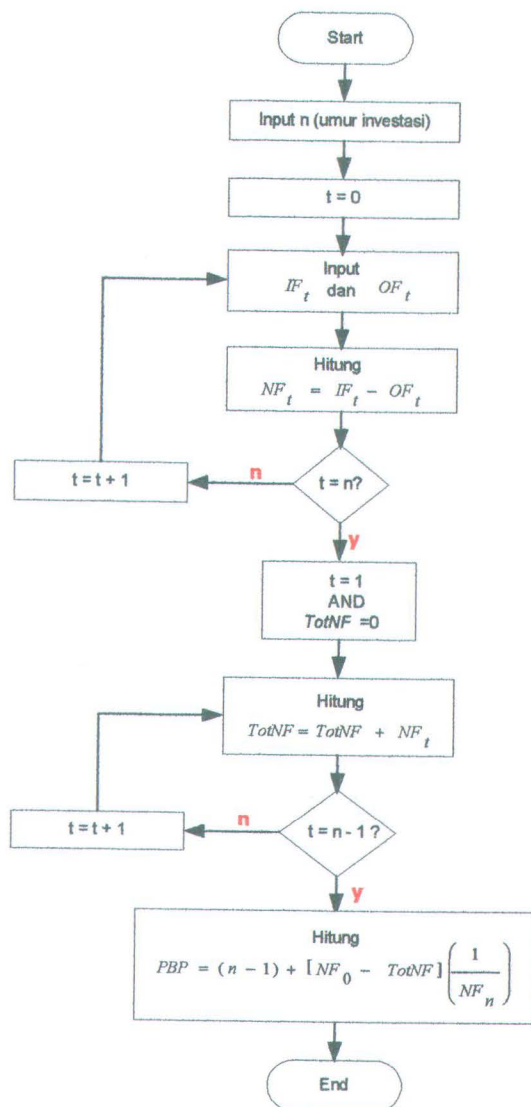
Biaya investasi yang pertama adalah NF_0 .

- 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.

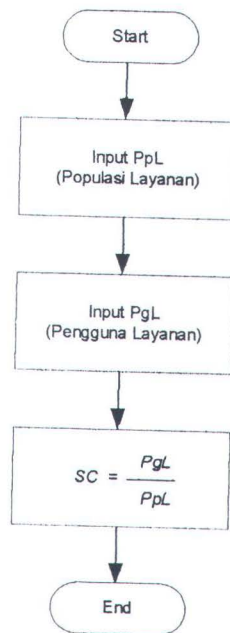
Nilai PL ini akan sangat tergantung pada : jenis layanan, dan wilayahnya. Misalkan: untuk investasi bangunan SD Inpres di kecamatan A, maka yang disebut populasi layanan (PpL) adalah jumlah anak yang berumur dari 7 – 12 tahun di kecamatan A.

- 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:

$$IPVTotal_j = \sum_{t=0}^n \frac{IF_t}{(1 + ROR)^t}$$

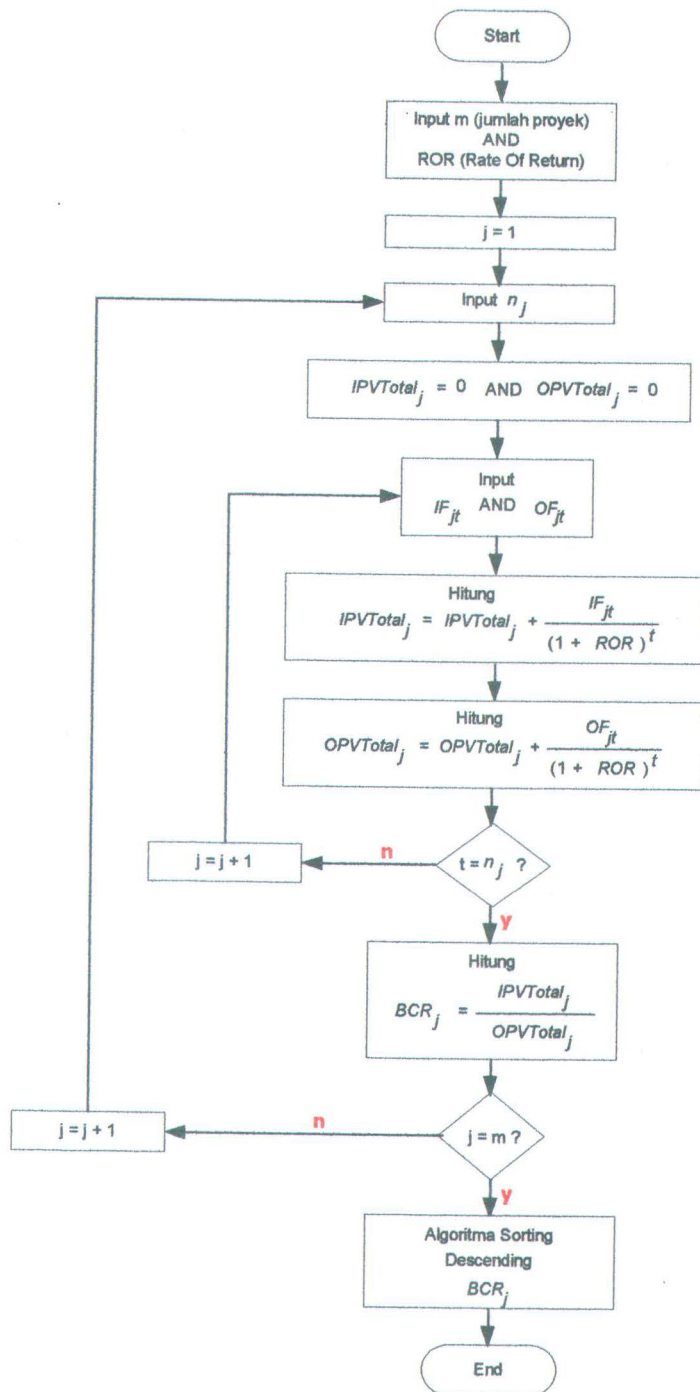
- 6) Lakukan perhitungan juga Total Nilai Sekarang atau Outflow Present Value Total (OPVTotal) untuk Outflow proyek j . Perhitungan $OPVTotal_j$ adalah:

$$OPVTotal_j = \sum_{t=0}^n \frac{OF_{jt}}{(1 + ROR)^t}$$

- 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].[PeggunaLayanan],
 [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].[PeggunaLayanan],
 [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] = [TbPBP].[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
<i>02.1</i>	<i>Pertanian rakyat</i>
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
<i>02.2</i>	<i>Perkebunan</i>
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
<i>02.3</i>	<i>Peternakan</i>
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
<i>02.4</i>	<i>Perikanan</i>
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
<i>02.5</i>	<i>Kehutanan</i>
02.5.01	Pembangunan dan pembinaan kehutanan
02.5.02	Pengembangan usaha hutan rakyat
03	Sumberdaya air dan irigrasi
<i>03.1</i>	<i>Pengembangan sumber daya air</i>
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
<i>03.2</i>	<i>Irigrasi</i>
03.2.01	Pengembangan dan pengelolaan jaringan irigrasi
04	Tenaga Kerja
<i>04.1</i>	<i>Tenaga kerja</i>
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

<i>05.1</i>	<i>Perdagangan</i>
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
<i>05.2</i>	<i>Pengembangan Usaha Daerah</i>
05.2.01	Pengembangan usaha golongan ekonomi lemah
05.2.02	Pengembangan penanaman modal
05.2.03	Penyertaan modal pemerintah
<i>05.3</i>	<i>Keuangan daerah</i>
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
<i>05.4</i>	<i>Koperasi dan Pengusaha Kecil</i>
05.4.01	Pembinaan dan Pengembangan Koperasi
05.4.02	Pembinaan dan Pengembangan Usaha Kecil
06	Transportasi
<i>06.1</i>	<i>Prasarana Jalan</i>
06.1.01	Rehabilitasi dan Pemeliharaan Jalan dan Jembatan
06.1.02	Peningkatan jalan dan penggantian jembatan
06.1.03	Pembangunan jalan dan jembatan
<i>06.2</i>	<i>Transportasi Darat</i>
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
<i>07.1</i>	<i>Pertambangan rakyat</i>
07.1.01	Pembangunan pertambangan
07.1.02	Pembangunan Sistem Informasi Geologi/Geografi
<i>07.2</i>	<i>Energi/Listrik Pedesaan</i>
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
<i>08.1</i>	<i>Pariwisata</i>
08.1.01	Pengembangan Pariwisata
08.1.02	Pengembangan dan Pengelolaan produk/obyek wisata
<i>08.2</i>	<i>Telekomunikasi Daerah</i>
08.2.01	Pengembangan telekomunikasi daerah

09	Pembangunan daerah dan pemukiman
<i>09.1</i>	<i>Pembangunan daerah bawahan</i>
09.1.01	Pembangunan daerah tingkat II
09.1.02	Pembangunan kawasan khusus
<i>09.2</i>	<i>Pembangunan kota</i>
09.2.01	Pembangunan prasarana kota
<i>09.3</i>	<i>Pembangunan desa</i>
09.3.01	Pembangunan desa
<i>09.4</i>	<i>Pemukiman</i>
09.4.01	Pemukiman dan Lingkungan
09.4.01	Penggerakan dan Pembinaan Transmigran
10	Lingkungan Hidup dan Tata ruang
<i>10.1</i>	<i>Lingkungan Hidup</i>
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
<i>10.2</i>	<i>Tata Ruang</i>
10.2.01	Penataan ruang
10.2.02	Pertanahan
11	Pendidikan kebudayaan nasional, Kepercayaan terhadap Tuhan YME, Pemuda dan Olahraga
<i>11.1</i>	<i>Pendidikan</i>
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
<i>11.2</i>	<i>Pendidikan luar sekolah dan kedisiplinan</i>
11.2.01	Luar sekolah
<i>11.3</i>	<i>Kebudayaan daerah dan kepercayaan terhadap Tuhan Yang Maha Esa</i>
11.3.01	Inventarisasi dan pembinaan nilai-nilai budaya
11.3.02	Pembinaan kebahasaan, kesusastraan 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
<i>11.4</i>	<i>Pemuda dan Olah Raga</i>
11.4.01	Pembinaan dan pengembangan kepemudaan
11.4.02	Pembinaan Keolahragaan
12	Kependudukan dan keluarga sejahtera

12.1	<i>Kependudukan dan keluarga berencana</i>
12.1.01	Kependudukan
12.1.02	Keluarga berencana
13	Kesehatan, kesejahteraan sosial, peranan wanita, anak dan remaja
13.1	<i>Kesehatan</i>
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	<i>Kesejahteraan sosial</i>
13.2.01	Pembinaan kesejahteraan sosial
13.2.02	Pelayanan dan rehabilitasi sosial
13.2.03	Pembinaan partisipasi sosial masyarakat
13.2.04	Penanggulangan bencana alam
13.3	<i>Peranan wanita, anak dan remaja</i>
13.3.01	Peningkatan peranan wanita dan PKK
13.3.02	Pembinaan anak dan remaja
14	Perumahan dan pemukiman
14.1	<i>Perumahan dan pemukiman</i>
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	<i>Penataan kota dan bangunan</i>
14.2.01	Penataan Kota
14.2.02	Penataan Bangunan
15	Agama
15.1	<i>Pelayanan kehidupan beragama</i>
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	<i>Pembinaan pendidikan agama</i>
15.2.01	Pembinaan pendidikan agama
16	Ilmu pengetahuan dan teknologi
16.1	<i>Ilmu pengetahuan dan teknologi</i>
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	<i>Tenik produksi dan teknologi</i>
16.2.01	Penerapan dan pengembangan teknik produksi
16.2.02	Penguasaan alih teknologi
16.2.03	Penerapan dan pengembangan model-model pemnagunan di daerah
16.3	<i>Penelitian</i>
16.3.01	Penelitian pendukung perencanaan pembangunan
16.3.02	Perancangan pembangunan daerah
16.3.03	Pembinaan kelembagaan ilmu pengetahuan dan teknologi
16.4	<i>Kelautan</i>
16.4.01	Inventarisasi dan Evaluasi Potensi kelautan
16.4.02	Pemanfaatan sumber daya kelautan
16.5	<i>Sistem Informasi dan statistik</i>
16.5.01	Pengembangan sistem informasi
16.5.02	Penyempurnaan dan pengembangan statistik
17	Hukum
17.1	<i>Hukum</i>
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	<i>Aparatur pemerintah</i>
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	<i>Pendayagunaan sistem dan pelaksanaan pengawasan</i>
18.2.01	Pendayagunaan sistem pengawasan
18.2.02	Peningkatan efektivitas pelaksanaan pengawasan
19	Politik, penerangan, komunikasi dan media massa
19.1	<i>Politik</i>
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	<i>Penerangan, komunikasi dan media massa</i>
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	<i>Keamanan dan ketertiban hukum</i>
20.1.01	Pembinaan kekuatan rakyat dan bela negara
20.1.02	Pembinaan ketentraman dan ketertiban
A	Bantuan pembangunan kepada daerah bawahan
A.1	<i>Bantuan pembangunan Dati I</i>
A.1.01	Subsidi Bantuan pembangunan Dati I