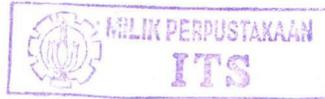


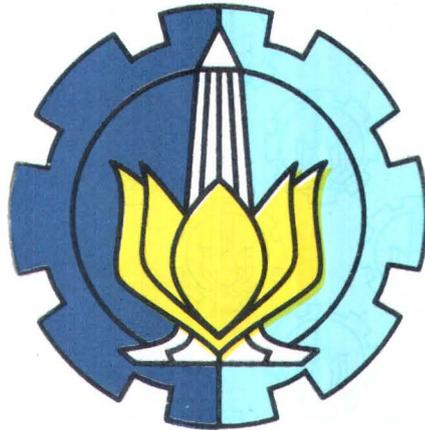
17.564 /H/03



PERANCANGAN DAN PEMBUATAN SISTEM OLAP UNTUK ANALISIS STUDI KELAYAKAN PROYEK

Studi Kasus Pemerintah Kabupaten Kediri

TUGAS AKHIR



R514
005.1
Dam
P-1

2002

Oleh :

RULY DAMARATRI
5198.100.079

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2002

PERPUSTAKAAN ITS	
Tgl. Terima	13-8-2002
Terima	14
No. Agenda Exp.	215 809

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

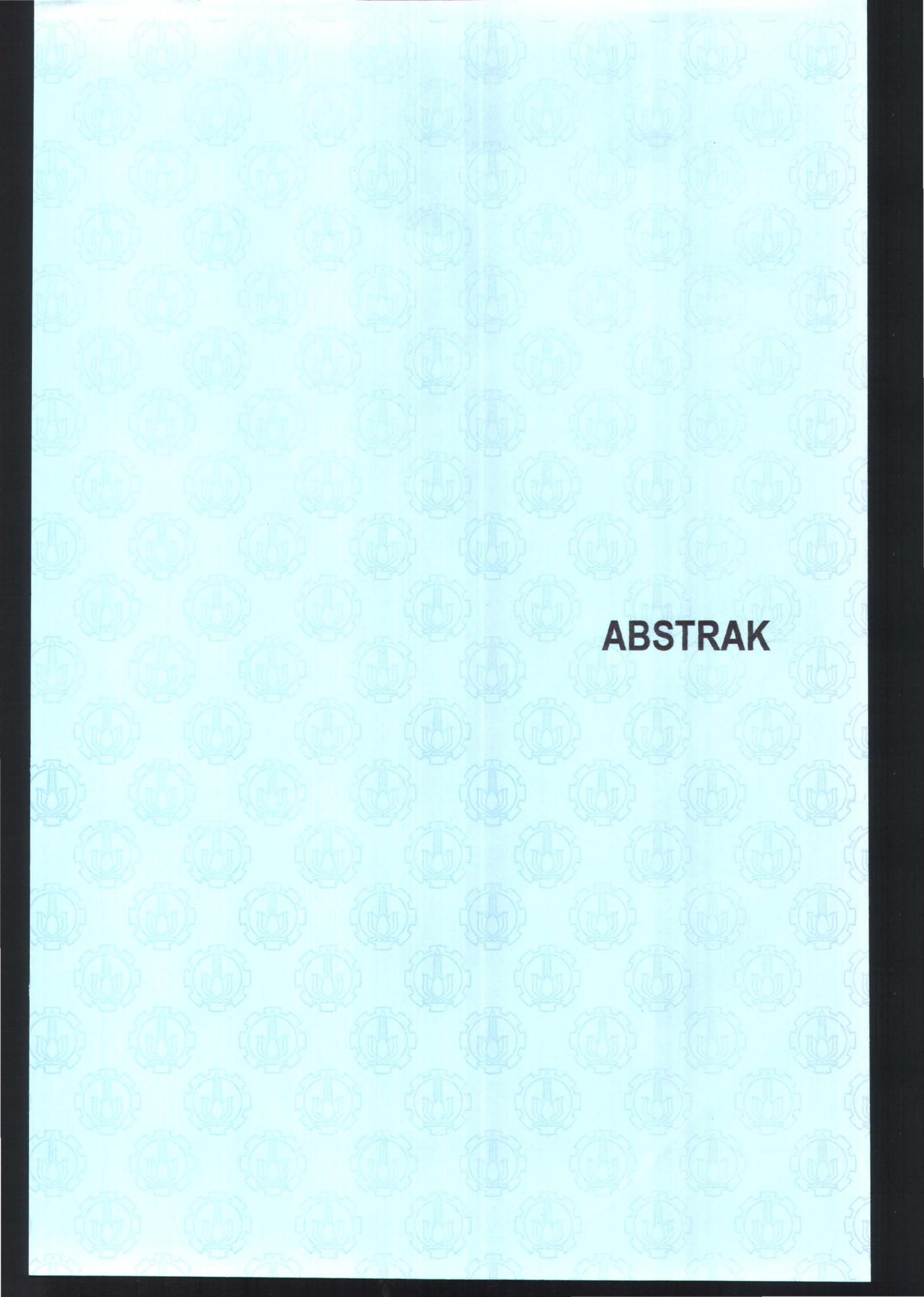
Pembimbing I
Dosen

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



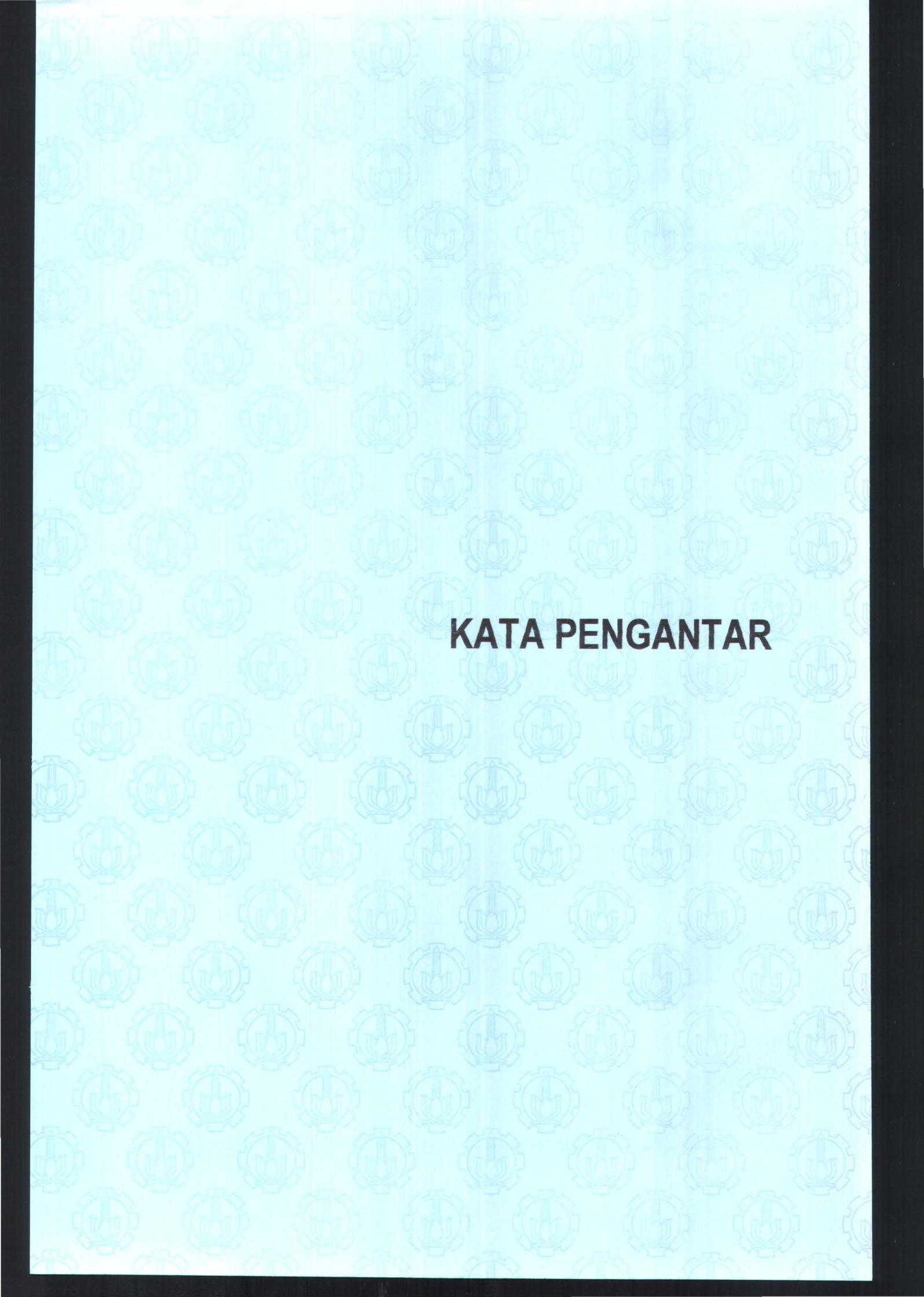
ABSTRAK

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

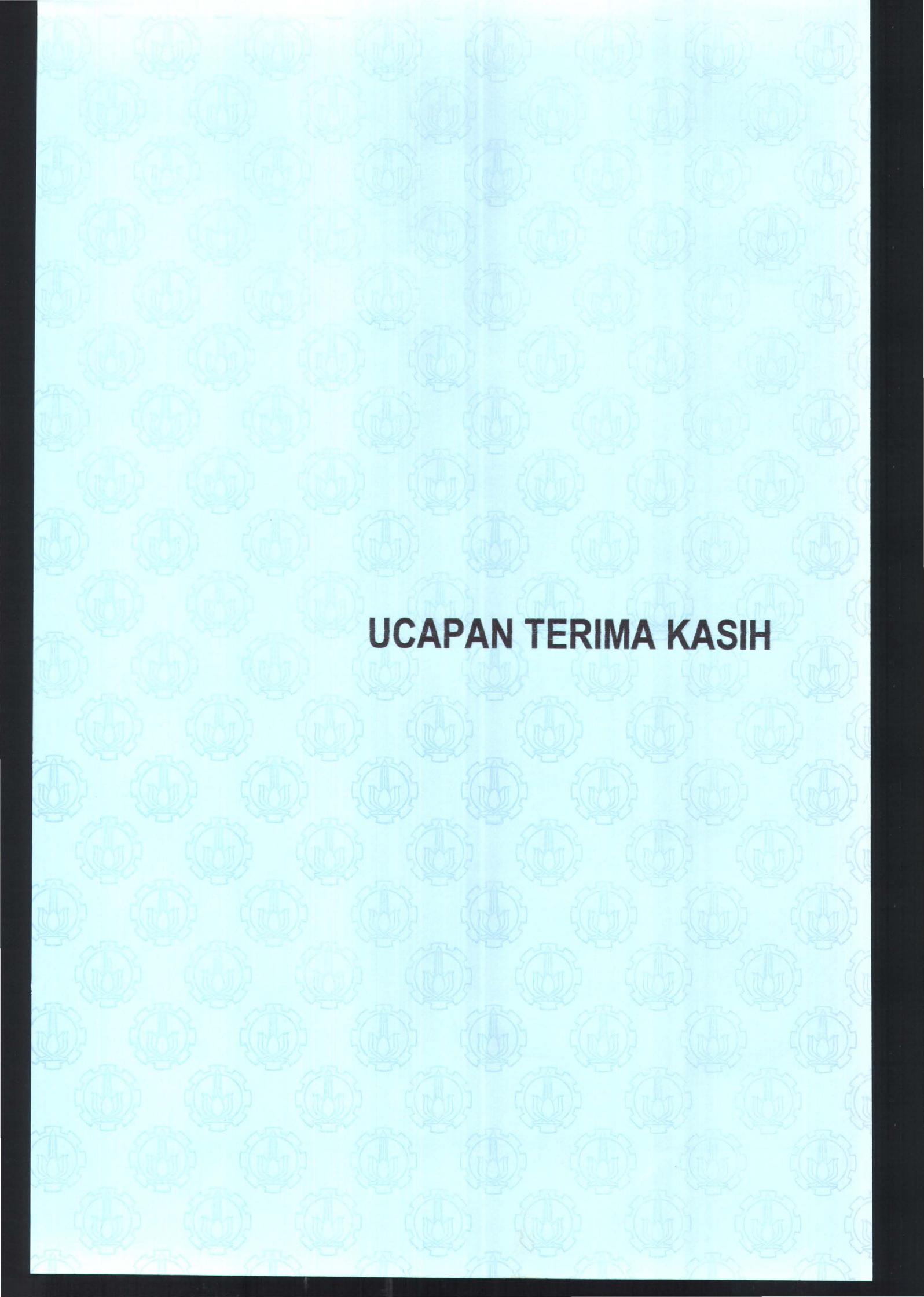
Segala puji syukur ke hadirat Tuhan Yang Maha Esa, Maha Kasih dan Penyayang yang telah memberikan berkat dan rahmat, serta kasih setiaNya, sehingga penulis telah dapat menyelesaikan penyusunan tugas akhir ini, dengan judul *“Perancangan dan Pembuatan Sistem OLAP untuk Analisa Studi Kelayakan Proyek Pemerintah 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 mujizat 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 kuatir. *God Bless Us!*
6. Agus Zainal SKom, MKom, selaku Ketua Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember

7. Bapak Dr.Ir. Joko Lianto Buliali,M.Sc serta Bapak Sugeng Sundoro, MM, selaku pembimbing yang telah sangat banyak membantu memberikan saran dan masukan dalam penyelesaian Tugas Akhir ini. Terima kasih Pak atas bantuan dan dukungannya.
8. Bapak Dwi Sunaryono, SKom selaku dosen wali yang telah memberikan nasehat, semangat dan dorongan kepada penulis sehingga penulis dapat menyelesaikan studinya dengan cepat.
9. Seluruh Dosen-dosen Teknik Informatika yang tidak dapat penulis sebutkan satu-persatu. Terima kasih atas bimbingan dan ilmunya selama masa perkuliahan.
10. Seluruh staf dan karyawan Teknik Informatika: Mas Yudi, Mas Sugeng, Mas Sholeh, Mbak Irna, Mbak Davi dan yang lainnya, yang telah membantu dalam kemudahan administrasi.
11. Mas Ilham yang telah meluangkan waktunya untuk membantu penulis dalam pengambilan data dan pencarian informasi di Bappeda Kediri.
12. Seluruh staf Bappeda yang telah banyak direpotkan selama pengerjaan tugas akhir ini.
13. Bang Romeo dan Mbak Betty yang sudah sangat membantu dalam memberikan “referensi” dalam pengerjaan tugas akhir ini. Minta maaf kalo penulis sudah terlalu merepotkan. Semoga sukses selalu.
14. Semua “Philean – PhiLeo Chorale” atas kebersamaanya, kegilaan, semangat dan doanya serta pelayanan-pelayanan yang telah ada yang membuat penulis menjadi bersemangat melewati semua masalah-masalah.

15. Rekan-rekan serta seluruh staf Indonesian Airlines yang ikut memberikan dukungan serta semangat. Terima kasih untuk Pak Edi Surono dan Pak Messy yang telah memberikan 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 COE 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.



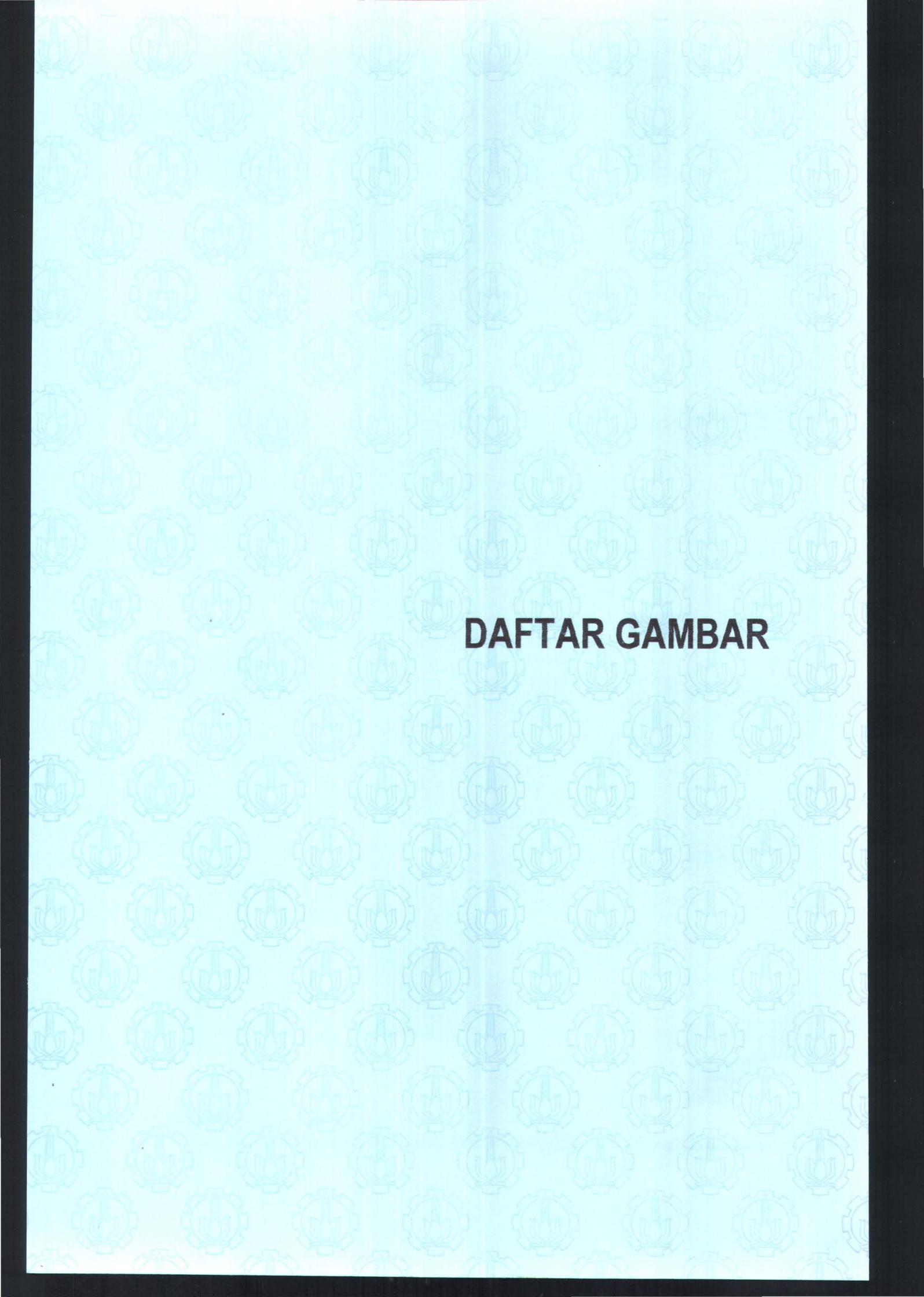
DAFTAR ISI

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 <i>Data Struktur Proyek</i>	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 <i>Kebutuhan Data Struktur Proyek</i>	42
4.1.2.2 <i>Kebutuhan Data Keuangan Proyek</i>	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 <i>DAD Level 2 Login</i>	49
5.3.3.2 <i>DAD Level 2 User Management</i>	50
5.3.3.3 <i>DAD Level 2 OLAP Entri</i>	51
5.3.3.4 <i>DAD Level 2 OLAP Report</i>	52
5.3.4 DAD Level 3.....	54
5.3.4.1 <i>DAD Level 3 Penentuan Cube dan Layout</i>	54
5.3.4.2 <i>DAD Level 3 Data Display Customizing</i>	55
5.2 DESAIN DATA.....	57
5.2.1 Desain Data <i>User Management</i>	57
5.2.2 Desain Data OLAP Entri.....	58
5.2.3 Desain Data OLAP Report.....	59
5.2.3.1 <i>Desain Data Struktur Database</i>	59
5.2.3.2 <i>Desain Data Penyimpanan Display Analisis</i>	61
5.2.3.3 <i>Desain Query Data Cube</i>	61
5.3 DESAIN ANTARMUKA	63
5.3.1 Desain Antarmuka <i>User Management</i>	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 <i>User Management</i> 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 <i>User Management</i>	102
7.2.3 Uji Coba OLAP Entri	103
7.2.4 Uji Coba OLAP Report	105
7.3 ANALISIS HASIL UJI COBA	113
BAB VIII KESIMPULAN DAN SARAN	114
8.1 KESIMPULAN	114
8.2 SARAN	114
DAFTAR PUSTAKA	
LAMPIRAN	
LAMPIRAN A DESAIN ALGORITMA PERHITUNGAN INDIKATOR KEUANGAN	
LAMPIRAN B JOIN-QUERY PEMBUATAN CUBE	
LAMPIRAN C DAFTAR ISIAN SEKTOR/SUBSEKTOR/PROGRAM PROYEK PEMBANGUNAN	

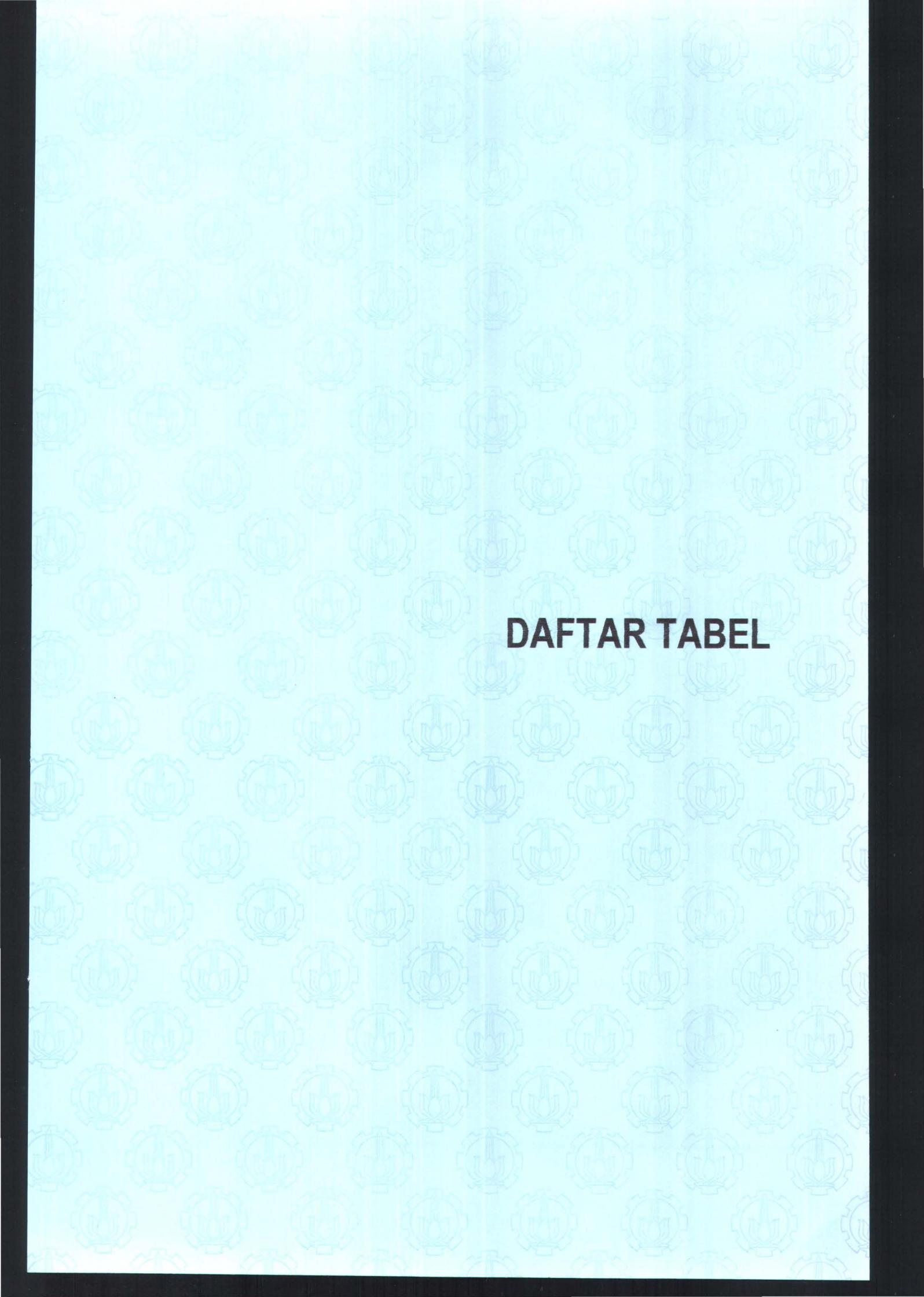


DAFTAR GAMBAR

DAFTAR GAMBAR

GAMBAR 2.1	ILUSTRASI FEASIBILITY STUDY PROYEK PEMERINTAH.....	11
GAMBAR 2.2	MULTIDIMENSIONAL DATA VIEW.....	20
GAMBAR 2.3	ARSITEKTUR ROLAP.....	23
GAMBAR 2.4	ARSITEKTUR MOLAP.....	25
GAMBAR 2.5	ARSITEKTUR MQE.....	26
GAMBAR 2.6	SYSTEM DEVELOPMENT LIFE CYCLE.....	30
GAMBAR 3.1	STRUKTUR PROYEK.....	35
GAMBAR 5.1	DAD LEVEL 0.....	47
GAMBAR 5.2	DAD LEVEL 1.....	47
GAMBAR 5.3	DAD LEVEL 2 LOGIN.....	49
GAMBAR 5.4	DAD LEVEL 2 USER MANAGEMENT.....	50
GAMBAR 5.5	DAD LEVEL 2 OLAP ENTRI.....	52
GAMBAR 5.6	DAD LEVEL 2 OLAP REPORT.....	53
GAMBAR 5.7	DAD LEVEL 3 PENENTUAN CUBE DAN LAYOUT.....	55
GAMBAR 5.8	DAD LEVEL 3 DATA DISPLAY CUSTOMIZING.....	56
GAMBAR 5.9	ERD USER MANAGEMENT.....	58
GAMBAR 5.10	ERD OLAP ENTRI.....	60
GAMBAR 5.11	ERD STRUKTUR DATABASE OLAP REPORT.....	61
GAMBAR 5.12	ERD PENYIMPANAN DISPLAY ANALISIS OLAP REPORT.....	61
GAMBAR 5.13	DESAIN ANTARMUKA USER MANAGEMENT.....	63
GAMBAR 5.14	DESAIN ANTARMUKA OLAP ENTRI.....	64
GAMBAR 5.15	DESAIN ANTARMUKA OLAP REPORT.....	65
GAMBAR 6.1	FORM DATA USER.....	69
GAMBAR 6.2	FORM DEPARTEMEN.....	70
GAMBAR 6.3	FORM GROUP USER.....	70
GAMBAR 6.4	FORM PASSWORD USER.....	71
GAMBAR 6.5	MDI OLAP ENTRI.....	71
GAMBAR 6.6	TOOLBAR MDI OLAP ENTRI.....	73
GAMBAR 6.7	TOOLBAR ACTIVEFORM PADA OLAP ENTRI.....	74
GAMBAR 6.8	PENGGAMBARAN TREE-VIEW PADA STRUKTUR PROYEK.....	75
GAMBAR 6.9	FORM ENTRI SEKTOR.....	77
GAMBAR 6.10	FORM ENTRI SUBSEKTOR.....	77
GAMBAR 6.11	FORM ENTRI PROGRAM.....	78
GAMBAR 6.12	FORM ENTRI PROYEK.....	79
GAMBAR 6.13	FORM ENTRI KELAS PROYEK.....	80
GAMBAR 6.14	FORM ENTRI BALANCE SHEET.....	80
GAMBAR 6.15	FORM ENTRI INCOME STATEMENT.....	81

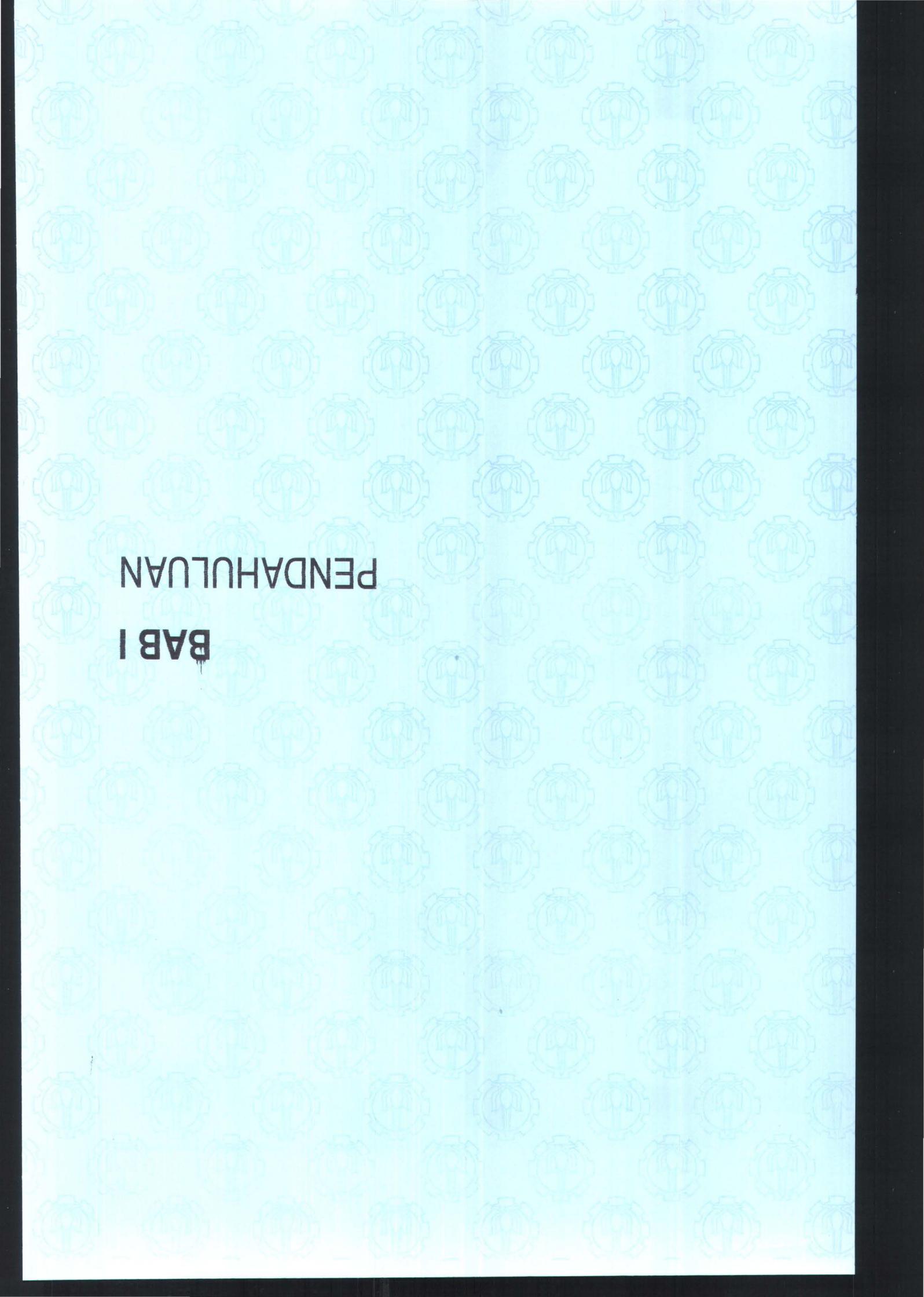
GAMBAR 6.16	FORM ENTRI BENEFIT COST.....	82
GAMBAR 6.17	FORM ENTRI CASH FLOW	82
GAMBAR 6.18	FORM REPORT BCR.....	83
GAMBAR 6.19	FORM REPORT IRR.....	86
GAMBAR 6.20	FORM REPORT PBP	86
GAMBAR 6.21	FORM REPORT ROA	88
GAMBAR 6.22	FORM REPORT ROI.....	89
GAMBAR 6.23	MDI OLAP REPORT.....	91
GAMBAR 6.24	TOOLBAR OLAP REPORT.....	92
GAMBAR 7.1	PERINGATAN USER GROUP TIDAK MEMILIK HAK AKSES TERHADAP SEBUAH MODUL.....	101
GAMBAR 7.2	PERINGATAN INPUTAN KOMBINASI USER NAME DAN PASSWORD YANG KELIRU	102
GAMBAR 7.3	PERINGATAN PENUTUPAN APLIKASI SEGERA OLEH SISTEM.	102
GAMBAR 7.4	MENU OLAP ENTRI YANG DIPERKENANKAN UNTUK USER GROUP DATA	103
GAMBAR 7.5	MENU OLAP ENTRI YANG DIPERKENANKAN UNTUK USER GROUP KEUANGAN	103
GAMBAR 7.6	STRUKTUR PROYEK HASIL UJI COBA PEMASUKAN SEKTOR,SUBSEKTOR, PROGRAM DAN PROYEK BARU.....	105
GAMBAR 7.7	HASIL UJI COBA PEMBUATAN DOKUMEN ANALISIS BARU	106
GAMBAR 7.8	HASIL UJI COBA MANIPULASI DENGAN CUSTOM WIZARD....	108
GAMBAR 7.9	HASIL UJI COBA MANIPULASI DENGAN RANGE.....	109
GAMBAR 7.10	HASIL UJI COBA MANIPULASI DENGAN FIELDS PROPERTIES	110
GAMBAR 7.11	PEMILIHAN SEL DATA GRAFIK.....	110
GAMBAR 7.12	HASIL UJI COBA PENGGAMBARAN GRAFIK.....	111
GAMBAR 7.13	HASIL UJI COBA EXPORT DATA KE EXCELL WORKBOOK.....	111
GAMBAR 7.14	HASIL UJI COBA PENYIMPANAN DOKUMEN FAVORITE ANALYSIS	112
GAMBAR A.1	FLOWCHART ALGORITMA PERHITUNGAN ROA	117
GAMBAR A.2	FLOWCHART ALGORITMA PERHITUNGAN ROI.....	118
GAMBAR A.3	FLOWCHART ALGORITMA PERHITUNGAN IRR	120
GAMBAR A.4	FLOWCHART ALGORITMA PERHITUNGAN PBP.....	121
GAMBAR A.5	FLOWCHART ALGORITMA PERHITUNGAN SC.....	122
GAMBAR A.6	FLOWCHART ALGORITMA PERHITUNGAN BCR.....	124



DAFTAR TABEL

DAFTAR TABEL

TABEL 2.1	CONTOH BENEFIT, DISBENEFIT, COST	15
TABEL 6.1	MENU-MENU MDI OLAP ENTRI	72
TABEL 6.2	FUNGSI ICON-ICON PADA TOOLBAR MDI OLAP ENTRI.....	73
TABEL 6.3	FUNGSI ICON-ICON PADA TOOLBAR ACTIVEFORM OLAP ENTRI...	74
TABEL 6.4	MENU-MENU PADA MDI OLAP REPORT	91
TABEL 6.5	FUNGSI ICON-ICON PADA TOOLBAR OLAP REPORT	92
TABEL C.1	DAFTAR ISIAN SEKTOR/SUBSEKTOR/PROGRAM PROYEK PEMBANGUNAN	128



BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan beberapa hal dasar yang meliputi latar belakang, tujuan, permasalahan, batasan permasalahan, metodologi, dan sistematika pembahasan. Diharapkan bab ini akan memberikan gambaran umum terhadap permasalahan dan pemecahannya yang dapat digunakan untuk menyelesaikan tugas akhir ini.

1.1 Latar Belakang

Memasuki era reformasi sejak pertengahan tahun 1998, sangat disadari pentingnya membangun kembali manajemen pemerintahan melalui paradigma baru menuju *good governance*. Menurut Bank Dunia (1992), *good governance* adalah sistem pemerintahan yang handal, pelayanan publik yang efisien, serta pemerintahan yang akuntabel terhadap publik. Pemakaian istilah *good governance* direkomendasikan oleh Bank Dunia sebagai pilihan dari *good government* atau *clean government*. *Good governance* berlaku terhadap keseluruhan lembaga negara dalam penyelenggaraan negara (ROSJIDI 2001: 141-142).

Dalam setiap penyelenggaraan, *good governance* harus berlandaskan pada tiga prinsip dasar, yaitu: transparansi, partisipasi, dan akuntabilitas. Ketiga prinsip tersebut merupakan bagian tak terpisahkan dalam setiap penentuan kebijakan publik, implementasi, dan pertanggungjawabannya dalam bingkai *good governance*.

Good governance berlaku untuk keseluruhan lembaga negara dalam penyelenggaraan negara. Salah satu hal penting dalam dimensi fungsi lembaga-

lembaga negara adalah jenis-jenis layanan fungsionalnya kepada *stakeholders* masing-masing lembaga. Layanan fungsional ini di beberapa negara lain sudah merupakan layanan baku (*standard services*) yang bersifat wajib diberikan oleh lembaga pemerintah, baik kepada masyarakat umum secara langsung maupun lewat lembaga-lembaga pemerintah lain yang mempunyai kegiatan yang searah. Layanan-layanan dasar ini antara lain adalah layanan di bidang kesehatan masyarakat, di bidang prasarana umum, di bidang pendidikan. Bahkan bidang-bidang baru seperti teknologi informasi juga sudah diberikan bobot yang cukup besar penanganannya.

Perlunya layanan fungsional sebagai *standard services* yang diberikan oleh pemerintah dan lembaga-lembaga pemerintah juga harus dijawab oleh daerah dalam pengembangan potensi-potensi yang dimiliki oleh daerah, sejalan dengan rencana pembangunan dan pengembangan potensi-potensi yang dimiliki oleh daerah. Untuk mewujudkan hal ini, pemerintah daerah memerlukan partisipasi aktif lembaga-lembaga yang ada dalam lingkup pemerintahan daerah, seperti Bappeda, Pemda, DPRD, Dinas-Dinas Daerah, dan lembaga-lembaga daerah terkait lainnya. Lembaga-lembaga daerah ini menjalankan tugas dan fungsinya berkaitan dengan pembangunan daerah secara terstruktur dan koordinatif untuk membantu kelancaran tata alir prosedur pembangunan infrastruktur daerah, sarana dan prasarana, serta pengembangan potensi-potensi yang dimiliki daerah.

Pemerintah daerah dalam hal ini perlu mengambil inisiatif untuk mengidentifikasi dan merencanakan program-program pembangunan dan pengembangan potensi-potensi yang mereka miliki dan segera menjabarkan kebutuhan layanan masyarakat untuk meningkatkan pendapatan asli daerah, seperti pelayanan administrasi (identitas penduduk dan perijinan), pelayanan infrastruktur

(jalan raya, jaringan irigasi, jembatan), pelayanan kebutuhan dasar (pangan, air minum, kesehatan, pendidikan), dan pelayanan penerimaan daerah. Selain itu, pemerintah daerah juga menjabarkan dengan sistematis indikator-indikator penilaian program pengembangan potensi daerah sebagai sebuah proyek pengembangan yang ditangani oleh swasta (private, proses investasi), pemerintah daerah sendiri lewat RAPBD, atau nasional (pemerintah pusat). Sehingga dalam proses pembangunan tersebut juga diperlukan tertib administrasi dan keuangan.

Sebagaimana yang telah dijabarkan di atas, wacana pengelolaan sistem pemerintahan yang transparan dalam lingkup sistem pemerintahan nasional dan regional yang telah berkembang saat ini, membuat pemerintah daerah perlu menjabarkan jenis-jenis layanan fungsional pemerintah kepada masyarakat umum dan kalangan dunia usaha secara luas sehubungan dengan pengembangan potensi-potensi yang dimiliki daerah untuk meningkatkan taraf hidup masyarakat di daerah. Jenis-jenis layanan fungsional pemerintah memiliki beberapa dimensi kajian fungsional, yaitu Finansial, Administrasi, Proses, dll.

Penjabaran informasi layanan secara transparan secara langsung maupun tidak langsung membuka peluang dalam proses pelaksanaan pembangunan. Informasi kinerja proyek-proyek pembangunan beserta perencanaan tujuan dari masing-masing proyek membuka peluang yang lebih besar untuk terjadinya investasi bagi sektor bersangkutan. Informasi kinerja finansial yang cukup lengkap akan banyak membantu dalam proses investasi.

Untuk menjawab tantangan diatas, maka perlu dibuat sebuah sistem yang mampu memberi kemudahan kepada Pemerintah Daerah khususnya Bappeda dalam melakukan presentasi (promosi) terhadap potensi-potensi daerahnya dan juga dalam

perencanaan pembangunan yang akan dilakukan. OLAP (*Online Analytical Processing*) sebagai *analysis tool* merupakan salah satu metode yang digunakan untuk menampilkan data dalam bentuk suatu rekapitulasi yang dapat digunakan dalam sebuah Analisis.

Oleh sebab itu, OLAP akan memudahkan pengguna untuk melihat dan menganalisis sekumpulan data dari berbagai sudut pandang, sehingga akan membantu pengguna dalam mengambil keputusan secara tepat, misalnya: dalam sebuah proses menentukan kelayakan apakah sebuah proyek dapat dilaksanakan, maka pengguna dapat menganalisa dari berbagai kondisi yang ada, misal dengan Analisis *Benefit Cost Ratio*, analisis *Return On Investment*, atau bahkan melakukan analisis kombiner terhadap data-data yang ada sampai pada kurun waktu tertentu secara langsung (*online*).

1.2 Tujuan

Tujuan pembuatan tugas akhir ini adalah menghasilkan suatu perangkat lunak yang akan menerapkan pemakaian teknologi OLAP sebagai sebuah *analysis tool*, sehingga pengguna dapat memanipulasi tampilan, memindahkan sudut pandang data, merubah urutan data, menambahkan field-field, dan melakukan perubahan parameter dengan menggunakan fungsi-fungsi *aggregate* untuk dapat menjawab kebutuhan analisis yang dikehendaki. Dengan kelebihan dari OLAP tersebut, maka diharapkan dapat membantu pemerintah (khususnya Bappeda Pemda Kabupaten Kediri) dalam menentukan kebijakan dalam menentukan kelayakan proyek-proyek pembangunan.

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.

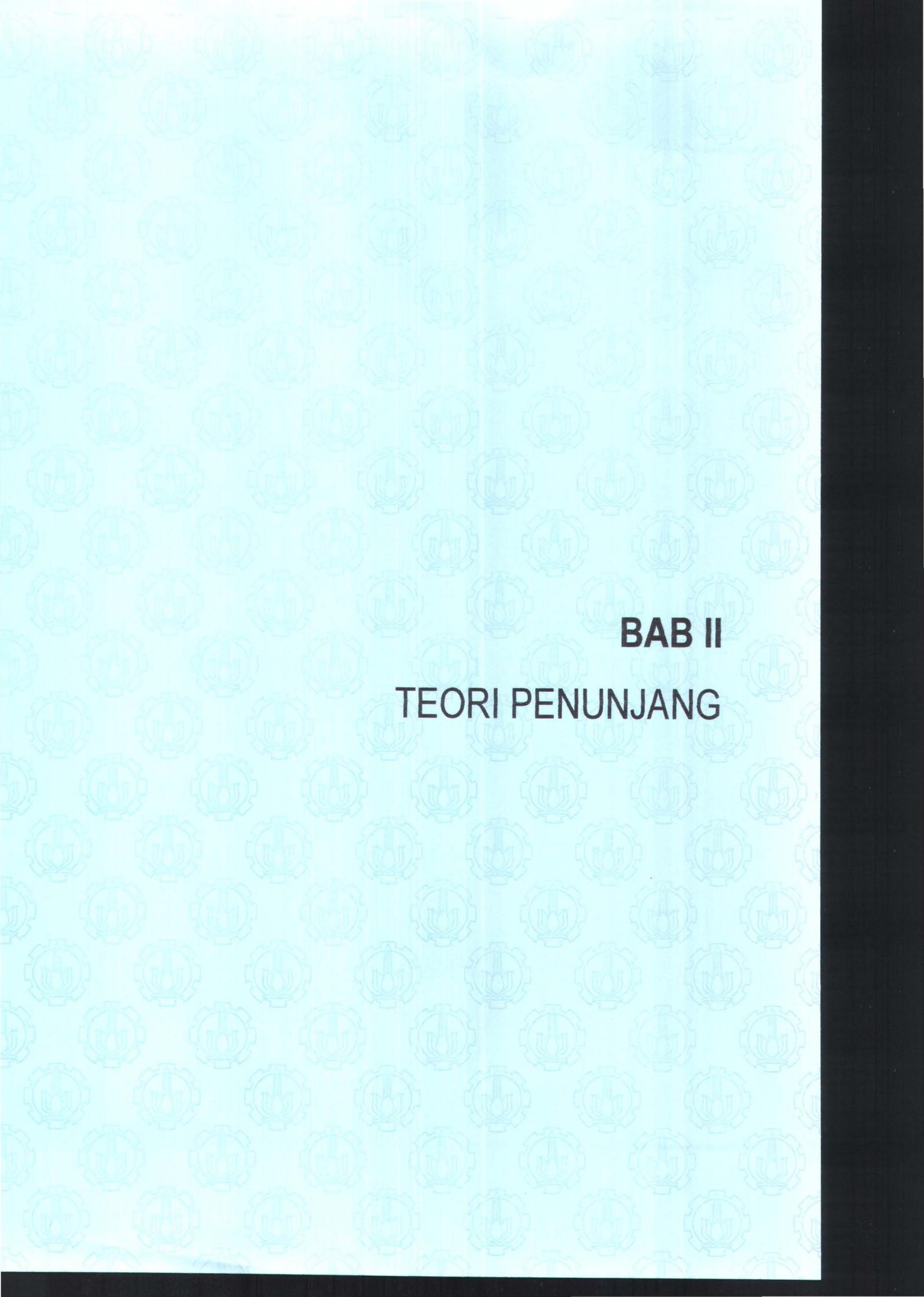
Bab IV, *Analisis Kebutuhan Sistem*, bab ini berisi identifikasi kebutuhan pada proses perencanaan di Bappeda dan obyektif sistem OLAP yang akan dibuat sesuai dengan hasil identifikasi tersebut, dan juga berkaitan dengan analisis aspek keuangan.

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.

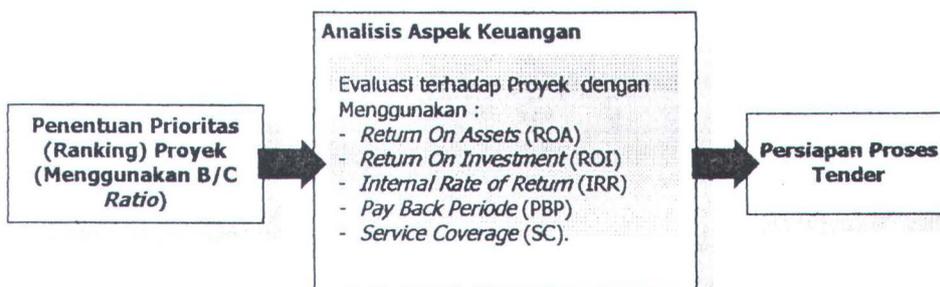


2.1.1 Analisis Keuangan dalam *Feasibility Study*

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:

$$PBP = (n-1) + \left[NF_0 - \sum_{t=1}^{n-1} NF_1 \right] \left(\frac{1}{NF_n} \right)$$

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
<ul style="list-style-type: none"> ▪ Penurunan biaya operasional kendaraan, termasuk bahan bakar ▪ Waktu perjalanan menjadi lebih singkat dan lancar ▪ Peningkatan keamanan lalu lintas ▪ Kemudahan mengendarai kendaraan ▪ Peningkatan harga tanah di sekitar jalan tol
Dampak negatif (disbenefit) bagi masyarakat umum
<ul style="list-style-type: none"> ▪ Pengurangan lahan pertanian ▪ Terganggunya saluran air untuk irigrasi ▪ Peningkatan polusi udara
Ongkos yang harus ditanggung oleh Pemerintah
<ul style="list-style-type: none"> ▪ Ongkos konstruksi ▪ Ongkos perawatan ▪ Ongkos administratif
Pendapatan bagi Pemerintah
<ul style="list-style-type: none"> ▪ 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)

Quarter	City			
	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
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 (*Rata-rata total pendapatan untuk setiap staf penjualan = Total pendapatan / Jumlah staf penjualan*)

Waktu respon dari sebuah query multidimensional bergantung pada seberapa banyak jumlah sel yang harus ditambahkan pada setiap proses yang berlangsung (*on the fly*). Seiring dengan bertambahnya dimensi, jumlah dari sel dari sebuah cube meningkat secara eksponensial. Keunggulan dari query multidimensional adalah data terkalkulasi atau data-data laporan. Oleh karena itu, solusi untuk membangun sebuah basisdata multidimensional yang efisien adalah untuk melakukan pre-agregasi atau konsolidasi terhadap semua data total atau sub total pada semua dimensi. Proses pre-agregasi ini akan sangat berarti, karena pada dasarnya dimensi-dimensi itu adalah sebuah hirarki. Sebagai contoh, dimensi waktu dapat terdiri dari hirarki tahun, triwulan, bulan, minggu dan hari, dan dimensi lokasi dapat terdiri dari kantor cabang, area, kota dan negara. Dengan diketahuinya definisi hirarki dari sebuah dimensi memberikan kemampuan untuk melakukan proses pre-agregasi dan sebaliknya. Proses *drill-down* (penjabaran) sebagai contoh dari data pendapatan tahunan, pendapatan triwulan sampai pendapatan bulanan. (CONNOLLY, 1998 : 950)

2.2.3 Kategorisasi OLAP

Alat-alat OLAP dikelompokkan berdasarkan arsitektur dari basisdata yang digunakan sebagai sumber dalam proses analisis. Ada beberapa jenis kelompok OLAP antara lain (CONNOLLY, 1998 : 955-957):

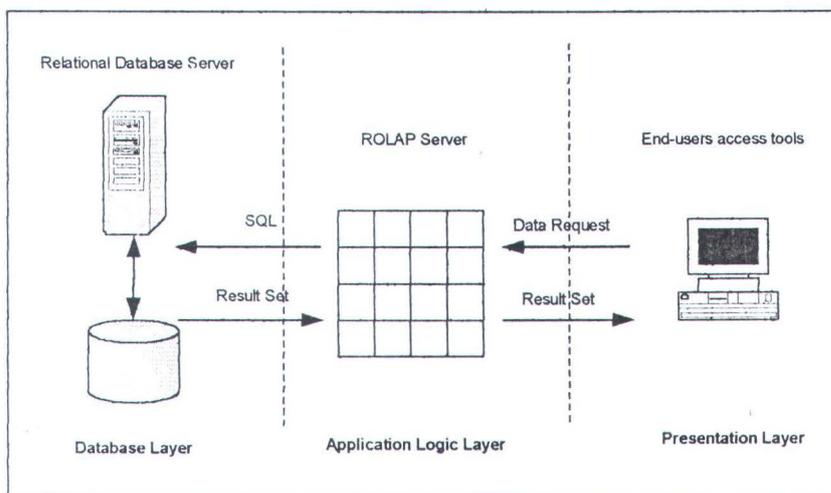
2.2.3.1 ROLAP

Relational OLAP (ROLAP) merupakan salah satu teknologi OLAP 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

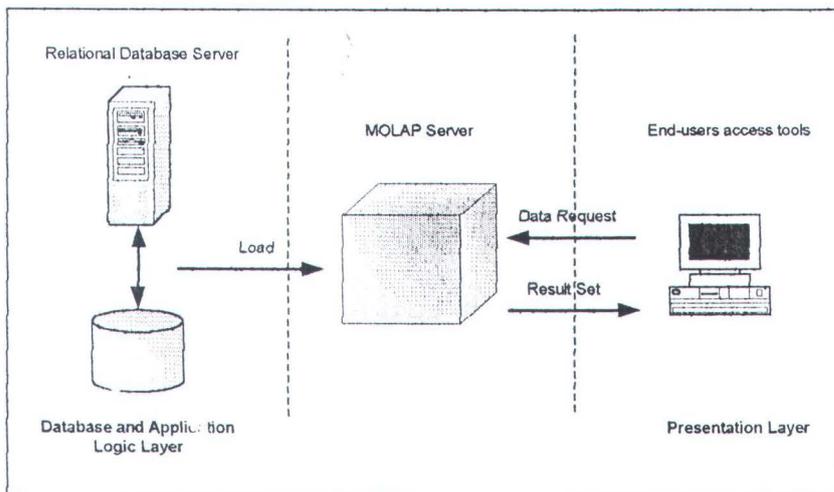
Permasalahan-permasalahan pada proses pengembangan ROLAP adalah sebagai berikut:

- 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

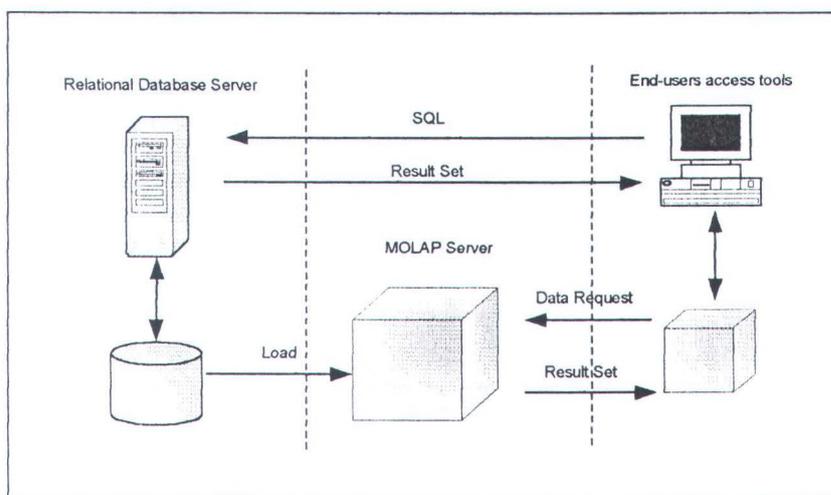
Permasalahan yang dihadapi dalam pengembangan sebuah alat MOLAP antara lain adalah:

- 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

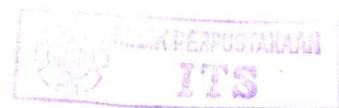
pembuatan tugas akhir ini akan dipergunakan metodologi *System Development Life Cycle* (SDLC), dengan tahap-tahap sebagai berikut (DAVIS 1994: 11-16) :

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.

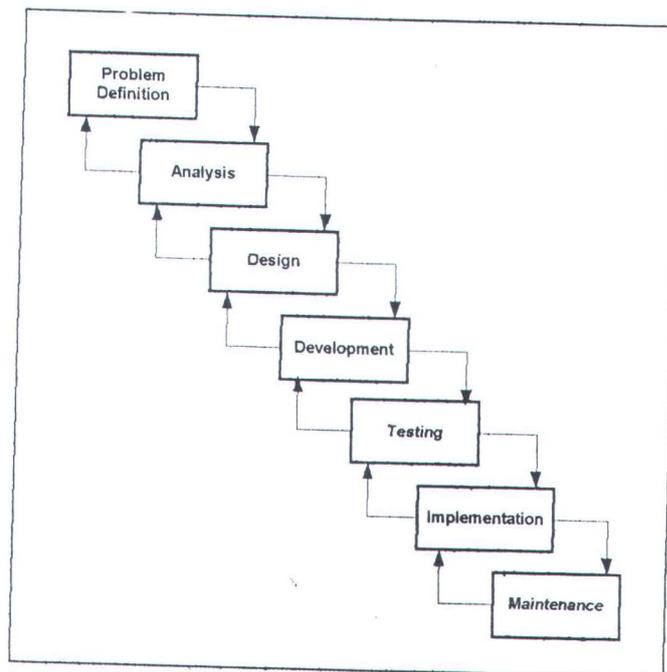
6. *Implementation*

Setelah proses uji coba selesai dilakukan dan semua kesalahan telah diperbaiki, maka sistem/aplikasi tersebut siap untuk *di-release*.

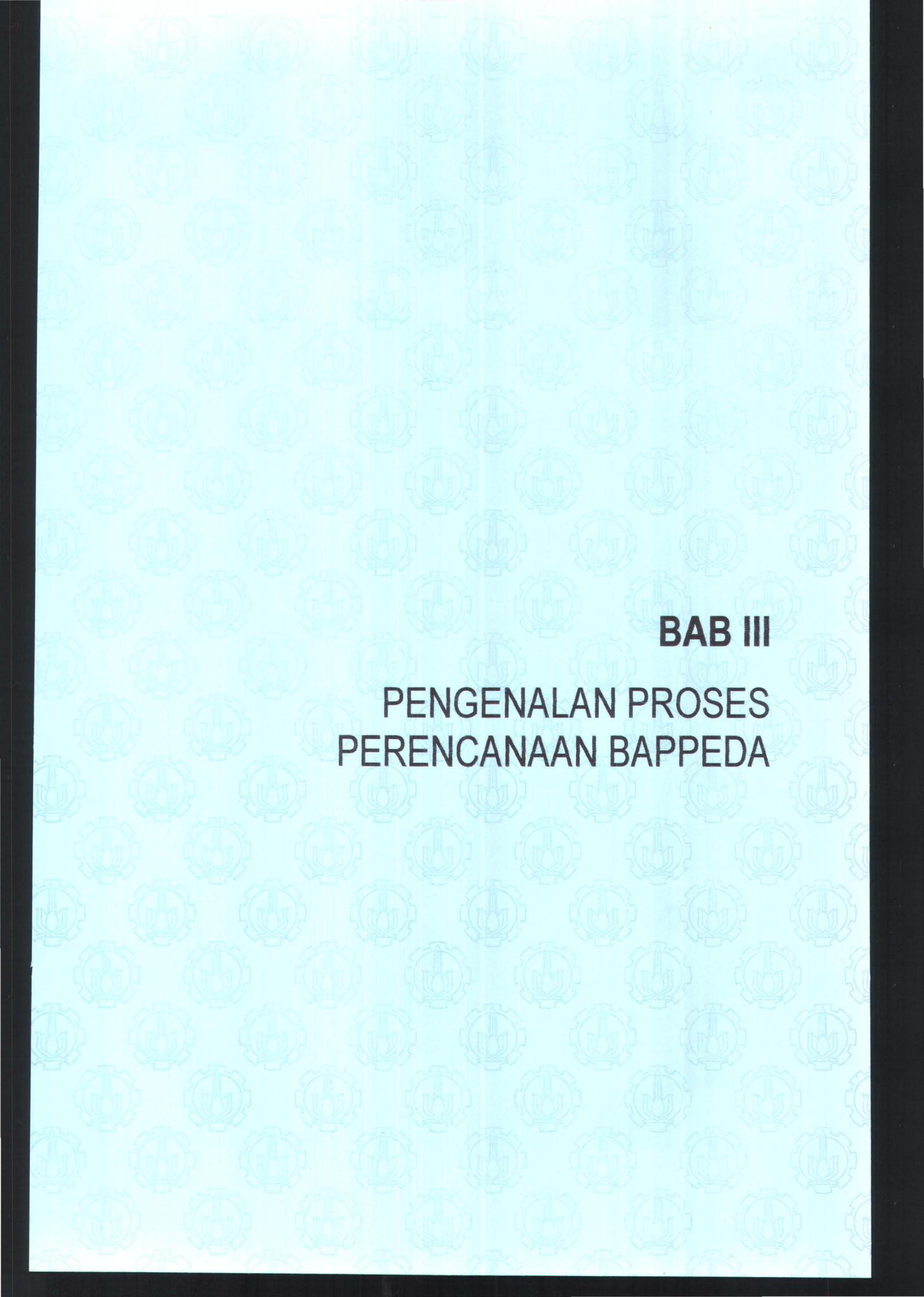
7. Maintenance

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

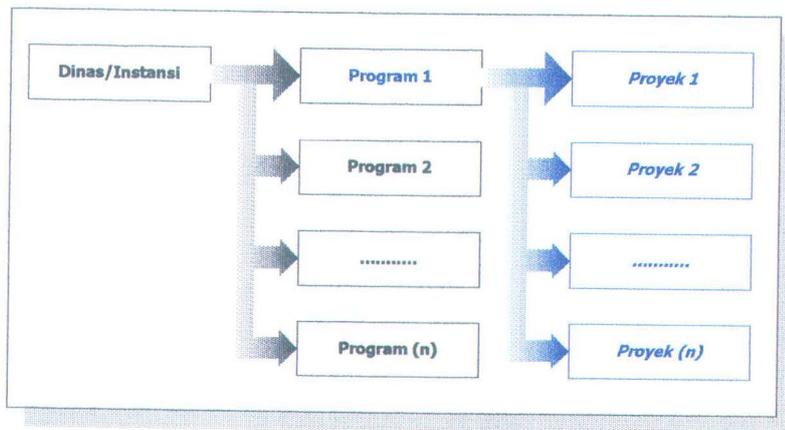
Data struktur proyek berisi kegiatan-kegiatan atau proyek-proyek pembangunan yang dilakukan oleh pemerintah daerah (Kabupaten Kediri) dalam setiap tahun perencanaan. Perencanaan proyek-proyek pembangunan tersebut merupakan kegiatan rutin dari pemerintah daerah dalam rangka melaksanakan aktivitas pembangunan 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.

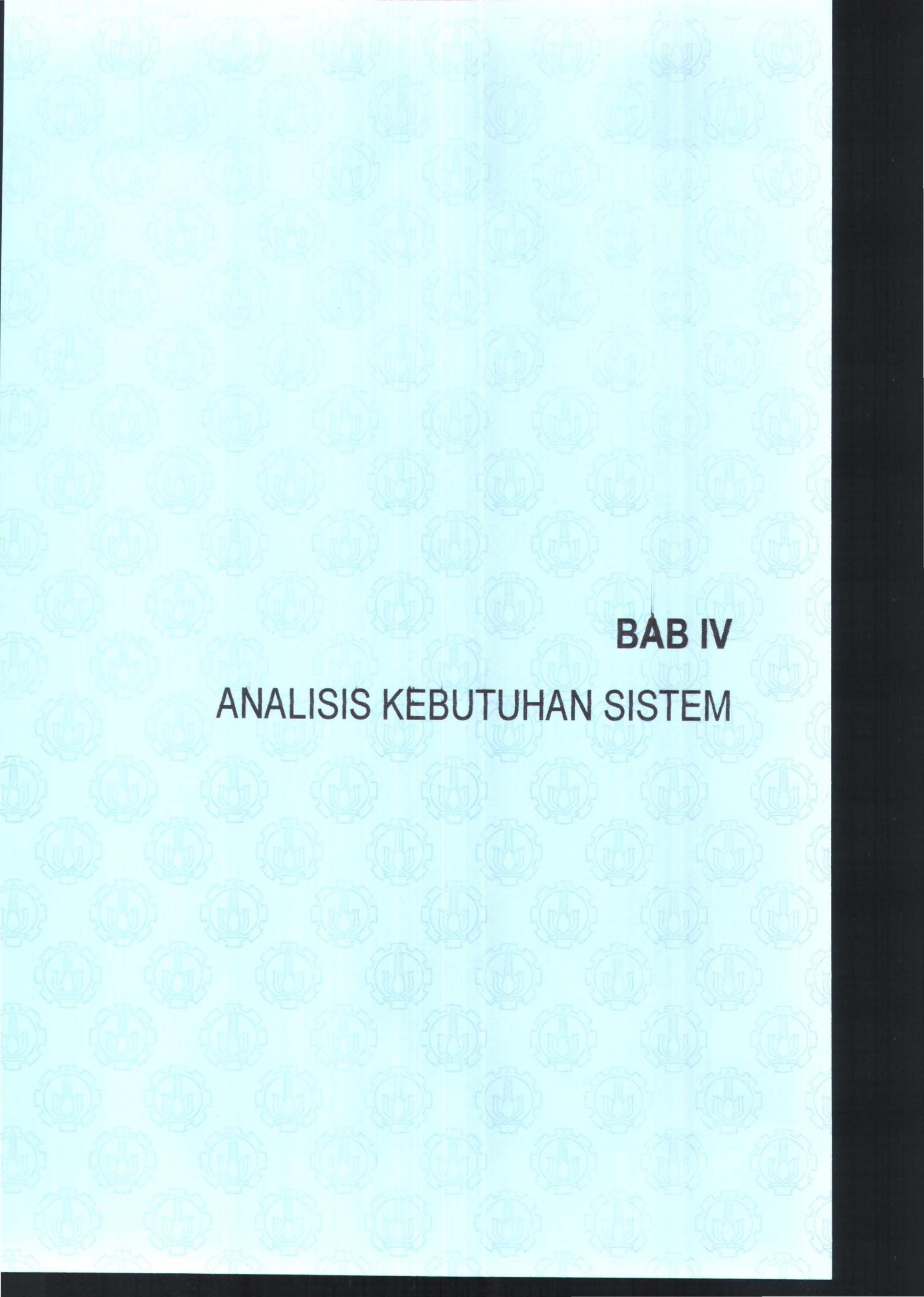
Format Studi Kelayakan yang akan dibuat diharapkan dapat memenuhi semua masukan yang dibutuhkan Bappeda dalam perencanaan program-program pembangunan daerah melalui Rakorbang.

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 perencana yang berkepentingan hadir dalam forum Rakorbang. Masalah waktu merupakan kunci utama yang harus dipenuhi.
- Ditunjang dengan ketersediaan data-data yang akan membantu proses perencanaan. Ketersediaan data berkaitan dengan ada tidaknya data yang diperlukan pada saat dibutuhkan dan dapat tidaknya data yang dibutuhkan tersebut dapat dicapai. Dengan kata lain, sebuah sistem pengolahan data yang terintegrasi akan sangat mendukung proses perencanaan ini.
- Sistem dapat memberikan respon yang cepat terhadap sebuah proses analisis data.
- Adanya fungsi-fungsi standar yang dapat mempercepat pengguna untuk mendapatkan informasi-informasi yang dibutuhkan pada proses analisis. Fungsi-fungsi standar itu berkaitan dengan kebutuhan sebuah proses analisis terhadap data-data, baik itu data hasil rekapitulasi sampai data pada level detail. Hal ini akan memberikan fleksibilitas pada proses analisis data.
- Untuk beberapa proses analisis yang sering dan telah dilakukan dapat disimpan dan diambil kembali untuk proses analisis-analisis sejenis

selanjutnya, sehingga proses analisis selanjutnya akan lebih cepat dilakukan karena tidak perlu lagi didefinisikan dari awal.

- Didukung dengan visualisasi data yang bermacam-macam seperti dalam bentuk *spreadsheet*, grafik dan tampilan yang berbasis grafis (*Graphical User Interface*).



BAB IV
ANALISIS KEBUTUHAN SISTEM

BAB IV

ANALISIS KEBUTUHAN SISTEM

Dalam bab IV ini akan diberikan sejumlah analisis terhadap kebutuhan sistem yang diharapkan akan membantu meningkatkan kinerja proses perencanaan pembangunan di Kabupaten Kediri, khususnya di Bappeda. Analisis sistem yang akan dilakukan ini merupakan langkah kedua —yaitu *analysis*— dalam metodologi SDLC.

4.1 Analisis Kebutuhan

Sebuah perencanaan pembangunan memerlukan proses analisis yang lebih dalam lagi terhadap data-data yang tersedia, sehingga mampu menjadi sarana yang efektif untuk menetapkan kebijakan dan strategi pembangunan. Peranan data-data tersebut sebenarnya sangat penting, karena berfungsi antara lain untuk:

- a. *Referensi*, sebagai rujukan dan bahan pertimbangan dalam menentukan kebijakan dan strategi pembangunan serta sasaran dari setiap kegiatan pembangunan;
- b. *Monitoring*, sebagai bahan dalam melakukan pengendalian pelaksanaan pembangunan yang sedang diselesaikan;
- c. *Evaluasi*, sebagai bahan untuk melihat tingkat perkembangan (kemajuan atau kemunduran) potensi dari suatu wilayah (kabupaten atau kecamatan) dalam suatu periode;

- d. *Presentasi*, sebagai media penyampaian hasil-hasil pelaksanaan pembangunan dan potensi yang dimiliki oleh Kabupaten Kediri kepada masyarakat maupun kalangan industri atau investor.

Namun, dengan melihat kondisi-kondisi yang ada di Pemda Kediri dalam menanggapi sebuah usulan dari Dinas atau Instansi dalam rangka melaksanakan proses perencanaan pembangunan yang masih konvensional, maka akan menjadi suatu hal yang sangat sulit untuk dapat merealisasikan poin-poin tersebut di atas, walaupun dalam beberapa hal, telah dilakukan komputerasi 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

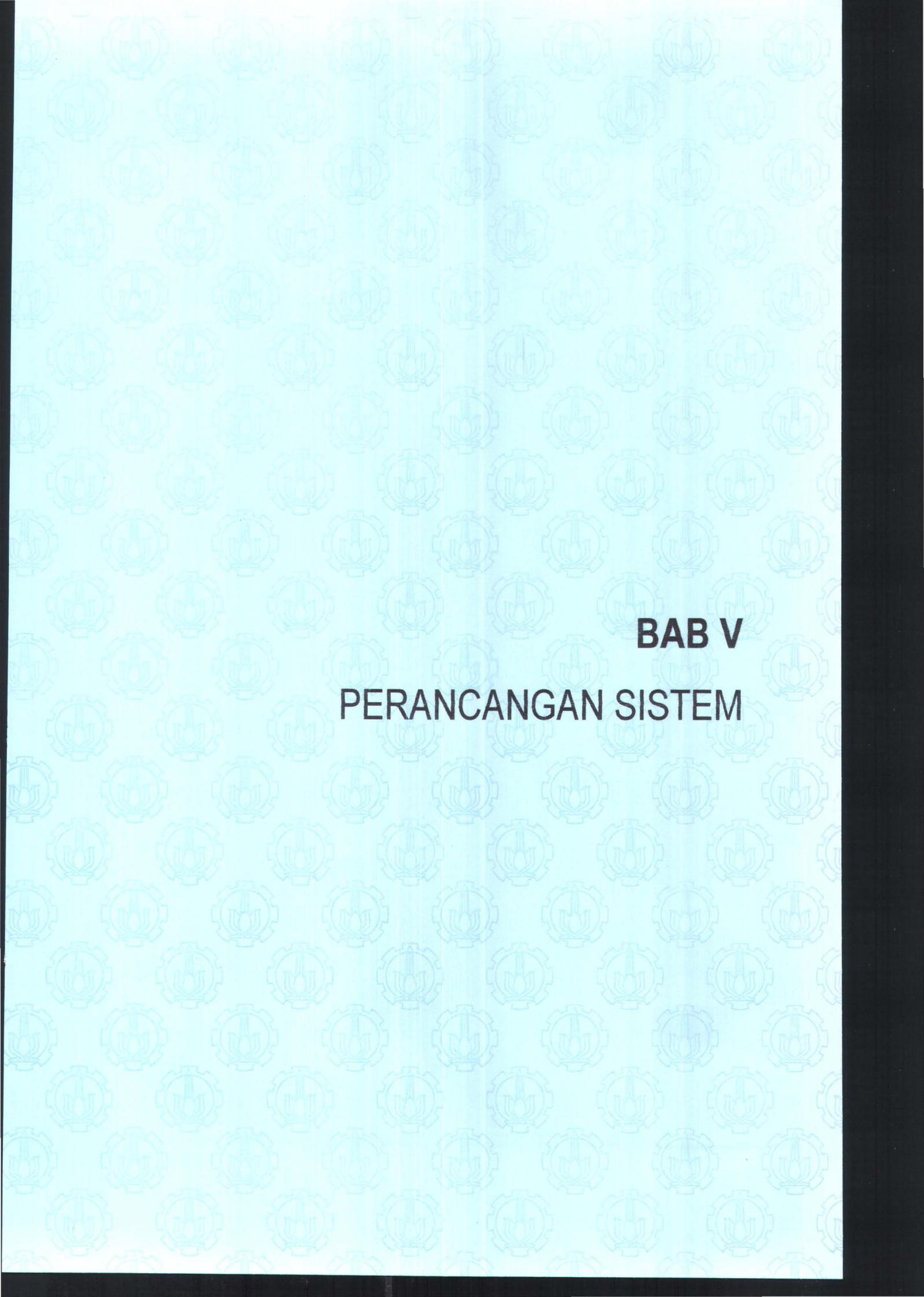
- Windows 9X / NT 4.0 / 2000 / XP
- Microsoft Excel 9X/ 2000

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.

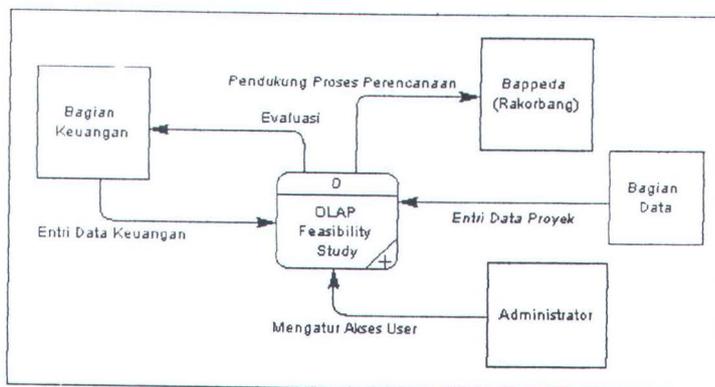
5.3.1 DAD Level 0

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:

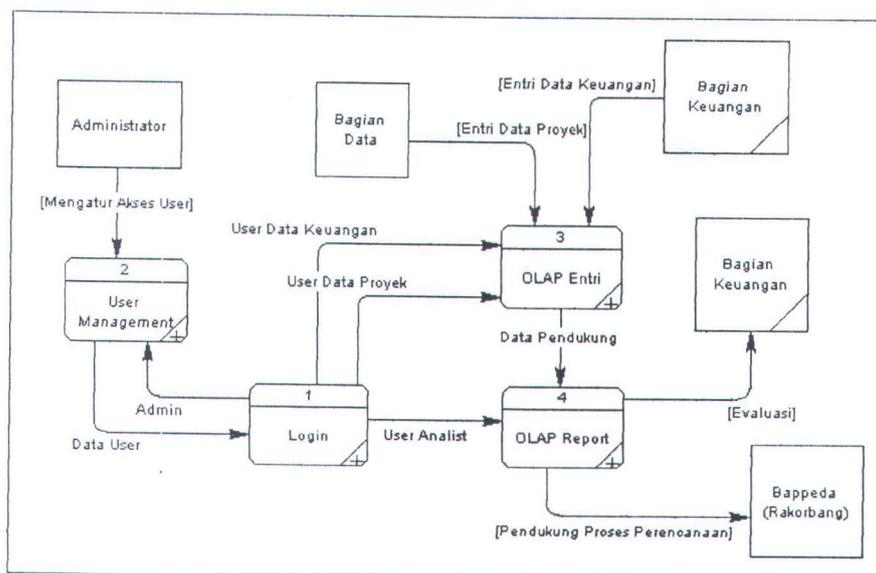
- Bagian Data, melakukan entri data-data proyek berkaitan dengan struktur dan kelas proyek.

- 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

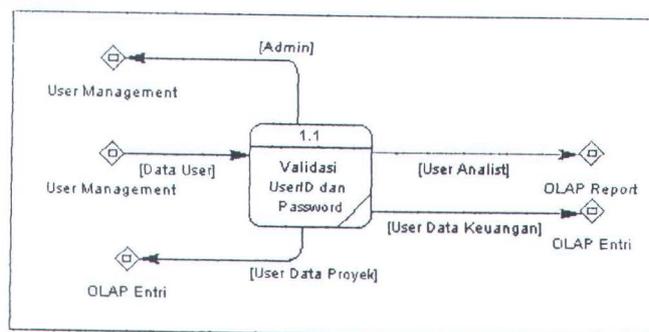
perencanaan yang dilakukan Bappeda dalam Rakorbang dan juga digunakan Bagian Keuangan untuk melakukan evaluasi terhadap proyek.

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*

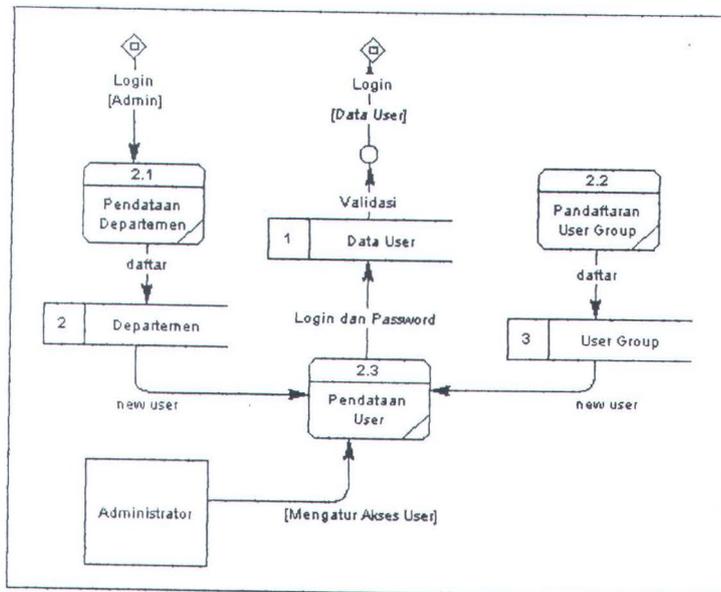
Dari data *user* yang terdapat pada *User Management*, maka proses Validasi kombinasi UserID dan Password akan menentukan kelompok *user group* bersangkutan. Jika dalam proses validasi dengan data *user* tersebut menyatakan kedudukan seorang pengguna dalam sebuah kelompok *user group* adalah sebagai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

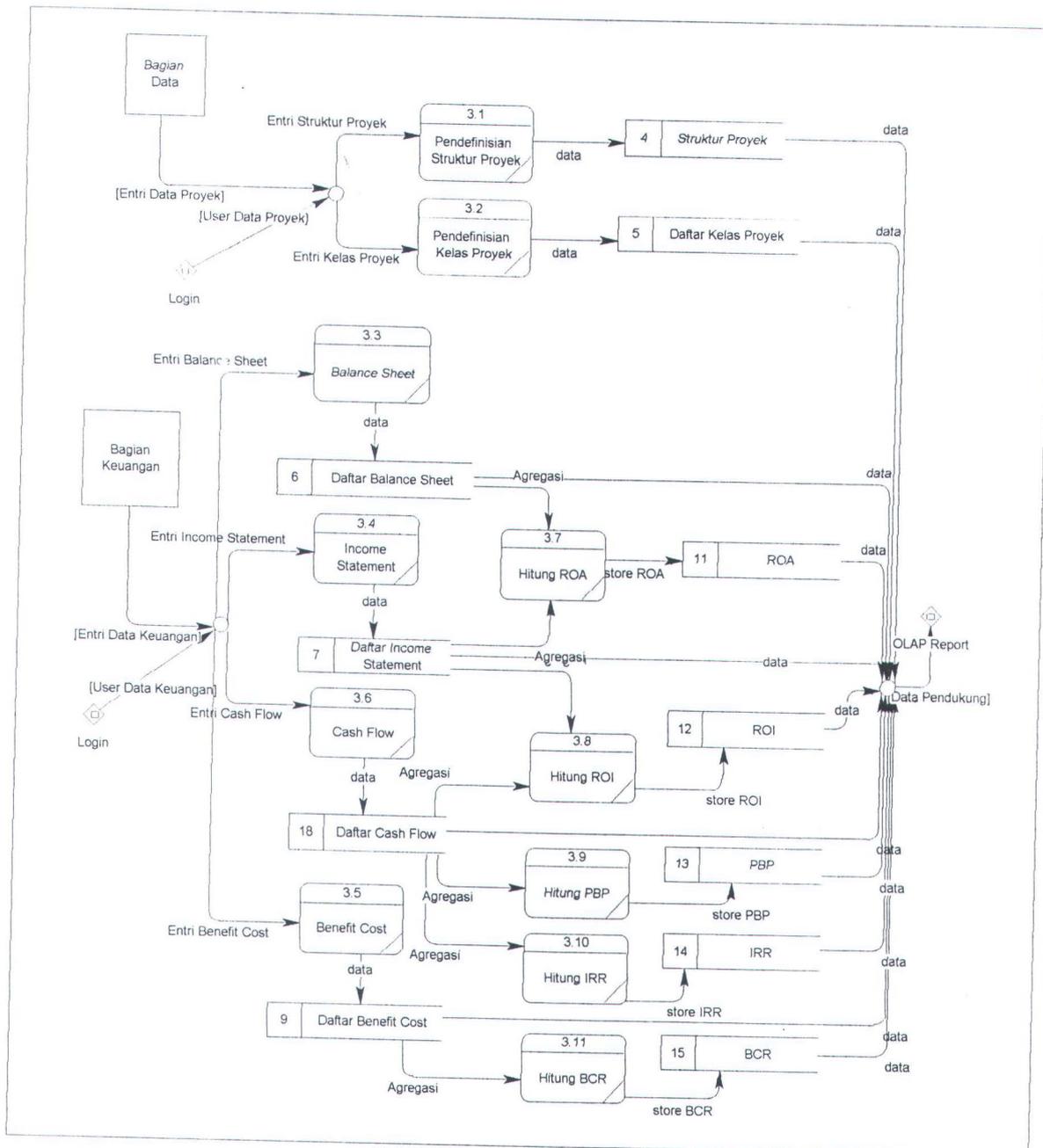
DAD level 2 OLAP Entri menggambarkan proses entri (input) data-data pendukung yang nantinya akan mendukung sistem OLAP yang akan dibuat. Data-data pendukung berasal dari bagian keuangan dan bagian data.

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.

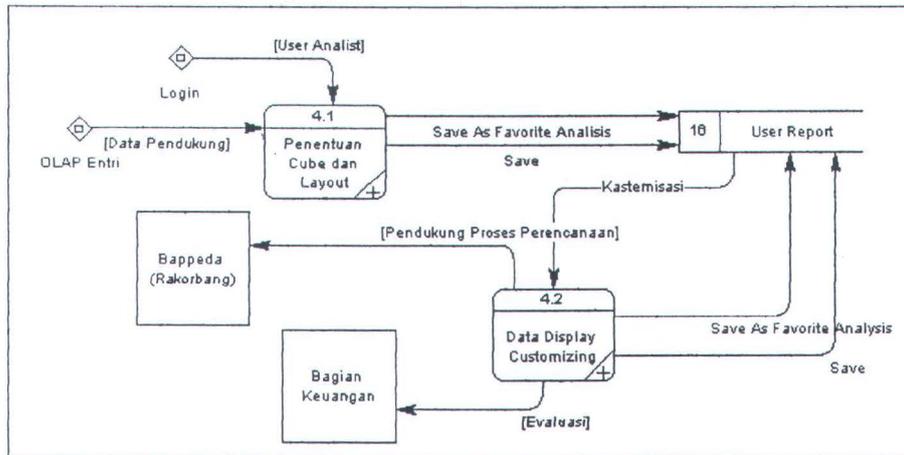


Gambar 5.5 DAD Level 2 OLAP Entri

5.3.3.4 DAD Level 2 OLAP Report

- DAD level 2 OLAP Report menggambarkan proses pembuatan dokumen OLAP Report yang hasilnya akan merupakan sebuah aplikasi OLAP yang

sesungguhnya, yang akan yang terdiri atas 2 (dua) proses utama yaitu Penentuan *Cube* dan *Layout* serta proses *Data Display Customizing*, yang digambarkan seperti Gambar 5.6.



Gambar 5.6 DAD Level 2 OLAP Report

Proses yang terjadi pada OLAP Report secara garis besar terdiri atas:

- Penentuan *Cube* dan *Layout*, merupakan proses pertama yang harus dilakukan pengguna terhadap sistem. Proses ini memerlukan data-data yang telah diinputkan pada proses sebelumnya yaitu pada OLAP Entri. Setelah melakukan aktivitas-aktivitas yang ada dalam proses ini, maka dokumen analisis yang telah dibuat *user*, dapat langsung disimpan (*save*) ataupun disimpan sebagai *favorite analysis* (*Save As Favorite Analysis*).

Data Display Customizing, proses *customization* terhadap hasil dokumen report yang telah disimpan, untuk semakin merubah penampakan dari data-data yang ada. Kemudian, setelah proses tersebut, analisis ini dapat disimpan lagi dengan nama yang berbeda, karena proses penyimpanan yang dilakukan oleh sistem dibuat secara incremental, hal ini akan berguna untuk tetap menjaga keberadaan dokumen analisis lama, untuk keperluan perbandingan dengan dokumen yang baru dibuat.

5.3.4 DAD Level 3

Penggambaran DAD level 3 sebagai penjabar dari masing-masing proses yang ada di level 2 yang masih bisa didetilkkan.

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

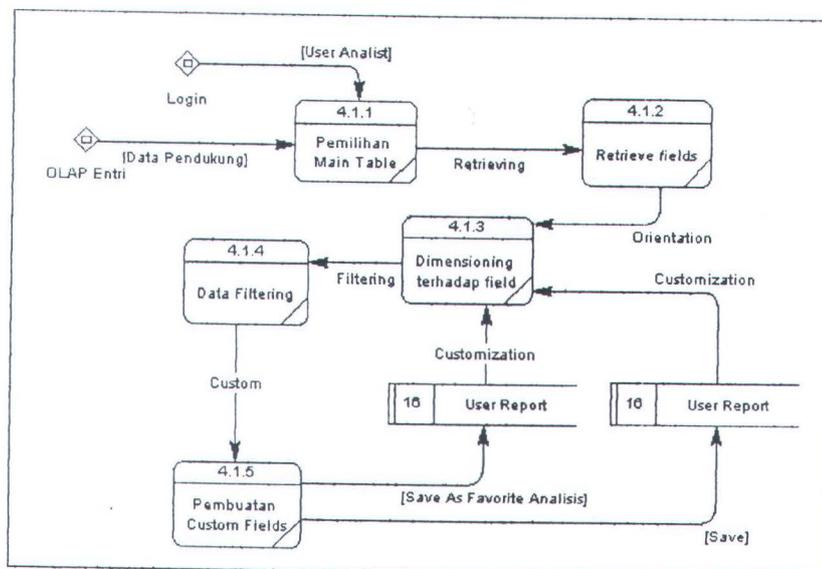
DAD level 3 Penentuan *Cube* dan *Layout* menggambarkan proses pemilihan terhadap field-field pada sebuah tabel utama (*main table*) untuk kemudian ditentukan orientasi field tersebut dalam penampakan (*layout*) pada sebuah dokumen analisis. DAD ini juga akan menggambarkan proses *filtering* data dan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

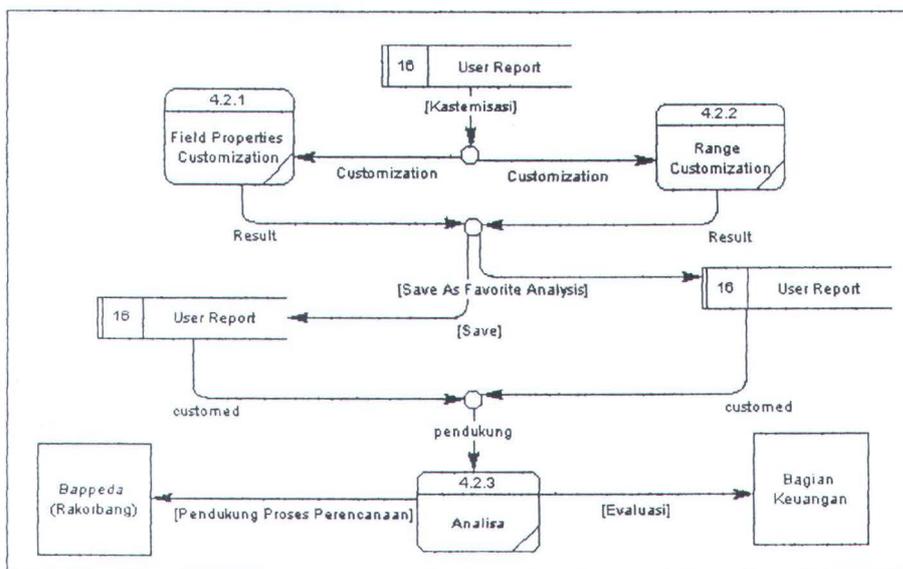
5.3.4.2 DAD Level 3 Data Display Customizing

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)*.

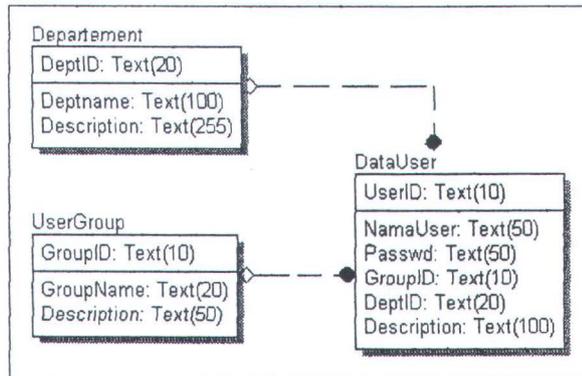
5.2.1 Desain Data *User Management*

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.

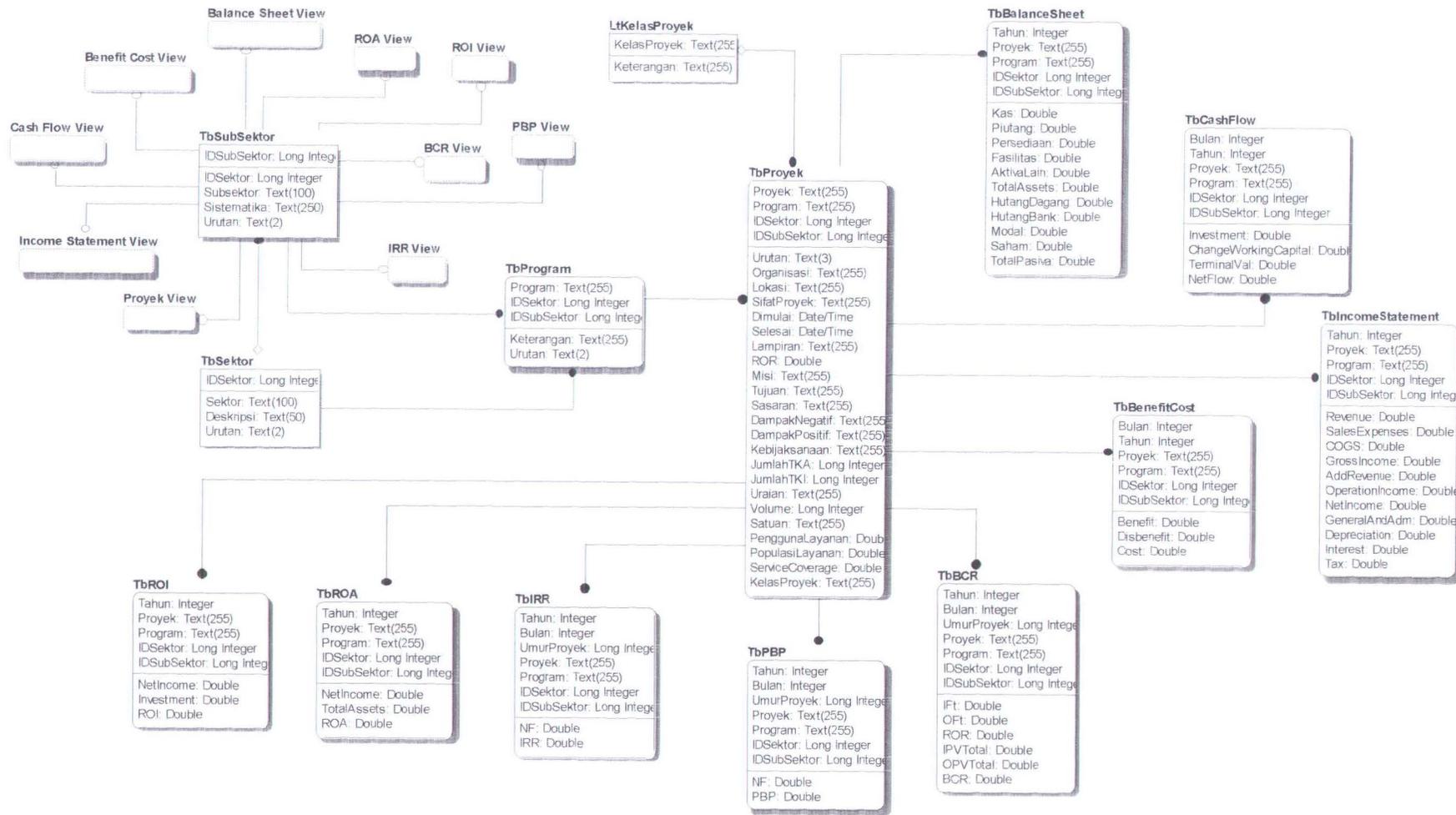
5.2.3 Desain Data OLAP Report

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.

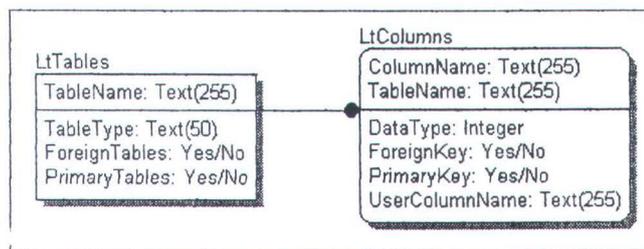


Gambar 5.1 ERD OLAP Entri

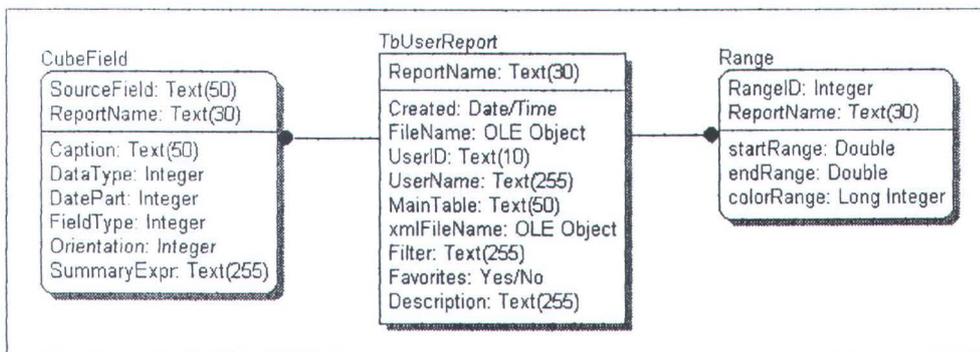
5.2.3.2 Desain Data Penyimpanan Display Analisis

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.

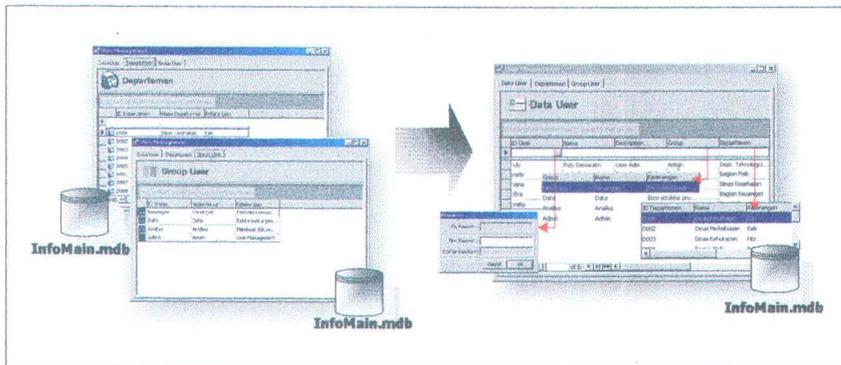
5.3 Desain Antarmuka

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.

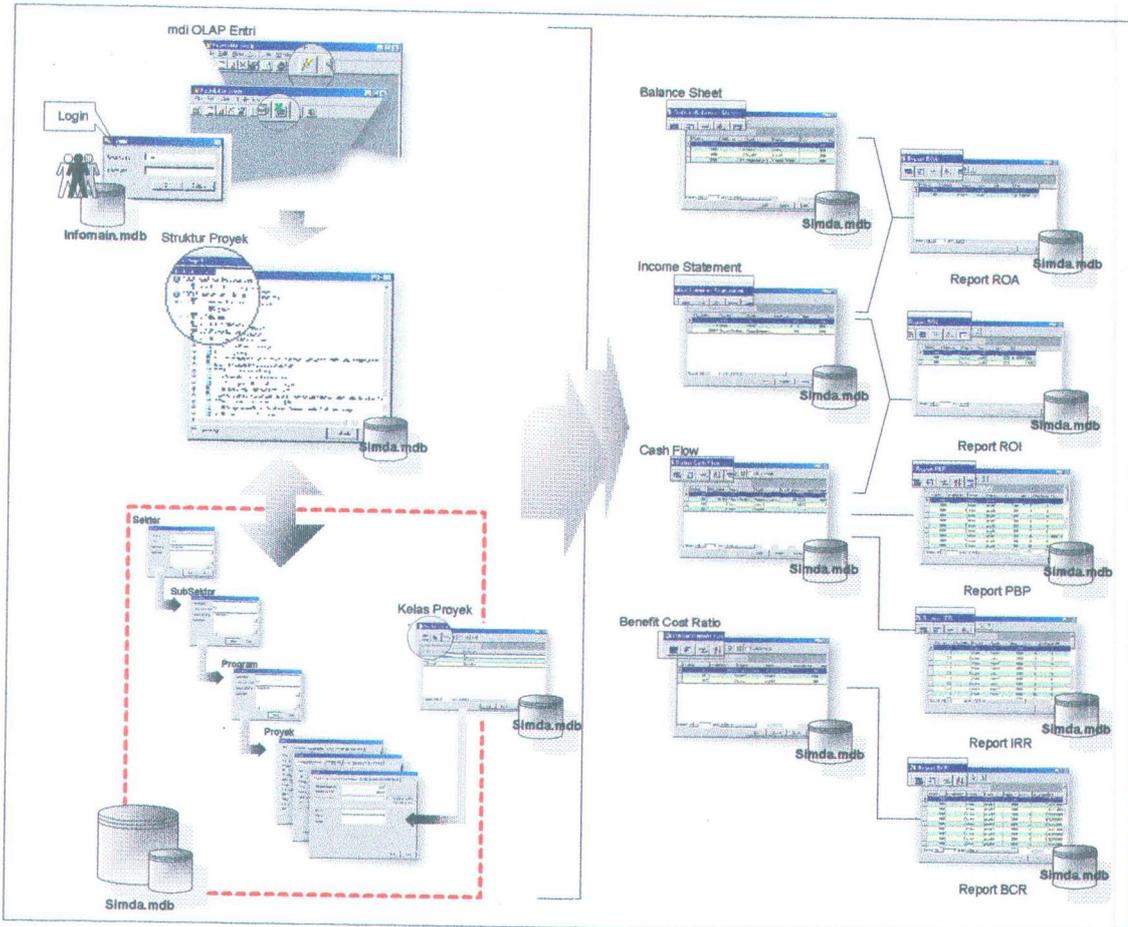


Gambar 5.13 Desain Antarmuka User Management

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.

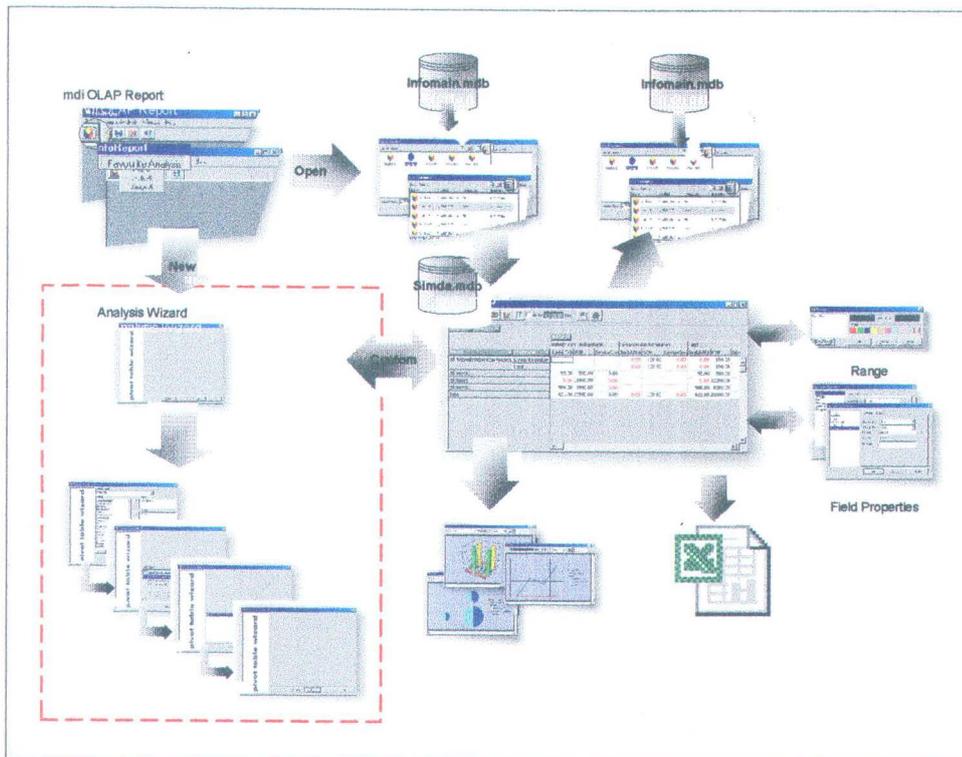


Gambar 5.14 Desain Antarmuka OLAP Entri

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..

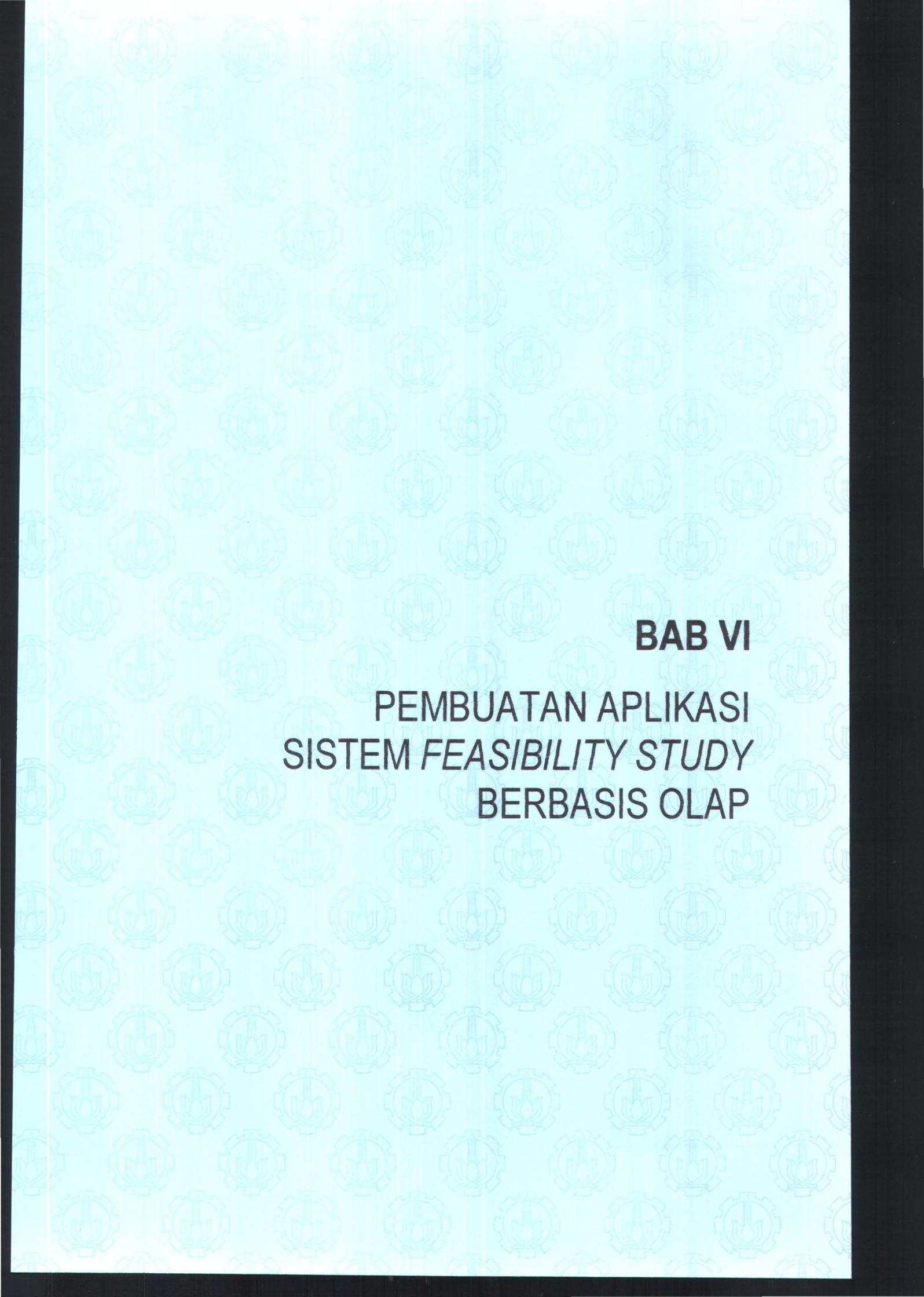
Secara garis besar, penggambaran antarmuka untuk modul OLAP Report akan memiliki hubungan antar form-form sebagai berikut pada gambar 5.15.



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.

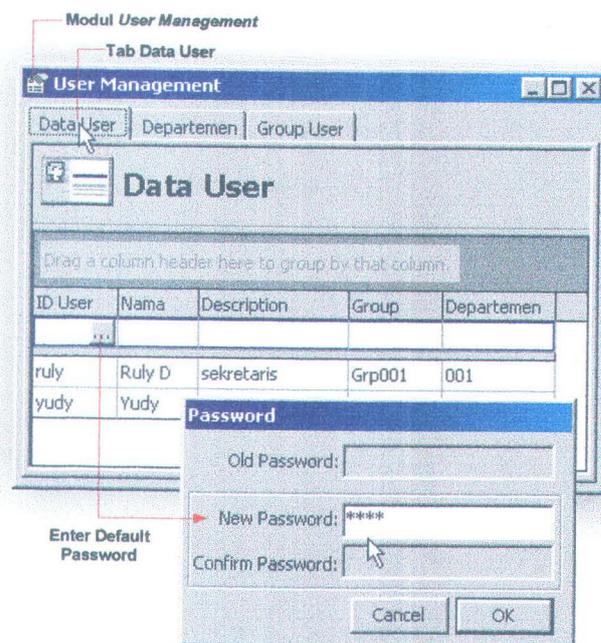
6.2.1 Pembuatan Modul User Management

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.



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	embah

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.

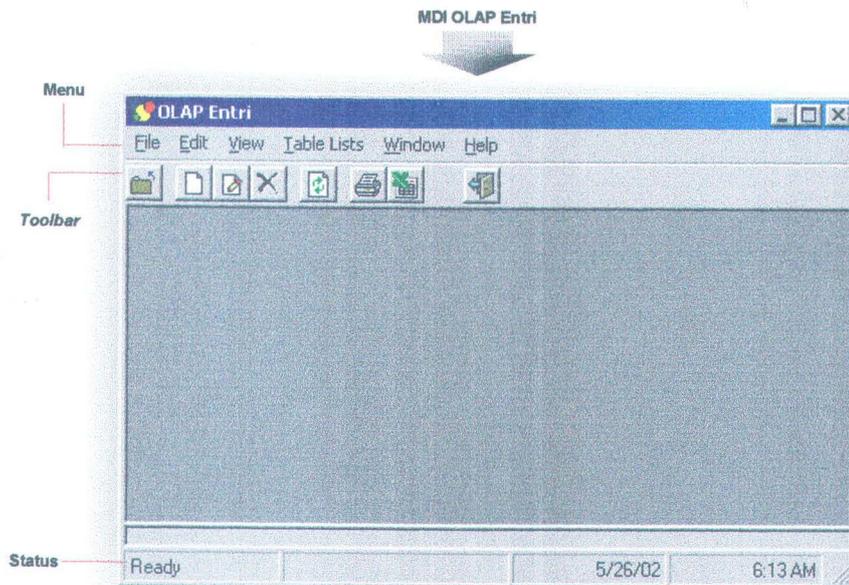
A screenshot of a Windows-style dialog box titled "Password". It contains three text input fields: "Old Password:", "New Password:", and "Confirm Password:". At the bottom of the dialog, there are two buttons: "Cancel" and "OK".

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**

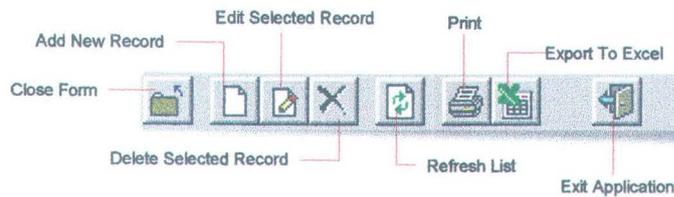
MDI OLAP Entri memiliki beberapa menu yang akan dirinci dalam tabel berikut:

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

<i>Menu</i>	<i>List</i>	<i>Keterangan</i>
	Cascade	Pengaturan window bertumpuk
	Close All	Menutup semua window
	Minimize All	Meminimasi semua window
<i>Help</i>	About	Tentang OLAP Entri

▪ *Toolbar*



Gambar 6.6 Toolbar MDI OLAP Entri

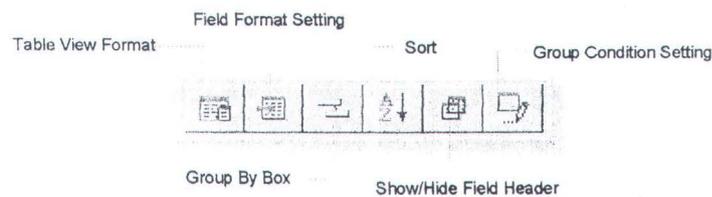
Gambar Icon pada MDI OLAP Entri akan digambarkan dan dijelaskan seperti pada Gambar 6.5 dan Tabel 6.2:

Tabel 6.2 Fungsi Icon-icon pada Toolbar MDI OLAP Entri

<i>Nama Icon</i>	<i>Fungsi</i>
<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 kustomisasi 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*

<i>Nama Icon</i>	<i>Fungsi</i>
<i>Table View Format</i>	Digunakan untuk melakukan setting pada tabel, misalnya pemilihan font.
<i>Field Format Setting</i>	Digunakan untuk merubah tampilan type data dari field tanpa mempengaruhi database
<i>Group By Box</i>	Digunakan jika ingin melakukan grouping pada data berdasarkan field.
<i>Sort</i>	Digunakan untuk melakukan multisort dari field yang ada.
<i>Show/Hide Field Header</i>	Digunakan untuk menampilkan/ menyembunyikan field-field yang diinginkan.

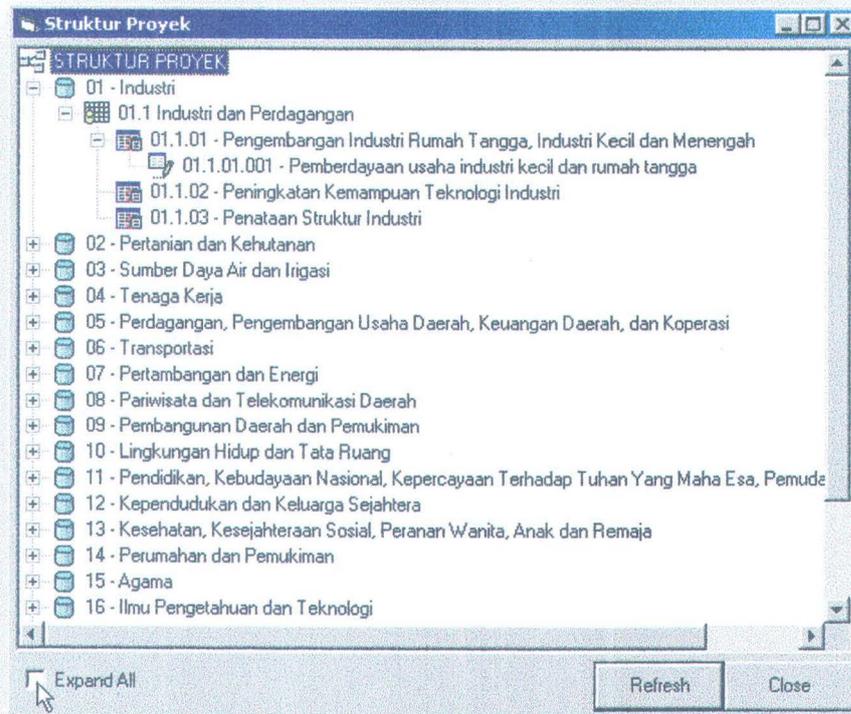
- **Status**

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)

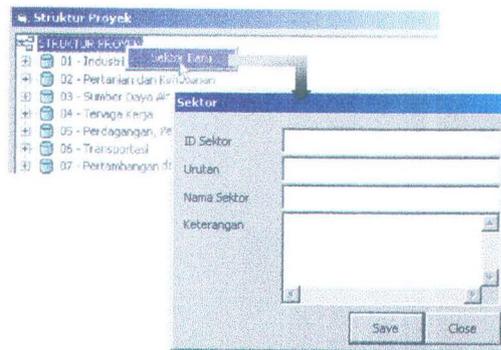
'//SELECT Proyek
rs4.Open "SELECT Proyek, Urutan FROM TbProyek _
WHERE IDSektor=" & rs1("IDSektor") & _
" AND IDSubSektor=" & rs2("IDSubSektor") _
& " AND Program=" & rs3("Program") & _
"", ConnMain, adOpenForwardOnly, _
adLockReadOnly
Do While Not rs4.EOF
Set nodex =
TreeView1.Nodes.Add(rs1("IDSektor")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

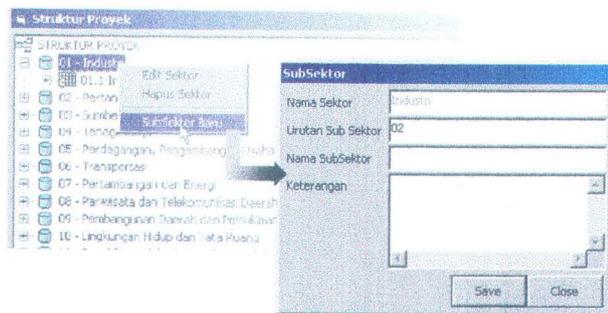
Add/Edit/Delete data Sektor dilakukan dengan *click* kanan pada *node* Struktur Proyek. *Event click* ini akan menampilkan form Sektor. Isian Urutan akan dipergunakan sebagai kode sebuah Sektor yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree*.



Gambar 6.9 Form Entri Sektor

▪ Pembuatan form entri Sub Sektor

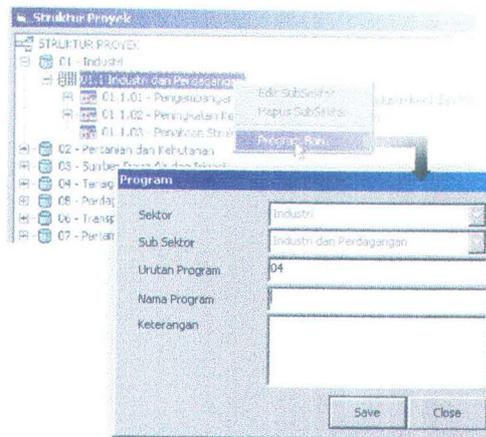
Add/Edit/Delete data SubSektor dilakukan dengan *click* kanan pada *node* tipe Sektor. *Event click* ini akan menampilkan *form* Sub Sektor. Isian Urutan akan dipergunakan sebagai kode sebuah Sub Sektor yang didefinisikan *user*, dan urutan ini akan ditampilkan pada struktur *tree* setelah sebelumnya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

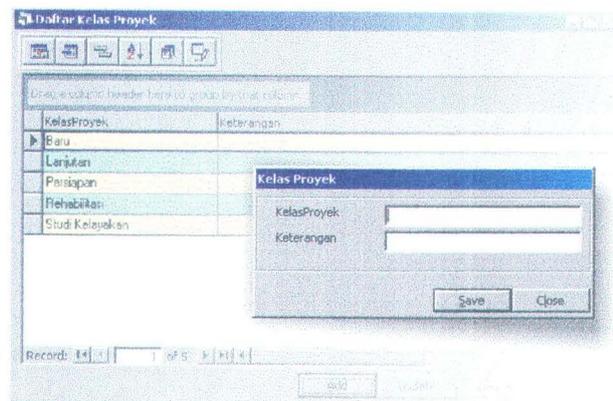
Tab Service Coverage dan Target Proyek

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.

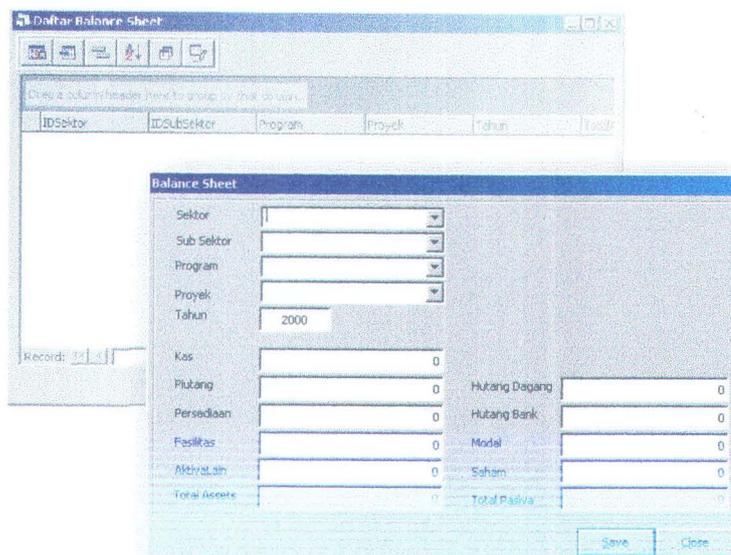
6.2.2.2 Pembuatan *Form* Entri Kelas Proyek

Entri Kelas Proyek akan digunakan sebagai referensi oleh Proyek untuk menentukan kategori dari suatu proyek. Pemasukan data Kelas Proyek melalui form berikut ini:



Gambar 6.13 *Form* Entri Kelas Proyek

6.2.2.3 Pembuatan *Form* Entri Balance Sheet



Gambar 6.14 *Form* Entri Balance Sheet

Balance Sheet digunakan untuk menunjukkan posisi keuangan sebuah proyek pada suatu periode tertentu. Entri di lajur sebelah kiri merupakan pencatatan Harta dan entri yang berada pada lajur sebelah kanan dipergunakan untuk mencatat Kewajiban dan Modal. Pencatatan tersebut disusun berdasarkan urutan kemampuan kepemilikan untuk digunakan sewaktu-waktu dibutuhkan.

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

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

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

Daftar Benefit Cost

Detail in Month

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
Mai	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

Daftar Cash Flow

Detail in Month

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
Mai	0	0	0	0
Juni	0	0	0	0
Juli	0	0	0	0
Agustus	0	0	0	0

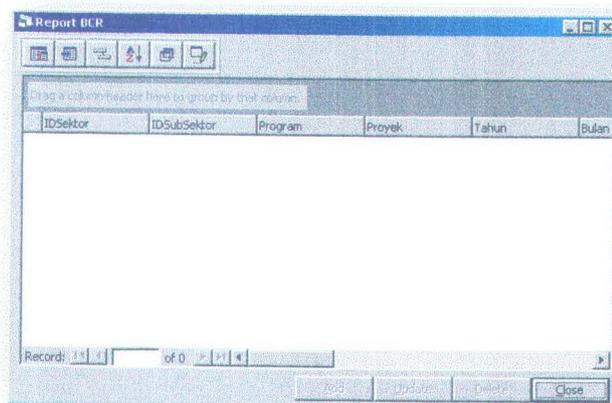
Save Close

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 & "'"
    End With
```

```

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
```

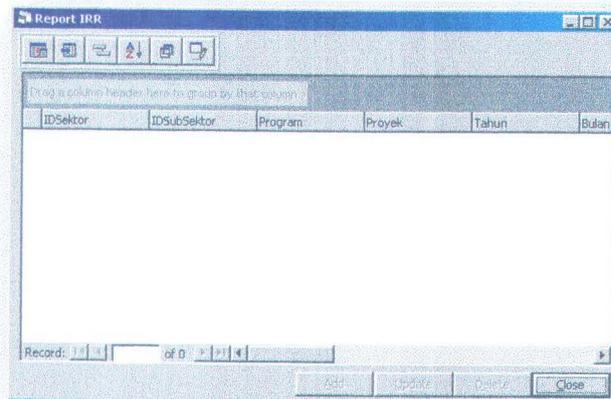
```
Exit Function
```

```
ElseIf NPVTotal > 0 Then
```

```

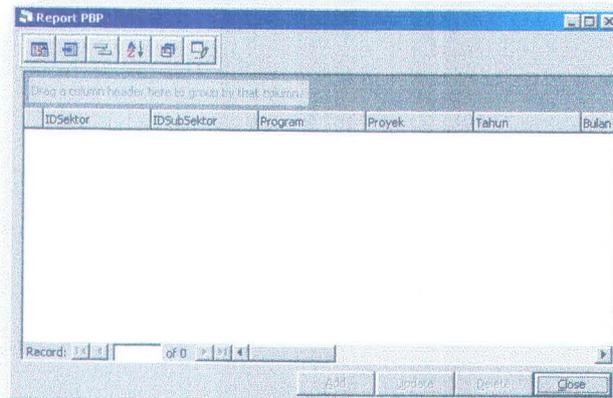
CalcIRR = IRR - ((NPVTotal * (IRR - IRRx)) / _
(NPVTotal - NPVTotalx)) / _
Exit Function
ElseIf NPVTotal < 0 Then
IRRx = IRR
NPVTotalx = NPVTotal
IRR = IRR + 1
GoTo ULANG
End If
End Function

```



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

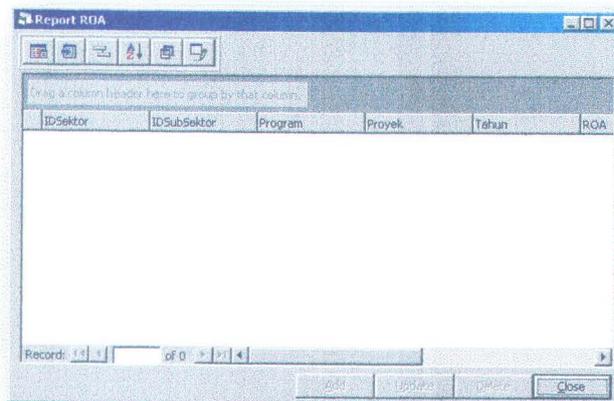
```

```

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 & ")"

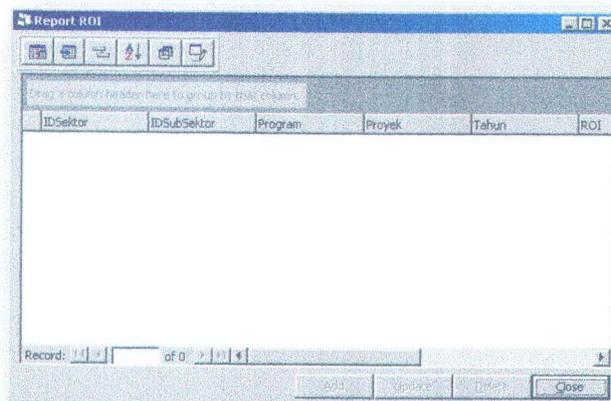
.MoveNext
Loop
End With

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,IDSSubSektor,Program,Proyek, Tahun,
NetIncome,SUM(Investment) AS SumInvestment " & "FROM
HitungROI Group BY Tahun,IDSektor,IDSSubSektor,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 " & "IDSSubSektor=" & !IDSSubSektor & "
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,
IDSSubSektor,Program,Proyek,Tahun," & "NetIncome,
Investment,ROI) VALUES (" & !IDSektor & "," &
!IDSSubSektor & "," & !Program & "," & !Proyek & "," &
!Tahun & ", (" & !NetIncome & "), (" & !suminvestment &
"), (" & i & ")")

        .MoveNext
    Loop
End With

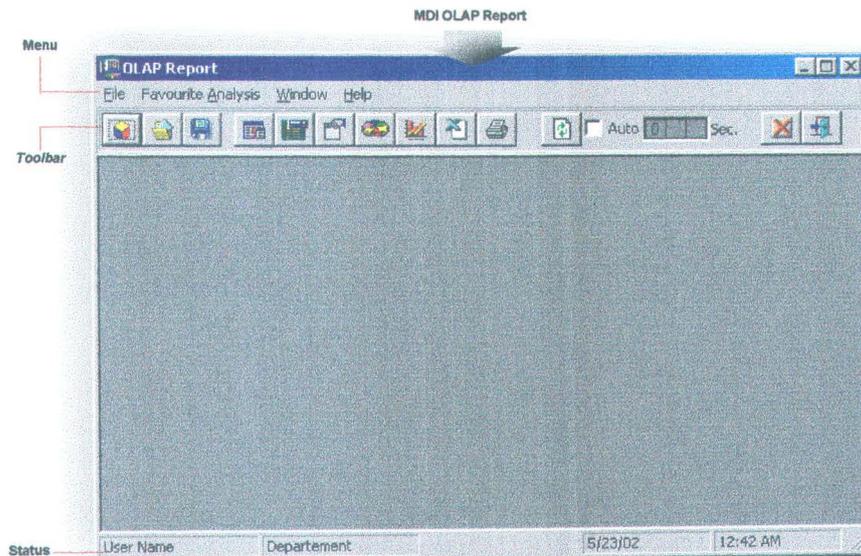
Set cat = Nothing
Set cmd = Nothing
Set rs = Nothing
Exit Sub

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

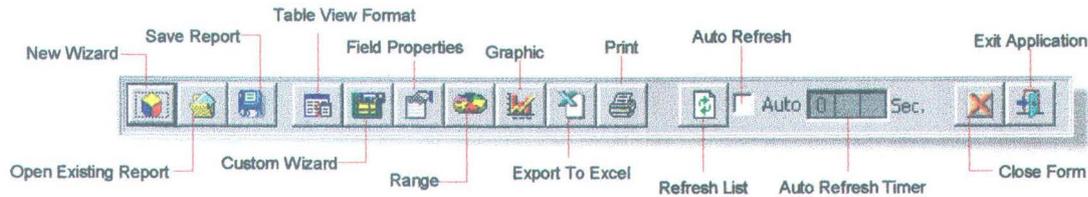
MDI OLAP Report memiliki beberapa menu yang akan dirinci dalam tabel berikut:

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- <i>save</i> dengan status <i>Favourite Analysis</i> = TRUE
Window	Tile Horisontally	Pengaturan window horisontal

<i>Menu</i>	<i>List</i>	<i>Keterangan</i>
	Tile Vertically	Pengaturan window vertikal
	Cascade	Pengaturan window bertumpuk
<i>Help</i>	About	Tentang OLAP Report

▪ *Toolbar*



Gambar 6.24 Toolbar OLAP Report

Tabel 6.5 Fungsi Icon-icon pada Toolbar OLAP Report

<i>Nama Icon</i>	<i>Fungsi</i>
<i>New Wizard</i>	Membuat <i>Report</i> baru dengan wizard
<i>Open Existing Report</i>	Membuka <i>Report</i> lama yang telah dibuat
<i>Save Report</i>	Menyimpan <i>Report</i> yang baru dibuat
<i>Table View Format</i>	Digunakan untuk melakukan setting pada tabel, misalnya pemilihan font
<i>Custom Wizard</i>	Digunakan untuk melakukan konfigurasi ulang terhadap dokumen analisis yang aktif.
<i>Field Properties</i>	Digunakan untuk meng-custom field-field pada <i>Report</i>
<i>Range</i>	Melakukan pewarnaan (<i>colouring</i>) pada field
<i>Graphic</i>	Memindahkan <i>Report</i> dalam bentuk grafik
<i>Export To Excel</i>	Mengirim data ke Excel
<i>Print</i>	Mencetak data
<i>Refresh List</i>	Refresh data
<i>Auto Refresh</i>	Refresh data otomatis berdasarkan setting <i>timer</i>
<i>Auto Refresh Timer</i>	<i>Timer</i> untuk melakukan Auto Refresh
<i>Close Form</i>	Menutup <i>report</i> yang aktif
<i>Exit Application</i>	Keluar dari aplikasi OLAP <i>Report</i>

- **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:

```

...
ClQueryFrame1.Item.MoveToRoot
If ClQueryFrame1.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:

```

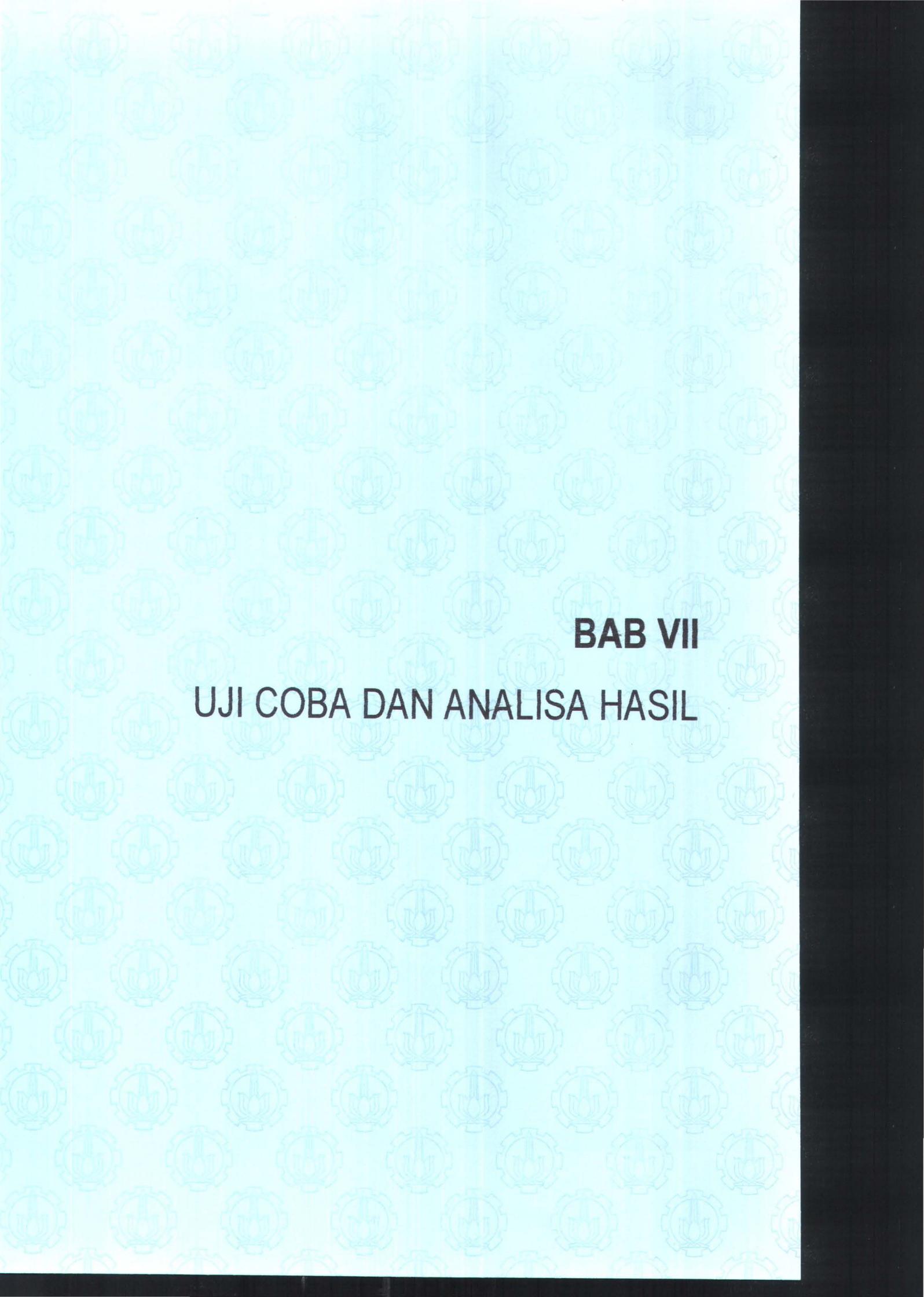
...
For Each rng In Ranges
    If cellAttrib.value < rng.EndRange And cellAttrib.value >= _
        rng.StartRange Then
        cellAttrib.ForeColor = rng.Color
        Exit Sub
    End If
Next
...

```

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

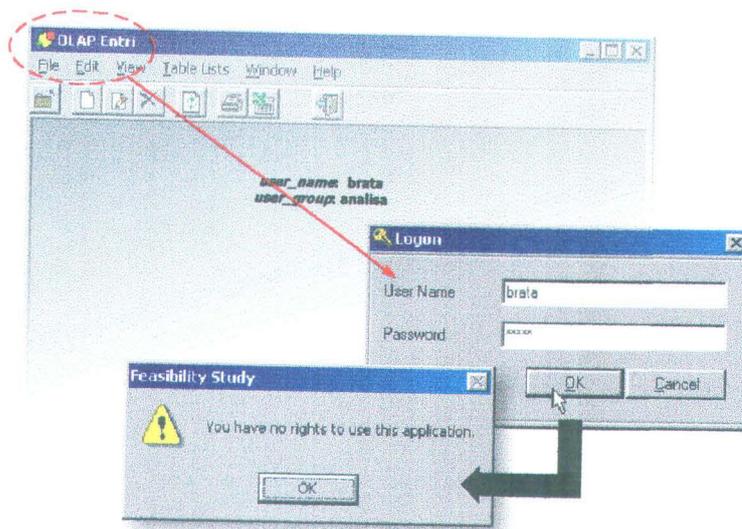
- AMD Duron 1 GHz
- RAM 128 MB
- 20 GB harddisk

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).

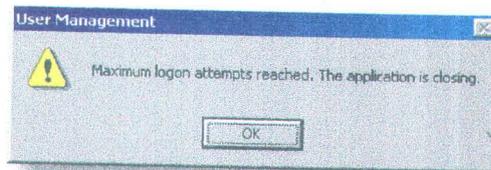


Gambar 7.1 Peringatan *user group* tidak memiliki hak akses terhadap sebuah modul

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



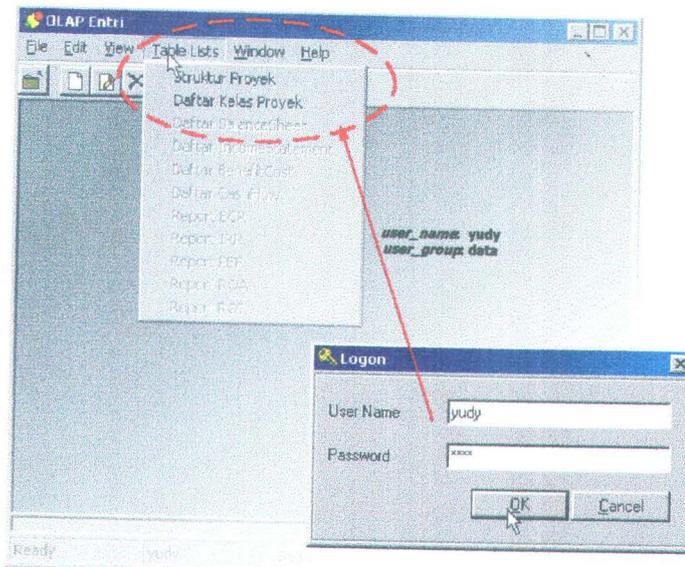
Gambar 7.3 Peringatan penutupan aplikasi segera oleh sistem

7.2.2 Uji Coba *User Management*

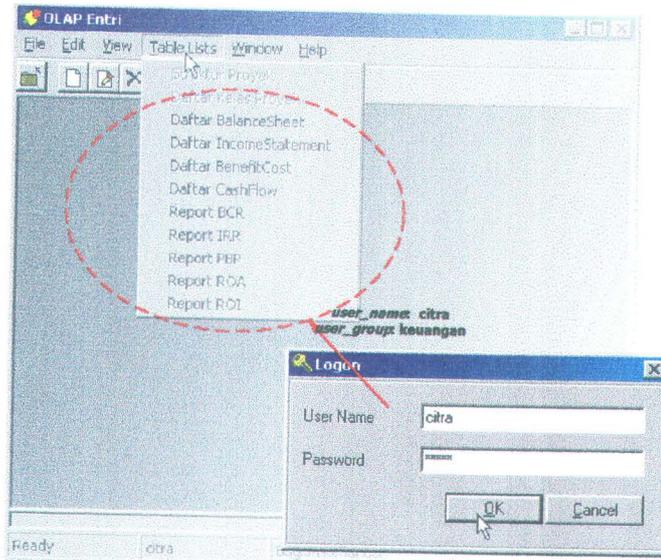
Uji coba ini dilakukan untuk menguji modul *User Management* yang telah dibuat. Modul *user management* ini hanya dapat diakses oleh *user group* Admin. Pengujian yang dilakukan meliputi penambahan *user* baru, *edit* data *user* dan penghapusan hak akses *user*. Seorang *user* bisa didaftarkan untuk 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 Statement, Benefit Cost dan Cash Flow, serta *report-report* keuangan yang ada.

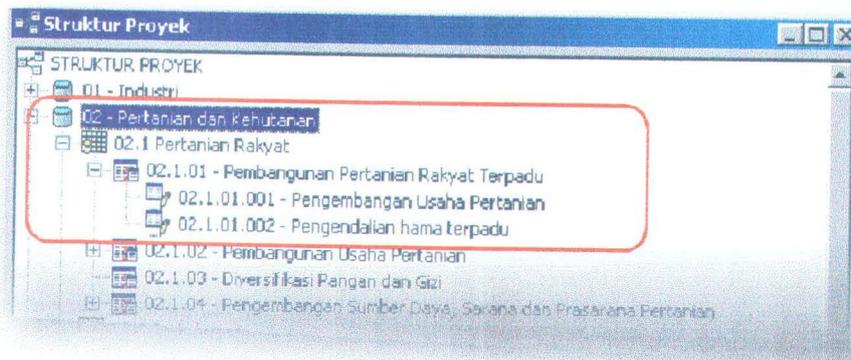
Penggambaran menu OLAP Entri yang dimanajemen berdasarkan hak akses group user, digambarkan pada gambar 7.4 dan 7.5.

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
PeggunaLayanan

Custom Field:
Formula1

SQL Select String:

```
SELECT [KelasProyek], [Proyek], [Sektor], [JumlahTKI], [PeggunaLayanan], ([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.

The screenshot shows a window titled 'UjiCoba1' with a data table. The table has columns for 'KelasProyek', 'JumlahTKI', 'PeggunaLayanan', and 'Formula1'. The data is as follows:

KelasProyek	JumlahTKI	PeggunaLayanan	Formula1	JumlahTKI	PeggunaLayanan	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

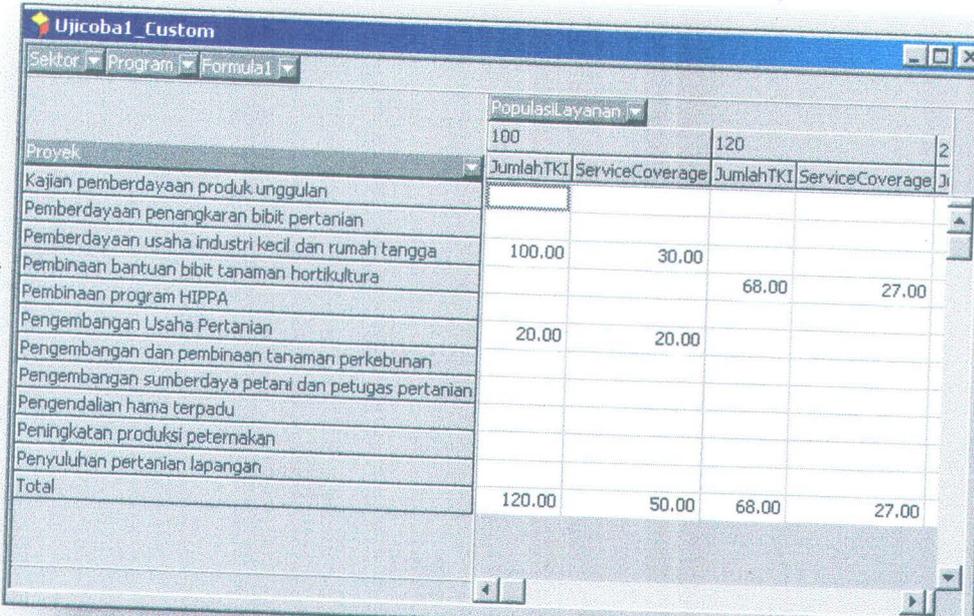
Custom Field:

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 containing "Sektor", "Program", and "Formula1". Below the menu bar is a table with the following data:

Proyek	PopulasiLayanan, 100		120		2
	JumlahTKI	ServiceCoverage	JumlahTKI	ServiceCoverage	
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 software application window titled "Ujicoba1_Custom". At the top, there are dropdown menus for "Sektor", "Program", and "Formula1". Below these, there 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 7.9 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.

Ujicoba1_Custom_FP

Sektor Program Formula1

Sort Type

Jumlah Penduduk

100

120

2

Proyek

Proyek	JumlahTKI	ServiceCoverage	JumlahTKI	ServiceCoverage
Pengembangan Usaha Pertanian	20.00	40.00%		
Pengendalian hama terpadu				
Pemberdayaan penangkaran bibit pertanian				
Kajian pemberdayaan produk unggulan				
Pembinaan program HIPPA				
Pengembangan dan pembinaan tanaman perkebunan				
Penyuluhan pertanian lapangan				
Peningkatan produksi peternakan				
Pengembangan sumberdaya petani dan petugas pertanian				
Pembinaan bantuan bibit tanaman hortikultura			68.00	100.00%
Pemberdayaan usaha industri kecil dan rumah tangga	100.00	60.00%		
Total Proyek	120.00	100.00%	68.00	100.00%

Display Type

Group Footer

Gambar 7.10 Hasil Uji Coba Manipulasi dengan Fields Properties

Uji Coba Penggambaran Grafik

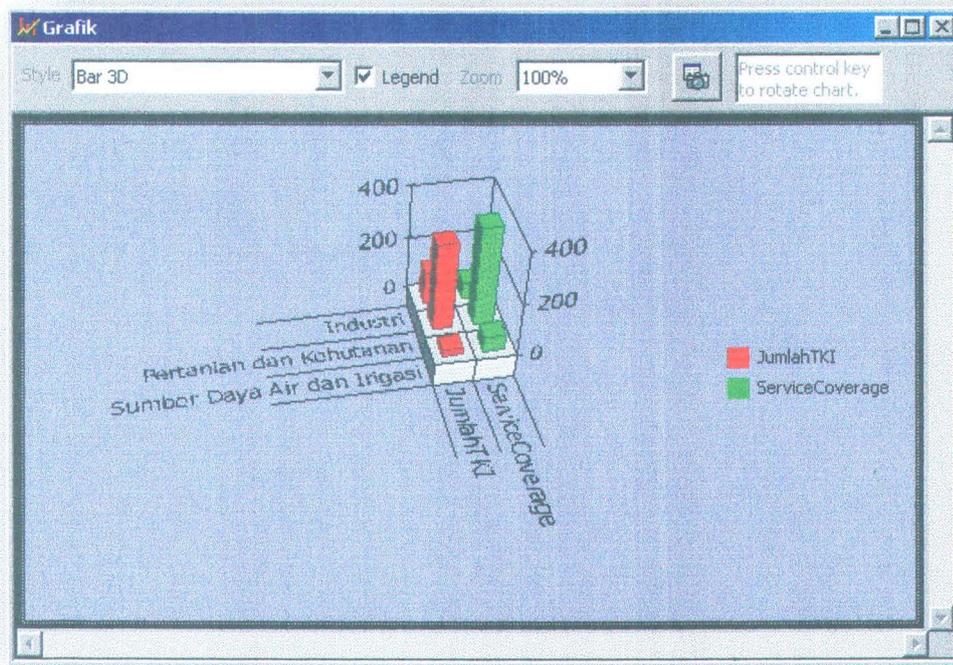
Uji coba penggambaran grafik dilakukan dengan melakukan pemilihan sekumpulan sel data pada dokumen analisis Ujicoba1_Custom. Pemilihan ini dilakukan dengan mengadakan perubahan orientasi dari beberapa field yang ada. Dari hasil pemilihan ini sel pada gambar 7.11, maka didapatkan gambar grafik seperti pada gambar 7.12.

Formula1 Proyek Program Populasi syanan

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

Sel data grafik

Gambar 7.11 Pemilihan Sel Data Grafik



Gambar 7.12 Hasil Uji Coba Penggambaran Grafik

Uji Coba Export Data ke Excel Workbook

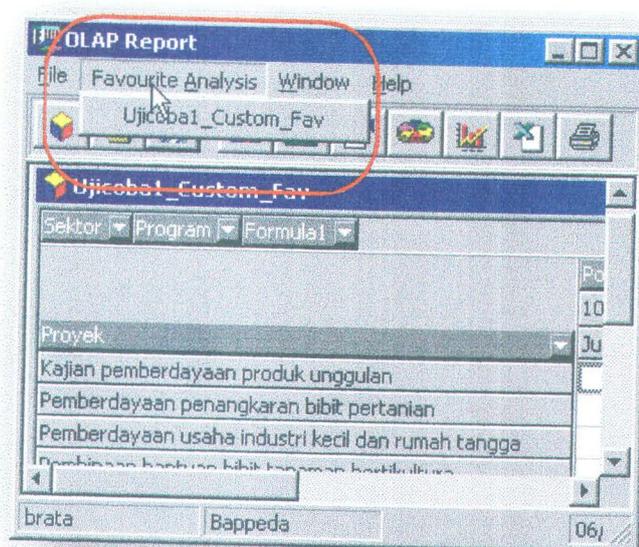
Proyek	Data	43	50	54	55	66	76
Kajian pemberdayaan produk ungg	Sum of JumlahTKI					25	
	Sum of ServiceCoverage					36	
Pemberdayaan penangkaran bibit	Sum of JumlahTKI						24
	Sum of ServiceCoverage						45
Pemberdayaan usaha industri kec	Sum of JumlahTKI						
	Sum of ServiceCoverage						
Pembinaan bantuan bibit tanaman	Sum of JumlahTKI						
	Sum of ServiceCoverage						
Pembinaan program HIPPA	Sum of JumlahTKI	26					
	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*.



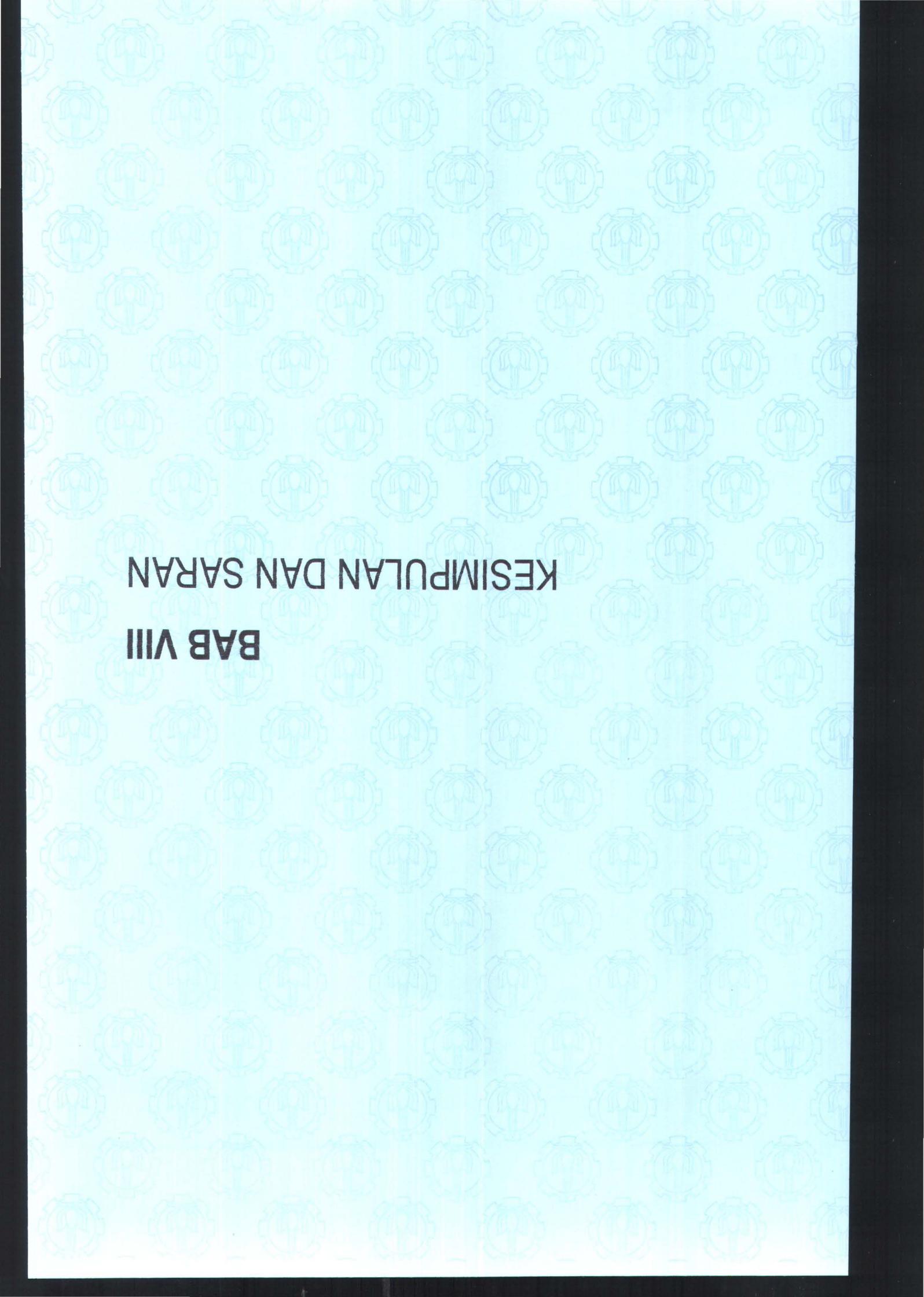
Gambar 7.14 Hasil Uji Coba Penyimpanan Dokumen Favorite Analysis

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

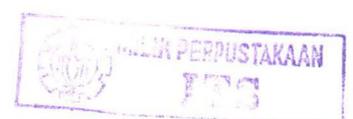
8.1 Kesimpulan

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

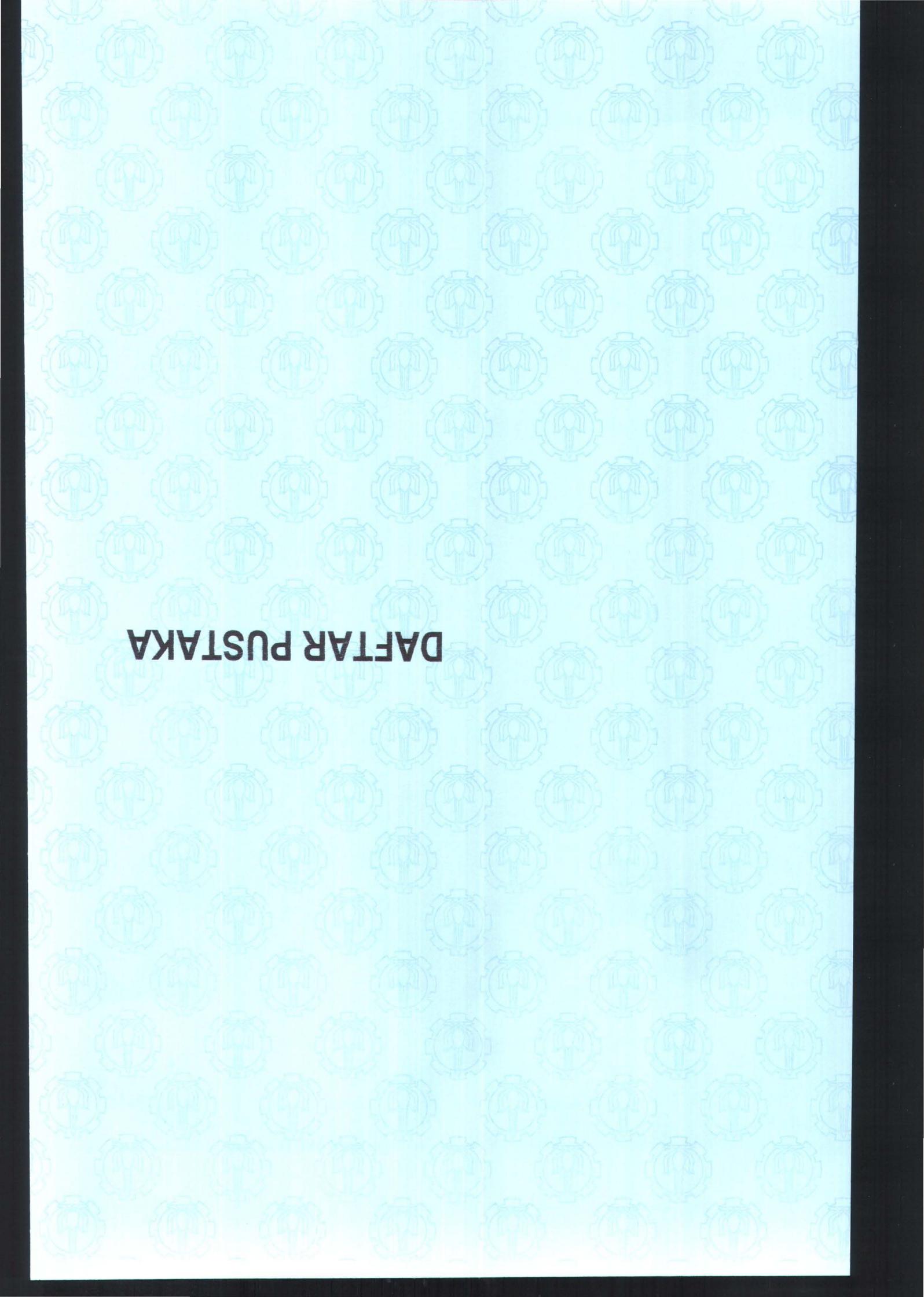
Saran bagi pengembangan aplikasi *Sistem Feasibility Study* yang menerapkan teknologi OLAP untuk proses investasi yang dikerjakan pada pembuatan tugas akhir ini antara lain adalah:



- 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.

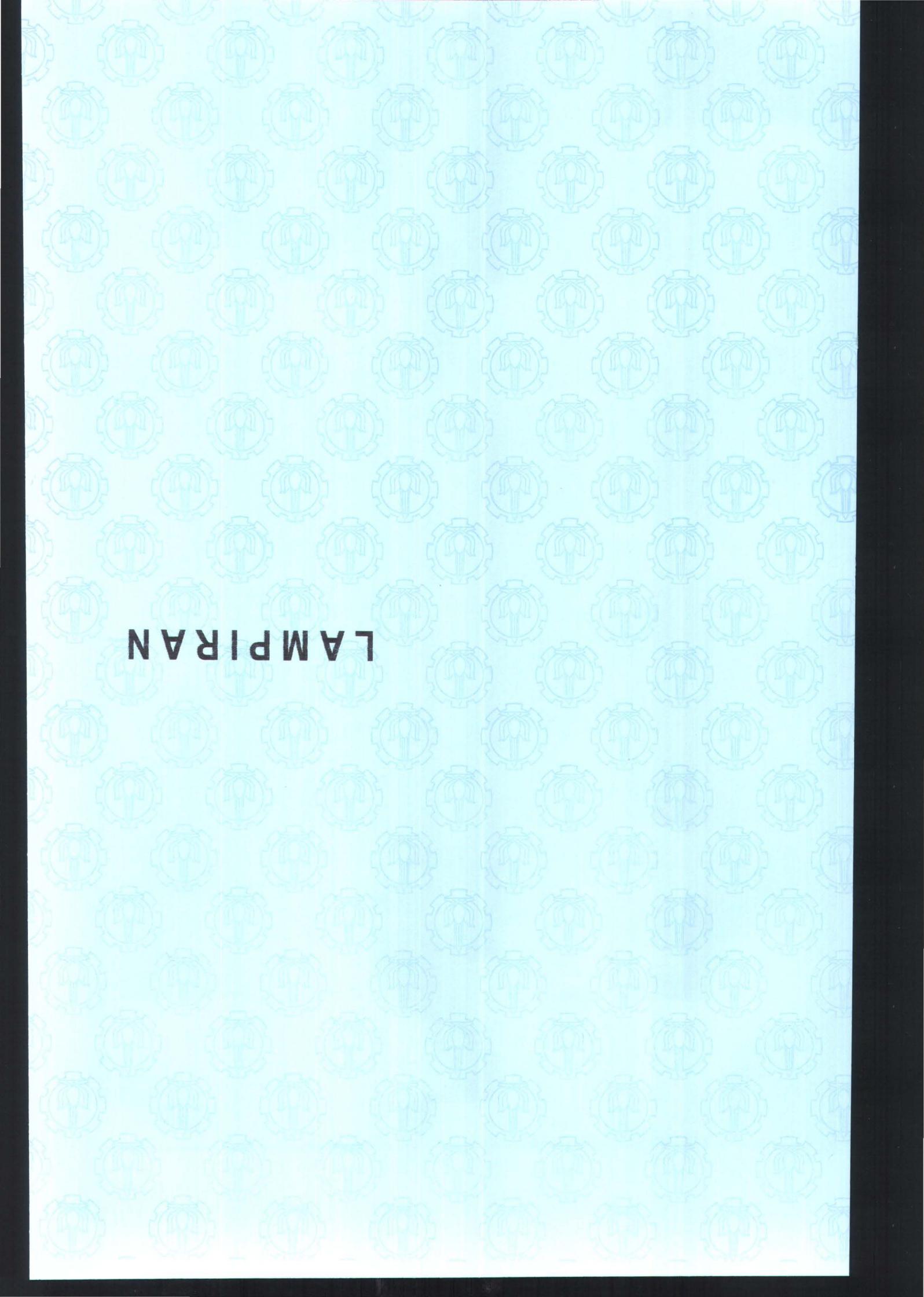
Oleh sebab itu, hal utama yang disarankan bagi pihak Bappeda adalah pengintegrasian komputer-komputer tersebut dalam sebuah sistem yang terintegrasi dengan baik, sehingga diharapkan Bappeda akan benar-benar siap dalam mengejar ketinggalan teknologi khususnya dibidang Teknologi Informasi. Dengan adanya 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

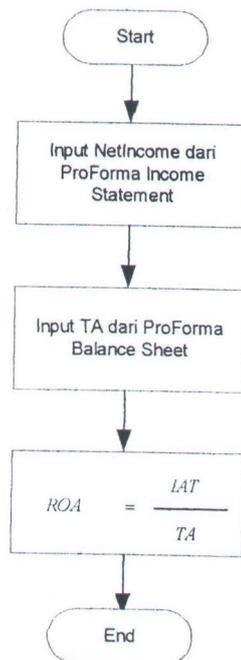
Desain Algoritma Perhitungan Indikator Keuangan

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

$$\text{rumus: } ROA = \frac{\text{NetIncome}}{TA}$$



Gambar A.1 Flowchart Algoritma Perhitungan ROA

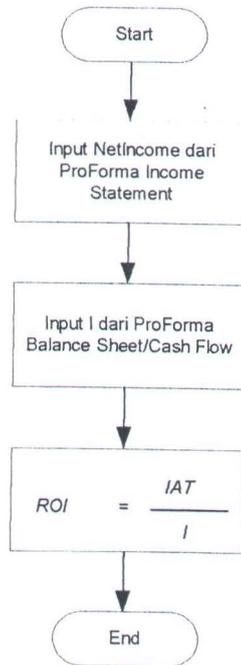
A.2 Algoritma Perhitungan ROI

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 :

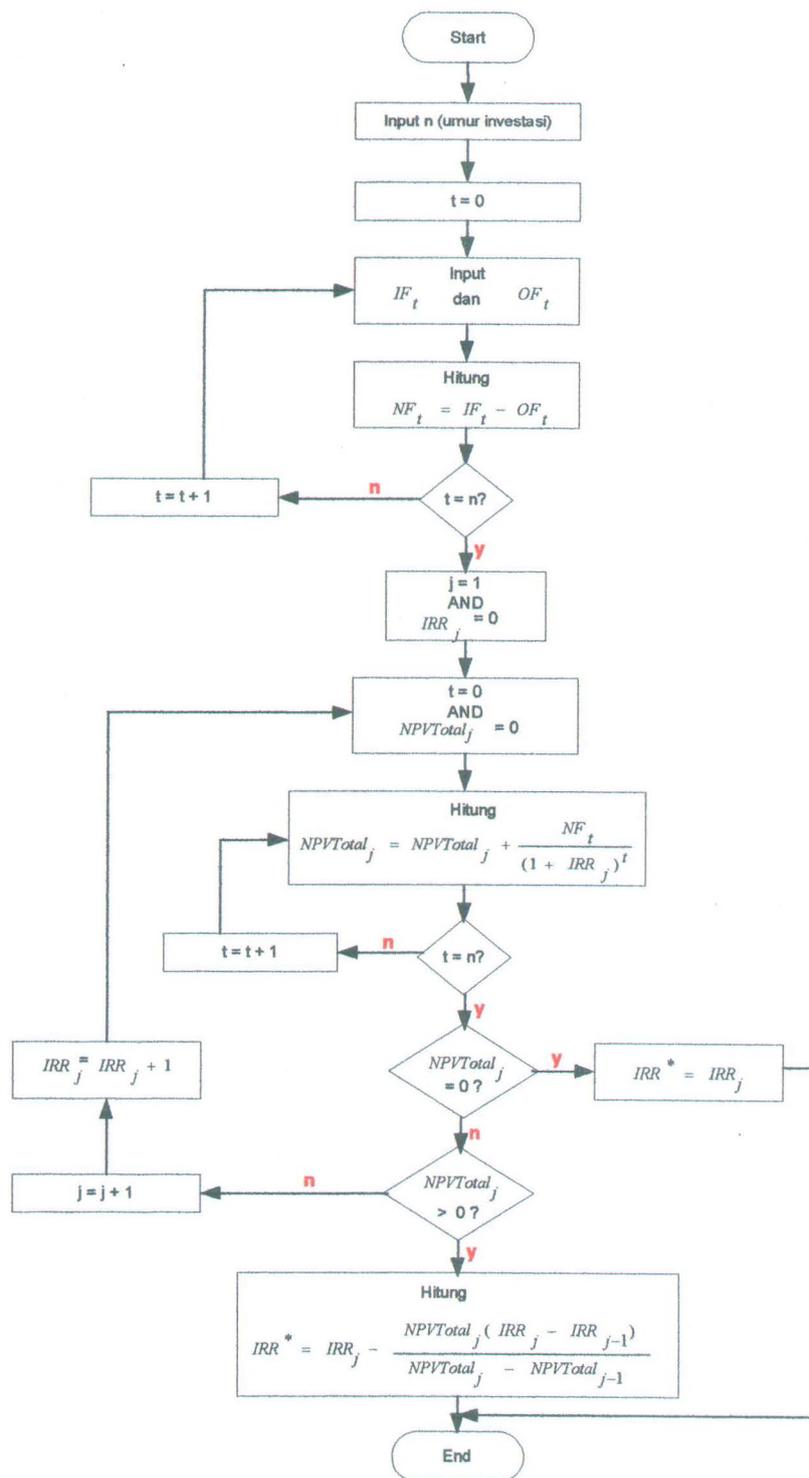
- 1) Masukkan umur proyek (n) dalam satuan bulan.
- 2) Masukkan Inflow (IF) dan Outflow (OF) untuk setiap bulannya sepanjang umur proyek tersebut .
- 3) Hitung Netflow (NF_t) untuk setiap bulannya sepanjang umur proyek tersebut. Mulai dari $t = 0$ sampai dengan n, hitung

$$NF_t = IF_t - OF_t$$

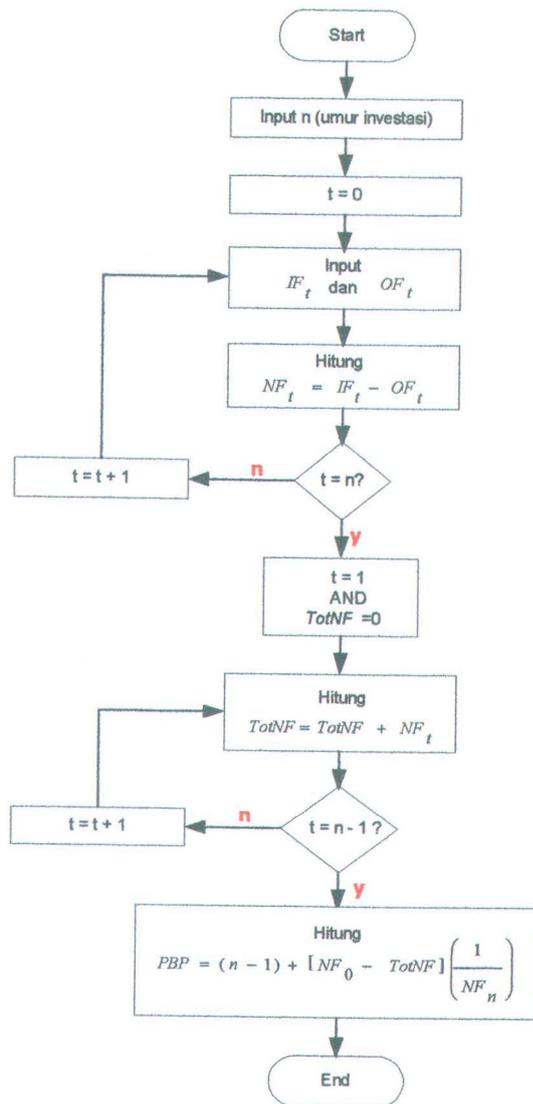
Biaya investasi yang pertama adalah NF_0 .

- 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.

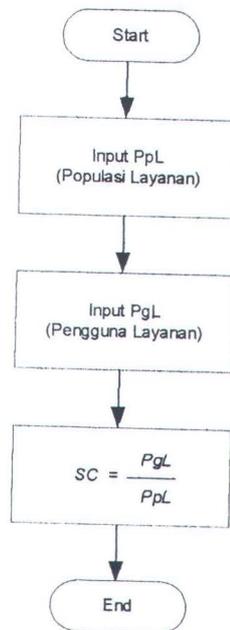
Nilai PL ini akan sangat tergantung pada : jenis layanan, dan wilayahnya. Misalkan: untuk investasi bangunan SD Inpres di kecamatan A, maka yang disebut populasi layanan (PpL) adalah jumlah anak yang berumur dari 7 – 12 tahun di kecamatan A.

- 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:

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

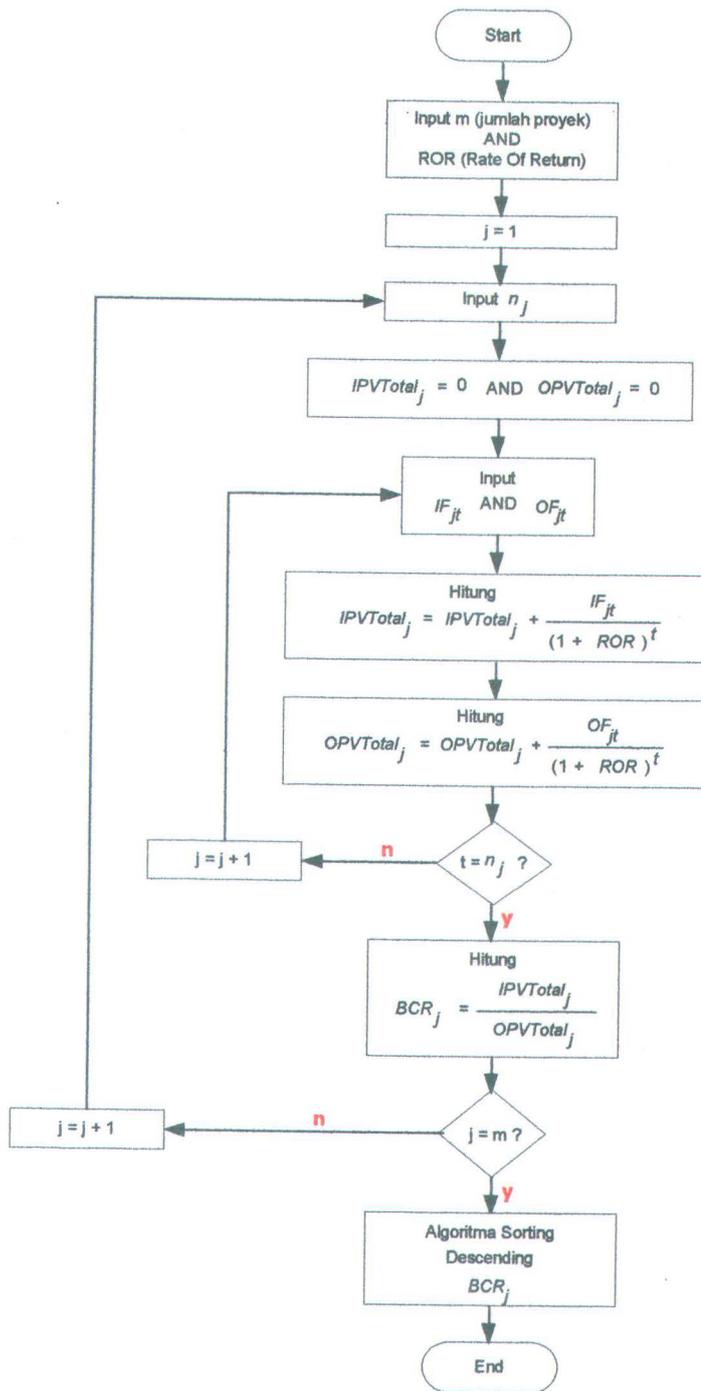
- 6) Lakukan perhitungan juga Total Nilai Sekarang atau Outflow Present Value Total (OPVTotal) untuk Outflow proyek j . Perhitungan $OPVTotal_j$ adalah:

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

- 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>Irigasi</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

05.1	<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
05.2	<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
05.3	<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
05.4	<i>Koperasi dan Pengusaha Kecil</i>
05.4.01	Pembinaan dan Pengembangan Koperasi
05.4.02	Pembinaan dan Pengembangan Usaha Kecil
06	Transportasi
06.1	<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
06.2	<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
07.1	<i>Pertambangan rakyat</i>
07.1.01	Pembangunan pertambangan
07.1.02	Pembangunan Sistem Informasi Geologi/Geografi
07.2	<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
08.1	<i>Pariwisata</i>
08.1.01	Pengembangan Pariwisata
08.1.02	Pengembangan dan Pengelolaan produk/obyek wisata
08.2	<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	Pengeralahan 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 kedinasan</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, kesusateraan dan kepustakaan
11.3.03	Pembinaan kesenian daerah
11.3.04	Pembinaan tradisi, peninggalan sejarah dan museum
11.3.05	Pembinaan penganut kepercayaan terhadap Tuhan Yang Maha Esa
<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 pembnagunan 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
<i>A.1</i>	<i>Bantuan pembangunan Dati I</i>
A.1.01	Subsidi Bantuan pembangunan Dati I