26.369/H/06

MILIK PERPUSTAKAAN
ITS

PENGAPLIKASIAN ORACLE WAREHOUSE BUILDER 10G UNTUK MEMBANGUN OLAP DATA WAREHOUSE STUDI KASUS : FRS ONLINE ITS

TUGAS AKHIR

RSIf
005.756 5
Wah
p-1
2006

Disusun Oleh :

ABDUL WAHIB
5199 100 037

PERPUSTAKAAN ITS	
Tgl. Terima	20-2-06
Terima Dari	H
No. Agenda Prp.	724231

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

PENGAPLIKASIAN ORACLE WAREHOUSE BUILDER 10G UNTUK MEMBANGUN OLAP DATA WAREHOUSE STUDI KASUS : FRS ONLINE ITS

TUGAS AKHIR

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

Mengetahui / Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Febriliyan Samopa, S.Kom., M.Kom.	**Darlis Heru Murti, S.Kom.**
NIP. 132.206.858	**NIP. 132.306.430**

SURABAYA
JANUARI, 2006

ABSTRAK

Berkembangnya jumlah data pada database akan semakin menambah beban mesin, untuk itu diperlukan spesifikasi mesin yang tinggi untuk menangani tugas ini. Semakin canggih mesin akan semakin mahal harganya. Berangkat dari ide untuk memperkecil biaya, muncullah teori Data Warehouse oleh W.H Inmon, bapak Data Warehouse. Pada awalnya belum ada aplikasi yang mampu membantu *user* membangun sebuah Data Warehouse mulai dari nol. Baru kemudian bermunculan aplikasi dengan tujuan ini, salah satunya adalah Oracle Warehouse Builder.

Oracle Warehouse Builder (OWB) mampu membantu user mulai dari proses ekstraksi, transformasi sampai proses *loading* data, sehingga menjadi sebuah Data Warehouse. Tugas Akhir ini mencoba untuk membuat sebuah Data Warehouse dengan memakai OWB. Mendesain alur-alur ekstraksi, mentransformasi data, dan me-*load*-nya ke dalam Data Warehouse. Dan sekaligus mengintegrasikannya dengan teknologi OLAP untuk merepresentasikan data.

Uji coba pada aplikasi ini adalah operasi-operasi OLAP yaitu *drill-down*, *consolidation*, *slice/dice*, dan juga operasi *rotation/pivoting*, dengan bantuan aplikasi lain yaitu Analytic Warehouse Manager.

Kata kunci: OLAP, Data Warehouse, ETL, Oracle Warehouse Builder

KATA PENGANTAR

Segala puji syukur ke hadirat Allah SWT, karena hanya dengan kehendak dan kuasa-Nya, Penulis dapat menyelesaikan Tugas Akhir yang berjudul **"PENGAPLIKASIAN ORACLE WAREHOUSE BUILDER 10G UNTUK MEMBANGUN OLAP DATA WAREHOUSE. STUDI KASUS:FRS ONLINE ITS".**

Tugas Akhir dengan beban 4 SKS ini disusun dan diajukan sebagai salah satu syarat untuk menyelesaikan program Strata Satu (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Semoga dengan selesainya Tugas Akhir ini, dapat membantu dalam proses pengembangan teknologi informasi di jurusan. Penulis menyadari bahwa Tugas Akhir ini jauh dari sempurna, masih banyak kekurangan-kekurangan. Oleh karena itu sangat diharapkan kritik dan saran yang membangun untuk perbaikan kedepan.

Akhirnya, penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu terselesaikannya Tugas Akhir ini.

Surabaya, Januari 2006

Penulis

UCAPAN TERIMA KASIH

Syukur Alhamdulillah, Penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala yang telah memberi begitu banyak nikmat dan kemudahan kepada penulis, terlebih pada pengerjaan Tugas Akhir ini.

Shalawat serta salam tak lupa Penulis ucapkan untuk junjungan Nabi Muhammad Salallahu Alaihi Wassalam.

Pada kesempatan ini, Penulis ucapkan penghormatan setinggi-tingginya serta rasa terima kasih kepada pihak-pihak yang telah membantu Penulis dalam pengerjaan Tugas Akhir ini baik moril maupun material kepada:

1. Almarhum Bapak (*Allahummaghfirlahu*), terimakasih atas segala bimbingan dan kasih sayangnya, semoga Allah membalasnya dengan surga. Juga Ibu penulis yang dengan sabar membimbing dan mendoakan, semoga Allah memberikan kesehatan dan memberkahi sisa umur serta berkenan memasukkan Ibu dalam golongan orang-orang yang husnul khotimah.
2. Bapak Febriliyan Samopa, S.Kom, M.Kom dan Bapak Darlis Heru Murti, S.Kom sebagai dosen pembimbing, atas bimbingan dan kesediaannya meluangkan waktu untuk mendengar keluh kesah Penulis selama pengerjaan Tugas akhir ini. Semoga Allah membalasnya dengan balasan yang setimpal.
3. Bapak Yudhi Purwananto, S.Kom, M.Kom selaku Ketua Jurusan Teknik Informatika, ITS; dan Bapak Prof. Ir. Arif Djunaidy, M.Sc, Ph.D, selaku Dekan Fakultas Teknologi Informasi, ITS.
4. Bapak Ir. Muhammad Husni, M.Kom selaku Dosen Wali Penulis.

5. Seluruh Dosen Fakultas Teknologi Informasi ITS, terima kasih atas segala ilmu yang diberikan selama masa perkuliahan.
6. Saudara-saudariku, Mas Diq, Mas Jik dan Mba' Iqo' (semoga segera dipercaya untuk punya momongan), Neng Jiyah dan Mas Imam (semoga bisa *'ngasih* ponakan yang lucu dan sholeh/sholihah), Neng Us, Neng Jijah, Neng Sih dan Mas Hasan, serta adikku tercinta Firman (mau 'kan jadi Dokter ? ☺). Terimakasih atas semangat dan kebersamaan yang membuat Penulis menjadi bisa menjalani hari-hari 'berat' dan melelahkan dengan penuh semangat.
7. Fajar, *I owe you a lot friend.* Edy "Mumun" dan Cherry, kapan kita main bulutangkis lagi? Meme, makasih bukunya, *they are very useful.* Om Priyo, Tante Rina juga Tante Fitri, makasih atas tiket gratisnya nonton di 21. ☺
8. Teman-teman C0F tercinta, Ade Reza, Pak dosen Shidiq, Hendra "Gumon", Nafis "Bang Ical", Dwi "Inuk", Rono "Rontog", Ardhi "Erzu", Maman, Medi, Adeta, Rifqi, Roy, Rahma, Nu'man (terima kasih suntikan semangatnya waktu sidang ☺), Kholid, Ian, dan semua anggota C0F yang tidak bisa Penulis sebutkan satu-persatu. *Bravo TC!!*
9. Teman-teman Admin Lab. IBS, Fauzan (*you've got the throne, at last!!*), Icoex, Galih, Mahmud "Zimux", Rile, Catur dan Febi. Juga para mantan Admin, Budi, Heri '98 dan Warna Agung. Terimakasih atas segala 'layanan' selama Penulis mengerjakan Tugas Akhir.
10. Teman-teman seperjuangan dalam pengerjaan Tugas Akhir, Yunan00, Dimas02, Yunan NSNT, Nu'man, Ian, Fajar, dan lain-lainnya.

11. Dan pihak-pihak yang telah membantu Penulis dalam menyelesaikan tugas Akhir ini, baik secara langsung maupun tidak langsung.

Tiada untaian kata yang dapat penulis sampaikan sebagai penghormatan dan balasan atas apa yang telah diberikan melainkan harapan dan doa semoga Allah Subhana Wa Ta'ala membalas semua amal dan kebaikan tersebut. Jazakumullah Khairan Katsiran.

DAFTAR ISI

ABSTRAK i
KATA PENGANTAR ii
UCAPAN TERIMA KASIH iii
DAFTAR ISI vi
DAFTAR GAMBAR viii
DAFTAR TABEL x
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Permasalahan 2
1.3 Batasan Permasalahan 2
1.4 Tujuan 3
1.5 Metodologi Pembuatan Tugas Akhir 3
1.6 Sistematika Penulisan 4
BAB II TEORI PENUNJANG 6
2.1 Data Warehouse 6
2.1.1 Extraction 9
2.1.2 Transportation 10
2.1.3 Transformation dan Loading 10
2.2 OLAP (On Line Analytical Processing) 12
2.2.1 Data Multidimensi 12
2.2.2 Operasi pada Data Multidimensi 13
2.2.3 Skema Data Pada OLAP 14
2.2.4 OLAP Metadata Model 16
2.3 Oracle Warehouse Builder (OWB) 19
2.3.1 Lingkungan Desain 20
2.3.2 Runtime Environment 20
BAB III PERANCANGAN DATA WAREHOUSE 24
3.1 Database Sumber 24
3.1.1 Pemodelan Data 24
3.2 Desain Skema Data Warehouse 25
3.2.1 Analisa Nilai Indeks Prestasi Semester (IPS) 26
3.2.2 Analisa Total Kredit Mengajar 27
3.3 Flowchart Data Warehouse 28
BAB IV IMPLEMENTASI ORACLE WAREHOUSE BUILDER 30
4.1 Menyiapkan Lingkungan Warehouse 30
4.1.1 Membangun Warehouse Builder Repository 30
4.1.2 Membuat Runtime Repository 32
4.2 Menyiapkan Oracle Transparent Gateway 35
4.2.1 Membuat File UDL (Universal Data Link) 35
4.2.2 Mengubah File inittg4msql.ora 36
4.2.3 Mengubah File listener.ora 37
4.2.4 Mengubah File tnsnames.ora 38
4.2.5 Mengubah File initholedb.ora 38

4.2.6 Membuat Database Link 38
4.3 Log in ke OWB 39
4.4 Membuat Project 41
4.4.1 Menentukan Lokasi Sumber 41
4.4.2 Menentukan Lokasi Tujuan 41
4.4.3 Membuat Modul Sumber (Ms SQL Server) 42
4.4.4 Mengimpor Obyek dari Sumber 43
4.4.5 Membuat Oracle Module 44
4.5 Membuat Dimensi 46
4.6 Membuat Cube 48
4.6.1 Cube Nilai 49
4.7 Membuat Mapping 51
4.7.1 Mapping Dimensi Waktu 51
4.7.2 Mapping Dimensi Mahasiswa 54
4.7.3 Mapping Cube Nilai 55
4.8 Membuat Alur Proses 57
4.9 Proses Penggunaan (Deploying) dan Eksekusi 60
4.10 Integrasi Dengan OLAP 63
4.10.1 Menggunakan Analytic Workspace Manager 63
BAB V UJI COBA DAN EVALUASI 68
5.1 Lingkungan Uji Coba 68
5.1.1 Server 68
5.1.2 Client 69
5.2 Uji Coba 69
5.2.1 *Drill-down* 69
5.2.2 *Consolidation* 71
5.2.3 Rotation/Pivoting 72
5.2.4 *Slice/dice* 73
5.3 Evaluasi 74
BAB VI KESIMPULAN DAN SARAN 75
6.1 Kesimpulan 75
6.2 Saran 75
DAFTAR PUSTAKA 76

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Integrated 7
Gambar 2.2 Ilustrasi Non Volatile 7
Gambar 2.3 Arsitektur Data Warehouse 8
Gambar 2.4 Multistage Data Transformation 11
Gambar 2.5 Pipelined Data Transformation 12
Gambar 2.6 Ilustrasi Data Multidimensi 13
Gambar 2.7 Proses Slice 14
Gambar 2.8 Proses Dice 14
Gambar 2.9 Star Schema 15
Gambar 2.10 Snowflake Schema 16
Gambar 2.11 Tabel Fakta 19
Gambar 2.12 Arsitektur OWB 21
Gambar 3.1 CDM Database Akademik ITS 25
Gambar 3.2 Star Schema Nilai IPS 27
Gambar 3.3 Star Schema totSKS 28
Gambar 3.4 Diagram Alur 29
Gambar 4.1 Form Pengisian Properti untuk Warehouse Builder Repository 31
Gambar 4.2 Ringkasan Runtime Repository 34
Gambar 4.3 Ringkasan Target Schema 34
Gambar 4.4 Setting Koneksi UDL File 36
Gambar 4.5 Query sederhana untuk menguji Database Link 39
Gambar 4.6 Informasi Koneksi 40
Gambar 4.7 Form Login OWB 40
Gambar 4.8 Jendela Utama OWB Client 41
Gambar 4.9 Lokasi Sumber 41
Gambar 4.10 Lokasi Target 42
Gambar 4.11 *Link* Sumber 43
Gambar 4.12 Import Obyek Database 44
Gambar 4.13 Informasi Koneksi ke target Data Warehouse 46
Gambar 4.14 *Level* Dimensi Mahasiswa 48
Gambar 4.15 Susunan Hirarki Dimensi Waktu 48
Gambar 4.16 Foreign Key pada Cube 50
Gambar 4.17 Measure IPS 51
Gambar 4.18 Memilih Tabel 52
Gambar 4.19 Kondisi operasi Join 53
Gambar 4.20 Pemetaan Dimensi Mahasiswa 55
Gambar 4.21 Pemetaan ke obyek Cube 57
Gambar 4.22 Alur Proses 60
Gambar 4.23 Kondisi END Activity 60
Gambar 4.24. Kondisi setelah Proses Create 62
Gambar 4.25 Data Dimensi Waktu 63
Gambar 4.26 Simpul TA 64
Gambar 4.27 Simpul Dimensi Mahasiswa 65
Gambar 4.28 Pemetaan Data Cube 66

Gambar 4.29 Crosstab IPS 67
Gambar 5.1 Tampilan awal Crosstab 70
Gambar 5.2 Hasil operasi *Drill-down* pada Fakultas MIPA cube IPS 70
Gambar 5.3 Hasil operasi Drill-down pada cube totSKS 71
Gambar 5.4 Hasil operasi *Consolidation* pada cube Nilai 71
Gambar 5.5 Hasil operasi Consolidation pada cube totSKS 72
Gambar 5.6 Hasil operasi Rotasi/Pivoting pada cube Nilai 72
Gambar 5.7 Hasil operasi Rotation pada cube totSKS 73
Gambar 5.8 Hasil operasi Slice/Dice pada cube Nilai 73
Gambar 5.9 Hasil operasi Slice/Dice pada cube totSKS 74

DAFTAR TABEL

Tabel 4.1 Dimensi Lain 47
Tabel 4 2 Foreign Key pada cube Nilai 49
Tabel 4.3 Foreign Key pada cube totSKS 50

BAB I

PENDAHULUAN

MILIK PERPUSTAKAAN
ITS

BAB I
PENDAHULUAN

Di dalam bab ini akan dijelaskan beberapa hal dasar mengenai Tugas Akhir yang akan dikerjakan. Hal-hal tersebut meliputi latar belakang, permasalahan, batasan permasalahan, tujuan serta sistematika pembahasan Tugas Akhir. Dari uraian tersebut diharapkan bisa memberikan gambaran umum permasalahan dan solusinya sehingga dapat dipahami dengan baik.

1.1 Latar Belakang

Penggunaan data dewasa ini semakin memegang peranan penting dalam sebuah institusi, baik itu institusi pendidikan maupun institusi sosial, terlebih pada institusi bisnis. Pengumpulan dan analisa data menjadi satu kewajiban bagi perusahaan jika ingin tetap mampu bersaing di era global seperti saat ini.

Semakin besar institusi tersebut mengakibatkan semaikin besar/banyak data yang dikumpulkan. Apalagi jika keberadaan institusi tersebut tersebar di berbagai tempat. Pemisahan tempat penyimpanan data di lokal masing-masing cabang institusi menjadi salah satu solusi dalam meningkatkan kinerja tiap cabang, mengingat kecepatan untuk mengakses data lebih cepat daripada ketika penyimpanan data berada di satu tempat yang secara fisik berjauhan antara satu dengan yang lain.

Pemisahan ini memungkinkan adanya ketidakkonsistenan data yang tersimpan pada masing-masing cabang, yang bisa menyebabkan masalah dikemudian hari. Untuk itu diperlukan satu metode untuk mengatasi masalah,

seperti konflik penamaan dan ketidakkonsistenan pengunaan satuan ukuran, ini yaitu dengan Data Warehouse.

Dalam Tugas Akhir ini akan dibuat sebuah Data Warehouse dengan memanfaatkan aplikasi yang dibuat oleh Oracle, Oracle Warehouse Builder. Aplikasi ini, seperti namanya, ditujukan untuk membuat sebuah Data Warehouse yang didalamnya sekaligus menampung beberapa teknologi pendukung Data Warehouse seperti teknologi OLAP (*On Line Analytical Processing*).

1.2 Permasalahan

Permasalahan yang akan dihadapi pada Tugas Akhir ini meliputi beberapa hal sebagai berikut :

1. Bagaimana cara mengekstraksi, mentransformasi dan memasukkan data dari Database ke Data Warehouse.
2. Bagaimana cara mengintegrasikan teknologi OLAP pada Data Warehouse yang sudah dibuat.

1.3 Batasan Permasalahan.

Dalam penerapan Tugas Akhir ini ada beberapa batasan, agar Tugas Akhir ini lebih fokus dan tidak melebar dari pokok permasalahan. Batasan-batasan itu antara lain :

1. Tugas Akhir ini tidak membahas bagaimana cara membuat Database yang baik dan benar.
2. Database yang akan digunakan untuk Tugas Akhir ini adalah database Akademik ITS yang berbasis MS SQL Server 2000.

3. Aplikasi yang digunakan untuk membangun Data Warehouse adalah Oracle Warehouse Builder 10g (10.1)
4. Untuk integrasi dengan teknologi OLAP, digunakan Analytic Workspace Manager (AWM).
5. RDBMS yang digunakan sebagai tujuan Data Warehouse adalah Database Oracle 10g.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah mengaplikasikan Oracle Warehouse Builder 10g untuk membangun sebuah Data Warehouse dan, dengan bantuan AWM, mengintegrasikannya dengan teknologi OLAP.

Secara khusus Tugas Akhir ini bertujuan untuk membuat sebuah Data Warehouse Akademis, dalam hal ini adalah data yang bersumber pada data akademis ITS.

1.5 Metodologi Pembuatan Tugas Akhir

Pembuatan Tugas Akhir ini terbagi menjadi beberapa tahap, sebagai berikut :

a. Studi Literatur

   Mencari dan mempelajari berbagai pustaka yang berkaitan dengan rumusan masalah, teori-teori yang berhubungan dengan aplikasi, dan juga literatur tentang teknologi yang akan diterapkan. Pustaka yang berkaitan dengan Tugas Akhir ini antara lain Data Warehouse, OWB, OLAP dan Oracle.

b. Desain Proses Pembangunan Data Warehouse Menggunakan OWB

Pada tahap ini akan dilakukan penggambaran desain pembangunan Data Warehouse menggunakan OWB. Desain yang dilakukan adalah dengan menggambarkan aliran proses menggunakan Diagram Alur (Flowchart).

c. Proses Pembangunan Data Warehouse Menggunakan OWB

Pada tahap ini akan dijelaskan proses pembangunan Data Warehouse menggunakan OWB secara bertahap. Tahapan tersebut dimulai dari penentuan sumber data dan diakhiri dengan pengintegrasian Data Warehouse dengan teknologi OLAP menggunakan Analytic Workspace Manager (AWM).

d. Uji Coba dan Evaluasi

Pada tahap ini Data Warehouse yang telah dibuat akan diuji coba, apakah sudah sesuai dengan hasil yang diharapkan. Selanjutnya, akan dilakukan evaluasi terhadap hasil yang diperoleh.

e. Penyusunan Buku Tugas Akhir

Pada tahap akhir ini disusun sebuah buku sebagai dokumentasi dari pelaksanaan Tugas Akhir. Pendokumentasian ini juga dibuat agar pengguna lain bisa memanfaatkan sebagaimana mestinya.

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini dibagi dalam beberapa bab. Bab I, berisi pendahuluan. Dalam bab ini akan dibahas latar belakang pembuatan Tugas Akhir, perumusan masalah, pembatasan masalah, tujuan serta manfaat yang diperoleh

dari penyusunan Tugas Akhir ini. Metodologi pembuatan Tugas Akhir juga akan dibahas pada bab ini.

Bab II, pada bab ini akan dibahas dasar teori yang akan digunakan dalam pembuatan Tugas Akhir. Teori dan konsep pembelajaran yang melandasi pembuatan Tugas Akhir ini akan menjadi pembahasan utama dalam bab ini. Dasar teori yang akan diterapkan pada Tugas Akhir ini meliputi Data Warehouse, OLAP, dan OWB.

Selanjutnya pada Bab III akan dibahas mengenai desain dari database sumber yaitu database FRS-Online ITS. Dari desain ini nantinya akan dibuat skema yang menjadi sumber acuan bagi data warehouse.

Bab IV membahas tentang proses pembangunan Data Warehouse menggunakan OWB secara detail, di dalamnya juga akan disertakan langkah-langkah pembangunan disertai dengan *screenshot*.

Uji coba dan evaluasi Data Warehouse yang telah dibuat dengan menggunakan OWB akan dibahas pada Bab V. Dalam melakukan ujicoba diperlukan skenario untuk menguji Data Warehouse yang sudah dibangun, dan akan disertakan pada bab ini.

Bab terakhir berisi kesimpulan penulis terhadap keseluruhan proses pembuatan Tugas Akhir. Saran untuk pengembangan ke depan juga diberikan pada bab ini.

BAB II

TEORI PENUNJANG

BAB II
TEORI PENUNJANG

Untuk menunjang konsep perencanaan Tugas Akhir ini, maka akan dibahas beberapa dasar teori yang menjadi landasan dalam penyusunan Tugas Akhir. Dasar teori ini meliputi Data Warehouse, OLAP, dan OWB sebagai aplikasi utama dalam Tugas Akhir ini.

2.1 Data Warehouse

Data Warehouse pada dasarnya adalah juga sebuah database tetapi berbeda dengan database pada umumnya yang digunakan untuk proses transaksi, Data Warehouse secara khusus di desain untuk kebutuhan *query* dan analisis.

Data Warehouse juga didesain untuk mendukung Sistem Pendukung Keputusan (SPK). Pada skala enterprise, Data Warehouse meningkatkan produktifitas pengambil keputusan melalui proses konsolidasi.

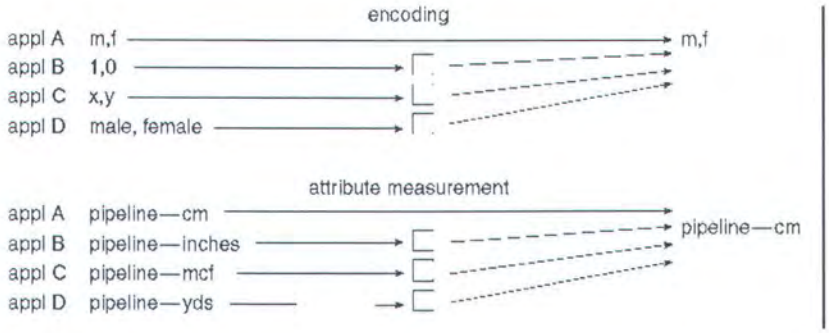
Ada beberapa karakteristik Data Warehouse yaitu [INM-02]:

a. Subject Oriented

Data Warehouse didesain untuk membantu dalam menganalisa data. Misalnya, untuk membantu memahami tentang data penjualan sebuah perusahaan maka bisa dibangun sebuah Data Warehouse yang berkonsentrasi hanya pada data penjualan. Dengan Data Warehouse ini maka pertanyaan-pertanyaan seperti "Siapa pelanggan dengan pembelian terbesar untuk produk A ?", atau "Apa produk yang paling banyak terjual di negara A selama tahun 2004 ?" akan dengan mudah dijawab.

b. Integrated

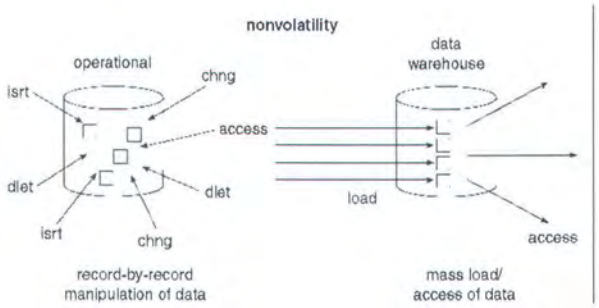
Data Warehouse menggabungkan data dari berbagai sumber transaksi. Untuk itu Data Warehouse harus mampu menyelesaikan masalah seperti konflik penamaan dan ketidakkonsistenan.



Gambar 2.1 Ilustrasi Integrated

c. Nonvolatile

Sekali data dimasukkan ke dalam Data Warehouse, data tidak bisa berubah. Hal ini wajar, mengingat tujuan dari Data Warehouse adalah memungkinkan user untuk melakukan proses analisis terhadap data.

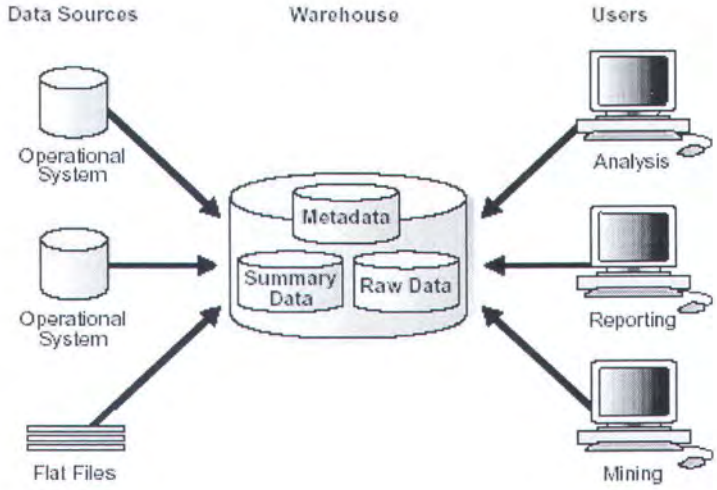


Gambar 2.2 Ilustrasi Non Volatile

d. Time Variant

Untuk memprediksikan sebuah tren diperlukan data yang sangat besar, berbeda dengan OLTP yang membutuhkan performa lebih agar data historis dipindahkan menjadi arsip/berkas.

Arsitektur Data Warehouse secara sederhana bisa digambarkan sebagai berikut.



Gambar 2.3 Arsitektur Data Warehouse

Data perlu dimasukkan ke dalam Data Warehouse secara rutin sehingga bisa memenuhi tugasnya menganalisa bisnis. Untuk melakukannya, data dari satu atau beberapa sistem opearsional diekstraksi dan disalin ke dalam Data Warehouse. Proses pengekstraksian data dari sumber dan memasukkannya ke dalam Data Warehouse lebih umum disebut dengan proses ETL (*Extraction, Transformation and Loading*). Sebenarnya ada proses lain yang terlibat yaitu *Transportation*, tetapi bagi sebagian besar praktisi Data Warehouse istilah ETL lebih dikenal secara umum.

Proses ETL ini sering dianggap sebagai proses yang paling rumit dalam membangun sebuah Data Warehouse. Seringkali digunakan perangkat pembantu untuk proses ini, salah satunya adalah Oracle Warehouse Builder.

2.1.1 Extraction

Extraction adalah proses pengekstraksian data dari sumber data untuk diolah lebih lanjut dalam Data Warehouse. Sumber data dari Data Warehouse biasanya berupa aplikasi pemroses transaksi.

Mendesain dan membangun proses ekstraksi data terkadang menjadi pekerjaan yang memerlukan banyak waktu pada proses ETL. Sistem sumber terkadang adalah sebuah sistem yang kompleks dan sayangnya tidak terdokumentasi dengan baik. Sehingga menyulitkan dalam proses ekstraksi datanya.

Proses ekstraksi normalnya tidak dilakukan hanya sekali, tapi berkali-kali secara berkala untuk menjamin data yang ada dalam warehouse adalah data terakhir (*up to date*).

Untuk mendesain proses ekstraksi perlu diperhatikan dua aspek berikut :

a. Metode ekstraksi yang dipakai
b. Bagaimana mempersiapkan data yang sudah diekstraksi untuk proses selanjutnya.

Pemilihan metode ekstraksi sangat tergantung pada sistem sumber, ada dua metode ekstraksi.

1. Metode ekstraksi logikal

   Metode ini mempunyai dua tipe *Full Extraction* dan *Incremental Extraction*. Pada tipe Ekstraksi penuh, data secara penuh diekstraksi dari sistem sumber, sedang pada tipe kedua hanya data yang berubah pada titik waktu tertentu saja yang akan diekstrak.

2. Metode ekstraksi fisik

Metode ekstraksi fisik juga mempunyai dua tipe, yaitu *Online Extraction* dan *Offline Extraction*. Metode Online mengekstraksi data langsung dari sistem sumber, sedang pada metode Offline data lebih dulu ditampung disuatu tempat di luar sistem sumber baru kemudian diekstraksi ke dalam Data Warehouse. Untuk metode offline, penampungan data sebelum diekstraksi bisa berupa flat files, dump files, redo dan archive logs serta transportable tablespaces.

2.1.2 Transportation

Proses ini memindahkan data dari satu sistem ke sistem yang lain. Dari keseluruhan proses ETL, proses ini dianggap sebagai proses yang paling mudah, bahkan pada umumnya proses ini tergabung menjadi satu dengan proses ekstraksi.

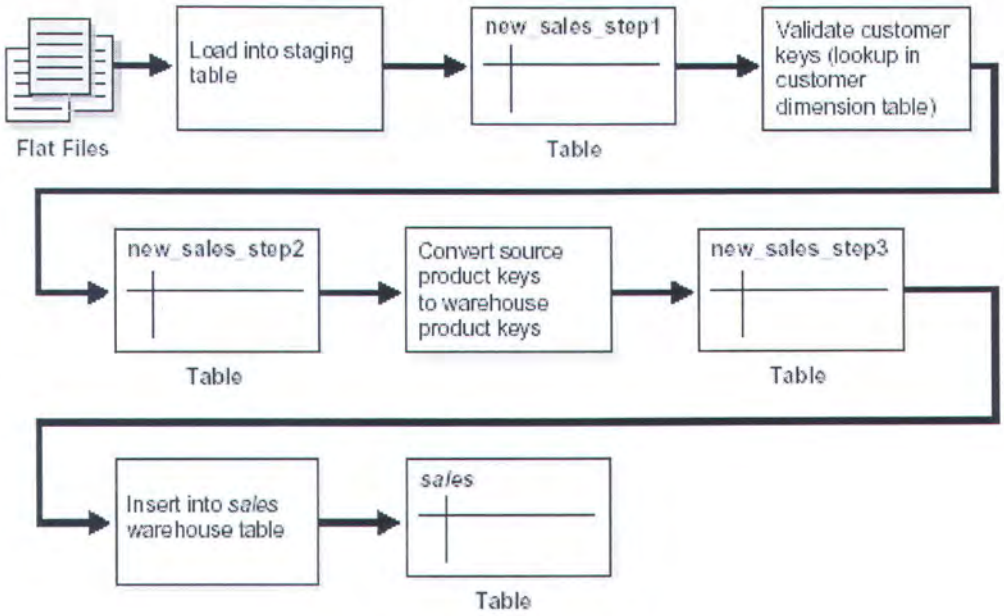
2.1.3 Transformation dan Loading

Proses ini, dibandingkan dengan proses ETL lainnya, merupakan proses yang paling rumit. Pada umumnya proses ini terjadi dalam Database Oracle, meskipun terkadang ada juga yang mengimplementasikannya di luar database, dengan flat files misalnya.

Secara arsitektural, data bisa ditransformasikan melalui dua cara.

1. Multistage Data Transformation

Secara umum proses transformasi data dalam Data Warehouse terdiri dari beberapa langkah.untuk lebih jelasnya bisa dilihat pada gambar dibawah ini.

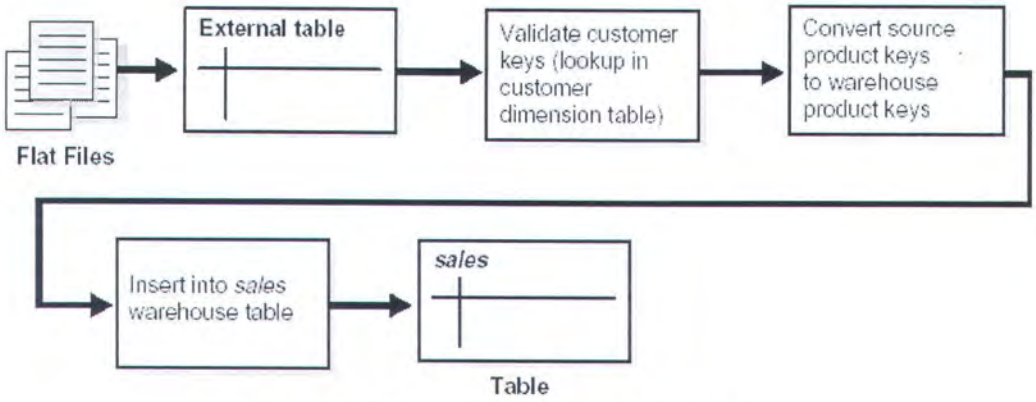


Gambar 2.4 Multistage Data Transformation

Dengan menggunakan Oracle sebagai alat transformasi, strategi yang umum dipakai adalah mengimplementasikan tiap transformasi sebagai operasi SQL yang terpisah dan membuat tabel penampung sementara untuk menampung hasilnya pada tiap langkah. Keunggulan memaki cara ini adalah tersedianya cek poin sehingga memudahkan dalam monitoring dan pengulangan (*restart*), di lain pihak cara ini membutuhkan ruang dan waktu yang lebih banyak.

2. Pipelined Data Transformation

Berbeda dengan cara *Multistaging* yang membuat tabel pada tiap langkah, metode *Pipelined* tidak memerlukan penampungan sementara sehingga dengan sekali jalan eksternal tabel langsung ditransformasi ke dalam tabel tujuan.



Gambar 2.5 Pipelined Data Transformation

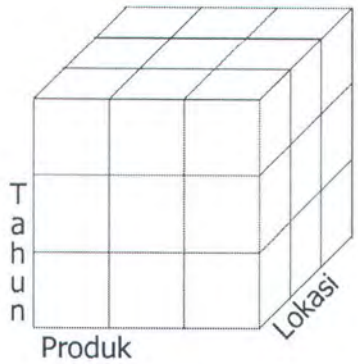
2.2 OLAP (On Line Analytical Processing)

OLAP adalah suatu pemodelan data multidimensional untuk memenuhi kebutuhan analisis terhadap data. OLAP menyajikan data dalam bentuk multidimensional untuk memberikan pengetahuan dan pemahaman akan data dan pola-pola tersembunyi.

Hal ini memungkinkan analisa terhadap data yang tidak mungkin dilakukan dengan menggunakan query biasa. Tapi OLAP tidak diperuntukkan untuk pencarian tren yang tersembunyi dan relasi yang tidak biasa seperti yang dilakukan oleh Data Mining.

2.2.1 Data Multidimensi

Pengambilan data dengan menggunakan query hanya bisa menampilkan data secara satu dimensi. Artinya, hanya ada 2 sudut pandang saja, yaitu baris dan kolom. Sedangkan Data Multidimensi adalah ketika sebuah data dapat dipandang dari berbagai sudut, misal hasil penjualan suatu barang dipandang dari dimensi waktu, lokasi, pembeli dan lain-lain. Jika masing-masing (waktu, lokasi dan pembeli) diwakili oleh sumbu x, y dan z, maka bentuk grafiknya menjadi grafik 3 Dimensi seperti gambar 2.6.



Gambar 2.6 Ilustrasi Data Multidimensi

2.2.2 Operasi pada Data Multidimensi

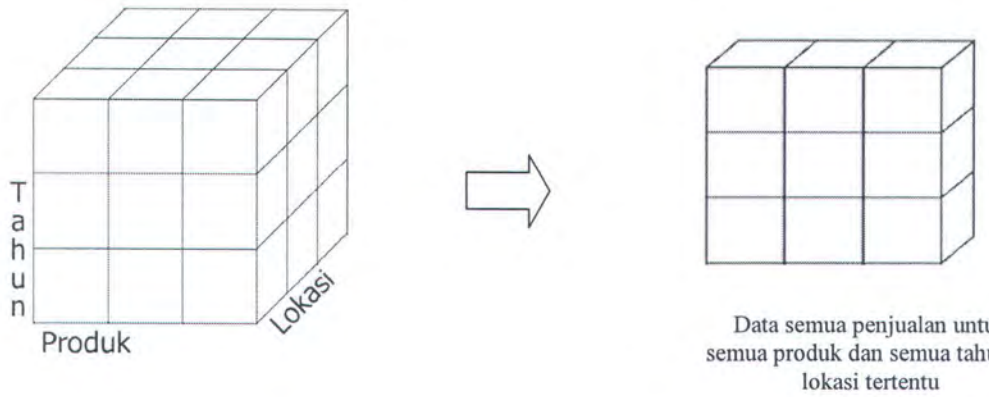
Ada beberapa operasi yang bisa dilakukan terhadap data multidimensi untuk mendapatkan data yang diinginkan.

2.2.2.1 Rotation/Pivoting

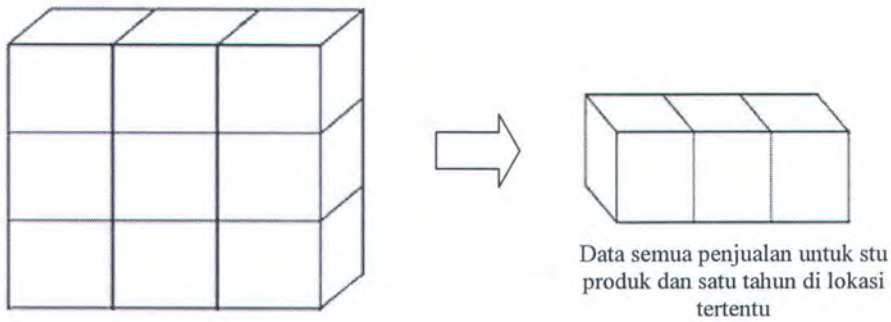
Operasi ini mengubah sudut pandang pada data. Dengan operasi ini, perubahan sudut pandang pada data dapat dilakukan dengan mudah. Perputaran atau rotasi yang dilakukan untuk masing-masing dimensi, artinya rotasi suatu sumbu digantikan dengan sumbu yang lain.

2.2.2.2 Slice dan Dice

Operasi ini memungkinkan pemilihan subset pada data. Proses slice adalah proses pemotongan data pada cube berdasarkan nilai pada satu atau beberapa dimensi. Sedangkan proses dice adalah pemotongan hasil slice menjadi bagian subset data yang lebih kecil. Dua proses ini menjadikan proses navigasi menjadi lebih mudah, terlebih pada data yang banyak.



Gambar 2.7 Proses Slice



Gambar 2.8 Proses Dice

2.2.2.3 *Drill-down* dan *Consolidation*

Dengan proses *drill-down* subset data bisa dilihat lebih detil. Proses ini melibatkan proses agregasi data. Sebaliknya pada proses *Consolidation* data dilihat secara global. Proses-proses ini memanfaatkan hirarki pada dimensi yang membentuk cube.

MILIK PERPUSTAKAAN ITS

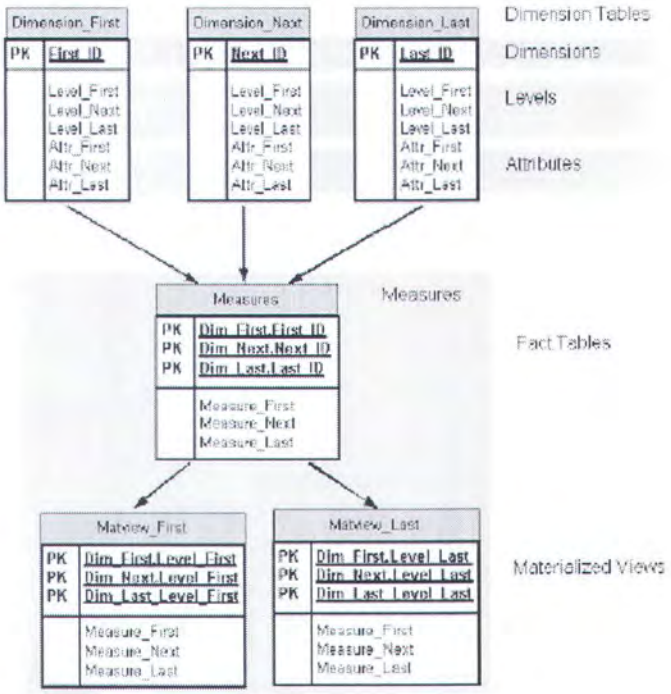
2.2.3 Skema Data Pada OLAP

Skema adalah koleksi dari obyek-obyek database, dan biasanya senama dengan nama user pemilik skema. Tiap user pada Oracle mempunyai skemanya masing-masing.

Skema adalah struktur penyimpanan data lojikal. Obyek skema tidak mempunyai hubungan one-to-one terhadap file fisik pada disk yang menyimpan informasi file.

2.2.3.1 Star Schema

Skema Star terdiri dari satu atau lebih tabel fakta.dan satu atau tabel dimensi [KUN-04]. Tabel Fakta menjadi pusat dari skema jenis star ini yang menjadi pengikat diantara tabel-tabel dimensi yang ada di sekelilingnya. Hubungan diantara tersebut menggunakan foreign, metadata, atau keduanya.

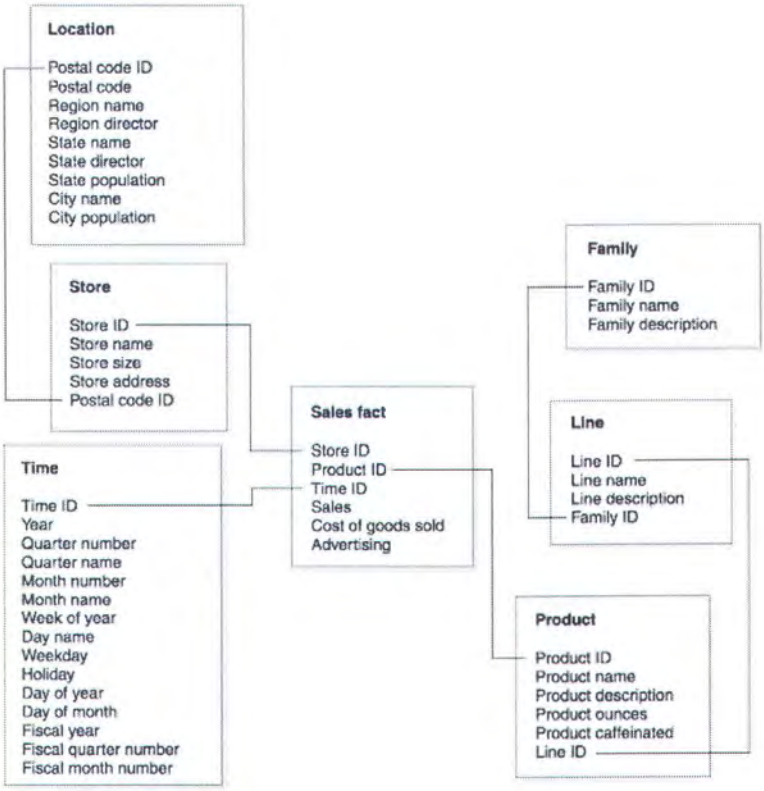


Gambar 2.9 Star Schema

2.2.3.2 Snowflake Schema

Skema ini merupakan pengembangan dari skema star, hanya saja pada skema ini tabel dinormalisasi sebagian atau seluruhnya untuk mengurangi duplikasi pada tabel. Normalisasi ini menyebabkan tabel dimensi pada skema snowflake lebih dari satu. Dengan banyaknya tabel, sehingga memerlukan banyak operasi join mengakibatkan performance menjadi lambat [OLA-02].

Dalam snowflake skema hanya satu tabel yang dihubungkan dengan tabel fakta, sedang tabel lainnya dihubungkan dengan tabel utama.



Gambar 2.10 Snowflake Schema

2.2.3.3 Fact Constellation

Model skema ini adalah modifikasi dari skema snowflake. Pada skema model ini terdapat beberapa tabel fakta dan adanya dimensi yang merujuk pada lebih dari satu tabel fakta [DJU-05]. Model skema ini dikenal juga dengan sebutan *galaxy schema*.

2.2.4 OLAP Metadata Model

OLAP mempunyai beberapa metadata model yang mempunyai peran dan fungsi masing-masing.

2.2.4.1 Dimensi

OWB menggunakan dimensi untuk mengorganisasikan dan mengindeks data pada *cube*. Untuk membuat dimensi, harus didefinisikan dulu hirarki, *level* dan *level relationship*.

Dimensi terbentuk dari satu atau lebih tabel. Setiap kolomnya merepresentaikan *level* pada hirarki. *Primary Key*-nya akan terhubung dengan tabel fakta.

Secara umum karakteristik dari sebuah dimensi terdiri dari [OWB-03]:

- Kolom *Primary Key* tunggal yang berisi data yang disebut Warehouse key.
- Warehouse Key yang memegang kendali atas dimensi, teknik pendukung yang menjaga histori dimensi dan mengurangi ukuran *cube*.
- Satu atau lebih hirarki yang secara eksplisit didefinisikan sebagai obyek dimensi.

2.2.4.2 Cube

Cube tersusun dari measure, dimension, attribute. *Cube* terhubung dengan dimensi melalui *Foreign Key*. Ketika dimensi didesain dengan *warehouse key*, panjang baris *cube* biasanya akan berkurang karena *warehouse key* lebih pendek daripada duplikasi alaminya. Sebuah *cube* biasanya terdiri dari :

- Sebuah *Primary Key* yang didefinisikan pada sekelompok *Foreign Key* mengacu pada kolom-kolom atau pada sebuah A*rtificial Key* atau bisa juga pada sekumpulan kolom W*arehouse Key*.

- Sekumpulan kolom rujukan *Foreign Key* yang menghubungkan tabel dengan dimensinya.

2.2.4.3 Hirarki

Obyek OLAP yang satu ini menggunakan *level* secara bertingkat untuk mengumpulkan (*aggregrate*) data. Sebagai contoh pada dimensi waktu, hirarki mungkin akan melakukan agregasi data dari *level* bulan ke triwulanan selanjutnya ke *level* tahun. Sebuah dimensi bisa terdiri dari satu atau lebih hirarki, misalnya hirarki produk mungkin terdiri dari hirarki kategori produk dan hirarki supplier produk.

2.2.4.4 Level

Level merepresentasikan sebuah posisi pada hirarki. *Level* mengumpulkan data untuk agregasi dab digunakan untuk proses komputasi. Setiap *level* diatas *level* terendah merupakan agregasi dari *level* yang ada dibawahnya. Sebagai contoh, pada dimensi waktu mungkin didefinisikan beberapa *level* yaitu *level* hari, *level* bulan, tri-wulan, catur-wulan, dan *level* tahun.

2.2.4.5 Attribute

Attribute adalah informasi tambahan pada sebuah *level*. Nilai atribut inilah yang akan ditampilkan kepada user untuk mewakili *level* pada data multidimensi. Untuk itulah pada setiap *level* minimal harus ada satu attribute.

2.2.4.6 Fact Table

Model ini adalah pusat dari skema pada OLAP. Tabel fakta mempunyai dua tipe kolom, yaitu kolom yang menyimpan nilai-nilai numerik (disebut dengan

measure) dan kolom yang menyimpan *Foreign Key* yang mengacu pada tabel dimensi. Nilai numerik yang ada pada tabel fakta merupakan nilai agregat dari data yang berasal dari tabel dimensi yang ber-relasi *one-to-many*.

2.2.4.7 Measure

Measure dapat diartikan sebagai fakta. Nilai measure terletak pada tabel fakta. Pada umumnya, *measure* bertipe data numerik agar dapat dilakukan operasi penambahan terhadapnya.

Gambar 2.11 Tabel Fakta

2.3 Oracle Warehouse Builder (OWB)

Oracle Warehouse Builder atau OWB adalah sebuah aplikasi bisnis cerdas yang menyediakan solusi terintegrasi intuk mendesain dan mengimplemantasikan data warehouse, data mart dan aplikasi bisnis cerdas sampai skala enterprise. OWB menyediakan segala fungsi yang diperlukan untuk mengelola siklus hidup dari sistem yang akan dibuat.

OWB merupakan paduan dari alat (*tool*) ekstraksi, transformasi dan *loading* (ETL) dengan alat desain. Arsitektur dari OWB terdiri dari dua komponen *design environment* dan *runtime environment*. Design Environment bertugas mengelola metadata sedangkan Runtime Environment menangani data fisik.

2.3.1 Lingkungan Desain

Lingkungan atau komponen ini terdiri dari repository metadata yang tersimpan dalam sebuah Database Oracle dan sekumpulan desain klien dan alat pelaporan yang ditulis dalam bahasa Java atau HTML. Dengan menggunakan alat-alat tersebut, metadata bisa ditampilkan dan dimanipulasi.

Biasanya metadata didesain secara manual mulai dari nol untuk membuat sistem yang baru. Tetapi dengan menggunakan OWB metadata bisa diimport dari klien atau data sumber dan disimpan dalam repository. Karena itulah sistem sumber mempunyai peranan yang sangat penting dalam penyelesaian proses ETL.

Lebih jauh lagi, OWB menyediakan sistem pelaporan berbasis web yang memungkinkan pengembang dan pengguna bisnis menjelajahi dan menganalisa elemen sistem tanpa harus menggunakan alat desain. Fasilitas lainnya adalah Analisa Dampak yang memungkinkan pengguna untuk mengetahui dampak apa yang akan terjadi pada sistem bila ada perubahan yang akan diimplementasikan.

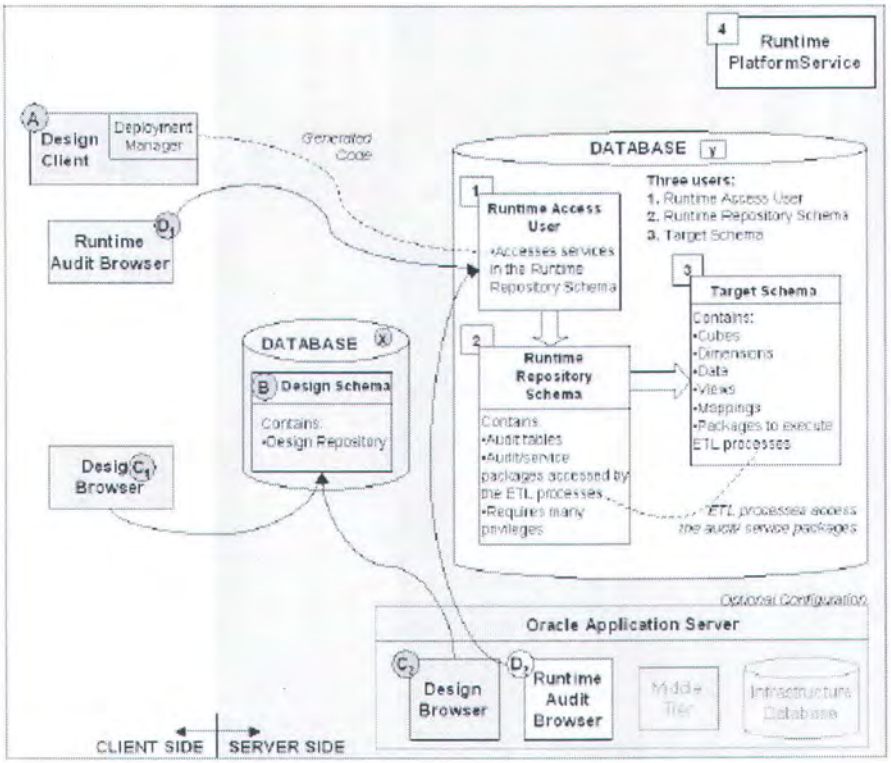
2.3.2 Runtime Environment

Setelah proses ETL didesain pada tahap lojikal, proses ETL perlu dipindah ke bentuk fisiknya. Untuk itu diperlukan informasi tentang lingkungan database yang akan ditambahkan pada desain lojikal ketika target telah ditentukan untuk proses implementasi. Setelah semuanya lengkap, baru kode program bisa digenerate.

Menjalankan proses ETL artinya menjalankan kode program dalam database, hal ini bisa dilakukan dengan menggunakan Warehouse Builder

Deployment Manager atau lewat alat eksternal seperti Oracle Enterprise Manager. Proses ETL kemudian akan menarik data ke database target.

Pada gambar 2.12. adalah arsitektur OWB.



Gambar 2.12 Arsitektur OWB

- **Design Client And Deployment Manager (A) :** menyediakan antarmuka grafis yang digunakan untuk mendefiniskan sumber dan mendesain target serta proses-proses ETL.
- **Design Repository Schema (B), Design Repository, dan Design Browser (C1) :** Desain Repository Schema memuat Desain Repository, di dalam Desain Repository termuat metadata dari sumber, target dan proses ETL yang berhubungan dengan metadata yang telah didesain. Isi dari Desain Repository bisa kita lihat lewat Desain Browser, tapi tidak bisa kita ubah (*read-only*).

- **Runtime Instance of Oracle Database (x) :** Biasanya terinstall pada sisi server dan secara default akan ada 3 user yang telah tercipta. Runtime Access User, Runtime Repository Schema, dan Target Schema.
- **Runtime Access User :** User ini mencegah adanya tindakan pelanggaran mengingat skema Runtime Repository membutuhkan beberapa hak system. Sebenarnya user ini tidak mempunyai hak apa-apa selain mengakses Runtime repository untuk men-*deploy* dan mengeksekusi proses-proses ETL.
- **Runtime Repository Schema :** Memiliki tabel audit dan paket-paket audit yang diakses oleh target skema. Komponen ini juga mengelola segala koneksi pada sembarang target pada skema target.
- **Runtime Audit Browser (D1 dan D2) :** Berkomunikasi dengan Runtime Repository Schema untuk menampilkan laporan atas hasil audit dan kesalahan-kesalahan yang terjadi ketika proses-proses ETL dijalankan.
- **Target Schema (3):** Adalah target sebenarnya dimana nantinya data akan di-*load*, dan didalamnya juga terdapat obyek-obyek data, seperti *cube*, *dimensions*, *views* dan *mappings*. Beberapa tujuan bisa berasosiasi dengan satu Repository Schema (2).
- **Runtime Platform Service (4) :** Mengelola eksekusi *native* dan memanggil Oracle Enterprise Manager untuk eksekusi dari jauh.

Komponen ini harus aktif agar bisa melakukan aktifitas pada *runtime environment*, dan harus ter-*install* pada mesin yang sama dengan database target.

BAB III

PERANCANGAN DATA WAREHOUSE

BAB III
PERANCANGAN DATA WAREHOUSE

3.1 Database Sumber

Database ITS merupakan database yang berisi, tidak hanya, tentang data akademik tapi juga berisi data pegawai (dosen dan karyawan).

Dari dua 'sub database' bisa dibuat dua macam Data Warehouse, satu difokuskan pada Akademik (Nilai, Kuliah, Mata Kuliah, dll) yang lain fokus pada Pegawai (Data Warehouse Sumber Daya Manusia – Human Resources Data Warehouse). Tetapi untuk Tugas Akhir ini hanya memfokuskan pada Data Akademik saja. Sehingga nantinya Data Warehouse yang dibuat terfokus pada satu masalah saja yaitu Akademik.

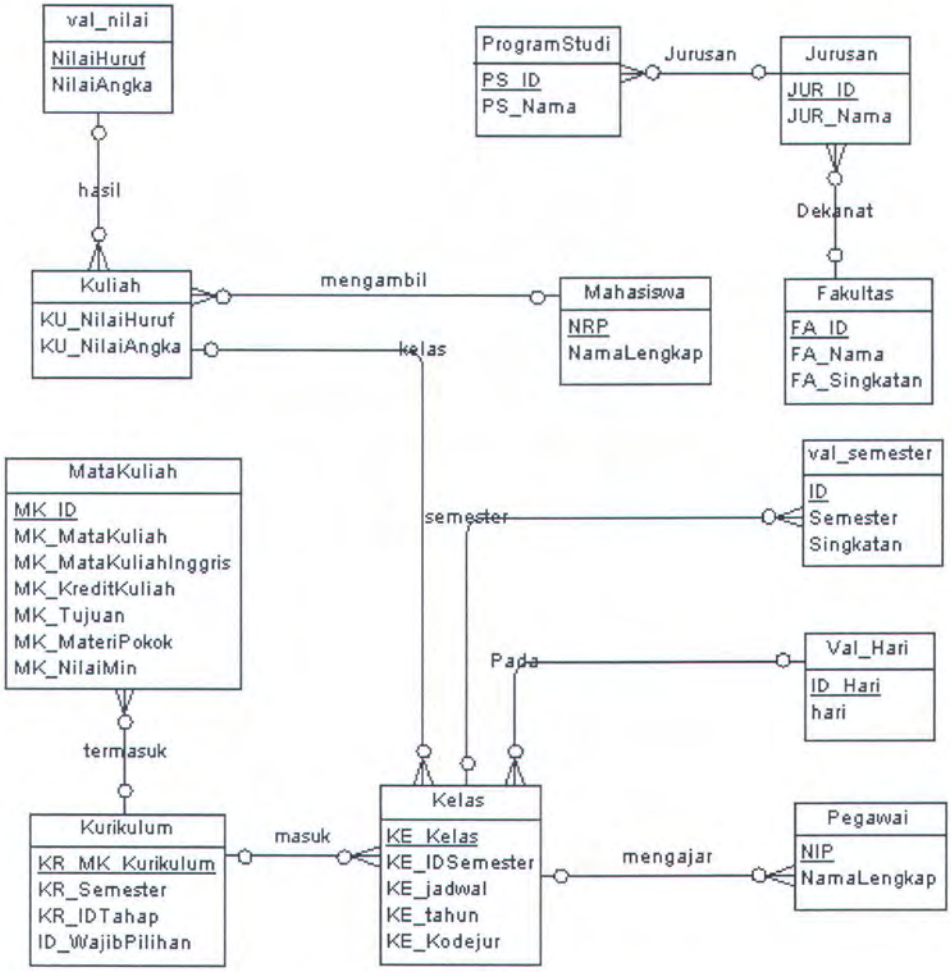
Karena hanya fokus pada data Akademik, ERD yang tampilkan hanya tabel-tabel yang berhubungan dengaan akademik saja. Berikut ini Diagram Relasi dari Database Akademik ITS

3.1.1 Pemodelan Data

Pemodelan Data dalam desain Sistem Informasi menurut Wikipedia (http://www.wikipedia.org) adalah analisis dan desain dari Sistem Informasi, yang fokus pada entitas lojikal, dan hubungan lojikal antar entitas yang ada.

Pada Tugas Akhir ini, tidak semua entitas/tabel yang akan di tampilkan pada diagram relasi, hanya yang berhubungan dengan Akademik saja yang akan

ditampilkan. Mengingat bahwa Data Warehouse yang akan dibuat adalah Data Warehouse Akademik.



Gambar 3.1 CDM Database Akademik ITS

3.2 Desain Skema Data Warehouse

Untuk membuat *star schema* dari sebuah Data Warehouse, perlu ditentukan terlebih dahulu dimensi dan measure yang akan dipakai. Untuk Tugas Akhir ini akan dibuat beberapa *star schema* yang merepresentasikan *cube* yang akan dibuat pada Data Warehouse.

Seperti telah disebutkan pada 2.2.3, ada beberapa skema yang bisa diadopsi untuk membuat sebuah Data Warehouse. Untuk Tugas Akhir ini skema yang akan dipakai adalah *star schema.*

Ada banyak pendapat tentang skema apa yang baik untuk sebuah Data Warehouse, menurut Ralph Kimball *snowflake schema* lebih baik, karena lebih mencerminkan kondisi dunia nyata. Sedangkan *star schema* menurut Kimball, 'lebih menyederhanakan' masalah sehingga membuat para pengambil keputusan merasa 'asing' dengan data yang ditampilkan.

Pendapat yang cenderung membela star schema karena lebih mudah dipahami dan pada umumnya mempunyai *performance* yang lebih baik. Star schema biasanya bersumber pada database yang dinormalisasi sampai level 2 (2NF) sedangkan snowflake schema sampai level 3 (3NF).

3.2.1 Analisa Nilai Indeks Prestasi Semester (IPS)

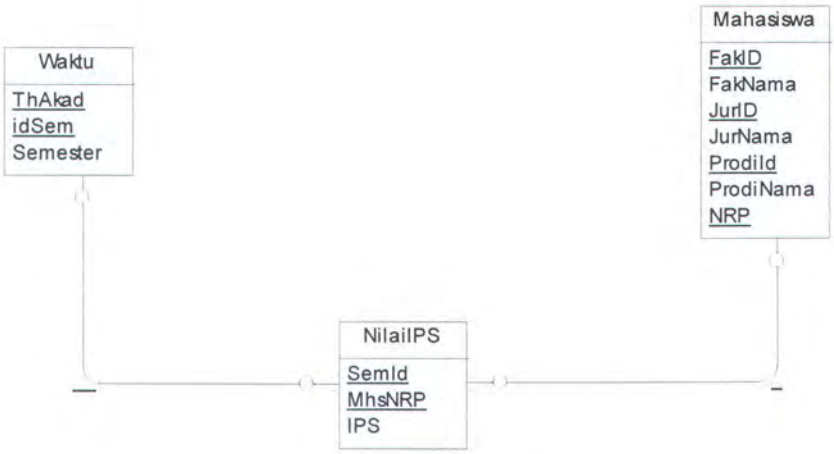
Skema pertama yang akan dibuat adalah untuk menampilkan laporan analisa terhadap mahasiswa. Dari laporan ini diharapkan bisa mendapatkan informasi mengenai nilai IPS Mahasiswa W, pada tahun Akademik Y di Semester Z .

Untuk mendapatkan informasi tersebut diperlukan beberapa Dimensi, tepatnya 2 Dimensi dan 1 Cube, yaitu :

- Dimensi Waktu – dimensi ini memuat informasi Semester beserta Tahun Akademiknya
- Dimensi Mahasiswa – data Mahasiswa termuat dalam dimensi ini. Data ini meliputi Fakultas, Jurusan, Program Studi, dan NRP

- Cube Nilai – dalam cube ini terdapat data IPS, Semester (FK dari Dimensi Waktu) dan NRP (FK dari Dimensi Mahasiswa).

Untuk skema ini hanya memuat satu *measure* yaitu IPS yang berisi data IPS Mahasiswa per semester.



Gambar 3.2 Star Schema Nilai IPS

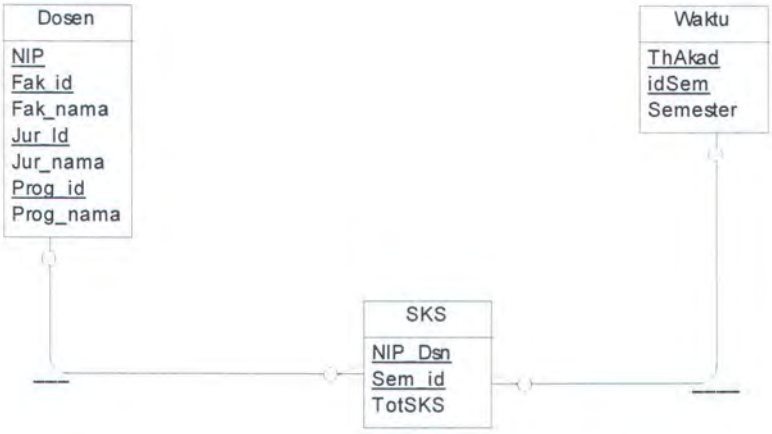
3.2.2 Analisa Total Kredit Mengajar

Skema ini bertujuan untuk mengetahui berapa jumlah beban kredit yang di tangung tiap Dosen pada semester dan tahun ajaran tertentu.

Tidak jauh berbeda dengan skema pertama, skema ini mempunyai dua dimensi dan satu *cube*/tabel fakta.

- Dimensi Waktu – memuat informasi tahun akademik dan semester.
- Dimensi Dosen – data dosen; NIP, Program Studi, Jurusan dan Fakultas ditampung pada dimensi ini.
- Cube SKS – selain foreign key dari dua dimensi (waktu dan dosen) tabel fakta sks mempunyai measure totsks.

Gambar 3.3 menunjukkan hubungan tabel fakta SKS dengan dimensi waktu dan Dosen.



Gambar 3.3 Star Schema totSKS

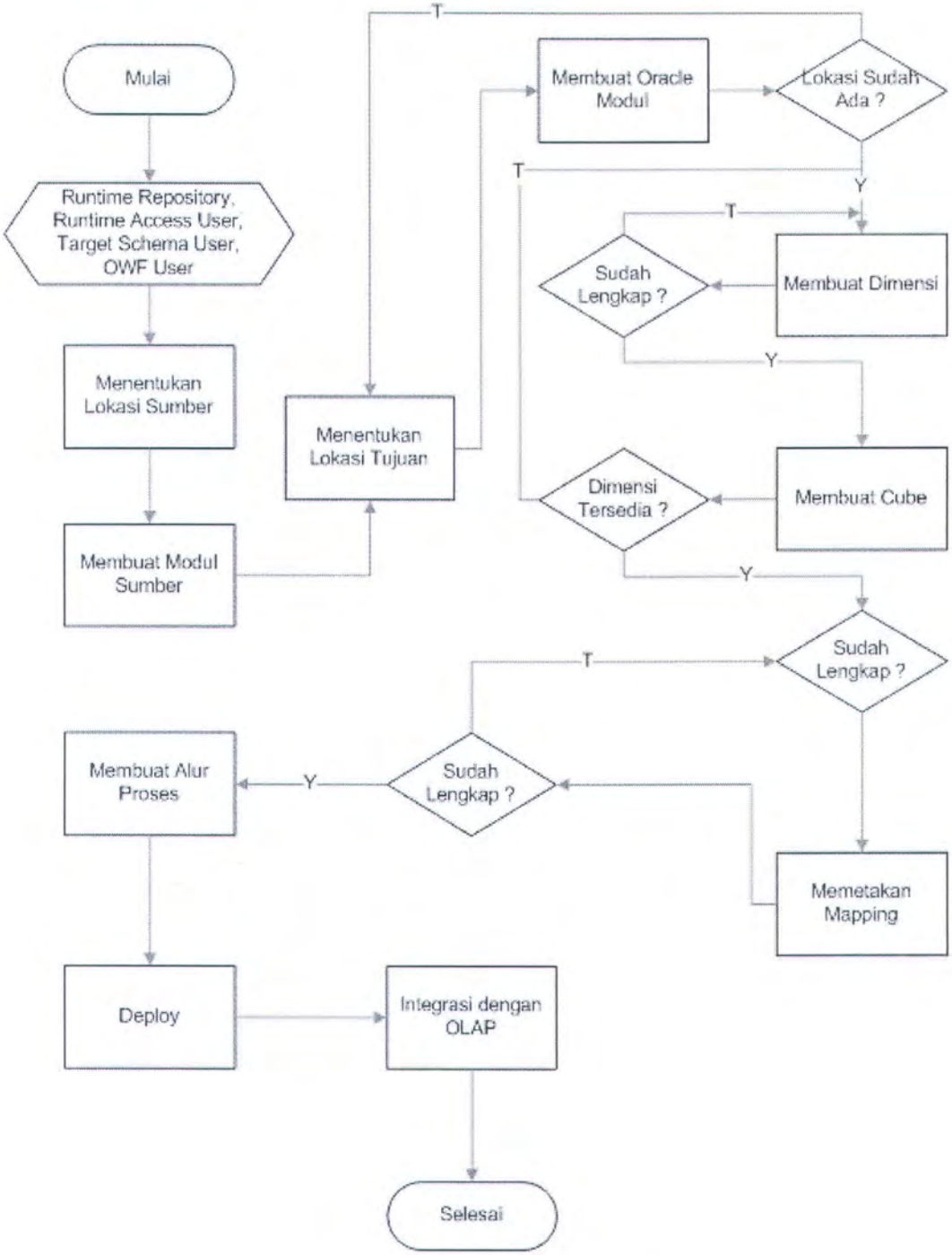
3.3 Flowchart Data Warehouse

Untuk me-*load* data dari database ke sebuah Data Warehouse dengan menggunakan OWB, beberapa langkah harus dilakukan.

Pada gambar 3.3 diilustrasikan langkah-langkah pembangunan Data Warehouse dari nol dengan menggunakan OWB.

Pertama adalah menyiapkan lingkungan yang akan dibuat untuk membangun sebuah Data Warehouse, dalam hal ini yang harus disiapkan adalah membuat Runtime Repository, Runtime Access User dan Target Schema. Perlu disiapkan juga Oracle Workflow Server untuk menangani alur proses saat proses *deploy*. Selanjutnya perlu ditentukan lokasi sumber data dan tujuan sebelum melangkah ke pembuatan Modul. Setelah dua modul dibuat (sumber dan tujuan), dimensi-dimensi yang diperlukan bisa dibuat, dilanjutkan dengan pembuatan cube. Pemetaan adalah langkah lanjutan untuk memetakan aliran data dari tabel ke

dimensi atau cube. Langkah terakhir adalah implementasi (*deploy*) setelah sebelumnya dibuat alur proses.



Gambar 3.4 Diagram Alur

BAB IV

IMPLEMENTASI ORACLE WAREHOUSE BUILDER

MILIK PERPUSTAKAAN
ITS

BAB IV
IMPLEMENTASI ORACLE WAREHOUSE BUILDER

Pada bab ini akan dijelaskan langkah demi langkah penggunaan OWB untuk membangun sebuah Data Warehouse. Kemudian akan dijelaskan juga integrasi teknologi OLAP terhadap Data Warehouse yang telah dibuat

4.1 Menyiapkan Lingkungan Warehouse

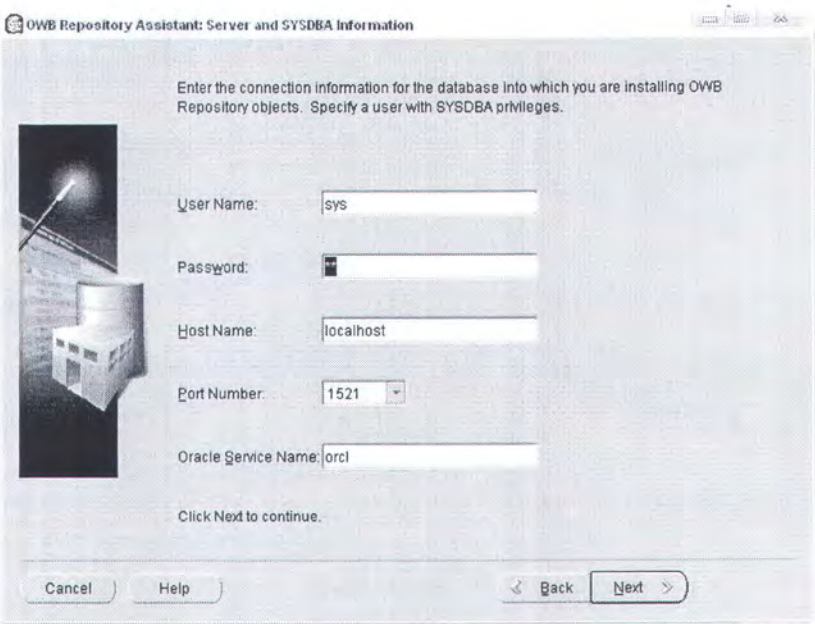
Sebelum melakukan proses ETL yang merupakan proses kunci dalam pembuatan sebuah Data Warehouse perlu disiapkan dulu lingkungan yang akan digunakan dalam pengimplementasian Data Warehouse, begitu juga dengan OWB. Berikut ini adalah beberapa poin yang perlu disiapkan sebelum melangkah ke proses ETL.

4.1.1 Membangun Warehouse Builder Repository

Untuk membuat Warehouse Repository gunakan wizard Repository Assitant (**Start > Program Files > Oracle – OWB10g > Warehouse Builder > OWB Repository Assistant**), kemudian ikuti langkah-langkahnya :

1. Yang pertama kali tampil adalah jendela yang menjelaskan fungsi wizard yang sedang berjalan. Beri tanda centang pada pilihan *Show this page next time* jika ingin menampilkan halaman ini lain waktu. Klik *Next*.

2. Isikan properti yang diminta pada *form* kedua. Contoh isian bisa dilihat pada gambar 4.1. Catatan : *User* yang dimasukkan pada form kedua harus mempunyai hak DBA pada Database yang akan dijadikan sumber Data Warehouse.



Gambar 4.1 Form Pengisian Properti untuk Warehouse Builder Repository

3. Selanjutnya akan muncul *form* dengan tiga pilihan, pilih **Create a new Warehouse Builder Repository** untuk membuat *Warehouse Builder Repository* baru.
4. Form selanjutnya adalah untuk menentukan pada skema mana *Warehouse Repository* akan diletakkan. Untuk Tugas Akhir ini skema yang digunakan adalah **rt_owner**.
5. Berikutnya adalah form tablespace dilanjutkan dengan form bahasa. Pilih default dan lanjutkan.

6. Form terakhir berisi ringkasan dari awal sampai akhir pembuatan *Warehouse Builder Repository*. Klik Finish untuk mengakhiri pembuatan Repository.

Warehouse Repository berfungsi untuk menampung segala definisi obyek yang akan digunakan oleh Data Warehouse. Untuk mengakses metadata dari Warehouse diperlukan user yang mempunyai hak akses terhadap metadata tersebut. Pada OWB user ini disebut Runtime User. Satu lagi akun yang berfungsi untuk mengakses runtime repository yaitu Runtime Access User.

Untuk satu database *instance* paling tidak diperlukan satu Runtime Repository yang berfungsi untuk menambahkan (meng-*install*) komponen tabel audit, paket-paket service untuk mengelola lingkungan Runtime, dan juga paket-paket nama dan alamat server Warehouse.

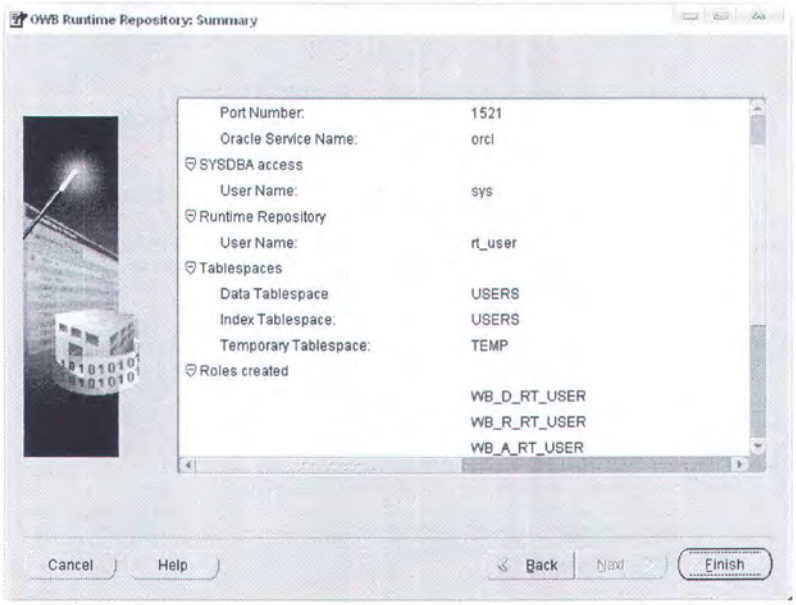
4.1.2 Membuat Runtime Repository

Seperti juga pada Warehouse Repository, pembuatan Runtime Repository ini bisa juga menggunakan wizard yang telah tersedia (**Start > Program Files > Oracle – OWB10g > Warehouse Builder > OWB Runtime Assistant**).

1. Form penjelasan tujuan wizard akan muncul pertama kali, klik Next untuk melanjutkan.
2. Selanjutnya adalah form untuk proses autentifikasi ke database, untuk itu diperlukan user dan password yang mempunyai hak setingkat Database Administrator (DBA).

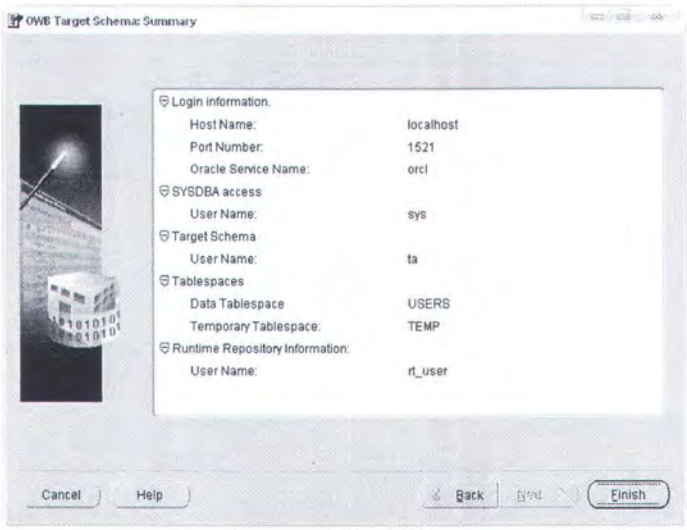
3. Kemudian form pilihan ditampilkan, pilih pilihan pertama (Runtime Repository atau RR) karena pada database yang akan digunakan komponen ini belum ada. Klik Next untuk melanjutkan.
4. Form berikutnya berisi pilihan antara membuat Runtime Repository baru dan menghapus Runtime Repository yang ada. Pilih untuk membuat baru kemudian klik Next.
5. Pilihan pada form selanjutnya adalah tempat dimana RR akan diinstall, pilih opsi pertama karena RR akan diinstall pada skema yang sudah ada.
6. Selanjutnya adalah form untuk mengisi user dan password pemilik dari Warehouse Repository (WR). User yang digunakan adalah **user_rt**.
7. Form selanjutnya adalah form untuk lokasi tablespace, klik Next untuk melanjutkan.
8. Dari form tablespace, kemudian akan muncul form pilihan untuk membuat user Runtime Access. User ini digunakan sebagai media untuk mengakses RR. Pilih pilihan pertama, karena belum ada user dengan hak ini sebelumnya.
9. Isikan pada field user name dan password yang akan digunakan untuk mengakses RR. Username yang dibuat adalah **user_ra**.
10. Kemudian tentukan port yang digunakan untuk mengakses server, secara default adalah **4040**. Klik *Next* untuk melanjutkan.

11. Form terakhir adalah form ringkasan, klik *Finish* untuk menginstall RR dan membuat semua user yang telah didefinisikan sebelumnya.



Gambar 4.2 Ringkasan Runtime Repository

12. Selanjutnya akan muncul pesan untuk membuat Target Schema, pilih *Yes* untuk menentukan skema yang akan dijadikan target.
13. Secara default user yang digunakan adalah rt_owner. Lihat gambar 4.3 untuk ringkasannya.



Gambar 4.3 Ringkasan Target Schema

4.2 Menyiapkan Oracle Transparent Gateway

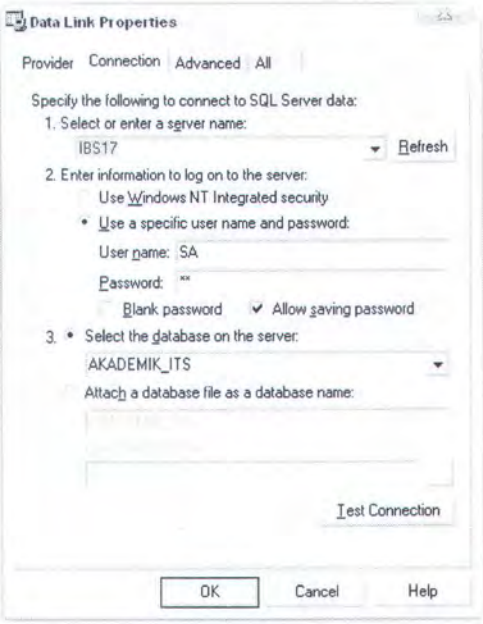
Oracle Tramsparent Gateway digunakan Oracle untuk berkomunikasi dengan sistem selain Oracle. Bersama-sama dengan *Heterogenous Service* (HS), Transparent Gateway menyediakan koneksi yang transparan bagi Oracle dengan sistem selain Oracle, sehingga user merasa dia terkoneksi langsung dengan database sumber. Ada beberapa tahapan atau langkah untuk menyetting atau mengaktifkan *Oracle Transparent Gateway*.

4.2.1 Membuat File UDL (Universal Data Link)

File *Universal Data Link* adalah file untuk membuat koneksi ke sebuah database yang didukung oleh OLEDB. Untuk membuat file ini ikuti langkah-langkah berikut:

1. Pada sebarang *folder* buat sebuah berkas (*file*) *text* (klik kanan > New > Text Document). Misalnya file *text* yang dibuat ada pada drive D:\Oracle.
2. Ubah nama *file* text tersebut menjadi *file* berekstensi .udl, misal: db.udl
3. Klik dua kali file UDL yang baru dibuat dan kemudian akan muncul jendela dengan 4 Tab (Provider, Connection, Advanced, dan All).
4. Pilih tab *Provider*, kemudian pilih jenis database yang akan kita koneksikan. Untuk Tugas Akhir ini RDBMS yang digunakan adalah Ms SQL Server.

5. Pilih tab *Connection* dan masukkan properti yang diminta untuk melakukan koneksi ke database (Ms SQL Server).



Gambar 4.4 Setting Koneksi UDL File

6. Klik tombol "Test Connection" untuk mengecek keberhasilan koneksi. Jika berhasil maka akan muncul jendela bertuliskan "*Test Connection Succeeded*"
7. Klik OK untuk mengakhiri pembuatan UDL file

4.2.2 Mengubah File inittg4msql.ora

File ini merupakan file konfigurasi untuk *Transparent Gateway* yang digunakan oleh OWB untuk berhubungan dengan database lain. File ini biasanya terletak pada lokasi *ORA_HOME/tg4msql/admin*.

Setelah membuka file tg4msql ubah nilai parameter menjadi terlihat seperti di bawah ini:

```
#--Ini Baris komentar--
#HS_FDS_CONNECT_INFO=nama_host.nama_db
#-------------------------
HS_FDS_CONNECT_INFO=10.126.14.54.AKADEMIK_ITS
HS_FDS_TRACE_LEVEL=OFF
HS_FDS_RECOVERY_ACCOUNT=RECOVER
HS_FDS_RECOVERY_PWD=RECOVER
```

4.2.3 Mengubah File listener.ora

File listener.ora memuat daftar listener yang dibaca oleh proses lsnrctl. Letak file ini ada pada folder admin tepatnya di *ORA_HOME/Network/Admin*. Agar bisa mendeteksi *Transparent Gateway*, perlu ditambahkan baris berikut di bawah parameter SID_LIST_LISTENER

```
(SID_DESC =
    (SID_NAME = hsoledb)
    (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
    (PROGRAM = hsolesql)
  )
```

Sehingga secara keseluruhan parameter SID_LIST_LISTENER menjadi:

```
SID_LIST_LISTENER =
 (SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = hsoledb)
    (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
    (PROGRAM = hsolesql)
  )
 )
```

4.2.4 Mengubah File tnsnames.ora

Pada file tnsnames.ora tambahkan satu parameter lagi yang menunjukkan bahwa ada satu tnsnames yang menggunakan HS (*Heterogenous Service*) *System*. File tnsnames.ora terletak satu folder dengan file listener.ora. Berikut ini adalah baris kode yang ditambahkan

```
simits =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = ibs17)(PORT = 1521))
  (CONNECT_DATA =
   (SID = hsoledb)
  )
 (HS=)
 )
```

4.2.5 Mengubah File initholedb.ora

File ini berfungsi untuk meletakkan parameter yang menunjukkan informasi koneksi yang akan dipakai oleh Transparent Gateway. Berikut adalah baris yang ditambahkan atau parameter yang diubah pada file initholedb.ora (terletak di folder *ORA_HOME/hs/admin*).

```
HS_FDS_CONNECT_INFO ="UDLFILE=D:\\oracle\\db.udl"
HS_FDS_TRACE_LEVEL = 0
```

4.2.6 Membuat Database Link

Setelah file-file yang harus di-setting ulang siap, kemudian sebagai "jembatan" diperlukan sebuah database link.

1. Masuk ke SQLPLUS atau OEM
2. Gunakan perintah (jika menggunakan SQLPLUS) untuk membuat database link secara manual:

```
Create Database link "simits" connect to "sa" identified by "sa" using 'simits'
```

3. Uji database link yang sudah dibuat dengan melakukan query sederhana.

```
C:\WINDOWS\system32\cmd.exe - sqlplus system/ta

SQL*Plus: Release 10.1.0.2.0 - Production on Wed Jan 18 00:08:38 2006

Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> select * from "agama"@simits;

A AG_Agama
- --------------------
_ - data kosong -
1 Islam
2 Katolik
3 Protestan
4 Hindu
5 Budhal
9 Lain

7 rows selected.

SQL>
```

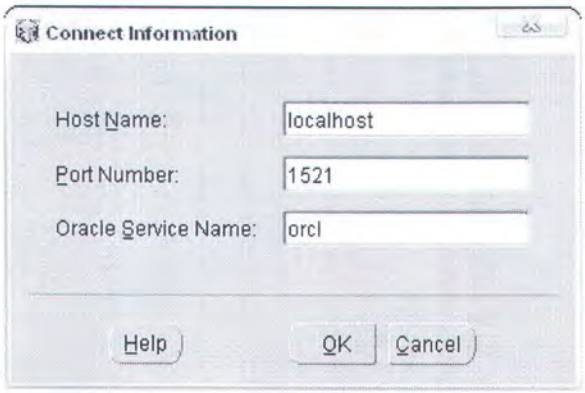
Gambar 4.5 Query sederhana untuk menguji Database Link

Setelah *database link* siap, maka *Transparent Gateway* sudah terkoneksi dengan sistem non-Oracle, dalam hal ini adalah Microsoft SQL Server yang terletak di komputer IBS14.

4.3 Log in ke OWB

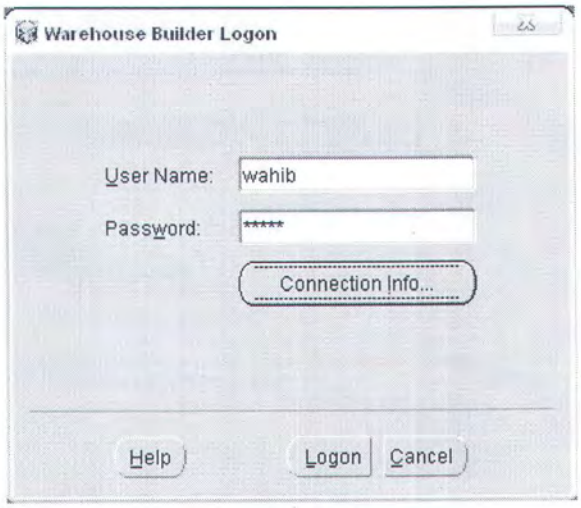
Untuk menampilkan OWB Client pilih **Start > Program Files > Oracle – OWB10g > Warehouse Builder > OWB Client.**

Sebelum Login, tentukan dahulu informasi koneksi pada OWB Client.



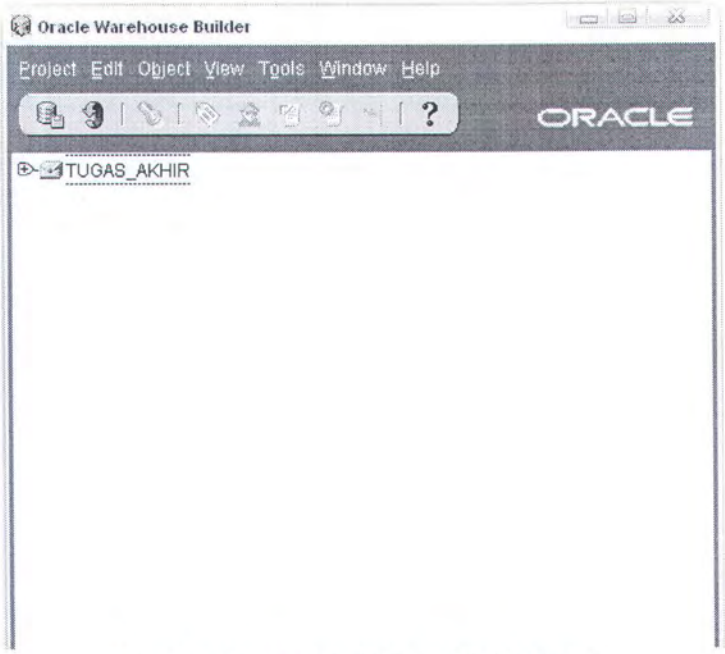
Gambar 4.6 Informasi Koneksi

Setelah informasi koneksi diisikan, maka log in ke OWB Client bisa dilakukan.



Gambar 4.7 Form Login OWB

Jika login berhasil maka akan muncul jendela utama OWB Client seperti pada gambar 4.8.



Gambar 4.8 Jendela Utama OWB Client

4.4 Membuat Project

Secara default OWB Client sudah membuat satu project pada saat pertama kali masuk/login. Untuk membuat project baru, pilih menu **Project > Create Project**.

4.4.1 Menentukan Lokasi Sumber

Agar OWB Client mengenali sumber database yang akan dijadikan sumber perlu ditentukan dulu lokasinya pada OWB Client.

1. Klik/urai simpul **Database > Other**
2. Klik kanan **Locations** dan pilih **Create Location**
3. Ikuti wizard yang muncul, beri nama lokasi SQL_SRC.
4. Klik Finish untuk mengkahiri. Dan dibawah simpul Locations akan muncul simpul baru dengan nama lokasi yang diberikan.

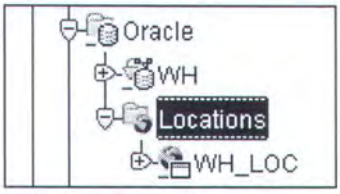
Gambar 4.9 Lokasi Sumber

4.4.2 Menentukan Lokasi Tujuan

Seperti juga lokasi sumber, lokasi target juga perlu di tentukan, yaitu dengan cara:

1. Urai simpul Databases > Oracle
2. Klik kanan pada simpul Locations dan pilih Create Oracle Location
3. Beri nama lokasi target WH_LOC, dan pada form berikutnya tentukan jenis database Oracle yang digunakan beserta versinya.

4. Klik Finish untuk mengakhiri dan dibawah simpul Locations akan muncul simpul baru bernama WH_LOC



Gambar 4.10 Lokasi Target

4.4.3 Membuat Modul Sumber (Ms SQL Server)

Untuk membuat modul sumber ikuti langkah-langkah berikut:

1. Urai simpul Databases > Other.
2. Klik kanan pada simpul Other, dan pilih Create Other Gateway Source Module.
3. Beri nama modul SQL_SRC
4. Masukkan parameter-parameter link setelah menekan tombol New Link. Pilih SQL*Net dan gunakan nama database link yang telah dibuat ("simits") pada field isiannya.
5. Jangan lupa untuk klik tombol *Test and Create* untuk menguji link sekaligus membuatnya.
6. Klik OK, selanjutnya pada form Connection Information ubah schema dengan menuliskan nama schema atau memilihnya dengan menekan tombol "Change Schema".

Gambar 4.11 *Link* Sumber

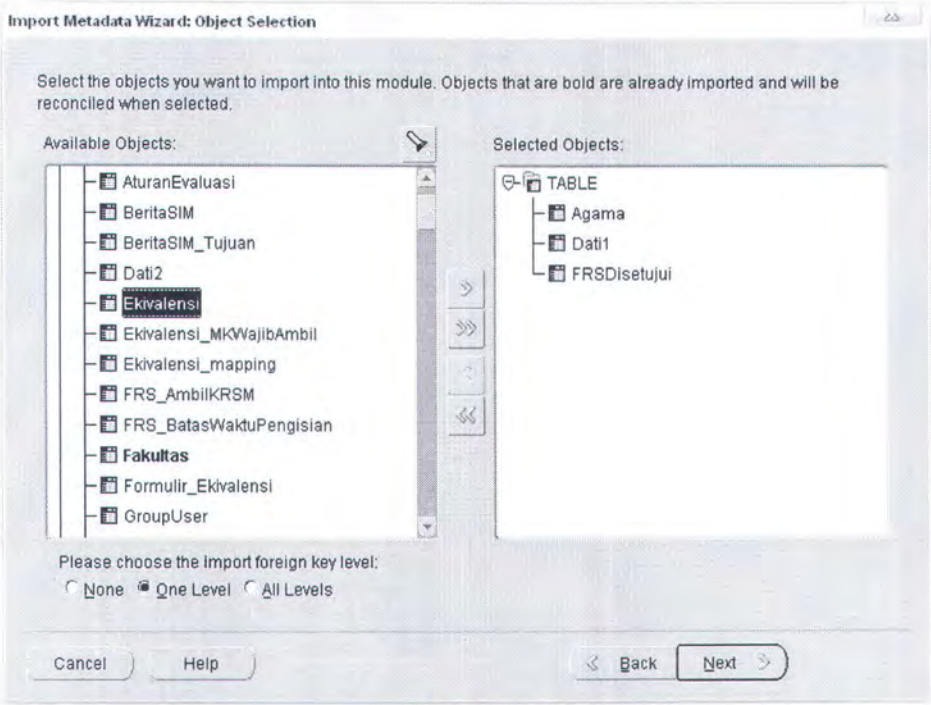
7. Tentukan lokasi sumber, kemudian klik next.
8. Pada form summary, jangan lupa untuk mencentang pilihan Proceed to The Import Metadata Wizard jika ingin melanjutkan dengan mengimport metadata. Kemudian Klik Finish.

4.4.4 Mengimpor Obyek dari Sumber

Sebelum obyek-obyek database pada sumber bisa digunakan oleh OWB, obyek-obyek tersebut harus diimpor terlebih dahulu.

1. Klik kanan pada modul SQL Server yang telah dibuat sebelumnya, dan pilih Import.
2. Selanjutnya pilih jenis obyek data yang akan diimportkan ke OWB, ada 3 macam yaitu, Tabel, View dan pilihan untuk mencari menggunakan sinonim obyek. Klik Next untuk melanjutkan.
3. Selanjutnya pilih obyek yang tampil di kolom sebelah kiri ke sebelah kanan dengan mengklik tombol ».

4. Pilih Next dan pada form berikutnya akhiri dengan mengklik tombol *Finish.*



Gambar 4.12 Import Obyek Database

4.4.5 Membuat Oracle Module

Modul ini menyimpan definisi target dari skema. Definisi ini bisa dibuat dengan menggunakan *wizards* atau mengimpornya dari tabel eksternal secara manual. Untuk Tugas Akhir ini akan digunakan metode dengan menggunakan *wizards.*

1. Dari jendela navigasi OWB, urai titik pada simpul **Databases** dengan mengklik tanda plus (+) di kiri simpul **Databases.**
2. Kemudian klik kanan pada simpul Oracle dan pilih **Create Oracle Module.**
3. Pada halaman pembuka yang muncul klik/pilih **Next**

4. Pada form selanjutnya isikan informasi berikut :
   a. Name the Module : WH
   b. Module type : Warehouse Target
   c. Version : Oracle Database 8i/9i/10g
5. Pada halaman Informasi, perlu diisikan beberapa informasi agar metadata bisa diimpor ke dalam Warehouse Module. Jika sudah ada link yang telah dibuat sebelumnya, link tersebut bisa dipilih. Jika belum maka perlu dibuat Link baru dengan mengklik tombol **New Link.** Masukkan parameter-parameter berikut:

   DB Link Name : wh_link

   Hostname : localhost

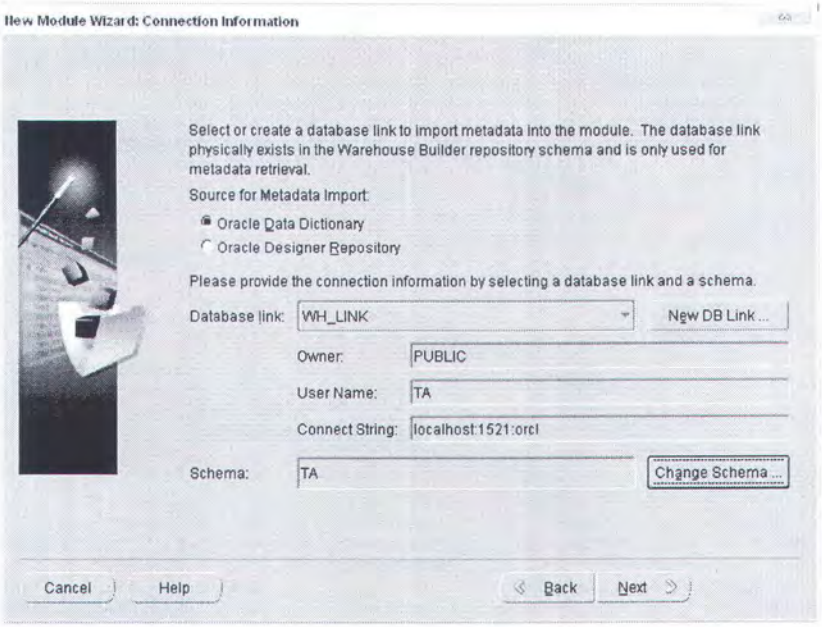
   Port number : 1521

   Oracle Service Name : orcl

   Username : ta

   Password : ta

   Jangan lupa untuk meng-klik tombol Create and Test untuk memastikan apakah link yang dibuat berhasil.



Gambar 4.13 Informasi Koneksi ke target Data Warehouse

6. Selanjutnya halaman location akan muncul. Pada halaman ini lokasi dari database sumber diidentifikasikan. Langkah ini bersifat pilihan, penentuan lokasi sumber bisa dilakukan dengan cara lain (4.3.1), klik **Next** untuk melanjutkan.
7. Halaman ini memuat seluruh parameter yang telah dimasukkan, jika sudah benar semua klik tombol **Finish**.

4.5 Membuat Dimensi

Untuk Tugas Akhir ini akan dibuat beberapa dimensi. Tiap dimensi mewakili tabel yang ada pada skema (gambar skema). Fungsi utama dimensi adalah untuk mengorganisir dan mengindeks data untuk *cube*. Berikut langkah-langkah membuat Dimensi menggunakan OWB :

1. Urai simpul WH, dan klik kanan pada simpul Dimensions.

2. Isikan parameter berikut pada form yang sesuai:

   a. Nama Dimensi : Waktu

   b. Nama *Level* : ThAkad

      ThAkad → char(4)

      Semester

      Id → char(1)

      semester → varchar(50)

   c. Nama Hirarki : h_wkt

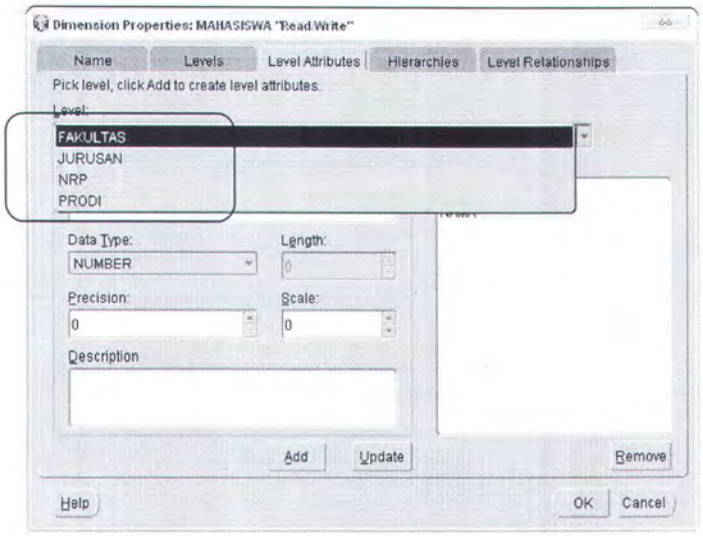
   d. Susunan Hirarki : ThAkad > Semester

3. Ulangi proses pembuatan dimensi dengan parameter-parameter dibawah ini.

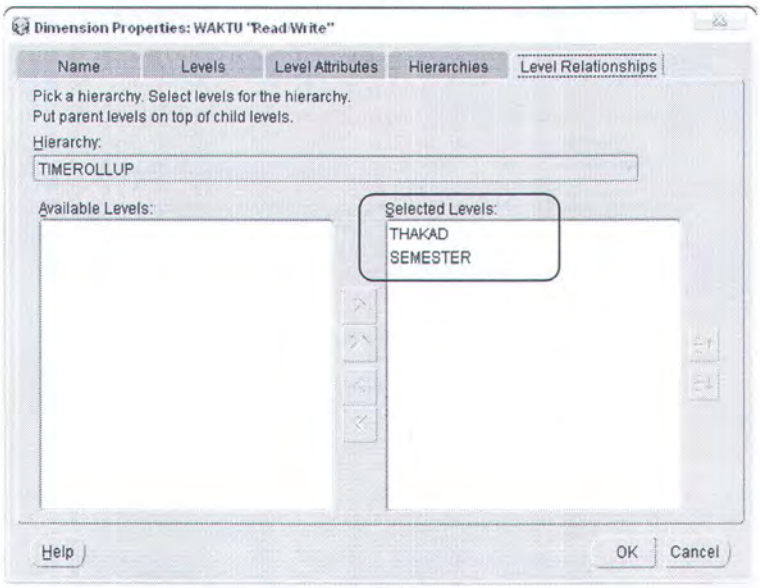
Tabel 4.1 Dimensi Lain

Dimensi : Mahasiswa	
Level : Fakultas	Id → char(1)
	NamaFakultas → varchar(50)
Level: Jurusan	Id → char(1)
	NamaJurusan → varchar(50)
Level : Prodi	Id → char(3)
	Prodi → varchar(50)
Level : NRP	Nrp → char(10)
Hirarki	h_mhs
Susunan Hirarki	Fakultas > Jurusan > Prodi > NRP

Satu lagi dimensi yang harus dibuat untuk skema totSKS yaitu dimensi Dosen. Atribut dimensi ini sama dengan atribut dimensi mahasiswa kecuali NRP diganti dengan NIP bertipe data Char(9).



Gambar 4.14 *Level* Dimensi Mahasiswa



Gambar 4.15 Susunan Hirarki Dimensi Waktu

4.6 Membuat Cube

Cube merupakan penerjemahan dari star skema, desain hubungan antar tabel yang akan digunakan untuk menggambarkan Data Warehouse yang akan dibuat. Sebelum membuat cube ada beberapa hal yang harus dipenuhi, yang pertama adalah keberadaan dimensi, hal ini bersifat mutlak karena pada cube ada

MILIK PERPUSTAKAAN ITS

tabel fakta (*Fact Table*) yang menampung *Primary Key* (PK) dari tiap-tiap dimensi untuk menjadi *field* pada Cube, ditambah dengan beberapa *measure.*

Untuk Tugas Akhir ini akan dibuat 2 cube sebagaimana mengacu pada star schema (gambar 3.2) dan (gambar 3.3).

4.6.1 Cube Nilai

Cube adalah representasi tabel fakta yang ada pada star schema. Cube mempunyai kolom-kolo yang terdiri dari dua jenis yaitu Measure dan *Foreign Key*. Measure pada cube nilai adalah Indeks Prestasi Semester, sedangkan *Foreign Key*-nya adalah *Primary Key* dari masing-masing dimensi yang terhubung dengan tabel fakta, dalam hal ini adalah Dimensi Mahasiswa (NRP) dan Dimensi Waktu (SemID).

Langkah-langkah membuat sebuah cube dalam OWB adalah sebagai berikut:

1. Klik kanan simpul Cube dan pilih Create Cube
2. Tekan tombol Next pada form yang muncul, pada form selanjutnya masukkan CNILAI sebagai nama cube, klik Next.
3. Pilih data berikut dan klik Add untuk menjadikannya *Foreign Key* pada cube.

Tabel 4.2 *Foreign Key* pada cube Nilai

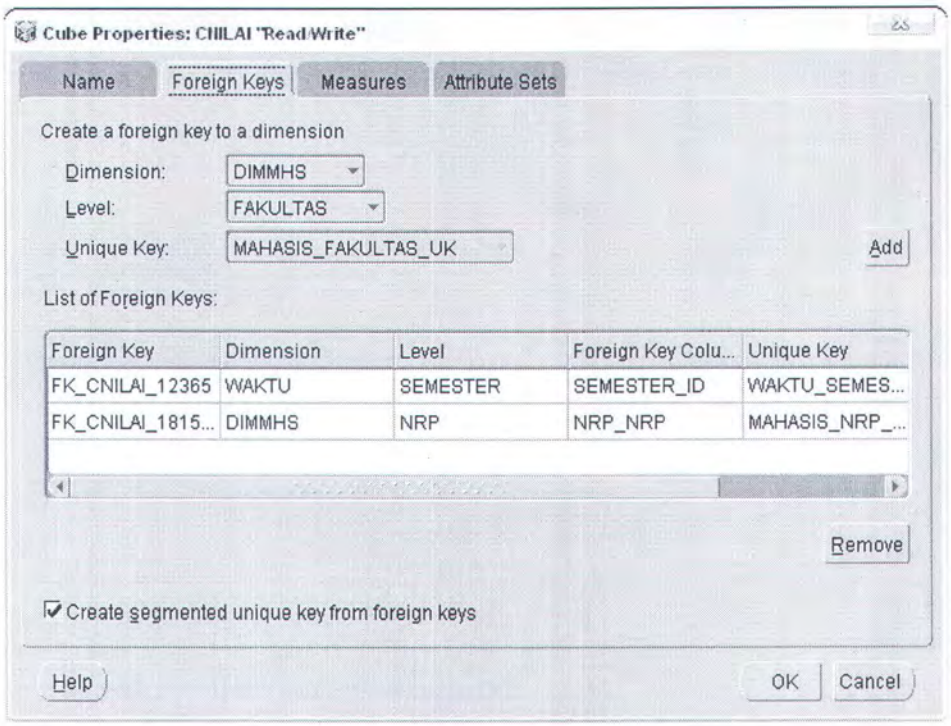
Dimension	: DIMMHS
Level	: NRP
Unique Key	: MAHASIS_NRP_UK

Dimension	: WAKTU
Level	: SEMESTER
Unique Key	: WAKTU_SEMESTER_UK

Tabel 4.3 *Foreign Key* pada cube TotSKS

Dimension	: DOSEN
Level	: NIP
Unique Key	: DOSEN_NIP_UK
Dimension	: WAKTU
Level	: SEMESTER
Unique Key	: WAKTU_SEMESTER_UK

Jangan lupa untuk memberi centang pada pilihan "*Create segmented unique key from Foreign Keys*"



Gambar 4.16 Foreign Key pada Cube

4. Di form selanjutnya, masukkan satu measure IPS dengan tipe data FLOAT dengan tingkat presisi 38. Klik Next untuk melihat ringkasan cube, kemudian klik Finish untuk mengakhiri.

List of measures:

Name	Data Type	Length	Precision	Scale
IPS	FLOAT		38	

Gambar 4.17 Measure IPS

4.7 Membuat Mapping

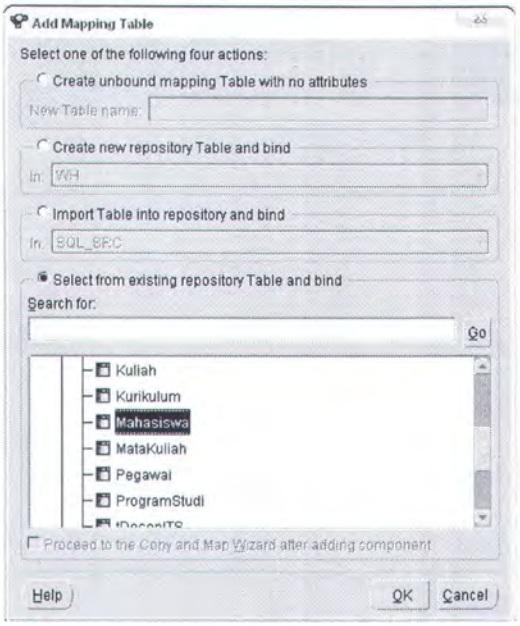
Pemetaan menggambarkan serangkaian operasi yang mengekstraksi data dari sumber, mentransformasi dan me-*load*-nya ke dalam tujuan [OWB-03]. Pemetaan juga memvisualisasikan alur data dan operasi-operasi (*join*, *aggregrate*, *function*, dll) yang dilakukan pada data.

Pemetaan biasanya dibuat sebanyak jumlah dimensi dan cube yang ada, karena tiap dimensi maupun tiap cube mempunyai alur perpindahan (pemrosesan) data yang berbeda untuk mendapatkan data yang ingin ditampilkan pada Data Warehouse.

4.7.1 Mapping Dimensi Waktu

Dalam dimensi waktu terdapat dua *level*, sehingga paling tidak dibutuhkan dua tabel yang berhubungan dengan masing-masing *level*. Untuk lebih jelasnya berikut ini adalah langkah-langkah pemetaan alur data pada dimensi waktu.

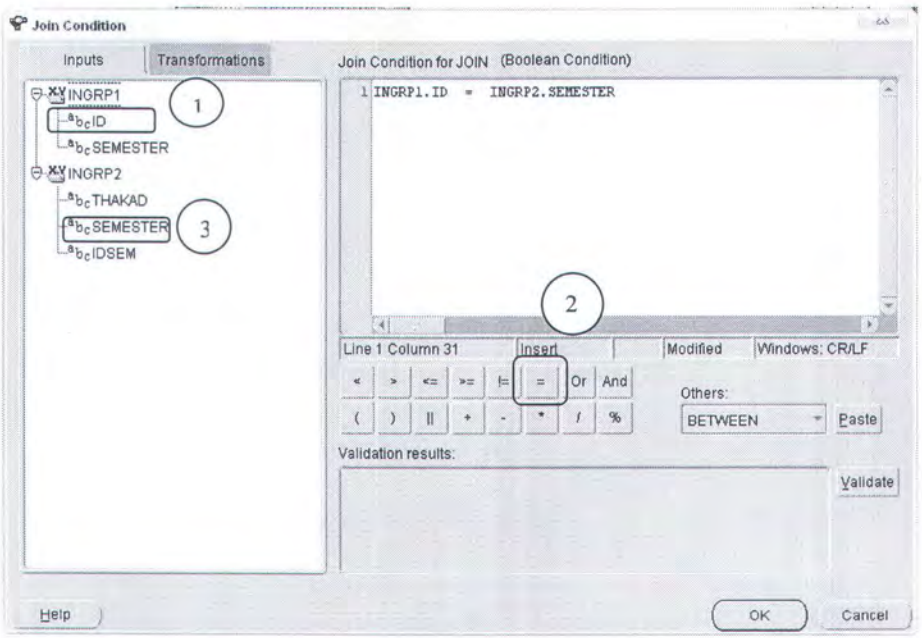
1. Seret icon ke area kerja, kemudian muncul gambar.



Gambar 4.18 Memilih Tabel

2. Pilih tabel val_semester yang akan dijadikan sumber data, untuk memilih lebih dari satu tabel gunakan tombol Ctrl pada papan ketik. Klik OK untuk konfirmasi pemilihan tabel.
3. Selanjutnya dengan menggunakan icon pilih view thakad dengan cara yang sama dengan (2).
4. Agar lebih memudahkan pemetaan, dengan menyeret icon pilih dimensi WAKTU untuk dijadikan tujuan akhir dari pemetaan.
5. Pada dimensi waktu terdapat atribut semester_id yang merupakan gabungan dari tahun akademik dan id semester. Untuk itu perlu dilakukan operasi penggabungan data dengan menggunakan Expression. Untuk menampilkannya seret icon . Sebagai data masukan seret KE_IDSemester dari view thakad ke Expr.Ingrp1 begitu pula dengan KE_Tahun → Expr.Ingrp1.

6. Sebagai keluaran (output) klik kanan pada Expr kemudian pilih Edit. Pada tab Output Attributes tambahkan atribut IDSEM dengan tipe data char sepanjang 5 karakter. Klik OK.
7. Dari dua obyek yang ada (val_semester dan thakad) perlu digabungkan agar bisa dipetakan pada Dimensi Waktu. Proses penggabungan dilakukan dengan operasi JOIN. Seret icon ke kanvas. Petakan seluruh atribut val_semester ke Join.Ingrp1. Petakan seluruh atribut thakad ke Jopin.Ingrp2, begitu juga dengan atribut output dari obyek Expr, petakan ke Join.Ingrp2.
8. Pemberian kondisi penggabungan dilakukan dengan klik kanan pada toolbar dan pilih Operator Properties. Sorot pilihan Join Condition kemudian klik [...] yang akan memunculkan jendela berikut.



Gambar 4.19 Kondisi operasi Join

9. Klik dua kali pada obyek yang diberi tanda pada gambar diatas sesuai urutan nomor akan memberikan hasil INGRP1.ID = INGRP2.SEMESTER sebagai kondisi dari operasi JOIN. Klik OK
10. Petakan masing-masing keluaran dari obyek join, kecuali Join.Outgrp1.Semester_Id, ke obyek Dimensi Waktu.
11. Untuk memastikan pemetaan berjalan dengan baik, lakukan validasi dengan mengklik icon pada toolbar.
12. Periksa kembali jika masih muncul kesalahan atau peringatan dari hasil validasi.

4.7.2 Mapping Dimensi Mahasiswa

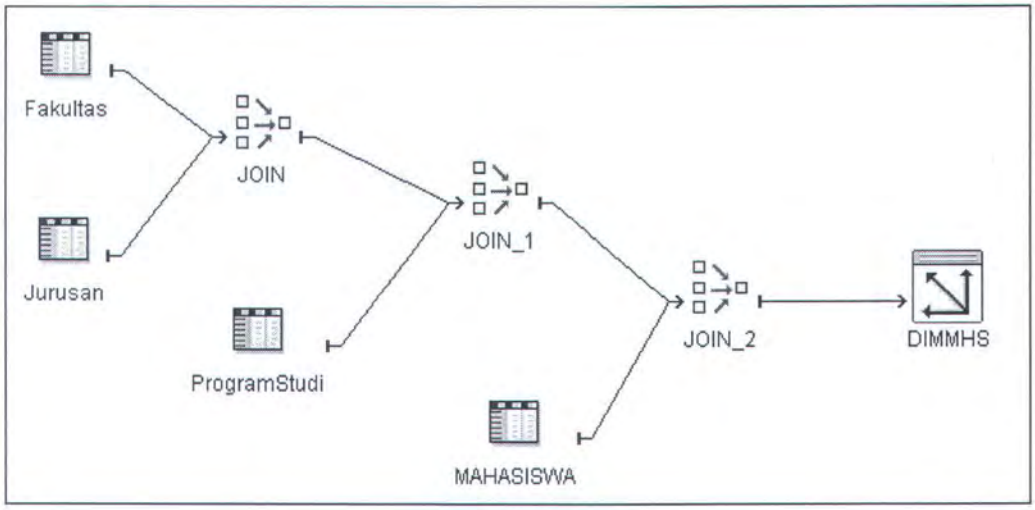
Dimensi Mahasiswa mempunyai 4 *level*, Fakultas, Jurusan, Prodi (Program Studi), dan NRP. Kecuali NRP, tiap *level* mempunyai dua atribut yaitu id dan nama. Sedangkan *level* NRP hanya mempunyai 1 atribut yaitu NRP.

Untuk memetakan aliran data menuju Dimensi Mahasiswa (DIMMHS), dibutuhkan 4 tabel yaitu, tabel Fakultas, tabel Jurusan, tabel Program Studi, dan tabel Mahasiswa.

Langkah-langkah pemetaan :

1. Seret icon ke kanvas, dari form yang muncul pilih tabel Fakultas, Jurusan, Program Studi dan tabel Mahasiswa.
2. Jangan lupa untuk tampilkan juga Dimensi Mahasiswa (gunakan icon) sebagai acuan dari operasi yang akan diterapkan.

3. Seret icon (join) kemudian gabungkan tabel Fakultas dan tabel Jurusan melalui operator JOIN dengan menggunakan syarat JU_FA_ID = FA_ID
4. Hasil dari operasi JOIN gabungkan dengan tabel Program Studi melalui operator JOIN_1, gunakan syarat JU_ID = PS_JU_ID.
5. Kemudian hasil dari operasi JOIN_1 digabungkan dengan tabel Mahasiswa melalui operasi JOIN_2. Kondisi yang digunakan adalah (FA_ID || JU_ID || PS_ID) = (SUBSTR(MA_Nrp,1,1) || SUBSTR(MA_Nrp,2,1) || SUBSTR(MA_Nrp,5,3)).
6. Dari hasil JOIN_2 petakan masing-masing atribut yang sesuai ke Dimensi Mahasiswa (DIMMHS).



Gambar 4.20 Pemetaan Dimensi Mahasiswa

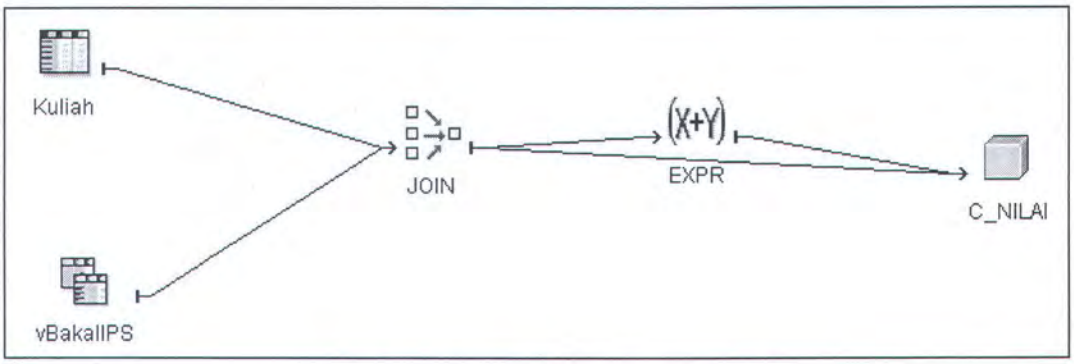
4.7.3 Mapping Cube Nilai

Sama juga dengan cara pemetaan dimensi, hanya saja tujuan dari pemetaan ini berupa cube.

Langkah-langkah untuk memetakan cube Nilai adalah sebagai berikut:

1. Seret icon kemudian pilih view vBakalIPS, dan pilih tabel Kuliah dengan menggunakan icon .
2. Selanjutnya tampilkan cube Nilai yang akan jadi tujuan dengan menggunakan icon .
3. Untuk menggabungkan tabel Kuliah dan view vBakalIPS, gunakan operator join. Hubungkan baris KU_IDSemester, KU_Tahun, dan KU_MA_Nrp dari tabel Kuliah Ingrp1 pada Join, dan baris NRP, Sem, ThAkad, TotSKS, TotSKSNilai dari view vBakalIPS ke Inngrp2 pada operator Join. Gunakan kondisi (Kuliah.KU_KE_IDSEMESTER = vBakalIPS.SEM And vBakalIPS.THAKAD = Kuliah.KU_KE_TAHUN And vBakalIPS.NRP = Kuliah.KU_MA_NRP) sebagai syarat dari operasi Join.
4. Untuk menggabungkan kolom Tahun dan IdSemester sebagai masukan dari kolom cube Semester_ID, gabungkan dengan menggunakan operasi expression.
5. Klik kanan pada toolbar expression kemudian pilih edit.
6. Pada tab Output Attributs tambahkan satu variabel ThSem bertipe data char sepanjang 5 karakter. Akhiri dengan mengklik Ok.
7. Klik kanan pada kolom atribut ThSem di bagian bawah OUTGRP1 dari operator expression dan pilih attribute properties.
8. Pada jendela yang muncul, pilih Expression dan klik [...]

9. Isikan operasi yang akan dilakukan dengan masukan. Klik Ok untuk mengakhiri.
10. Selanjutnya tambahkan keylookup dan pilih waktu sebagai kuncinya. Petakan hasil keluaran dari expression ke grup masukan keylookup.
11. Terakhir petakan masing masing kolom yang sesuai.
    a. vIPS.IPS ➔ CNILAI.IPS
    b. vIPS.MA_NRP ➔ CNILAI.NRP_NRP
    c. WAKTU.SEMESTER_ID ➔ CNILAI.SEMESTER_ID
12. Lakukan validasi untuk memastikan tidak ada kesalahan atau peringatan yang muncul.



Gambar 4.21 Pemetaan ke obyek Cube

4.8 Membuat Alur Proses

Alur proses menunjukkan rangkaian proses yang harus dieksekusi oleh OWB untuk me-loading data dari database sumber. Ada empat tahapan yang harus dilakukan sebelum pengaturan alur proses bisa dilakukan.

1. Menentukan lokasi Alur Proses, yaitu dengan mengklik kanan simpul Locations yang berada dibawah simpul Process Flow, dan

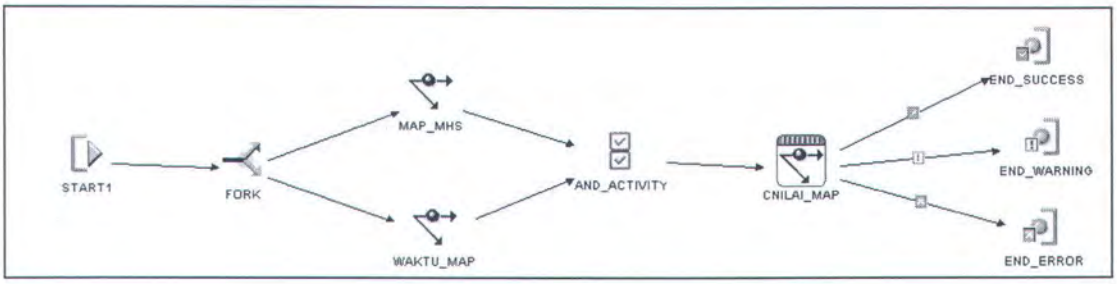
melengkapi field yang dibutuhkan yaitu nama (PF_LOC) dan tipenya (Oracle WorkFlow 2.6.3)

2. Selanjutnya adalah membuat Modul Alur Proses.
    a. Klik kanan pada simpul Proces Flow, dan pilih Create Process Module.
    b. Isikan nama modul yang dikehendaki, misalnya PF_MOD, klik Next.
    c. Pilih lokasi yang sudah dibuat PF_LOC, dan lanjutkan ke form berikutnya.
    d. Klik Finish pada form summary untuk mangakhiri pembuatan modul alur proses.
3. setelah modul dibuat, yang perlu dilakukan selanjutya adalah membuat paket alur proses. Caranya:
    a. Klok kanan pada modul yang telah dibuat dan pilih create process package.
    b. Gunakan PF_PKG sebagai nama paketnya. Klik Ok untuk mengakhiri.
4. Langkah terakhir adalah membuat prosesnya.
    a. Pada PF_PKG klik kanan dan pilih Create Process.
    b. Beri nama PF_PROC dan klik Ok.

Untuk mendesain alur proses klik dua kali pada PF_PROC dan akan muncul form untuk mendesain alur proses. Alur proses OWB selain untuk

menggambarkan rangkaian aliran proses juga bisa berfungsi untuk mengirimkan laporan dari hasil rangkaian proses ke email yang berkepentingan.

1. Seret icon untuk memecah alur menjadi dua
2. Hubungkan dengan menggunakan icon (transisi) antara Start dengan Fork
3. Seret icon kemudian pilih MAP_MHS, WAKTU_MAP dan CNILAI_MAP.
4. Selanjutnya teruskan proses dengan menghubungkan Fork ke MAP_MHS dan WAKTU_MAP.
5. Seret icon (AND Activity) untuk menggabungkan dua aliran proses.
6. Gabungkan kedua obyek (no. 4) dengan menghubungkannya pada AND Activity menggunakan
7. Hubungkan Proses AND Activity dengan proses CNILAI_MAP.
8. Kemudian seret icon dan dari form yang muncul beri centang pada pilihan Warning End Activity dan Error End Activity.
9. Pari CNILAI_MAP hubungkan dengan ke tiga END Activity. Pada tiap alurnya set kondisi tarnsisinya sesuai dengan keadaan akhirnya.



Gambar 4.22 Alur Proses

Transition View

FROM	TO	CONDITION	DESC
CNILAI_MAP	END_WARNING	WARNING	
CNILAI_MAP	END_ERROR	ERROR	
CNILAI_MAP	END_SUCCESS	SUCCESS	
AND_ACTIVIT	CNILAI_MAP		

Gambar 4.23 Kondisi END Activity

4.9 Proses Penggunaan (Deploying) dan Eksekusi

Proses *deploy* adalah proses loading metadata dalam rangkaian proses ETL untuk membuat sebuah Data Warehouse. Dengan mengikuti pemetaan sesuai Pemetaan yang telah dibuat sebelumnya dan mengikuti alur proses yang sudah didesain metadata akan di-*load* ke Data Warehouse.

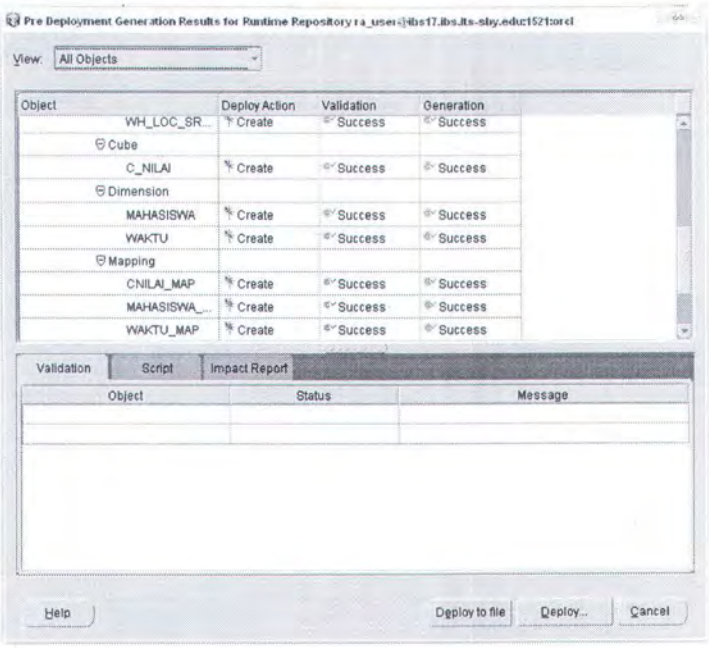
1. Pada jendela utama OWB, pilih Project > Deployment Manager.
2. Jika ada lebih dari satu koneksi Repository yang dibuat sebelumnya, pilih koneksi dimana Data Warehouse akan diimplementasikan. Untuk Tugas Akhir ini hanya ada satu koneksi yaitu RR_CON.

3. Selanjutnya, dari 3 lokasi yang muncul pada panel sebelah kiri, yang perlu didaftarkan ada dua lokasi, yaitu lokasi alur proses (PF_LOC) dan lokasi Data Warehouse (TRGT_LOC).
4. Untuk mendaftarkan lokasi alur proses klik kanan pada PF_LOC dan pilih register. Masukkan nilai berikut pada form yang muncul

Schema Name	: owf_mgr
Password	: owf_mgr
Service Name	: orcl
Host name	: ibs14.ibs.its-sby.edu
Port Number	:1521

   Begitu juga dengan TRGT_LOC, gunakan username user_ts dan kata sandi user_ts untuk mendaftarkannya.
5. Pada pilihan PF_LOC disebelah kanan terdapat obyek yang akan di-*deploy*, begitu juga dengan TRGT_LOC mempunyai beberapa obyek yang siap untuk di-*deploy*.
6. Klik icon pada toolbar untuk men-*deploy* setelah sebelumnya ditentukan dulu aksi pen-*deploy*-an dengan mengklik tombol "Default Action".

7. Tunggu sampai proses *create* berhenti, dan lanjutkan dengan mengklik tombol D*eploy*.



Gambar 4.24. Kondisi setelah Proses Create

8. Jika semua sukses dan lancar maka di bagian kanan akan ada keterangan yang menyatakan kondisi proses pen-*deploy*-an.
9. Langkah selanjutnya adalah pengeksekusian obyek pemetaan untuk mentransfer data dari sumber. Dengan memilih obyek PF_LOC klik untuk mengeksekusi semua obyek pemetaan. Tunggu sampai proses selesai.
10. Data yang sudah di-load ke Data Warehouse bisa dilihat melalui Oracle Enterprise Manager. Tabel yang dibuat berada dibawah skema user_ts.

Table Editor : "USER_TS"."WAKTU" - system@ORCL

SEMESTER_ID	SEMESTER_SEMESTER	SEMESTER_SEM_ID	THAKAD_THAKAD
19942	Genap	2	1994
19952	Genap	2	1995
19962	Genap	2	1996
19972	Genap	2	1997
19982	Genap	2	1998
19992	Genap	2	1999
20 2	Genap	2	20
20001	Gasal	1	2000
20003	Pendek	3	2000
20012	Genap	2	2001
20021	Gasal	1	2002
20023	Pendek	3	2002
20032	Genap	2	2003
20041	Gasal	1	2004
20043	Pendek	3	2004
20052	Genap	2	2005
20142	Genap	2	2014
19941	Gasal	1	1994

Execute time (s): 0.268 Rows returned: 35 Apply Revert Show SQL Close Help

Gambar 4.25 Data Dimensi Waktu

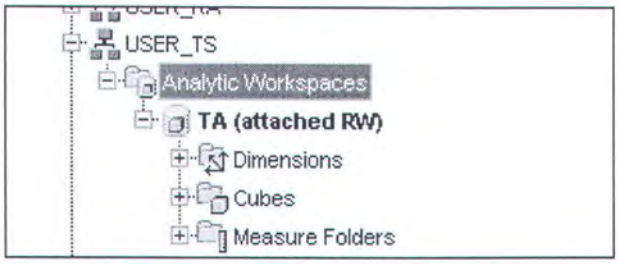
4.10 Integrasi Dengan OLAP

4.10.1 Menggunakan Analytic Workspace Manager

Integrasi Data Warehouse dengan teknologi Online Analytical Processing (OLAP) bisa dilakukan dengan menggunakan beberapa aplikasi, salah satunya adalah dengan aplikasi Analytic Workspace Manager (AWM) buatan Oracle.

Berikut ini adalah cara pengintegrasiannya.

1. Masuk ke AWM dengan menggunakan username dan password user_ts
2. Urai schema > user_ts, kemudian klik kanan pada simpul Analytic Workspace untuk membuat Workspace baru.
3. Pada form yang muncul beri nama workspace dengan nama TA
4. Urai simpul TA yang telah terbentuk sampai muncul simpul Dimensions, Cubes, dan Measures Folder.



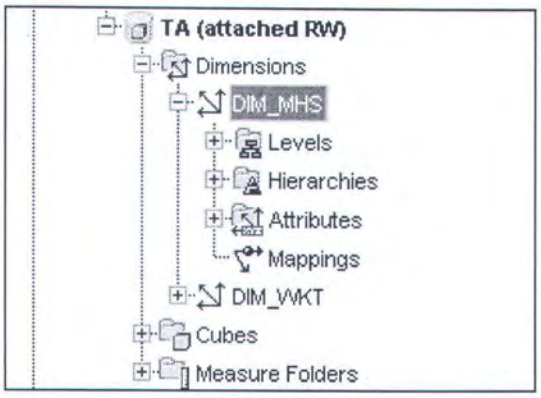
Gambar 4.26 Simpul TA

a. Klik kanan pada simpul Dimensions untuk membuat dimensi. Dimensi yang dibuat ada 2 yaitu, Dimensi Mahasiswa dan Waktu.

b. Urai simpul dimensi, dan klik kanan pada simpul Levels kemudian pilih Create Level untuk membuat level pada dimensi bersangkutan.

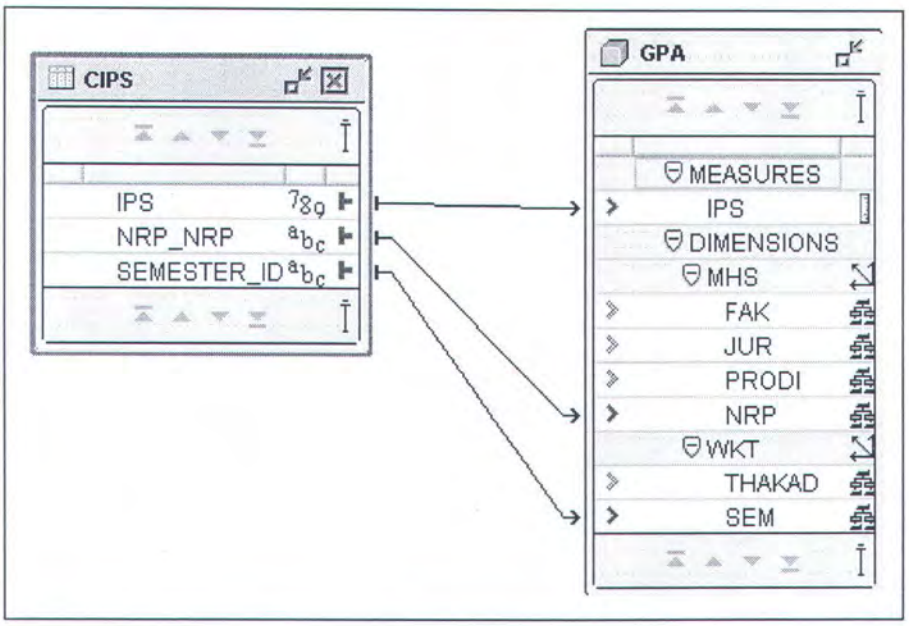
c. Untuk membuat hirarki, klik kanan pada simpul Hierarchies, kemudian pilih menu Create Hierarchy... untuk tiap dimensi cukup satu hirarki saja.

d. Untuk pemetaan klik simpul Mappings dan akan muncul dua kolom jendela tambahan di sebelah kanan. Dari kolom tengah urai simpul user_ts > Tables. Seret tabel Dim untuk dipetakan ke dimensi Mahasiswa dan tabel Waktu untuk Dimensi Waktu. Klik tombol Apply untuk menyimpan perubahan



Gambar 4.27 Simpul Dimensi Mahasiswa

5. Unuk membuat cube klik kanan simpul Cubes yang berada dibawah simpul Dimensions dan pilih *Create cube...* Pada tab General masukkan semua dimensi (Mahasiswa dan Waktu) yang akan terhubung dengan Cube dengan memindahkan dari kolom kiri ke kolom kanan. Pada tab Rule ubah parameter Operator dari SUM menjadi AVG pada semua Dimensi. Langkah terakhir untuk cube adalah memetakan data dari tabel fakta. Pilih simpul *Mappings* yang ada di bawah simpul Cube, seret tabel CIPS ke kolom paling kanan, dan petakan sesuai atributnya.



Gambar 4.28 Pemetaan Data Cube

6. Measure berada di bawah simpul cube, untuk Tugas Akhir ini diperlukan satu measure. Untuk membuatnya klik kanan simpul Measures dan pilih Create Measure. Masukkan nama measure, IPS, dan tentukan tipe datanya, Decimal.
7. Untuk me-loda data dari tabel sumber ke AWM, klik kanan simpul TA dan pilih "Maintain Analytic Workspace TA". Tunggu sampai selesai.
8. Untuk melihat hasil yang berupa tabel crosstab, klik kanan pada simpul IPS (dibawah simpul *Measures*) kemudian pilih "View Data IPS".

	PerSem									
	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Fakultas Matematika dan Ilmu Pengetahuan Alam	3	3	2	2	2	2	2	3	3	0
Fakultas Teknologi Industri		2	2	2	2	2	2	3	2	0
Fakultas Teknik Sipil dan Perencanaan			2	2	2	2	3	3	3	0
Fakultas Teknologi Kelautan	3		3	2	2	2	2	3	2	0
Fakultas Teknologi Informasi			3	3	3	3	3	3	3	0
Magister Manajemen Teknologi					3	3	3	3	3	0

Gambar 4.29 Crosstab IPS

BAB V

UJI COBA DAN EVALUASI

BAB V
UJI COBA DAN EVALUASI

Dalam bab ini akan dijelaskan coba aplikasi Data Warehouse yang telah terintegrasi dengan OLAP apakah berjalan dengan baik pada lingkungan yang telah ditentukan dan apakah operasi-operasi-nya berhasil.

5.1 Lingkungan Uji Coba

Lingkungan uji coba adalah komputer-komputer yang digunakan untuk uji coba aplikasi. Ada dua komputer yang akan digunakan, *server* dan *client*.

5.1.1 Server

Pada komputer yang bertindak sebagai server ter-*install* beberapa aplikasi yang dibutuhkan oleh OWB. Database sumber dan Database target juga berada di komputer ini. Komputer ini mempunyai spesifikasi sebagai berikut:

Prosesor	: Intel Pentium IV 2,8 GHz
RAM	: DDRRAM 1 Gb
Sistem Operasi	: Microsoft Windows XP Service Pack 2
Alamat IP	: 10.126.14.54
Aplikasi Pendukung	:

- Oracle Database Release 1 versi 10.1.0.4
- Oracle Warehouse Builder 10g Server
- Oracle Workflow 2.6.3

- Microsoft SQL Server 2000

5.1.2 Client

Pada sisi *client* diperlukan aplikasi pendukung seperti OWB Client, Analytic Workspace Manager, dengan spesifikasi lengkap seperti dibawah ini:

Prosesor : Intel Pentium IV 2.8 GHz

RAM : DDRRAM 1 Gb

Sistem Operasi : Microsoft Windows XP Service Pack 2

Alamat IP : 10.126.14.15

Aplikasi Pendukung :

- Oracle Warehouse Builder 10g Client
- Analytic Workspace Manager

5.2 Uji Coba

Uji coba dilakukan untuk menerapkan operasi-operasi dasar yang biasa dilakukan pada OLAP seperti *drill-down* dan *consolidation*, dan operasi ini dilakukan pada aplikasi pendukung AWM.

5.2.1 *Drill-down*

Operasi *drill-down* adalah operasi memecah salah satu parameter dalam cube menjadi sub-parameternya, contohnya adalah memecah parameter Fakultas menjadi jurusan-jurusan, atau memecah jurusan menjadi program studi-program studi.

Pertama, menampilkan tabel crosstab dari cube yang akan dikenai operasi. Tampilan awalnya biasanya menampilkan nilai data yang ada pada level tertinggi pada hirarki.

	PerSem									
	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Fakultas Matematika dan Ilmu Pengetahuan Alam	3	3	2	2	2	2	2	3	3	0
Fakultas Teknologi Industri		2	2	2	2	2	2	3	2	0
Fakultas Teknik Sipil dan Perencanaan			2	2	2	2	3	3	3	0
Fakultas Teknologi Kelautan	3		3	2	2	2	2	3	2	0
Fakultas Teknologi Informasi			3	3	3	3	3	3	3	0
Magister Manajemen Teknologi					3	3	3	3	3	0

Gambar 5.1 Tampilan awal Crosstab

Operasi *drill-down* bisa dilakukan dengan mengklik icon ▸ pada parameter yang akan dipecah. Misalnya, Fakultas Matematika dan Ilmu Pengetahuan Alam dipecah, maka hasilnya akan tampak pada gambar berikut.

	IPS						
	Fakultas Mat	Fisika	Matematika	Statistik	Kimia	Biologi	Fakultas
THAKAD_1995							
THAKAD_1996							
THAKAD_1997	3		3				
THAKAD_1998	2	2	2				
THAKAD_1999	2	2	2	3	2	1	
THAKAD_20							
THAKAD_200							
THAKAD_2000	2	2	1	2	2	1	
THAKAD_2001	2	2	2	1	2	1	
THAKAD_2002	2	2	1	2	2	2	

Gambar 5.2 Hasil operasi *Drill-down* pada Fakultas MIPA cube IPS

	CREDIT					
	20	2001	2002	2003	2004	2005
Fakultas Matematika dan Ilmu Pengetahuan Alam					119	946
Fakultas Teknologi Industri					57	1,822
Fakultas Teknik Sipil dan Perencanaan					308	1,221
Fakultas Teknologi Kelautan					276	610
Fakultas Teknologi Informasi					544	242
Teknik Informatika					373	148
Sistem Informasi					171	94

Gambar 5.3 Hasil operasi Drill-down pada cube totSKS

5.2.2 *Consolidation*

Operasi *Consolidation* adalah kebalikan dari operasi *drill-down*, yaitu menyatukan parameter-parameter yang setingkat menjadi parameter yang setingkat lebih tinggi. Misalnya dari parameter Program Studi menjadi Jurusan.

	PerSem									
	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Fakultas Matematika dan Ilmu Pengetahuan Alam	3	3	2	2	2	2	2	3	3	0
Fisika			2	2	2	2	2			
S1 Fisika			2	2	2	2	2			
S2 Fisika										
Matematika	3	3	2	2	2	2	2	2	2	0
Statistik				2	2	3	3	3	3	0
Kimia	3	2	3	2	2	2	3	3	3	0
Biologi				2	2	2	2	3	3	0
Fakultas Teknologi Industri		2	2	2	2	2	2	3	2	0
Fakultas Teknik Sipil dan Perencanaan			2	2	2	2	3	3	3	0
Fakultas Teknologi Kelautan	3		3	2	2	2	2	3	2	0
Fakultas Teknologi Informasi			3	3	3	3	3	3	3	0
Magister Manajemen Teknologi					3	3	3	3	3	0

Gambar 5.4 Hasil operasi *Consolidation* pada cube Nilai

	CREDIT					
	20	2001	2002	2003	2004	2005
Fakultas Matematika dan Ilmu Pengetahuan Alam					119	946
Fakultas Teknologi Industri					57	1,822
Fakultas Teknik Sipil dan Perencanaan					308	1,221
Fakultas Teknologi Kelautan					276	610
Fakultas Teknologi Informasi					544	242
Teknik Informatika					373	148
S1 Teknik Informatika					373	148
Sistem Informasi					171	94

Gambar 5.5 Hasil operasi Consolidation pada cube totSKS

5.2.3 Rotation/Pivoting

Operasi Rotasi pada OLAP adalah pengubahan sudut pandang, pada gambar 5.3 misalnya, variabel Fakultas berada pada sumbu Y dan variabel tahun akademik berada pada simbu X. Dengan operasi Rotasi/pivoting kedua variabel bisa berpindah sumbu, Fakultas ke sumbu X dan sebaliknya.

	PerSem					
	Fakultas Mate	Fakultas Tekr	Fakultas Tekr	Fakultas Tekr	Fakultas Tekr	Magister Man
1996	3			3		
1997	3	2				
1998	2	2	2	3	3	
1999	2	2	2	2	3	
2000	2	2	2	2	3	3
2001	2	2	2	2	3	3
2002	2	2	3	2	3	3
2003	3	3	3	3	3	3
2004	3	2	3	2	3	3
2005	0	0	0	0	0	0

Gambar 5.6 Hasil operasi Rotasi/Pivoting pada cube Nilai

	CREDIT				
	Fakultas Matemat	Fakultas Teknolog	Fakultas Teknik S	Fakultas Teknolog	Fakultas Teknolog
1996					
1997					
1998					
1999					
20					
200					
2000					
2001					
2002					
2003					
2004	119	57	308	276	544
2005	946	1,822	1,221	610	242

Gambar 5.7 Hasil operasi Rotation pada cube totSKS

5.2.4 *Slice/dice*

Operasi *slice/dice* memungkinkan untuk melihat data dengan terfokus pada variabel tertentu, misalnya nilai IPS di semua Fakultas pada tahun akademik 1998 saja. Gambar 5.5 mengilustrasikan operasi *slice/dice* pada OLAP.

Page Items TIME 1998

	PerSem
Fakultas Matematika dan Ilmu Pengetahuan Alam	2
Fakultas Teknologi Industri	2
Fakultas Teknik Sipil dan Perencanaan	2
Fakultas Teknologi Kelautan	3
Fakultas Teknologi Informasi	3
Magister Manajemen Teknologi	

Gambar 5.8 Hasil operasi Slice/Dice pada cube Nilai

MILIK PERPUSTAKAAN ITS

Page Items TIME 2004

	CREDIT
Fakultas Matematika dan Ilmu Pengetahuan Alam	119
Fakultas Teknologi Industri	57
Fakultas Teknik Sipil dan Perencanaan	308
Fakultas Teknologi Kelautan	276
Fakultas Teknologi Informasi	544

Gambar 5.9 Hasil operasi Slice/Dice pada cube totSKS

5.3 Evaluasi

Dari hasil uji coba 5.2.1, 5.2.2, 5.2.3 dan 5.2.4 telah menunjukkan aplikasi berjalan dengan lancar. Baik pada uji coba pertama maupun kedua Data Warehouse telah terintegrasi dengan teknologi OLAP. Terbukti dengan berhasilnya operasi-operasi yang diujicobakan.

BAB VI

KESIMPULAN DAN SARAN

BAB VI
KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan yang bisa diambil dari pembuatan Tugas Akhir ini serta saran untuk pengembangan lebih lanjut.

6.1 Kesimpulan

Dari pembuatan Tugas Akhir ini bisa ditarik beberapa kesimpulan:

1. OWB memudahkan dalam mendesain sebuah Data Warehouse dengan pemetaan secara visual.
2. Keterjaminan dari kesalahan kode program PL/SQL karena di-*generate* secara otomatis oleh OWB.

6.2 Saran

Berikut ini adalah saran kemungkinan penggunaan dan pengembangan lebih lanjut dari pengaplikasian OWB dalam Tugas Akhir ini:

1. Penggunaan database dari sumber yang berbeda-beda sehingga tingkat produktifitas OWB semakin terlihat.
2. Penggunaan bahasa *scripting* OMB+ untuk menjalankan operasi secara terjadwal dan otomatis.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

[DJU-05] Djunaidy, Arief. *Pengantar Datawarehousing*. PENS Workshop. 2005

[INM-02] Inmon, W.H. *Building Data Warehouse 3rd Ed.* John Wiley and Sons, Inc. 2002.

[KUN-04] Kunaefi, Anang. *Perancangan dan Pembuatan Perangkat Lunak OLAP Terdistribusi Dengan Memanfaatkan TeknologiLocal-Global Data-Warehouse/OLAP StationBerbasis Oracle9i* . September 2003

[OLA-02] Oracle Corporation. *Oracle 9i OLAP User's Guide*. http://www.oracle.com/.

[ORA-03] Oracle Corporation. *Oracle Database - Data Warehouse Guide*. http://www.oracle.com/. Desember 2003

[OWB-03] Oracle Corporation. *Oracle Warehouse Builder 10g:User's Guide*. http://www.oracle.com/. November 2003

[WIK-06] Wikipedia.org. *Wikipedia Online Ecyclopedia*. http://www.wikipedia.org/