

PENGOLAHAN DATA *WAREHOUSE* TERHADAP SUMBER DATA AKADEMIK DAN KEMAHASISWAAN STIKOM SEBAGAI SISTEM PENDUKUNG KEPUTUSAN

Ditya Octavianto ¹⁾, Tutut Wuriyanto ²⁾

¹⁾ Jurusan Sistem Informasi, STIKOM Surabaya, email: ditya@stikom.edu

²⁾ Jurusan Sistem Informasi, STIKOM Surabaya, email: tutut@stikom.edu

Abstract: This paper introduces the data warehouse and the online analysis process (olap). According to the power management information system, the paper analyzes the power-oriented multidimensional data-base modeling and the presentation of multidimensional views. The system also builds power-oriented data warehouse using oracle database, oracle warehouse builder, oracle discoverer. Then, this paper discusses the concrete applications of data warehouse in completing extract transform and load (ETL) of system Information Academic at STIKOM Surabaya, includes the optimization of the system after creating the MQT tables using oracle warehouse builder and the presentation of the analysis system. System built helps power corporations make decision.

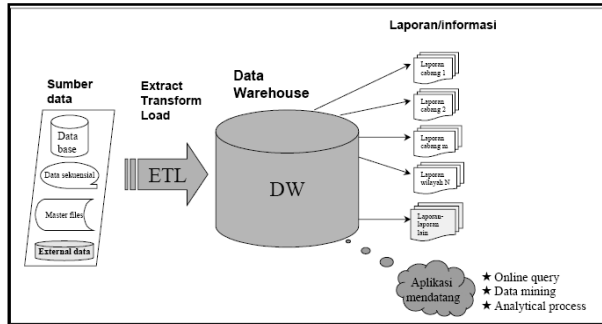
Keywords: Data Warehouse; Power Management; Olap; Multidimensional Views

Data dan Informasi merupakan salah satu faktor yang handal dalam pengambilan keputusan untuk penentuan kebijakan dan perencanaan, menyediakan data dan informasi yang akurat, *up to date* dan tidak terpengaruh oleh pendapat pribadi dengan bentuk laporan yang beragam, meskipun semuanya menggunakan sumber data yang sama. Selain format (*layout*), strukturnya pun juga bervariasi. Ada yang diurutkan berdasar besarnya transaksi, jenis laporan terdahulu, maupun kebutuhan informasi dari bagian lain yang diurutkan dan dikelompokkan per bagian dalam suatu instansi. Sehingga, semakin banyak jumlah data/informasi yang tersedia, semakin sulit untuk mengelolanya, berdampak pula pada pertumbuhan yang pesat dari akumulasi data itu telah menciptakan kondisi yang sering disebut sebagai “*rich of data but poor of information*”, karena data yang terkumpul itu tidak dapat digunakan untuk aplikasi yang berguna, hal tersebut sama halnya dengan otak kanan yang bertugas untuk menangkap dan menyimpan data dalam volume yang besar, sementara otak kiri tidak tahu bagaimana memakainya, kemampuan manajer untuk menganalisis data yang sangat besar tersebut sangat terbatas, sehingga perlu dilakukan pengaturan tertentu sehingga lebih mudah untuk dianalisis dan dipilah datanya menurut ketentuan yang ada.

Berdasarkan latar belakang tersebut, dapat dirumuskan permasalahan yaitu bagaimana merancang suatu *Data Warehouse*, menggunakan terminologi *Multi Dimensional Database* sebagai pelaporan analisa informasi yang terpercaya serta diolah menggunakan *Bussiness Intellegent* untuk mengelola data dan menganalisa guna membantu proses pengambilan keputusan.

Dengan mengacu pada perumusan masalah, maka tujuan yang hendak dicapai dalam penyusunan penelitian ini adalah: (1) bisa menghasilkan informasi yang bermanfaat, baik bagi manajemen operasional maupun eksekutif dengan menyediakan kebutuhan dan penyimpanan data yang bersifat historis, yang terpisah dari aplikasi operasional, sehingga kinerja aplikasi bisa lebih baik dan memungkinkan untuk menghasilkan berbagai bentuk informasi seperti laporan, data analisis, data mining yang terintegrasi dari Data mentah, (2) membantu manajemen dalam menganalisa informasi yang dibutuhkan, dilihat dari berbagai dimensi dan melihat informasi secara detail sehingga membantu dalam membuat keputusan dengan cepat dan tepat untuk menyusun strategi, (3) menyediakan konsep *view* data secara multidimensional (dimensi dalam hal ini adalah dimensi waktu, lokasi, dan lain-lain), termasuk dukungan penuh untuk *view* data yang berhirarki *Data Warehouse* merupakan Pengumpulan dan

mengintegrasikan data-data tersimpan yang sangat besar dari berbagai sumber yang berbeda pada suatu perusahaan besar secara keseluruhan ke dalam satu repositori, mengorganisasikannya sesuai dengan tanggung jawab dan tingkat kemampuan teknis mereka.



Gambar 1 Arsitektur Data Warehouse secara umum

Ada 4 sifat yang mencirikan data yang disimpan di dalam *Data Warehouse* ini didefinisikan oleh salah satu mahaguru data warehousing (Bill Inmon):

1. Berorientasi pada Subyek: Aplikasi untuk operasi perusahaan (*operational system*) berorientasi pada proses (mengotomasi fungsi-fungsi dari proses bersangkutan function oriented). Misalnya, di bank aplikasi kredit mengotomasi fungsi-fungsi: verifikasi lamaran dan credit checking, pemeriksaan kolateral, *approval*, pendanaan, tagihan, dan seterusnya. Di dalam *Data Warehouse* data-data yang dihasilkan dari proses kredit ini, diatur kembali (dikelompokkan) dan diintegrasikan (digabung) dengan data-data dari fungsi-fungsi lain, agar berorientasi pada misalnya nasabah dan produk.
2. Terintegrasi: Data dari macam-macam aplikasi transaksi (untuk bank misalnya: tabungan, kredit, rekening koran) semua mengandung data nasabah, ada yang sama ada yang spesifik (yang sama misalnya: nama dan alamat, yang spesifik misalnya: untuk kredit ada kolateral, untuk rekening Koran ada *overdraft*) – didalam *Data Warehouse* data-data yang sama harus diintegrasikan disatu *database*, termasuk misalnya diseragamkan formatnya (sederhana tetapi paling sering terjadi – aplikasi-aplikasi sering dibeli *vendor* berbeda, dibuat dengan/dijalankan di teknologi berbeda-beda)
3. Variasi Waktu: *Data Warehouse* menyimpan sejarah (*historical data*). Bandingkan dengan kebutuhan sistem operasional yang hampir semuanya adalah data mutakhir, waktu merupakan tipe atau bagian data yang sangat penting didalam *Data Warehouse*. Di dalam *Data Warehouse* sering disimpan macam-macam

waktu, seperti waktu suatu transaksi terjadi, diubah, dibatalkan, kapan efektifnya, kapan masuk ke komputer, kapan masuk ke *Data Warehouse*; juga hampir selalu disimpan versi, misalnya terjadi perubahan definisi kode pos, maka yang lama dan yang baru ada semua di dalam *Data Warehouse* kita. Sekali lagi, *Data Warehouse* yang bagus adalah yang menyimpan sejarah

4. *Non-volatile*: Sekali masuk ke dalam *Data Warehouse*, data-data, terutama data tipe transaksi, tidak akan pernah di *update* atau dihapus (*delete*).

Terlihat, bahwa keempat karakteristik ini saling terkait-kesemuanya harus diimplementasikan agar suatu *Data Warehouse* bisa efektif memiliki data untuk mendukung pengambilan-keputusan. Dan, implementasi keempat karakteristik ini membutuhkan struktur data dari *Data Warehouse* yang berbeda dengan *database* sistem operasional.

Suatu *Data Warehouse* bisa memiliki banyak sumber data. Data dari sumber-sumber tersebut diekstrak, ditransformasi, kemudian dimasukkan ke dalam *Data Warehouse*. Proses-proses inilah yang disebut sebagai *data acquisition process* atau disebut juga sebagai *load*. Proses ini memiliki tiga komponen yaitu [3]: (a) *Extractor*, berfungsi mengekstrak data dari sistem operasional, (b) *Transform*, melakukan transformasi data dari suatu format operasional sistem menjadi format *Data Warehouse*, (c) *Insert/load*, memasukkan atau memperbaharui *Data Warehouse* dengan data yang diterimanya, (d) ketiga komponen diatas disebut sebagai Proses *Extract, Transform, Load (ETL)*.

Data Warehouse memiliki beberapa keunggulan, jika dibandingkan dengan *system* operasional biasa antara lain: (a) *Data Warehouse* dapat memproses informasi dengan kredibilitas tinggi dan efisien, (b) *Data Warehouse* meningkatkan produktifitas pembuat keputusan perusahaan melalui konsolidasi, konversi, transformasi dan integrasi data operasional, serta menyediakan representasi informasi yang konsisten bagi organisasi.

Desain *database* sistem operasional tidak memadai untuk digunakan pada aplikasi pendukung keputusan. Desain *database* untuk *system* operasional biasanya dirancang untuk OLTP, mengotomasi operasi bisnis dan menekankan efisiensi dalam memperbaharui ratusan data individual setiap saat dalam hitungan detik. Sedangkan, desain *database* untuk *data warehouse* harus memperhatikan kemampuan sistem dalam mengolah jutaan *record* data yang biasanya memiliki ukuran ratusan *gigabytes* dalam sekali proses.

Salah satu komponen dalam *data warehouse* yang sangat penting adalah metadata. Metadata merupakan penjelasan makna secara terperinci mengenai suatu data dalam *Data Warehouse*. Metadata memetakan data saat data ditransformasikan dari lingkungan operasional ke lingkungan *Data Warehouse*.

Metadata merupakan arahan untuk membantu alokasi isi data warehouse, Metadata berisi antara lain: (1) struktur data, (2) algoritma pengolahan

Pemetaan data dari lingkungan operasional ke data warehouse.

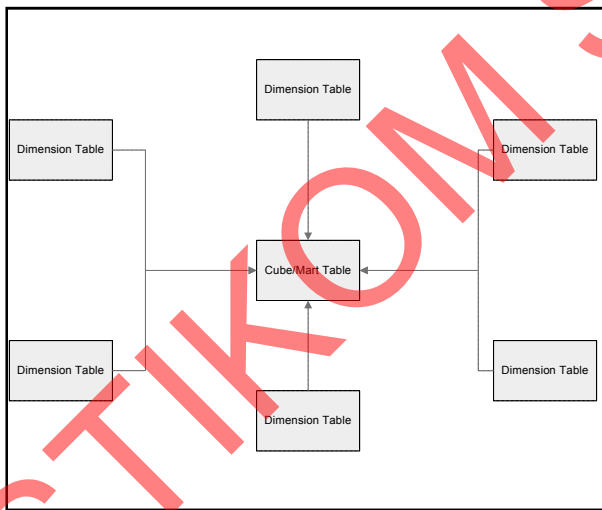
Pemodelan Data Warehouse

Multidimensional Model

Pada sistem pengembangan Data Warehouse ini, dikenal istilah skema-skema disini merupakan kumpulan dari objek-objek database, meliputi table, view, indeks, dan synonym. Ada banyak cara untuk mengatur objek skema untuk mendesain Data Warehouse. Pada sistem pengembangan Data Warehouse di sini, menggunakan model multidimensional. Skema dalam multidimensional dikenal ada 2 skema, yakni Star Schema dan Snowflake Schema.

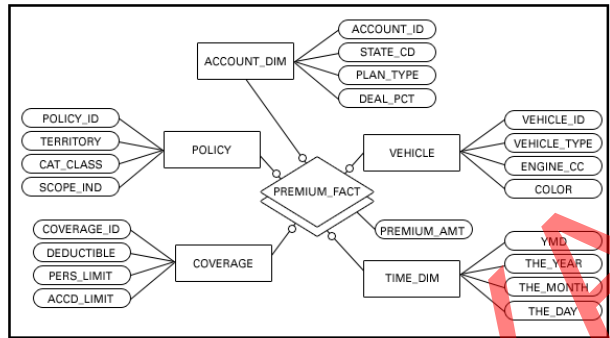
Star Schema

Star Schema merupakan skema Data Warehouse yang paling simple. Disebut Star Schema karena bentuknya yang mirip bintang, dengan titik sudut yang terfokus di tengah, sudut tengah dari bintang tersebut adalah tabel fact yang telah di aggregated atau lebih sering disebut table mart, dan sudut-sudut yang lain adalah tabel parameter atau lebih sering dikenal dengan sebutan tabel dimension, yang bisa dilihat pada Gambar 2.



Gambar 2 Struktur Star Schema

Tidak seperti struktur database yang lain, pada star schema, tabel-tabel dimensi ini mengalami proses denormalisasi. Adapun Star Schema yang di design untuk pengembangan Data Warehouse ini dapat dilihat pada Gambar 3.



Gambar 3 Contoh Model STAR Schema

Online Analytical Processing (OLAP)

OLAP adalah suatu pernyataan yang bertolak belakang atau kontras dengan OLTP. Istilah OLAP muncul pertama kali pada tahun 1993 yang diperkenalkan oleh Edgar F. Codd, S. B. Codd, dan C. T. Salley dalam dokumen untuk Arbor berjudul "Providing OLAP to User-Analysts: An IT Mandate". OLAP menggambarkan sebuah klas teknologi yang dirancang untuk analisa dan akses data secara khusus. Apabila pada proses transaksi pada umumnya semata-mata adalah pada relational database, OLAP muncul dengan sebuah cara pandang multidimensi data. Cara pandang multimensi ini didukung oleh teknologi multidimensi database. Cara ini memberikan tehnik dasar untuk kalkulasi dan analisa oleh sebuah aplikasi bisnis.

OLTP mempunyai karakteristik beberapa user dapat creating, updating, retrieving untuk setiap record data, lagi pula OLTP sangat optimal untuk updating data. OLAP aplikasi digunakan untuk analisa dan mengatur frekuensi level dari agregat/jumlah data. OLAP database biasanya di update pada kumpulan data, jarang sekali dari multiple source dan menempatkan kekuatan analisa pada pada back-end pada operasi aplikasi. Sebab itulah maka OLAP sangat optimal digunakan untuk analisis.

Relational database merupakan suatu bentuk yang baik untuk mendapatkan suatu record dalam kapasitas jumlah record yang kecil, namun tidak cukup baik dalam mendapatkan suatu record dalam kapasitas jumlah record yang sangat besar serta membuat suatu summaries data untuk di analisa, ini memerlukan response time yang lambat dan membutuhkan cukup waktu.

Aplikasi menggunakan OLTP cenderung atomized untuk "record-at-a-time"? data. Dengan OLAP aplikasi lebih cenderung pada summarized data. Sedangkan, OLTP aplikasi lebih cenderung tidak mempunyai historical data, hampir setiap aplikasi OLAP dikaitkan dengan kebutuhan historical data. Jadi, OLAP database membutuhkan kemampuan untuk menangani "time series data"? Aplikasi dan database menggunakan OLTP lebih cenderung pada proses pengelompokan data (data entry). Sedangkan, OLAP lebih cenderung pada "subject oriented"?

Table 2 Perbedaan OLTP dan OLAP

<i>OLTP (Relational)</i>	<i>OLAP (Multidimensional)</i>
<i>Automized</i>	<i>Summarized</i>
<i>Present</i>	<i>Historical</i>
<i>Record at a the time</i>	<i>Many record at a time</i>
<i>Process Oriented</i>	<i>Subject Oriented</i>

Informasi yang dihasilkan oleh *Data Warehouse* bersifat multidimensi. Aplikasi aplikasi *RDBMS* biasa yang banyak beredar, tidak mampu menampilkan sudut pandang multidimensi tersebut, yang diperlukan untuk membuat analisa yang kompleks *Tool OLAP* yang memiliki kemampuan menampilkan data dalam format multidimensi dan memiliki fungsi fungsi analisa yang lengkap muncul sebagai solusi untuk mendukung pengambilan keputusan bisnis suatu perusahaan. *OLAP* merupakan sebuah teknologi yang didesain untuk mengolah pengaksesan dan analisa data data *ad hoc* yang berformat multidimensi.

Kebanyakan pendekatan pada *OLAP* berpusat pada gagasan memformulasi ulang data relasional atau *flat file* menjadi penyimpanan data multidimensi yang dioptimalkan untuk analisa data. Penyimpanan data multidimensi atau dikenal juga sebagai *cube*, menyimpan data beserta dengan dimensi-dimensinya, memungkinkan pengguna menganalisa data mereka bersamaan dengan sumbu-sumbu bisnis mereka. Dua bentuk analisa yang paling relevan dengan para pengguna kebanyakan adalah *slice and dice* dan *drill down*.

OLAP memungkinkan manajer dan penganalisis untuk mengamati dan menggunakan sejumlah besar data yang terkumpul dan detail dari *Data Warehouse* dari berbagai sudut pandang. Dengan menggunakan *OLAP*, analisis dari multidimensional *database* dapat mengungkapkan pola, tren, dan kondisi tertentu. *OLAP* dapat dilakukan secara *online* pada waktu nyata dengan tanggapan yang cepat terhadap *query* yang diajukan, sehingga sangat mendukung proses pengambilan keputusan. *OLAP* memiliki fungsi – fungsi sebagai berikut: (a) *Roll-up (consolidation)*, untuk mendapatkan rangkuman informasi dengan tingkatan dimensi yang lebih luas/umum, (b) *Drill-down* agar user memperoleh gambaran yang lebih detail dari dimensi terkait, (c) *Slice and Dice* untuk dapat melihat *database* dalam sudut pandang yang berbeda. Dengan *slice* dari *database* penjualan, dapat memperlihatkan seluruh penjualan dari berbagai produk berdasarkan wilayah dan *dice* memperlihatkan secara lebih spesifik yaitu *sub-cube* dari *cube* secara keseluruhan, (d) *Pivot* yang memberikan visualisasi nilai *cube* dengan cara yang lebih alami dan secara intuisi.

Karena data *cube* menyarankan suatu penafsiran ruang data dalam multi-dimensional, sejumlah penjual *OLAP* memilih untuk secara fisik memakai model dadu/kubus dalam pengaturan multi-dimensional. Produk *MOLAP* menawarkan waktu respon yang cepat atas *query OLAP* dengan mengindeks secara langsung ke dalam struktur data *cube* untuk mendapatkan kembali subsets data dikumpulkan. *MOLAP* tidak selalu efektif, dikarenakan tingkat dimensi data yang tinggi menjadikan banyak data dalam data *cube* menjadi terus meningkat jarang. Dengan kata lain, banyak dari kombinasi atribut yang diwakili oleh struktur data kubus (*Cube*) tidak berisi data apapun. Dengan demikian, suatu *MOLAP* yang susunannya dapat berisi suatu jumlah sel kosong yang sangat besar, dan tidak dapat tertampung dalam *storage*. Tetapi teknik Tekanan dan internal *indexing* dapat dipekerjakan untuk mengurangi masalah ini.

OLAP memungkinkan pengguna untuk *slice and dice* informasi terkonsolidasi untuk melihat data dari berbagai sudut pandang yang berbeda. Pengguna dapat memotong atau memutar sebagian data tertentu berdasarkan dimensi apapun yang diinginkan.

OLAP juga memungkinkan pengguna untuk *drill* atau bernavigasi sepanjang informasi untuk mendapat informasi yang lebih detail. Kebanyakan *tool OLAP* mendukung *mandrill down* untuk melihat informasi pada level yang lebih detail, tetapi *tool OLAP* yang paling lengkap memiliki kemampuan untuk *drill* informasi kemanapun, tidak hanya *drill down*. *Tool* ini mampu *mandrill up* untuk melihat informasi pada level lebih umum, atau *mandrill across* untuk bergerak kesamping dari satu data set ke data set lain dalam level yang sama, sehingga saat ini kita dapat mendaftarkan butir-butir berikut sebagai dasar ilmiah untuk *OLAP*

- Tipe: *TABLE*, *REPORT*, dan *CUBE*
- Operasi: *GROUPING*, *ROLLUP*, *CUBE*

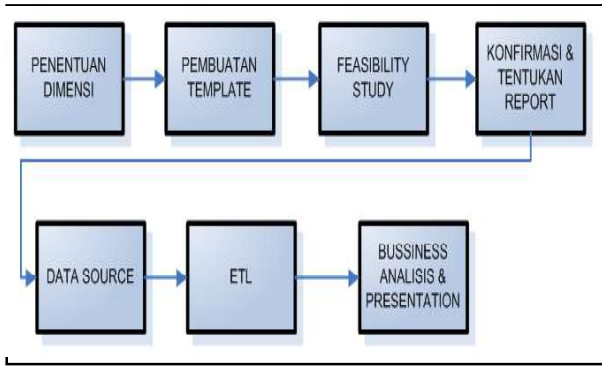
Nilai: Pemakaian nilai *NULL* bagi nilai-nilai ‘sel’ dikubus (*cube*) dan *report* untuk data agregat

Pemanfaatan data *Warehouse* untuk masa depan nantinya bisa dikembangkan sampai bisa menentukan Keputusan, dengan mengambil salah satu *sample* data dari *measure* suatu kubus kita bisa menggabungkan dengan kubus yang berbeda bisa menarik korelasinya, dan akhirnya dikembangkan sampai dengan data mining.

Penggunaan Komponen yang sesuai dengan keadaan sumber datanya saat membuat proses *Mapping* menggunakan *Warehouse Builder*, akan mempengaruhi tingkat keakuratan dan kecepatan saat pemindahan sumber data ke *Data Warehouse*.

Procedure Pengembangan

Procedure pengembangan dalam analisa data *warehouse* berupa *Analytical Intellegensi Report* pada Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Surabaya (STIKOM). Terlihat pada Gambar



Gambar 4 Bagan Tahapan Pengembangan

Bagan tahapan Pengembangan berikut penjelasan tahapan-tahapan yang dilakukan selama proses pengembangan Gambar 4.

HASIL DAN PEMBAHASAN

Proses pengujian dan implementasi dari aplikasi Analisa Data Warehouse dengan OLAP saat melakukan proses metadata baik dari Internal Database OLTP maupun External Database terdiri dari proses pengambilan data, proses segmentasi data, ekstraksi, tahap pelatihan dan tahap pengujian.

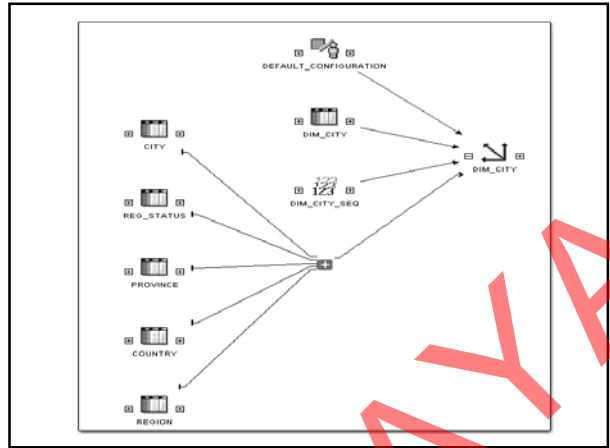
Proses Pengambilan Data

Training data set dan sekumpulan tabel-tabel database untuk pengujian diperoleh dari hasil transaksi. Proses pengambilan data, ini melibatkan tabel master yang ada di OLTP dijadikan dimensi menurut level sebagai objek dengan setiap relasinya yang ada, saling berhubungan. Pada proses pengambilan data ini metadata haruslah disajikan dengan benar sesuai dengan studi kasus yang diinginkan, sama halnya pada saat kita query SQL di Oracle untuk kasus tertentu, mempertimbangkan konsistensi data dan keakurasian, bedanya semua ini (perintah SQL) di jalankan pada saat yang sama dengan banyak perintah, jika dilakukan dengan SQL biasa, maka proses tersebut memakan banyak waktu.

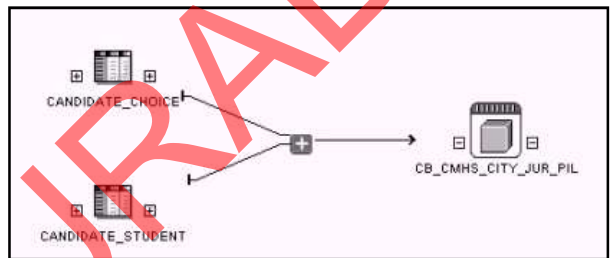
Proses Segmentasi

Proses Segmentasi adalah proses pemisahan data yang akan diambil disesuaikan dengan status yang ada. Tujuan utama dari proses ini adalah pada saat pengambilan data akan dipilah dulu melalui proses ini dan akan dimasukkan dalam dimensi melalui proses metadata.

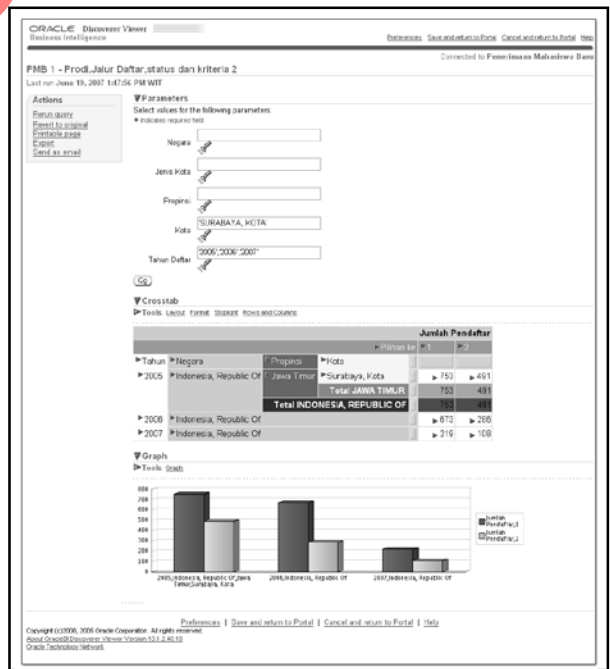
Setelah itu, pada tahap implementasi aplikasi OLAP yang berbasis Web ini akan dijelaskan proses-proses yang terjadi dimulai dari login, lalu setelah itu menganalisa bermacam laporan yang sudah tersedia berdasarkan data-data yang ada di Warehouse.



Gambar 5 Integrasi Tabel yang ada di OLTP dengan Dimensi DIM_CITY



Gambar 6 Integrasi Tabel yang ada di OLTP dengan Kubus CB_CMHS_CITY_JUR_PIL



Gambar 7 Tabel Prodi dan Jalur Daftar yang Siap di Migrasikan ke Excell

The screenshot shows a Microsoft Excel spreadsheet with the following data:

Jenis Kelamin		Pilihan tes		LAKI-LAKI				PEREMPUAN					
1	2	3	4	5	6	7	8	9	10	11	12	13	14
2000	12	9	1	59	12	644	957	203	248	241	149	103	120
2001	103	96	4	142	80	787	567			311	217		
2004	150	118	1	153	123	654	479			172	23		
2005	195	152	1	125	86	405	204			91	88		
2006	120	95	1	111	71	319	156			65	20		
2007	147	86		79	47								

Gambar 7 Hasil Migrasi ke Excell untuk laporan Prodi dan Jalur Daftar

SIMPULAN

Dari hasil analisis terhadap pengujian yang dilakukan terhadap proses *Extract Load Transform* baik oleh Dimensi dan Kubus dapat diambil beberapa kesimpulan berikut: (1) dengan membuat proses *Mapping / Metadata* secara benar dan sesuai dengan yang diharapkan, tentunya butuh ketelitian dan harus disesuaikan dengan keadaan Sumber Data, karena mungkin relasinya tidak sesuai dengan pada saat kita mau mengimplementasikan ke dalam bentuk tabel dimensi, (2) pembuatan Kubus yang baik adalah harus bisa mengatasi semua kebutuhan *summary* terhadap ukuran data di tingkat manajemen, karena efisiensi data dan waktu sangatlah diperhatikan dalam membantu membuat keputusan, (3) keberhasilan proses Penjadwalan saat menjalankan *metadata* dipengaruhi oleh

pembuatan *mapping* dan kondisi data, (4) aspek pembuatan *metadata* dengan *Warehouse Builder* lebih efisien karena kita hanya memodelkan alur *Flow Diagram* yang sesuai dengan studi kasus tertentu untuk menentukan ukuran (*measurement*), sudah secara otomatis terbentuk data dengan dimensi-dimensi, memang alternatifnya kita bisa menggunakan fasilitas *PL/SQL* yang ada di *Oracle*, tetapi memerlukan banyak waktu untuk membuat proses *metadata*

RUJUKAN

- Garmany, J.JR. 2004. *Oracle Application Server 10g Administration Handbook*. Build and Maintain a Robust E-Business Infrastructure. Helping you save time and money anytime you move data, http://oracle.com/technology/products/warehouse/pdf/Architecture%20White_Paper.pdf
- Mcgraw-Hill. 1997. *Data Warehousing, Data Mining & OLAP*. Boston.
- Mattison, R. 1991. *Data Warehousing Strategies Technologies and Techniques*. United States of America of America: McGraw-Hill Companies Inc.
- Oracle Warehouse Builder 10g
- Oracle University team. 2002. *Learn Oracle from Oracle, Data Warehousing Fundamentals*
- Oracle University team. 2002. *Learn Oracle from Oracle, Data Warehousing Implementation*. Volume 1.
- Singh dan Hary, S. 1998. *Data Warehousing: Concepts, Technologies, Implementations, and Management*, New Jersey: Prentice-Hall.