

**MODUL DATA MINING
DATA WAREHOUSE
PERTEMUAN 2 (ONLINE)**



Disusun Oleh
Syefira Salsabila

Data mining muncul setelah banyak dari pemilik data baik perorangan maupun organisasi mengalami penumpukan data yang telah terkumpul selama beberapa tahun, misalnya data pembelian, data penjualan, data nasabah, data transaksi, email dan sebagainya. Kemudian muncul pertanyaan dari pemilik data tersebut, apa yang harus dilakukan terhadap tumpukan data tersebut. Misalnya, perangkat lunak *data mining* bisa membantu perusahaan ritel untuk menemukan pelanggan yang memiliki ketertarikan tertentu. Istilah ini umumnya dipersempit artinya yaitu hanya untuk menggambarkan perangkat lunak yang merepresentasikan data dengan cara-cara yang baru. Namun sebenarnya perangkat lunak data mining tidak hanya berfungsi mengubah presentasi tersebut, melainkan juga menemukan relasi tak dikenal antar-data.

Data warehouse adalah kumpulan data yang dirancang untuk mendukung pengambilan keputusan manajemen. Data warehouse berisi data yang berbeda-beda yang memberikan gambaran kondisi bisnis pada saat tertentu. Data *warehouse* juga merupakan tempat penyimpanan saat ini dan data historis dari potensi ketertarikan manager seluruh perusahaan. Data yang disimpan itu saling berelasi dan dirancang berdasarkan query dan analisis daripada proses transaksinya. Data tersebut kemudian ditransformasikan menjadi sebuah informasi yang dapat diakses kapan saja dan selalu up-to-date. Informasi ini kemudian akan digunakan untuk dianalisis untuk menghasilkan keputusan yang tepat.

Beberapa manfaat *data warehouse* adalah *data warehouse* menyediakan model data umum untuk semua data yang menarik terlepas dari sumber data itu sehingga lebih mudah untuk melaporkan dan menganalisa informasi, informasi dalam *data warehouse* berada di bawah kontrol pengguna *data warehouse* sehingga jika sistem sumber data dibersihkan informasi dalam *warehouse* dapat disimpan dengan aman untuk waktu yang lama, *data warehouse* menyediakan pengambilan data tanpa memperlambat sistem operasional, dan *data warehouse* memfasilitasi aplikasi *decision support system*.

Datawarehouse mendukung kemampuan melakukan query untuk mendukung pengambilan keputusan, data warehouse lebih mudah dipahami oleh customer dilihat kebutuhan informasi mereka dalam menaksir keadaan finansial, produk dan layanan, serta waktu untuk memasarkan produk dan layanan baru. Yang terpenting – mungkin juga harapan yang beresiko – data warehousing dapat dilihat sebagai teknologi, dengan kemampuan unik untuk mengolah informasi, bertujuan untuk menghasilkan pendapatan yang tinggi dan memperbesar keuntungan.

Tujuan utama dari pembuatan data warehouse adalah untuk menyatukan data yang beragam ke dalam sebuah tempat penyimpanan dimana pengguna dapat dengan mudah menjalankan query (pencarian data), menghasilkan laporan, dan melakukan analisis. Salah satu keuntungan yang diperoleh dari keberadaan data warehouse adalah dapat meningkatkan efektifitas pembuatan keputusan. Selain itu secara khusus tujuan dari data warehouse meliputi;

- a. Memberikan kemudahan untuk mengakses informasi yang ada.
Kemudahan disini berbicara tentang efisiensi. Data *warehouse* harus efisien sehingga dengan mudah dipahami oleh *user* bukan hanya oleh *devoleper* saja. Selain itu, pengguna juga dapat mengkombinasikan data dalam data *warehouse* dengan berbagai cara (*slicing and dicing*). Untuk mengakses data *warehouse* disarankan sebaiknya dapat dilakukan dengan sederhana dan mudah dioperasikan.
- b. Menyediakan informasi yang konsisten.
Data *warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan *user* untuk pengambil keputusan. Oleh karena itu, kredibilitas data yang terdapat dalam data *warehouse* harus dapat dipertanggungjawabkan.
- c. Mampu beradaptasi dan tahan terhadap perubahan
Perubahan-perubahan yang terjadi harus dapat diatasi oleh data *warehouse*. Dengan kata lain, data *warehouse* harus dirancang agar mampu menghadapi setiap perubahan dengan terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.
- d. Mampu mengamankan informasi.
Informasi yang tersimpan dalam data *warehouse* harus tersimpan dengan aman. Dengan kata lain, informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Oleh karena itu, data *warehouse* harus mampu mengendalikan setiap akses dari informasi yang ada.
- e. Mampu memberikan dukungan dalam pengambilan keputusan.
Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan data *warehouse*. Data *warehouse* bisa digambarkan sebagai kumpulan teknologi pendukung keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan informasi, dapat membuat keputusan dengan cepat dan tepat.
- f. User friendly.
Seperti pada tujuan data *warehouse* pertama, data *warehouse* harus dirancang agar dapat dioperasikan dengan mudah oleh *user*. Tidak seperti sistem operasional dimana seringkali *user* tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan tetapi *user* data *warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan *user* data *warehouse* merupakan faktor yang sangat penting.

Pengembangan data *warehouse* meliputi pengembangan sistem untuk mengekstrak data dari sistem dan instalasi sistem basisdata *warehouse* yang memungkinkan manajer dapat mengakses data secara fleksibel. Secara umum istilah data *warehouse* mengacu kepada basis data yang berbeda-beda yang dikumpulkan dari seluruh bagian perusahaan. Data yang disimpan di dalam data *warehouse* ini memiliki empat karakteristik, yaitu:

- a. Subject Oriented

Data *warehouse* diorganisasikan pada subjek-subjek utama, seperti pelanggan, barang/produk, dan penjualan. Berfokus pada model dan analisis pada data untuk

membuat keputusan, jadi bukan pada setiap proses transaksi atau bukan pada OLTP. Menghindari data yang tidak berguna dalam mengambil suatu keputusan.

Data warehouse dirancang untuk membantu user dalam pengambilan keputusan. Contohnya untuk mengetahui tentang data penjualan perusahaan, kita bisa membangun data *warehouse* yang berfokus pada penjualan. Dengan menggunakan data *warehouse*, kita bisa menjawab pertanyaan seperti “Siapa pembeli terbaik untuk barang ini tahun lalu?” Kemampuan untuk mendefinisikan sebuah data *warehouse* sebagai sebuah subjek, dalam hal ini penjualan, membuat data *warehouse subject oriented*.

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. Didalam 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.

b. Integrated

Dibangun dengan menggabungkan/menyatukan data yang berbeda. relational database, flat file, dan on-line transaction record. Menjamin konsistensi dalam penamaan, struktur pengkodean, dan struktur atribut diantara data satu sama lain.

Data *warehouse* dikonstruksikan dengan cara mengintegrasikan sejumlah sumber data yang berbeda. Data yang terintegrasi menyebabkan data tersebut lebih konsisten, sehingga lebih mudah dipahami oleh para pembuat keputusan.

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).

c. Time Variant

Data disimpan untuk menyediakan informasi dari perspektif historical, data yang tahun lalu/ 4-5 thn. Waktu adalah elemen kunci dari suatu data warehouse/ pada saat peng-*capture*-an.

Data *warehouse* harus bisa menghasilkan informasi dari sudut pandang historical (misalnya informasi 5-10 tahun yang lalu atau bahkan lebih). Atau bisa dikatakan bahwa data *warehouse* berfokus pada perubahan setiap waktunya.

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 di dalam data warehouse. Didalam data warehouse sering disimpan macam-macam waktu, seperti waktu suatu transaksi terjadi/dirubah/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 didalam data warehouse kita. Sekali lagi, data warehouse yang bagus adalah yang menyimpan sejarah.

d. Non Volatile

Setiap kali proses perubahan, data akan di tampung dalam tiap-tiap waktu. Jadi tidak di perbaharui terus menerus. Data warehouse tidak memerlukan pemrosesan transaksi dan recovery. Hanya ada dua operasi initial loading of data dan access of data.

Sekali masuk kedalam 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.

Data yang ada dalam data *warehouse* tidak bisa di-*edit* ataupun di-*update*. Data *warehouse* dibuat untuk melayani *user* (*analyst* dan pengambil keputusan). Sehingga data *warehouse* wajib dirancang sesuai dengan persyaratan berikut:

- a. Harus bisa memberikan kepuasan kepada setiap *user*.
- b. Memiliki *function* sendiri tanpa mengganggu OLTP *systems*.
- c. Menyediakan pusat tempat penyimpanan data yang konsisten.
- d. Menjawab setiap *complex queries* dengan cepat.
- e. Menyediakan berbagai analisis *tools* yang kuat, seperti OLAP dan data *mining*.

Arsitektur Data Warehouse.

Arsitektur *data warehouse* terbagi dalam tiga bagian yaitu:

- a. *Data warehouse*, yang terdiri dari data-data dan software yang berasosiasi.
- b. *Data acquisition (back-end)* software, yang mengekstrak data dari sistem legal dan sumber-sumber eksternal, mengkonsolidasi dan merangkumnya, dan memprosesnya ke dalam *data warehouse*.

- c. *Client (front-end)* software, yang mengizinkan user mengakses dan menganalisis data dari warehouse.

Data warehouse bukan hanya suatu arsip atau fasilitas penyimpanan data. Pengembangan *data warehouse* ditujukan untuk mengintegrasikan, menggali dan membuat intisari informasi yang penting dari data yang tersebar. *Data warehouse* dikembangkan untuk mempertemukan suatu permintaan bagi Manajemen Informasi dan Analisis yang tidak bisa dijumpai pada basis data.

Aplikasi basis data tidak mampu mempertemukan kebutuhan ini untuk beberapa pertimbangan: pengolahan pelaporan data yang banyak/besar dapat mengurangi waktu tanggap aplikasi basis data; perancangan aplikasi basis data tidaklah dioptimalkan untuk analisis informasi dan pelaporan; kebanyakan organisasi mempunyai lebih dari satu aplikasi basis data, sehingga pelaporan secara keseluruhan organisasi tidak bisa didukung dari sistem tunggal; pengembangan laporan pada aplikasi basis data sering memerlukan program komputer spesifik yang mahal.

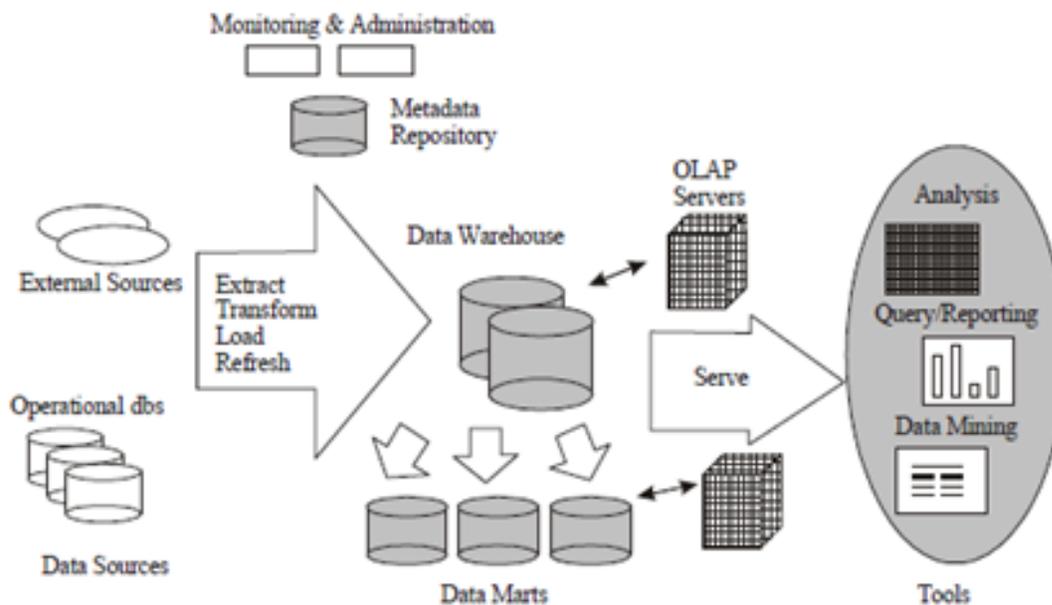
Aktivitas lebih lanjut kemudian adalah analisis informasi dan data yang ada di *data warehouse*, yang memungkinkan untuk menampilkan dan menganalisis informasi dari berbagai sudut pandang (dimensi), teknik-teknik *data mining* kemudian digunakan untuk menggali data dan mendapatkan pemahaman yang tersembunyi dari data serta informasi yang ada untuk membantu dalam kegiatan pengambilan keputusan, dalam hal ini membantu siswa dalam menentukan program peminatan studi yang akan diambilnya.

Membuat *data warehouse* tidak sekedar memindahkan data operasional ke dalam *data warehouse*, memindahkan data dari beberapa sumber berpotensi menimbulkan masalah besar terutama terkait dengan isu tidak konsistennya data. *Data warehouse* tidak hanya digunakan dalam melakukan *loading*, integrasi, dan menyimpan data dalam jumlah besar, namun juga berpotensi untuk mendapatkan sudut pandang baru dari sebaran data, dan memungkinkan untuk memberikan pelaporan dan jawaban dari pertanyaan pengguna yang bersifat *ad hoc* secara cepat dan lebih baik.

Online Analytical Processing (OLAP) merupakan salah satu tools yang digunakan untuk mengakses informasi dalam *data warehouse*. Teknologi OLAP memungkinkan *data warehouse* digunakan secara efektif untuk proses *online analysis*, memberikan respon yang cepat terhadap *analytical queries* yang kompleks. *Multidimensional data model* dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar, sehingga dapat dievaluasi secara cepat dengan menggunakan *online analysis* dan *graphical tool*. Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk melakukan *support* analisis secara *real time*.

Tahapan untuk membangun data warehouse terdiri atas:

Arsitektur *data warehouse* (lihat gambar di bawah) mencakup proses ETL (*Extraction, Transformation, Loading*) untuk memindahkan data dari *operational data source* dan sumber data eksternal lainnya ke dalam *data warehouse*.



Data warehouse dapat dibagi menjadi beberapa *data mart*, berdasarkan fungsi bisnisnya (contoh *data mart* untuk penjualan, pemasaran, dan keuangan). Data dalam *data warehouse* dan *data mart* diatur oleh satu atau lebih server yang mewakili *multidimensional view* dari data terhadap berbagai *front end tool*, seperti *query tools*, *analysis tools*, *report writers*, dan *data mining tools*.

a. Identifikasi sumber data

Langkah pertama sebelum mulai mengembangkan *data warehouse* yaitu identifikasi sumber data. Perlu mencari tahu apa data yang dibutuhkan untuk diletakkan ke dalam *data warehouse*.

b. Membangun ETL (Extraction, Transformation, Loading) tool

Setiap *data warehouse* memiliki data yang berbeda persyaratan, oleh karena itu, *ETL tool* yang sesuai adalah solusi yang lebih baik untuk memenuhi persyaratan.

c. Ekstraksi

Perlu menentukan sistem *database* yang akan digunakan dan juga mencari tahu data apa yang diperlukan sebelum mendapatkannya. Penurunan biaya perangkat keras dan penyimpanan telah mengatasi masalah pada menghindari duplikasi data dan juga kekhawatiran pada kurangnya ruang penyimpanan sebagai tempat

penyimpanan data yang berlebihan atau yang tidak diperlukan. Namun, mungkin tidak ada alasan untuk menyimpan data yang tidak diperlukan dan telah diidentifikasi tidak berguna dalam proses pengambilan keputusan. Oleh karena itu, diperlukan untuk mendapatkan hanya ekstrak data yang relevan sebelum membawa ke *data warehouse*.

Ekstraksi data adalah proses dimana data diambil atau diekstrak dari berbagai sistem operasional, baik menggunakan *query*, atau aplikasi ETL. Terdapat beberapa fungsi ekstraksi data, yaitu:

1. Ekstraksi data secara otomatis dari aplikasi sumber.
2. Penyaringan atau seleksi data hasil ekstraksi.
3. Pengiriman data dari berbagai *platform* aplikasi ke sumber data.
4. Perubahan format *layout* data dari format aslinya.
5. Penyimpanan dalam *file* sementara untuk penggabungan dengan hasil ekstraksi dari sumber lain.

d. Transformasi

Setelah penggalian data dari berbagai sumber, transformasi dibutuhkan untuk menjamin konsistensi data. Agar mengubah data ke dalam *data warehouse* dengan benar, perlu mengetahui cara pemetaan bidang sumber data eksternal ke *data warehouse*. Transformasi dapat dilakukan selama ekstraksi data atau saat memuat data ke dalam *data warehouse*. Integrasi ini bisa menjadi masalah yang kompleks ketika jumlah sumber data menjadi lebih besar.

Transformasi adalah proses dimana data mentah (*raw data*) hasil ekstraksi disaring dan diubah sesuai dengan kaidah bisnis yang berlaku. Langkah-langkah dalam transformasi data adalah sebagai berikut:

1. Memetakan data input dari skema data aslinya ke skema *data warehouse*.
2. Melakukan konversi tipe data atau format data.
3. Pembersihan serta pembuangan duplikasi dan kesalahan data.
4. Penghitungan nilai-nilai derivat atau mula-mula.
5. Penghitungan nilai-nilai agregat atau rangkuman.
6. Pemeriksaan integritas referensi data.
7. Pengisian nilai-nilai kosong dengan nilai *default*.
8. Penggabungan data.

e. Loading

Setelah proses penggalian, mengubah dan pembersihan telah dilakukan, data diambil ke dalam *data warehouse*. Pemuatan data dapat dikategorikan ke dalam dua jenis; pemuatan data yang saat ini ada dalam *database* operasional dan pemuatan pembaruan *data warehouse* dari perubahan yang telah terjadi dalam *database* operasional. Untuk menjamin kesegaran data, *data warehouse* perlu untuk terus memperbaharui datanya. Banyak persoalan yang perlu untuk dipertimbangkan terutama saat memuat data yang sudah diperbaharui ke *data*

warehouse. Sementara melakukan pemutakhiran *data warehouse*, perlu memastikan bahwa tidak ada data yang dilepaskan dan juga untuk memastikan *overhead* minimum selama proses *scanning* file yang ada.

Online Transaction Processing atau yang sering disebut dengan OLTP adalah system yang berorientasi proses yang memproses suatu transaksi secara langsung melalui komputer yang terhubung dalam jaringan. Seperti misalnya kasir pada sebuah super market yang menggunakan mesin dalam proses transaksinya.

Data *warehouse* bisa dikatakan sebagai suatu salinan dari OLTP (*On-Line Transaction Processing*) yang terstruktur yang digunakan untuk memenuhi kebutuhan analisis, *reporting*, maupun data *mining*. OLTP sendiri adalah sebuah proses yang menitikberatkan pada transaksinya, seperti input data dan lain-lain. Pada Tabel 1 akan memperlihatkan perbedaan OLTP *systems* dan data *warehouse*.

1.2. Operational Systems vs Data Warehousing Systems

Operational	Data Warehouse
Data saat ini	Menangani data historis
Data bersifat dinamis	Data bersifat statik
Akses Read/Write	Akses Read only
Untuk keperluan Transaksi sehari hari	Untuk keperluan analisa
Berorientasi pada Aplikasi	Berorientasi pada Subyek
Used by clerical staff for day-to-day operations	Digunakan oleh top manager untuk proses analisa
Normalisasi data (ER model)	Denormalisasi data (model dimensional)
Harus dioptimasi untuk query dalam bentuk yang singkat	Query melibatkan data dalam jumlah besar (query lebih kompleks).

Online Analytical Processing yang merupakan bagian dari Business Intelligence yang berguna untuk menyediakan laporan analisis, seperti penjualan, marketing, BPM (Business Process Management), *budgeting*, *forecasting*, keuangan dan sebagainya. model multidimensi, yang mengijinkan query analisis yang kompleks dengan kecepatan eksekusi yang tinggi.

Dengan dua sumber diatas, dapat kita ketahui jika OLTP dan OLAP adalah berbeda, walaupun sama – sama online. Dari segi pengertiannya saja dapat kita ketahui bahwa yang satu berorientasi proses dan yang satunya lagi berorientasi subjek. Seperti yang disampaikan [Devi Munandar, th tidak disebutkan] dalam makalahnya yang berjudul "OLAP dan Terminologi Multi-Dimensional Database" yang menceritakan bahwa OLTP dan OLAP adalah suatu pernyataan yang bertolak belakang atau sangat kontras. Dimana dalam makalah tersebut juga disebutkan bahwa OLAP menggambarkan sebuah klas teknologi yang dirancang untuk analisa dan akses data secara khusus sedangkan OLTP

mempunyai karakteristik beberapa user dapat *creating, updating, retrieving* untuk setiap record data, lagi pula OLTP sangat optimal untuk *updating* data.

Dari semua sumber yang ada dapat kita simpulkan bahwa OLTP dan OLAP itu berbeda jauh. Mungkin keduanya hanya sama – sama online dan keduanya sangat erat kaitannya dengan database ataupun data warehouse. OLTP merupakan suatu system yang berorientasi proses, yang berfungsi sebagai operasi harian dan datanya dapat di-edit, di-ganti, atau di-hapus. Sedangkan OLAP suatu system yang berorientasi subjek, yang fungsinya lebih diutamakan dalam mendukung pengambilan keputusan dalam data warehouse sehingga data – data disini bukanlah data yang bisa di-edit, di-ganti, atau di-hapus seperti data – data yang ada dalam OLTP. OLAP biasanya digunakan oleh manager dalam mencari suatu keputusan.

1. **User**

Dalam OLTP, penggunaanya adalah IT Professional sedangkan OLAP penggunaanya adalah Knowledge worker maksudnya penggunaanya adalah seorang yang bertindak dalam subyek tertentu, atau petinggi dalam suatu perusahaan.

2. **Function**

OLTP digunakan sehari-hari untuk proses bisnis seperti toko atau swalayan, sedangkan OLAP digunakan untuk pengambilan keputusan.

3. **Design DB**

Desain dalam OLTP bersifat Entity Relational atau databasenya dinormalisasi dulu sebelum digunakan. Untuk OLAP desain databasenya di de-normalisasi.

4. **Data**

Dalam OLTP datanya adalah hari ini, update setiap saat sedangkan OLAP datanya adalah sekarang dan hari ini yang berguna untuk melakukan analisis ke depan.

5. **Penggunaan**

OLTP digunakan setiap saat, sedangkan OLAP digunakan seperlunya saja.

6. **Access**

OLTP aksesnya bisa write, read dan lain-lain. Sedangkan OLAP sering dibaca karena digunakan untuk analisa.

7. **Unit Pekerjaan**

Kalau OLTP pekerjaannya hanya sederhana misalnya transaksi dalam swalayan. Untuk OLAP query untuk menampilkan data sangat kompleks

8. **Jumlah rekaman yang di akses**

Kalau OLTP sekitar ratusan sampai ribuan, tapi jika OLAP data yang diakses bisa sampai jutaan bahkan milyaran.

9. **Jumlah Pengguna**

Untuk OLTP penggunaanya adalah puluhan, tapi kalau OLAP penggunaanya bisa sampai ratusan bahkan ribuan

10. **Ukuran Database**

Ukuran database untuk OLTP sekitar MB-GB, sedangkan OLAP bisa sampai GB-TB

Dari definisinya jelas, bahwa keduanya merupakan sistem yang menggunakan suatu basis data.

1. **Berdasarkan sumber data**

OLTP : Data Operasional, data OLTP adalah data asli.

OLAP : Data konsolidasi, data OLAP di peroleh dari beberapa OLTP.

2. **Berdasarkan Tujuan datanya**

OLTP : Untuk mengendalikan dan menjalankan tugas2 utama.

OLAP : Untuk membantu dalam perencanaan, memecahkan masalah dan mendukung keputusan.

3. **Berdasarkan Data apa yg di tampilkan**

OLTP : Bisnis proses yg berkelanjutan.

OLAP : Menampilkan data dari berbagai macam aktivitas bisnis.

4. **Query yg digunakan**

OLTP : Simple Query.

OLAP : Complex Queries.

5. **Kecepatan proses**

OLTP : Pada dasarnya sangat cepat.

OLAP : Tergantung dari data yg dilibatkan, proses akan lebih cepat dengan menggunakan fungsi indexing.

6. **Space yg dibutuhkan**

OLTP : Relativ kecil.

OLAP : Lebih besar, karena membutuhkan lebih banyak indexing dibandingkan OLTP.

7. **Database Design**

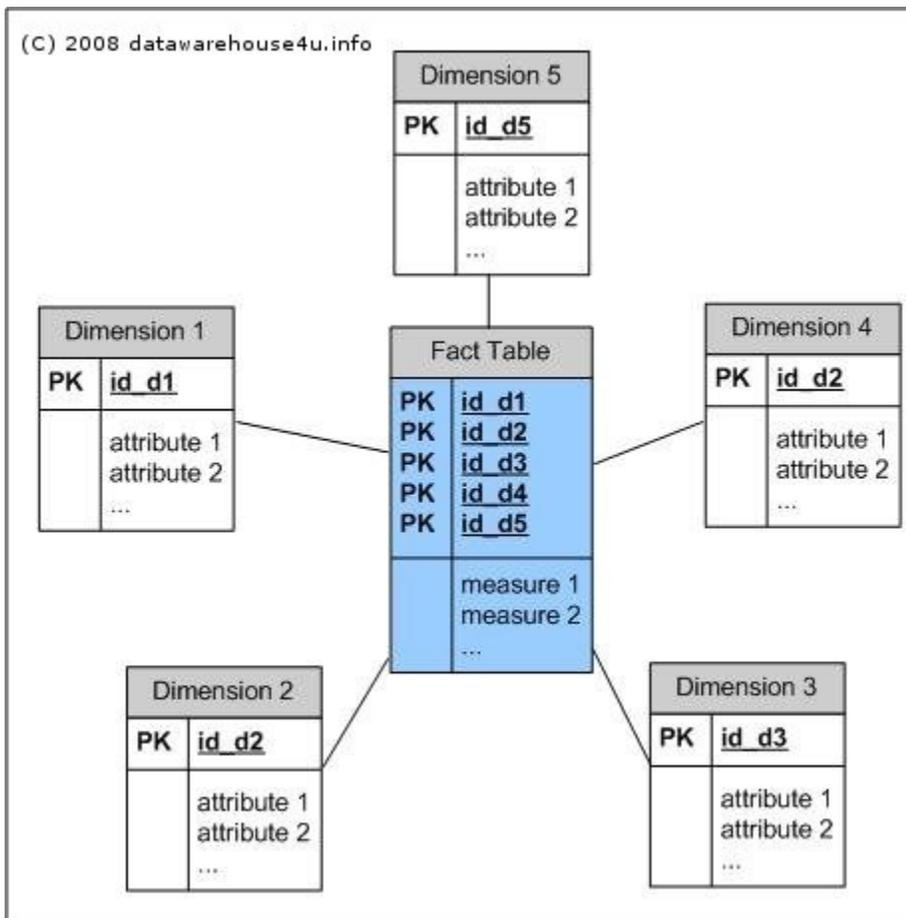
OLTP : Normalized dengan banyak table.

OLAP : De-normalized dengan sedikit table dan menggunakan star / showflake schemas.

Pemodelan Konsep Data Warehouse:

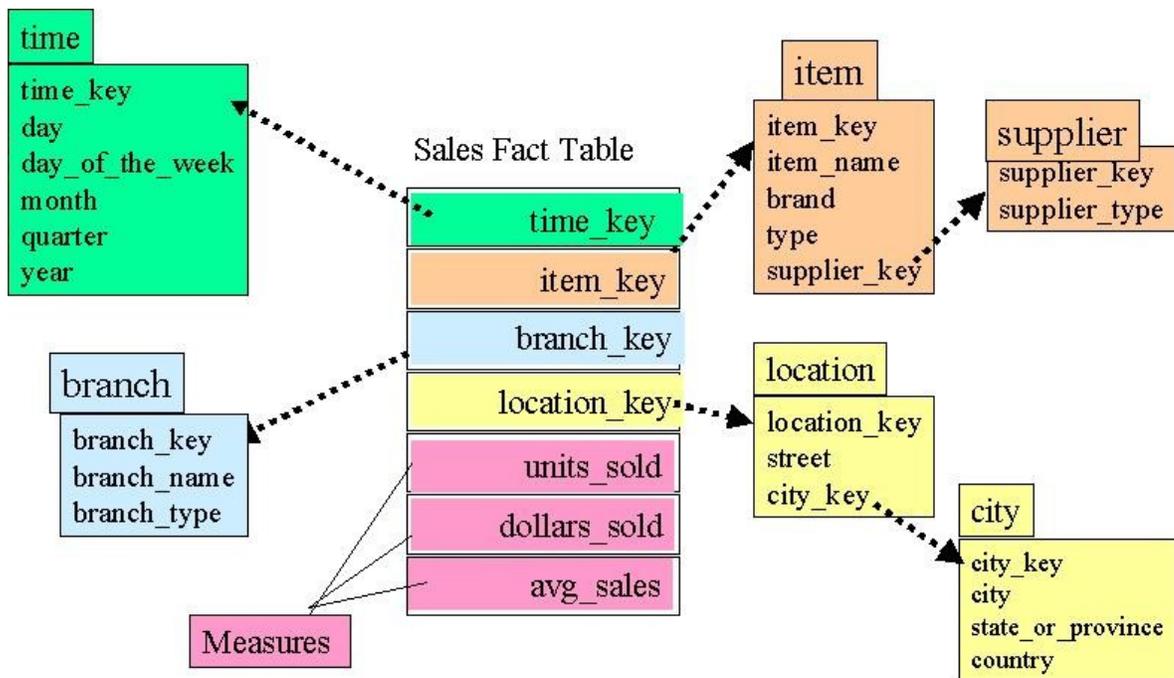
1. **Skema Bintang (Star Skema)**

Skema ini seperti bentuk bintang, dimana tabel fakta(fact) dipusatkan di tengah dimensi tabel, Tabel fakta memiliki beberapa key yang merupakan kunci indeks individual dalam tabel dimensi.



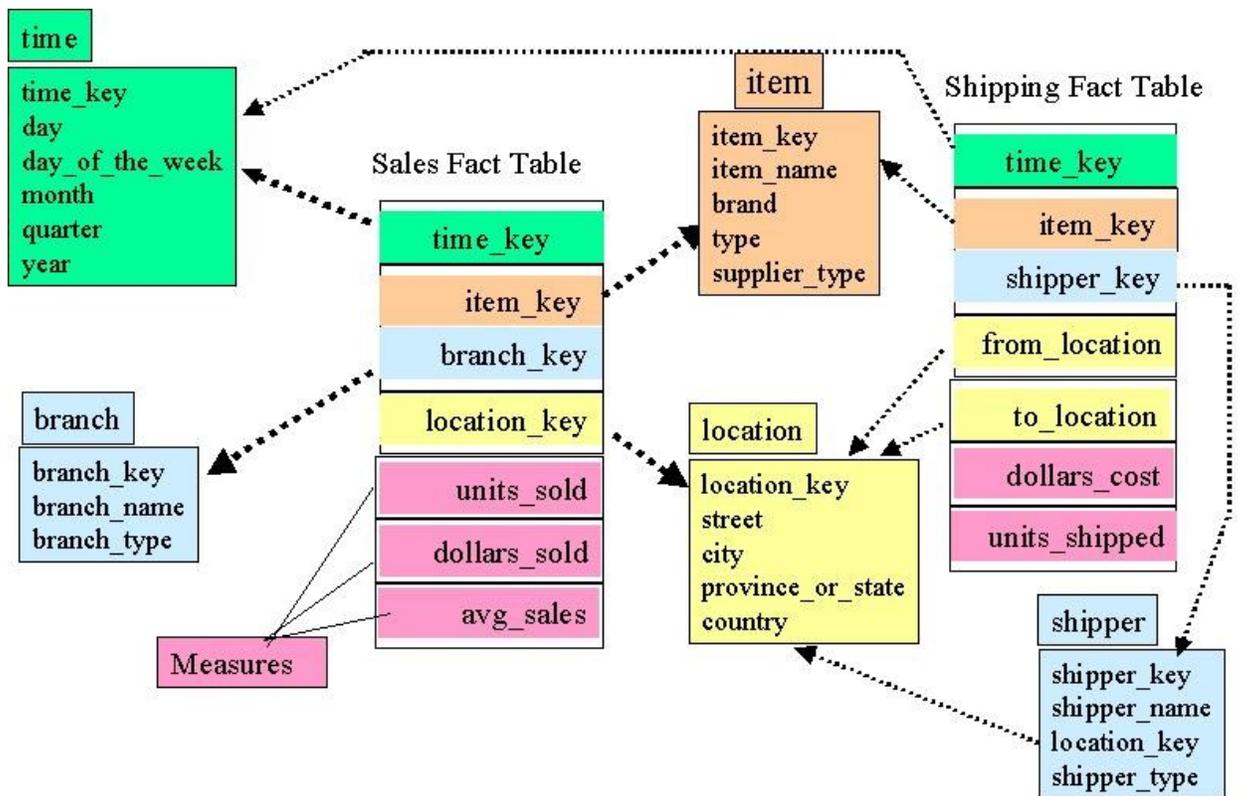
2. Skema Bola Salju (Snowflake Schema)

Skema ini berdasarkan perluasan dari Skema bintang, dengan tambahan beberapa dimensi yang tidak berhubungan dengan tabel fakta (Fact table)



3. Fact Constellations

Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi secara bersama-sama sehingga jika digambarkan akan terlihat seperti sekumpulan bintang.



Kelebihan Data Warehouse:

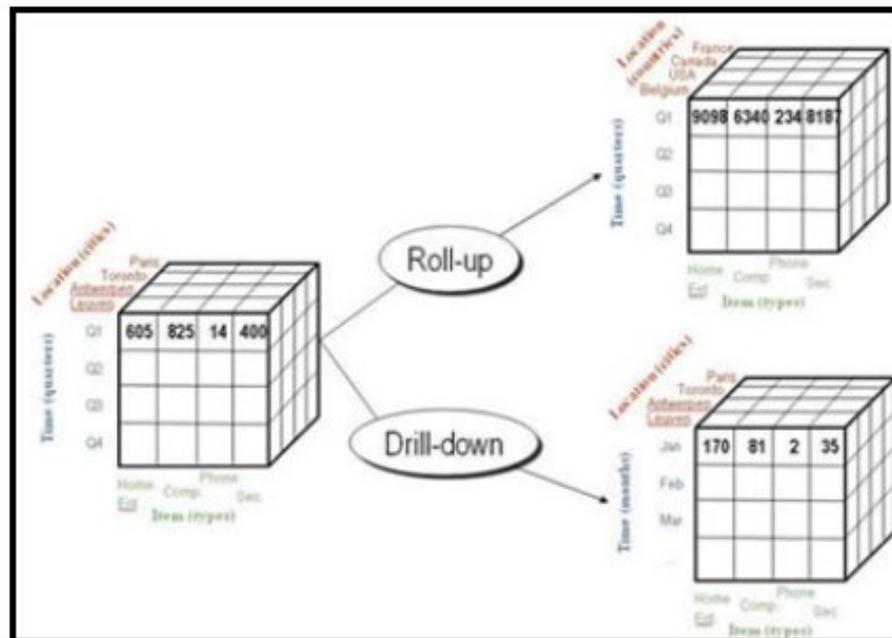
1. Data terorganisir dengan baik untuk query analisis dan sebagai bahan yang baik untuk proses transaksi.
2. Perbedaan struktur data yang banyak macamnya dari sumber yang berbeda dapat di atasi
3. Memiliki aturan transformasi untuk memvalidasi dan menkonsolidasi data dari OLTP ke datawarehouse.
4. Masalah keamanan dan kinerja dapat dipecahkan tanpa perlu mengubah sistem produksi.
5. Memiliki model data yang banyak macamnya, dan tidak tergantung dari format data awal/sumbernya, sehingga memudahkan dalam menciptakan laporan.
6. Proses transformasi/ perpindahan dapat di monitoring. jika terjadi kesalahan dapat di arahkan / di luruskan.
7. Informasi yang disimpan dalam data warehouse, jadi ketika OLTP data sumber nya hilang, informasi yang diolah tetap terjaga dalam suatu datawarehouse.
8. Datawarehouse tidak memperlambat kerja operasional transaksi.
9. Dapat menyediakan laporan yang bermacam-macam

Kekurangan Data Warehouse

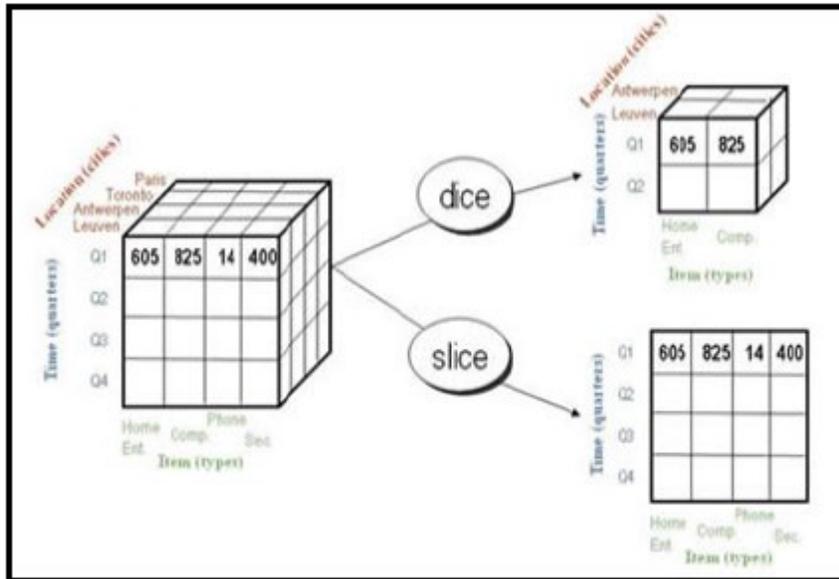
1. Datawarehouse bukan merupakan lingkungan yang cocok untuk data yang tidak terstruktur.
2. Data perlu di ekstrak, diubah (ETL) dan di load ke datawarehouse sehingga membutuhkan tenggang waktu untuk memindahkannya.
3. Semakin lama dipelihara, semakin besar biaya untuk merawat sebuah datawarehouse.
4. Datawarehouse dapat menjadi ketinggalan dari data terbaru yang relatif cepat, karena data yang digunakan di datawarehouse tidak di update secara cepat. sehingga data yang ada tidak optimal.

OLAP dapat digunakan untuk melakukan hal-hal seperti:

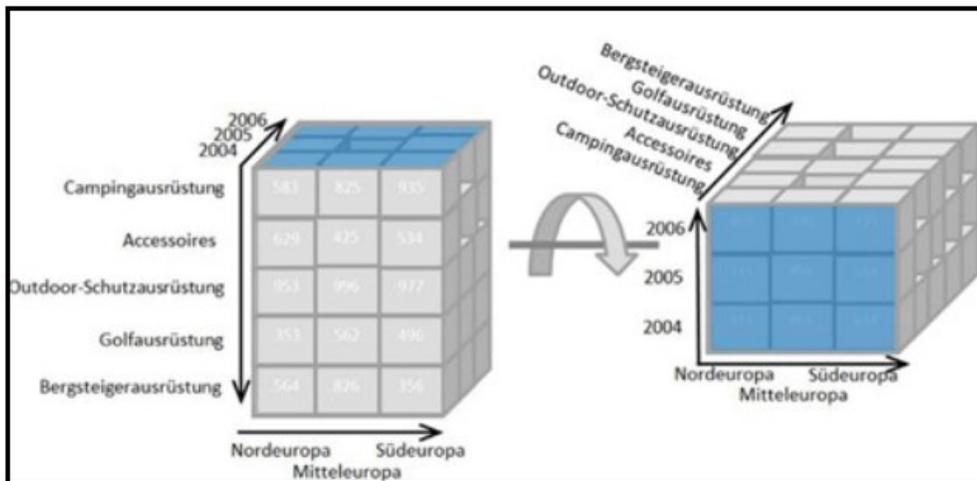
- a. *Consolidation (roll-up)* : Konsolidasi melibatkan pengelompokkan data.
- b. *Drill-down*: Suatu bentuk yang merupakan kebalikan dari konsolidasi, untuk mendapatkan lebih detail tentang suatu dimensi serta bisa dikatakan sebagai suatu navigasi dari tingkat yang lebih umum ke tingkat yang lebih spesifik.
- c. *Slicing and dicing*: menjabarkan pada kemampuan untuk melihat data dari berbagai sudut pandang.
- d. *Pivot*: Menampilkan nilai-nilai ukuran dalam tata letak tabel yang berbeda dan juga bisa mengatur kembali dimensi dalam OLAP *cube*.



Gambar 5. Roll-up dan Drill-down



Gambar 6. *Slicing and Dicing*



Gambar 7. *Pivot*

Data *mining* merupakan teknologi yang mengaplikasikan algoritma yang canggih dan kompleks untuk menganalisis data dan mencari informasi yang menarik dari kumpulan data tersebut. Perbedaan mendasar antara OLAP dan data *mining* yaitu terletak pada apa yang akan dianalisisnya. Pada OLAP, yang dianalisis adalah modelnya, tetapi pada data *mining* yang dianalisis adalah datanya (harus berjumlah besar).