

Data Warehouse

Pendahuluan

Decision Support System merupakan suatu sistem yang berfungsi sebagai penunjang keputusan. Dengan adanya DSS, pekerjaan dari para pengambil keputusan akan lebih terbantu secara signifikan.

Makalah yang kami susun berikut ini menjelaskan suatu database yang biasa digunakan sebagai pengambil keputusan yaitu data warehouse. Dikarenakan banyak organisasi atau perusahaan kurang dapat menggunakan database operasional dalam mendukung secara langsung pengambilan keputusan.

Penyusun sadar dalam penulisan makalah ini terdapat banyak kekurangan, oleh karena itu, kami menerima saran dan kritik yang membangun agar dikemudian hari kami dapat menyusun suatu makalah dengan lebih baik lagi.

Daftar Isi

Pendahuluan	1
Daftar Isi	3
BAB I	4
Definisi Data Warehouse.....	4
I.1. Pengertian Data, Informasi dan Database	4
I.2. Pengertian Data Warehouse	5
I.3. Istilah-istilah yang berhubungan dengan data warehouse	6
I.4. Karakteristik <i>Data Warehouse</i>	7
 BAB. II	 11
Kegunaan Data Warehouse.....	11
II.1 Perlunya Data Warehouse.....	12
II.2 Tugas-tugas Data warehouse	12
II.3. Keuntungan Data Warehouse.....	14
 BAB III	 16
Membangun Data Warehouse	16
III.1. Menentukan Bentuk Data Warehouse.....	16
III.2. Anatomi Data Warehouse	16
III.3 Arsitektur dan Infrastruktur Data Warehouse	19
III.4. Struktur Data Warehouse	23
III.5 Metodologi Perancangan Database untuk <i>Data Warehouse</i>	26
III.6. Model untuk Data Warehouse	29
III.6.1. Model Dimensional.....	29
 BAB IV	 37
Hubungan Data Warehouse Dengan DSS.....	37
IV.1. Definisi Decision Support System	37
IV.2. Hubungan antar DSS dengan Data Warehouse	38
IV.3. Contoh Data Warehouse Yang Ada.....	38
 Bab V	 39
Kesimpulan dan Penutup	39
V.1. Kesimpulan.....	39
 Daftar Pustaka.....	 40

BAB I

Definisi Data Warehouse

I.1. Pengertian Data, Informasi dan Database

Sebelum kita membahas tentang data warehouse, hal yang harus dipahami terlebih dahulu yaitu pengertian tentang data, informasi dan database.

Menurut Steven Alter, data merupakan fakta, gambar atau suara yang mungkin atau tidak berhubungan atau berguna bagi tugas tertentu.

Menurut McLeod, data terdiri dari fakta-fakta dan angka yang secara relatif tidak berarti bagi pemakai. Sedangkan informasi adalah data yang sudah diproses atau data yang memiliki arti.

Disini kita dapat melihat bahwa data merupakan suatu bentuk keterangan-keterangan yang belum diolah atau dimanipulasi sehingga belum begitu berarti bagi sebagian pemakai. Sedangkan informasi merupakan data yang sudah di olah sehingga memiliki arti.

Menurut James A. O'Brien Database adalah suatu koleksi terintegrasi dimana secara logika berhubungan dengan record dari file.

Menurut Fatansyah, Database adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan(redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Jadi Database adalah tempat penyimpanan data yang saling berhubungan secara logika, sehingga bisa digunakan untuk mendapatkan suatu informasi yang diperlukan oleh suatu organisasi atau perusahaan.

Sedangkan data yang diperoleh suatu organisasi atau perusahaan umumnya didapat dari kegiatan operasional sehari-hari atau hasil dari transaksi.

Dari perkembangan model database, muncullah apa yang disebut dengan data warehouse.

I.2. Pengertian Data Warehouse

Pengertian Data Warehouse dapat bermacam-macam namun mempunyai inti yang sama, seperti pendapat beberapa ahli berikut ini :

Menurut W.H. Inmon dan Richard D.H., *data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi, time-variant, dan bersifat tetap dari koleksi data dalam mendukung proses pengambilan keputusan management.

Menurut Vidette Poe, *data warehouse* merupakan database yang bersifat analisis dan *read only* yang digunakan sebagai fondasi dari sistem penunjang keputusan.

Menurut Paul Lane, *data warehouse* merupakan database relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya mengandung *history* data dari proses transaksi dan bisa juga data dari sumber lainnya. *Data warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber.

Jadi, *data warehouse* merupakan metode dalam perancangan database, yang menunjang DSS(Decision Support System) dan EIS (Executive Information System). Secara fisik *data warehouse* adalah database, tapi perancangan data warehouse dan database sangat berbeda. Dalam perancangan database tradisional menggunakan normalisasi, sedangkan pada *data warehouse* normalisasi bukanlah cara yang terbaik.

Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah database yang saling bereaksi yang dapat digunakan untuk *query* dan analisis, bersifat orientasi subjek, terintegrasi, time-variant, tidak berubah yang digunakan untuk membantu para pengambil keputusan.

I.3. Istilah-istilah yang berhubungan dengan data warehouse

Istilah-istilah yang berkaitan dengan data warehouse :

1. Data Mart

Adalah suatu bagian pada data warehouse yang mendukung pembuatan laporan dan analisa data pada suatu unit, bagian atau operasi pada suatu perusahaan.

2. On-Line Analytical Processing(OLAP)

Merupakan suatu pemrosesan database yang menggunakan tabel fakta dan dimensi untuk dapat menampilkan berbagai macam bentuk laporan, analisis, query dari data yang berukuran besar.

3. On-Line Transaction Processing(OLTP)

Merupakan suatu pemrosesan yang menyimpan data mengenai kegiatan operasional transaksi sehari-hari.

4. Dimension Table

Tabel yang berisikan kategori dengan ringkasan data detail yang dapat dilaporkan. Seperti laporan laba pada tabel fakta dapat dilaporkan sebagai dimensi waktu(yang berupa perbulan, perkwartal dan pertahun).

5. Fact Table

Merupakan tabel yang umumnya mengandung angka dan data history dimana *key* (kunci) yang dihasilkan sangat unik, karena *key* tersebut terdiri dari *foreign key*(kunci asing) yang merupakan *primary key* (kunci utama) dari beberapa dimension table yang berhubungan.

6. DSS

Merupakan sistem yang menyediakan informasi kepada pengguna yang menjelaskan bagaimana sistem ini dapat menganalisa situasi dan mendukung suatu keputusan yang baik.

I.4. Karakteristik *Data Warehouse*

Karakteristik data warehouse menurut Inmon, yaitu :

1. *Subject Oriented* (Berorientasi subject)

Data warehouse berorientasi subject artinya data warehouse didesain untuk menganalisa data berdasarkan subject-subject tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu.

Data warehouse diorganisasikan disekitar subjek-subjek utama dari perusahaan(customers,products dan sales) dan tidak diorganisasikan pada area-area aplikasi utama(customer invoicing,stock control dan product sales). Hal ini dikarenakan kebutuhan dari data warehouse untuk menyimpan data-data yang bersifat sebagai penunjang suatu keputusan, dari pada aplikasi yang berorientasi terhadap data.

Jadi dengan kata lain, data yang disimpan adalah berorientasi kepada subjek bukan terhadap proses. Secara garis besar perbedaan antara data operasional dan data warehouse yaitu :

Data Operasional	Data Warehouse
Dirancang berorientasi hanya pada aplikasi dan fungsi tertentu	Dirancang berdasar pada subjek-subjek tertentu(utama)
Focusnya pada desain database dan proses	Focusnya pada pemodelan data dan desain data
Berisi rincian atau detail data	Berisi data-data history yang akan dipakai dalam proses analisis
Relasi antar table berdasar aturan terkini(selalu mengikuti rule(aturan) terbaru)	Banyak aturan bisnis dapat tersaji antara tabel-tabel

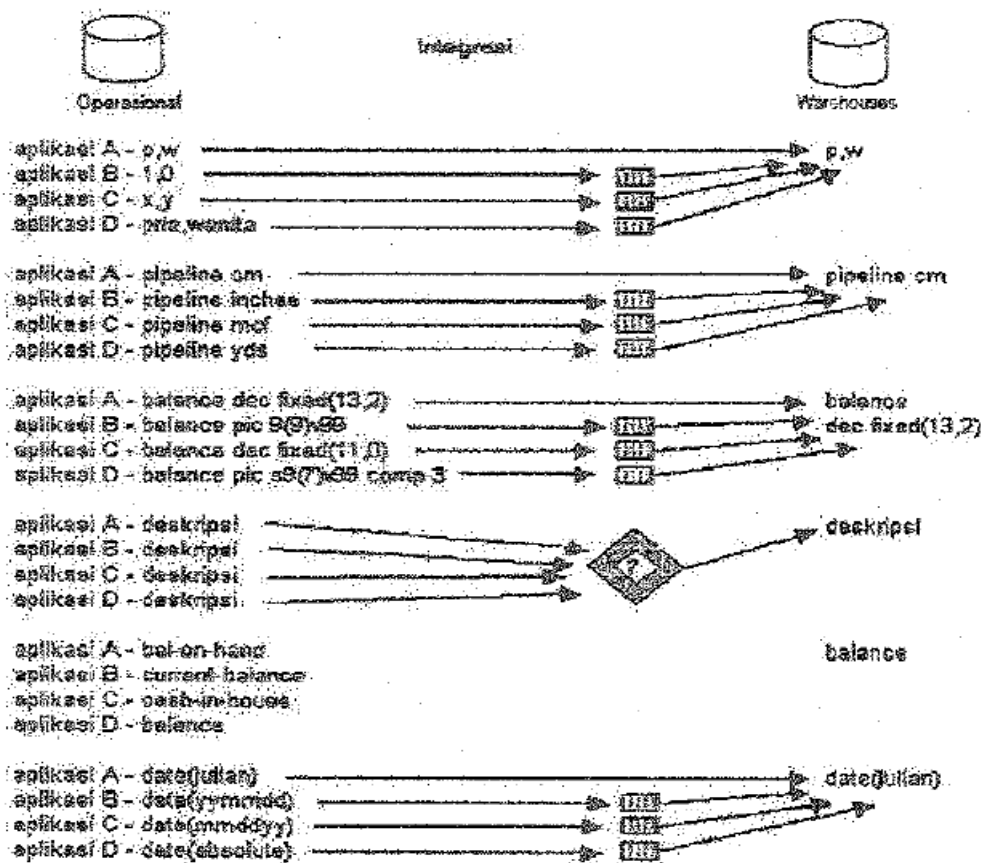
2. *Integrated* (Terintegrasi)

Data Warehouse dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah kedalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Dengan demikian data tidak bisa dipecah-pecah karena data yang ada merupakan suatu kesatuan yang menunjang keseluruhan konsep data warehouse itu sendiri.

Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara seperti konsisten dalam penamaan variable, konsisten dalam ukuran variable, konsisten dalam struktur pengkodean dan konsisten dalam atribut fisik dari data.

Contoh pada lingkungan operasional terdapat berbagai macam aplikasi yang mungkin pula dibuat oleh developer yang berbeda. Oleh karena itu, mungkin dalam aplikasi-aplikasi tersebut ada variable yang memiliki maksud yang sama tetapi nama dan format nya berbeda.

Variable tersebut harus dikonversi menjadi nama yang sama dan format yang disepakati bersama. Dengan demikian tidak ada lagi kerancuan karena perbedaan nama, format dan lain sebagainya. Barulah data tersebut bisa dikategorikan sebagai data yang terintegrasi karena kekonsistennannya.



Integrasi Data Warehouse

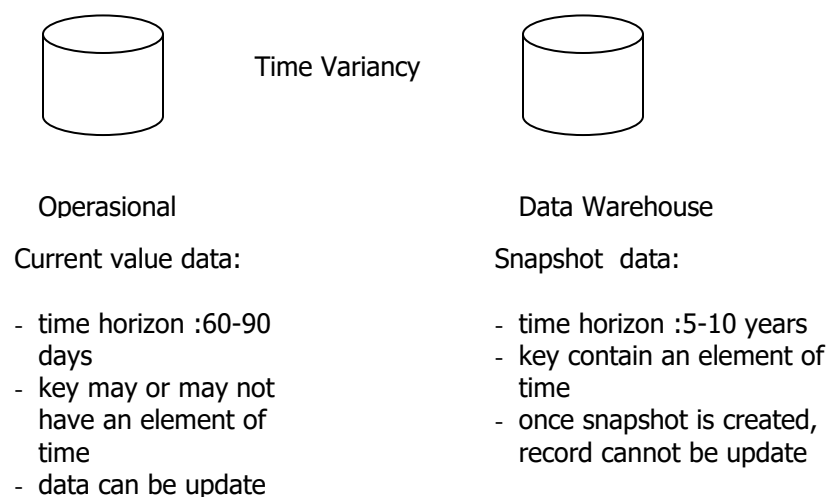
Sumber :

http://www.cait.wustl.edu/papers/prism/vol1_no1/integration/home.html

3. Time-variant (Rentang Waktu)

Seluruh data pada data warehouse dapat dikatakan akurat atau valid pada rentang waktu tertentu. Untuk melihat interval waktu yang digunakan dalam mengukur keakuratan suatu data warehouse, kita dapat menggunakan cara antara lain :

- Cara yang paling sederhana adalah menyajikan data warehouse pada rentang waktu tertentu, misalnya antara 5 sampai 10 tahun ke depan.
- Cara yang kedua, dengan menggunakan variasi/perbedaan waktu yang disajikan dalam data warehouse baik implicit maupun explicit secara explicit dengan unsur waktu dalam hari, minggu, bulan dsb. Secara implicit misalnya pada saat data tersebut diduplikasi pada setiap akhir bulan, atau per tiga bulan. Unsur waktu akan tetap ada secara implisit didalam data tersebut.
- Cara yang ketiga, variasi waktu yang disajikan data warehouse melalui serangkaian snapshot yang panjang. Snapshot merupakan tampilan dari sebagian data tertentu sesuai keinginan pemakai dari keseluruhan data yang ada bersifat read-only.



Time Variance Data Warehouse

Sumber :

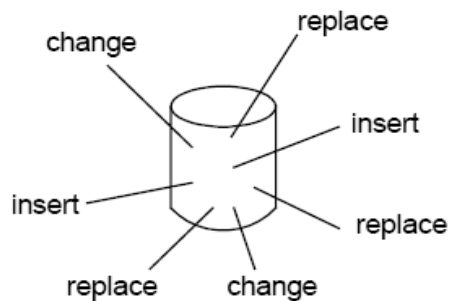
http://www.cait.wustl.edu/papers/prism/vol1_no1/time_variance/home.html

4. Non-Volatile

Karakteristik keempat dari data warehouse adalah non-volatile, maksudnya data pada data warehouse tidak di-*update* secara *real time* tetapi di *refresh* dari sistem operasional secara reguler. Data

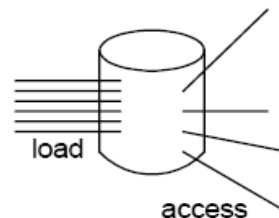
yang baru selalu ditambahkan sebagai suplemen bagi database itu sendiri dari pada sebagai sebuah perubahan. Database tersebut secara kontinyu menyerap data baru ini, kemudian secara incremental disatukan dengan data sebelumnya.

Berbeda dengan database operasional yang dapat melakukan update, insert dan delete terhadap data yang mengubah isi dari database sedangkan pada data warehouse hanya ada dua kegiatan memanipulasi data yaitu loading data (mengambil data) dan akses data (mengakses data warehouse seperti melakukan query atau menampilkan laporan yang dibutuhkan, tidak ada kegiatan updating data).



operational

data is updated on a record-by-record basis regularly



data warehouse

data is loaded into the warehouse and is accessed there, but once snapshot of data is made, the data in the warehouse does not change

Non Volatile Data Warehouse

Sumber :

http://www.cait.wustl.edu/papers/prism/vol1_no1/nonvolatile/home.html

BAB. II

Kegunaan Data Warehouse

II.1 Perlunya Data Warehouse

Seperti pengertian-pengertian yang kita sebutkan sebelumnya, data warehouse diperlukan bagi para pengambil keputusan manajemen dari suatu organisasi/perusahaan.

Dengan adanya data warehouse, akan mempermudah pembuatan aplikasi-aplikasi DSS dan EIS karena memang kegunaan dari data warehouse adalah khusus untuk membuat suatu database yang dapat digunakan untuk mendukung proses analisa bagi para pengambil keputusan.

II.2 Tugas-tugas Data warehouse

Ada empat tugas yang bisa dilakukan dengan adanya data warehouse Menurut Williams, keempat tugas tersebut yaitu:

a. Pembuatan laporan

Pembuatan laporan merupakan salah satu kegunaan data warehouse yang paling umum dilakukan. Dengan menggunakan query sederhana didapatkan laporan perhari,perbulan, pertahun atau jangka waktu kapanpun yang diinginkan.

b. On-Line Analytical Processing (OLAP)

Dengan adanya data warehouse, semua informasi baik detail maupun hasil summary yang dibutuhkan dalam proses analisa mudah didapat.

OLAP mendayagunakan konsep data multi dimensi dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL. Hal ini dimungkinkan karena pada konsep multi dimensi, maka data yang berupa fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada software OLAP adalah fasilitas rool-up dan drill-down. Drill-down adalah

kemampuan untuk melihat detail dari suatu informasi dan roll-up adalah kebalikannya.

c. Data mining

Data mining merupakan proses untuk menggali(mining) pengetahuan dan informasi baru dari data yang berjumlah banyak pada data warehouse, dengan menggunakan kecerdasan buatan (Artificial Intelligence), statistik dan matematika. Data mining merupakan teknologi yang diharapkan dapat menjembatani komunikasi antara data dan pemakainya.

Beberapa solusi yang diberikan data mining antara lain :

1. Menebak target pasar

Data mining dapat mengelompokkan (clustering) model-model pembeli dan melakukan klasifikasi terhadap setiap pembeli dan melakukan klasifikasi terhadap setiap pemebeli sesuai dengan karakteristik yang diinginkan.

2. Melihat pola beli dari waktu ke waktu

Data mining dapat digunakan untuk melihat pola beli dari waktu ke waktu.

3. *cross-market analysis*

Data mining dapat dimanfaatkan untuk melihat hubungan antara satu produk dengan produk lainnya.

4. Profil pelanggan

Data mining bisa membantu pengguna untuk melihat profil pembeli sehingga dapat diketahui kelompok pembeli tertentu cenderung kepada suatu produk apa saja.

5. Informasi *summary*

Data mining dapat membuat laporan summary yang bersifat multi dimensi dan dilengkapi dengan informasi statistik lainnya.

d. Proses informasi executive

Data warehouse dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan data warehouse segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan data warehouse menjadi target informative bagi user.

II.3. Keuntungan Data Warehouse

Data warehouse merupakan pendekatan untuk menyimpan data dimana sumber-sumber data yang heterogen(yang biasanya tersebar pada beberapa database OLTP) dimigrasikan untuk penyimpanan data yang homogen dan terpisah. Keuntungan yang didapatkan dengan menggunakan data warehouse tersebut dibawah ini (Ramelho).

- Data diorganisir dengan baik untuk *query* analisis dan sebagai bahan untuk pemrosesan transaksi.
- Perbedaan diantara struktur data yang heterogen pada beberapa sumber yang terpisah dapat diatasi.
- Aturan untuk transformasi data diterapkan untuk memvalidasi dan mengkonsolidasi data apabila data dipindahkan dari database OLTP ke *data warehouse*.
- Masalah keamanan dan kinerja bisa dipecahkan tanpa perlu mengubah sistem produksi.

Membangun data warehouse tentu saja memberikan keuntungan lebih bagi suatu perusahaan, karena data warehouse dapat memberikan keuntungan strategis pada perusahaan tersebut melebihi pesaing-pesaing mereka. Keuntungan tersebut diperoleh dari beberapa sumber (Sean Nolan,Tom Huguelet):

- Kemampuan untuk mengakses data yang besar
- Kemampuan untuk memiliki data yang konsisten
- Kemampuan kinerja analisa yang cepat
- Mengetahui adanya hasil yang berulang-ulang
- Menemukan adanya celah pada *business knowledge* atau *business process*.
- Mengurangi biaya administrasi
- Memberi wewenang pada semua anggota dari perusahaan dengan menyediakan kepada mereka informasi yang dibutuhkan agar kinerja bisa lebih efektif.

BAB III

Membangun Data Warehouse

III.1. Menentukan Bentuk Data Warehouse

Data warehouse memiliki berbagai macam bentuk yang sering digunakan. Jadi sebelum membangun suatu data warehouse kita harus memutuskan bentuk data warehouse seperti apa yang dibutuhkan oleh aplikasi yang kita rancang.

III.2. Anatomi Data Warehouse

Penerapan awal dari arsitektur *data warehouse* dibuat berdasarkan konsep bahwa *data warehouse* mengambil data dari berbagai sumber dan memindahkannya ke dalam pusat pengumpulan data yang besar. Konsep ini sebenarnya lebih cenderung kepada sebuah lingkungan *mainframe* yang terpusat.

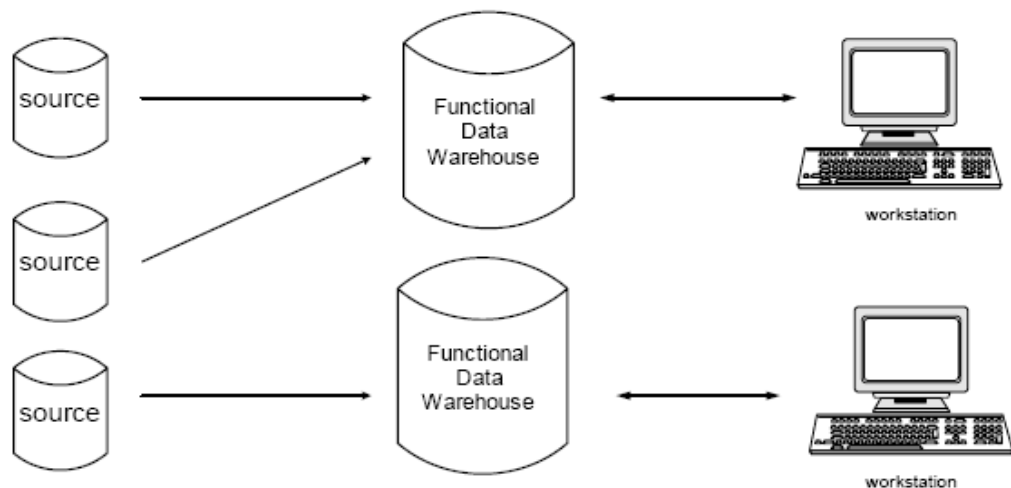
Keunggulan teknologi *Client Server* memungkinkan *data warehouse* diterapkan dalam berbagai macam cara untuk menampung kebutuhan pemakai sistem secara lebih proposional. Dalam suatu kasus, misalkan saja pemakai tertentu perlu menggabungkan data dari sebuah sistem pengumpulan data yang statis dengan data dari sistem operasional yang dinamis hanya dengan sebuah *query* saja.

Berikut ini adalah tiga jenis dasar sistem *Data Warehouse* :

1. Functional Data Warehouse (*Data Warehouse Fungsional*)

Kata operasional disini merupakan database yang diperoleh dari kegiatan sehari-hari. Data warehouse dibuat lebih dari satu dan dikelompokkan berdasar fungsi-fungsi yang ada di dalam perusahaan seperti fungsi keuangan(*financial*),marketing,personalia dan lain-lain.

Keuntungan dari bentuk data warehouse seperti ini adalah, sistem mudah dibangun dengan biaya relatif murah sedangkan kerugiannya adalah resiko kehilangan konsistensi data dan terbatasnya kemampuan dalam pengumpulan data bagi pengguna.

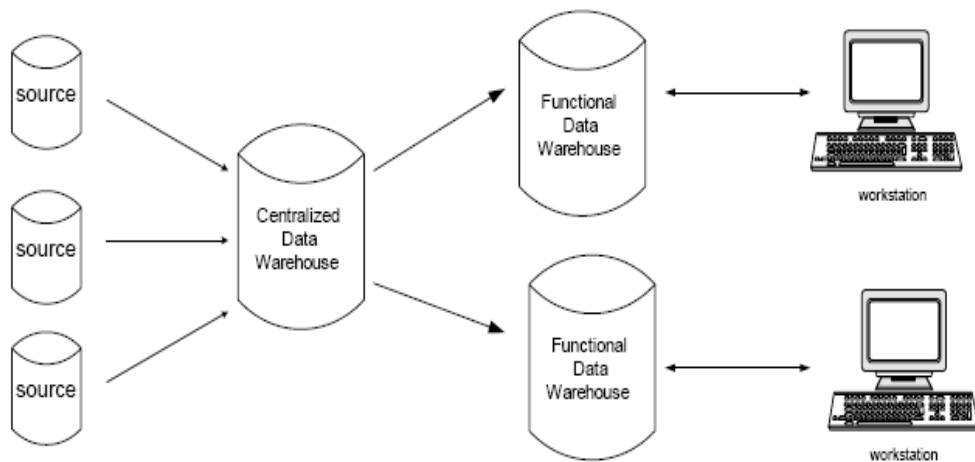


Bentuk data warehouse fungsional

2. Centralized Datawarehouse (*Data Warehouse Terpusat*)

Bentuk ini terlihat seperti bentuk data warehouse fungsional, namun terlebih dahulu sumber data dikumpulkan dalam satu tempat terpusat, kemudian data disebar ke dalam fungsinya masing-masing, sesuai kebutuhan perusahaan. Data warehouse terpusat ini, biasa digunakan oleh perusahaan yang belum memiliki jaringan eksternal.

Keuntungan dari bentuk ini adalah data benar-benar terpadu karena konsistensinya yang tinggi sedang kerugiannya adalah biaya yang mahal serta memerlukan waktu yang cukup lama untuk



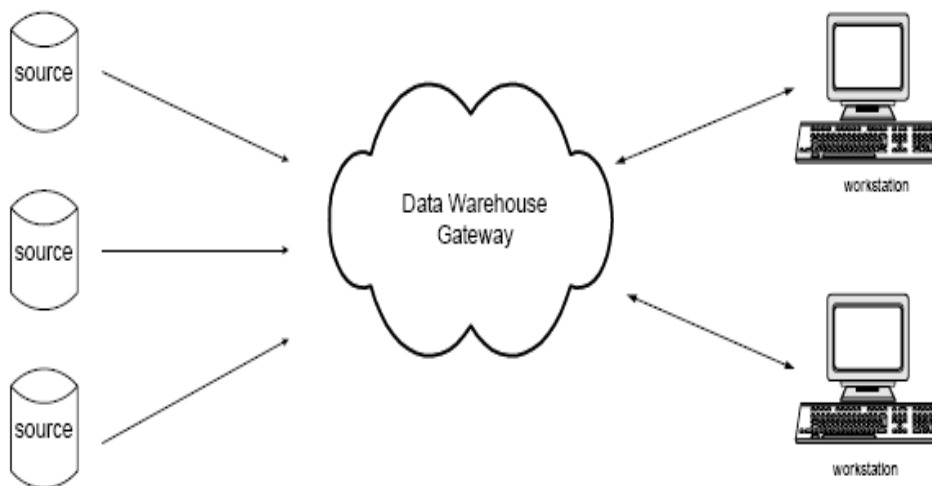
membangi

Bentuk data warehouse terpusat

3. Distributed Data Warehouse (*Data Warehouse terdistribusi*)

Pada data warehouse terdistribusi ini, digunakan gateway yang berfungsi sebagai jembatan penghubung antara data warehouse dengan workstation yang menggunakan sistem beraneka ragam. Dengan sistem terdistribusi seperti ini memungkinkan perusahaan dapat mengakses sumber data yang berada diluar lokasi perusahaan(eksternal).

Keuntungannya adalah data tetap konsisten karena sebelum data digunakan data terlebih dahulu di sesuaikan atau mengalami proses sinkronisasi. Sedangkan kerugiannya adalah lebih kompleks untuk diterapkan karena sistem operasi dikelola secara terpisah juga biayanya yang paling mahal dibandingkan dengan dua bentuk data



warehouse lainnya.

III.3 Arsitektur dan Infrastruktur Data Warehouse

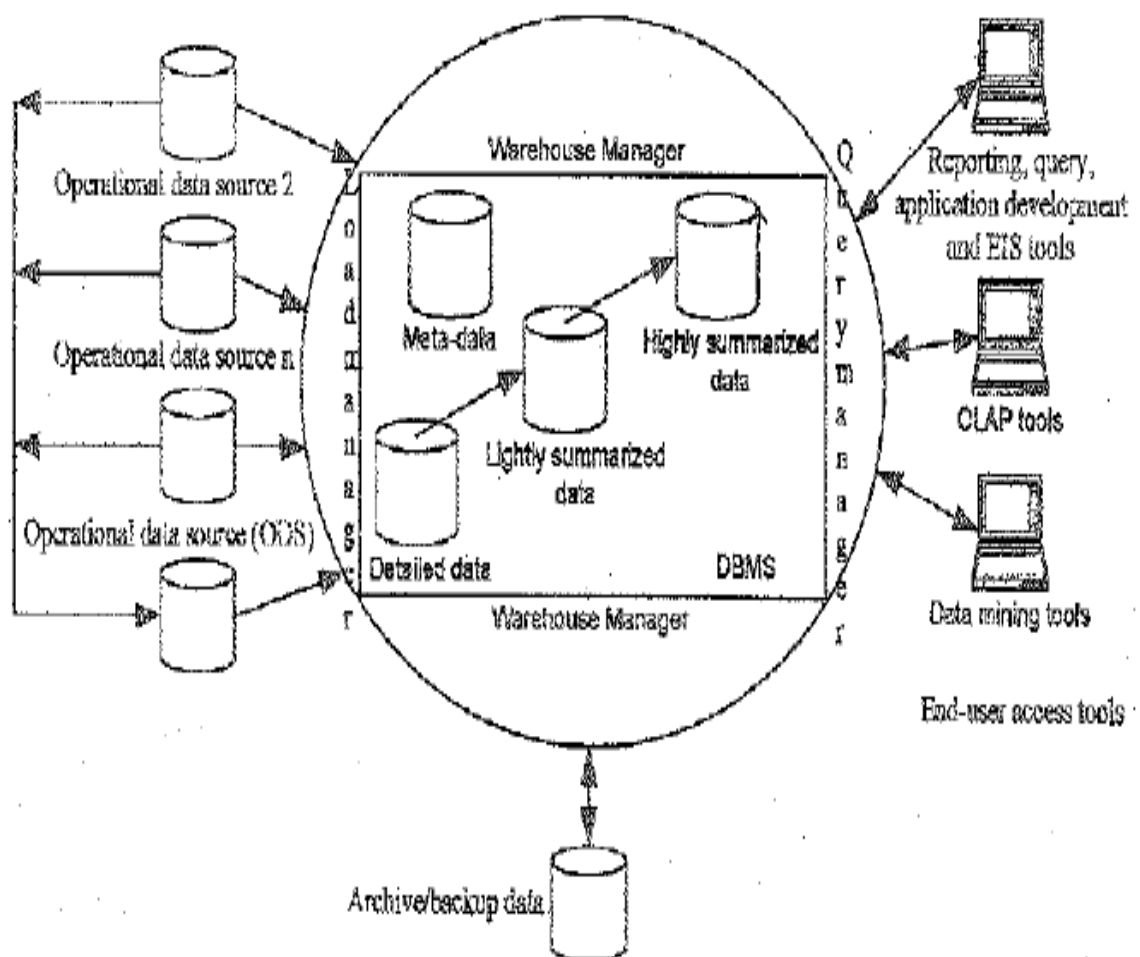
III.3.1. Arsitektur Data Warehouse

Menurut Poe, arsitektur adalah sekumpulan atau struktur yang memberikan kerangka untuk keseluruhan rancangan suatu sistem atau produk. Ada arsitektur *client-server*, arsitektur *networking* dan masih banyak arsitektur lainnya. Arsitektur data menyediakan kerangka dengan mengidentifikasi dan memahami bagaimana data akan pindah melalui sistem dan digunakan dalam perusahaan. Arsitektur data untuk *data warehouse* mempunyai komponen utama yaitu *read-only database*.

Karakteristik arsitektur *data warehouse* (Poe) :

1. Data diambil dari sistem asal (sistem informasi yang ada), database dan file.
2. Data dari sistem asal diintegrasikan dan ditransformasi sebelum disimpan ke dalam *Database Management System* (DBMS) seperti Oracle, Ms SQL Server, Sybase dan masih banyak yang lainnya.
3. *Data warehouse* merupakan sebuah database terpisah bersifat hanya dapat dibaca yang dibuat khusus untuk mendukung pengambilan keputusan
4. Pemakai mengakses *data warehouse* melalui aplikasi *front end tool*

Arsitektur dan komponen utama dari *data warehouse* dapat dilihat pada gambar berikut ini :



Sumber : Conolly,T.M.,Begg

a. *Operational Data*

Sumber data dari *data warehouse* dapat diambil langsung dari *mainframe*, basis data relasional seperti Oracle, Ms SQL server dan sebagainya. Selain itu dapat melalui *Operational Data Source(ODS)*. ODS menampung data yang diekstrak dari sistem utama atau sumber-sumber data yang ada dan kemudian data hasil ekstraksi tersebut dibersihkan.

b. *Load manager*

Load manager juga disebut sebagai komponen *front-end* yang bertugas melakukan seluruh operasi yang berhubungan dengan ekstraksi dan me-*load* data ke *warehouse*.

c. *Warehouse Manager*

Warehouse manager melakukan seluruh operasi-operasi yang berhubungan dengan kegiatan manajemen data di dalam *warehouse*. Operasi-operasi tersebut meliputi :

- Analisis terhadap data untuk memastikan konsistensi
- Transformasi dan penggabungan sumber data dari tempat penyimpanan sementara menjadi tabel-tabel *data warehouse*.
- Penciptaan indeks-indeks dan view berdasarkan tabel-tabel dasar
- Melakukan denormalisasi dan agregasi jika diperlukan
- *Backing-Up* dan mengarsipkan data

d. *Query manager*

Query manager juga disebut komponen *back-end*, melakukan operasi-operasi yang berhubungan dengan manajemen *user queries*.

Operasi-operasi yang dilakukan oleh komponen ini termasuk mengarahkan *query* kepada tabel-tabel yang tepat dan menjadwalkan eksekusi dari *query* tersebut.

e. *End-user Access Tools*

Prinsip atau tujuan utama dari dibangunnya *data warehouse* adalah untuk menyediakan informasi bisnis kepada *user-user* untuk dapat melakukan pengambilan keputusan secara cepat dan tepat. *User* ini berinteraksi dengan *warehouse* melalui *end-user access tools*. *Data warehouse* harus secara efisien mendukung secara khusus kebutuhan *user* serta secara rutin melakukan analisis. Performa yang baik dapat dicapai dengan merencanakan dahulu keperluan-keperluan untuk melakukan *joins, summations* dan laporan-laporan per periode dengan *end-users*.

Berdasarkan kategori yang dikemukakan oleh Berson dan Smith terdapat lima grup utama dari tools tersebut, antara lain :

1. *Reporting and query tools*
2. *Application development tools*
3. *Executive information System (EIS) tools*
4. *Online Analytical Processing (OLAP) tools*
5. *Data mining tools*

Arsitektur dan infrastruktur dari *data warehouse* sangat erat hubungannya dan satu dengan lainnya saling berkaitan.

III.3.2. Infrastruktur Data Warehouse

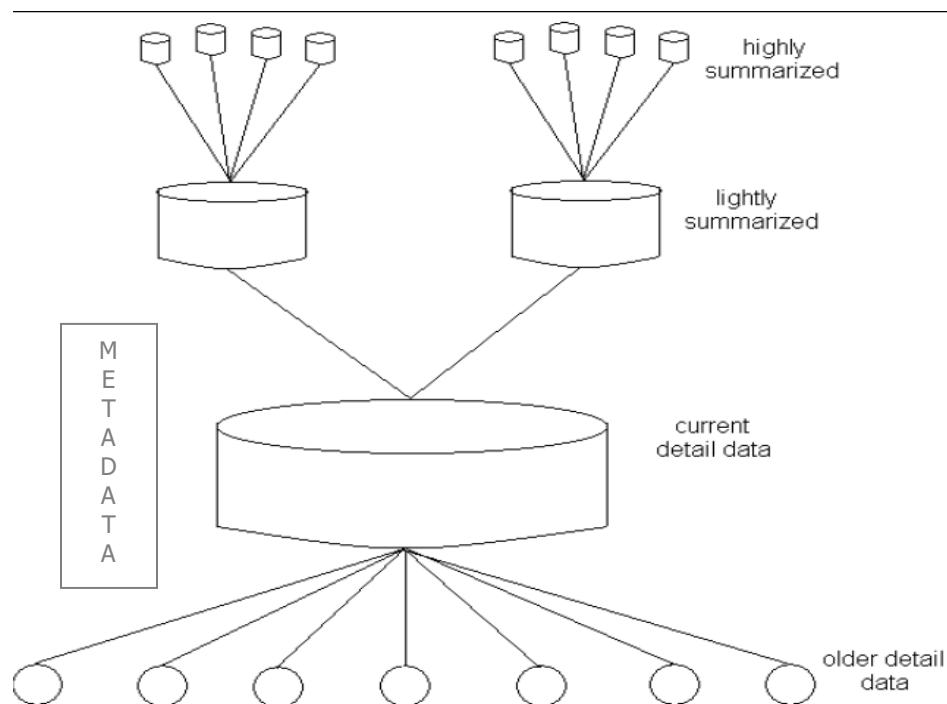
Infrastruktur *data warehouse* adalah software, hardware, pelatihan dan komponen-komponen lainnya yang memberikan dukungan yang dibutuhkan untuk mengimplementasikan *data warehouse* (Poe).

Salah satu instrumen yang mempengaruhi keberhasilan pengembangan *data warehouse* adalah pengidentifikasian arsitektur mana yang terbaik dan infrastruktur apa yang dibutuhkan. Arsitektur yang sama, mungkin memerlukan infrastruktur yang berbeda, tergantung pada lingkungan perusahaan ataupun organisasi.

III.4. Struktur Data Warehouse

Seperti yang kita lihat sebelumnya pada arsitektur *data warehouse*, ada beberapa struktur yang spesifik terdapat pada bagian *warehouse manager*. Bagian tersebut merupakan struktur data warehouse.

Menurut Poe, Vidette, data warehouse memiliki struktur yang spesifik dan mempunyai perbedaan dalam tingkatan detail data dan umur data.



Komponen dari struktur data warehouse adalah:

➤ *Current detail data*

Current detail data merupakan data detil yang aktif saat ini, mencerminkan keadaan yang sedang berjalan dan merupakan level terendah dalam *data warehouse*. Didalam area ini warehouse menyimpan seluruh detail data yang terdapat pada skema basis data. Jumlah data sangat besar sehingga memerlukan *storage* yang besar pula dan dapat diakses secara cepat. Dampak negatif yang ditimbulkan adalah kerumitan untuk mengatur data menjadi meningkat dan biaya yang diperlukan menjadi mahal.

Berikut ini beberapa alasan mengapa *current detail data* menjadi perhatian utama :

1. Menggambarkan kejadian yang baru terjadi dan selalu menjadi perhatian utama
2. Sangat banyak jumlahnya dan disimpan pada tingkat penyimpanan terendah.
3. Hampir selalu disimpan dalam *storage* karena cepat di akses tetapi mahal dan kompleks dalam pengaturannya.
4. Bisa digunakan dalam membuat rekapitulasi sehingga *current detail data* harus akurat.

➤ *Older detail data*

Data ini merupakan data historis dari *current detail data*, dapat berupa hasil cadangan atau archive data yang disimpan dalam *storage* terpisah. Karena bersifat *back-up*(cadangan), maka biasanya data disimpan dalam *storage* alternatif seperti *tape-desk*.

Data ini biasanya memiliki tingkat frekuensi akses yang rendah. Penyusunan file atau *directory* dari data ini di susun berdasarkan umur dari data yang bertujuan mempermudah untuk pencarian atau pengaksesan kembali.

➤ *Lightly summarized data*

Data ini merupakan ringkasan atau rangkuman dari *current detail data*. Data ini dirangkum berdasar periode atau dimensi lainnya sesuai dengan kebutuhan.

Ringkasan dari *current detail data* belum bersifat total *summary*. Data-data ini memiliki detail tingkatan yang lebih tinggi dan mendukung kebutuhan *warehouse* pada tingkat departemen. Tingkatan data ini di sebut juga dengan *data mart*. Akses terhadap data jenis ini banyak digunakan untuk view suatu kondisi yang sedang atau sudah berjalan.

➤ *Highly summarized data*

Data ini merupakan tingkat lanjutan dari *Lightly summarized data*, merupakan hasil ringkasan yang bersifat totalitas, dapat di akses misal untuk melakukan analisis perbandingan data berdasarkan urutan waktu tertentu dan analisis menggunakan data multidimensi.

➤ *Metadata*

Metadata bukan merupakan data hasil kegiatan seperti keempat jenis data diatas. Menurut Poe, metadata adalah 'data tentang data' dan menyediakan informasi tentang struktur data dan hubungan antara struktur data di dalam atau antara *storage*(tempat penyimpanan data).

Metadata berisikan data yang menyimpan proses perpindahan data meliputi database structure, contents, detail data dan summary data, matrices, versioning, aging criteria, versioning, transformation criteria. Metadata khusus dan memegang peranan yang sangat penting dalam data warehouse.

Metadata sendiri mengandung :

- Struktur data
Sebuah direktori yang membantu user untuk melakukan analisis *Decision Support System* dalam pencarian letak/lokasi dalam data warehouse.
- Algoritma
Algoritma digunakan untuk summary data. Metadata sendiri merupakan panduan untuk algoritma dalam melakukan pemrosesan *summary* data antara *current detail data* dengan *lightly summarized data* dan antara *lightly summarized data* dengan *highly summarized data*.
- Mapping
Sebagai panduan pemetaan(*mapping*) data pada saat data di transform/diubah dari lingkup operasional menjadi lingkup data warehouse.

III.5 Metodologi Perancangan Database untuk *Data Warehouse*

Menurut Kimball ada **sembilan tahap metodologi** dalam perancangan database untuk *data warehouse*, yaitu :

DATA WAREHOUSE NILAI MAHASISWA

Langkah 1 : **Pemilihan proses**

- Data mart yang pertama kali dibangun haruslah data mart yang dapat dikirim tepat waktu dan dapat menjawab semua pertanyaan bisnis yang penting
- Pilihan terbaik untuk data mart yang pertama adalah yang berhubungan dengan *sales*, misal *property sales*, *property leasing*, *property advertising*.

Proses : perkuliahan, perwalian, ujian, pengerjaan tugas

Langkah 2 : Pemilihan sumber

- Untuk memutuskan secara pasti apa yang diwakili atau direpresentasikan oleh sebuah tabel fakta.
- Misal, jika sumber dari sebuah tabel fakta properti sale adalah properti sale individual maka sumber dari sebuah dimensi pelanggan berisi rincian pelanggan yang membeli properti utama

Tabel nilai, tabel mahasiswa, tabel dosen, tabel ujian,

Langkah 3 : Mengidentifikasi dimensi

- Set dimensi yang dibangun dengan baik, memberikan kemudahan untuk memahami dan menggunakan data mart
- Dimensi ini penting untuk menggambarkan fakta-fakta yang terdapat pada tabel fakta
- Misal, setiap data pelanggan pada tabel dimensi pembeli dilengkapi dengan id_pelanggan, no_pelanggan, tipe_pelanggan, tempat tinggal, dan lain sebagainya.
- Jika ada dimensi yang muncul pada dua data mart, kedua data mart tersebut harus berdimensi sama, atau paling tidak salah satunya berupa subset matematis dari yang lainnya.
- Jika sebuah dimensi digunakan pada dua data mart atau lebih, dan dimensi ini tidak disinkronisasi, maka keseluruhan *data warehouse* akan gagal, karena dua data mart tidak bisa digunakan secara bersama-sama

Nilai : kode, nama MK, nama dosen, nilai

Mahasiswa : Nim, nama, jurusan

Langkah 4 : Pemilihan fakta

- Sumber dari sebuah tabel fakta menentukan fakta mana yang bisa digunakan dalam data mart.
- Semua fakta harus diekspresikan pada tingkat yang telah ditentukan oleh sumber

Nim	Nama	MK	Jurusan	Nilai
1124	Aldi	Basis data	Inf	A

Langkah 5 : Menyimpan pre-kalkulasi di tabel fakta

- Hal ini terjadi apabila fakta kehilangan statement

Field yang ,ingiin akan ditambahkan

Nilai angka

Langkah 6 : Melengkapi tabel dimensi

- Pada tahap ini kita menambahkan keterangan selengkap-lengkapnya pada tabel dimensi
- Keterangannya harus bersifat intuitif dan mudah dipahami oleh pengguna

Nim	Nama	MK	Jurusan	Nilai	Nilai	Keterangan
1124	Aldi	Basis data	Inf	80	A	Nilai semester 1

Langkah 7 : Pemilihan durasi database

- Misalnya pada suatu perusahaan asuransi, mengharuskan data disimpan selama 10 tahun atau lebih

Pertumbuhana data :

1. Berapa tabel yang dipake
2. Setiap tabel berapa field
3. 1 hari berapa tabel yang di update
4. Berapa kapasitas pertumbuhan per hari (5mb)
10 tahun, kapsitas 1 TB

Langkah 8 : Menelusuri perubahan dimensi yang perlahan

- Ada tiga tipe perubahan dimensi yang perlahan, yaitu :

- Tipe 1. Atribut dimensi yang telah berubah tertulis ulang
- Tipe 2. Atribut dimensi yang telah berubah menimbulkan sebuah dimensi baru
- Tipe 3. Atribut dimensi yang telah berubah menimbulkan alternatif sehingga nilai atribut lama dan yang baru dapat diakses secara bersama pada dimensi yang sama.

1. Mahasiswa : **nim**, nama, jurusan
2. Mata kuliah : **Kd mk**, nama Mk , SKS
3. Dosen : kd _dosen, nama dosen, alamat
4. Nilai : **nim**, nama, **kd mk**, nilai angka , huruf mutu

Langkah 9 : Menentukan prioritas dan mode *query*

- Pada tahap ini kita menggunakan perancangan fisik.

Query, : create table Nilai

Var

Dengan langkah-langkah tadi, seharusnya kita bisa membangun sebuah *data warehouse* yang baik.

III.6. Model untuk Data Warehouse

Berikut di bawah ini adalah penjelasan dari model untuk data warehouse

III.6.1. Model Dimensional

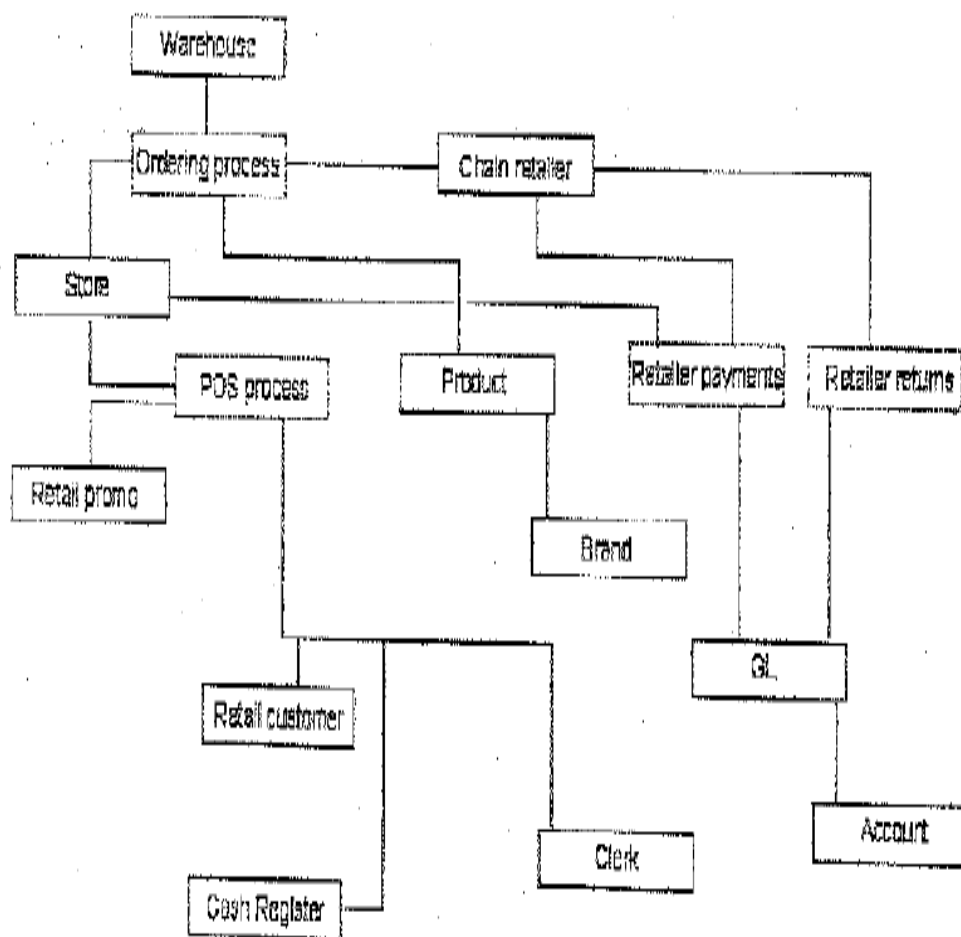
Model dimensional merupakan **rancangan logikal yang bertujuan untuk menampilkan data dalam bentuk standar dan intuitif yang memperbolehkan akses dengan performa yang tinggi.**

Model dimensional menggunakan konsep model hubungan antar **entity (ER)** dengan beberapa batasan yang penting. Setiap model dimensi terdiri dari sebuah tabel dengan sebuah komposit *primary key*, disebut dengan table fakta, dan satu set table yang lebih kecil disebut table dimensi. Setiap table dimensi memiliki sebuah *simple primary key* yang merespon tepat pada satu komponen *primary key* pada table fakta. Dengan kata lain *primary key* pada table fakta terdiri dari dua atau lebih *foreign key*. Struktur karakteristik ini disebut dengan skema bintang atau join bintang.

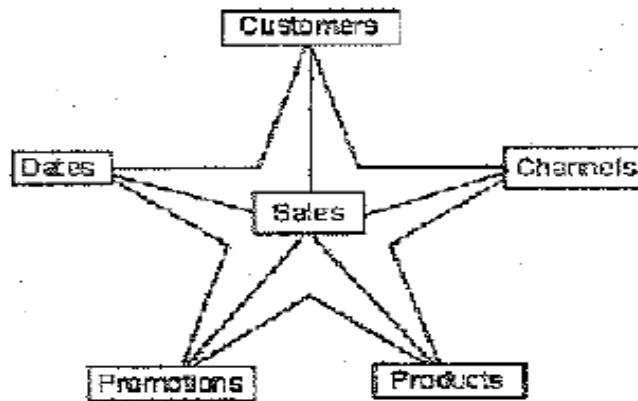
Fitur terpenting dalam model dimensional ini adalah semua *natural keys* diganti dengan kunci pengganti(*surrogate keys*). Maksudnya yaitu setiap kali join antar table fakta dengan table dimensi selalu didasari kunci pengganti. Kegunaan dari kunci pengganti adalah memperbolehkan data pada *data warehouse* untuk memiliki beberapa kebebasan dalam penggunaan data, tidak seperti halnya yang diproduksi oleh sistem OLTP.

Sebuah sistem OLTP memerlukan normalisasi untuk mengurangi redundansi, validasi untuk input data, mendukung volume yang besar dari transaksi yang bergerak sangat cepat. Model OLTP sering terlihat seperti jaring laba-laba yang terdiri atas ratusan bahkan ribuan tabel sehingga sulit untuk dimengerti.

Sebaliknya, dimension model yang sering digunakan pada *data warehouse* adalah skema bintang atau *snowflake* yang mudah dimengerti dan sesuai dengan kebutuhan bisnis, mendukung *query* sederhana dan menyediakan performa *query* yang superior dengan meminimalisasi tabel-tabel join. Berikut contoh perbandingan diagram antara model data OLTP dengan dimension table data warehouse :



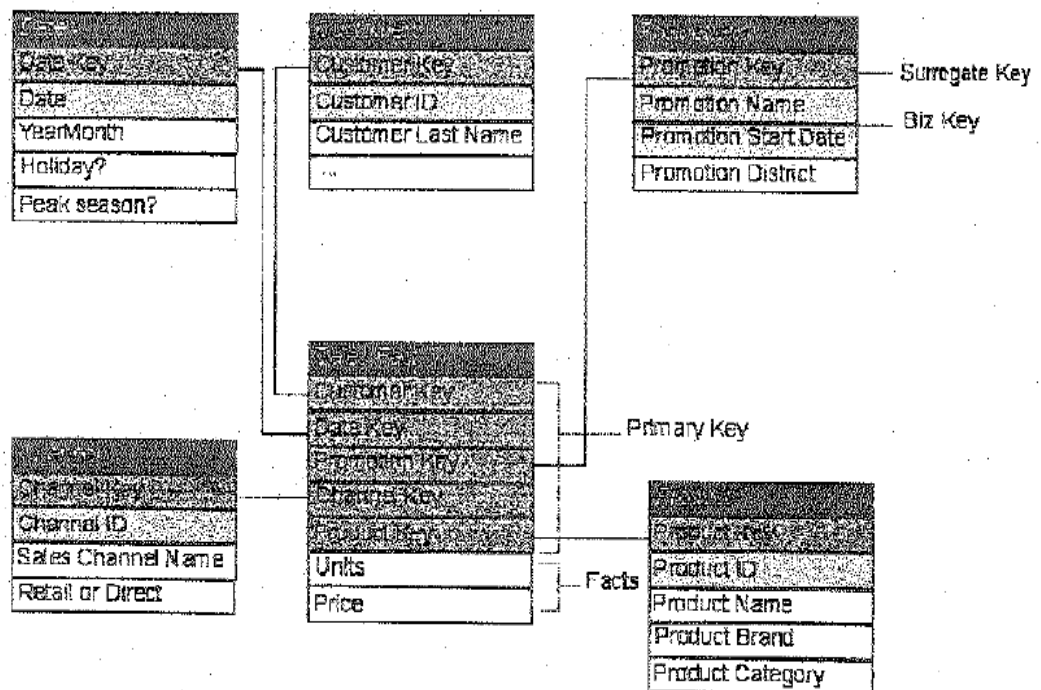
Model data OLTP



Dimension Model

III.6.2.1. Schema Bintang

Skema bintang merupakan struktur logikal yang memiliki **tabel fakta** yang terdiri atas **data faktual ditengahnya**, dan dikelilingi oleh **tabel-tabel dimensi** yang berisi **referensi data**.

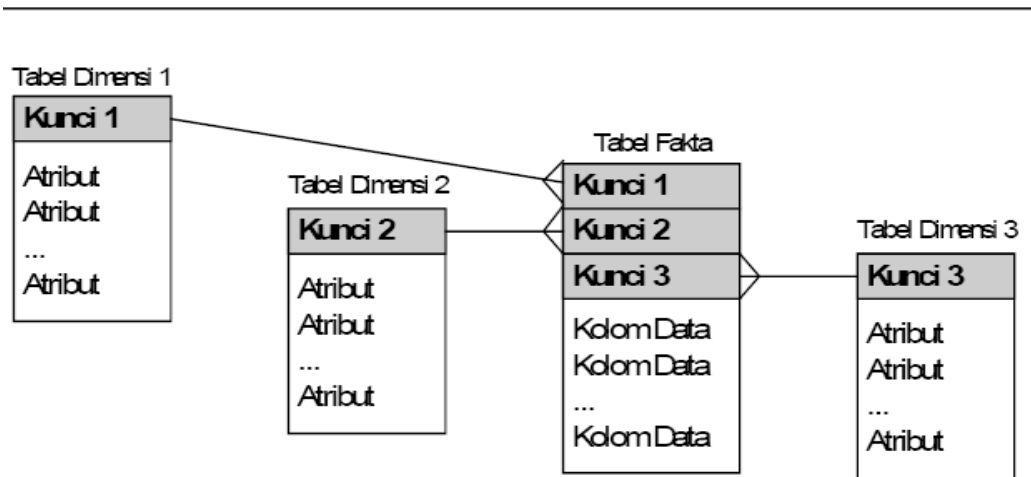


Jenis-jenis Skema Bintang

1. Skema bintang sederhana

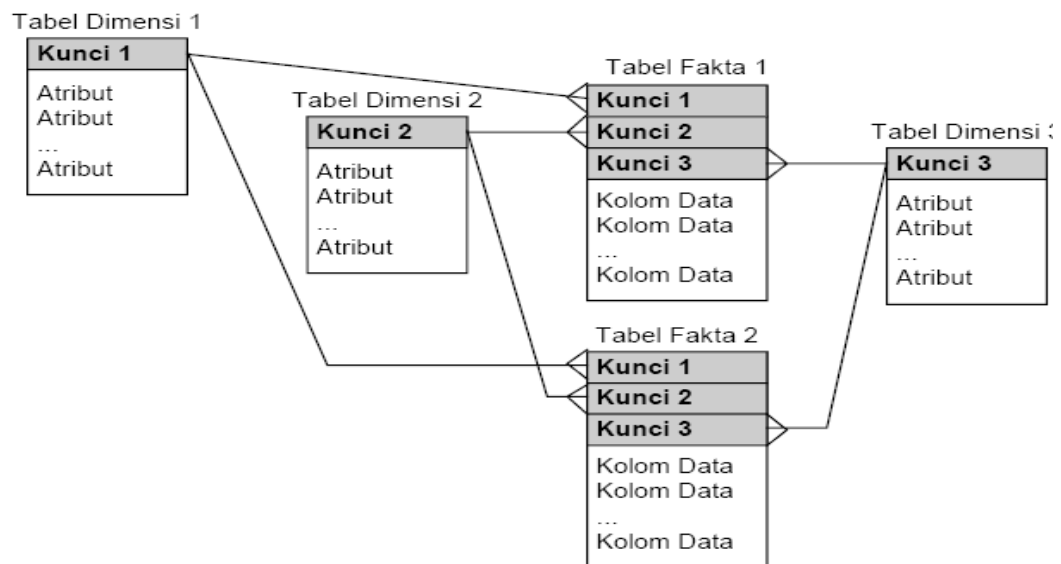
Dalam skema ini, setiap table harus memiliki *primary key* yang terdiri dari satu kolom atau lebih.

Primary key dari table fakta terdiri dari satu atau lebih *foreign key*. *Foreign key* merupakan *primary key* pada table lain.



2. Skema bintang dengan banyak table fakta

Skema bintang juga bisa terdiri dari satu atau lebih table fakta. Dikarenakan karena table fakta tersebut ada banyak, misalnya disamping penjualan terdapat table fakta *forecasting* dan *result*. Walaupun terdapat lebih dari satu table fakta, mereka tetap menggunakan table dimensi bersama-sama.



Adapun ketentuan dalam pembacaan skema bintang adalah :

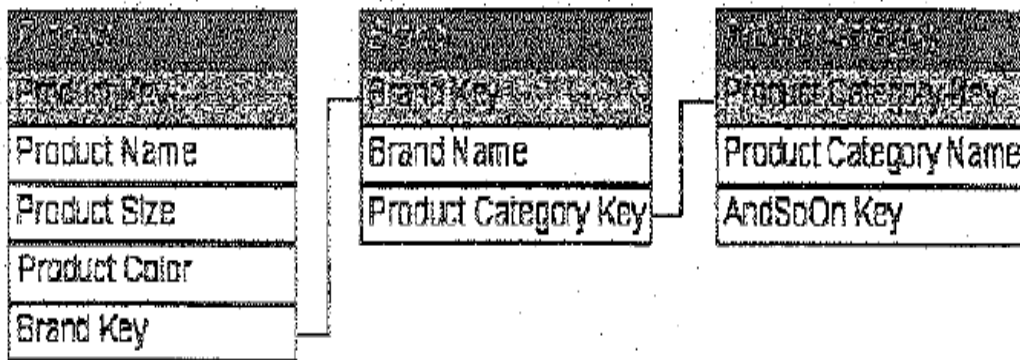
- Bagian yang ada di bawah judul tabel merupakan kolom-kolom tabel tersebut
- Primary key dan Foreign key diberi kotak
- Primary key diarsir sedang Foreign key yang bukan primary tidak
- Foreign key yang berhubungan ditunjukkan dengan garis yang menghubungkan tabel.

Kolom yang bukan kunci disebut kolom data pada table fakta dan atribut pada table dimensi

III.6.2.3 Snowflake Schema

Merupakan varian dari skema bintang dimana table-table dimensi tidak terdapat data yang di denormalisasi. Dengan kata lain satu atau lebih table dimensi tidak bergabung secara langsung kepada table fakta tapi pada table dimensi lainnya. Sebagai contoh, sebuah dimensi yang mendeskripsikan produk

dapat dipisahkan menjadi tiga table (*snowflaked*) seperti contoh dibawah ini :



Snowflake Schemes

III.6.2.4. Star atau Snowflake

Keduanya merupakan model-model dimensional, perbedaannya terletak pada implementasi fisik. Skema snowflake memberi kemudahan pada perawatan dimensi, dikarenakan strukturnya yang lebih normalisasi. Sedangkan skema bintang lebih efisien serta sederhana dalam membuat query dan mudah diakses secara langsung oleh pengguna.

Adapun starflake merupakan gabungan diantara keduanya. Keuntungan menggunakan masing-masing model tersebut dalam data warehouse antara lain :

- Efisien dalam hal mengakses data
- Dapat beradaptasi terhadap kebutuhan-kebutuhan user
- Bersifat fleksibel terhadap perubahan yang terjadi khususnya perubahan yang mengarah pada perkembangan

- Memiliki kemampuan dalam memodelkan situasi bisnis secara umum
- Meskipun skema yang dihasilkan sangat kompleks, tetapi pemrosesan query dapat diperkirakan, hal ini dikarenakan pada level terendah, setiap table fakta harus di query secara independen.

BAB IV

Hubungan Data Warehouse Dengan DSS

IV.1. Definisi Decision Support System

Istilah dari *decision support system* telah digunakan dengan banyak cara (Alter 1980) dan menerima banyak definisi yang berbeda menurut pandangan dari sang penulis (Druzzdel dan Flynn 1999). Finlay (1994) dan lainnya mendefinisikan DSS kurang lebih sebagai sebuah sistem berbasis komputer yang membantu dalam proses pengambilan keputusan.

Turban (1995) mendefinisikan secara lebih spesifik dengan, sesuatu yang interaktif, flexible dan dapat menyesuaikan diri (*adaptable*) dari sistem informasi berdasarkan komputer, khususnya pengembangan untuk mendukung pemecahan masalah dari non-struktur management, untuk meningkatkan pengambilan keputusan. Dengan menggunakan data, mendukung antar muka yang mudah digunakan dan memberikan wawasan untuk sang pengambil keputusan.

Definisi lainnya bisa jadi gugur dibandingkan dengan dua pandangan ekstrim berikut, Keen dan Scott Morton (1978), DSS adalah dukungan berdasar kan komputer untuk para pengambil keputusan management yang berurusan dengan masalah semi-struktur. Sprague dan Carlson (1982), DSS adalah sistem berdasarkan komputer interaktif yang membantu para pengambil keputusan menggunakan data dan model-model untuk memecahkan masalah yang tak terstruktur (*unstructured problem*). Menurut Power (1997), istilah DSS mengingatkan suatu yang berguna dan istilah inklusif untuk banyak jenis sistem informasi yang mendukung pembuatan pengambilan keputusan. Dia dengan penuh humor menambahkan bahwa jika suatu sistem komputer yang bukan OLTP, seseorang akan tergoda untuk menyebutnya sebagai DSS.

Seperti yang kita lihat, DSS memiliki banyak arti dengan maksud yang kurang lebih hampir sama, yaitu suatu sistem komputer yang berguna bagi para pengambil keputusan untuk memecahkan masalah mereka yang kurang lebih berhadapan dengan masalah non-struktur atau semi-struktur.

IV.2. Hubungan antar DSS dengan Data Warehouse

Setelah kita lihat dan selami tentang data warehouse, kita dapat menyimpulkan bahwa data warehouse adalah sebuah model database yang berguna untuk menyimpan dan memproses data dengan pendekatan kepada kegunaan data dalam pengambilan keputusan bagi EIS atau DSS.

Sebuah DSS (tergantung dengan yang disupport-nya)membutuhkan data warehouse agar dapat menjalankan kerjanya dengan baik. Dan memang data warehouse sendiri dibangun untuk memenuhi kebutuhan DSS.

IV.3. Contoh Data Warehouse Yang Ada

Contoh dari data warehouse yang digunakan bersamaan dengan DSS, misalkan saja pegawai peminjaman bank memverikasi data peminta pinjaman atau suatu perusahaan engineer melakukan tawar menawar dalam beberapa project dan ingin tahu jika dia bisa kompetitive dalam harga terhadap para pesaingnya.

Contoh yang lain masih lebih banyak lagi, yang kesemuanya membutuhkan kecepatan dalam pengambilan keputusan dan kemudahan dalam penggunaannya.

Bab V

Kesimpulan dan Penutup

V.1. Kesimpulan

Data warehouse merupakan suatu cara/metode dari suatu database yang berorientasi kepada subjek, non-volatile, time-variance dan terintegrasi yang digunakan untuk mempermudah para pengambil keputusan dalam memecahkan masalah.

Keberadaan data warehouse sangat penting sebagai tools dari DSS, karena data warehouse memang digunakan untuk itu. Dengan adanya data warehouse, diharapkan suatu perusahaan dapat lebih unggul dari kompetitornya dan lebih jeli lagi dalam melihat peluang pasar.

Daftar Pustaka

http://en.wikipedia.org/wiki/Decision_support_system

Conolly, Thomas dan Begg Carolyn (2002). *Database systems – A Practical Approach to Design, Implementation and Management*, edisi-3. Addison Wesley Longman, Inc., USA

Fatansyah (2002). *Buku Teks Ilmu Komputer – Basis Data*, cetakan-4. Informatika

Inmon, W.H. (2002). *Building the Data Warehouse*, edisi-3. Wiley Computer Publishing.

Kimball, R., Merz, R (1998). *The Data Warehouse Lifecycle Toolkit. Expert Methods for Designing, Developing and Deploying Data Warehouses*. Wiley Computer Publishing, Canada.

McLeod, Raymond (1996). *Sistem Informasi Manajemen*, Jilid-1. Terjemahan Teguh, H. PT. Prenhallindo, Jakarta.

Nolan, Sean And Huguelet, Tom (2000). *Microsoft SQL Server 7.0 Data Warehousing Training Kit*. Microsoft Press, USA

Poe, Vidette (1998). *Building Data Warehouse for Decision Support*, edisi-2. Prentice Hall.