BAB 2

LANDASAN TEORI

2.1 Pengertian Data

Menurut Turban (2010, p41), data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan tetapi belum terorganisir untuk menyampaikan arti tertentu.

Menurut Inmon (2005, p493), data adalah kumpulan dari fakta, konsep, atau instruksi pada penyimpanan yang digunakan untuk komunikasi, perbaikan dan diproses secara otomatis yang mempresentasikan informasi yang dapat di mengerti oleh manusia.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar tetapi belum diolah.

2.2 Pengertian Sistem

Menurut O'Brien dan Marakas (2010,p26), Sistem adalah sekelompok komponen yang saling bekerja sama menuju tujuan bersama dengan *input* dan menghasilkan *output* dalam proses transformasi yang teroganisir.

Menurut O'Brien dan Marakas (2005,p5), Sistem adalah sekelompok komponen yang bekerja sama menuju tujuan bersama dengan menerima *input* dan memproduksi *output* di dalam proses transformasi yang terorganisasi

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, Sistem adalah kumpulan komponen-komponen yang saling berhubungan dan bekerja sama dalam mencapai suatu tujuan tertentu.

2.3 Pengertian Informasi

Menurut O'Brien (2010, p34), informasi adalah data yang telah diubah ke dalam suatu konteks yang memiliki arti dan berguna bagi *end user* tertentu.

Menurut Turban (2010, p41), informasi adalah data yang sudah diorganisasi sehingga memiliki arti dan nilai untuk penerima.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, informasi adalah data yang telah diorganisir sehingga memiliki arti dan berguna bagi *end user*.

2.4 Pengertian Sistem Informasi

Menurut O'Brien dan Marakas (2010,p4), Sistem Informasi adalah dapat berupa kombinasi yang teroganisir antara orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber data yang terkumpul, berubah, dan menyebarkan informasi dalam sebuah organisasi.

2.4.1 Sumber daya manusia

Menurut O'Brien dan Marakas (2010,p32) manusia dibutuhkan untuk pengoperasian semua sistem informasi. Sumber daya manusia ini meliputi pemakai akhir dan pakar SI.

a. Pemakai Akhir

adalah orang-orang yang menggunakan sistem informasi atau informasi yang dihasilkan sistem tersebut.

b. Pakar SI

adalah orang-orang yang mengembangkan dan mengoperasikan sistem informasi.

2.4.2 Sumber Daya *Hardware*

Menurut O' Brien dan Marakas (2010,p32) konsep sumber daya *hardware* meliputi semua peralatan dan bahan fisik yang digunakan dalam pemrosesan informasi.

2.4.3 Sumber Daya *Software*

Menurut O'Brien dan Marakas (2010,p33), Konsep sumber daya *software* meliputi semua rangkaian perintah pemrosesan informasi. Konsep umum *software* ini meliputi tidak hanya rangkaian perintah operasi yang disebut progam, dengan *hardware* komputer pengendalian dan langsung, tetapi juga rangkaian perintah pemrosesan informasi yang disebut prosedur. Berikut ini adalah contoh-contoh sumber daya *software* :

a. ***Software* sistem,** seperti progam sistem operasi, yang mengendalikan serta mendukung operasi sistem komputer.
b. ***Software* aplikasi,** yang memprogam pemrosean langsung bagi pengguna tertentu komputer oleh pemakai akhir.

c. **Prosedur,** yang mengoperasikan perintah bagi orang-orang yang akan menggunakan sistem informasi.

2.4.4 Sumber Daya Data

Menurut O'Brien dan Marakas (2010,p33) data lebih daripada hanya bahan baku mentah sistem informasi. Data dapat berupa banyak bentuk, termasuk data alfa numerik tradisional, yang terdiri dari angka dan huruf serta karakter lainnya yang menjelaskan transaksi bisnis dan kegiatan serta entitas lainnya. Data teks, terdiri dari kalimat dan paragraf yang digunakan dalam menulis komunikasi, data gambar, seperti bentuk grafik dan angka, serta gambar video grafis dan video, serta data audio, suara manusia dan suara-suara lainnya, juga merupakan bentuk data yang penting.

2.5 Pengertian *Database*

Menurut Inmon (2005, p493), *database* adalah sekumpulan data yang saling berhubungan dan disimpan berdasarkan suatu skema.

Menurut Connolly dan Begg (2010, p65), *database* adalah kumpulan berbagai data logika terkait dan deskripsi, yang dirancang untuk memenuhi kebutuhan informasi organisasi.

Menurut O'Brien (2010, p173), *database* adalah kumpulan elemen data yang terintegrasi yang berhubungan secara logikal.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, *database* adalah kumpulan data yang berhubungan secara logikal dan disimpan berdasarkan suatu skema untuk memperoleh informasi yang dibutuhkan oleh organisasi.

2.6 *Database Management System (DBMS)*

Menurut Connolly dan Begg (2010, p16), *database management system* adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendifinisikan, membuat, memelihara, dan mengontrol akses ke *database*.

Menurut Turban (2010, p94), *database management system* adalah program *software* atau kumpulan program yang menyediakan akses ke *database*.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, *database management system* adalah suatu program *software* yang menyediakan akses ke *database* dan memungkinkan *user* untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke *database* tersebut.

Keuntungan DBMS, yaitu :

- Mengontrol redudansi data.
- Konsistensi data.
- Lebih banyak informasi dari jumlah data yang sama.
- *Share* data.
- Meningkatkan integritas data.
- Meningkatkan keamanan.
- Standar pelaksanaan.
- Skala ekonomi (data operasional organisasi dijadikan satu *database* dan membuat aplikasi pada satu sumber data sehingga dapat menghemat biaya).
- Keseimbangan aksesibilitas data dan data *responsiveness*.
- Meningkatkan produktivitas.
- Meningkatkan pemeliharaan melalui data *independence*.

- Meningkatkan *konkurensi* (mengurangi *loss* informasi dan *loss* integrasi).
- Meningkatkan layanan *backup* dan *recovery*.

Kerugian DBMS, yaitu :

- Kompleksitas.
- Ukuran.
- Biaya DBMS.
- Biaya penambahan perangkat keras.
- Biaya konversi (biaya staf spesialis, biaya pelatihan).
- *Performance* (tidak dapat berjalan secepat yang diinginkan).
- Resiko kesalahan lebih tinggi.

2.7 Pengertian *Online Transaction Processing (OLTP)*

Menurut Kimball dan Ross (2002, p408), OLTP adalah deskripsi awal dari setiap aktifitas dan sistem yang berhubungan dengan proses memasukkan data ke dalam sebuah *database*.

Menurut Connolly dan Begg (2005, p1149), OLTP adalah sistem yang dirancang untuk menangani jumlah hasil transaksi yang tertinggi, dengan transaksi yang biasanya membuat perubahan kecil bagi data operasional organisasi, yaitu data yang memerlukan penanganan operasi setiap hari.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, OLTP adalah sistem yang dirancang untuk menangani jumlah hasil transaksi yang tertinggi dan data yang memerlukan penanganan operasi setiap hari ke dalam sebuah *database*.

2.8 Pengertian *Online Analytical Processing (OLAP)*

Menurut Connolly dan Begg (2010, p1250), OLAP adalah sintesis, analisis, dan konsolidasi dinamis dari sejumlah besar multidimensional data.

Menurut Kimball dan Ross (2002, p408), OLAP adalah kumpulan aturan yang menyediakan sebuah kerangka dimensional untuk mendukung keputusan.

Menurut Kuo-Ming (2007) OLAP dirancang untuk menyimpan data dan meringkas informasi dengan cepat serta menjawab beberapa pertanyaan misalnya tentang peramalan pasar dan laporan keuangan.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, OLAP adalah sintesis, analisis, dan konsolidasi dinamis dari sejumlah besar multidimensional data untuk mendukung keputusan.

2.9 Pengertian *Data Warehouse*

Menurut Connoly dan Begg (2010, p1197), *datawarehouse* merupakan kumpulan data yang berorientasi subjek, integrasi, berdasarkan waktu, dan tidak mengalami perubahan dalam mendukung proses pengambilan manajemen.

Menurut Inmon (2005, p29), *data warehouse* adalah sekumpulan data yang bersubjek orientasi, terintegrasi, *nonvolatile*, *and time-variant* yang digunakan untuk mendukung proses pengambilan keputusan.

Menurut O'Brien (2010, p191), *data warehouse* adalah kumpulan data yang diekstrak dari *database* operasional, historis, dan eksternal, yang dibersihkan, diubah, dan dikatalogkan untuk penelusuran dan analisis untuk pengambilan keputusan bisnis.

Menurut Hsiang-Yuan(2007) *data warehouse* adalah solusi untuk mengonvergensikan data dari database yang terdistribusi dan sumber data.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, *data warehouse* adalah tempat penyimpanan data yang berorientasi pada subjek, terintegrasi, tidak mudah berubah, dan memiliki rentang waktu, yang diambil dari *database* operasional, historis, dan eksternal, yang diproses agar dapat dianalisis untuk mendukung proses pengambilan keputusan.

2.10 Karakteristik *Data Warehouse*

Karakteristik *data warehouse* menurut Inmon, yaitu

1. *Subject Oriented* (Berorientasi subjek)

Data warehouse berorientasi *subject* artinya *data warehouse* didesain untuk menganalisis data berdasarkan subjek-subjek 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 :

Tabel 2.1 Perbandingan antara sistem OLTP dan sistem *Data Warehouse*

(Connolly dan Begg, 2010, p1153)

Data Operasional (OLTP)	***Data Warehouse***
Mengandung data terkini	Mengandung data historis
Menyimpan data yang detail	Menyimpan data rinci, sedang, ringkas
Data bersifat dinamis	Data bersifat statis
Prosesnya berulang	Proses tidak terstruktur, tergantung tujuan
Digunakan untuk transaksi	Digunakan untuk analisis
Jumlah transaksi tinggi	Jumlah transaksi sedang dan kecil
Berorientasi pada aplikasi	Berorientasi pada subjek
Penggunaan bisa diprediksi	Penggunaan tidak bisa diprediksi
Mendukung keputusan harian	Mendukung keputusan strategis
Digunakan oleh user operasional	Digunakan oleh user manajerial

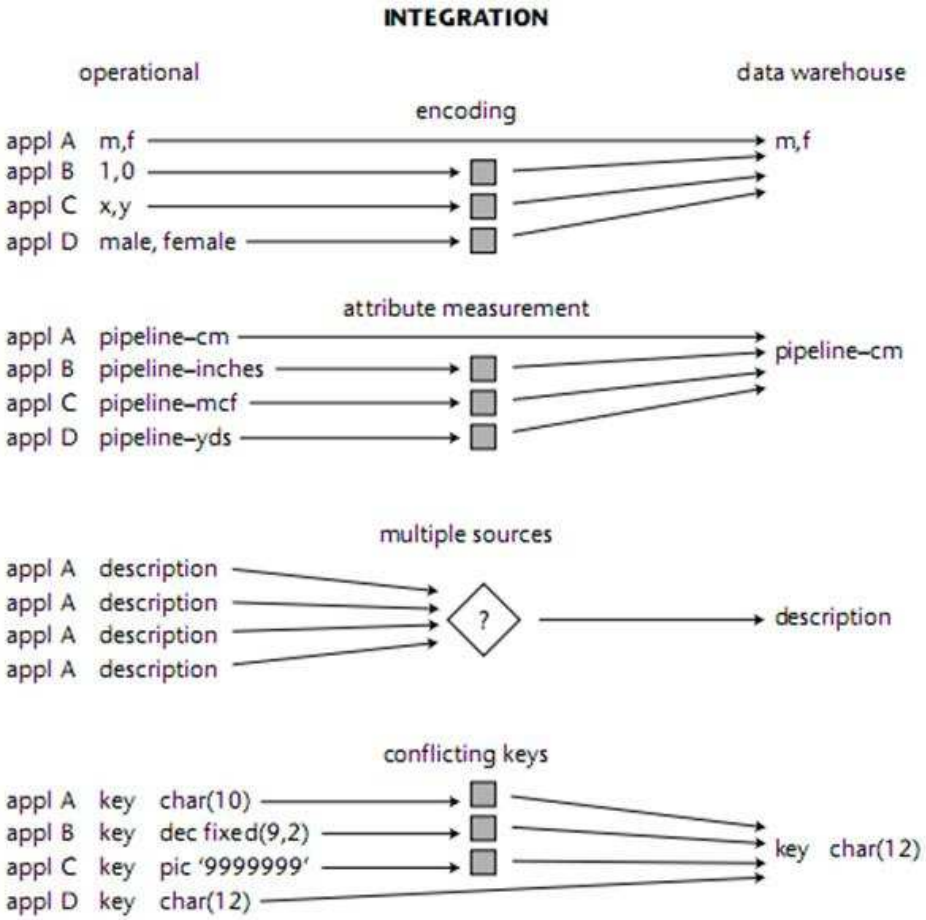
2. *Integrated* (Terintegrasi)

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 sepeti 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 kekonsistenannya.



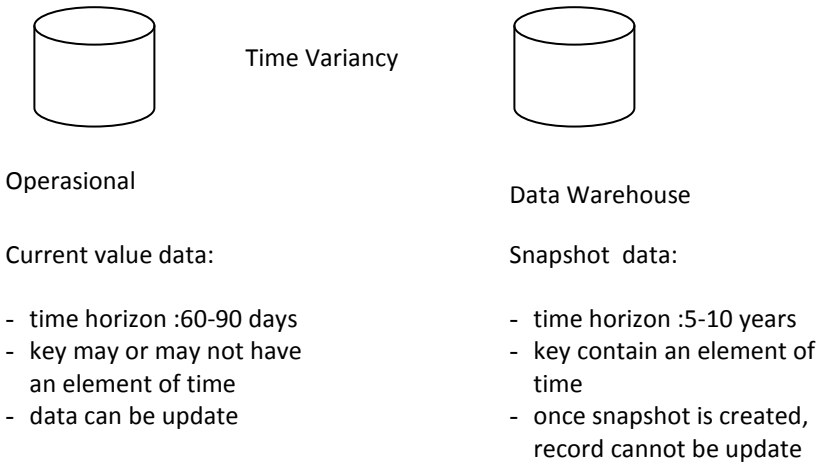
Gambar 2.1 Integrasi *Data Warehouse*

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*.

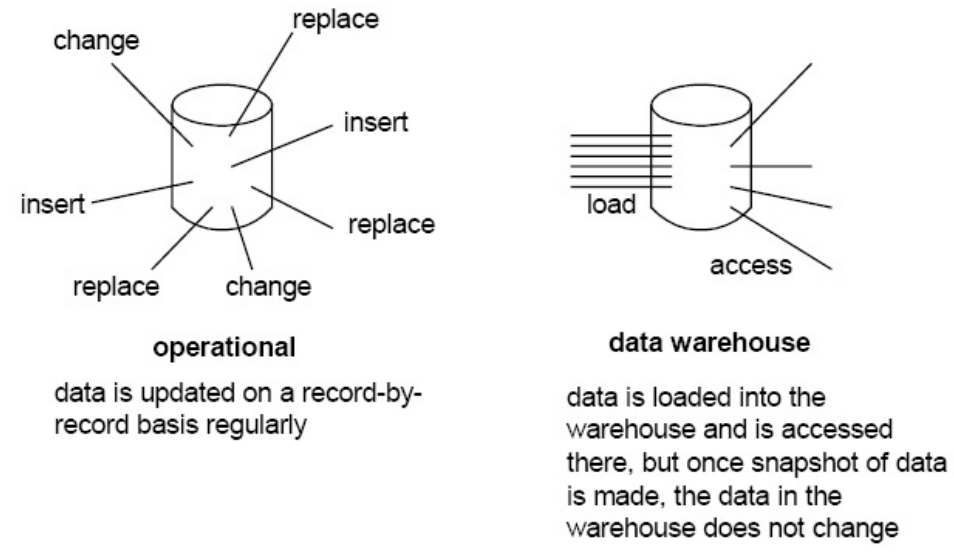


Gambar 2.2 *Time Variance Data Warehouse*

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 menampilan laporan yang dibutuhkan, tidak ada kegiatan *updating* data).



Gambar 2.3 *Non Volatile Data Warehouse*

2.11 Pengertian *Data Mart*

Menurut Inmon (2005, p494), *data mart* adalah struktur data per departemen yang diperoleh dari *data warehouse* di mana data tersebut didenormalisasi berdasarkan kebutuhan informasi departemen.

Menurut Connolly dan Begg (2010, p1214), *data mart* adalah database yang berisi data perusahaan subjek untuk mendukung persyaratan analitis dari unit bisnis tertentu.

Menurut Huisman (2009), Sebuah *data mart* adalah sub bagian dari *data warehouse* yang terkait dengan proses bisnis yang spesifik.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, *data mart* adalah bagian dari *data warehouse* yang telah didenormalisasi berdasarkan kebutuhan informasi manajemen.

2.12 Keuntungan dan Kegunaan *Data Warehouse*

Menurut Connolly dan Begg (2010, p1198), *data warehouse* yang telah diimplementasikan dengan baik dapat membawa keuntungan besar bagi perusahaan antara lain :

1. Pengembalian yang besar dari investasi yang ada.
2. Keuntungan kompetitif.
3. Meningkatkan produktivitas para pembuat keputusan.

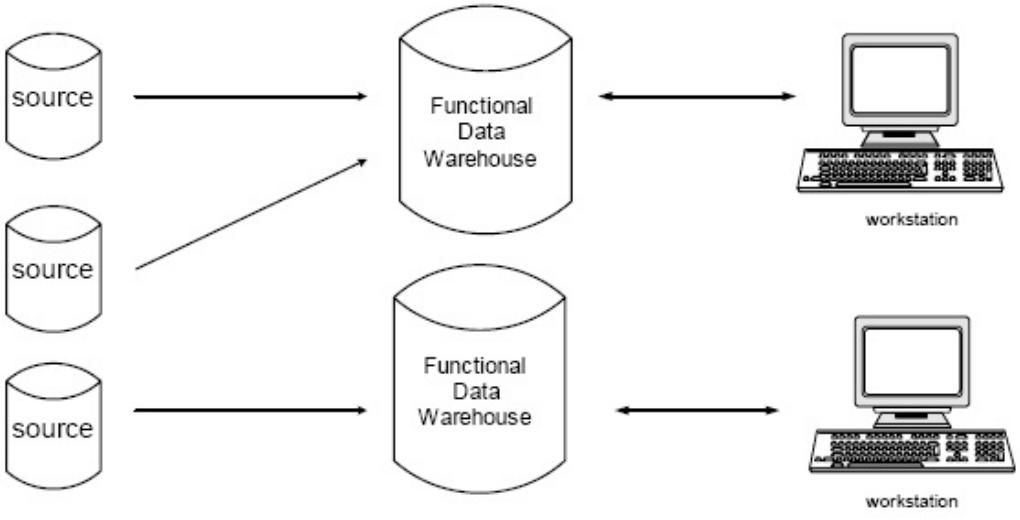
2.13 Bentuk *Data Warehouse*

Bentuk dari *data warehouse* yaitu :

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.



Gambar 2.4 Bentuk *Data Warehouse* fungsional

2. *Centralized Datawarehouse (*Data Warehouse Terpusat)

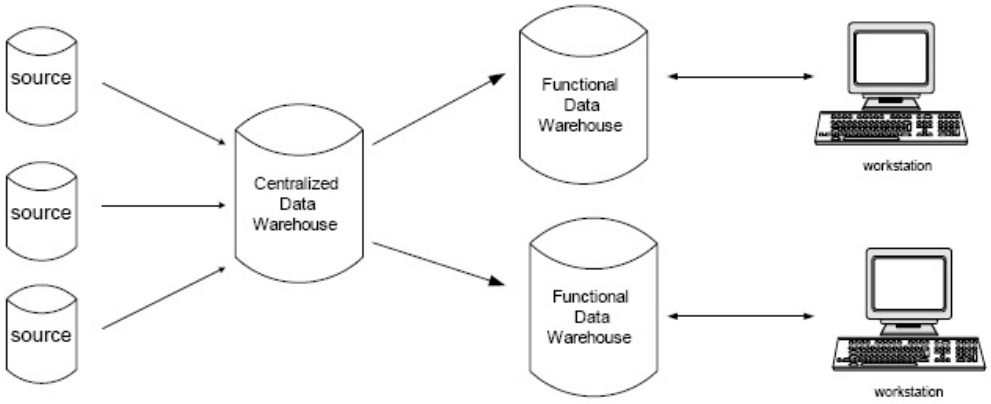
Bentuk ini terlihat seperti bentuk *data warehouse* fungsional, namun terlebih dahulu sumber data dikumpulkan dalam satu tempat terpusat, kemudian data disebar ke dalam fungsinya masing-masing, sesuai kebutuhan perusahaan. *Data warehouse* terpusat ini, biasa digunakan oleh perusahaan yang belum memiliki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

Keuntungan:

- Benar-benar terpadu karena konsistensinya yang tinggi.

Kerugiannya :

- Biaya yang mahal serta memerlukan waktu yang lama untuk membangunnya.



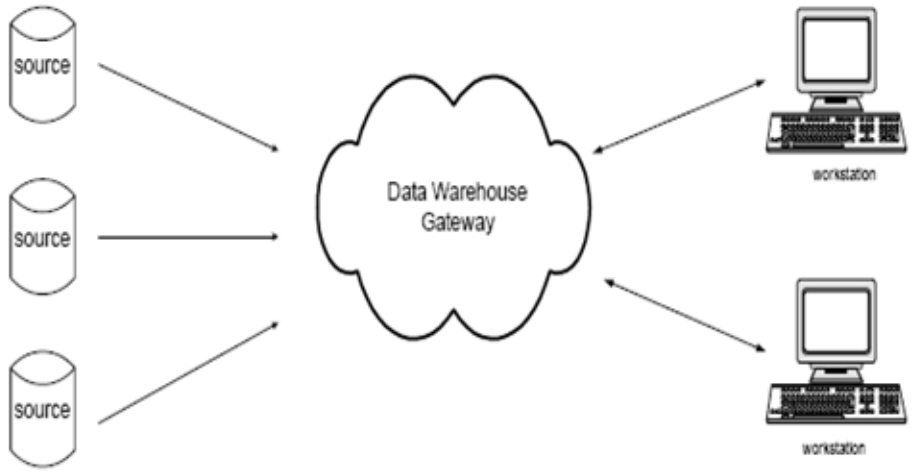
Gambar 2.5 Bentuk *Data Warehouse* terpusat

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 biaya nya yang paling mahal dibandingkan dua bentuk *data warehouse* lainnya.



Gambar 2.6 Bentuk *Data Warehouse* terdistribusi

2.14 Arsitekur *Data Warehouse*

Menurut Connoly dan Begg(2010, P1156), Karakterisitik arsitektur *data warehouse*:

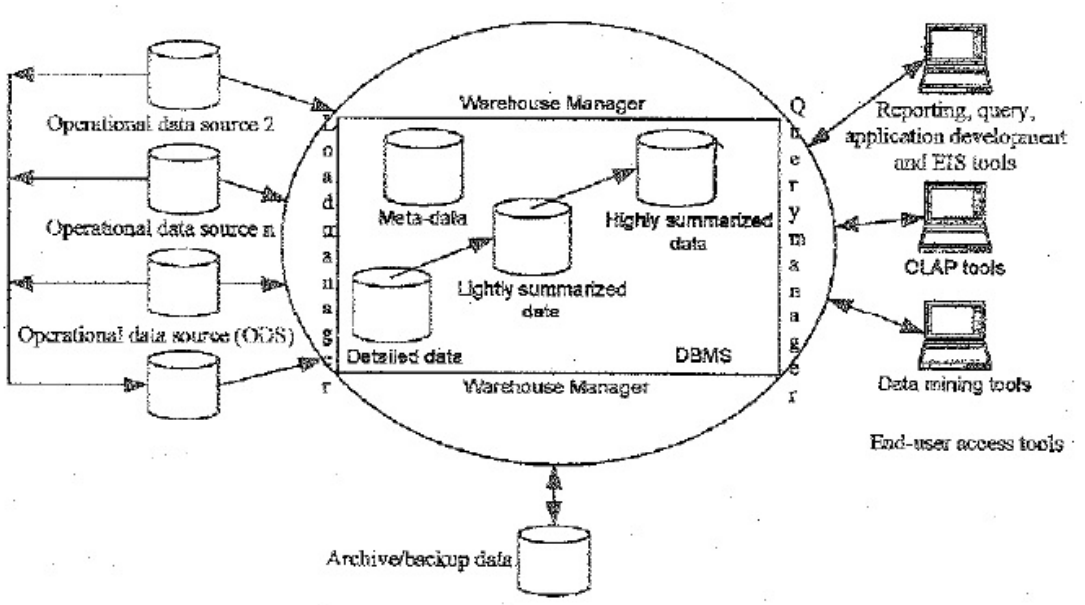
1. Data diambil dari sistem asal (sistem informasi yang ada), *database* dan file.

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 :



Gambar 2.7 :Arsitektur dan Komponen Utama *Data Warehouse*

Sumber : Conolly,T.M.,Begg (2010, p1157)

a. *Operational Data*

Sumber data dari *data warehouse* dapat diambil langsung dari *mainframe*, basis data relasional seperti Oracle, Ms SQL server dan sebagainya.

b. *Operational Data Store*

Operasional *Data Store* menampung data yang diekstrak dari sistem utama atau sumber-sumber data yang ada dan kemudian data hasil ekstrasi tersebut dibersihkan.

c. ETL *manager*

ETL *manager* melakukan semua operasi yang berhubungan dengan ETL data ke dalam gudang. data dapat diambil langsung dari sumber data atau lebih umum, dari toko data operasional.

d. *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.

e. *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.

f. *Detailed Data*

Dalam *data warehouse*, *detailed data* menyimpan semua detail dari data di dalam skema *database*. Biasanya *detailed data* tidak disimpan secara *online* melainkan dibuat dengan melakukan agregasi data. Tetapi pada dasarnya, *detailed data* ditambahkan ke *warehouse* untuk melengkapi data *aggregate*. *Detailed data* dibagi menjadi dua, yaitu *current detailed data* (tempat penyimpanan *detailed data* saat ini) dan *old detailed data* (tempat penyimpanan *detailed* histori).

g. *Lightly and Highly Summarized Data*

Dalam data *warehouse*, *lightly* and *highly summarized data* adalah tempat penyimpanan semua *data predefined lightly* dan *highly summarized* yang dihasilkan oleh *Warehouse Manager*. Tujuan dari ringkasan informasi ini adalah mempercepat tanggapan terhadap permintaan *user*. Ringkasan data di-*update* terus-menerus seiring dengan bertambahnya jumlah data dalam *data warehouse*.

h. *Archieve / Backup Data*

Dalam *data warehouse*, *archieve / backup data* digunakan sebagai tempat penyimpanan *detailed data* dan data yang telah diringkas. Data yang telah diringkas dan disimpan akan ditransfer ke media penyimpanan seperti *magnetic tape* dan *optical disc*.

i. *Metadata*

Dalam *data warehouse*, *metadata* digunakan sebagai tempat penyimpanan semua definisi metadata(keterangan mengenai data) yang digunakan di seluruh proses *data warehouse*. *Metadata* bertujuan untuk:

1. Proses *extracting* dan *loading*

   Melakukan pemetaan sumber data pada *data warehouse*.

2. Proses *Warehouse Management*

   Mengotomatiskan produksi pada tabel-tabel produksi.

3. Sebagian proses *Query Management*

   Mengarahkan permintaan ke sumber data yang tepat.

j. *End-user Access Tools (EUAT)*

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.* End-User Access Tools dikelompokkan menjadi 5 golongan:

1. *Reporting* dan *Query Tools*

Reporting tools meliputi *production reporting tools* dan *report writers*. *Production repoting tools* digunakan untuk menghasilkan laporan

operasional biasa, sedangkan *report writer* adalah *desktop tools* yang dirancang untuk *end user*.

Query Tools untuk *data warehouse* relasional dirancang untuk menerima SQL atau menghasilkan pernyataan SQL untuk *query* data yang disimpan dalam *data warehouse*.

2. *Application Development Tools*

Aplikasi yang dapat digunakan *user* yaitu *graphical data access* yang dirancang untuk sisi *client server*. Beberapa *application development tools* terintegrasi dengan OLAP *tools* dan dapat mengakses semua sistem basis data utama, mencakup Oracle, Sybase dan Infomix.

3. *Executive Information System (EIS) Tools*

Executive Information System (EIS) *tools* semula dikembangkan untuk mendukung pengambilan keputusan tingkat tinggi kemudian meluas untuk mendukung semua tingkat manajemen. EIS tools yang terisolasi dengan *mainframe* memungkinkan *user* membuat aplikasi pendukung pengambilan keputusan untuk menyediakan *overview* data dan mengakses sumber data eksternal. Saat ini perbedaan antara EIS dan *decision support tools* lainnya semakin tidak jelas, sejak EIS menyediakan aplikasi *custom build* untuk area bisnis seperti penjualan, *marketing*, dan keuangan.

4. *Online Analysis Processing (OLAP) Tools*

Online Analytical Processing (OLAP) *tools* berbasis pada konsep *database* multidimensional dan memungkinkan pengguna untuk menganalisis data menggunakan *view* yang kompleks dan multidimensional. Tools ini mengasumsikan data diatur dalam model multidimensi yang

didukung special *multidimensional database* (MDDB) atau basis data relasional yang dirancang untuk mendapatkan multidimensional *query*.

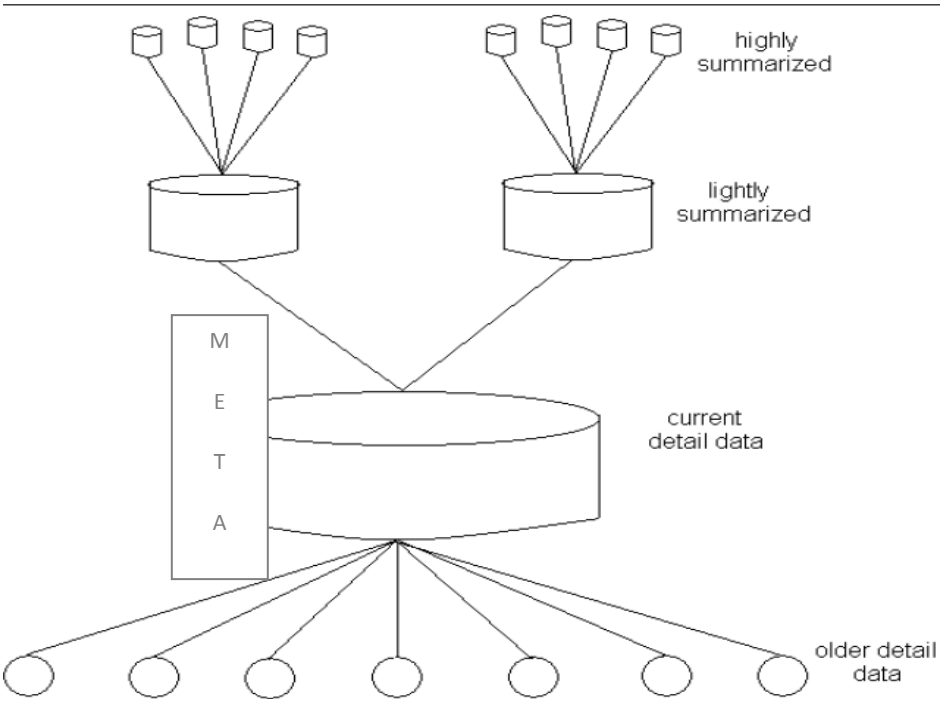
5. *Data Mining Tools*

Data Mining adalah proses menemukan korelasi, pola dan arah baru yang berarti ‘menambang’ sejumlah besar data dengan menggunakan teknik statistik, matematika dan *artificial intelligence*. *Data mining* berpotensi mengganti kemampuan dari OLAP *tools*.

2.15 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 Inmon(2005,p33) data mengalir dari lingkungan opersional ke dalam lingkungan *data warehouse* dimana data mengalami transformasi dari tingkatan opersional ke tingkatan *data warehouse*. Pada perumusan data dapat dilihat di gambar, data disampaikan dari *current detail* ke *older detail*. Setelah data diringkas,data tersebut disampaikan dari *current detail* ke *lightly summarized data*,kemudian dari *lightly summarized* data ke *highly summarized data*.



Gambar 2.8 Struktur *Data Warehouse*

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 memilki 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.

- ***Lighlty 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 detil 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. *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*, *matrics*, *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 *Decission 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 *hightly summaried data*.

- *Mapping*

  Sebagai panduan pemetaan(*mapping)* data pada saat data di *transform*/diubah dari lingkup operasional menjadi lingkup *data warehouse*.

2.16 Aliran Data dalam *Data Warehouse*

Menurut Connolly dan Begg (2005, p1161), *data warehouse* fokus pada manajemen lima arus data primer, yaitu :

- ***Inflow***

  Proses yang berhubungan dengan *extraction*, *transformation*, dan *loading* (ETL) data dari sumber data ke dalam *data warehouse*. Proses *inflow* berperan dalam proses pengambilan data dari sumber sistem dan memasukkannya ke dalam *datawarehouse*. Cara lainnya yaitu dengan memasukkan data ke dalam *operational data store* (ODS) sebelum dikirim ke *data warehouse*. Proses rekonstruksi dari data meliputi :

  - Membersihkan data yang kotor.
  - Restrukturisasi data yang dicocokkan dengan kebutuhan *data warehouse*, contohnya menambah atau membuang *field-field*.
  - Memastikan sumber data konsisten dengan dirinya sendiri dan data lainnya yang sudah ada di dalam *data warehouse*.

- ***Upflow***

Proses penambahan nilai ke data di dalam *data warehouse* melalui proses meringkas, mengemas, dan menyalurkan data. Aktivitas yang berhubungan dengan proses *Upflow* meliputi :

- Meringkas data dengan proses memilih, memperhitungkan, menggabungkan, dan mengelompokkan data relasional ke dalam tampilan yang lebih baik dan berguna bagi *user*.
- Pengepakan data dengan mengubah data yang detail ke dalam format yang lebih berguna seperti *spread sheets*, teks dokumen, diagram, grafik, *database* pribadi dan animasi.
- Mendistribusikan data ke kelompok-kelompok yang tepat untuk meningkatkan ketersediaan agar dapat diakses.

- ***Downflow***

Proses pengarsipan dan *backup* data di dalam *data warehouse*. Menyimpan data lama memainkan peranan yang penting dalam mempertahankan penampilan dan efektifitas dari *warehouse* dengan mengirimkan data lama dengan nilai yang terbatas ke sebuah tempat penyimpanan seperti *magnetic tape* atau *optical disk*.

Downflow dari data juga meliputi proses yang memastikan bahwa kondisi sekarang dari *data warehouse* dapat dibangun kembali juka terjadi kehilangan data, kegagalan *software* atau *hardware*.

- ***Outflow***

Proses untuk membuat data tersedia untuk *user*. Dua aktivitas yang dilakukan adalah pengaksesan dan pengiriman data, yaitu :

a. Pengaksesan, berhubungan dengan proses memuaskan *end-user* dengan menyediakan data yang dibutuhkan. Frekuensi dari pengaksesan ini bervariasi mulai dari *ad hoc* secara rutin sampai *real time*. Selain itu, dipastikan bahwa sumber sistem digunakan dengan cara yang paling efektif di dalam penjadwalan pengeksekusian terhadap *query* dari *user*.
b. Pengiriman berhubungan secara aktif dalam pengiriman informasi ke *work station* dari *user*. Ini merupakan area baru dari *data warehouse* dan sering dihubungkan dengan proses *publish* dan *subsciribe*. *Data warehouse* akan mempublikasi objek bisnis yang bermacam-macam dan *user* akan berlangganan terhadap objek bisnis yang dibutuhkan oleh mereka.

- ***Metaflow***

Proses yang berhubungan dengan mengatur *metadata*. *Metaflow* merupakan proses memindahkan *metadata* (data tentang *flow* yang lainnya). *Metadata* merupakan deskripsi dari data yang ditampung di dalam *data warehouse*, apa yang ada di dalamnya, dari mana asalnya, dan apa yang telah dilakukan terhadap data tersebut dengan cara *cleansing, integrating*, dan *summarizing.*

2.17 *ETL(Extraction, Transformation, Loading)*

Menurut Inmon (2005,p497) ETL adalah proses melakukan pencarian data, mengintegrasikan dan menempatkan data ke dalam sebuah *data warehouse*.

Menurut James O'Brien (2005,p38),ETL adalah mengumpulkan, menyaring, mengolah, dan menggabungkan data-data yang relevan dari berbagi sumber untuk disimpan ke dalam data *warehouse.*

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, ETL adalah proses menyiapkan data yang meliputi pencarian data, pengintegrasian data dan penempatan data dari *operational source* ke dalam *data warehouse.*

Proses ETL ini terdiri dari 3 tahap,yaitu:

1. *Extraction*

Langkah pertama dari proses ETL adalah proses penarikan data dari satu atau lebih sistem operasional sebagai sumber data (misalnya diambil dari sistem OLTP). Kebanyakan proyek *data warehouse* menggabungkan data dari sumber-sumber yang berbeda. Proses ektraksi ini merupakan proses penguraian dan pembersihan data yang diekstrak untuk mendapatkan pola atau struktur data yang diinginkan.

2. *Transformation*

Proses membersihkan data yang telah diambil pada proses *extract* tersebut dilakukan agar data tersebut sesuai dengan struktur *data warehouse*.

Hal-hal yang dapat dilakukan dalam tahap transformasi:

a. Hanya memilih kolom tertentu saja untuk dimasukkan ke dalam *data warehouse.*

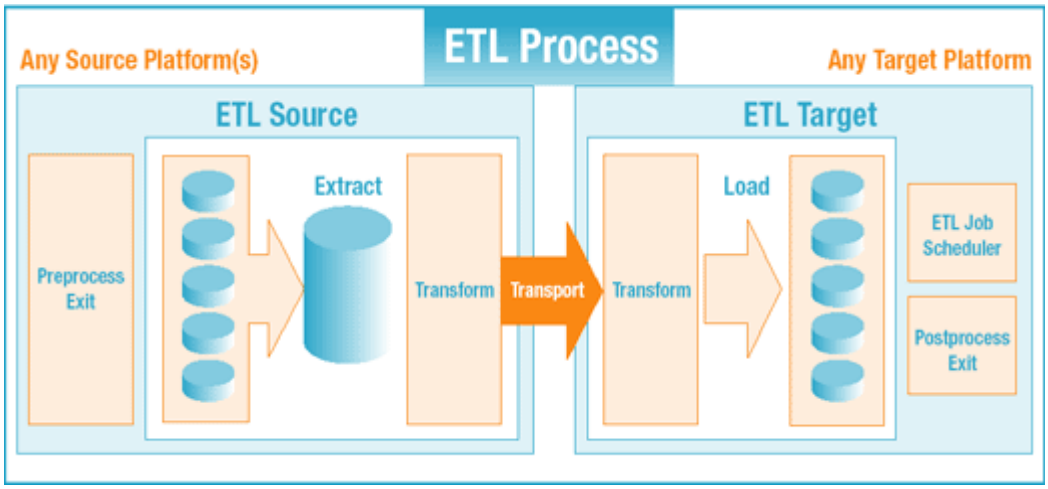
b. Menterjemahkan nilai berupa kode(misalnya saja, *database* sumber menyimpan nilai 1 untuk pria dan 2 untuk wanita tetapi dalam data *warehouse* menyimpan M untuk pria dan F untuk wanita).Proses yang dilakukan tersebut disebut *automated* atau *cleansing*. Tidak ada pembersihan manual selama proses ETL.
c. Mengkodekan nilai-nilai ke dalam bentuk bebas(misalnya memetakan "male","1", dan "Mr" ke dalam "M").
d. Melakukan perhitungan nilai-nilai baru(misalnya sale_amount=qty*price).
e. Menggabungkan data dari berbagai sumber bersama-sama.
f. Membuat ringkasan dari sekumpulan baris data(misalnya, total penjualan untuk setiap bagian).

Kesulitan yang terjadi dalam proses transformasi adalah:

a. Data harus digabungkan dari beberapa sistem yang terpisah.
b. Data harus dibersihkan sehingga konsisten.
c. Data harus diagregasi untuk mempercepat analisis.

3. *Loading*

Merupakan tahap akhir dalm proses ETL, yaitu proses memasukkan data ke dalam target akhir, dalam hal ini adalah *data warehouse*. Data berasal dari proses transformasi. Setelah data yang dihasilkan dari proses transformasi sesuai dengan kondisi yang diinginkan di data warehouse maka proses *loading* akan berjalan. Data dari *staging area* akan dipindahkan ke dalam *data warehouse*



Gambar 2.9 Proses ETL

Sumber :http://www.iwaysoftware.com/etl-tools.html

2.18 Metodologi Perancangan *Database* untuk *Data Warehouse*

Menurut Kimball (Connolly dan Begg, 2005, p1187), ada sembilan tahap metodologi dalam perancangan *database* untuk *data warehouse*, yaitu :

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.*

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.

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.

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.

Langkah 5 : Menyimpan pre-kalkulasi di tabel fakta

- Hal ini terjadi apabila fakta kehilangan *statement*.

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.

Langkah 7 : Pemilihan durasi *database*

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

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.

Langkah 9 : Menentukan prioritas dan mode *query*

- Pada tahap ini kita menggunakan perancangan fisik.

2.19 Model untuk *Data Warehouse*

Berikut di bawah ini adalah penjelasan dari model untuk *data warehouse*:

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.

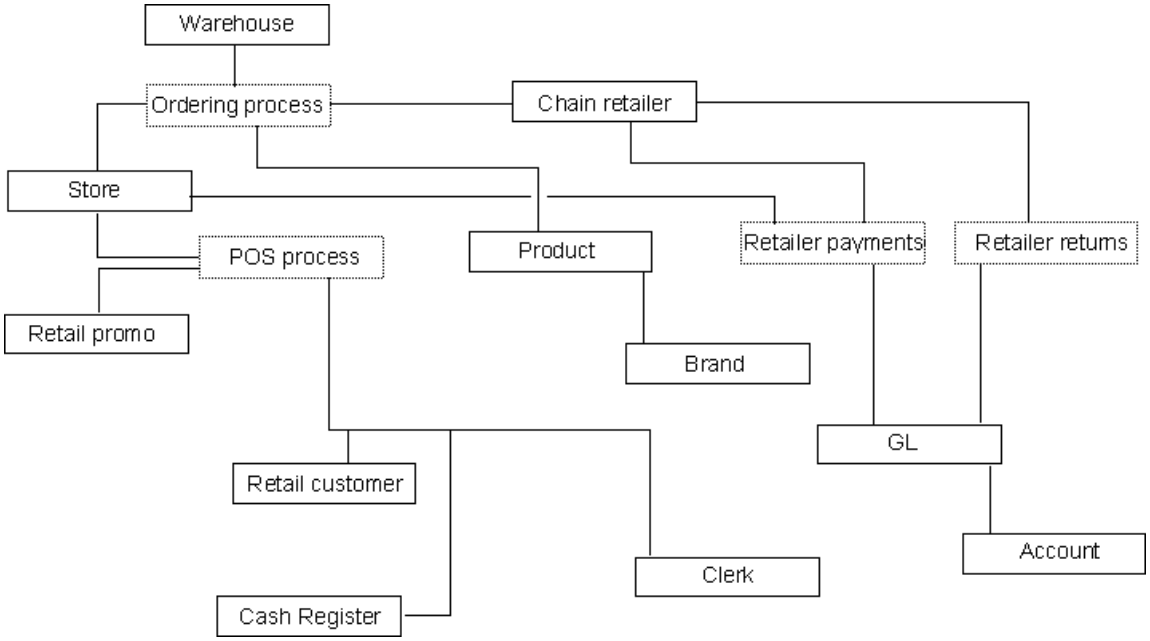
Model dimensional menggunakan konsep model hubungan antar *entity* (ER) dengan beberapa batasan yang penting. Setiap model dimensi terdiri dari sebuah

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 tabel fakta. Dengan kata lain *primary key* pada tabel fakta terdiri dari dua atau lebih *foreign key*. Struktur karakteristik ini disebut dengan skema bintang atau *join* bintang.

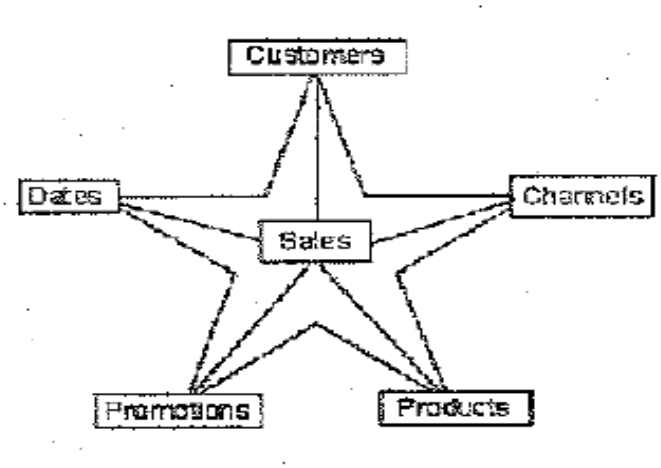
Fitur terpenting dalam *model dimensional* ini adalah semua *natural keys* diganti dengan kunci pengganti(*surrogate keys).* Maksudnya yaitu setiap kali join antar *table* fakta dengan *table* dimensi selalu didasari kunci pengganti. Kegunaan dari kunci pengganti adalah memperbolehkan data pada *data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

Sebuah sistem OLTP memerlukan normalisasi untuk mengurangi redudansi, 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 tabel *data warehouse* :



Gambar 2.10 Model data OLTP

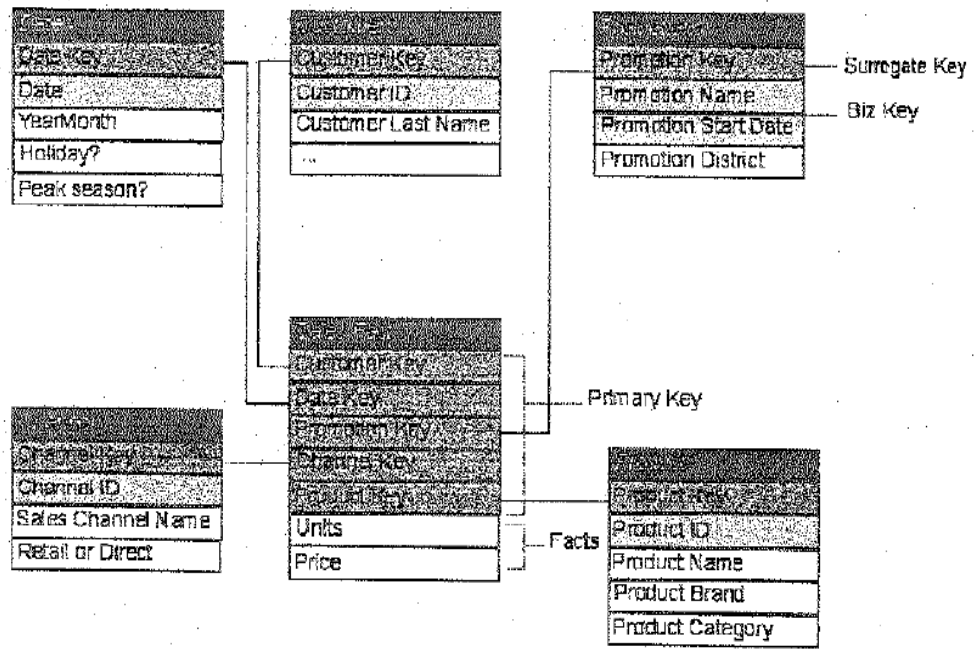


Gambar 2.11 Dimension Model

2. *Schema* Bintang

Menurut Inmon (2005,P126), skema bintang memberikan beberapa keuntungan yang tidak terdapat pada struktur relasional yang biasa. Skema bintang merupakan standar rancangan *warehouse* :

- Membentuk rancangan *database* yang memberikan waktu respon yang cepat.
- Menghasilkan rancangan yang dapat dimodifikasi dengan mudah atau ditambahkan sesuai dengan perkembangan dan pertumbuhan *data warehouse.*
- Paralel dalam rancangan *database*, bagaimana *user* bisa memandang dan menggunakan data.
- Mempermudah pemahaman dan navigasi *metadata* baik untuk perancangan maupun pemakai.
- Memperluas pilihan *software* akses data (*front end data access tools*), sebagai beberapa produk yang memerlukan rancangan skema bintang.



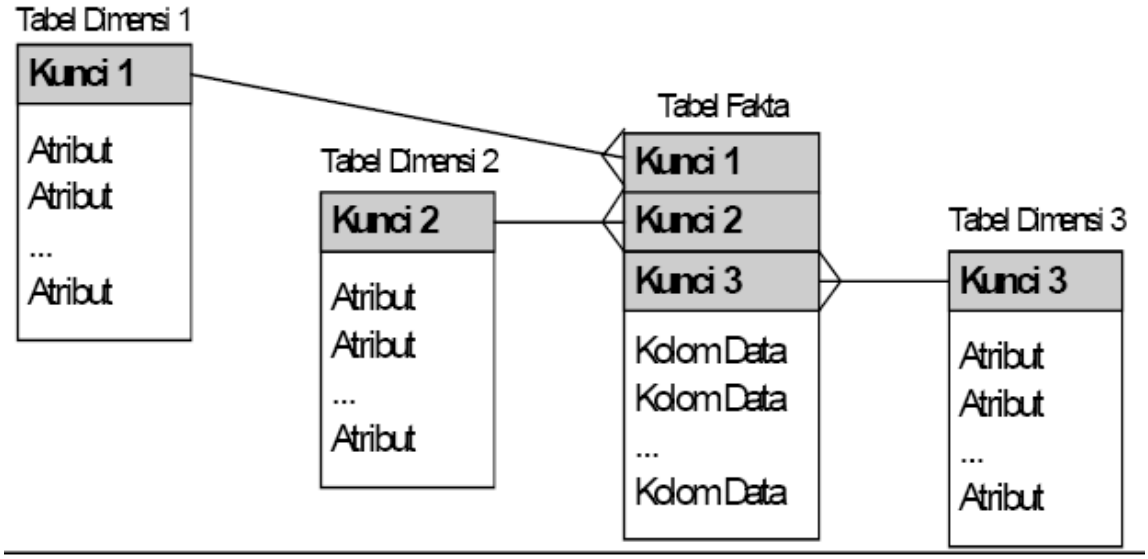
Gambar 2.12 Contoh Skema Bintang

Jenis-jenis Skema Bintang, yaitu :

a. Skema Bintang Sederhana

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

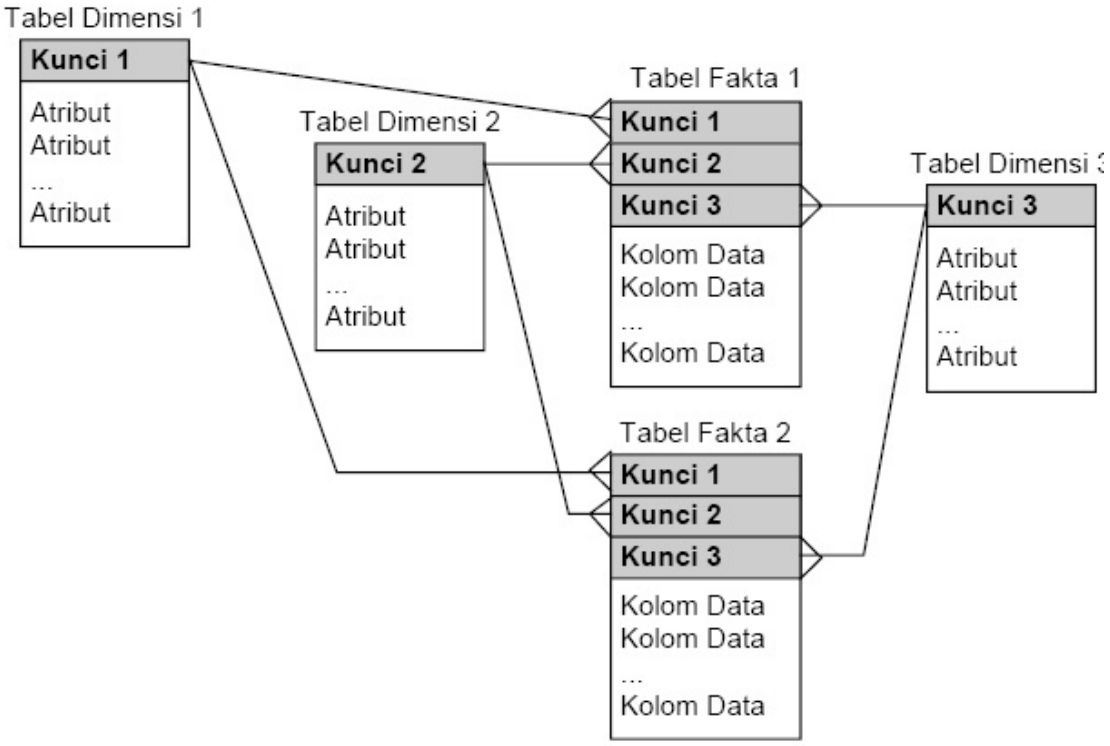
Primary key dari table fakta terdiri dari satu atau lebih *foreign key. Foreign key* merupakan *primary key* pada table lain.



Gambar 2.13 Skema Bintang Sederhana

b. 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.



Gambar 2.14 Skema Bintang Dengan Banyak Tabel Fakta

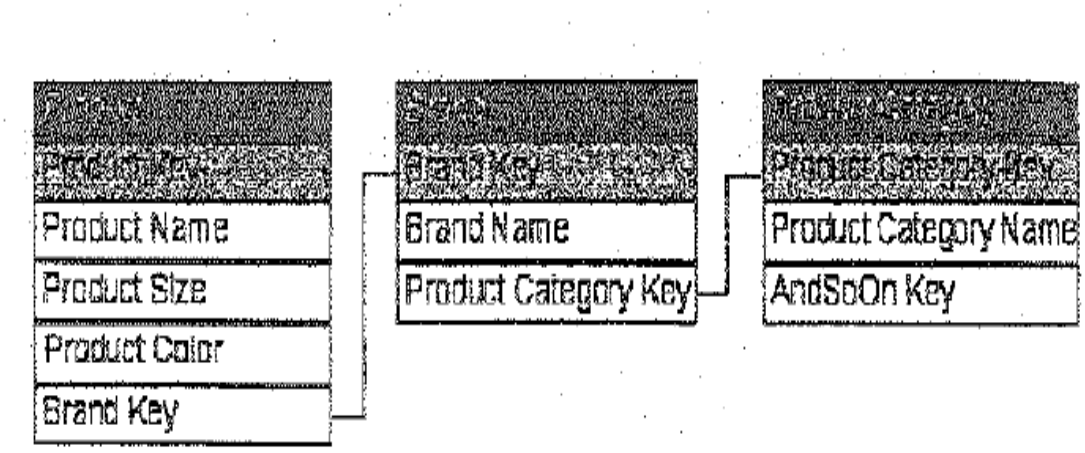
Adapun ketentuan dalam pembacaan skema bintang adalah :

- 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.

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 (snowflake)* seperti contoh dibawah ini :



Gambar 2.15 *Snowflake Schema*

4. *Star* atau *Snowflake*

Keduanya merupakan model-model dimensional, perbedaannya terletak pada implementasi fisikal. 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.

5. *Fact Table*(Tabel Fakta)

Menurut Inmon(2005,p497), *table* fakta adalah tabel pusat dari skema bintang dimana data sering muncul akan ditempatkan di tabel tersebut. Tabel fakta disebut juga tabel utama(*major table*), merupakan inti dari skema bintang dan berisi data *actual* yang akan dianalisis(data kuantitatif dan transaksi). Tabel fakta adalah tabel yang umumnya mengandung angka dan data historis dimana *key*(kunci) yang dihasilkan sangat unik karena merupakan kumpulan *foreign key* dan *primary key* yang ada pada masing-masing tabel dimensi yang berhubungan atau merupakan tabel terpusat dari skema bintang. Tabel fakta menyimpan tipe-tipe *measure* yang berbeda, seperti *measure* yang secara langsung berhubungan dengan tabel dimensi dan *measure* yang tidak berhubungan dengan tabel dimensi.

Menurut Connoly dan Begg(2005,p1183), tabel fakta adalah tabel pada dimensional model yang isinya *composite primary key*(PK). Jadi *primary key* pada tabel fakta merupakan beberapa *foreign key*.

6. *Dimensional Table*(Tabel Dimensi)

Menurut Inmon(2005,p495), tabel dimensi adalah tempat dimana data-data yang tidak berhubungan yang berelasi dengan tabel fakta ditempatkan di dalam tabel dimensional.

Tabel dimensi juga disebut tabel kecil(*minor table*), karena memegang data deskriptif yang mencerminkan dimensi suatu bisnis. Tabel dimensi merupakan tabel yang berisi kategori dengan ringakasan data detail yang dapat dilaporkan, seperti laporan keuntungan pada tabel fakta dapat dilaporkan sebagai dimensi waktu(berupa per bulan atau per tahun).

2.20 Pengertian *Granularity*

Menurut Inmon (2005,p43) *granularity* merupakan suatu level dari detil atau ringkasan pada unit data di dalam data *warehouse*. Semakin banyak detil atau ringkasan pada unit data maka akan semakin rendah level pada *granularity*.

Contohnya adalah sebuah transaksi yang sederhana akan berada pada tingkat *granularity* yang rendah, sedangkan keseluruhan dari transaksi dalam satu bulan akan berada pada level *granularity* yang lebih tinggi. *Granularity* merupakan permasalahan utama dalam mendesain lingkungan pada *data warehouse* karena berpengaruh besar pada *volume* dari data yang terletak di dalam *data warehouse*.

Keuntungan-keuntungan *granularity*:

- Dapat digunakan kembali

  Dikatakan dapat digunakan kembali karena dapat digunakan oleh banyak orang dengan cara-cara yang berbeda. Contohnya data yang sama dapat digunakan untuk memenuhi kebutuhan dalam bidang pemasaran, penjualan

dan keuangan. Pemasaran menginginkan melihat data bulanan berdasarkan area geografi, dan keuangan menginginkan melihat pendapatan setiap kuarter berdasarkan produk.

- Kemampuan untuk mencocokkan data

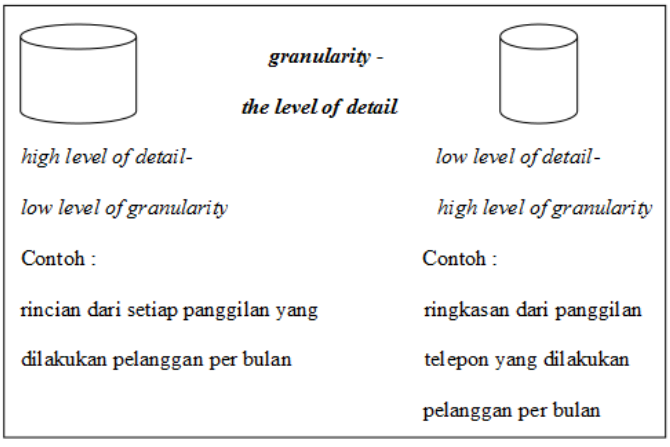
  Jika memiliki satu dasar yang sama untuk semuanya, maka jika terjadi perbedaan dalam analisis antara dua atau lebih departemen, proses pencocokan akan menjadi lebih sederhana dan mudah.

- Fleksibel

  Dimana para user dapat merubah data sesuai dengan tampilan yang mereka inginkan sehingga pekerjaan dapat diselesaikan dengan baik dan mudah.

- Kebutuhan yang tidak jelas yang akan datang dapat diakomodasi.

  Saat ada kebutuhan yang baru dan ada kebutuhan informasi, *data warehouse* sudah siap untuk melakukan analisis dan organisasi disiapkan untuk menangani kebutuhan yang baru.



Gambar 2.16 *Granularity*

2.21 Teori Khusus

2.21.1 Pengertian Penjualan

Menurut Kotler (2006,P457), penjualan merupakan sebuah proses dimana kebutuhan pembeli dan kebutuhan penjual dipenuhi, melalui antar pertukaran informasi dan kepentingan.

Menurut Reeve (2009,P255), penjualan adalah total biaya yang dibebankan kepada *customer* untuk barang yang dijual termasuk penjualan tunai maupun penjualan kredit.

2.21.2 Pengertian Produksi

Menurut Groover (2005,P1), produksi merupakan suatu kumpulan orang,peralatan, dan aturan-aturan yang dikelola sedemikian rupa untuk melaksanakan operasi-operasi manufaktur dalam sebuah pabrik.

Menurut Nasution (2003,P1) produksi adalah metode dan teknik yang digunakan dalam mengolah bahan baku menjadi suatu produk jadi atau setengah jadi.

2.21.3 Pengertian Inventori / Persediaan

Menurut Alfredson, K., et al (2007,P342) persediaan adalah *asset* yang tersedia untuk dijual dalam proses bisnis, *asset* yang ada dalam proses

produksi seperti untuk dijual, *asset* dalam bentuk material atau *supplier* untuk digunakan dalam proses produksi.

Menurut Reeve (2005,P355) persediaan digunakan untuk menunjukkan barang dagangan yang dimiliki untuk dijual dalam kegiatan usaha normal. bahan dalam proses produksi atau yang dimiliki bagian produksi.