BAB 2

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Pengertian data

Menurut Laudon (2006, p13), data adalah kumpulan fakta yang masih mentah yang menjelaskan aktivitas – aktivitas yang terjadi dalam organisasi atau lingkungan fisik, sebelum terorganisir dan diubah menjadi bentuk yang dimengerti dan dapat digunakan.

Menurut W.H Inmon (2002, p388), data adalah sebuah rekaman fakta, konsep atau instruksi pada sebuah media penyimpanan untuk komunikasi, pencarian dan pemrosesan secara otomatis dan dapat memberikan informasi yang dapat dimengerti oleh pemiliknya atau pihak yang bersangkutan.

Dari teori – teori diatas maka dapat disimpulkan bahwa data adalah kumpulan fakta yang masih mentah dan tersimpan dalam media penyimpanan yang setelah diproses akan menghasilkan informasi yang dapat dimengerti dan dapat digunakan oleh pihak yang bersangkutan.

2.1.2 Pengertian Informasi

Menurut Mc.Leod (2007, p9), informasi adalah data yang sudah diproses dan memiliki arti, biasanya memberitahu pemakai sesuatu yang belum mereka ketahui. Karakteristik penting yang harus dimiliki oleh informasi adalah :

a. Relevansi

Informasi tersebut berhubungan dengan keputusan yang akan diambil dalam usaha mencapai tujuan yang telah ditetapkan.

b. Akurat

Informasi dapat diandalkan dan disajikan secara tepat.

c. Tepat Waktu

Informasi harus dapat diterima oleh penerima, tidak boleh terlambat karena informasi yang terlambat menjadi tidak bernilai.

d. Kelengkapan

Informasi harus mampu menyajikan gambaran lengkap dari suatu permasalahan atau penyelesaian.

Menurut O'Brien (2003, p13), informasi adalah data yang telah dikonversikan ke dalam konteks yang penuh arti dan berguna bagi pengguna tertentu.

Berdasarkan pendapat ahli diatas maka dapat disimpulkan bahwa informasi adalah data yang telah diproses atau dikonversikan ke dalam bentuk yang penuh arti dan berguna baik bagi manusia secara umum maupun bagi pengguna tertentu.

2.1.3 Pengertian *Database*

Menurut Connolly dan Begg (2005, p15) *database* adalah kumpulan data yang berhubungan satu sama lain yang digunakan secara bersama-sama, dan kumpulan data ini didesain untuk memenuhi kebutuhan informasi suatu perusahaan.

Menurut W.H Inmon (2005, p493), *database* merupakan koleksi data – data yang saling berhubungan yang tersimpan (biasanya dengan *redundancy* yang terkendali dan terbatas) berdasarkan suatu skema tertentu. *Database* dapat digunakan untuk aplikasi tunggal ataupun berganda.

Jadi, dapat disimpulkan *database* adalah kumpulan data yang saling berhubungan, tersimpan berdasarkan skema tertentu, dan digunakan untuk memenuhi kebutuhan informasi suatu perusahaan.

2.1.4 Pengertian DBMS (*Database Management System*)

Menurut Connolly dan Begg (2005, p16), *Database management system* adalah sistem perangkat lunak yang memungkinkan user untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke *database.*

2.1.5 Pengertian *Data Warehouse*

Menurut W.H Inmon (2005, p495) *data warehouse* merupakan kumpulan dari *database* yang memiliki sifat berorientasi subjek, terintegrasi, yang dirancang untuk dapat mendukung pengambilan keputusan dalam organisasi, dimana tiap datanya berhubungan dengan suatu kejadian yang terjadi pada suatu waktu tertentu.

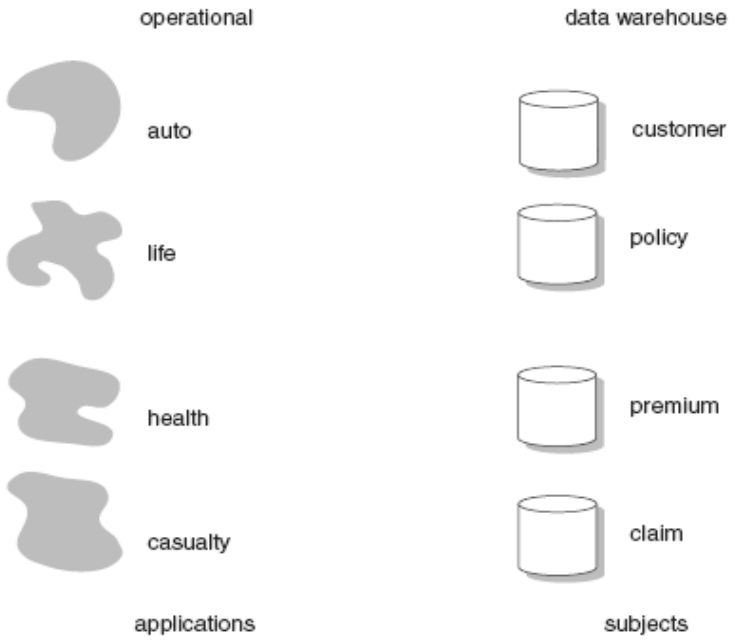
Menurut Connolly dan Begg (2005,p1151), *data warehouse* adalah kumpulan data yang berorientasi subjek, terintegrasi, *time-variant,* dan *non-volatile,* yang mendukung proses pengambilan keputusan dalam manajemen.

2.1.6 Karakteristik *Data Warehouse*

Menurut W.H Inmon (2005, p29) karakteristik *data warehouse* adalah sebagai berikut :

a. *Subject Oriented* (Berorientasi subyek)

Data Warehouse bersifat *subject oriented* artinya *data warehouse* digunakan untuk menganalisa data berdasarkan subyek-subyek tertentu dalam organisasi, bukan berorientasi pada aplikasi-aplikasi tertentu, yang mempermudah *user* dalam pengambilan keputusan.



Gambar 2.1 Karakteristik *Data Warehouse : Subject Oriented*

(Sumber : W.H Inmon, 2005, p30)

Beberapa perbedaan antara data primitif (data operasional) dan *data warehouse* diantaranya sebagai berikut :

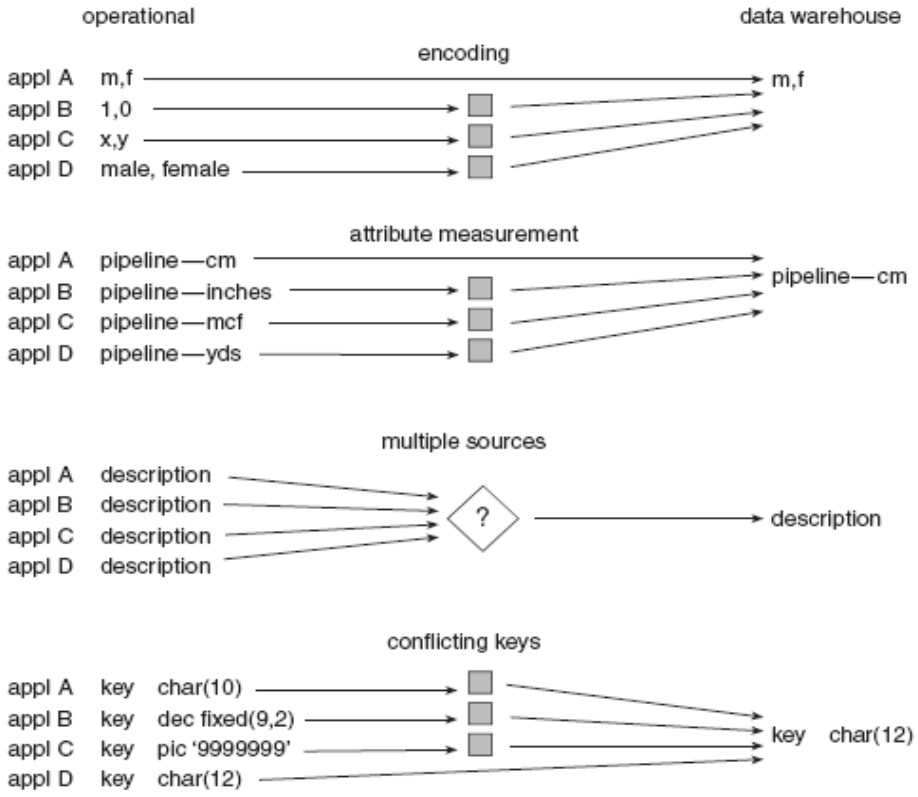
Tabel 2.1 Perbedaan data operasional dan *data warehouse*

Data Operasional	***Data Warehouse***
Berorientasi pada aplikasi	Berorientasi subyek
Dapat berubah	Tidak dapat berubah
Dapat diakses oleh sebuah unit dalam satu waktu	Dapat diakses oleh sebuah set unit dalam satu waktu
Jumlah data yang diproses kecil	Jumlah data yang diproses besar
Tidak ada *redudancy* data	Ada *redudancy* data
Untuk komunitas karyawan	Untuk komunitas manajer

b. *Integrated* (Terintegrasi)

Data warehouse bersifat *integrated* artinya *data warehouse* menyimpan data dari berbagai sumber yang berbeda yang disimpan ke dalam suatu format yang konsisten dan data tersebut terintegrasi satu sama lain, dimana data-data tersebut merupakan suatu kesatuan dan tidak dapat dipecah-pecah.

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

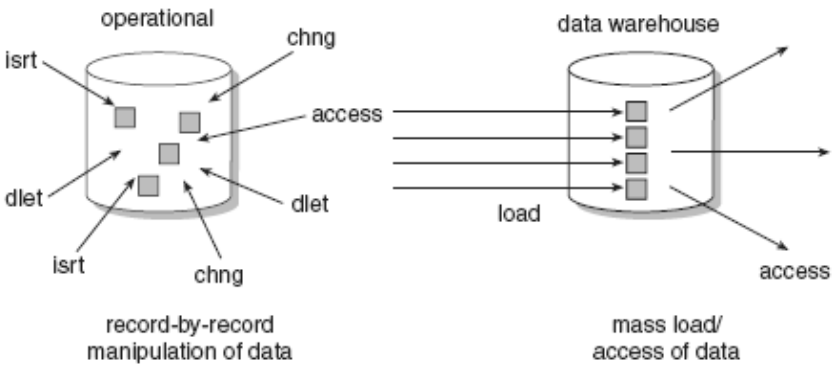


Gambar 2.2 Karakteristik *Data Warehouse : Integrated*

(Sumber : W.H Inmon, 2005, p31)

c. *Non volatile (*Tidak dapat diubah*)*

Data warehouse bersifat *non volatile*, artinya *data warehouse* tidak dapat diubah. Pengguna tidak dapat mengubah *data warehouse* yang sudah ada. Berbeda dengan *database* operasional yang memiliki tiga kegiatan operasi yaitu *insert, update,* dan *delete*, *data warehouse* hanya memiliki dua kegiatan yaitu *loading* dan akses data.



Gambar 2.3 Karakteristik *Data Warehouse : Non volatile*

(Sumber : W.H Inmon, 2005, p32)

d. *Time variant* (Variasi Waktu)

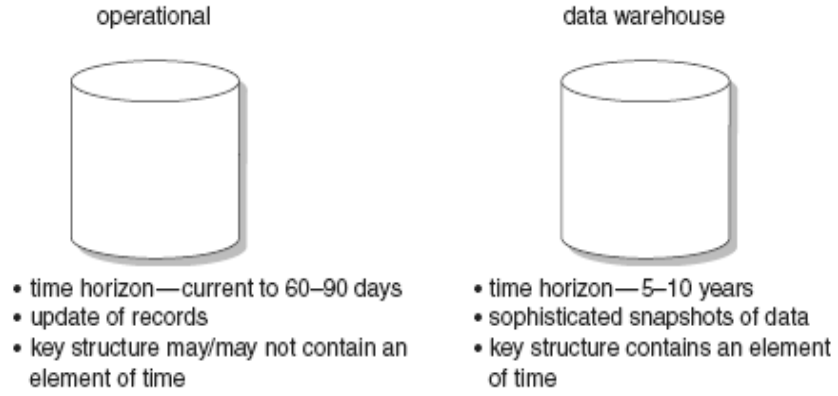
Dalam OLTP, *record* yang ada merupakan *record* terbaru sebab OLTP tidak menyimpan *record-record* yang lama untuk mempercepat proses karena semakin sedikit data yang disimpan maka waktu pemrosesan data semakin cepat. Lain halnya dengan *data warehouse* yang berisi *record* - *record* yang bersifat historis dan dapat tetap berada dalam sistem untuk jangka waktu 5-10 tahun sehingga dapat digunakan sebagai bahan analisis pengambilan keputusan. Tetapi, *record* yang terlalu lama juga bisa tidak efektif karena bisa memberikan hasil analisa yang kurang sesuai dengan *trend* yang ada di masa mendatang. Oleh karena itulah data pada *data warehouse* bersifat *time variant* atau akurat pada periode tertentu.

Selain itu, data pada *data warehouse* dikatakan memiliki perbedaan / rentang waktu *(time variance)* karena *data warehouse* juga mempunyai dimensi waktu sehingga *data warehouse* akurat

selama periode waktu tertentu dan dapat digunakan untuk mempelajari *trend* dan perubahan. Hal ini sangat berbeda dengan data pada OLTP, dimana data hanya akurat untuk waktu sesaat setelah diakses.

Aspek yang menunjukkan karakteristik *time variant* dalam *data warehouse* adalah sebagai berikut :

- *Data warehouse* merepresentasikan data untuk kurun waktu 5-10 tahun, sedangkan OLTP merepresentasikan data untuk jangka waktu yang lebih singkat, yaitu sekitar 60-90 hari. Karena aplikasi yang digunakan pada OLTP harus memiliki waktu respon yang singkat maka data yang diproses harus optimal.
- Setiap struktur data pada *data warehouse* mengandung elemen waktu seperti tahun, bulan, minggu, hari, dan sebagainya.
- Data pada *data warehouse* merupakan serangkaian *snapshot*, yaitu potongan data yang dikelompokkan sesuai dengan urutan waktu.



Gambar 2.4 Karakteristik *Data Warehouse : Time Variant*

(Sumber : W.H Inmon, 2005, p32)

2.1.7 Anatomi *Data Warehouse*

Dalam menentukan bentuk *data warehouse* yang akan digunakan oleh suatu perusahaan, terlebih dahulu kita harus mengetahui kebutuhan informasi yang diperlukan oleh perusahaan. *Data warehouse* terdiri atas tiga jenis dasar sistem, yaitu :

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

   Data warehouse fungsional dibangun berdasarkan kebutuhan informasi dari tiap bagian fungsi bisnis perusahaan. *Data warehouse* fungsional merupakan pendekatan yang digunakan untuk membangun suatu sistem *data warehouse* dengan biaya investasi yang rendah. Keuntungan dari bentuk ini adalah sistem mudah dibangun dengan biaya yang relatif murah sedangkan kerugiannya adalah resiko kehilangan konsistensi data dan terbatasnya kemampuan dalam pengumpulan data bagi pengguna.

2. ***Data warehouse* Terpusat *(Centralized Data Warehouse)***

   Data warehouse terpusat dibangun dari data operasional yang dikumpulkan dalam pusat penyimpanan data yang digunakan oleh pengguna untuk membangun *data warehouse* fungsional masing-masing.

   Menurut W.H Inmon (2005, p193), kebanyakan organisasi membangun dan memelihara lingkungan *data warehouse* terpusat yang tunggal. Pengaturan ini masuk akal karena alasan sebagai berikut :

a. Data di dalam *data warehouse* terintegrasi antar perusahaan dan gambaran terintegrasi digunakan hanya pada kantor pusat.
b. Perusahaan beroperasi pada model bisnis terpusat.
c. *Volume* dari data dalam *data warehouse* seperti tempat penyimpanan tunggal yang terpusat.
d. Sekalipun data dapat terintegrasi dan diedarkan antar area lokal yang beragam, data tersebut akan tidak praktis untuk diakses.

Keuntungan dari bentuk ini adalah data benar-benar terpadu karena konsistensinya yang tinggi sedangkan kerugiannya adalah biaya yang mahal serta perlu waktu yang cukup lama dalam membangun bentuk ini.

3. *Data warehouse* Terdistribusi *(Distributed Data Warehouse)*

Perusahaan yang memiliki cabang tersebar di seluruh dunia membutuhkan informasi yang mencakup tidak hanya wilayah lokal saja tetapi juga wilayah global. Global *data warehouse* membutuhkan informasi terpadu dari *data warehouse* tempat infomasi dikumpulkan. Disamping itu, ada kebutuhan yang lain untuk *data warehouse* yang terpisah di setiap cabang perusahaan. Dalam kasus ini, menurut W.H Inmon (2005, p193), *data warehouse* terdistribusi dibutuhkan. Tiga tipe dari *data warehouse* terdistribusi :

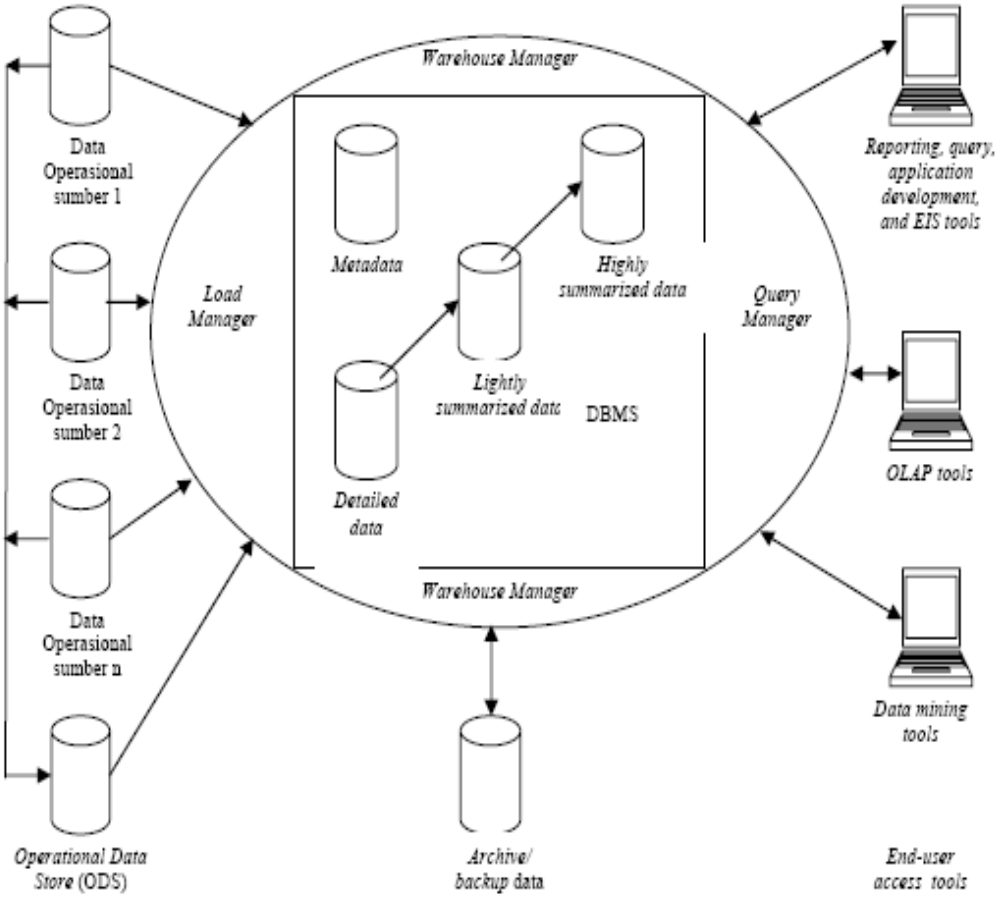
a. *Data warehouse* yang terdistribusi secara geografi terdiri dari *data warehouse* lokal dan *data warehouse* global.

b. *Data warehouse* yang terdistribusi dalam banyak prosesor, secara logis ada satu *data warehouse* tetapi secara fisiknya ada banyak *data warehouse* yang saling berhubungan.
c. *Data warehouse* yang tumbuh dalam sumber yang tidak terkoordinasi.

Keuntungan dari bentuk ini adalah kelebihan dalam mengakses data dari luar perusahaan yang telah mengalami sinkronisasi terlebih dahulu dan tetap terjaga konsistensinya, namun kerugian dari bentuk ini adalah bentuk yang paling mahal dan kompleks untuk diterapkan, karena sistem operasinya dikelola secara terpisah.

2.1.8 Arsitektur *Data Warehouse*

Menurut Connolly dan Begg (2005, p1157), arsitektur *data warehouse* dapat digambarkan sebagai berikut :



Gambar 2.5 Arsitektur *Data Warehouse*

(Sumber : Connolly dan Begg, 2005, p1157)

2.1.8.1 Data Operasional

Sumber data untuk *data warehouse* berasal dari (Connolly dan Begg, 2005, p1156) :

- *Mainframe* operasional data memegang kendali di hirarki generasi pertama dan di *database* jaringan.

- Bagian data memegang kendali di kepemilikan *system file* seperti VSAM, RMS, dan relasi DBMS seperti *Informix* dan *Oracle.*
- *Private* data memegang kendali di *workstation* dan *private servers*.
- *External System* seperti Internet, *database* komersial yang tersedia atau *database* yang berhubungan dengan pemasok organisasi atau konsumen.

2.1.8.2 Operasional *Datastore*

Operasional *datastore* adalah sebuah tempat penyimpanan untuk data operasional saat ini dan terintegrasi yang digunakan untuk analisis. Operasional *database* seringkali terstruktur dan terisi dengan data yang sama seperti *data warehouse*, tetapi kenyataannya operasional *datastore* adalah suatu tempat untuk menampung data yang akan masuk ke dalam *data warehouse.*

2.1.8.3 *Load Manager*

Load Manager menampilkan semua operasi yang berhubungan dengan *extraction* dan *loading* data ke dalam *data warehouse*. Datanya mungkin diekstrak secara langsung dari

sumber data atau lebih umumnya berasal dari operasional *datastore*.

2.1.8.4 *Warehouse Manager*

Warehouse Manager menampilkan semua operasi yang berhubungan dengan manajemen data di dalam *data warehouse*. Operasi yang dilakukan oleh *warehouse manager* antara lain :

- Analisis data untuk memastikan konsistensi data.
- Perubahan dan menyatukan sumber data dari penyimpanan sementara ke dalam tabel di *data warehouse*.
- Membuat indeks-indeks dan *views* di tabel awal.
- Membuat denormalisasi (jika perlu).
- Membuat aggregrasi (jika perlu).
- Mem-*backup* data dan menyimpan data.

2.1.8.5 *Query Manager*

Query Manager menampilkan semua operasi yang berhubungan dengan manajemen dari pengguna *queries*. Operasi yang dilakukan oleh *query manager* antara lain memerintah *query* untuk dimasukkan kedalam tabel yang benar dan menjadwal eksekusi *query*. Di beberapa kasus, *query manager* juga membuat *profile query* untuk membolehkan

warehouse manager untuk menentukan indeks dan agregrasi mana saja yang diperlukan.

2.1.8.6 *Detailed data*

Di beberapa kasus, detil data tidak disimpan secara *online* tetapi ditentukan oleh aggregrasi data ke *level* berikutnya. Bagaimanapun, di basis regular, detil data dimasukkan ke dalam *data warehouse* untuk menambah agregrasi data.

2.1.8.7 *Lightly and Highly Summarized Data*

Tujuan dari ringkasan informasi adalah untuk menaikkan kemampuan *queries*. Walaupun ada harga operasional yang berhubungan secara inisial dengan ringkasan data, ini ditutupi dengan menghilangkan keperluan untuk melanjutkan ringkasan operasi-operasi (seperti *sorting* atau *grouping*) di pengguna *queries*. Ringkasan data diperbaharui secara terus menerus sebagai data baru yang diisi ke dalam *data warehouse*.

2.1.8.8 *Archive / Backup Data*

Walaupun ringkasan data didapat dari detil data, mungkin diperlukan untuk mem-*backup* ringkasan data *online* jika data tersebut tetap melebihi penyimpanan untuk detil data.

Data tersebut dipindah ke tempat penyimpanan seperti *magnetic tape* atau *optical disk*.

2.1.8.9 *Metadata*

Metadata digunakan untuk berbagai tujuan antara lain :

- Proses *extraction* dan *loading* - *metadata* digunakan untuk memetakan sumber data ke *view* umum data dalam *data warehouse.*
- Proses manajemen *warehouse* - *metadata* digunakan untuk mengotomatiskan produksi dari tabel ringkasan.
- Sebagai bagian dari proses manajemen *warehouse* - *metadata* digunakan untuk mengatur *query* ke sumber data yang paling penting.

2.1.8.10 *End-User Access Tools*

Tujuan utama dari *data warehouse* adalah untuk menyediakan informasi untuk pemakai bisnis dalam membuat keputusan strategis. Pemakai ini berinteraksi dengan *data warehouse* menggunakan *end-user access tools*. *End-user access tools* dapat dibagi menjadi 5 kelompok, yaitu :

- Laporan dan Alat *Query*

  Menghasilkan program laporan dan laporan tertulis, sedangkan *query tools* didesain untuk menerima SQL atau

menghasilkan pernyataan SQL untuk men-*query* data di dalam *data warehouse*.

- Alat Pengembangan Aplikasi.

  Keperluan *end user* dari laporan dan alat *query* terkadang tidak cukup karena analisis yang diperlukan tidak dapat ditampilkan atau karena interaksi pengguna memerlukan keahlian yang tinggi dari *user*. Beberapa dari alat pengembangan aplikasi ini terintegrasi dengan alat OLAP yang terkenal, dan dapat mengakses semua sistem *database* utama, termasuk *Oracle*, *Sybase*, dan *Informix*.

- Alat *Executive Information System* (EIS)

  Executive Information System, dikembangkan untuk mendukung pembuatan keputusan tingkat tinggi. Alat EIS berhubungan dengan *mainframe* pengguna untuk membangun kebiasaan-kebiasaan, aplikasi grafik pendukung keputusan untuk menyediakan sebuah gambaran data-data organisasi dan akses ke sumber data luar.

- Alat *Online Analytical Processing* (OLAP)

  Alat OLAP berdasarkan dari konsep *multi-dimensional database* dan membolehkan pengguna untuk menganalisa data menggunakan kompleks, *multi-dimensional views*. Aplikasi bisnis khusus untuk alat ini

menilai dari keefektifan *marketing*, perkiraan *sales* produk, dan rencana kapasitas.

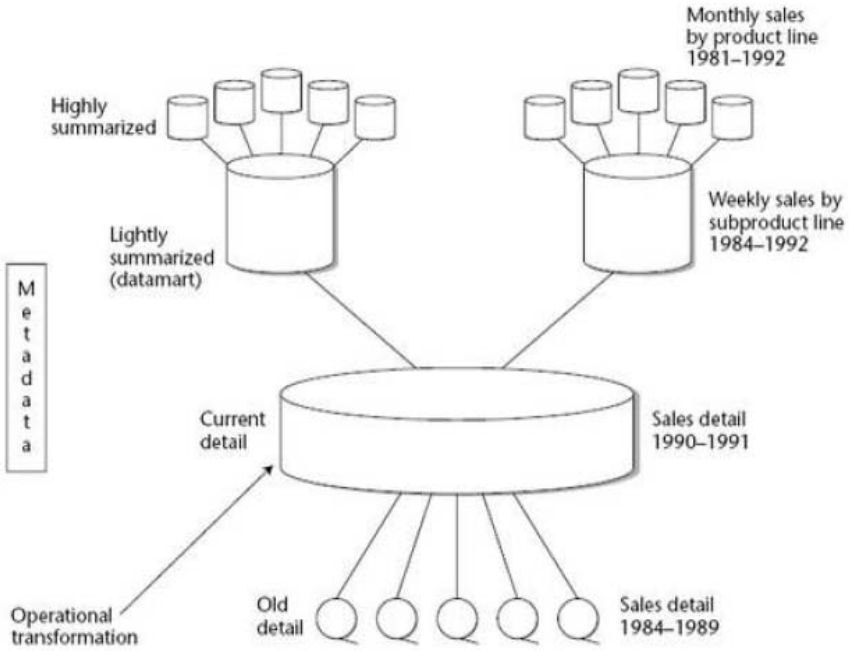
- Alat *Data Mining*

  Data Mining adalah proses menemukan korelasi baru, pola, arah yang baru dengan menambang sejumlah besar data menggunakan statistik, matematika, dan teknik *artificial intelligence* (AI).

2.1.9 Struktur Data Warehouse

Menurut W.H Inmon (2005, p33), data dalam *data warehouse* maupun adanya perubahan data didalamnya berasal dari lingkungan *database* operasional dimana data akan mengalami memiliki transformasi-transformasi yang diperlukan dalam perpindahannya. Apabila data dalam *data warehouse* sudah menua (data yang lama) maka data tersebut akan berpindah dari *current level of detail* ke *older level of detail.* Jika data sudah rangkum maka data akan berpindah dari *current level of detail* ke *level of lightly summarized data* kemudian akhirnya akan berpindah ke *level of highly summarized data*.

Struktur *data warehouse* dapat dilihat pada gambar :



Gambar 2.6 Struktur *Data Warehouse*

(Sumber : W.H Inmon, 2005, p34)

Menurut Vikas R. Agrawal (2005, p13), tingkatan-tingkatan data menurut perbedaan pada tingkatan *summary* data dan umur *data warehouse* adalah dijelaskan sebagai berikut :

- ***Current Detail Data*** **(detil data saat ini)**

  Current Detail Data adalah data detil yang aktif saat ini, yang merupakan *level* terendah dari *data warehouse* dan mencerminkan keadaan yang sedang berjalan. *Current Detail Data* biasanya memerlukan tempat penyimpanan yang cukup besar.

- ***Old Detail Data* (detil data historis)**

  Old Detail Data merupakan data historis, yang dapat berupa hasil *backup* yang disimpan dalam media penyimpanan yang terpisah dan dapat diakses sewaktu-waktu pada saat dibutuhkan. Penyusunan direktori untuk data ini harus menggambarkan umur dari data agar memudahkan apabila ingin diakses kembali.

- ***Highly Summarized Data* (ringkasan data *level* tinggi)**

  Highly Summarized Data merupakan hasil proses ringkasan yang bersifat total dan mudah untuk diakses. Digunakan untuk melakukan analisa perbandingan data berdasarkan urutan waktu dan analisa yang menggunakan data multi dimensi. *Database* multi-dimensi adalah suatu teknologi *software* komputer yang dirancang untuk meningkatkan efisiensi dalam mencari tabel atau *query* sehingga media penyimpanan menjadi lebih baik, serta memudahkan pengambilan data dalam jumlah yang besar.

- ***Lightly Summarized Data* (ringkasan data *level* menengah)**

  Lightly Summarized Data merupakan ringkasan dari detil data, tetapi belum bersifat total *summary*. Data-data ini memiliki tingkatan yang lebih tinggi dan mendukung kebutuhan *data warehouse* pada tingkatan departemen. Tingkatan data ini disebut juga *data mart*. Akses terhadap data jenis ini banyak digunakan untuk *view* dari suatu kondisi yang sedang dan sudah berjalan.

- ***Metadata***

Menurut Bradley W Klenz (1999, p527), sebuah komponen penting didalam lingkungan *data warehouse* adalah *metadata* atau data mengenai data, yang menyediakan informasi tentang darimana datangnya nilai - nilai data dan bagaimana cara nilai data tersebut diperoleh. *Metadata* juga menjelaskan bagaimana data – data tersebut saling berhubungan.

Menurut W.H Inmon (2005, p102), sebuah komponen penting didalam lingkungan *data warehouse* adalah *metadata* (atau data mengenai data), yang merupakan sebuah bagian dari aturan pemrosesan informasi dimana jika masih terdapat program dan data. Dalam lingkungan *data warehouse*, *metadata* memiliki tingkat kepentingan yang berbeda, karena *metadata* memungkinkan cara penggunaan *data warehouse* yang paling efektif.

Metadata memungkinkan *end-user* atau DSS *analyst* untuk menavigasi melalui kemungkinan-kemungkinan yang ada didalam *metadata* tersebut.

Dilain pihak, pada saat *user* melakukan pendekatan terhadap *data warehouse* yang tidak ada *metadata*nya, maka *user* tidak tahu harus memulai analisis dari mana. *User* harus mencari tahu didalam *data warehouse* bahwa data apa saja yang tidak terdapat didalamnya sehingga banyak waktu akan terbuang sia-sia. Bahkan setelah *user* berhasil mencari tahu, tetap tidak ada jaminan bahwa dia akan menemukan data yang benar atau menerjemahkan data tersebut

dengan benar. Dengan bantuan *metadata* maka *end-user* dapat langsung menemukan data yang dibutuhkan atau dapat mengetahui ada atau tidaknya data tersebut dan mengetahui tentang apa sebenarnya data tersebut. Jadi dapat dikatakan bahwa *metadata* berperan sebagaimana daftar isi dalam sebuah buku. Ada beberapa *item* dalam penyimpanan *metadata* sebagai berikut :

- Struktur dari data yang dapat digunakan oleh *programmer*
- Struktur dari data yang dapat digunakan oleh DSS *analyst*
- Sumber data yang menyokong *data warehouse*
- Transformasi data didalam perpindahan kedalam *data warehouse*
- Model data
- Hubungan antara model data dan *data warehouse*
- Catatan dari penggunaan data (*History of Extracts*)

Metadata memuat informasi yang penting mengenai data dalam *data warehouse* yang berfungsi sebagai :

1. Direktori yang akan dipakai oleh *user* dalam mencari lokasi dalam *data warehouse.*
2. Suatu panduan untuk *summary* data dari detail data menjadi *lightly summarized data* dan kemudian menjadi *highly summarized data*.
3. Merupakan penuntun pemetaan (*mapping*) dalam proses transformasi dari operasional ke *data warehouse.*

Karena *data warehouse* harus melayani banyak fungsi, maka *metadata* penting untuk menjawab kebutuhan dari suatu fungsi tertentu, karena setiap departemen dalam perusahaan biasanya menggambarkan struktur data yang spesifik meskipun asal datanya sama.

2.1.10 Kegiatan Inti *Data Warehouse*

Untuk melakukan penganalisaan dan pelaporan informasi bagi pihak-pihak pengambil keputusan maka dalam merancang *data warehouse* terdapat kegiatan-kegiatan yang harus ada didalamnya.

Kegiatan-kegiatan inti itu adalah:

- Memperoleh dan menggabungkan data

  Mendapatkan data dari berbagai sumber dan melakukan penggabungan pada suatu tempat tertentu, data-data yang digabung adalah data-data yang akan membantu kita dalam pembuatan laporan, karena data tersebut merupakan suatu bentuk kesatuan.

- Transformasi data

  Pengolahan data dari awal ke bentuk data yang telah disepakati. Dengan mengalami pemrosesan atau pengolahan terlebih dahulu, yang sama artinya dengan pengubahan data ke bentuk yang diharapkan.

- Pendistribusi data

  Data-data yang akan kita gunakan dalam *data warehouse* berkaitan dengan lingkungan kerja dalam perusahaan. Bagi perusahaan yang terhubung dengan jaringan, pemakaian *data warehouse* mendukung kegiatan ini, dimana pengguna dapat menggunakan *data warehouse* ini secara lebih fleksibel dan merata pada masing-masing bagian yang ada dalam perusahaan.

- Penggunaan data

  Data yang telah disaring akan menghasilkan ringkasan-ringkasan yang dapat memudahkan pengguna dalam mengambil suatu keputusan.

2.1.11 *Data Flow Data Warehouse*

Proses ini terdiri dari :

- *Inflow*

  Berhubungan dengan *loading*, pembersihan dan pembacaan data dari sumber sistem ke dalam *data warehousing*.

- *Upflow*

  Proses ini berhubungan dengan penambahan nilai ke data dalam *data warehousing* seperti ringkasan, pengepakan, dan distribusi data.

- *Downflow*

  Proses ini berhubungan dengan pengarsipan dan data *backup* dalam *data warehousing*.

- *Outflow*

  Proses ini berhubungan dengan pembuatan data yang dapat dipakai oleh *end-users*. 2 kunci dari aktivitas ini terdiri dari :

  - *Accessing* : berkonsentrasi pada kepuasan permintaan pemakai atas data yang mereka perlukan.
  - *Delivering* : berkonsentrasi pada pengiriman informasi yang proaktif kepada *workstation* pemakai.

- *Metaflow*

  Proses ini berhubungan dengan deskripsi isi dari data dari *data warehousing*.

2.1.12 *Granularity*

Salah satu faktor penting yang harus diperhatikan oleh pengembang *data warehouse* adalah *granularity*. *Granularity* mempengaruhi efisiensi dari penggunaan data dalam analisis yang dilakukan.

Menurut W.H Inmon (2005, p41) *granularity* merupakan sebuah *level* kedetilan / *summarization* dari unit data yang ada dalam *data warehouse*. Semakin tinggi tingkat kedetilan data maka semakin rendah level *granularity* dan juga sebaliknya.

2.1.13 Agregasi

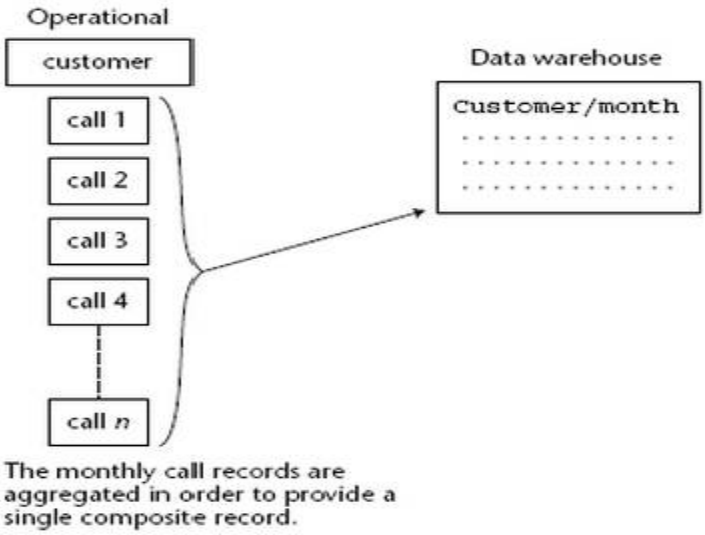
Menurut W.H Inmon (2005, p114), terdapat banyak kasus dimana data dalam *data warehouse* tidak memenuhi kriteria stabilitas dan tidak sering berubah, kasus lainnya dimana jumlah data menjadi terlalu banyak, sering terjadi perubahan isi data, dan sebagainya. Dalam kasus-kasus seperti demikian, dapat dilakukan agregasi yang mengelompokkan beberapa data detil operasional yang berbeda ke dalam satu *record* tunggal. *Record* tunggal itu disebut sebagai *profile record* atau *aggregate record*.

Sebuah *profile record* dibuat untuk mengelompokkan *record* detil yang sangat banyak jumlahnya. Sebagai contoh, sebuah perusahaan telepon pada akhir bulan mengumpulkan semua data-data aktivitas telepon para pelanggan dalam sebulan kedalam *data record* pelanggan dalam *data warehouse*.

Agregrasi dari data operasional kedalam sebuah *record* tunggal dalam data *warehouse* dapat dilakukan dengan menggunakan cara, seperti :

- Nilai-nilai yang diambil dari data operasional yang dapat diringkas.
- Unit-unit data operasional dapat dihitung / dijumlahkan, dimana jumlah dari unit data tersebut disimpan.
- Unit-unit data dapat diproses untuk menentukan yang paling tinggi, paling rendah, rata-rata, dan lain-lain.
- Kemunculan pertama dan terakhir sebuah data dapat ditangkap.

- Tipe data tertentu, yang berbeda pada batasan parameter tertentu dapat diukur.
- Data yang efektif pada momem waktu tertentu dapat terperangkap.
- Data yang paling muda dan yang paling tua dapat ditangkap.



Gambar 2.7 Pembentukan satu *profile record* dari beberapa *record* detil

(Sumber : W.H Inmon, 2005, p115)

Faktor yang mendukung pembuatan agregrasi adalah :

- meningkatkan performa *query*.
- mengurangi jumlah penggunaan CPU *cycle*.

2.1.14 Denormalisasi

Menurut W.H Inmon (2005, p495), denormalisasi adalah suatu teknik untuk menempatkan data hasil normalisasi ke dalam lokasi fisikal yang dapat mengoptimasi kinerja sistem. Alasan dilakukannya denormalisasi adalah untuk mengurangi jumlah penggabungan yang

harus diproses dalam rata-rata *query*, sehingga meningkatkan performa. Sekarang ini, pengembang *data warehouse* telah memperbaiki teknik denormalisasi dan menghasilkan pendekatan skema bintang.

2.1.15 Keuntungan *Data Warehouse*

Menurut Connolly dan Begg (2005, p1152), *data warehouse* yang telah diimplementasikan dengan baik dapat memberikan keuntungan yang besar bagi organisasi, yaitu :

1. Potensi nilai balik yang besar pada investasi

   Sebuah organisasi harus mengeluarkan uang dan sumber daya dalam jumlah yang cukup besar untuk memastikan kalau *data warehouse* telah diimplementasikan dengan baik dan biaya yang dikeluarkan dapat berkisar antara 50.000 *pound* sampai 10.000.000 *pound*, tergantung dari solusi teknikal yang diinginkan.

2. Keuntungan Kompetitif

   Keuntungan kompetitf didapatkan apabila pengambil keputusan mengakses data yang dapat mengungkapkan informasi yang sebelumnya tidak diketahui, tidak tersedia, misalnya informasi mengenai konsumen, *trend*, dan permintaan.

3. Meningkatkan produktifitas para pengambil keputusan perusahaan

   Data warehouse meningkatkan produktifitas para pengambil keputusan perusahaan dengan menciptakan sebuah *database* yang terintegrasi secara konsisten, berorientasi pada subjek, dan data historis. *Data Warehouse* mengintegrasikan data dari beberapa *system*

yang tidak kompatibel ke dalam bentuk yang menyediakan satu pandangan yang konsisten dari organisasi. Dengan mengubah data menjadi informasi yang berguna, maka seorang manajer bisnis dapat membuat analisa yang lebih akurat dan konsisten.

2.1.16 Skema Bintang

Menurut Connolly dan Begg (2005, p1183), skema bintang adalah struktur logikal yang memiliki tabel fakta yang berisi data faktual, dikelilingi oleh tabel dimensi yang berisi *data reference* (dimana dapat didenormalisasikan).

Menurut Bradley W Klenz (1999, p528), struktur skema bintang memiliki satu tabel fakta pusat dengan beberapa kunci utama yang berhubungan dengan beberapa tabel dimensi. Tiap tabel dimensi berisikan atribut – atribut yang berfungsi untuk memisahkan tiap kategori. Atribut – atribut tersebut menyediakan akses ke nilai fakta secara optimal dengan memungkinkan *subsetting* diselesaikan pada tabel dimensi dan kemudian mengakses tabel fakta untuk dianalisa.

2.1.17 Keuntungan Menggunakan Skema Bintang

Menurut Connolly dan Begg (2005, p1185) keuntungan menggunakan skema bintang antara lain :

- Efisiensi, struktur *database* yang konsisten membuat akses data lebih efisien dengan menggunakan alat untuk menampilkan data termasuk laporan tertulis dan *query*.

- Kemampuan untuk mengatasi perubahan kebutuhan, skema bintang dapat beradaptasi terhadap perubahan kebutuhan, karena semua tabel dimensi memiliki kesamaan dalam hal menyediakan akses ke tabel fakta.
- *Extensibility*, model dimensional dapat dikembangkan. Contohnya menambah tabel fakta selama data masih konsisten, menambah tabel dimensi selama masih ada nilai tunggal di tabel dimensi tersebut yang mendefinisikan setiap *record* tabel fakta yang ada.
- Kemampuan untuk menggambarkan situasi bisnis pada umumnya.
- Proses *query* yang bisa diprediksi, aplikasi *data warehouse* yang mencari data dari *level* yang dibawahnya akan dengan mudah menambah jumlah atribut pada tabel dimensi dari sebuah skema bintang.

2.1.18 Tipe Tabel Skema Bintang

Di dalam *data warehouse* terdapat dua macam tipe tabel, yaitu :

1. **Tabel fakta (*fact table*)**

   Tabel fakta sering disebut juga tabel *mayor*. Tabel ini berisi data aktual yang akan dianalisis (data kuantitatif dan transaksi). *Field-field* tabel fakta sering disebut *measure* dan juga selalu berisi *foreign key* dari masing-masing tabel dimensi.

2. Tabel dimensi (*dimension table*)

Tabel dimensi sering disebut juga tabel *minor*. Tabel dimensi biasanya lebih kecil daripada tabel fakta dan berisi data yang merupakan deksripsi dari data-data yang ada pada tabel fakta.

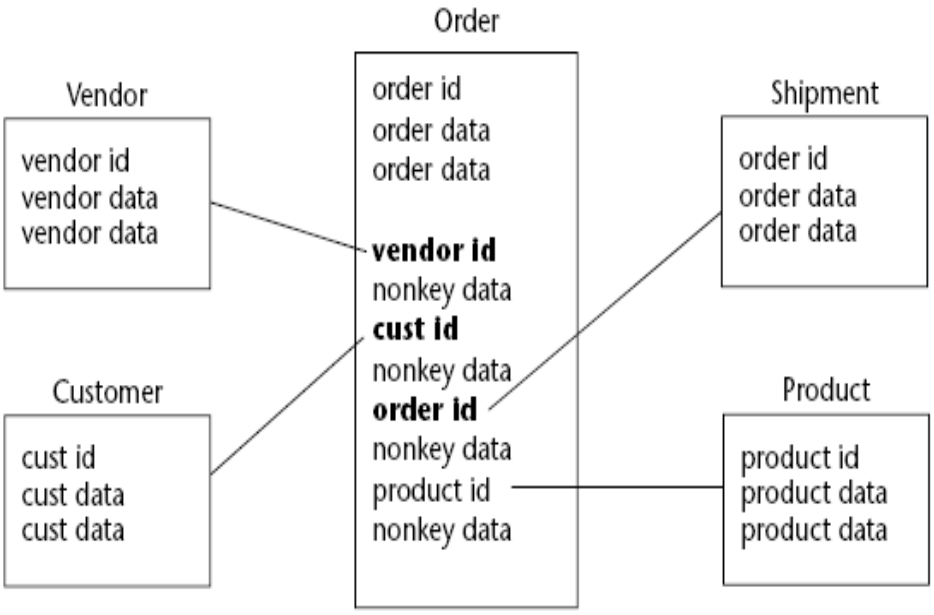
2.1.19 Jenis Skema Bintang

Skema bintang merupakan suatu rancangan *database* di dalam *data warehouse* yang menggambarkan hubungan yang jelas antara struktur tabel fakta dan tabel dimensi. Skema bintang terdiri dari beberapa jenis, yaitu :

1. Skema bintang sederhana

Dalam skema bintang sederhana ini, setiap tabel harus mempunyai *primary key* yang dapat terdiri dari satu kolom atau lebih. *Primary key* pada tabel fakta dapat terdiri dari satu atau lebih *foreign key* dan *primary key* tersebut harus *unique*.

Gambar dibawah ini menggambarkan hubungan antara tabel fakta dan tabel dimensi. Di dalam tabel fakta terdapat empat *foreign key* yang berasal dari *primary key* tabel dimensi.



Gambar 2.8 Skema Bintang Sederhana

(Sumber : W.H Inmon, 2005, p129)

2. Skema Bintang dengan Banyak Tabel Fakta

Skema bintang juga dapat memiliki lebih dari satu tabel fakta. Hal ini terjadi karena skema ini berisi lebih banyak tabel fakta, misalnya disamping penjualan terdapat tabel fakta pembelian dan tabel fakta persediaan. Tetapi meskipun terdapat banyak tabel fakta, skema ini menggunakan tabel dimensi secara bersama-sama.

2.1.20 Proses ETL *(Extract – Transform – Load)*

Menurut Hoffer (2005, p456), tujuan dari proses ETL adalah untuk menyediakan sumber data tunggal dan ter-autorisasi yang

mendukung pembuatan keputusan. Proses ETL dapat dilihat pada gambar berikut ini.

2.1.20.1 *Extract*

Extract merupakan proses mengambil data yang relevan dari *database* operasional untuk mengisi *data warehouse* perusahaan dan dilakukan berdasarkan analisa terhadap *database* operasional dan *data warehouse*. Alasannya yaitu tidak semua data yang ada di *database* operasional dibutuhkan, melainkan hanya sebagian. Proses *extract* terdiri dari dua tipe, yaitu :

1. *Static extract* : metode mengambil data yang dibutuhkan pada waktu tertentu dan digunakan untuk mengisi *data warehouse* untuk pertama kali.
2. *Incremental extract* : metode yang hanya mengambil perubahan yang terjadi pada data sejak data tersebut diambil terakhir kali. Artinya, data yang diambil adalah data setelah data terakhir yang diambil sebelumnya.

Hoffer menyatakan bahwa menurut English dan White kriteria umum dari kualitas data yang akan diekstrak tergantung pada hal-hal berikut :

- Kejelasan nama data sehingga perancang *data warehouse* tahu dengan pasti data apa saja yang ada dalam *database* operasional.

- Kelengkapan dan keakuratan proses bisnis yang langsung mempengaruhi keakuratan data
- Format data

2.1.20.2 *Cleanse*

Data scrubbing (data cleansing) adalah teknik yang menggunakan pola pengenalan dan teknik lainnya untuk meningkatkan kualitas dari data mentah sebelum mentransformasi data tersebut dan memindahkannya ke dalam *data warehouse*. *Data scrubbing* dilakukan untuk meningkatkan kualitas dari data. Kegiatan *cleansing* pada umumnya meliputi hal-hal berikut ini :

- *Decoding* data untuk membuat data dapat dimengerti dalam aplikasi *data warehouse*.
- Men-format ulang dan mengubah tipe data serta melakukan fungsi-fungsi lainnya untuk memasukan data dari *database* operasional ke dalam *data warehouse* dengan format yang siap untuk ditransformasikan
- Menambahkan *time stamp* untuk membedakan atribut yang sama sepanjang waktu
- Mengubah ukuran data yang memiliki nilai unit yang berbeda
- Membuat *primary key* untuk setiap tabel

- Menyesuaikan dan menggabungkan data hasil ekstraksi yang terpisah menjadi satu tabel atau satu *file* dan menyesuaikan data yang akan dimasukan ke dalam tabel yang baru dibuat
- Mencatat *error* yang terdeteksi, memperbaikinya, dan memproses ulang data tersebut tanpa membuat duplikat
- Menemukan data yang kurang untuk melengkapi data yang diperlukan pada proses *loading*

2.1.20.3 *Data Transformation*

Data transformation (transformasi data) adalah proses mengubah format data yang berasal dari *database* operational menjadi format data pada *data warehouse*. Transformasi data menerima data yang telah di ekstrak kemudian mengubah formatnya dan mengirimnya untuk melakukan *load* dan *index*.

Pada umumnya, tujuan *data scrubbing* adalah untuk memperbaiki *error* pada data *value* yang terdapat dalam *database* operasional, sedangkan tujuan transformasi data adalah untuk mengubah format data dari *database* operasional ke *data warehouse*. Fungsi transformasi data terbagi menjadi 2 kategori, yaitu :

1. *Record level function*

- *Selection (subsetting)* yaitu proses membagi data berdasarkan kriteria tertentu. Pada aplikasi *data warehouse*, digunakan untuk mengekstrak data yang relevan dari *database* operasional yang akan digunakan untuk mengisi *data warehouse*.
- *Joining* adalah proses menggabungkan data dari berbagai *database* operasional menjadi tabel tunggal. Pada aplikasi *data warehouse*, *joining* penting dilakukan untuk mengabungkan data karena data berasal dari berbagai sumber.
- Normalisasi yaitu proses mendekomposisi relasi yang memiliki anomali untuk menghasilkan relasi yang lebih kecil dan terstruktur dengan baik.
- Agregasi merupakan proses mentransformasi data dari *level detail* menjadi *level summary*.

2. *Field level function*

Mengubah format data *record* yang berasal dari *database* operasional menjadi format untuk *data warehouse*. *Field level function* terbagi menjadi 2 tipe, yaitu :

- *Single-field transformation* : mengubah data dari *field* tunggal pada *database* operasional ke dalam *field* tunggal pada *data warehouse*.

- *Multifield transformation* : mengubah data dari satu *field* atau lebih dari *database* operasional ke dalam satu atau lebih *field data warehouse*.

2.1.20.4 *Load and Index*

Langkah terakhir dalam mengisi *data warehouse* adalah memasukkan data yang telah dipilih ke dalam *data warehouse* dan membuat *index* yang diperlukan. Dua model dasar dari *loading* data ke dalam *data warehouse* adalah *refresh* dan *update*.

- *Mode refresh* digunakan untuk mengisi *data warehouse* yang menggunakan proses penulisan ulang ke target data secara berkala.
- *Mode update* adalah pendekatan dimana hanya perubahan pada sumber data yang akan ditulis pada *data warehouse*. Dengan kata lain *record* baru biasanya ditulis ke dalam *data warehouse* tanpa menimpa atau menghapus *record* sebelumnya.

2.1.21 Istilah-Istilah Lain yang Berhubungan dengan *Data Warehouse*

- ***Data Mart***

  Menurut W.H Inmon (2005, p494), *data mart* adalah struktur data per departemen yang berasal dari *data warehouse* dimana data di denormalisasi berdasarkan kebutuhan informasi tiap departemen.

  Menurut Connolly dan Begg (2005, p1171) *data mart* dengan sebuah area fungsional dari perusahaan atau memiliki lingkup yang terbatas.

  Ada beberapa karakteristik yang membedakan antara *data mart* dengan *data warehouse*, di antaranya :

  1. *Data mart* berfokus pada kebutuhan pengguna yang berhubungan dengan satu bagian departemen atau fungsi bisnis.
  2. *Data mart* tidak berisi data operasional secara detil, tidak seperti *data warehouse*.

  Data mart lebih mudah dimengerti dan digunakan karena berisi data yang lebih sedikit dibandingkan dengan *data warehouse*.

- **OLAP *(Online Analytical Processing)***

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

- **OLTP *(Online Transaction Processing)***

  Merupakan suatu pemrosesan yang menyimpan data mengenai kegiatan operasional atau transaksi perusahaan sehari-harinya.

- ***Dimension table* (tabel dimensi)**

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

- ***Fact Table* (tabel fakta)**

  Merupakan tabel yang pada umumya mengandung angka dan data *history* dimana *key* (kunci) yang dihasilkan sangat unik karena *key*nya merupakan kumpulan *foreign key* dari *primary key* yang ada pada masing-masing tabel dimensi yang berhubungan.

- **DSS *(Decision Support System)***

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

- ***Data Mining***

  Menurut Sid Adelman (2000, p145) *data mining* adalah proses pencarian pola data yang tidak diketahui atau tidak diperkirakan sebelumnya.

2.1.22 Metodologi Perancangan *Data Warehouse*

Menurut Kimball (Connolly, 2005, p1087) ada 9 langkah dalam perancangan *data warehouse* yang dikenal dengan *nine-step methodology*, yaitu :

2.1.22.1 Memilih Proses *(Choosing the Process)*

Memilih proses berarti menentukan subyek utama. Subyek utama merujuk pada suatu kegiatan bisnis perusahaan yang dapat menjawab semua pertanyaan bisnis yang penting serta memiliki ciri-ciri tersendiri.

2.1.22.2 Memilih *Grain (Choosing the Grain)*

Memilih *grain* artinya menentukan apa yang akan diwakili atau direpresentasikan oleh sebuah tabel fakta. Setelah menentukan *grain* dari tabel fakta maka untuk selanjutnya dapat ditentukan tabel-tabel dimensi yang berhubungan dengan tabel fakta tersebut. *Grain* pada tabel fakta juga menentukan *grain* untuk tabel dimensi.

2.1.22.3 Identifikasi dan Membuat Dimensi yang Sesuai *(Identifying and Conforming the Dimensions)*

Identifikasi dan hubungkan tabel dimensi dengan tabel fakta. Dimensi merupakan kumpulan sudut pandang yang penting untuk menggambarkan fakta-fakta yang terdapat pada tabel fakta.

2.1.22.4 Memilih Fakta (*Choosing the Facts*)

Grain dari suatu tabel fakta menentukan fakta-fakta yang bisa digunakan. Pada tahap ini, tentukan *measure* (ukuran) yang dibutuhkan pada tabel fakta.

2.1.22.5 Menyimpan Pra-Kalkulasi Dalam Tabel Fakta (*Storing Pre-Calculations In the Fact Table*)

Umumnya hasil perhitungan dari atribut di *database* tidak disimpan pada suatu atribut khusus pada *database* tersebut, namun pada tahap ini, perlu dipertimbangkan kembali penyimpanan hasil perhitungan pada suatu atribut tersendiri di *database* dengan alasan mengurangi resiko kesalahan pada program setiap kali melakukan perhitungan pada atribut-atribut tersebut.

2.1.22.6 Melengkapi Tabel Dimensi (*Rounding Out the Dimension Tables*)

Dari dimensi-dimensi yang telah diidentifikasi, pada tahap ini dibuat deskripsi yang memuat informasi terstruktur mengenai atribut-atribut pada tabel dimensi. Tabel dimensi tersebut harus diberi keterangan selengkap-lengkapnya dan keterangannya harus mudah dipahami oleh *user*.

2.1.22.7 Memilih Durasi Dari Database (*Choosing the Duration of the Database*)

Pada tahap ini ditentukan durasi atau periode waktu dari data-data yang akan dimasukkan ke dalam *data warehouse.* Misalnya saja pada perusahaan asuransi, data harus disimpan selama 10 tahun atau lebih.

2.1.22.8 Mencari Perubahan Pada Dimensi (*Tracking Slowly Changing Dimensions*)

Dimensi dapat berubah sehingga untuk mengantisipasinya ada tiga cara untuk mengubah data di dimensi, yaitu :

a) Menulis ulang atribut yang berubah.
b) Membuat *record* baru pada dimensi
c) Membuat suatu atribut alternatif untuk menampung nilai yang baru, sehingga nilai lama dan nilai baru dari atribut tersebut bisa diakses secara bersamaan.

2.1.22.9 Memutuskan Prioritas dan Cara *Query* (*Deciding the Query Priorities and the Query Modes*)

Pada tahap terakhir ini, dilakukan perancangan fisik dari *data warehouse* dan menentukan masalah-masalah yang mungkin ada pada perancangan fisik seperti administrasi, *backup, indexing*, dan *security.*

2.2 Teori Khusus

2.2.1 Penjualan Angsuran

Menurut Utoyo Widayat dan Sugito Wibowo (1993, p2), Penjualan angsuran yaitu penjualan barang dagang atau jasa yang dilaksanakan dengan perjanjian dimana pembayaran dilakukan secara bertahap atau berangsur. Biasanya pada saat barang atau jasa diserahkan kepada pembeli, penjual menerima uang muka *(down payment)* sebagai pembayaran pertama dan sisanya diangsur dengan beberapa kali angsuran.

Menurut Hadori Yunus dan Harnanto (1992,p109), Penjualan angsuran adalah penjualan yang dilakukan dengan perjanjian dimana pembayarannya dilaksanakan secara bertahap, yaitu :

1. Pada saat barang diserahkan kepada pembeli, penjual menerima pembayaran pertama sebagian dari harga penjualan (diberikan *down payment*).
2. Sisanya dibayar dalam beberapa kali angsuran.

Untuk mengurangi atau menghindarkan kemungkinan kerugian yang terjadi dalam pemilikan kembali, maka faktor – faktor yang harus diperhatikan oleh penjual adalah sebagai berikut :

1. Besarnya pembayaran pertama (*down payment*) harus cukup untuk menutup semua kemungkinan terjadinya penurunan harga barang tersebut dari semula barang baru menjadi barang bekas.
2. Jangka waktu pembayaran diantara angsuran yang satu dengan yang lain hendaknya tidak terlalu lama, kalau dapat tidak lebih dari satu bulan.

3. Besarnya pembayaran angsuran periodik harus diperhitungkan cukup untuk menutup kemungkinan penurunan nilai barang – barang yang ada selama jangka pembayaran yang satu dengan pembayaran angsuran berikutnya.

2.2.2 Penagihan

Menurut Kamus Besar Bahasa Indonesia (1999, p989), penagihan adalah proses, pembuatan, cara menagih; permintaan (peringatan dan sebagainya) supaya membayar hutang dan sebagainya.

2.2.3 Penarikkan

Menurut Kamus Besar Bahasa Indonesia (1999, p1012), penarikkan adalah proses, cara, perbuatan menarik. Penarikkan dilakukan apabila konsumen tidak dapat lagi melakukan pembayaran sesuai dengan perjanjian yang sudah disepakati oleh kedua belah pihak.