BAB 2

LANDASAN TEORI

2.1. Teori – Teori Umum

2.1.1. Pengertian *Data, Informasi, dan Database*

Menurut Turban Rainer Potter (2001,p17), "*Data are raw facts or elementary descriptions of things, events, activities, and transactions, that are captured, recorded, stored and classified, but not organized to convey any specific meaning*," yang berarti *data* merupakan fakta 'mentah' atau deskripsi awal mengenai suatu barang, kejadian, aktifitas, dan transaksi yang diperoleh, disimpan, dan diklasifikasikan, namun tidak diorganisasikan untuk menyampaikan arti yang spesifik.

Menurut Turban Rainer Potter (2001,p17), "*Information is a collection of facts(data) organized in some manner so that they are meaningful to a recipient*," yang berarti informasi adalah sekumpulan data yang sudah diolah, dibentuk, atau dimanipulasi sesuai dengan keperluan tertentu sehingga mempunyai arti bagi penerima. Informasi dapat berbentuk dokumen, laporan ataupun multimedia.

Menurut Thomas Connoly (2005, p14), "*Database is a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization* ," yang berarti bahwa *database* merupakan suatu kumpulan dari data yang

berhubungan secara logika dan diskripsi dari data – data tersebut, yang didesign untuk kebutuhan informasi dari suatu organisasi.

Jadi, dapat disimpukan bahwa *data* merupakan suatu bentuk keterangan dalam bentuk teks maupun numerik yang didapat dari berbagai macam sumber yang belum diolah atau dimanipulasi menjadi informasi. *Data* suatu perusahaan pada umumnya diperoleh dari hasil kegiatan operasional sehari-hari atau hasil transaksi. Informasi adalah hasil dari pengolahan data-data untuk membuat analisa kegiatan operasional yang telah dilakukan oleh perusahaan, yang pada akhirnya dapat membantu perusahaan mengambil keputusan untuk menentukan strategi perusahaan selanjutnya. Sedangkan *database* merupakan kumpulan data yang disimpan dalam tabel – tabel yang saling berhubungan, dan digunakan oleh sistem aplikasi dalam perusahaan, yang dirancang agar sesuai dengan informasi yang dibutuhkan oleh suatu perusahaan.

2.1.2. Pengertian *On–Line Transaction Processing* (OLTP) dan *On–Line Analytical Processing* (OLAP)

OLTP (*On-Line Transaction Processing*) menggambarkan kebutuhan sistem dalam ruang lingkup operasional dan merupakan proses yang mendukung operasi bisnis sehari-hari. Sebuah perusahaan biasanya memiliki sejumlah sistem – sistem OLTP yang berbeda untuk proses - proses bisnis seperti pengontrolan *inventory*, faktur pelanggan, dan *point-of-sale*. Sistem – sistem ini menghasilkan

data operasional yang mendetil, bersifat *current* (terbaru), dan dapat diperbaruhi. Sistem - sistem OLTP dioptimalkan untuk sejumlah transaksi yang bersifat dapat diprediksi, berulang, dan dapat diperbaruhi. Data OLTP diatur menurut kebutuhan transaksi yang berhubungan dengan aplikasi bisnis dan mendukung keputusan sehari-hari untuk sejumlah pengguna operasional.

OLAP(*On-Line Analytical Processing*) adalah proses analisis sejumlah besar data *multidimension* yang bersifat dinamis. OLAP dideskripsikan sebagai sebuah teknologi yang menggunakan suatu *view* bersifat *multidimension* mengenai sejumlah data yang teragregasi dan menyediakan akses informasi yang strategis untuk kebutuhan analisis lebih lanjut. OLAP memungkinkan pengguna memperoleh pengertian dan pengetahuan yang lebih mendalam mengenai berbagai aspek data perusahaan melalui akses data yang cepat, konsisten, dan interaktif. OLAP dapat membantu proses pengambilan keputusan untuk menanggapi kejadian yang akan datang.

Berikut ini adalah table perbandingan OLTP dan OLAP :

OLTP	**OLAP**
Digunakan untuk mendukung kegiatan transaksi sehari-hari	Digunakan untuk mendukung kegiatan Analisis
Menggunakan *view single* dimensi	Menggunakan *view multi* dimensi
Mendukung keputusan sehari-hari	Mendukung keputusan masa depan

Tidak bergantung pada OLAP	Bergantung pada data yang tersimpan dalam sistem OLTP
Melayani pengguna operasional	Melayani pengguna managerial
Operasi *query*-nya sederhana dan berulang-ulang	Operasi *query*-nya lebih rumit, bersifat *ad hoc*, dan tidak melibatkan operasi *update* data
Menggunakan data sehari-hari	Menggunakan data yang terangkum dalam data *cube*

Tabel 2.1 Perbedaan OLAP dan OLTP

2.2. Konsep *Data Warehouse*

2.2.1. Pengertian *Data Warehouse*

Menurut W.H. Inmon (2005, p389), " *A data warehouse is a collection of integrated, subject oriented databases designed to support the DSS function, where each unit of data is relevant to some moment in time*," yang mengandung pengertian bahwa *Data Warehouse* merupakan kumpulan *database* yang bersifat berorientasi subjek dan terintegrasi yang dirancang untuk mendukung sistem pengambilan keputusan (DSS), dimana setiap data berhubungan dengan suatu kejadian pada suatu waktu.

Menurut Turban Rainer Potter (2001,G-4), "*A data warehouse is a centralized repository of corporate data, needed mainly for internally-oriented decision support extracted from transaction*

processing system, corporate suppliers data and external database," yang berarti sebuah *data warehouse* adalah penyimpanan gabungan data secara terpusat, diperlukan khususnya untuk *decision support internal* yang diekstrak dari sistem proses transaksi, gabungan data pemasok dan *database* eksternal.

Menurut Ralph Kimball(2004), "*A data warehouse is a copy of transaction data specifically structured for querying and reporting*." yang artinya *data warehouse* adalah salinan (*copy*) dari data transaksi yang tersusun untuk laporan dan *query***.**

Dari definisi-definisi diatas dapat ditarik suatu kesimpulan yaitu *data warehouse* adalah suatu tempat penyimpanan data yang mengintegrasikan data-data perusahaan dari berbagai sumber ke dalam sebuah penyimpanan tunggal dimana pengguna dapat lebih mudah mengambil data, menghasilkan laporan, dan melakukan analisis, sehingga dapat mendukung pengambilan keputusan pihak eksekutif perusahaan.

Data-data dari operational *database* dan *external source* diekstrak, disaring, di-*summary* dan kemudian di-*load* ke *data warehouse*. Karena *data warehouse* mengandung data-data historis, maka seringkali aktivitasnya meliputi pengambilan kembali data-data yang telah benar-benar di rangkum (*summarized)*.

2.2.2. Karakteristik *Data Warehouse*

Karakteristik *data warehouse* menurut W.H. Inmon (2005, p31) dan Connolly (2005, p1047) adalah sebagai berikut:

1. *Subject-oriented*

   Data warehouse diatur sesuai dengan subjek-subjek utama perusahaan (seperti pelanggan, produk, dan penjualan), bukan sesuai dengan area aplikasi utama (seperti faktur pelanggan, pengontrolan stok, dan penjualan produk). Ini didasari dengan kebutuhan untuk menyimpan data yang mendukung keputusan, bukan data berorientasi aplikasi.

2. *Integrated*

   Data dalam *data warehouse* bersifat terintegrasi karena bersumber dari sistem-sistem aplikasi yang berbeda dalam perusahaan. Sumber data demikian sering tidak konsisten, misalnya karena berbeda format. Sumber data yang terintegrasi ini harus dibuat konsisten untuk memberikan data yang seragam pada para pengguna.

3. *Time-variant*

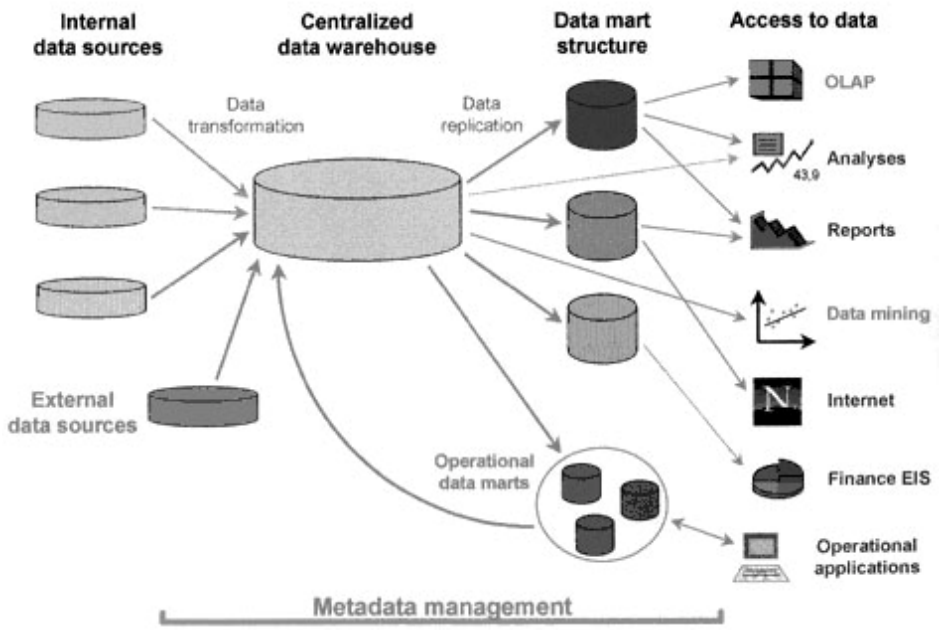
   Data dalam *data warehouse* bersifat tepat dan valid dalam jangka waktu tertentu. Data dalam *data warehouse* terdiri dari serangkaian *snapshot*, masing-masing menunjukkan data operasional yang diambil pada suatu waktu tertentu.

4. *Non-volatile*

   Data dalam *data warehouse* tidak di-*update* dalam *real time* melainkan diperbaruhi secara periodik dari sistem operasional. Data baru selalu ditambahkan sebagai tambahan bagi *data waerhouse*, bukan sebagai pengganti. *data waerhouse* secara terus menerus mengambil data baru, menambahkannya dan mengintegrasikannya dengan data sebelumnya.

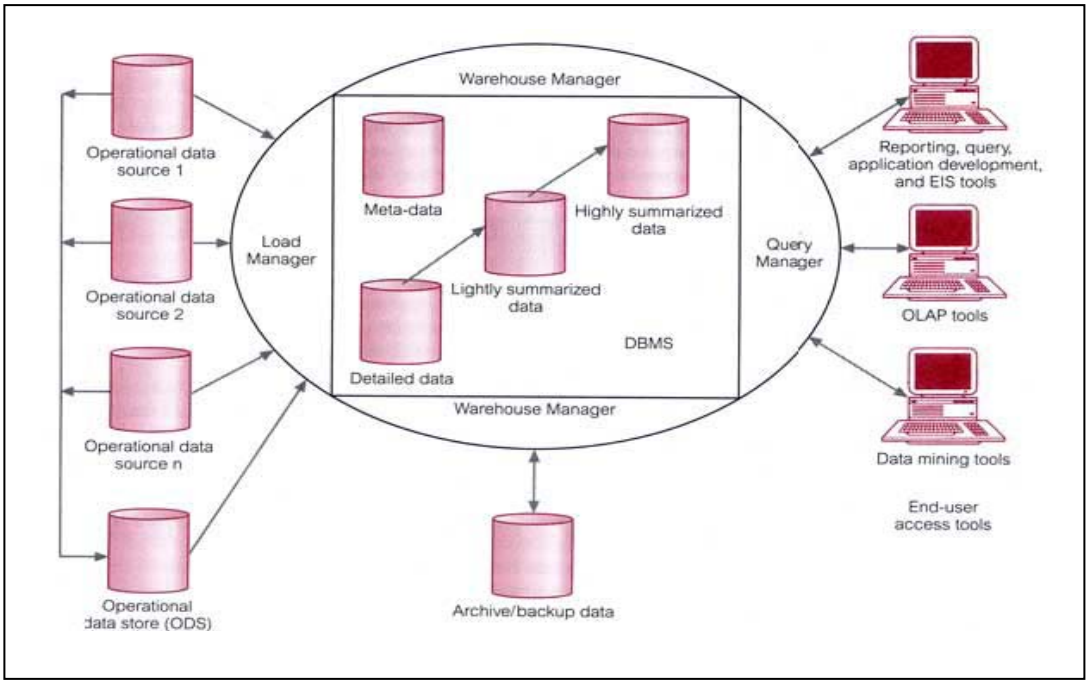
2.2.3. Anatomi *Data Warehouse*

Anatomi yang digunakan adalah *data warehouse terpusat*. *Data warehouse* terpusat merupakan pendekatan *data warehouse* dimana data diambil dari seluruh sistem yang ada kemudian disimpan di dalam pusat penyimpanan data, yang kemudian dapat digunakan untuk membuat *data warehous*e fungsional sesuai dengan kebutuhan. *User* yang membutuhkan data dapat langsung mengambil data dari kumpulan data tersebut, karena itu pendekatan terpusat ini merupakan pendekatan paling baik yang digunakan. Penggunaan pendekatan terpusat ini membutuhkan *cost* yang cukup besar dan membutuhkan waktu yang lama untuk membangun sistem tersebut.



Gambar 2.1 Data Warehouse Terpusat
Sumber: http://www.dmreview.com/editorial/dmdirect/080198_bhend_2.gif

2.2.4. Arsitektur *Data Warehouse*



Gambar 2.2 Arsitektur *Data Warehouse*
Sumber : Database Systems(Connoly,2005)

Komponen – komponen utama sebuah *data warehouse* antara lain:

1. *Operational Data*

   Data-data untuk *data warehouse* yang dapat bersumber dari :

   – *Mainframe* data operasional yang berada pada tingkatan *database* generasi pertama dan *database* jaringan.

   – Data departemen yang berada pada sistem file seperti VSAM, RMS , dan DBMS *Relational*

   – Data pribadi yang ada di server pribadi , bila ada.

   – Sistem-sistem *eksternal* seperti internet atau *database* yang berbungan dengan pemasok atau pelanggan perusahaan.

2. *Operational Data Store*

   Operational Data Store atau ODS merupakan sebuah tempat penyimpanan data operasional yang akan digunakan untuk kebutuhan analisis . ODS merupakan tempat penyimpanan sementara sebelum data dipindahkan ke *data warehouse*.

3. *Load Manager*

   Load Manager melakukan semua operasi yang berhubungan dengan mengambil data dan meletakkan data ke *data warehouse* . Data – data diambil atau diekstrak dari *operational datastore* .

4. *Warehouse Manager*

   Warehouse manager melakukan semua operasi yang berhubungan dengan manajemen data di dalam *data warehouse*. Operasi-operasi yang dilakukan antara lain :

   - Melakukan analisis data untuk menjaga konsistensi data.
   - Melakukan transformasi dan penggabungan data sumber dari ODS ke dalam-dalam tabel-tabel *data warehouse*.
   - Menciptakan indeks dan view pada *base tables*.
   - Melakukan denormalisasi (jika diperlukan).
   - Melakukan agregasi (jika diperlukan).
   - Melakukan *back-up* dan *archive* / penyimpanan data.

5. *Query Manager*

   Query manager melakukan semua operasi yang berhubungan dengan manajemen *user queries*. Komponen ini dibangun menggunakan *vendor end-user data access tools, data warehouse monitoring tools*, fasilitas *database*, dan *custom-built programs*. Kompleksitas *query manager* ditentukan oleh fasilitas yang disediakan oleh *end-user access tools* dan *database*. Operasi yang dilakukan oleh komponen ini berupa pengarahan *query* pada tabel-tabel yang tepat dan penjadwalan eksekusi *query*.

6. *Detailed Data*

Menyimpan semua data detil dalam skema *database*.Detil data ini dapat diperoleh dengan melakukan agregasi. Data detil ditambahkan ke dalam *data warehouse* secara periodik untuk mendukung agregasi data. Alasan perlu diperhatikannya *detailed data* adalah sebagai berikut:

a. Menggambarkan kejadian yang baru terjadi dan selalu menjadi perhatian utama.
b. Hampir selalu disimpan di media penyimpanan karena cepat diakses tetapi mahal dan kompleks dalam pengaturannya.
c. Dapat digunakan dalam membuat rekapitulasi sehingga *current detail data* harus akurat.

7. *Lightly and Highly Summarized Data*

Menyimpan semua data yang telah diringkas oleh *data warehouse manager* . Tujuan dari peringkasan data yang ada yaitu untuk mempercepat performa *query*. Setiap ada data baru yang masuk ke dalam *data warehouse* , ringkasan data akan terus diperbaharui.

Lightly summary data adalah ringkasan dari data tingkat rendah yang terdapat pada data detail yang sedang aktif (*current detail data*), tetapi belum bersifat *total summary* (ringkasan secara total). Data ini banyak ditampilkan dalam bentuk *view* dari kondisi yang sedang berjalan atau sudah berjalan.

Highly summary data adalah suatu hasil data yang bersifat *total summary* (ringkasan secara total). Data ini tersusun rapi dan mudah diakses terutama untuk melakukan analisis perbandingan data berdasarkan urutan waktu dan analisis yang menggunakan data multi dimensi. Sumber data tersebut dapat berasal dari data yang sedang aktif ataupun di luar *data warehouse.*

8. *Archive/Backup Data*

   Menyimpan detil data dan juga ringkasan data dengan tujuan untuk mem-*backup* data. Data-data yang di-*backup* kemudian dapat disimpan di media pemyimpanan data seperti *magnetic tape* atau *optical disc*.

9. *Meta Data*

   Meta data merupakan “data tentang data”. Yaitu data yang berisi informasi mengenai data-data yang terdapat pada *data warehouse* tersebut.

   Meta data dapat digunakan untuk :

   - Proses *extract* dan *loading data* , memetakan sumber data dalam *data warehouse*.
   - Proses dalam manajemen *data warehouse*. Meringkas data dari data detail menjadi *lightly summary data* dan dari *lightly summary data* menjadi *highly summary data*.

– Sebagai bagian dari proses manajemen *query*, mengarahkan *query* ke sumber data yang tepat. Suatu panduan pemetaan data pada saat data ditransformasi/diubah dari lingkup data operasional menjadi lingkup *data warehouse.*

10. End-User Access Tools

Merupakan tools yang digunakan oleh para pengguna untuk berinteraksi dengan *data warehouse*.

– *Reporting and Query Tools*

  Digunakan untuk menghasilkan laporan secara berkala, sedangkan query tools digunakan untuk menerima SQL atau menghasilkan *SQL statements* untuk proses *query* data yang ada di *data warehouse*.

– *Application Development Tools*

  Menggunakan *Graphical Access Tools* yang digunakan untuk *client server environment*. Aplikasi yang ada diintegrasikan dengan OLAP *tools* dan dapat mengakses semua sistem *database* utama.

– *Executive Information System(EIS) Tools*

  Suatu tools yang dikembangkan untuk mendukung keputusan *high level* yang strategis.

– *Online Analitical Processing(OLAP) Tools*

  Tools yang mengsumsikan bahwa data diatur dengan model multidimensi yang didukung dengan *Database* Multi Dimensi atau sebuah *relational database*.

– *Data Mining Tools*

  Tools untuk melalukan penggalian sejumlah data menggunakan teknik statistik, matematis, dan artificial intelligence.

2.2.5. Granularity

Menurut W.H. Inmon (2005, p43), " *Granularity refers to the level of detail or summarization of the units of data in the data warehouse*," yang mengandung pengertian bahwa *granularity* merupakan tingkat detail ringkasan dari data – data yang ada. Semakin detail ringkasannya maka semakin rendah tingkat *granularity* tersebut, semakin tidak detail ringkasannya maka semakin tinggi tingkat *granularity* yang ada. Hal yang perlu diperhatikan adalah tingkat *granularity* tidak boleh terlalu rendah maupun tinggi. Ketika *granularity* yang diperlukan sudah ditentukan maka desain dan implementasi akan berjalan dengan baik. *Granularity* akan berpengaruh bagaimana efisiennya data yang disimpan untuk sebagai pertimbangan analisis oleh manajemen suatu perusahaan.

2.2.6. *Data Flow* dalam *Data Warehouse*

Aliran data dalam *data warehouse* terdapat lima aliran data utama (Connolly ,2005 ,p1162), yaitu:

1. *Inflow*

   Inflow adalah proses penggabungan dengan *Extraction, Cleansing, and Loading data* dari sistem sumber ke *data warehouse.*

2. *Upflow*

   Upflow adalah proses penggabungan dengan penambahan nilai ke data di dalam *data warehouse* melalui ringkasan, *packaging*, dan distribusi data.

3. *Downflow*

   Downflow adalah proses penggabungan dengan arsip dan *back – up* data di dalam *data warehouse.*

4. *Outflow*

   Proses penggabungan dengan membuat data menjadi siap pakai bagi *end user*, alias informasi.

5. *Metaflow*

   Proses penggabungan dengan manajemen *meta data*.

2.2.7. Skema *Data Warehouse*

2.2.7.1.Tabel Fakta

Menurut Thomas Connoly (2005,p1183), "*Every dimensional model(DM) is composed of one table with a*

composite primary key, called the fact table," yang berarti tabel fakta adalah satu tabel pada *DM* yang isinya *composite Primary Key(PK).* Jadi *PK* pada tabel fakta merupakan beberapa *Foreign Key(FK).*

2.2.7.2.Tabel Dimensi

Menurut Thomas Connoly (2005,p1183), "*a set of smaller tables called dimesion tables*," yang berarti tabel dimensi adalah sekumpulan tabel-tabel yang lebih kecil dari tabel fakta pada *DM*. Setiap tabel dimensi mempunyai *non-composite PK.*

2.2.7.3.Pemodelan Dimensional

Menurut Thomas Connoly (2005,p1183), "*Dimensionality modelling is a logical design technique that aims to present the data in a standard, intuitive form that allows for high-performance access,"* yang berarti pemodelan dimensional adalah teknik desain logika yang bertujuan untuk menyajikan data dalam *standard,* bentuk intuitif yang mempunyai akses performa tinggi.

2.2.7.4.Skema Bintang

Skema bintang atau *Star Schema* adalah struktur logika yang mempunyai sebuah tabel fakta yang berisi

data faktual di tengahnya dan dikelilingi oleh tabel dimensi yang berisikan data referensi (dimana data bisa didenormalisasi).

Keuntungan memakai Skema Bintang adalah sebagai berikut :

a. Efisiensi, struktur relational skema bintang yang sederhana membuat data menjadi lebih efisien, sehingga membantu proses pengaksesan data menggunakan alat atau *tool* untuk menampilkan data termasuk yaitu laporan tertulis dan *query.*
b. Kemampuan untuk mengatasi perubahan kebutuhan, skema bintang dapat beradaptasi terhadap perubahan kebutuhan pengguna, karena semua tabel dimensi memiliki kesamaan dalam hal menyediakan akses ke tabel fakta. Artinya bahwa desain skema sebaiknya mampu mendukung *ad hoc query* dari pengguna.
c. *Extensibility,* model dimensional dapat dikembangkan. Seperti menambah tabel fakta selama data masih konsisten, menambah tabel dimensi selama ada nilai tunggal di tabel dimensi tersebut yang mendefinisikan setiap *record* tabel fakta yang ada, menambahkan attribute tabel dimensi, dan memecah *record* tabel dimensi yang

ada menjadi *level* yang lebih rendah dari *level* sebelumnya.

d. Kemampuan untuk menggambarkan situasi bisnis pada umumnya, pendekatan standar untuk menangani situasi umum didunia bisnis terus bertambah.

e. Proses *query* lebih mudah dilakukan, aplikasi *data warehouse* yang mencari data dari *level* yang dibawahnya akan dengan mudah menambah jumlah atribut pada tabel dimensi dari sebuah skema bintang. Aplikasi yang mencari data dari *level* yang setara akan menghubungkan tabel fakta yang terpisah melalui tabel dimensi yang dapat diakses bersama.

2.2.8. Keuntungan Penggunaan *Data Warehouse*

Keuntungan yang dapat diberikan oleh sebuah *data warehouse* adalah sebagai berikut :

1. Meningkatkan kemudahan *user* dalam mengakses data yang ada dengan cakupan yang lebih luas.
2. Membantu eksekutif perusahaan terutama di bagian manajemen untuk mengambil keputusan dari laporan-laporan yang ada.
3. Meningkatkan konsistensi data yang ada pada perusahaan.
4. Meningkatkan produktifitas pada pengolahan data.

5. Dapat mengintegrasikan atau mengkombinasikan data yang bersumber dari tempat atau cabang yang berbeda-beda.

2.3. Perancangan *Data Warehouse*

Dalam merancang *data warehouse*, perlu dilakukan analisis pada sistem yang sedang berjalan, fungsi bisnis yang ada di perusahaan dan kendala yang ada. Hal ini dilakukan untuk mendapatkan informasi mengenai data – data yang diperlukan dalam membangun aplikasi *data warehouse.* Analisis sistem dilakukan dengan cara menganalisis *database* yang ada apakah sudah ternormalisasi dengan benar dan data-data yang diperlukan tersedia.

2.3.1. Pengertian *Dimensionality Modelling*

Dimensionality modeling merupakan sebuah teknik perancangan logikal yang bertujuan untuk menampilkan data dalam bentuk standar dan memiliki akses performa tinggi. Setiap *Dimentionality Model* terbentuk dari satu tabel dengan *primary key* yang biasa disebut dengan tabel fakta, dan serangkaian tabel yang lebih kecil yang disebut tabel dimensi. Tiap tabel dimensi memiliki sebuah *primary key* yang berhubungan dengan sebuah komponen *composite key* pada tabel fakta. Dengan kata lain, *primary key* pada tabel fakta, terbentuk dari dua atau lebih *foreign key*. Struktur karakteristik yang menyerupai bintang ini disebut skema bintang.

Hal penting lainnya dari *Dimentionality Model* yaitu semua *natural key* digantikan dengan *surrogate key*, dimana setiap

penggabungan dengan tabel fakta harus menggunakan *surrogate key*, bukan *natural key*. Setiap *surrogate key* harus memiliki struktur umum berdasarkan *integer* yang sederhana. Penggunaan *surrogate key* memperbolehkan data pada *data warehouse* tidak memiliki ketergantungan dengan data yang digunakan dan diciptakan oleh sistem OLTP.

DM pada umumnya digunakan untuk mendesain komponen database pada *data warehouse*. Sedangkan ER digunakan untuk mendeskripsikan sistem database OLTP. Model ER merupakan sebuah teknik untuk mengidentifikasi hubungan antara entitas. Fungsi utama dari model ER adalah untuk menghilangkan perulangan data (*data redundancy*). Hal ini menguntungkan dalam pemrosesan transaksi, karena transaksi - transaksi tersebut menjadi lebih sederhana. Contohnya: Pada proses *update* nama produk, akan lebih mudah apabila hanya melakukan *update* untuk satu *record* pada tabel master daripada harus meng-*update* banyak record untuk setiap transaksi. Model ER tidak mendukung perancangan *data warehouse*, karena performanya kurang baik dalam pemrosesan data secara massal.

Kunci utama yang dapat digunakan dalam memahami hubungan antara ER dengan DM yaitu: sebuah model ER pada umumnya dibagi menjadi beberapa model DM. Model DM saling berhubungan melalui tabel-tabel dimensi.

2.3.2. Denormalisasi

Menurut Adelman (2004, p244) denormalisasi adalah suatu proses penggabungan tabel yang dilakukan dengan cermat dan hati-hati yang bertujuan untuk meningkatkan kinerja pengaksesan data. Pada dasarnya denormalisasi merupakan proses yang melanggar aturan *third normal form* (*3NF*).

Menurut Mannino (2001, p553) denormalisasi menggabungkan tabel-tabel sehingga dapat mempermudah dalam melakukan *query*. Denormalisasi merupakan kebalikan dari normalisasi. Denormalisasi berguna untuk meningkatkan kinerja *query*.

Keuntungan melakukan proses denormalisasi adalah

1. Mengurangi jumlah relasi yang terjadi antar tabel yang harus diproses pada saat pencarian, sehingga akan meningkatkan kecepatan proses *query* data.
2. Memetakan struktur fisik *database* agar mudah dimengerti oleh pengguna. Struktur tabel yang dibuat sesuai dengan hasil yang dikehendaki oleh pengguna, dan memungkinkan terjadinya akses langsung terhadap data.

Sedangkan kelemahan melakukan proses denormalisasi adalah:

1. Proses denormalisasi secara tidak langsung akan membuat redundansi data.
2. Proses denormalisasi memerlukan alokasi *memory* dan *storage* (tempat penyimpanan) yang besar.

2.3.3. Agregasi

Agregasi merupakan suatu proses penciptaan nilai yang diperoleh dengan cara menggabungkan (penjumlahan, pengambilan nilai minimum/maksimum, dsb) nilai kolom pada satu atau banyak *record* dalam *database* yang bertujuan untuk meningkatkan performa *query*, karena nilai yang ditampilkan merupakan gabungan dari banyak *record* yang ada dalam *database*.

2.3.4. *Nine-Step Methodology*

Metodologi perancangan *data warehouse* yang dikemukakan oleh Kimball adalah '*Nine-Step Methodology*' (Kimball, 2004). Pendekatan Kimball ini membagi perancangan *data warehouse* ke dalam beberapa bagian yang dapat diatur dengan mudah, disebut *data mart*. Kemudian integrasi dari semua *data mart* tersebut membentuk sebuah *data warehouse* perusahaan. Langkah-langkah dalam *Nine-Step Methodology* :

1. Menentukan proses *(Choosing the process)*

   Yang dimaksud dengan proses adalah subjek yang akan digunakan pada *data mart*. *Data mart* pertama yang perlu diutamakan yaitu *data mart* yang dapat memberikan jawaban terhadap permasalahan bisnis yang utama dan dapat diselesaikan dengan biaya yang terjangkau dalam waktu yang sesingkatnya. Pada umumnya *data mart* pertama yang perlu diutamakan adalah *data mart* yang berhubungan dengan fungsi bisnis penjualan.

2. Menentukan tingkatan kedetailan *(Choosing the grain)*

Memilih *grain* berarti menentukan data apa yang diwakilkan oleh sebuah tabel fakta. Dengan ditentukannya *grain* untuk tabel fakta, maka dapat diidentifikasikan dimensi - dimensi untuk tabel fakta tersebut. Keputusan *grain* untuk sebuah tabel fakta juga menentukan *grain* untuk masing-masing tabel dimensi.

3. Identifikasi dimensi yang ada *(Identifying and conforming the dimensions).*

Dimensi menentukan konteks informasi mengenai fakta - fakta di dalam tabel fakta. Dimensi yang ditentukan dengan benar dapat membuat *data mart* lebih mudah dimengerti dan digunakan. Dimensi diidentifikasi dengan data detil yang cukup untuk mendeskripsikan hal-hal misalnya produk, karyawan, dsb. Isi dimensi yang tidak lengkap dapat mengurangi kegunaan sebuah *data mart* bagi perusahaan. Jika terdapat dimensi yang sama pada dua *data mart*, maka *data mart* tersebut harus menggunakan satu dimensi yang sama. Hanya dengan cara ini, dua atau lebih *data mart* dapat berbagi dimensi pada aplikasi yang sama. Ketika sebuah dimensi digunakan pada dua *data mart* atau lebih, dimensi tersebut disebut '*conformed*'.

4. Memilih fakta fakta *(Choosing the facts)*

Grain dari tabel fakta menentukan fakta - fakta mana yang dapat digunakan dalam *data mart*. Semua fakta harus berupa angka dan dapat dijumlahkan.

5. Menyimpan penghitungan sebelumnya dalam tabel fakta *(Storing pre-calculations in the fact table*)

Setelah fakta terpilih, masing-masing fakta harus diperiksa kembali untuk menentukan apakah terdapat kemungkinan untuk melakukan *pre-calculation*. Jika ya, maka hasil dari perhitungan tersebut disimpan kedalam suatu kolom terpisah pada tabel fakta.

6. Mengatur tabel – tabel dimensi di sekitar tabel fakta (*Rounding out the dimension tables)*

Pada tahap ini, terdapat penambahan deskripsi teks sebanyak mungkin pada tabel dimensi. Deskripsi teks harus dapat dimengerti oleh pengguna. Kegunaan *data mart* ditentukan oleh ruang lingkup dan atribut pada tabel dimensi.

7. Memilih durasi *database (Choosing the duration of the database)*

Pada tahap ini ditentukan durasi transaksi yang dapat dicakup oleh sebuah tabel fakta. Di banyak perusahaan, terdapat kebutuhan untuk melihat data hingga dua tahun yang lalu. Untuk perusahaan seperti perusahaan asuransi, terdapat kebutuhan untuk melihat data hingga lima tahun yang lalu atau lebih. Pertumbuhan data yang sangat besar pada tabel fakta dapat menyebabkan dua buah masalah dalam perancangan *data warehouse*. Pertama, untuk sumber data yang lama, terdapat kesulitan dalam membaca dan

menginterpretasikan informasi yang terdapat pada media penyimpanan yang lama. Kedua, adalah sebuah keharusan untuk menggunakan versi lama dari dimensi yang penting, bukan versi barunya. Kejadian ini dikenal sebagai masalah '*slowly changing dimension'* yang akan dibahas di langkah kedelapan.

8. Melacak secara perlahan perubahan tabel dimensi *(Tracking slowly changing dimensions).*

Terdapat 3 jenis 'slowly *changing dimensions*' :

– Sebuah atribut dimensi yang berubah di-*overwritten* (nilai atribut yang lama diubah dengan yang baru)

Keuntungan :

Jenis ini merupakan cara yang paling mudah untuk menangani masalah '*slowly changing dimension'*, karena tidak perlu menyimpan atau mencari informasi yang lama.

Kerugian:

Semua sejarah mengenai data yang lama hilang. Dengan menggunakan metodologi ini, tidak memungkinkan untuk dilakukan pencarian sejarahnya. Informasi yang lalu mengenai data tidak diketahui.

– Sebuah atribut dimensi yang berubah menyebabkan sebuah *record* baru diciptakan di dalam dimensi

Keuntungan :

Jenis ini memungkinkan tersimpannya semua informasi yang lalu.

Kerugian:

Jenis kedua ini akan menyebabkan ukuran tabel tumbuh dengan cepat. Dengan tabel yang memiliki jumlah baris yang banyak, penyimpanan dan performa menjadi suatu hal yang perlu diperhatikan. Jenis kedua ini juga membuat proses ETL menjadi rumit.

- Sebuah atribut dimensi yang berubah menyebabkan sebuah atribut alternatif diciptakan, sehingga kedua nilai atribut yang lama dan baru dapat diakses secara bersamaan dalam suatu *record* dimensi yang sama.

  Keuntungan :

  Jenis ini tidak menyebabkan bertambah besarnya ukuran tabel, karena informasi baru di-*update*. Jenis ketiga ini memungkinkan tersimpannya sebagian informasi yang lalu.

  Kerugian :

  Jenis ketiga ini tidak dapat menyimpan semua informasi lalu bila suatu atribut diubah lebih dari sekali.

9. Menentukan prioritas *query* dan mode *query (Deciding the query priorities and the query modes).*

   Pada langkah ini, dipertimbangkan masalah perancangan secara fisik. Masalah perancangan fisikal yang mempengaruhi persepsi pengguna akhir mengenai *data mart* adalah kehadiran ringkasan atau agregasi data yang telah tersimpan, dan hal-hal

lainnya yang mencakup administrasi, *backup*, performa index, dan keamanan.

Pada akhir metodologi ini, telah dihasilkan sebuah rancangan *data mart* yang mendukung kebutuhan sebuah proses bisnis tertentu dan juga memungkinkan integrasi yang mudah dengan *data mart* lainnya untuk membentuk sebuah *data warehouse* perusahaan.

2.3.5. *Extraction Transformation Loading* (ETL)

Merupakan suatu cara bagaimana data dapat dimasukkan ke dalam *data warehouse*. Dimana caranya terbagi menjadi tiga tahap yaitu:

1. *Extract*

   Tahap *Extract* merupakan proses mengambil data dari sumber data. *Data warehouse* biasa mengambil data dari berbagai sumber yang terpisah. Dan setiap sistem dimana data tersebut diambil mungkin juga menggunakan format data yang berbeda. Format sumber data yang paling umum adalah *relational databases* dan *flat files*, tetapi dapat juga meng-*include non-relational database structure* seperti IMS atau struktur data yang lain seperti VSAM atau ISAM. Di bagian *Extraction* termasuk mengubah data ke dalam format untuk transformation processing.

2. *Transform*

Pada tahap ini mengaplikasikan beberapa *rules* dan *function* pada data-data yang telah di *extract* sehingga data siap untuk di *load*. Beberapa sumber data mungkin memerlukan sedikit manipulasi data . Dalam kasus lain diperlukan juga beberapa tipe – tipe transformasi seperti :

- Memilih hanya beberapa kolom yang di *load*.
- *Translating coded values*, biasa disebut *data cleansing*.
- Membuat lambang untuk nilai-nilai tertentu(*encoding free form values*) .contoh : *mapping* "*Male*" dan "1" dan "Mr" menjadi M.
- Membuat suatu hasil perhitungan baru.
- Menggabungkan data dari banyak sumber.
- Meringkas banyak baris data.
- Meng-*generate* nilai-nilai *surrogate key*.
- *Transposing* atau *pivoting* (mengubah *multiple columns* menjadi *multiple rows*).
- Memecah sebuah kolom menjadi banyak kolom.

3. *Load*

Pada tahap ini data di *load* ke *data warehouse*. Proses ini tergantung dari kebutuhan dari organisasi. Waktu dan *scope* untuk mengganti bergantung dari waktu yang tersedia dan kebutuhan dari bisnis.

2.3.6. *Data Transformation* Services (DTS)

Kebanyakan perusahaan memiliki berbagai format dan lokasi penyimpanan data yang berbeda. Ketika perusahaan akan melakukan *decision-making*, meningkatkan performa sistem atau *upgrade* sistem yang ada, data akan sering dipindahkan dari tempat yang satu ke tempat lainnya. DTS adalah seperangkat piranti yang bisa digunakan untuk *import*, *export* dan *transform* berbagai macam data antara satu atau lebih sumber data, contohnya *Microsoft SQL Server*, *Microsoft Excel* atau *Microsoft Access*.

2.3.7. Pengertian *Data Mart*

Menurut Peterson (2000, p54), *Data Mart* adalah kumpulan data yang lebih kecil dari data warehouse yang digunakan untuk melakukan analisis bisnis di satu divisi.

Data mart adalah sebuah bagian dari *data warehouse* yang mendukung kebutuhan sebuah departemen atau fungsi bisnis tertentu. Sebuah *data mart* menyimpan sebagian data dalam *data warehouse*, biasanya berupa data berisi ringkasan yang berhubungan dengan sebuah departemen atau fungsi bisnis tertentu.

Karakteristik yang membedakan *data mart* dan *data warehouse* antara lain :

a. Sebuah *data mart* berfokus hanya pada kebutuhan pengguna yang berhubungan dengan sebuah departemen atau fungsi bisnis.

b. *Data mart* biasanya tidak berisi data operasional yang mendetil, tidak seperti *data warehouse*.

c. Karena *data mart* berisi data yang lebih sedikit dibandingkan dengan *data warehouse*, *data mart* lebih mudah dimengerti.

Ada beberapa pendekatan untuk membangun *data mart*. Salah satunya adalah membangun beberapa *data mart* yang akan berkelanjutan untuk diintegrasikan menjadi sebuah *data warehouse*. Pendekatan lainnya adalah membangun infrastruktur *data warehouse* perusahaan pada saat yang bersamaan dibangun pula satu *data mart* atau lebih untuk memenuhi kebutuhan bisnis yang berlangsung.

2.3.8. *Analysis Services*

Analysis Services adalah *middle-tier server* yang digunakan untuk *On – Line Analytical Processing* (OLAP) dan *data mining*. Akses data yang cepat ke *data warehouse* disediakan oleh *Microsoft® SQL Server™ 2000 Analysis Services*. Data dari data warehouse di *extract*, diringkas, diatur, dan disimpan di dalam struktur multidimensional untuk mempercepat respon ke *query end user*. Hasil dari proses ini dinamakan dengan *cube*.

Cube merupakan satu set data yang dibangun dari *subset data warehouse* dan terorganisasi. *Cube* diringkas dalam struktur multidimensional yang dibatasi oleh *dimensions* dan *measures*. *Cube* menyediakan mekanisme yang mudah digunakan untuk mengambil data dengan cepat. *End user* menggunakan aplikasi klien untuk

menghubungkan ke *analysis server* dan melakukan *query* pada *cube* di server. Pada beberapa aplikasi , *end user* menggunakan *query* pada *cube* dengan memanipulasi *user interface control* yang kemudian mendeterminasikan isi dari *query*.

Kemampuan yang disediakan oleh *analysis services* yaitu :

- Mendesign, menciptakan, mengatur *cube* dari *data warehouse* dan menyediakan akses ke data OLAP.
- Mendistribusikan data di *cube* melewati banyak server untuk menyediakan kapasitas penyimpanan yang besar.
- Menciptakan *linked cubes* untuk mendistribusikan akses *end-user* ke informasi tanpa harus menduplikat data di *cube*.
- Menciptakan *cube* yang ter-*update* secara *real time* ketika data berubah.
- Menciptakan *cube* yang mengalamatkan kebutuhan bisnis secara spesifik.

2.3.9. Analisis Matriks

Analisis matriks dilakukan untuk mendapatkan informasi mengenai perusahaan, baik itu mengenai bagian – bagian dari organisasi, fungsi bisnis yang dilakukan, serta subjek data dalam perusahaan. Kemudian informasi ini dilanjutkan dengan menganalisis *critical success factor* perusahaan. Hasil dari analisis – analisis ini

diharapkan untuk membantu memahami kebutuhan informasi perusahaan.

Dalam analisis matriks dibutuhkan pemetaan antara elemen – elemen berikut, yaitu:

1. Unit organisasi, yaitu bagian dari organisasi yang memilki tugas khusus yang mendukung pencapaian tujuan dari organisasi.
2. Subjek data, merupakan kumpulan dari *entity* mengenai data yang disimpan.
3. Lokasi, yaitu tempat unit organisasi berada.
4. Fungsi bisnis, yaitu sekumpulan aktivitas yang mendukung salah satu aspek untuk mencapai misi perusahaan. Contohnya : marketing, produksi, dan distribusi.

2.4. Terminologi dalam Penjualan

2.4.1. Penjualan

2.4.1.1. Pengertian Penjualan (*selling*)

Menurut Mulyadi (2001, p202), penjualan terdiri dari transaksi penjualan barang atau jasa , baik secara kredit maupun secara tunai. Di dalam perusahaan yang bergerak di bidang distribusi, penjualan adalah proses penting dalam melaksanakan tujuan dari perusahaan untuk kemajuan perusahaan yang pesat.

Dari definisi diatas dapat ditarik suatu kesimpulan yaitu penjualan adalah proses yang terjadi antara penjual

dengan pembeli dimana terjadi suatu transaksi baik itu barang ataupun jasa oleh penjual kepada pembeli yang menguntungkan keduanya.

2.4.1.2. Sistem Penjualan

Fungsi yang terkait dalam sistem penjualan adalah:

1. Fungsi Penjualan

   Fungsi ini bertanggung jawab melayani kebutuhan barang pelanggan dengan mengisi faktur penjualan yang memungkinkan fungsi gudang dan pengiriman melaksanakan penyerahan barang kepada pelanggan.

2. Fungsi Gudang

   Fungsi ini menyediakan barang yang diperlukan oleh pelanggan sesuai dengan yang tercantum dalam tembusan faktur penjualan yang diterima dari bagian penjualan.

3. Fungsi Pengiriman

   Fungsi ini bertanggung jawab untuk menyerahkan barang yang kuantitas, mutu, dan spesifikasinya sesuai dengan yang tercantum dalam tembusan faktur penjualan yang diterima dari bagian penjualan. Selain itu fungsi ini juga bertanggung jawab untuk memperoleh tanda tangan dari pelanggan di atas faktur penjualan kartu kredit

sebagai bukti telah diterimanya barang yang dibeli oleh pelanggan.

4. Fungsi Akuntansi

   Fungsi ini bertanggung jawab untuk mencatat transaksi penjualan di dalam jurnal penjualan

2.4.2. Retur

2.4.2.1. Pengertian Retur

Retur adalah tindakan pengembalian barang yang dilakukan oleh konsumen dikarenakan ketidaksesuaian dengan yang dipesan sebelumnya atau adanya kecacatan produk atau menukar jenis barang yang telah dipesan dimana barangnya ditukar dengan barang lagi.

2.4.2.2. Sistem Retur Penjualan

Pengembalian barang oleh pelanggan harus diotorisasi oleh fungsi penjualan dan diterima oleh fungsi penerimaan. Fungsi yang terkait dalam melaksanakan transaksi retur penjualan adalah (Mulyadi, 2001, p226)

1. Fungsi Penjualan

   Fungsi ini bertanggung jawab atas penerimaan pemberitahuan mengenai pengembalian barang yang telah dibeli oleh pembeli. Otorisasi penerimaan kembali barang yang telah dijual tersebut dilakukan dengan cara

membuat memo kredit yang dikirimkan kepada fungsi penerimaan.

2. Fungsi Penerimaan

   Fungsi ini bertanggung jawab atas penerimaan barang berdasarkan otorisasi yang terdapat dalam memo kredit yang diterima dari fungsi penjualan.

3. Fungsi Gudang

   Fungsi ini bertanggung jawab atas penyimpanan kembali barang yang diterima dari retur penjualan setelah barang tersebut diperiksa oleh fungsi penerimaan. Barang yang diterima dari transaksi retur penjualan ini dicatat oleh fungsi gudang dalam kartu gudang.

4. Fungsi Akuntansi

   Fungsi ini bertanggung jawab atas pencatatan transaksi retur penjualan ke dalam jurnal umum dan pencatatan berkurangnya piutang dan bertambahnya persediaan akibat retur penjualan dalam kartu piutang dan kartu persediaan.

2.4.3. Pengertian Persediaan (*stock*)

Persediaan adalah persediaan barang dagangan, yaitu merupakan barang yang dibeli untuk tujuan dijual kembali (Mulyadi, 2001, p553).

Terdapat lima jenis konsep persediaan :

- Bahan baku(*raw-materials*).
- Komponen(*components*).
- Produk dalam proses pengerjaan(work in process).
- Barang jadi(*final goods*).
- Barang pasokan(*supplies*).

Sistem dan prosedur yang bersangkutan dengan sistem persediaan adalah :

1. Prosedur pencatatan produk jadi
2. Prosedur permintaan dan pengeluaran barang gudang
3. Sistem penghitungan fisik persediaan.

2.4.4. Pengertian *Stock Point*

Stock point adalah penjual barang ke *end user* dimana dapat berupa *mini market*, *hypermart*, *supermarket*, warung dan lain – lain.

2.4.5. Pengertian *End User*

End User atau biasa disebut dengan *customer* adalah para pemakai produk yang dijual oleh suatu pihak tertentu.

2.4.6. Pengertian *Salesman*

Salesman adalah seorang individual yang mengelola sebuah kelompok perwakilan penjualan(Jeff Madura, 2001, p179).