

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

Pada Sub bab ini berisi tentang teori-teori pendukung yang digunakan dalam proses analisis dan implementasi dalam tugas akhir ini.

##### **2.1.1 Pengertian Data**

Menurut Jogianto (1990), data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk jamak dari datum, berasal dari bahasa Latin yang berarti “sesuatu yang diberikan”. Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra. [2]

Menurut Immon, data adalah sebuah rekaman dari fakta-fakta, konsep-konsep atau intruksi-intruksi pada media penyimpanan untuk komunikasi, perolehan dan pemrosesan dengan cara otomatis dan presentasi sebagai informasi yang dapat dimengerti oleh manusia [1].

Menurut Laudon & Laudon, data adalah fakta-fakta yang mewakili kejadian-kejadian yang berlangsung dalam organisasi atau lingkungan fisik sebelum didata dan diatur kedalam bentuk yang dapat dipahami dan digunakan orang [2].

Dari dua teori tersebut dapat disimpulkan bahwa data merupakan kumpulan fakta yang secara umum tidak dapat digunakan karena belum diolah yang terdapat pada media penyimpanan untuk melalui pemrosesan sehingga dapat menyajikan informasi yang dapat dimengerti.

#### **A. Jenis Data Menurut Cara Memperolehnya**

Jenis data dapat diuraikan berdasarkan cara memperolehnya, yaitu :

##### **1. Data Primer**

Data Primer adalah secara langsung diambil dari objek penelitian oleh peneliti perorang maupun organisasi [5].

Contoh : Mewawancarai langsung penonton bioskop 21 untuk meneliti preferensi konsumen bioskop.

## **2. Data Sekunder**

Data sekunder adalah data yang didapat tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial [5].

## **3. Macam-Macam Data Berdasarkan Sumber Data**

Macam-macam data dapat dibedakan berdasarkan sumber data yang diperoleh yaitu :

### **1 Data Internal**

Data *Internal* adalah data yang menggambarkan situasi dan kondisi pada suatu organisasi secara internal [5].

Contoh : Data keuangan, data pegawai, data penjualan suatu perusahaan atau organisasi.

### **2 Data Eksternal**

Data *eksternal* adalah data yang menggambarkan situasi serta kondisi yang ada di luar organisasi [5].

Contohnya : Data jumlah penggunaan suatu produk pada konsumen tingkat preferensi pelanggan, persebaran penduduk dan lain sebagainya.

### **2.1.2 Pengertian Informasi**

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan saat ini atau saat mendatang [2]. Dapat disimpulkan bahwa :

1. Informasi bermula pada data.
2. Memberikan suatu nilai tambah atau pengetahuan bagi yang menggunakannya.
3. Dapat digunakan untuk pengambilan keputusan.

### 2.1.3 Konsep Basis Data

Basis data dapat didefinisikan sebagai himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Atau dapat juga didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersamaan sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan [3].

*“ Menurut Gordon C. Everest, Database atau basis data merupakan sebuah koleksi atau kumpulan dari data yang bersifat mekanis, terbagi, terdefinisi secara formal dan terkontrol. Pengontrolan dari sistem database tersebut adalah terpusat, yang biasanya dimiliki dan juga dipegang oleh suatu organisasi. “*

*“ Menurut Rogayah, dalam modulnya mengenai sistem basis data, mengatakan bahwa yang dimaksud dengan sistem basis data merupakan suatu sistem yang dapat menyusun dan mengelola record-record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pengguna untuk proses mengambil keputusan. “*

#### 2.1.3.1 Pengertian Basis Data

Basis Data (database) adalah kumpulan data-data yang umumnya menjabarkan aktivitas-aktivitas dari suatu atau lebih dari satu organisasi yang terkait [2]. Basis Data adalah sebuah cara untuk mendokumentasikan berbagai macam data yang kemudian di manajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan. Dalam basis data, data yang ada tidak hanya disimpan begitu saja melainkan akan dikelola dengan sistem pengaturan basis data yang sering disebut *Database Managemenet System (DBMS)*. Dengan begitu pengaksesan data dapat dilakukan dengan mudah dan cepat oleh pengguna.

Keuntungan basis data :

1. Mereduksi redudansi
2. Data dapat di-*share* antar aplikasi.
3. Standarisasi dapat dilakukan.
4. Batasan *security* dapat diterapkan.

5. Mengelola integritas data.
6. Independensi data.

### 2.1.3.2 Bahasa dalam Basis Data

Ada Beberapa Bahasa yang digunakan dalam basis data diantaranya adalah sebagai berikut :

1. *Data Definiton Language (DDL)*

*Data Definition Language (DDL)* ini adalah sekumpulan perintah-perintah yang terdapat dalam SQL yang berfungsi untuk menspesifikasikan atau menggambarkan suatu skema basis data. Dengan Bahasa ini hal-hal yang dapat dilakukan adalah membuat table baru, membuat indeks, mengubah struktur table, menentukan struktur penyimpanan table, dan lain-lain. Hal yang dasar yang dapat dilakukan DDL ini adalah menciptakan, mengubah, dan menghapus basis data.

2. *Data Manipulation Language (DML)*

*Data Manipulation Language (DML)* adalah sekumpulan perintah-perintah dapat dikatakan seperti Bahasa pemrograman atau sintak yang dapat digunakan untuk mengubah, memanipulasi, dan mengambil data pada basis data. Dapat melakukan aksi crud seperti menghapus, mengubah, menyisipkan dan mengambil data. DML dibagi menjadi 2 jenis yaitu :

- A. Prosedural

Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkan dari data tersebut.

- B. Non-Prosedural

Non-prosedural menuntut pengguna menentukan terlebih dahulu data apa saja yang diperlukan dan yang dibutuhkan, tetapi tidak perlu mendeskripsikan atau menyebutkan bagaimana cara data tersebut mendapatkannya.

3. *Transaction Control*

*Transaction Control* adalah bahasa pada basis data yang dapat mengatur transaksi dimana transaksi dilakukan oleh *Data Manipulation Language (DML)*. *Transaction Control* ini memiliki peran yang sangat besar untuk

menentukan suatu aktivitas dilakukan atau tidaknya perubahan-perubahan data yang terdapat dalam basis data. Contoh dari *transaction control* adalah perintah *commit* dan *rollback*.

### 2.1.3.3 DBMS (Database Management System)

DBMS (*Database Management System*) atau disebut dengan sistem manajemen basis data, berfungsi sebagai suatu sistem dalam pengolahan basis data sehingga menjadikan sebuah data menjadi sebuah informasi baru yang berguna. Sistem ini memungkinkan untuk menyusun, mengolah dan memperbaharui data yang terdapat dalam suatu basis data. DBMS memiliki kemampuan untuk mengolah data dalam jumlah yang besar dan memanipulasi data dengan cepat dan mudah. Tujuan utama dari DBMS adalah menyediakan cara untuk menyimpan dan mengambil informasi dari *database* dengan baik, nyaman, dan efisien [4].

### 2.1.3.4 RDBMS (Relational Database Management System)

RDBMS (*Relational Database Management System*) merupakan sekumpulan data yang saling berhubungan yang disimpan sedemikian rupa sehingga mudah diambil informasinya oleh pengguna.

Ada Tiga prinsip dalam RDBMS.

#### 1. *Data Definition*

Mendefinisikan data-data yang akan dibuat, seperti nama table dan *field* serta keterangan dari *field* tersebut.

#### 2. *Data Manipulation*

Melakukan manipulasi data yang sudah dibuat, seperti mengubah na a *field* dan keterangannya, dan meghapus *record*.

#### 3. *Data Control*

Bagian ini berkenaan pada mengendalikan data kepada siapa saja yang bisa melihat isi data.

### 2.1.3.5 Database Administrator

*Database Administrator* adalah orang memiliki kemampuan dalam melakukan *control* utama terhadap keseluruhan sistem basis data (mencakup data dan program) yang mempunyai fungsi sebagai berikut :

#### 1. Pendefinisian skema.

2. Pendefinisian struktur penyimpanan dan metode akses.
3. Modifikasi skema dan organisasi fisik.
4. Pemberian otorisasi bagi pengaksesan data
5. Mendefinisikan bagian basis data yang mana dapat diakses oleh seorang
6. Pengguna, termasuk operasi-operasu yang dapat dilakukan.
7. Spesifikasi batasan integrasi.

#### 2.1.4 Data Warehouse

*Data Warehouse* adalah sekumpulan informasi *logic* yang dikumpulkan dari berbagai operasional *database* yang berfungsi untuk mendukung aktivitas para analis dalam melakukan pengambilan suatu keputusan. Penyajian *data warehouse* sendiri memiliki sifat multidimensional data, dimana *data warehouse* terdiri dari beberapa layer, kolom dan baris. Layer yang terdapat pada *data warehouse* menunjukkan suatu informasi yang ditampilkan dan disampaikan pada dimensi yang berbeda.

*Data warehouse* juga dapat dikatakan sebagai wadah penyimpanan data histori yang berasal dari berbagai departemen, organisasi atau perusahaan yang bertujuan untuk mengolah data, dimana data yang diolah dapat menghasilkan sebuah informasi yang efektif dan efisien dimana data tersebut dapat digunakan dalam pengambilan suatu keputusan. *Data warehouse* merupakan sekumpulan dari *database* yang berbeda, *data warehouse* mengkombinasikan informasi dengan cara merangkum dan mengelompokan. *Data warehouse* merupakan metode dalam melakukan perancangan basis data yang dapat menunjang DSS (*Decision Support System*) dan EIS (*Executive Information System*).

*Data warehouse* juga bisa dikatakan sebagai sekumpulan data yang berorientasi subje, terintegrasi tidak dapat di *update*, memiliki dimensi waktu yang dapat digunakan untuk pengambilan suatu keputusan yang dilakukan oleh pihak pimpinan perusahaan. Berdasarkan definisi tersebut, maka *data warehouse* memiliki karakteristik [6]. Sebagai berikut :

1. *Subject Oriented* (berorientasi Subjek)

*Data warehouse* memiliki entitas tertinggi dalam suatu perusahaan, contoh subjek atau entitas tertinggi di dalam suatu perusahaan manufaktur yaitu di

penjualan, marketing, dan lain-lain. *Data warehouse* pun dirancang untuk membantu pengguna dalam mengambil sebuah keputusan. Contohnya untuk mengetahui tentang data penjualan pada perusahaan. Kita bisa membangun *data warehouse* yang berfokus pada penjualan.

2. *Integrity* (integrasi)

*Data warehouse* dikonstruksikan dengan cara mengintegrasikan sejumlah sumber data yang berbeda-beda. Data yang terintegrasi menyebabkan data tersebut lebih konsisten. Sehingga lebih mudah dipahami ketika akan membuat laporan dan laporan tersebut akan dijadikan acuan sebagai pengambilan suatu keputusan.

3. *Time-variant* (variasi waktu)

*Data warehouse* harus bisa menyimpan histori dari suatu data atau laporan dalam jangka waktu panjang. *Data warehouse* bisa berfokus pada perubahan setiap waktunya.

4. *Non-volatile*

Data yang ada dalam *data warehouse* tidak bisa di ubah. Melainkan datanya tetap dan tidak berubah-ubah.

*Data warehouse* dibuat untuk melayani pengguna (*analyst* dan pengambil keputusan). Sehingga *data warehouse* wajib dirancang sesuai dengan persyaratan [7] berikut :

1. Harus bisa memberikan kepuasan kepada setiap pengguna.
2. Memiliki *function* (fungsi) sendiri tanpa menggunakan OLTP *system*.
3. Menyediakan pusat tempat penyimpanan data yang konsisten.
4. Menjawab setiap *complex queries* dengan cepat.
5. Menyediakan berbagai analisis *tools* yang kuat, seperti OLAP dan *data mining*.

Sebagai besar *data warehouse* yang sukses selain memenuhi persyaratan di atas juga memiliki beberapa karakteristik [7] berikut:

1. Berdasarkan model dimensional.
2. Mengandung *historical data*.
3. terdiri dari *detailed* dan *summarized data*.

4. Tetap mempertahankan konsistensi data walaupun berasal dari sumber data yang berbeda.
5. Fokus dalam *single subject*, seperti penjualan, keuangan, atau inventaris.

Menurut W.H. Inmon, *data warehouse* merupakan kumpulan koleksi data yang mempunyai sifat berorientasi subjek, *integrated*, *time-variant*, dan bersifat tetap/tidak berubah (*non-volatile*) dari koleksi data dalam mendukung proses pengambilan keputusan manajemen [8].

Ada empat tingkat data di dalam arsitektur basis data yang dapat dilihat pada Gambar 2.3 yaitu sebagai berikut :

1. Tingkat Operasional

Pada tingkatan ini data yang akan diolah merupakan data yang digunakan dalam kehidupan sehari-hari, terperinci, dan berorientasi aplikasi.

2. Tingkat *Data Warehouse*

Merupakan kumpulan koleksi data yang mempunyai sifat berorientasi subjek, *integrated*, *time-variant*, dan *non-volatile*. *Data warehouse* merupakan kumpulan dari beberapa *data mart*.

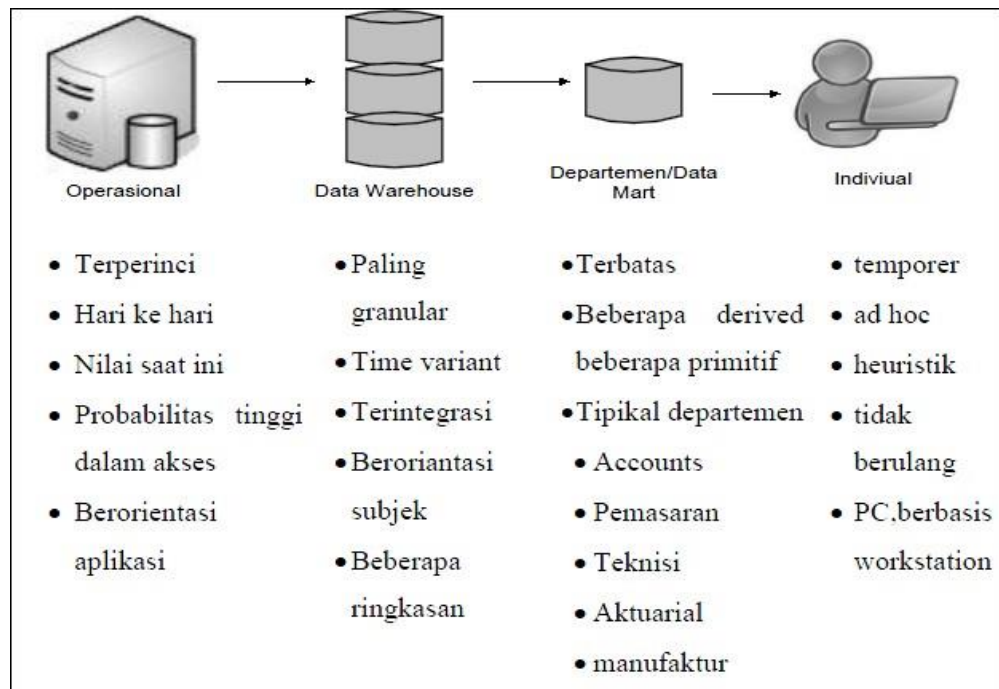
3. Tingkat Departemen/*data mart*

*Data mart* merupakan bentuk sederhana dari sebuah *data warehouse*, data marthanya fokus pada sebuah subjek tunggal atau suatu area fungsional seperti bagian penjualan, produksi, keuangan dan lainnya.

4. Tingkat individual

Data individual bersifat temporer dan hanya dalam ruang lingkup kecil. Didukung oleh perangkat komputer di mana proses EIS yang berlangsung hanya berjalan di dalam level individu.





**Gambar 2.1 Arsitektur Data Warehouse**

Data warehouse bisa dikatakan sebagai suatu salinan dari OLTP (*On-Line Transaction Processing*) yang terstruktur yang digunakan untuk memenuhi kebutuhan analisis, *reporting*, maupun *data mining*. OLTP sendiri adalah sebuah proses yang menitikberatkan pada transaksinya, seperti input data dan lain-lain. Pada Tabel 2.1 akan memperlihatkan perbedaan OLTP system dan *data warehouse*.

**Tabel 2.1 Perbedaan OLTP dan Data Warehouse [9].**

OLTP	Data Warehouse
Dirancang untuk operasi <i>real-time</i> bisnis	Dirancang untuk analisis dari suatu bisnis berdasarkan atribut dan kategori
Menangani data saat ini.	Menangani data saat ini dan data masa lalu
Data disimpan pada beberapa <i>platform</i> .	Data disimpan pada satu platform saja
Data diorganisir berdasarkan fungsi atau operasinya	Data diorganisir berdasarkan subjek.
Prosesnya bersifat berulang.	Prosesnya dilakukan setiap saat dan harus berorientasikan waktu (historical).
Untuk operasional.	Untuk managerial.
Berorientasi pada transaksi.	Berorientasi pada analisis.

#### 2.1.4.1 Tujuan Data Warehouse

Tujuan dari data *warehouse* adalah sebagai berikut [9] :

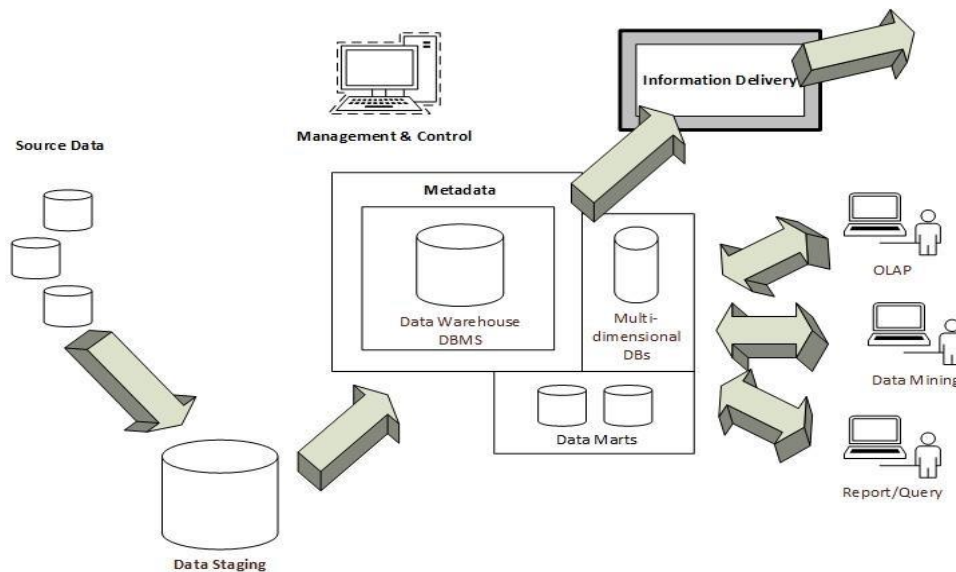
1. Memberikan kemudahan untuk mengakses informasi yang ada.  
Kemudahan disini berbicara tentang efisiensi. Data *warehouse* harus efisien sehingga dengan mudah dipahami oleh pengguna bukan hanya *olive developer* saja. Selain itu, pengguna juga dapat mengkombinasikan data dalam data *warehouse* dengan berbagai cara (*slicing and dicing*). Untuk mengakses data *warehouse* disarankan sebaiknya dapat dilakukan dengan sederhana dan mudah dioperasikan.
2. Menyediakan informasi yang konsisten.  
Data *warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan pengguna untuk mengambil suatu keputusan. Oleh karena itu, kredibilitas data yang terdapat dalam data *warehouse* harus dapat dipertanggungjawabkan.
3. Mampu beradaptasi dan tahan terhadap perubahan.  
Perubahan-perubahan yang terjadi harus dapat diatasi oleh data *warehouse*. Dengan kata lain, data *warehouse* harus dirancang agar mampu menghadapi setiap perubahan dengan terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.
4. Mampu mengamankan informasi.  
Informasi yang tersimpan dalam data *warehouse* harus tersimpan dengan aman. Dengan kata lain, informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Oleh karena itu, data *warehouse* harus mampu mengendalikan setiap akses dari informasi yang ada.
5. Mampu memberikan dukungan dalam pengambilan suatu keputusan.  
Ini merupakan tujuan yang paling penting dan harus ada dalam setiap pembuatan data *warehouse*. Data *warehouse* bisa digambarkan sebagai kumpulan teknologi pendukung suatu keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan sebuah informasi dapat membuat keputusan dengan cepat dan tepat.

6. *User friendly.*

Seperti pada tujuan data *warehouse* pertama, data *warehouse* harus dirancang agar dapat dioperasikan dengan mudah oleh pengguna. Tidak seperti sistem operasional dimana seringkali pengguna tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan terjadi pengguna data *warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan pengguna data *warehouse* merupakan factor yang sangat penting.

### 2.1.4.2 Arsitektur Data Warehouse

Arsitektur pada data *warehouse* dikelompokkan menjadi 5 bagian Gambar 2.2 [10].



**Gambar 2.2 Arsitektur Data Warehouse [10]**

1. *Source Data*

Merupakan sumber data, atau bisa dikatakan bahwa darimana data itu berasal. Untuk membangun sebuah data *warehouse* maka *source* data berasal dari operasional *system* atau *OLTP database*.

2. *Data Staging*

Merupakan sebuah proses yang diperlukan sebelum data *source* masuk ke dalam data *warehouse*. Proses ini dinamakan proses ETL (*Extracting, Transformation, Loading*).

### 3. *Data Warehouse*

Merupakan tempat penyimpanan data yang multidimensional, dimana data yang tersimpan berupa *metadata*, *summary* data dan *row* data.

### 4. *Data Mart*

Merupakan bagian dari sebuah data *warehouse*. Seluruh data *mart* jika digabungkan akan menjadi satu data *warehouse*.

### 5. *User*

Merupakan pengguna yang akan menggunakan data *warehouse*. *User* bisa memiliki tugas yang berbeda-beda, ada yang untuk dilakukan analisis, ada yang melakukan *reporting*, dan ada yang untuk membentuk data *mining*. Intinya informasi yang berasal dari data *warehouse* akan diolah menjadi “sesuatu” menggunakan *tools* yang ada dan berguna untuk pengambilan suatu keputusan.

#### **2.1.4.3 Dimensional Data Warehouse**

Kebutuhan pengguna dan realitas data yang menjadi factor penentu untuk merancang dimensional model data *warehouse*, seperti bisnis apa yang paling diperlukan, detailnya seperti apa dan dimensi-dimensi serta fakta-fakta apa yang harus dilakukan [6].

Maka dimensional model harus sesuai dengan kebutuhan dari pengguna. Model juga harus dirancang sedemikian rupa agar dapat bertahan dan dapat beradaptasi dari segala perubahan yang akan terjadi. Desain model yang dihasilkan dibentuk menjadi sebuah *database* relasional yang nantinya dapat mendukung OLAP *cubes* untuk menyediakan secara “instan” hasil *query* dimana hasil *query* tersebut digunakan untuk analisis.

#### 1. Tabel Dimensi (*Dimension Tables*)

Tabel dimensi menjelaskan tentang entitas bisnis dari suatu *enterprise* [6]. Tabel dimensi umumnya berisi data keterangan, dimana data tersebut jarang sekali mengalami perubahan.

#### 2. Tabel Fakta (*Fact Tables*)

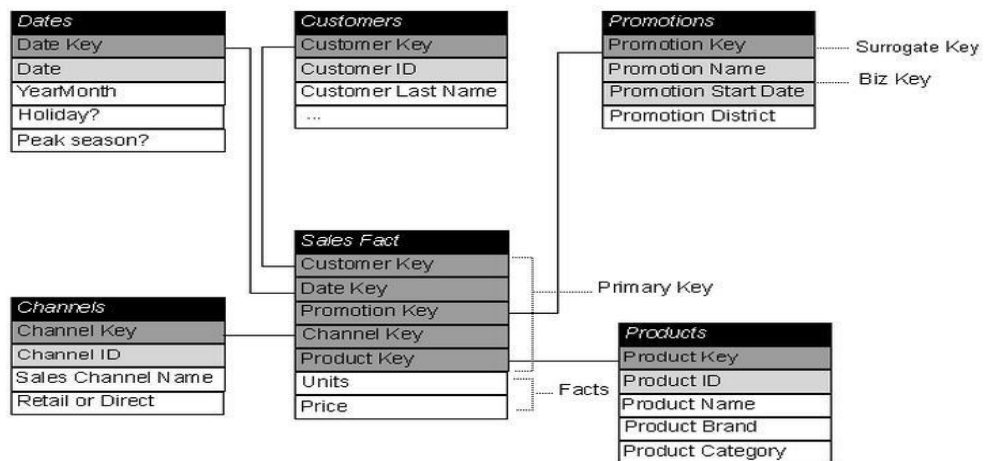
Tabel fakta merupakan sebuah table yang menjelaskan tentang transaksi bisnis dari suatu *enterprise* biasanya disebut table detail [6]. Tabel fakta umumnya berisi data yang berkaitan langsung dengan proses bisnis.

### 3. Skema *Dimensional Model*

Berikut ini adalah beberapa skema yang biasa digunakan dalam membangun sebuah data *warehouse* :

#### A) Skema *Star (Star Schema)*

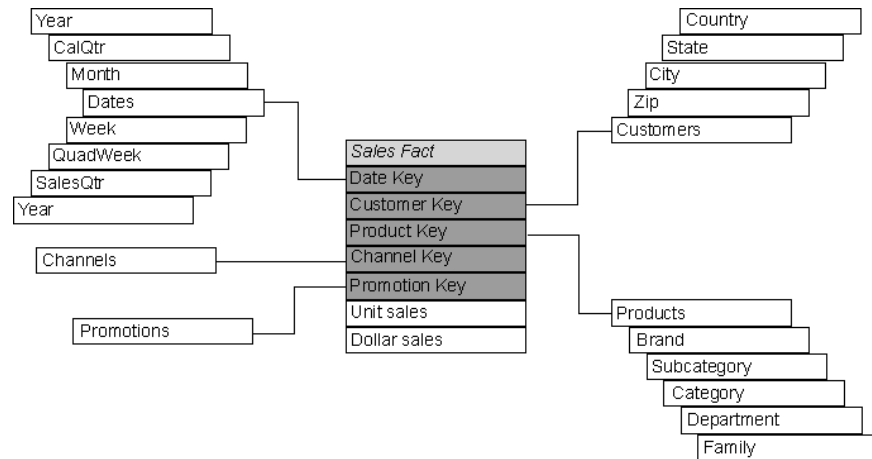
Suatu skema disebut skema *star* jika seluruh tabel dimensi dihubungkan secara langsung dengan tabel fakta dan satu tabel fakta wajib memiliki relasi minimal dengan satu buah tabel dimensi [6]. Gambar 2.3 dari skema *star*.



**Gambar 2.3 Skema Star [7].**

#### A) Skema *Snowflake*

Suatu skema disebut skema *snowflake* jika satu atau lebih tabel dimensi tidak berhubungan langsung dengan tabel fakta, tetapi harus berhubungan melalui dimensi lain [6]. Gambar 2.4 dari skema *snowflake*.

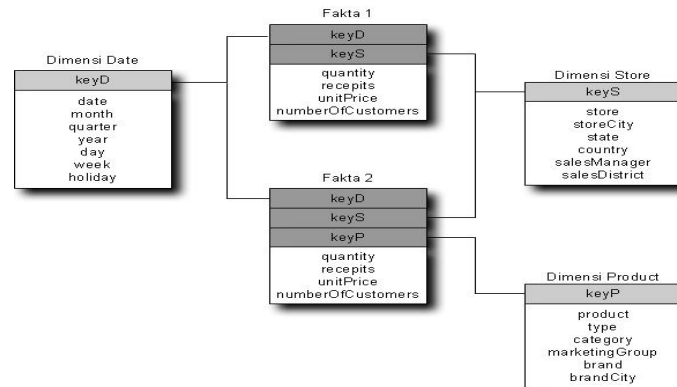


**Gambar 2.4 Skema Snowflake [7]**

B) Skema *Constellation*

Suatu skema dikatakan sebagai skema *constellation* jika ada satu tabel dimensi yang dipakai bersamaan oleh satu atau lebih tabel fakta [11].

Gambar 2.5 dari skema *constellation*.



**Gambar 2.5 Skema Constellation [11]**

### 2.1.5 Data Mart

Data mart merupakan suatu bagian dari data warehouse yang mendukung pembuatan laporan dan analisis data pada suatu unit. Data mart berisi informasi yang relevan bagi pengguna yang ingin mengambil keputusan.

Ada empat tugas yang dapat dilakukan dengan adanya data mart [10], empat tugas tersebut adalah sebagai berikut.

1. Pembuatan laporan

Pembuatan laporan merupakan salah satu kegunaan data mart yang paling umum dilakukan. Dengan menggunakan query sederhana didapatkan laporan per hari, per bulan, per tahun, atau jangka waktu kapanpun yang diinginkan.

2. On-Line Analytical Processing (OLAP)

Dengan adanya data mart, semua informasi baik detail maupun hasil summary yang dibutuhkan dalam proses analisa mudah didapat. OLAP merupakan konsep data multidimensi dan memungkinkan pada pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah SQL.

3. Data Mining

Data mining merupakan proses untuk menggali (mining) pengetahuan dan informasi baru dari data yang berjumlah banyak pada data mart.

4. Proses informasi executive.

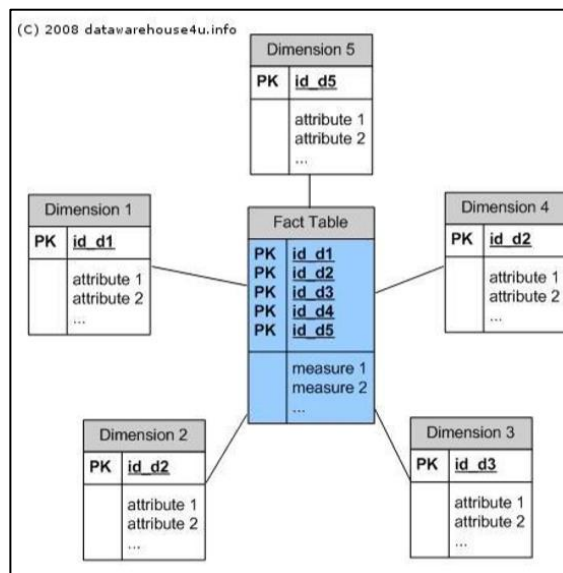
Data mart dapat membuat ringkasan informais yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan data mart segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan.

### 2.1.5.1 Model Dimensional Data Mart

Model dimensional pada data mart terdiri dari tabel fakta dan tabel dimensi, Ada beberapa model skema yang terdapat pada pemodelan data mart, yaitu skema star, skema snowflake, dan skema constellation. Penjelasan dari masing-masing model skema adalah sebagai berikut.

#### 1 Star Schema

Skema ini mengikuti bentuk bintang, dimana terdapat satu tabel fakta di pusat bintang dengan beberapa tabel dimensi yang mengelilinginya. Semua tabel dimensi berhubungan dengan ke tabel fakta. Tabel fakta memiliki beberapa primary key dalam tabel dimensi. Berikut adalah contoh skema bintang dapat dilihat pada Gambar 2.6.

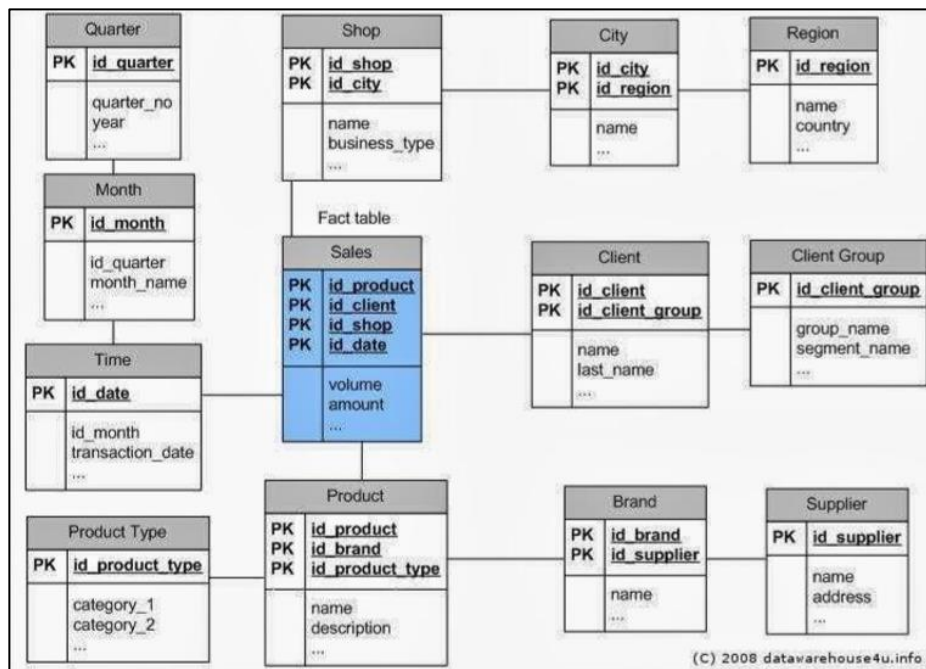


**Gambar 2.6 Star Schema**

#### 2 Snowflake Schema

*Snowflake Schema* merupakan perluasan dari *star schema* dengan tambahan beberapa tabel dimensi yang tidak berhubungan secara langsung dengan tabel fakta. Tabel dimensi tersebut berhubungan dengan tabel dimensi yang lain. Berikut adalah contoh skema bola salju dapat dilihat pada Gambar 2.7.

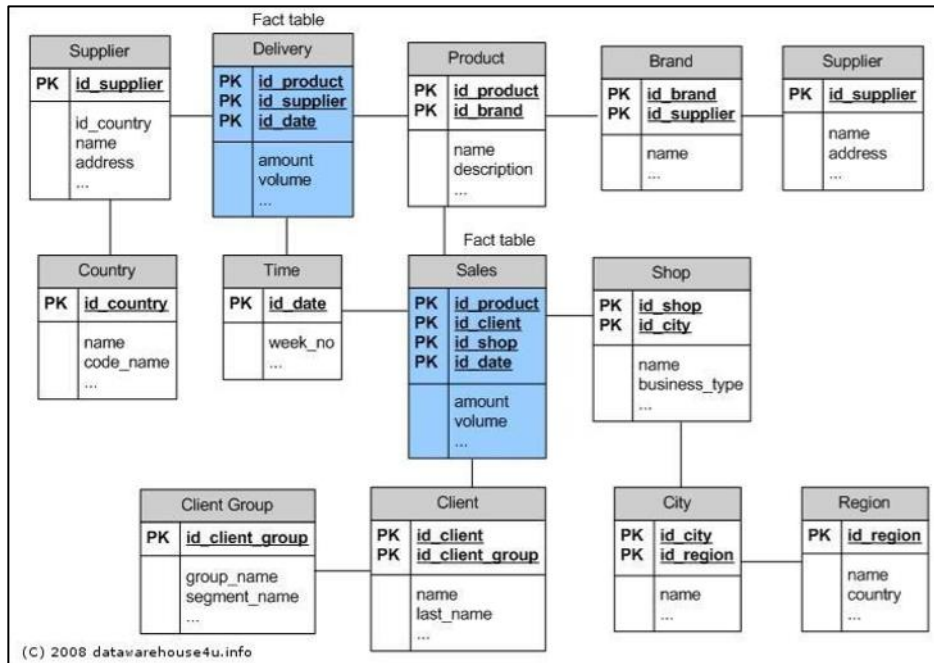




**Gambar 2.7 Snowflake Schema**

### 3 Skema Constellation

Pada skema ini terdapat beberapa tabel fakta yang menggunakan satu atau beberapa tabel dimensi. Berikut adalah contoh skema constellation dapat dilihat pada Gambar 2.8.



**Gambar 2.8 Skema Constellation**

### 2.1.5.2 ETL (Extract, Transform, Loading)

Proses ETL (Extract, Transform, Loading) merupakan proses yang harus dilalui dalam pembentukan data mart [11]. Tujuan ETL adalah mengumpulkan, menyaring, mengolah dan menggabungkan data dari berbagai sumber untuk disimpan ke dalam data mart. Berikut adalah penjelasan dari tiap proses ETL :

#### 1. Ekstraksi Data (Extract)

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

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

## 2. Transformasi data (Transformation)

Transformasi adalah proses dimana data hasil ekstraksi disaring dan diubah sesuai dengan aturan bisnis yang berlaku. Langkah-langkah dalam transformasi data adalah sebagai berikut.

- a. Memetakan data input dari skema data aslinya ke skema data *mart*.
- b. Melakukan konversi tipe data.
- c. Membersihkan dan membuang data yang sama (duplikasi).
- d. Memeriksa referensi data.
- e. Mengisi nilai-nilai yang kosong dengan nilai *default*.
- f. Menggabungkan data.

## 3. Pengisian Data (*Loading*)

Pengisian data adalah proses memasukkan data yang didapatkan dari hasil transformasi ke dalam data mart. Cara untuk memasukkan data adalah dengan menjalankan *SQL script* secara periodik.

### 2.1.5.3 OLAP (*On-Line Analytical Processing*)

OLAP (*On-Line Analytical Processing*) adalah teknologi yang memproses data ke dalam struktur multidimensi, menyediakan jawaban yang cepat untuk *query* analisis yang kompleks dengan tujuan untuk mengorganisir sejumlah data yang besar, agar bisa dianalisa dan dievaluasi dengan cepat serta menyediakan kecepatan dan fleksibilitas untuk mendukung analisa dalam waktu nyata [6].

Ada beberapa karakteristik OLAP, yaitu :

1. Mengizinkan pelaku bisnis melihat data dari sudut pandang logical dan multidimensional pada *data warehouse*.
2. Memfasilitasi *query* yang kompleks dan analisa bagi pengguna.
3. Mengizinkan pengguna melakukan *drill-down* untuk menampilkan data yang lebih detail atau *roll-up* untuk agregasi dari suatu dimensi atau beberapa dimensi.
4. Menyediakan proses kalkulasi dan perbandingan data.
5. Menampilkan hasil dalam bentuk tabel atau grafik.

Keuntungan dari OLAP, yaitu :

1. Meningkatkan produktivitas pemakai akhir bisnis, pengembang IT, dan keseluruhan organisasi.
2. Pengawasan yang lebih dan akses tepat waktu terhadap informasi strategis dapat membuat pengambilan keputusan lebih cepat.
3. Mengurangi pengembangan aplikasi bagi staf IT dengan membuat pemakai akhir dapat mengubah skema dan membuat model sendiri.
4. Penyimpanan pengawasan organisasi melalui integritas data korporasi sebagai aplikasi OLAP tergantung pada *data warehouse* dan sistem OLTP untuk memperbaharui sumber tingkatan data.

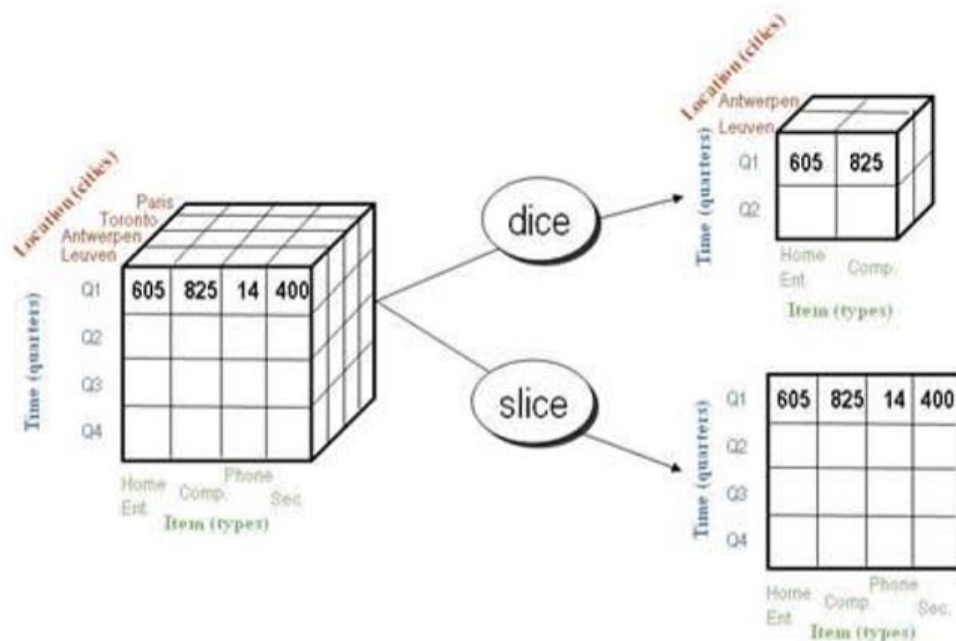
OLAP dapat digunakan untuk melakukan seperti [6] :

1. Konsolidasi (*roll-up*)

Konsolidasi melibatkan pengelompokan data

2. *Drill-down*

Suatu bentuk yang merupakan kebalikan dari konsolidasi untuk menjabarkan data yang ringkas menjadi data yang lebih detail. Gambaran untuk roll-up dan drill-down dapat dilihat pada Gambar 2.9.



**Gambar 2.9 Slicing dan Dicing**

#### 2.1.5.4 Fact Table and Dimension Tables

Tabel fakta adalah tabel utama dalam tabel dimensi dimana pengukuran kinerja numerik dari bisnis di simpan. Sebuah baris dalam tabel fakta sesuai dengan pengukuran, sebuah pengukuran adalah baris dalam tabel fakta [9]. Berikut contoh tabel fakta :

Daily Sales Fact Table
DateKey (FK)
Product Key (FK)
Store Key (FK)
Quantity Sold
Dollar Sales Amount

Tabel fakta tidak menyimpan informasi tekstual berlebihan dalam tabel fakta kecuali teks unik untuk setiap baris tabel fakta, itu termasuk dalam tabel dimensi.

Tabel dimensi adalah integral dari tabel fakta, tabel dimensi berisi deskripsi tekstual dari bisnis. Atribut tabel dimensi sangat berperan penting dalam pembuatan laporan, tabel dimensi adalah kunci untuk membuat data warehouse, atribut dimensi produk akan mencakup deskripsi singkat (10 sampai 15 karakter). Berikut contoh tabel dimensi :

Product Dimension Table
Product Key (PK)
Product
Description
SKU Number (Natural Key)
Brand Description
Category Description
Department
Description Package
Type Description
Package Size
Fat Content
Description Diet Type
Description Weight
Weight Units of Measure
Storage Type
Shelf Life
Type Shelf
Width Shelf
Height Shelf
Depth
... and many more

## 2.1.6 Tahapan Pembentukan Data *Mart*

### 2.1.6.1 Analisis Sumber Data

Ada beberapa data yang dapat dijadikan sebagai sumber data masukan untuk data *mart* di antaranya :

#### A. Data Transaksional

Data transaksional adalah data yang dihasilkan dari suatu proses bisnis transaksi yang dilakukan oleh perusahaan. Contohnya : transaksi penjualan barang.

#### B. Data Internal

Data internal yaitu data-data yang menggambarkan situasi dan kondisi pada suatu organisasi secara intern. Misalkan data keuangan, data pegawai, data produksi, dan sebagainya.

#### C. Data History

Data history yaitu data-data yang berasal dari masa lampau yang di simpan di suatu penyimpanan.

#### D. Data External

Data yang menggambarkan situasi dan kondisi yang ada di luar organisasi. Contoh persebaran pelanggan, produk, dan sebagainya.

### 2.1.6.2 Data Staging

Data *Staging* yaitu tempat dan area dengan sekumpulan fungsi untuk membersihkan, merubah, mengkombinasikan, penggandaan dan menyiapkan sumber data untuk penyimpanan dan digunakan pada data *mart*. Pada data *staging* terdapat suatu proses yang disebut proses ETL (*Extract, Transform dan Load*).

#### A. *Extraction*

*Extraction* merupakan proses untuk mengidentifikasi seluruh sumber data yang relevan dan mengambil data dari sumber data tersebut. Penentuan pendekatan yang digunakan pada proses ekstraksi sangat terkait dengan analisis bisnis proses, pendefinisian area subjek , serta desain logik/fisik data *mart*. Proses ini meliputi penyaringan data yang akan digunakan dalam pembuatan data *mart*. Dapat langsung dimasukkan langsung dalam data *mart* atau dimasukkan dalam tempat penampungan sementara terlebih

dahulu. Pada hakikatnya bagian dari ekstraksi melibatkan penguraian dari data yang telah diekstrak, menghasilkan suatu pengecekan jika data bertemu dengan suatu struktur atau pola yang diharapkan. Jika bukan, data tersebut mungkin ditolak secara keseluruhan. Adapun fungsi ekstraksi diantaranya, yaitu :

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

#### B. *Transformasi*

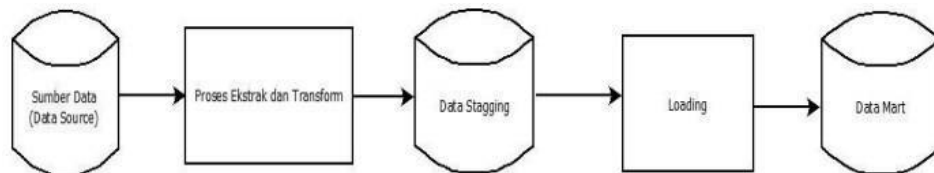
*Transformasi* data yang telah diekstrak ke dalam format yang diperlukan. Hal ini perlu dilakukan mengingat data yang diambil berasal dari sumber yang berbeda yang kemungkinan memiliki standardisasi yang berbeda pula. Data dari beberapa sistem perlu ditransformasi ke dalam format umum yang disepakati dan digunakan dalam data *mart*. *Transformation* merupakan proses yang mempunyai peran dalam melakukan perubahan dan integrasi skema serta struktur yang berbeda-beda ke dalam skema dan struktur yang terdefinisi dalam data *mart*.

Ada beberapa tahapan dalam proses *transformasi* diantaranya sebagai berikut:

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

#### C. *Loading*

*Loading* merupakan proses pemindahan data secara fisik dan sistem operasional ke dalam data *mart*. Tahap load adalah men-load data ke dalam target akhir (end target), yang pada umumnya adalah data *mart*. Bergantung pada kebutuhan organisasi, proses ini bervariasi secara luas.



**Gambar 2.10 ETL Proses**

### 2.1.6.3 Online Analytical Processing (OLAP)

*Online Analytical Processing* (OLAP) merupakan salah satu *tools* yang digunakan untuk mengakses informasi dalam data *mart*. Teknologi OLAP memungkinkan data *mart* digunakan secara aktif untuk proses *online analisis*, memberikan respon yang cepat terhadap *analytical queries* yang kompleks.

Data model multidimensional dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar sehingga dapat di evaluasi secara cepat dengan menggunakan *online analysis* dan *grafical tool*.

Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk men- support proses analisis secara *real time* [19].

### 2.1.6.4 OLAP Service

OLAP *services* adalah sesuatu yang menggambarkan beberapa perangkat yang bekerja bersama-sama untuk membantu dalam menganalisa data yang ada. Informasi dari data *warehouse* atau data *mart* di ekstrak secara periodik dan digunakan untuk mengupdate objek yang ada dalam OLAP *service*. OLAP *server* mengambil data dari data *warehouse* atau data *mart* dan *relational OLAP* serta mengupdate informasi yang disimpan dalam *multidimensional OLAP* (MOLAP). Setelah informasi dikumpulkan maka *pivot table service* yang bekerja sama dengan *excel pivot table* atau perangkat lain yang mendukung OLE DB dengan ekstensi OLAP dapat melakukan akses dan memanipulasi data yang ada [19].



### 2.1.6.5 Teknik penyimpanan data dalam OLAP

OLAP digunakan untuk memproses informasi dan menampilkannya dalam bentuk multidimensi. Walaupun data yang ditampilkan dalam satu bentuk namun tidak berarti data-data yang ada disimpan dalam satu bentuk pula. Ada tiga teknik dasar untuk menyimpan data OLAP [19]

#### A. Multidimensional OLAP (MOLAP)

Salah satu cara umum yang digunakan untuk menyimpan data yaitu di dalam basis data multidimensional. Tidak seperti basis data relasional yang menyimpan informasi dalam sejumlah baris dalam tabel. Sebuah basis data multidimensional menyimpan informasi dalam sejumlah *query* multidimensional. Karena dimensi dapat diakses secara mudah, maka pengguna dapat melakukan *query* pada basis data MOLAP dengan sangat cepat. Selain mengandung data mentah, basis data MOLAP pun mengandung agregasi data sehingga dapat memberikan respon yang cepat terhadap *query*.

#### B. Relational OLAP (ROLAP)

Partisi menggunakan tabel *relational* dalam data *mart* untuk menyimpan agregasi, sedangkan detail dari *fact table* tersimpan dalam data *mart fact table*. Banyak orang berpendapat bahwa basis data yang dirancang secara khusus untuk sebuah keperluan analisis tidak dibutuhkan karena sebuah basis data relasional sudah cukup mampu untuk menampilkan informasi OLAP. Hal ini hanya berlaku pada tingkat tertentu saja, pada sebuah basis data yang terdiri dari ribuan atau ratusan ribu *record* maka menampilkan informasi OLAP akan menjadi sebuah masalah karena banyak data yang harus dihapus di *query*. Dan hal inilah yang menjadi keterbatasan partisi ROLAP.

#### C. OLAP *Cube* (Kubus OLAP)

Objek utama yang tersimpan dalam sebuah basis data OLAP adalah *cube* (kubus). Sebuah kubus merupakan representasi multidimensi dari sekumpulan data, yang mengandung data secara detail maupun rangkumannya. Sebuah basis data OLAP dapat memiliki beberapa buah

kubus sesuai dengan yang dibutuhkan, yang menggambarkan data yang ada dalam data *mart*.

Sebuah kubus dibangun menggunakan dua komponen utama yaitu ukuran (*measures*) dan dimensi. Ukuran merupakan nilai numerik dari *fact table* dalam data *mart* seperti harga dari unit maupun kualitas dari item. Sedangkan dimensi menggambarkan kategori dari ukuran yang ada. Seperti bagaimana ukuran berubah setiap waktu. Beberapa operasi yang dapat dilakukan pada kubus atau *report* multidimensi ini adalah :

#### **2.1.6.6 Teknik Dalam OLAP**

1. *Slice and Dice*

*Slice and Dice* memungkinkan untuk melihat kubus dari sudut pandang yang berbeda. Dengan *slice and dice* dapat ditentukan dimensi apa yang akan ditampilkan dan bagaimana mereka ditampilkan. Hal inilah yang menjadi keunggulan OLAP. Dengan melihat kubus dari sudut pandang yang berbeda maka akan dapat dipelajari banyak hal dari data yang dimiliki.

2. *Drill Down*

Sebagian besar informasi yang akan ditampilkan dalam OLAP merepresentasikan kesimpulan yang lebih detail. *Drilling Down* merupakan teknik untuk memecahkan sebuah informasi menjadi beberapa informasi yang lebih detail lagi. Sebagai contoh, jika dilakukan *drill down* terhadap data tahunan maka akan dapat diperoleh data dalam catur wulan dan tri wulan.

3. *Consolidation (Roll-Up)*

*Consolidation* atau lebih dikenal dengan *roll up* merupakan kebalikan dari *drill down*. Data-data sebelumnya dilihat dalam format triwulan akan dapat dilihat dalam format tahunan. Dengan *roll up* data dapat dilihat secara lebih global (*Summary*).

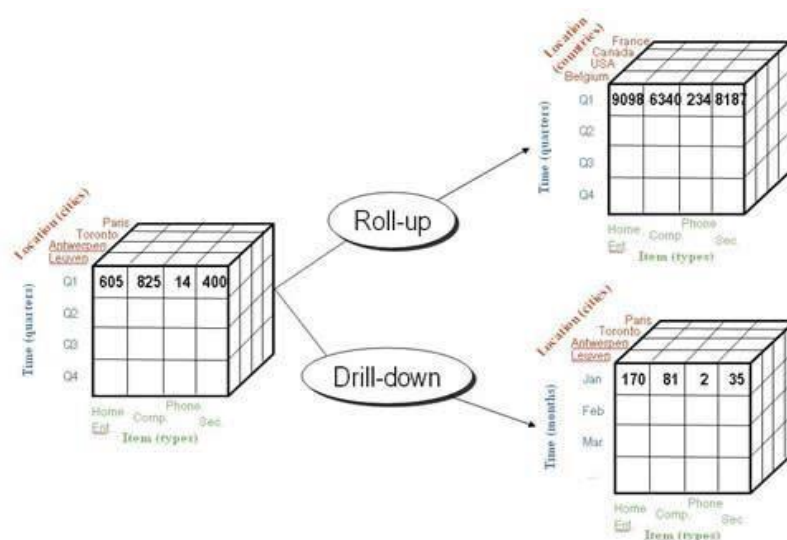
4. *Pivoting*

*Pivoting* merupakan suatu teknik untuk saling menukarkan suatu dimensi data. Dengan melakukan *pivoting*, maka dapat diamati suatu informasi atau

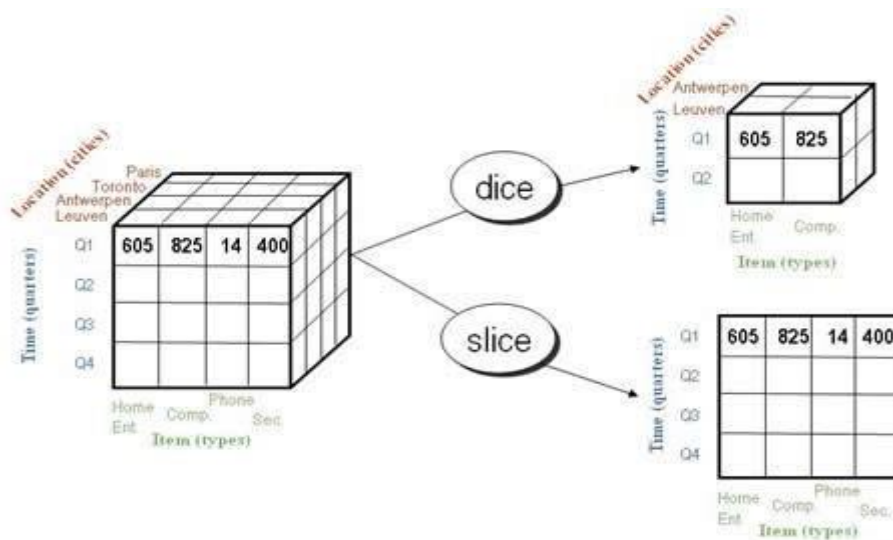
data dari sudut pandang yang berbeda, sehingga diharapkan akan dapat memperjelas analisis yang dilakukan.

### 5. Filtering

*Filtering* merupakan suatu teknik untuk menyaring informasi yang dibutuhkan, sehingga para analisis tidak digabungkan oleh banyaknya informasi yang tersedia.



**Gambar 2.11 Roll-Up Dan Drill Down**



**Gambar 2.12 Slice And Dice**

### 2.1.7 Visualisasi Data

Bentuk visualisasi data terdapat tiga macam yaitu *bar chart*, *pie chart*, dan *line char*. Berikut ini deskripsi dari masing-masing bentuk visualisasi data :

#### 1. *Pie Chart*

*Pie Chart* adalah sebuah tampilan deskriptif sederhana dari data, dimana data yang ditampilkan hanya jumlah dari total yang diberikan. *Pie Chart* mungkin adalah sebuah cara paling ilustratif untuk menampilkan kuantitas sebagai presentasi dari total yang diberikan. Total area dari sebuah *pie chart* mempresentasikan 100% dari kuantitas (jumlah dari nilai variabel pada seluruh kategori). Ukuran dari setiap potongan adalah presentasi dari total yang dipresentasikan oleh masing-masing kategori yang dapat dilihat pada Gambar 2.15 berikut:

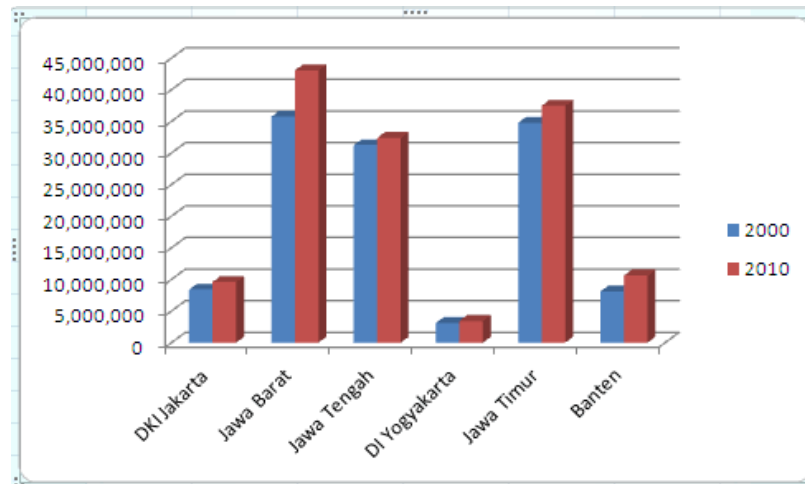


**Gambar 2.13** Contoh *Pie Chart*

#### 2. *Bar Chart*

*Bar chart* sering digunakan untuk menunjukkan data berdasarkan kategori tertentu dimana tidak ada penekanan total presentase pada setiap kategori. *Bar chart* dapat disajikan secara vertikal maupun horizontal. Skala pengukuran adalah nominal atau ordinal. *Bar chart* dapat digunakan untuk menampilkan data kontinu seperti ukuran sepatu atau warna mata dan data diskontinu seperti

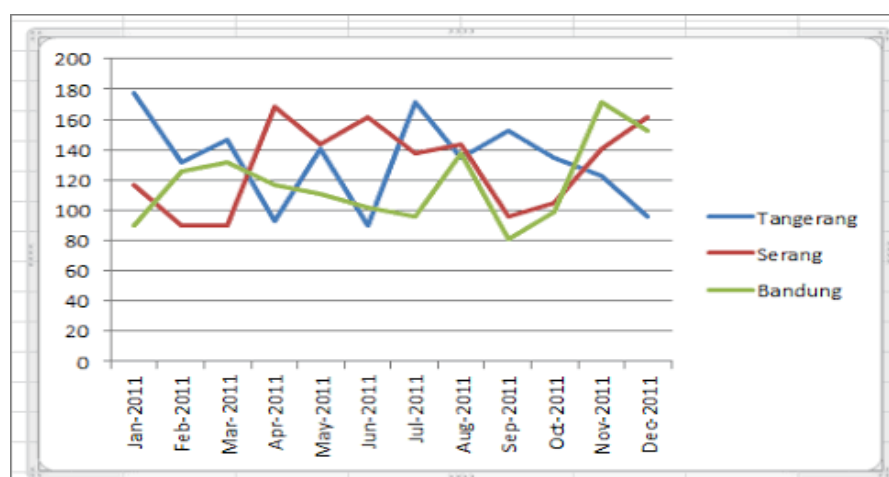
tinggi badan atau berat badan. Contoh dari bar chart dapat dilihat pada Gambar 2.16 berikut:



**Gambar 2.14 Contoh Bar Chart**

### 3. Line Chart

*Line chart* sering digunakan untuk menampilkan informasi dalam rangkaian titik data yang dihubungkan dengan segmen garis lurus. *Line chart* sering digunakan untuk memvisualisasikan trend data dalam interval waktu atau dalam kurun waktu tertentu. Contoh dari *line chart* dapat dilihat pada Gambar 2.13 berikut:



**Gambar 2.15 Contoh Line Chart**



### **2.1.8 Microsoft SQL Server**

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa query utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di sunia bisnis yang memiliki basis data yang berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar. Pengguna yang menggunakan SQL Server dapat menyimpan banyak data dan dapat untuk diimplementasi untuk kepentingan bisnis dan perusahaan [7].

### **2.1.9 SSIS (SQL Server Integration Service)**

SSIS (SQL Server Integration Services) adalah suatu platform untuk membangun sistem yang handal untuk integrasi data, extraction, transformation, dan loading yang digunakan pada data warehousing. SSIS menawarkan solusi dalam menghadapi permasalahan data integrasi. Selain itu, tools ini membantu untuk meningkatkan efisiensi waktu pembuatan [8].

Arsitektur SQL Server Integration Services secara umum berisi berbagai macam komponen, diantaranya yaitu :

1. SSIS Deginer. Merupakan tools yang digunakan untuk membuat dan mengatur paket integration service. Pada SQL Server 2012, tools ini sudah terintegrasi dengan Visual Studio 2010 yang merupakan bagian project Bussiness Intelegence.
2. Runtime Engine. Komponen ini berguna untuk menjalankan semua paket SSIS yang sudah dibuat.
3. Task dan executable binary.
4. Data Flow Engine dan Data Flow. Komponen data flow merupakan enkapsulasi data flow engine yang menyediakan buffer didalam memori dan bertugas memindahkan data dari sumber data ke tujuan data. Sedangkan data flow merupakan sumber data, tujuan data, dan transformasi.
5. Integration Services service. Memungkinkan SQL Server Management Studio dapat digunakan untuk melakukan monitoring paket SSIS dan mengatur SSIS storage yang digunakan.

6. SQL Server Import dan Export Wizard. Tools ini digunakan untuk melakukan copy data dari sumber ke tujuan data.

#### 2.1.10 Diagram Konteks

Diagram konteks adalah level teratas dari diagram aliran data, yaitu diagram yang tidak detail dari sebuah sistem informasi yang menggunakan aliran-aliran data kedalam dan keluar entitas eksternal. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada di luar sebuah sistem yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara entity luar, masukan dan keluaran sistem. Diagram konteks ini direpresentasikan dengan lingkaran tunggal yang mewakili seluruh sistem.

#### 2.1.11 Data Flow Diagram

Diagram Arus Data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dari mana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut

##### 1. Komponen Terminator

*Terminator* mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya *terminator* dikenal dengan nama entitas luar (*external entity*). Terdapat dua jenis *terminator*:

- a) *Terminator* Sumber (*source*) merupakan terminator yang dijadikan sebagai sumber.
- b) *Terminator* Tujuan (*sink*) merupakan terminator yang menjadi tujuan dari data-data yang dibutuhkan.

*Terminator* dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. *Terminator* dapat juga berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan. Ada tiga hal penting yang harus diingat tentang *terminator* :



- a) *Terminator* merupakan bagian/lingkungan luar sistem. Alur data yang menghubungkan *terminator* dengan berbagai proses sistem, menunjukkan hubungan sistem dengan dunia luar.
- b) Profesional sistem tidak dapat mengubah isi atau cara kerja organisasi, atau prosedur yang berkaitan dengan *terminator*.
- c) Hubungan yang ada antar *terminator* yang satu dengan yang lain tidak digambarkan pada DFD.

## 2. Komponen Proses

Komponen proses menggambarkan bagian dari sistem yang mentransformasikan input menjadi output. Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang/akan dilaksanakan. Pemberian nama proses dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan objek). Ada beberapa hal yang perlu diperhatikan tentang proses:

- A. Proses harus memiliki input dan output.
- B. Proses dapat dihubungkan dengan komponen terminator, data store atau proses melalui alur data.
- C. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

Umumnya kesalahan proses di DFD adalah sebagai berikut:

- A. Proses mempunyai input tetapi tidak menghasilkan output. Kesalahan ini disebut dengan black hole (lubang hitam), karena data masuk ke dalam proses dan lenyap tidak berbekas seperti dimasukkan ke dalam lubang hitam.
- B. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan miracle (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.

### 2.1.12 Kamus Data

Kamus data merupakan kumpulan data-data. Kamus data adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Kamus data merupakan tempat penyimpanan definisi dari aliran-aliran data, file-file dan proses-proses dalam sebuah sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat memberikan informasi mengenai definisi struktur pemakaian masing-masing elemen, dapat mendefinisikan data yang mengalir di sistem dengan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen.

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Kamus data mendefinisikan data elemen dengan cara:

1. Menguraikan arti dari alur data dan data *store* dalam DFD .
2. Menguraikan komposisi paket data pada alur data ke dalam alur yang lebih *elementary* (kecil). Contoh : alamat langganan yang terdiri dari nama jalan, kota dan kode pos.
3. Menguraikan komposisi paket data dalam data *store*.
4. Menspesifikasikan nilai dan unit informasi dalam alur data dan data *store*.
5. Menguraikan hubungan yang terinci antara data *store* dalam suatu *entity relationship diagram* (ERD).