

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Data Warehouse**

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

Menurut W.H.Inmon (2002, p388) data adalah sekumpulan fakta, konsep atau instruksi dalam media penyimpanan untuk komunikasi, pengambilan kembali dan pemrosesan yang bertujuan untuk penyediaan informasi yang dapat dipahami oleh manusia.

Menurut Whitten, Bentley, Dittman (2002, p45) Data adalah fakta murni tentang organisasi dan transaksi bisnisnya.

##### **2.1.2. Pengertian Basisdata**

Menurut Connolly dan Begg (2002, p14) *database* adalah kumpulan data yang saling berhubungan secara logika, dan memiliki penjelasan akan data tersebut, yang dirancang untuk memenuhi kebutuhan organisasi akan informasi.

Menurut Turban, Rainer, Porter (2001, p135) *database* adalah file – file yang saling berhubungan yang dikelompokkan secara logikal.

Menurut W.H.Inmon (2002, p388) *database* adalah kumpulan data yang saling berhubungan (biasanya dikontrol, tidak ada redundansi) yang disimpan dalam sebuah skema.

### **2.1.3. Pengertian *Data Warehouse***

Menurut W.H.Inmon (2002, p31) *data warehouse* adalah kumpulan data yang bersifat *subject-oriented*, terintegrasi, *nonvolatile*, dan *time-variant* yang mendukung keputusan manajemen.

Menurut Vidette Poe, Patricia Klaver, Stephen Brobs, (1998, p389) *data warehouse* adalah kumpulan basis data yang terintegrasi, berorientasi subjek untuk mendukung fungsi DSS (*Decision Support System*), dimana setiap unit data sesuai dengan waktu saat ini.

### **2.1.4. Pengertian *Data Mart***

Menurut W.H.Inmon (2002, p17) *data mart* adalah struktur data yang cakupannya lebih kecil dibandingkan *data warehouse* dimana data dibagi berdasarkan kebutuhan informasi setiap departemen.

## **2.2. Karakteristik *Data Warehouse***

Adapun karakteristik *data warehouse* menurut W.H.Inmon (2002, p31) adalah sebagai berikut :

### **2.2.1. Subject – oriented**

*Data warehouse* dibuat berdasarkan subjek-subjek utama dari suatu organisasi misalnya : Customer, Produk, Penjualan. Setiap area subjek utama secara fisik diimplementasikan sebagai kumpulan tabel yang saling berhubungan dalam *data warehouse*.

### **2.2.2. Integrated**

*Data warehouse* harus dibuat terintegrasi karena berasal dari sumber-sumber data pada aplikasi yang tersebar dalam perusahaan. Sumber data tersebut kemungkinan tidak seragam misalnya, menggunakan format yang berbeda. Karena itulah *data warehouse* harus dibuat konsisten.

### **2.2.3. Nonvolatile**

Data yang terdapat dalam *data warehouse* tidak boleh diupdate.

### **2.2.4. Time-variant**

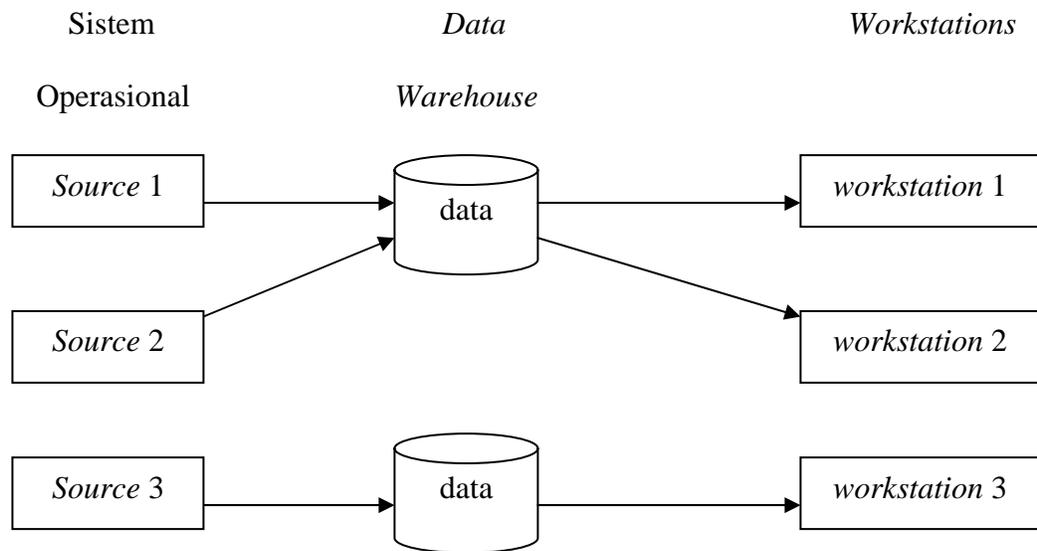
Data yang terdapat pada *data warehouse* hanya valid dalam kurun waktu tertentu saja.

## **2.3. Bentuk Data Warehouse**

Menurut Prabowo (1999, pp8-10), *data warehouse* dapat dibagi menjadi tiga bentuk, yaitu:

### **2.3.1. Data Warehouse Fungsional (*Functional Data Warehouse*)**

Data warehouse yang ada dikelompokkan berdasarkan bagian fungsi bisnis yang ada dalam perusahaan misalnya berdasarkan divisi atau departemen. Pendekatan ini dapat digunakan dengan biaya yang relatif murah, sedangkan risikonya adalah konsistensi data akan hilang jika berada di luar lingkungan fungsi bisnis yang ada. Selain itu, kemampuan dalam pengumpulan data bagi pengguna juga terbatas.

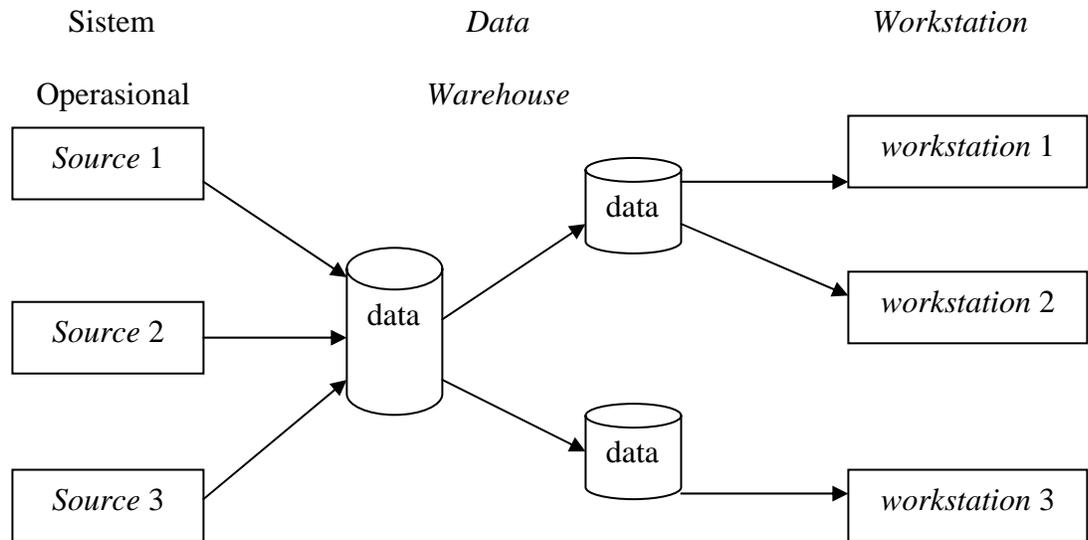


Gambar 2.1. *Data Warehouse* Fungsional

### 2.3.2. **Data Warehouse Terpusat** (*Centralized Data Warehouse*)

Data-data yang ada dikumpulkan terlebih dahulu ke dalam satu tempat terpusat, kemudian data-data yang telah terkumpul dibagi-bagi menurut fungsi masing-masing yang dibutuhkan dalam perusahaan. Bentuk ini yang paling umum digunakan oleh perusahaan.

Kelebihannya adalah data terintegrasi karena setiap data yang masuk akan mengalami sinkronisasi. Namun, karena sistem pengumpulan data yang besar maka biaya pemeliharannya akan lebih besar. Selain itu, waktu untuk membuatnya akan lebih lama.



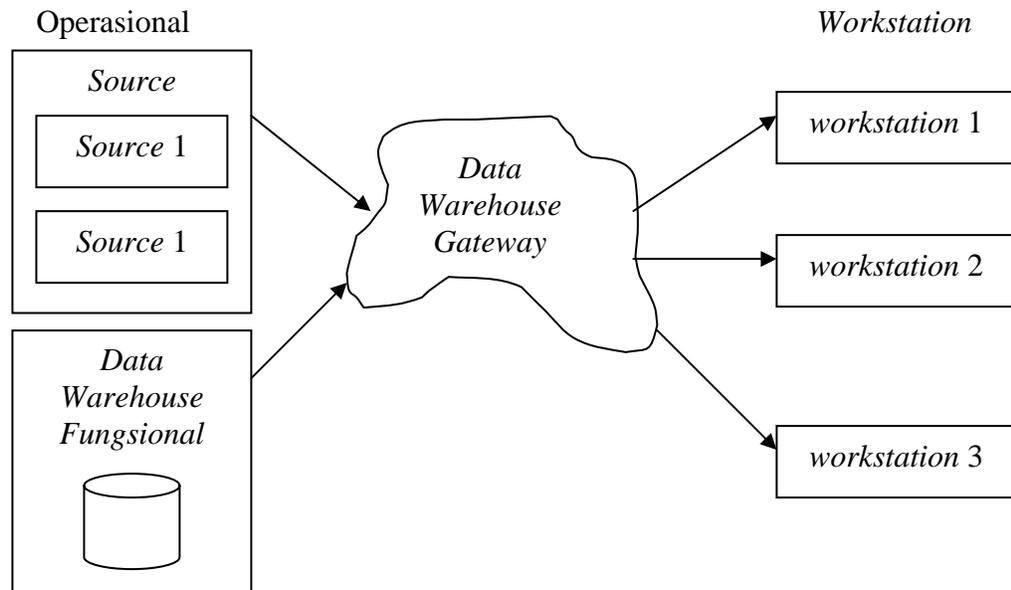
Gambar 2.2. Data Warehouse Terpusat

### 2.3.3. Data Warehouse Terdistribusi (*Distributed Data Warehouse*)

*Data warehouse* terdistribusi dikembangkan berdasarkan konsep *data warehouse* “gateway” yang memungkinkan pengguna untuk langsung berhubungan dengan sumber data maupun dengan pusat pengumpul data lainnya. Gambaran pengguna atas data adalah berupa gambaran logik karena data diambil dari sumber yang berbeda. Pendekatan ini mengandalkan keunggulan teknologi “client – server” untuk mengambil data dari berbagai sumber. Pendekatan ini memungkinkan pengumpulan data fungsional dan sistem operasionalnya menggabungkan bagian – bagian tersebut dengan teknologi “client – server”.

Keuntungan menggunakan bentuk ini adalah data yang ada telah tersinkronisasi. Kelemahan menggunakan bentuk ini adalah biaya yang sangat besar karena setiap sistem pengumpulan data fungsional dan sistem operasi dikelola secara terpisah. Bentuk ini akan sangat efektif bila data telah tersedia

dalam bentuk yang konsisten dan penambahan data tersebut dengan informasi baru yang dilakukan oleh pengguna bila ingin membantu gambaran baru atas informasi tersebut.



Gambar 2.3. Data Warehouse Terdistribusi

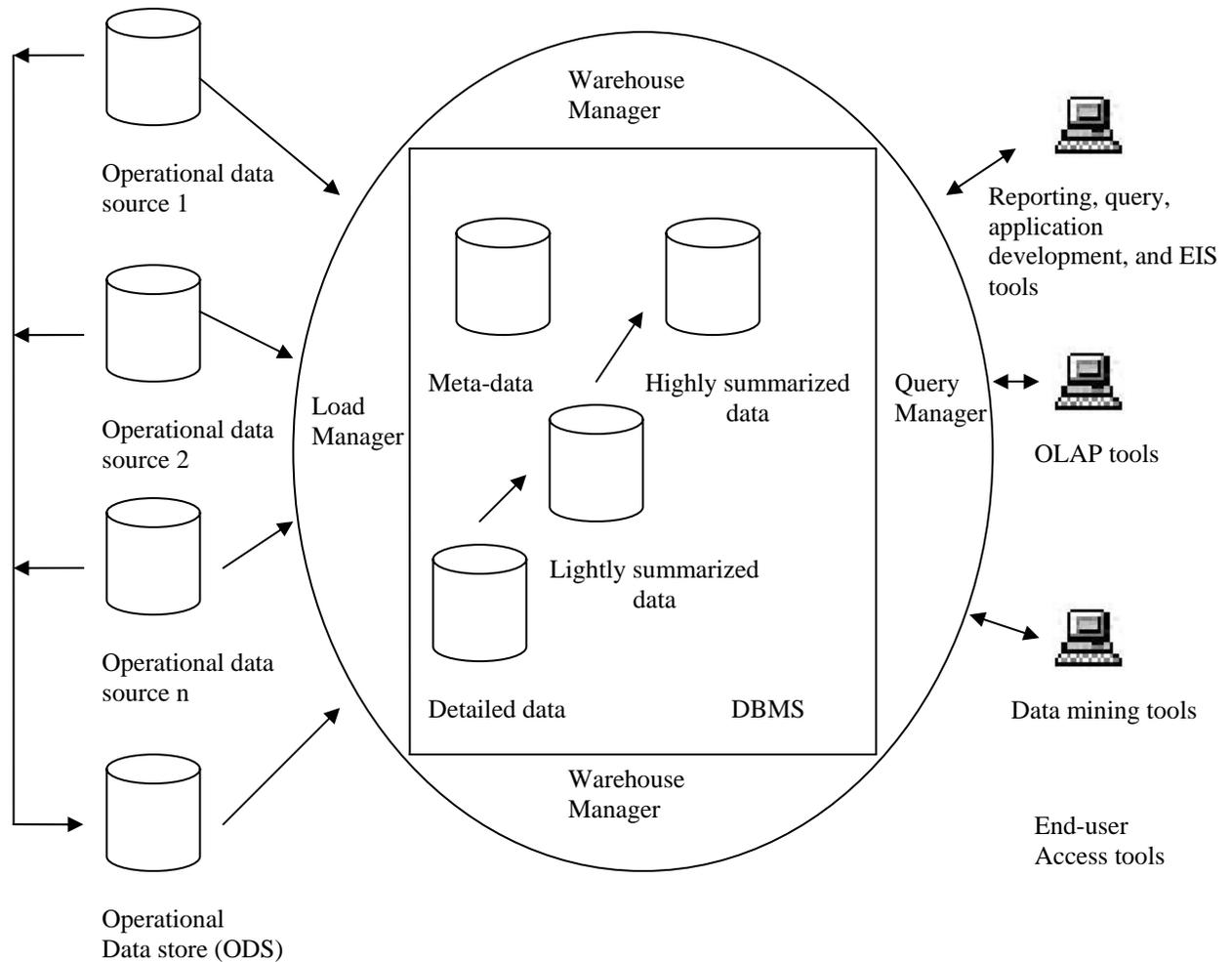
## 2.4. Arsitektur dan Infrastruktur Data Warehouse

### 2.4.1. Arsitektur Data Warehouse

Menurut Connolly dan Begg (2002, p1052) komponen utama dari *data warehouse* yaitu :

1. Data operasional : sumber data dari data warehouse
2. *Operational Datastore* (ODS) : tempat penyimpanan data operasional yang terintegrasi saat ini untuk analisa
3. *Load manager* : mengatur semua operasi yang berhubungan dengan penyaringan dan pemindahan data ke *data warehouse*

4. *Warehouse manager* : mengatur semua operasi yang berhubungan dengan manajemen data dalam *data warehouse*
5. *Query manager* : mengatur semua operasi yang berhubungan dengan manajemen query user
6. *Detailed data* : menyimpan semua data detail dalam skema basis data
7. *Lightly and Highly summarized data* : menyimpan *Lightly and Highly summarized data* yang dihasilkan dari *warehouse manager*
8. *Archive / Backup data*: menyimpan data untuk keperluan backup
9. Metadata : menyimpan metadata (data tentang data) mengenai semua proses pada *data warehouse*
10. *End-user access tools* : menyediakan informasi kepada user untuk pembuatan keputusan



Gambar 2.4. Arsitektur data warehouse

#### 2.4.2. Infrastruktur Data Warehouse

Menurut Poe, Klauer, dan Brobst (1996, p54) infrastruktur *data warehouse* adalah software, hardware, pelatihan dan komponen-komponen lainnya yang memberika dukungan untuk implementasi arsitektur data warehouse. Salah satu instrumen yang mempengaruhi keberhasilan *data warehouse* adalah pengidentifikasian arsitektur mana yang terbaik dan infrastruktur apa yang dibutuhkan.

Menurut O'Neil, Dakin, Whittmer, Matthew, Schader, Hardy (1997, p26) teknikal arsitektur *data warehouse* adalah salah satu komponen yang paling penting. Karena, teknikal arsitektur digunakan sebagai dasar untuk membangun semua komponen *data warehouse* lainnya. Hal ini menyebabkan teknikal arsitektur disebut sebagai infrastruktur. Dasar infrastruktur dimana *data warehouse* dibangun, disebut sebagai *platform*. Komponen – komponen yang membangunnya, yaitu:

- Perangkat keras. masuk sistem operasi. Sebagian besar, *data warehouse* membutuhkan jumlah perangkat keras yang sangat banyak, sekitar 6 sampai 8 atau bahkan 12 CPU dengan memory berukuran gigabytes atau bahkan terabytes.
- Jaringan. Untuk meminimalisasi kompleksitas dan memaksimalkan bandwidth. Jaringan yang ada harus berhubungan langsung dengan semua komponen dan lokasi perusahaan yang membutuhkan akses ke *data warehouse*.
- Perangkat lunak. Komponen perangkat lunak yang paling penting dalam *data warehouse* adalah DBMS ( *Database Management Systems* ). Selain itu, komponen penting perangkat lunak lainnya, antara lain alat untuk pemantauan, administrasi, pengelolaan jaringan untuk memelihara database, perangkat lunak untuk mendukung akses bagi pengguna, *data modeling tools* yang digunakan oleh staf pengembang untuk merancang, implementasi, and memelihara *data warehouse*.

- Sumber daya manusia. Komponen yang paling penting dalam *data warehouse*. Komponen ini terdiri dari kemampuan setiap orang dalam merancang dan mengembangkan arsitektur *data warehouse*.

## **2.5. Skema Bintang**

### **2.5.1. Pengertian Skema Bintang**

Menurut Peterson dan Pinkelman (2000, p70) skema bintang adalah perangkat logika untuk mendesain fakta multidimensi dalam sebuah basis data relational. Skema bintang terdiri dari dua jenis tabel yaitu tabel fakta dan tabel dimensi.

Menurut Poe, Klaver, dan Brobst (1998, p192) tabel fakta disebut juga tabel major, terdiri dari data sebenarnya dari suatu bisnis. Informasi ini biasanya menggunakan pengukuran angka dan bisa terdiri dari banyak baris dan kolom.

Tabel dimensi disebut juga tabel minor, lebih kecil dari tabel fakta dan menyimpan penjelasan data yang mencerminkan dimensi-dimensi dari sebuah bisnis.

### **2.5.2. Manfaat Skema Bintang**

Skema bintang sering digunakan dalam mendesain basis data *Data Warehouse* karena :

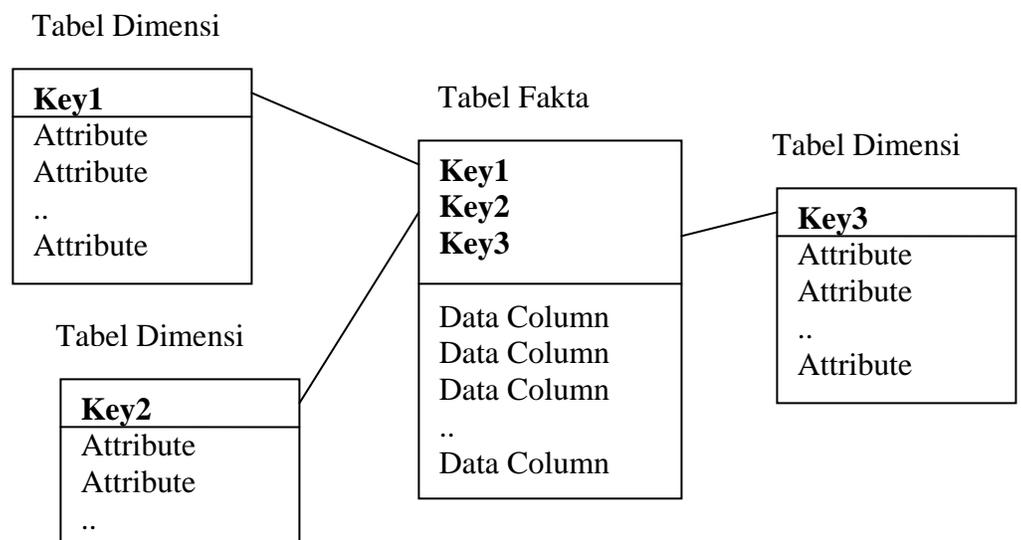
1. Membuat desain basis data yang menyediakan *response time* yang cepat

2. Memungkinkan *database optimizers* untuk bekerja dengan desain basis data yang lebih sederhana untuk menghasilkan rencana eksekusi yang lebih baik
3. Paralel, dalam desain basis data, bagaimana *end user* akan berpikir dan menggunakan data
4. Mempermudah dalam mengerti dan melakukan navigasi *metadata* baik bagi developer ataupun *end user*
5. memperluas pilihan perangkat akses data *front-end*, karena beberapa produk memerlukan desain skema bintang

### 2.5.3. Variasi Skema Bintang

#### 2.5.3.1. Skema Bintang Sederhana

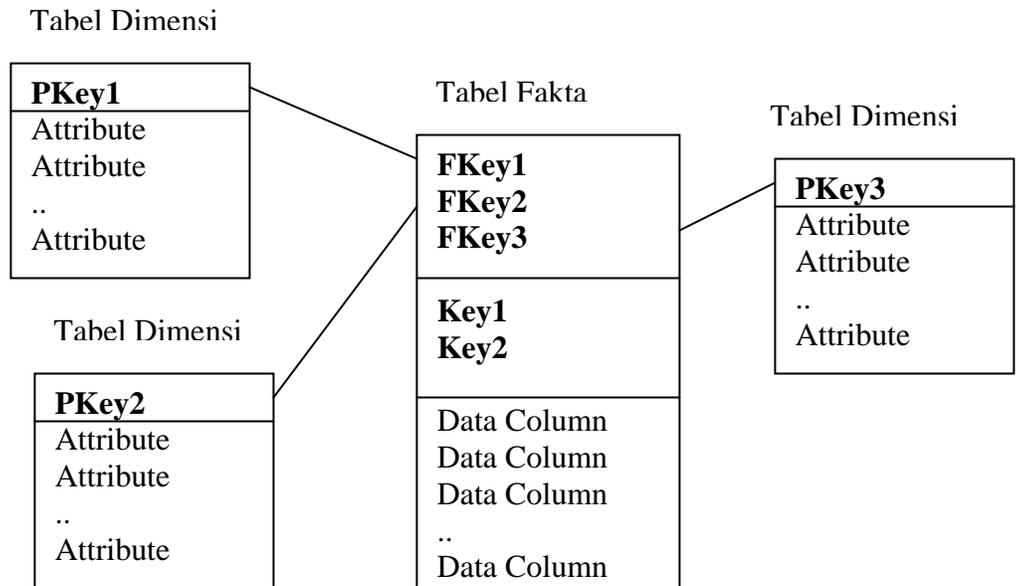
Pada skema bintang sederhana, primary key pada tabel fakta merupakan gabungan dari satu atau beberapa foreign key.



Gambar 2.5. Hubungan antara tabel fakta dan tabel dimensi desain skema bintang sederhana

### 2.5.3.2. Skema Multistar

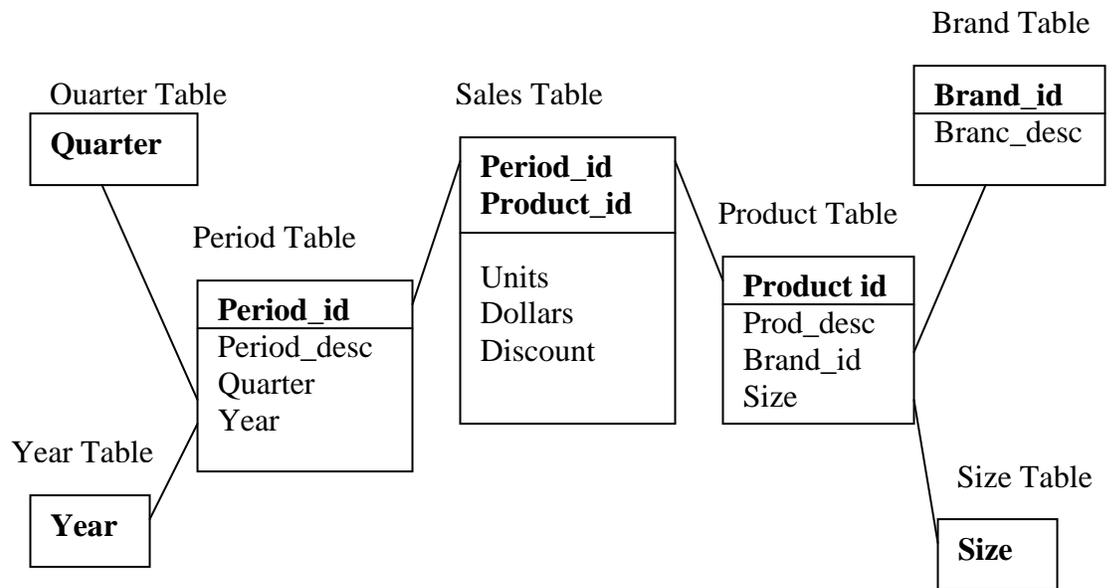
Pada skema multistar, tabel fakta memiliki foreign key dan juga primary key yang terdiri dari satu atau lebih kolom yang menyediakan pengenal unik untuk setiap baris.



Gambar 2.6. Desain skema bintang multistar

### 2.5.3.3. Skema Snowflake

Pada skema snowflake, semua informasi pada tabel dimensi disimpan dengan menggunakan *third normal form*, sementara tabel fakta tetap dalam struktur yang sama seperti skema lainnya.



Gambar 2.7. Skema bintang snowflake

## 2.6. Agregasi

Agregasi adalah proses mengumpulkan data fakta bersamaan dengan attribute yang sudah didefinisikan sebelumnya.

## 2.7. Denormalisasi

Denormalisasi adalah proses menggabungkan tabel dengan hati-hati dan penuh perhitungan untuk meningkatkan daya guna.

Hal ini dilakukan untuk :

1. Mengurangi jumlah join yang harus diproses sehingga dapat meningkatkan dayaguna basis data
2. Memetakan struktur basis data fisik untuk lebih dekat ke dimensi model bisnis user

## 2.8. Granularity

Granularity merupakan tingkat detail atau *summary* data dari unit data dalam *data warehouse*. Semakin rendah granularity yang ada maka semakin detail dan sebaliknya, semakin tinggi granularity maka semakin tidak detail datanya.

## 2.9. Metadata

Menurut Berson, Smith, Thearling (2000, p) Metadata adalah data tentang data yang menggambarkan *data warehouse*. Metadata digunakan untuk membangun, memelihara, mengelola dan menggunakan *data warehouse*.

Metadata dapat diklasifikasikan menjadi:

### 1. *Technical Metadata*

- Berisi informasi tentang *data warehouse* yang digunakan oleh perancang dan administrator *warehouse* saat melakukan tugas pengembangan dan pengelolaan data warehouse.
- Isi dokumentasi *technical metadata*:
  - a. Informasi tentang sumber data
  - b. Gambaran transformasi, yang berisikan metode pemetaan dari basis data operasional ke dalam *warehouse* dan algoritma untuk mentransformasikan data.
  - c. *Warehouse object* dan definisi struktur data untuk target data.
  - d. Aturan yang digunakan untuk melakukan pembersihan data dan perbaikan data.
  - e. Operasi pemetaan data saat pengambilan data dari sumber sistem dan memasukkannya ke dalam target *warehouse* basis data.

- f. Hak akses, *backup history*, *archieive history*, *information delivery history*, *data acquisition history*, pengaksesan data.

## 2. *Business Metadata*

- Berisi informasi yang memberikan kemudahan bagi pengguna untuk memahami informasi yang disimpan dalam *data warehouse*.
- Isi dokumentasi *business metadata*:
  - a. Area subjek dan tipe objek informasi, terdiri dari: pertanyaan, laporan, gambar dan klip video/audio.
  - b. *Internet homepages*.
  - c. Informasi lainnya yang mendukung komponen *data warehouse*. Seperti: informasi yang berhubungan dengan informasi sistem pengiriman, informasi penjadwalannya, detail tentang tujuan pengirimannya, dan objek permintaan bisnis, seperti pertanyaan, laporan dan analisa.

## 3. *Operational Information Data Warehouse*

- Berisi: *data history (snapshots, version)*, kepemilikannya, penggunaan data (*usage data*), *extract audit trail*.

Metadata menyediakan akses yang intraktif bagi pengguna untuk memahami isi dan pencarian data. Pengelolaan metadata disediakan melalui *metadata repository* dan *accompanying software*. Komponen paling utama dalam *metadata repository* adalah *information directory*.

*Information directory* berisi tentang metadata yang membantu *technical dan business user* untuk mengeksploitasi kekuatan *data warehouse*. Direktori ini membantu integrasi, pemeliharaan dan melihat isi sistem *data warehouse*.

## **2.10. Critical Success Factor**

Menurut Poe, Klaver, Smith (1998, p74) ada tiga faktor utama yang menentukan keberhasilan *data warehouse* yaitu :

1. Berfokus pada bisnis, bukan pada teknologi

Dalam membuat *data warehouse* harus ada tujuan bisnis yang jelas, tidak cukup mengatakan alasan bahwa “informasi akan menguntungkan”. Memperkirakan model Return On Investment (ROI) pada *data warehouse* juga penting, karena jika yang menjadi ukuran kesuksesan adalah teknologi, maka belum tentu *data warehouse* tersebut memberikan manfaat pada bisnis.

2. *Rapid Turnaround on Deliverable*

Untuk mengimplementasikan *data warehouse* tidak bisa menggunakan metode pengembangan biasa yaitu System Development Life Cycle (SDLC). Untuk membuat *data warehouse* metode yang digunakan adalah *rapid turnaround on deliverables* menggunakan metode pengembangan iterative. Intinya adalah memungkinkan spesifikasi dan implementasi untuk berkembang sesuai dengan perkembangan permintaan bisnis end user.

### 3. End user pada tim implementasi

Keterlibatan end user pada tim implementasi bisa menjadi faktor yang paling penting dalam implementasi data warehouse. Kesuksesan *data warehouse* diukur oleh user. Jika mereka tidak menerima sistem ataupun tidak menggunakannya untuk membuat keputusan, maka *data warehouse* tersebut dikatakan gagal.