



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## Bab II

### LANDASAN TEORI

Sebelum membahas lebih jauh mengenai metode penelitian dan analisis yang dilakukan di penelitian ini, beberapa teori-teori dasar yang penulis gunakan dalam penelitian ini perlu dipahami untuk mempermudah pemahaman di dalam penelitian ini. Beberapa teori dasar itu seperti data, *metadata*, *database*, dan *data warehouse*.

#### 2.1 Data

Data yang akan digunakan di dalam penelitian ini berupa fakta-fakta yang telah diukur mengenai setiap transaksi bisnis yang ada di dalam perusahaan. Fakta-fakta tersebut merupakan sebuah pengukuran atribut dari suatu entitas yang tersimpan sebagai *file* dan *database*.

Data adalah fakta atau observasi mentah, yang umumnya mengenai fenomena fisik atau transaksi bisnis. Pengertian secara lebih spesifik, bahwa data merupakan objek pengukuran atribut-atribut dari suatu entitas. (O'Brien, 2001) Turban (2010) memberikan pengertian data adalah deskripsi dasar dari suatu benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan dan disimpan tetapi belum terorganisir untuk menyampaikan arti tertentu.

Berdasarkan pengertian dari beberapa ahli dapat disimpulkan bahwa, data adalah suatu kumpulan fakta-fakta dari setiap transaksi bisnis yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar tetapi belum diolah.

## 2.2 *Metadata*

Menurut Inmon (2005), *metadata* adalah data tentang sebuah data yang mendeskripsikan struktur, isi, kunci, *index*, dan informasi lain mengenai data tersebut. Di dalam implementasinya, *metadata* memiliki beberapa kegunaan di dalam *data warehouse*, menurut Connolly dan Begg (2010) kegunaan *metadata* antara lain :

- 1) Didalam proses *extract* dan *load*, *metadata* digunakan untuk melakukan pemetaan dari sumber data ke *common view* dari data yang berbeda di *data warehouse*.
- 2) Metadata dapat digunakan untuk pembuatan tabel *summary* secara otomatis di dalam prosen manajemen *warehouse*.
- 3) Pada proses manajemen *query*, *metadata* dapat digunakan untuk mengarahkan *query* ke sumber data yang paling sesuai.

## 2.3 *Database*

Menurut Conolly dan Begg (2010, hal.65), *database* adalah sekumpulan koleksi dari data yang terhubung secara logikal, dan deksripsi dari data-data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Turban et al (2005) menambahkan untuk pengertian *database*, yaitu sebuah kumpulan file yang saling terhubung yang menyimpan data dan hubungan di dalamnya secara logikal.

Dari pengertian tersebut, penulis menyimpulkan bahwa *database* merupakan kumpulan data yang saling terhubung satu sama lain yang terhubung secara logikal yang berguna untuk memberikan informasi kepada suatu perusahaan atau organisasi.

#### **2.4 *Datamart***

Menurut Connolly dan Begg (2010, hal.1214), *data mart* adalah sebuah *database* yang mengandung bagian data dari suatu divisi di dalam perusahaan yang berguna untuk membantu analisa *requirement* di dalam suatu bagian bisnis unit (contohnya ada bagian penjualan) atau dilakukan untuk membantu *users* dengan *requirement* yang sama untuk menganalisa suatu bisnis proses.

Menurut Ponniah (2010), *data mart* adalah kumpulan data yang berhubungan baik data yang berasal dari *internal* maupun *external* yang bertransformasi, saling terintegrasi, dan disimpan dengan tujuan untuk memberikan strategi yang spesifik di dalam suatu bagian pada perusahaan.

Berdasarkan pengertian diatas, menurut penulis *data mart* merupakan suatu bagian kecil dari sebuah *data warehouse* yang dirancang untuk memenuhi kebutuhan suatu unit bisnis tertentu agar informasi yang diberikan dapat lebih akurat.

#### **2.5 *Data Warehouse***

Vaisman dan Zimanyi (2014, hal.72) memberikan pengertian mengenai *data warehouse* yaitu sebuah gudang data yang saling terintegrasi yang berasal dari berbagai sumber untuk tujuan khusus analisis data multidimensi. Lebih secara

teknikal, *data warehouse* merupakan kumpulan objek yang saling terpadu, *non-volatile*, dan data yang memiliki variasi waktu untuk mendukung manajemen dalam mengambil keputusan.

(Ponniah, 2010) memberikan pengertian berdasarkan studi terbarunya bahwa *data warehouse* merupakan sebuah informasi yang:

- A. Menyediakan data terintegrasi di dalam suatu perusahaan
- B. Membuat informasi data saat ini dan data *history* yang memiliki tujuan untuk strategi pengambilan keputusan.
- C. Membuat sistem pengambilan keputusan tanpa menghalangi sistem operasional data.
- D. Memberikan data di dalam organisasi menjadi konsisten.
- E. Menyajikan sumber informasi yang fleksibel dan interaktif.

Berdasarkan dua pengertian diatas, penulis menarik kesimpulan bahwa *data warehouse* merupakan sebuah konsep pengumpulan data transaksional di dalam perusahaan yang diolah dan disimpan untuk menjadi sebuah informasi yang berkualitas. Sehingga mempermudah pihak *top management* di dalam perusahaan mengambil keputusan guna untuk kepentingan perusahaan.

### **2.5.1 Karakteristik Data Warehouse**

Menurut Vaisman dan Zimanyi (2014, hal.73) karakteristik data dalam *data warehouse* adalah sebagai berikut :

- 1) *Subject Oriented* (Berorientasi Subjek)

*Data warehouse* berfokus pada analisis kebutuhan dari setiap area yang berbeda di dalam suatu organisasi. Area tersebut bervariasi,

tergantung pada jenis kegiatan yang dilakukan di dalam organisasi. Misalnya, dalam kasus perusahaan ritel, analisis dapat fokus pada penjualan produk dan *inventory management*.

2) *Integrated* (terintegrasi)

Pada karakteristik ini berarti data yang diperoleh dari setiap kegiatan operasional dan sistem eksternal yang berbeda harus digabung sehingga data tersebut harus membentuk sebuah integrasi. Sumber data tersebut sering kali memiliki perbedaan format seperti format data, sinonim (bidang dengan data yang berbeda tetapi memiliki arti yang sama), homonim (bidang dengan nama yang sama tetapi memiliki arti yang berbeda), banyaknya kejadian data, dan banyak lainnya. Tujuan dari integrasi data tersebut agar data tidak terbagi-bagi.

3) *Non-Volatile*

Hal ini berarti data dalam *data warehouse* cenderung tidak berubah. Ketidakperubahan tersebut dipastikan dengan pelarangan modifikasi data dan penghapusan data. Hal tersebut memperluas lingkup data ke periode yang lebih lama dari system operasional yang biasa ditawarkan. Jadi data yang lama dan yang baru sama-sama tersimpan di *data warehouse*.

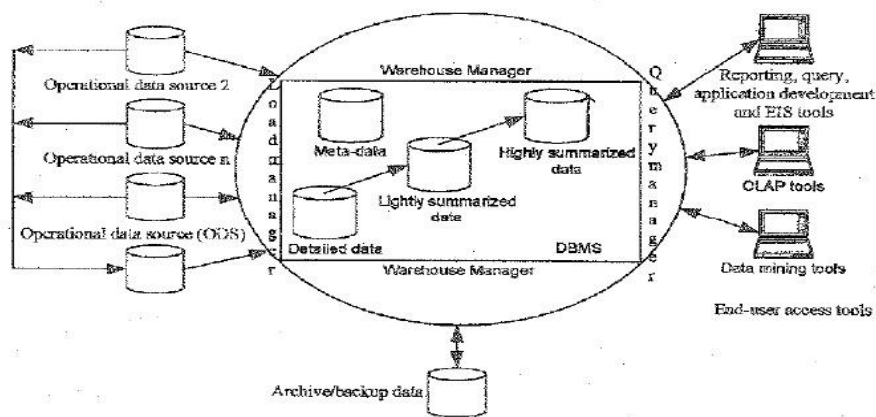
4) *Time Varying*

Data dalam *data warehouse* hanya akurat dan valid pada satu waktu tertentu atau dalam interval waktu tertentu. Contohnya, *data warehouse* di

bank mungkin menyimpan informasi tentang saldo rata-rata bulanan dari rekening nasabah untuk jangka waktu meliputi beberapa tahun.

## 2.5.2 Arsitektur *Data Warehouse*

Menurut Connolly dan Begg (2010) *data warehouse* memiliki arsitektur yang terdiri dari beberapa komponen atau struktur yang saling berhubungan dan saling mendukung satu sama lain. Secara garis besar arsitektur dan komponen-komponen penting dari *data warehouse* terlihat seperti gambar 2.1 di bawah ini



Gambar 2.1 Arsitektur *Data Warehouse* ( Connolly dan Begg, 2005, hal.1157)

Komponen-komponen di dalam *data warehouse* yang seperti terlihat di gambar 2.1 antara lain sebagai berikut :

### 1) *Operational Data*

*Operational data* digunakan sebagai sumber data yang utama di dalam proses perancangan *data warehouse*.

### 2) *Operational Data Store (ODS)*

ODS merupakan tempat penyimpanan untuk data operasional yang aktual dan terintegrasi yang digunakan untuk analisis. ODS sendiri

terstruktur dan terisi dengan data yang sama dengan *data warehouse*, atau hanya berfungsi sebagai media penyimpanan sementara sebelum dipindahkan ke *data warehouse*.

ODS umumnya dibuat ketika sistem operasional tidak mampu menyediakan kebutuhan *reporting*. Pembuatan ODS dapat membantu dalam proses membangun *data warehouse*, karena data yang ada di dalamnya sudah diisi atau di-*extract* dari sumber dan bersih.

### 3) *Load Manager*

*Load Manager* menjalankan semua kegiatan operasi yang berkaitan dengan proses ekstraksi dan *loading* yang berasal dari sumber data ke dalam *data warehouse*.

### 4) *ETL Manager*

*ETL manager* melakukan semua proses yang berhubungan dengan proses ETL dari setiap data yang akan masuk ke *data warehouse*.

### 5) *Warehouse Manager*

*Warehouse manager* menjalankan semua operasi yang berkaitan dengan manajemen data di dalam *data warehouse*. Operasi yang dikerjakan di dalam *warehouse manager*, antara lain :

- a) Analisis data untuk menjamin konsistensi;
- b) Transformasi dan penggabungan sumber data dari penyimpanan sementara ke dalam tabel *data warehouse*;
- c) Membangun *index* dan *view* pada tabel dasar;
- d) Pembuatan denormalisasi (jika diperlukan)



e) Pembuatan agregasi (jika dibutuhkan)

f) *Backup* dan *archiving data*.

6) *Query Manager*

*Query manager* melakukan semua operasi yang berkaitan dengan manajemen *query* yang dilakukan oleh *user*.

7) *Detailed Data*

Pada area ini, dimana menyimpan semua data rinci di dalam *database schema*.

8) *Lightly and Highly Summarized Data*

Pada area ini, *warehouse* menyimpan data yang telah ditetapkan *lightly* dan *highly summarized data* yang dihasilkan oleh *warehouse manager*,

9) *Archieve/ Backup Data*

Komponen ini bertujuan untuk melakukan penyimpanan dan *backup* terhadap data-data yang ada di dalam *data warehouse*.

10) *End-User Access Tools*

*End-User Access Tools* merupakan komponen yang digunakan untuk menyediakan informasi berkaitan dengan proses pengambilan keputusan. Secara umum, tools ini terdiri dari *reporting and query tools*, *application development tools*, *OLAP tools*, serta *data mining tools*.

### 2.5.3 Bentuk Data Warehouse

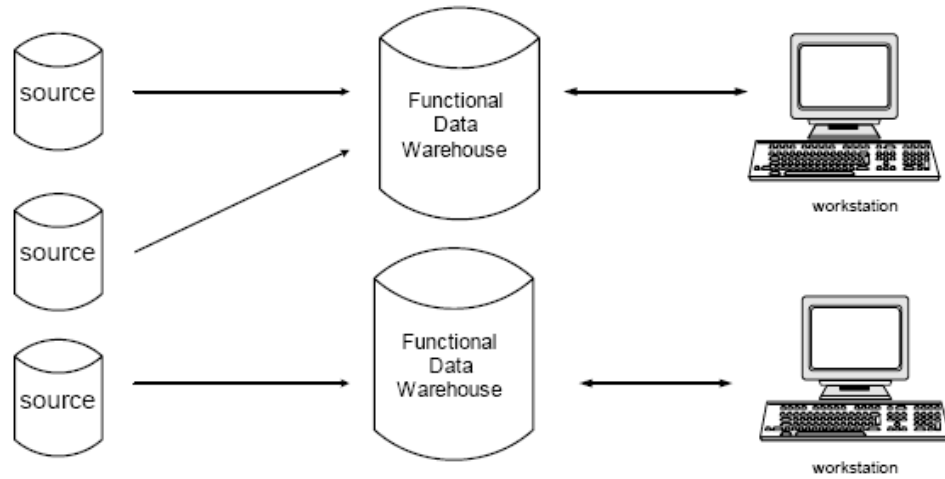
Dalam menentukan bentuk di dalam perancangan *data warehouse*, ada beberapa bentuk umum yang biasa digunakan di dalam merancang aplikasi yang akan dibuat. Beberapa bentuk umum tersebut adalah :

- 1) *Functional Data Warehouse* (Data Warehouse Fungsional)
- 2) *Centralized Data Warehouse* (Data Warehouse Terpusat)
- 3) *Distributed Data Warehouse* (Data Warehouse Terdistribusi)

#### 2.5.3.1 *Functional Data Warehouse* (Data Warehouse Fungsional)

*Data warehouse* dengan bentuk ini relatif mudah dibangun dengan biaya yang relatif lebih murah. Hal tersebut dikarenakan proses perancangan *data warehouse* dibuat lebih dari satu dan dikelompokkan berdasarkan masing-masing fungsi seperti fungsi keuangan (*financial*), fungsi marketing di dalam perusahaan, fungsi personalia, dan sebagainya. Namun, dibalik kelebihanannya, bentuk fungsional memiliki kerugian yaitu ada resiko kehilangan konsistensi data dan terbatasnya kemampuan pengguna dalam hal pengumpulan data.

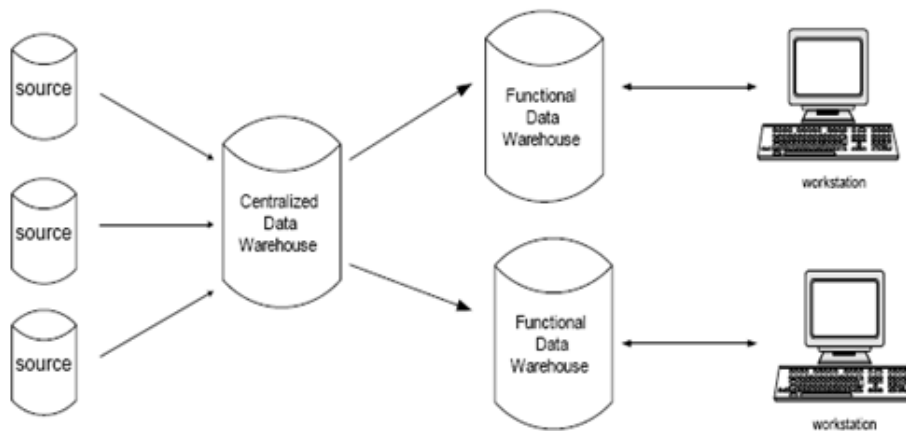
U  
M  
M  
N



**Gambar 2.2 Data Warehouse Fungsional**

### **2.5.3.2 Centralized Data Warehouse (Data Warehouse Terpusat)**

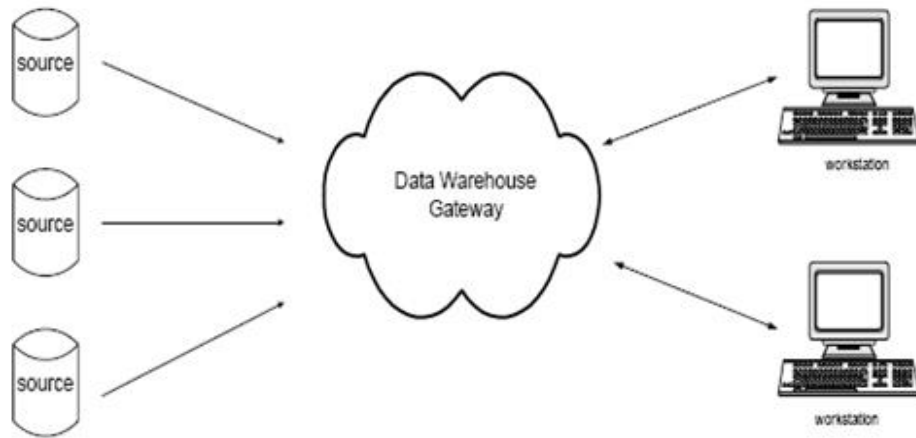
Bentuk *data warehouse* ini mirip dengan bentuk *data warehouse* fungsional, namun terlebih dahulu sumber data dikumpulkan ke dalam satu tempat terpusat yang kemudian data disebar ke dalam fungsinya masing-masing. Keuntungan dari menggunakan model ini adalah datanya terpadu sehingga memberikan konsistensi yang tinggi. Kerugiannya adalah biaya yang mahal, serta proses membangunnya yang memerlukan waktu yang cukup lama.



**Gambar 2.3 Data Warehouse Terpusat**

### **2.5.3.3 Distributed Data Warehouse (Data Warehouse Terdistribusi)**

Pada *data warehouse* terdistribusi ini, digunakan adanya *gateway* yang berfungsi sebagai jembatan penghubung antara *data warehouse* dengan *user* atau *workstation* yang memiliki kebutuhan terhadap data tersebut. Dengan sistem yang terdistribusi seperti ini memungkinkan perusahaan dapat mengakses sumber data yang berada di luar lokasi perusahaan. Keuntungannya adalah data tetap konsisten karena sebelum data digunakan terlebih dahulu disesuaikan atau mengalami proses sinkronisasi. Kerugiannya adalah sistem akan menjadi lebih kompleks, serta biaya yang tinggi dalam proses *implementasinya*.



Gambar 2.4 *Data Warehouse* Terdistribusi

#### 2.5.4 Keuntungan *Data Warehouse*

Menurut Connolly dan Begg (2010), implementasi dari *data warehouse* akan memberikan keuntungan sebagai berikut:

- 1) Pengembalian investasi yang tinggi

Sebuah perusahaan perlu mengeluarkan sumber daya dalam jumlah yang besar untuk memastikan kesuksesan implementasi dari *data warehouse*. Biaya yang perlu dikeluarkan juga bervariasi sesuai dengan solusi yang ingin didapati dari setiap bisnis proses. Sebuah penelitian dilakukan *International Data Corporation* (IDC) pada tahun 1996, memperlihatkan *data warehouse project* memberikan 3 tahun rata-rata *return of investment* (ROI) sebesar 401% (IDC,1996). Lebih lanjut, penelitian terbaru dari IDC dalam *business analytics*, memperlihatkan *analytical tools* yang mengakses *data warehouse* dapat memberikan 431 % rata-rata dalam waktu satu tahun ROI (IDC,2002).

## 2) Keunggulan Kompetitif

ROI yang tinggi juga membuat sebuah perusahaan mendapatkan keunggulan kompetitif yang besar dari dampak kesuksesan implementasi *data warehouse*. Keunggulan kompetitif didapatkan dengan memungkinkan pihak manajemen yang berwenang mengambil keputusan dapat mengakses data yang sebelumnya tidak tersedia, tidak diketahui, atau bahkan tidak terjangkau. Contohnya adalah, pelanggan, *trend*, dan permintaan.

## 3) Peningkatan produktifitas dari pengambil keputusan

*Data Warehousing* meningkatkan produktifitas para pengambil keputusan dengan membuat sebuah basis data yang saling terintegrasi, konsisten, berorientasi subjek, dan *historikal*. *Data warehouse* mengintegrasikan berbagai sistem yang digunakan menjadi sebuah *view* yang konsisten untuk perusahaan. Data tersebut dibentuk menjadi informasi yang bernilai sehingga pengambil keputusan dapat melakukan analisis yang lebih substansif, akurat dan konsisten.

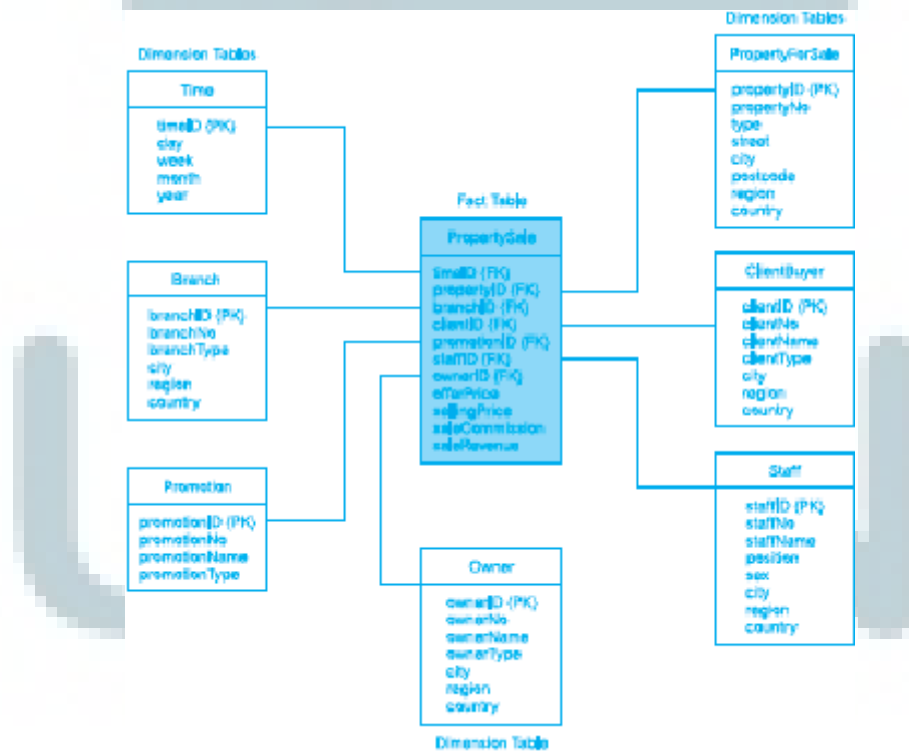
### 2.5.5 Permodelan *Data Warehouse*

Conolly dan Begg memberikan penjelasan terhadap komponen dari *data warehouse*. Komponen tersebut dapat dideskripsikan menggunakan teknik yang disebut *dimensionality modelling* yang berarti sebuah teknik logika desain yang memiliki tujuan untuk memberikan data standard (Connolly & Begg, 2010, hal. 1227). *Dimensionality modelling* menggunakan konsep *entity-relationship modelling* dengan beberapa batasan yang penting, yang terdiri dari satu tabel

bernama tabel fakta dan sekumpulan tabel-tabel lain yang lebih kecil yang bernama tabel dimensi. Setiap tabel dimensi memiliki satu *primary key* sehingga *primary key* dari tabel fakta terdiri dari dua atau lebih *foreign key* dari tabel-tabel dimensi.

### 2.5.5.1 Star Schema

Setiap tabel dimensi memiliki sebuah *primary key* yang berhubungan dengan tepat satu komponen dari *primary key* yang ada di tabel fakta. Dengan kata lain, tabel fakta akan memiliki *primary key* lebih dari satu atau disebut dengan *foreign key*. Karakteristik model data warehouse dimensional *star schema* yaitu dengan tabel fakta yang berada di tengah, dikelilingi oleh tabel-tabel dimensi yang masih bisa disederhanakan lagi ( belum dinormalisasi).



Gambar 2.5 Contoh Star Schema (Connolly,2005)

Berikut terdapat beberapa kelebihan dari *star schema* menurut Ponniah (2014, hal.241), yaitu :

a) Mudah Dimengerti

Skema bintang mendefinisikan hubungan antara tabel sama seperti bagaimana pengguna memvisualisasikan hubungannya pada umumnya sehingga mudah dimengerti oleh pengguna.

b) Mengoptimalkan Navigasi

Hubungan antar tabel yang sederhana pada *star schema* akan memungkinkan *user* untuk mengoptimalkan navigasi. Walaupun hasil *query* terlihat kompleks, tapi *user* masih dapat melakukan navigasi dengan mudah.

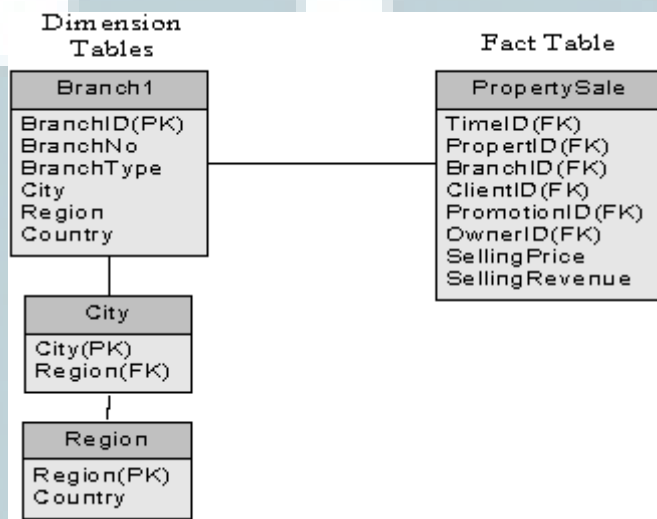
c) Paling cocok dengan pemrosesan query

*Star schema* merupakan bentuk yang paling cocok dalam pemrosesan *query* dikarenakan dari bentuknya yang terpusat pada tabel fakta. Tanpa bergantung pada banyak dimensi dan kerumitan *query*, yang pertama dilakukan adalah memilih baris dan tabel dimensi dan kemudian menemukan baris yang sama di tabel fakta.



### 2.5.5.2 Snowflake Schema

Connolly dan Begg (2010) memberikan penjelasan mengenai *snowflake schema*, dimana skema ini tetap memiliki tabel fakta di tengah dan dikelilingi dengan tabel dimensi yang sudah di normalisasi.



Gambar 2.6 Contoh Snowflake Schema (Connolly,2010)

UMMN

Ada beberapa kelebihan dan kekurangan pada *snowflake schema* menurut Ponniah (2014), adalah :

A. Keuntungan

- Penyimpanan yang relatif kecil
- Struktur yang sudah ternormalisasi sehingga mudah untuk di-*update* dan di-*maintain*

B. Kekurangan

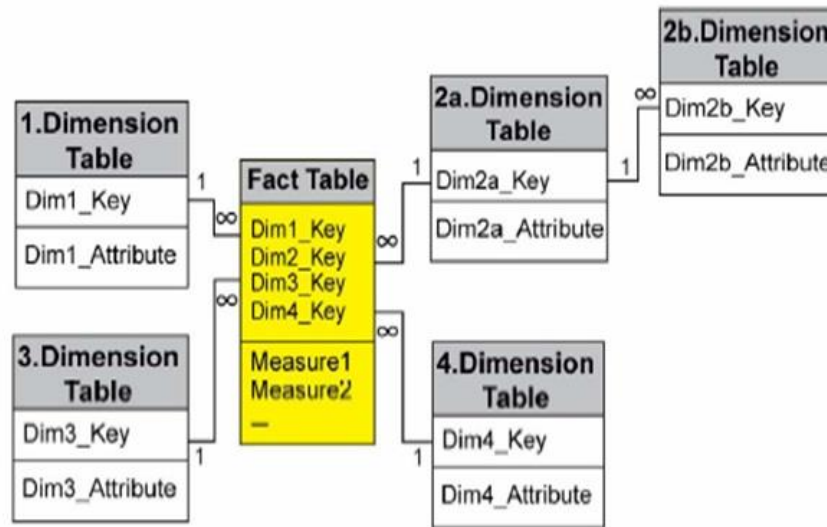
- Skema kurang intuitif dan terlalu kompleks untuk *end user*
- Sulit untuk melihat isi skema secara menyeluruh
- Performa *query* menurun karena adanya hubungan tambahan

*Snowflaking schema* tidak dianjurkan di dalam lingkungan *data warehouse*.

Performa *query* mengambil bagian tertinggi di dalam *data warehouse* dan skema *snowflaking* akan menghambat kinerja tersebut.

### 2.5.5.3 Starflake Schema

Variasi lain dari database dimensional adalah *starflake schema*. Dimana dalam skema ini menggabungkan karakteristik dari kedua skema sebelumnya. *Starflake schema* adalah dimensional data model yang memiliki tabel fakta di tengah dengan dikelilingi tabel dimensi yang sudah dinormalisasi maupun belum dinormalisasi (*denormalized*).



Gambar 2.7 Contoh Starflake Schema

### 2.5.6 Metodologi desain Data Warehouse

Dalam proses merancang *data warehouse*, terdapat dua pendekatan umum yang biasa digunakan yaitu menurut Kimball yang juga dikutip oleh Conolly dan Begg menggunakan pendekatan *bottom-up*. Sedangkan salah satu pencetus pendekatan *top-down* adalah William H. Inmon.

Menurut Inmon (2001), proses perancangan *data warehouse* dibuat sebagai pusat data yang terpusat yang kemudian dari *data warehouse* yang terpusat tersebut akan disusul dengan pembuatan *data marts*. Sedangkan, untuk pendekatan yang Kimball (2008) kemukakan bahwa *data marts* dibentuk dari setiap proses bisnis yang berorientasi pada suatu departemen. Oleh karena itu, pendekatan ini diutamakan untuk merancang *data marts* yang ada pada masing-masing departemen

atau subjek, yang kemudian data tersebut diintegrasikan menjadi *data warehouse* organisasi secara keseluruhan.

Metodologi	Keuntungan	Kekurangan
Inmon ( <i>Top Down</i> )	Memberikan potensi untuk memberikan data secara konsisten dan komprehensif untuk perusahaan	Waktu implementasi yang lebih lama
	Kontrol informasi dapat dilakukan secara sentralisasi	Resiko kegagalan yang tinggi karena rancangan yang kompleks, serta akan mengambil banyak waktu dan biaya
	Kerangka <i>data warehouse</i> lebih terstruktur	
Kimball ( <i>Bottom up</i> )	Dapat memberikan data yang lebih penting di dalam setiap departemen di dalam perusahaan	Data bisa menjadi tidak konsisten dan sulit untuk direkonsiliasi
	Resiko kegagalan yang kecil, sehingga alokasi waktu dan biaya juga relatif kecil	Memungkinkan terjadinya data yang duplikat dalam proses perancangan.

**Tabel 2.1 Kelebihan dan Kekurangan Metodologi Rancangan *Data Warehouse* Menurut Kimball dan Inmon**

## 2.6 Online Analytical Processing (OLAP)

Connolly dan Begg (2010) menyebutkan bahwa OLAP merupakan perpaduan dinamis, analisis dan konsolidasi dari data multimedimensi yang berukuran besar. Ponniah (2014) menabahkan bahwa OLAP mencakup spektrum multidimensional data yang kompleks yang melibatkan perhitungan yang rumit dan memerlukan waktu respon yang cepat.

Berdasarkan 2 pengertian diatas, penulis menyimpulkan bahwa OLAP memungkinkan *user* untuk mendapatkan pengertian dan pengetahuan yang mendalam untuk mengambil keputusan dari suatu data yang dapat diakses dengan cepat, konsisten, serta interaktif melalui variasi tampilan data tersebut.

Functional Area	Contoh dari aplikasi OLAP dalam kegiatan fungsional
Keuangan	<i>Budgeting, financial performance analysis</i> , dan model keuangan
Sales	Analisa penjualan dan <i>Sales forecasting</i>
Marketing	Analisa pasar, analisa promosi, analisa pelanggan, dan <i>market / customer segmentation</i>
Manufacturing	<i>Perencanaan produksi dan analisis kecacatan (defect)</i>

**Tabel 2.2 Contoh aplikasi OLAP pada setiap kegiatan fungsional**

Menurut Connolly dan Begg (2010) ada beberapa tipe OLAP, yaitu :

1) Multidimensional OLAP (MOLAP)

MOLAP menggunakan struktur data tertentu dan *database* multidimensi untuk melakukan organisir dalam analisa data.

2) *Relational* OLAP (ROLAP)

Tipe OLAP yang memiliki ketergantungan kepada *database* relasional atau RDBMS (*Relational Database Management System*) sebagai media penyimpanan data yang akan diolah. Keuntungan dari ROLAP adalah tidak memerlukan *storage* tambahan, sedangkan kekurangannya adalah jika data untuk suatu *cube* sangat besar maka performa pengambilan data akan sangat buruk.

3) *Hybrid* OLAP (HOLAP)

HOLAP merupakan kombinasi antar MOLAP dan ROLAP dimana HOLAP akan menyimpan data *precomputed aggregate* pada media penyimpanan (*storage*) HOLAP sendiri. Data tersebut berupa data yang berada di level atas atau *high level view*, sedangkan untuk data yang lebih rendah disimpan di *database* relasional.

#### 4) *Desktop* OLAP (DOLAP)

DOLAP *tools* merupakan tipe OLAP yang melakukan penyimpanan dalam *client-based file* dan mendukung proses multidimensi menggunakan mesin multidimensi *client*.

## 2.7 Proses ETL ( *Extract, Transform, Load* )

Kimball (2013) menjelaskan bahwa proses ETL merupakan segalanya diantara sumber sistem operasional dengan *data warehouse area*. Proses ETL merupakan kumpulan proses sumber data operasional sebelum masuk ke dalam *data warehouse*. Proses awal dari ETL ada ekstraksi (*Extraction*) data dimana data dari sumber dibaca dan dimengerti yang kemudian akan disimpan di sistem ETL untuk keperluan perubahan selanjutnya.

Setelah data diekstraksi ke dalam sistem ETL, terdapat banyak kemungkinan perubahan langkah ini disebut dengan *Transformation*. Contoh perubahan tersebut seperti membersihkan data (melakukan koreksi data, melakukan koreksi data yang konflik, atau merubah ke format yang standar), melakukan kombinasi data dari berbagai sumber, dan melakukan *de-duplicating data*.

Langkah terakhir adalah mengirimkan data yang sudah ditransformasikan tersebut ke tempat penyimpanan akhir sebelum akhirnya ditampilkan ataupun diolah kembali (*loading*).

## **2.8 Dashboard**

Menurut Vaisman dan Zimanyi (2014) *dashboard* adalah alat visualisasi data untuk *business intelligence* yang paling populer, dimana *dashboard* dapat melihat koleksi dari berbagai komponen visual (*charts* atau KPIs) di dalam *view* tunggal. *Dashboard* dapat memberikan kemudahan bagi organisasi untuk dapat melakukan pendekatan secara efektif, memonitor dan mengatur performa bisnis mereka. *Dashboard* digunakan untuk memvisualisasikan data organisasi dan memanfaatkan model pengukuran kinerja yang berbeda untuk mengidentifikasi dan menerapkan langkah-langkah untuk semua tingkatan dalam organisasi.

*Data warehouse* dan OLAP merupakan dua teknologi yang mendasari kesuksesan jangka panjang dari *dashboard*. Hal itu dikarenakan kedua teknologi ini dapat membantu *dashboard* dalam hal :

- 1) Memberikan tampilan data dari berbagai sumber yang sudah terintegrasi;
- 2) Melakukan pengukuran dan menampilkan hasil pengukuran tersebut;
- 3) Menyediakan informasi yang akurat dalam kurun waktu tertentu;

## **2.9 Business Intelligence (BI)**

Ponniah (2014) menyatakan, BI umumnya digunakan secara sinonim dengan informasi yang tersedia di dalam suatu perusahaan untuk membuat

keputusan yang strategis. Secara umum, BI merupakan sebuah informasi yang diolah dari setiap analisa terhadap berbagai bisnis proses di dalam perusahaan yang digunakan untuk menentukan pengambilan keputusan yang bertujuan meningkatkan performa bisnis. Secara khusus, BI dapat meningkatkan aset informasi dari proses-proses bisnis utama sehingga performa bisnis dapat meningkat.

### **2.10 Pentaho Data Integration (PDI) Kettle / Spoon**

*Pentaho data integration kettle* merupakan *software Open Source* yang dikembangkan oleh Pentaho dengan tujuan untuk proses integrasi dan pengelolaan data. *PDI Kettle* atau bisa disebut dengan *Spoon* menyediakan fasilitas ETL (*Extraction, Transformation, Loading*). *Spoon* sendiri juga dapat digunakan untuk membersihkan data, *loading* dari file ke *database* atau sebaliknya dapat digunakan untuk proses migrasi dari satu aplikasi dengan aplikasi lainnya.

Kelebihan dari *Pentaho Data Integration / Kettle* :

- a) Utilitas ETL *open source* yang paling populer.
- b) Designer GUI yang intuitif dan sangat mudah digunakan.
- c) *Multi Platform*.
- d) *Script* ETL dapat disimpan dalam bentuk *filesystem* maupun *repository*.
- e) Mendukung *multi pipelining* sehingga *load balance* maupun optimasi pekerjaan *data warehouse* dapat dilakukan dengan mudah.
- f) Mendukung *clustering engine* ETL.
- g) Terdiri atas lebih dari 200 *steps* yang mencakup *job* dan *transformation*



- h) Mendukung *Apache Virtual Filesystem* sehingga *filesystem* seperti HTTP Webdav, FTP, SFTP, dan lain sebagainya dapat dengan mudah diakses dengan konfigurasi yang minimal.

### **2.10.1 Pentaho Business Analytics Platform**

Komponen ini bertindak sebagai *connection point* untuk semua komponen. Ini memungkinkan untuk solusi unifikasi (kompak) dan integrasi end-to-end data menjadi data visualisasi. Ini juga menyediakan satu set alat untuk pengembangan, *deployment*, manajemen dari aplikasi-aplikasi yang ada.

### **2.10.2 Pentaho Analysis Services**

*Pentaho analysis services* dikenal juga sebagai *Mondrian*, yaitu sebuah *relational OLAP server*. *Mondrian* mendukung MDX (*multidimensional expression*) query dan XML, untuk analisis dan *olap4j* spesifikasi antarmuka. *Pentaho Analysis Services* membaca data dari setiap sumber data dan melakukan agregasi data di dalam sebuah *memory cache*.

### **2.10.3 Pentaho Data Mining**

Menggunakan *Waikato environment* untuk pengetahuan analisis (*weka*) untuk mencari pola data. *Weka* terdiri dari mesin algoritma untuk tugas *data mining*. Algoritma tersebut dapat menjalankan fungsi seperti *data processing*, *regression analysis*, *classification methods*, *cluster analysis*, sampai dengan visualisasi.

#### 2.10.4 *Pentaho Data Integration*

Dikenal juga dengan nama *Kettle*. Berisi sebuah *engine* integrasi data (ETL) dan *GUI* (*graphical user interface*) yang memungkinkan pengguna untuk melakukan transformasi data sumber, integrasi data sumber, sampai dengan penyimpanan di dalam *data warehouse*. Hal tersebut juga mendukung *deployment* dalam *node* tunggal dan *cloud* atau *cluster*.

The logo of Universitas Muhammadiyah Negeri (UMMN) is displayed in a light blue, semi-transparent style. It consists of a circular emblem containing a stylized building or tower structure, with the letters 'UMMN' written in a bold, sans-serif font directly below the emblem.