

MULTIDIMENSIONAL DATA WAREHOUSE DENGAN MENGUNAKAN MySQL

Kusnawi

Dosen STMIK AMIKOM Yogyakarta

Abstraksi

Dalam sebuah organisasi data adalah sebuah asset yang berguna untuk memutuskan suatu kebijakan, melakukan suatu kebijakan strategis, atau mengambil keputusan bisnis. Data diperoleh dari sistem atau aplikasi operasional yang terdapat pada organisasi. Perkembangan data membutuhkan pengelolaan yang pada akhirnya dapat dimanfaatkan untuk keperluan organisasi atau perusahaan. Data warehouse adalah suatu konsep dan kombinasi teknologi yang memfasilitasi organisasi atau perusahaan untuk mengelola dan memelihara data historis yang diperoleh dari sistem atau aplikasi operasional. Data warehouse mengumpulkan data historis yang kemudian dapat dijadikan sebagai bahan komprehensif bagi manajemen untuk dapat mengambil keputusan, analisis kebutuhan organisasi, hingga peramalan kondisi organisasi berdasar data. Untuk membangun Data warehouse banyak software yang dapat digunakan dengan memanfaatkan DBMS antara lain Mysql, SQL server, Oracle dan lainnya. Untuk merancang Data warehouse dimulai dengan kebutuhan berdasarkan proses bisnis suatu organisasi atau perusahaan yang kemudian menentukan model Data warehouse beserta arsitektur yang akan digunakan dan beberapa contoh pengaksesan data multlidimensi pemanfaatan Data warehouse.

Kata kunci : *Data warehouse, DBMS, Model data*

Pendahuluan

Data warehouse merupakan tempat penyimpanan untuk ringkasan data historis yang diambil dari basis data-basis data yang tersebar di suatu organisasi. *Data warehouse* mengumpulkan semua data perusahaan dalam satu tempat agar dapat diperoleh pandangan yang lebih baik dari suatu proses bisnis/kerja dan meningkatkan kinerja organisasi. *Data warehouse* mendukung proses pembuatan keputusan manajemen.

Tujuan utama dari pembuatan *data warehouse* adalah untuk menyatukan data yang beragam ke dalam sebuah tempat penyimpanan dimana pengguna dapat dengan mudah menjalankan *query* (pencarian data), menghasilkan laporan, dan melakukan analisis. Salah satu keuntungan yang diperoleh dari keberadaan *data warehouse* adalah dapat meningkatkan efektifitas pembuatan keputusan.

Menurut Rainardi(2008), dijelaskan tentang gambaran sistem *data warehouse* dimulai dari sistem OLTP yang dilanjutkan dengan ETL(*extract, transform, dan load*) untuk proses mengambil dan memilih data dari sumber yang dipilih sesuai dengan kebutuhan bisnis dan mengintegrasikan, menstransformasikan serta memuat data kedalam *dimensional data store* (DDS) dan *multidimensional databases* (MDBs). Secara singkat gambaran sistem dari *data warehouse* adalah proses yang meliputi *source system*, ETL dan DDS.

Menurut Inmon(2002), karakteristik *Data warehouse* adalah *Subject Oriented* atau berorientasi pada subyek. *Time-variant*, artinya bahwa perubahan data ditelusuri dan dicatat sehingga laporan dapat dibuat dengan menunjukkan waktu perubahannya. *Non Volatile* berarti bahwa data yang telah disimpan tidak dapat berubah. Dan bersifat *integrated*, artinya *data warehouse* akan mencakup semua data operasional organisasi yang disimpan secara konsisten.

Dalam pemilihan sebuah arsitektur *data warehouse* terlebih dahulu harus ditentukan dimana *data warehouse* ditempatkan dan juga dimana kendali kontrol data. Sebagai contoh data dapat ditempatkan di lokasi terpusat yang diatur secara terpusat (*centralized global warehouse*) atau data ditempatkan terdistribusi (*distributed global warehouse*). Secara fisik *centralized global warehouse* digunakan oleh semua organisasi. Sedangkan sebuah *distributed global warehouse* juga digunakan untuk semua organisasi tetapi distribusinya melalui beberapa lokasi fisik dalam organisasi.

Menurut Hoffer, dkk(2002) membedakan arsitektur *data warehouse* kedalam tiga komponen:

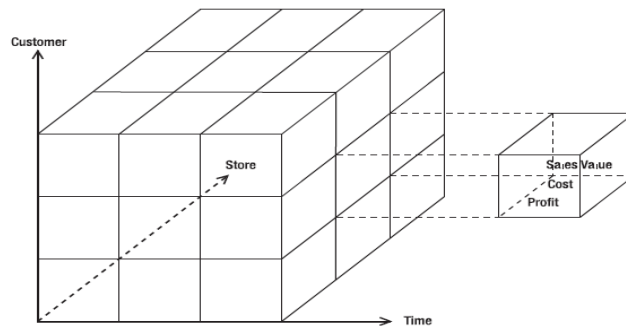
- *Data warehouse* yang berisi data dan perangkat lunak terkait.

- Perangkat lunak akuisisi data (*back end*) , yang mengekstraksi data dari system legacy dan sumber eksternal, memperkuat dan meringkasnya, dan memuatnya ke dalam data warehouse.
- Perangkat lunak klien (*front end*), yang mengizinkan para pengguna untuk mengakses dan meneliti data di dalam data warehouse

Pembahasan

Multidimensional Database (Cube)

Cube adalah sebuah bentuk database dimana data disimpan dalam bentuk Cell, dan posisi dari sel-sel tersebut ditentukan oleh beberapa variabel yang disebut dengan Dimension. Jumlah Dimension ini secara teori bisa tidak terbatas, tidak perlu terkuantifikasi untuk membentuk sebuah bangun 3 dimensi berupa cube. Istilah Cube dan penggambaran dalam bentuk cube (3 dimensi) ini dimaksud untuk mempermudah visualisasi kita tentang sifat multi dimensionalnya.



Gambar 1. Contoh cube tiga dimensi

Merancang dan membuat Data warehouse

Ada sembilan tahap metodologi dalam perancangan database untuk *data warehouse*, yaitu(kimball) :

1) Pemilihan proses

Untuk tahapan merancang data warehouse adalah sesuai dengan kebutuhan bisnis dari pihak perusahaan, sebagai contoh

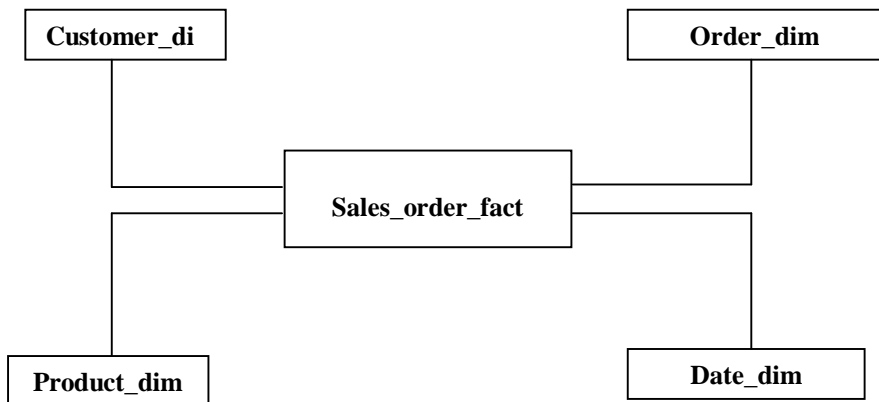
perusahaan ingin menganalisa penjualan yang berdasarkan dari berbagai sumber yang berbeda atau dimensi.

- 2) Pemilihan sumber
Perancangan data model data warehouse berdasarkan dari tabel fakta dan dimensi, untuk pemilihan sumber harus mengacu pada kemampuan untuk merepresentasikan fakta-fakta yang di ambil dari berbagai sumber dimensi pada OLTP. Sebagai contoh sales_order_fact, adalah menggambarkan rincian customer secara detail.
- 3) Mengidentifikasi dimensi
Dimensi ini untuk menggambarkan fakta-fakta pada tabel fakta, sebagai contoh setiap data pada tabel customer_dim dilengkapi dengan customer_sk, customer_number, customer_name, customer_street_address, customer_zip_code dan lainnya.
- 4) Pemilihan fakta
Pemilihan sumber dari sebuah tabel fakta akan menentukan fakta mana yang akan digunakan. Sebagai contoh pada sales_order_fact terdiri dari order_sk, customer_sk, product_sk, order_date_sk order_amount yang menggambarkan fakta dari berbagai tabel dimensi yaitu customer, order, product dan time yaitu order_date.
- 5) Menyimpan pre-kalkulasi di tabel fakta
Hal ini akan terjadi jika tabel fakta kehilangan statement sesuai dengan fakta yang dibutuhkan.
- 6) Melengkapi tabel dimensi
Pada tahap ini kita menambahkan keterangan selengkap-lengkapnyanya pada tabel dimensi dan harus bersifat intuitif dan mudah dipahami oleh pengguna.
- 7) Pemilihan durasi database
Berkaitan dengan penyimpanan data warehouse, misalnya untuk data penjualan yang diambil dari 5 tahun yang lalu dan maksimal 10 tahun yang akan datang.
- 8) Menelusuri perubahan dimensi yang perlahan

Ada tiga tipe perubahan dimensi yaitu atribut dimensi yang telah berubah tertulis ulang, menimbulkan sebuah dimensi baru dan dimensi yang telah berubah menimbulkan alternatif, sehingga nilai atribut lama dan yang baru dapat diakses secara bersama pada dimensi yang sama.

- 9) Menentukan prioritas dan mode *query*, berkaitan dengan perancangan fisik.

Untuk membuat, model *data warehouse* yang digunakan adalah model *dimensional* dalam bentuk *star schema*, yaitu terdiri dari fact table dan dimension table. Berikut ini adalah rancangan bentuk star schema dan struktur tabel yang digunakan.



Gambar 2. Star schema

Struktur query yang digunakan adalah:

```
TABLE customer_dim  
( customer_sk INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
, customer_number INT , customer_name CHAR (50) ,  
customer_street_address CHAR (50) , customer_zip_code INT (5)  
, customer_city CHAR (30) , customer_state CHAR (2) , effective_date DATE ,  
expiry_date DATE );
```

```
CREATE TABLE product_dim  
( product_sk INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
, product_code INT , product_name CHAR (30) , product_category CHAR (30)  
, effective_date DATE , expiry_date DATE );
```

```
CREATE TABLE order_dim  
( order_sk INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
, order_number INT , effective_date DATE , expiry_date DATE );
```

```
CREATE TABLE date_dim  
( date_sk INT NOT NULL AUTO_INCREMENT PRIMARY KEY  
, date DATE , month_name CHAR (9) , month INT (1) , quarter INT (1) , year  
INT (4) , effective_date DATE , expiry_date DATE );
```

```
CREATE TABLE sales_order_fact  
( order_sk INT , customer_sk INT , product_sk INT , order_date_sk INT ,  
order_amount DECIMAL (10, 2) );
```

Langkah – langkah membuat datawarehouse:

1. Login ke MySQL server

```
C:\>mysql -uroot -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 2 to server version: 5.0.15-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

2. Membuat database , misalkan DW, kemudian buat tabel yaitu *fact table* dan *dimension table*, yaitu sales_order_fact sebagai tabel faktanya, kemudian diikuti dengan tabel dimensi product_dim, date_dim, customer_dim dan order_dim.

```
mysql> show tables;
```

```
+-----+  
| Tables_in_dw |
```

```
+-----+
|customer_dim |
|date_dim     |
|order_dim    |
|product_dim  |
|sales_order_fact|
+-----+
```

5 rows in set (0.00 sec)

3. Inputkan semua isi data pada tabel dimensi dan fakta.

Slowly Changing Dimension(SCD)

Suatu teknik yang digunakan untuk menambah atau mengupdate data pada *dimension table* dari sumber yang berbeda, ini dilakukan biasanya jika kebutuhan data warehouse yang bersifat dinamis. Berikut contoh data sumber customer yang akan diupdate kedalam *customer_dim*.

CUSTOMER NO	CUSTOMER NAME	STREET ADDRESS	ZIP CODE	CITY	STATE
1	Really Large Customers	7500 Louise Dr.	17050	Mechanicsburg	PA
2	Small Stores	2500 Woodland St.	17055	Pittsburgh	PA
3	Medium Retailers	1111 Ritter Rd.	17055	Pittsburgh	PA
4	Good Companies	9500 Scott St.	17050	Mechanicsburg	PA
5	Wonderful Shops	3333 Rossmoyne Rd.	17050	Mechanicsburg	PA
6	Loyal Clients	7070 Ritter Rd.	17055	Pittsburgh	PA
7	Distinguished Partners	9999 Scott St.	17050	Mechanicsburg	PA

Gambar 3. Data sumber customer.csv

- 1) Untuk melakukan SCD harus membuat dulu tabel staging yang mempunyai fungsi untuk menghubungkan dengan data sumber. Berikut ini adalah contoh script untuk *customer_stg*, yang sumbernya adalah dari file *customer.csv*.

```
CREATE TABLE customer_stg
(customer_number INT
```

```
customer_name CHAR (30),customer_street_address CHAR
(30),customer_zip_code INT (5),
customer_city CHAR (30), customer_state CHAR (2) );
TRUNCATE customer_stg;
LOAD DATA INFILE 'customer.csv'
INTO TABLE customer_stg
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(customer_number,
customer_name,customer_street_address,customer_zip_code,
customer_city, customer_state);
```

2) Jalankan proses SCD dibawah ini:

```
UPDATE customer_dim a, customer_stg b
SET a.customer_name = b.customer_name
WHERE a.customer_number = b.customer_number
AND a.expiry_date = '9999-12-31'
AND a.customer_name <> b.customer_name ;
INSERT INTO customer_dim
SELECT NULL
, customer_number, customer_name, customer_street_address ,
customer_zip_code , customer_city , customer_state ,
CURRENT_DATE , '9999-12-31'
FROM customer_stg
WHERE customer_number NOT IN ( SELECT b.customer_number
FROM customer_dim a,
customer_stg b
WHERE
a.customer_number = b.customer_number );
```


3) Tampilkan output dari SCD

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
mysql> select * from customer_dim \G
***** 1. row *****
      customer_sk: 1
      customer_number: 1
      customer_name: Really Large Customers
      customer_street_address: 7500 Louise Dr.
      customer_zip_code: 17050
      customer_city: Mechanicsburg
      customer_state: PA
      effective_date: 2009-02-14
      expiry_date: 9999-12-31
***** 2. row *****
      customer_sk: 2
      customer_number: 2
      customer_name: Small Stores
      customer_street_address: 2500 Woodland St.
      customer_zip_code: 17055
      customer_city: Pittsburgh
      customer_state: PA
      effective_date: 2009-02-14
      expiry_date: 9999-12-31
***** 3. row *****
    
```

Gambar 4. Hasil SCD pada customer_dim

Berikut ini adalah beberapa contoh pengaksesan data multidimensi pada data warehouse.

Aggregate Queries

➔Daily Sales Aggregation

```

SELECT date, SUM (order_amount) , COUNT(*)
FROM sales_order_fact a , date_dim b
WHERE a.order_date_sk = b.date_sk
GROUP BY date ORDER BY date ;
    
```

Output :

date	SUM (order_amount)	COUNT(*)
2007-02-05	58000.00	10
2007-02-06	195000.00	6

2 rows in set (0.03 sec)

➔ Annual Aggregation

```
SELECT year, product_name, customer_city, SUM (order_amount),  
COUNT(*)  
FROM sales_order_fact a, date_dim b, product_dim c, customer_dim d  
WHERE a.order_date_sk = b.date_sk  
AND a.product_sk = c.product_sk  
AND a.customer_sk = d.customer_sk  
GROUP BY year, product_name, customer_city  
ORDER BY year, product_name, customer_city;
```

Output :

```
+-----+-----+-----+-----+  
-----+  
| year | product_name      | customer_city | SUM  
|COUNT(*) |  
| (order_amount) |  
+-----+-----+-----+-----+  
-----+  
| 2007 | Floppy Drive      | Mechanicsburg |  
70000.00 | 5 |  
| 2007 | Floppy Drive      | Pittsburgh    |  
8000.00 | 1 |  
| 2007 | Hard Disk Drive   | Mechanicsburg |  
46000.00 | 2 |  
| 2007 | Hard Disk Drive   | Pittsburgh    |  
34000.00 | 3 |  
| 2007 | LCD Panel         | Mechanicsburg |  
61000.00 | 3 |  
| 2007 | LCD Panel         | Pittsburgh    |  
34000.00 | 2 |  
+-----+-----+-----+-----+  
-----+  
6 rows in set (0.03 sec)
```

Specific Queries

➔ Monthly Storage Product Sales

```
SELECT product_name , month_name , year , SUM (order_amount) ,  
COUNT(*)  
FROM sales_order_fact a , product_dim b , date_dim c  
WHERE a.product_sk = b.product_sk AND a.order_date_sk = c.date_sk  
GROUP BY product_name , product_category , month_name , year  
HAVING product_category = 'Storage'  
ORDER BY year , month name ;
```

Output:

```
+-----+-----+-----+-----+  
-----+-----+  
| product_name | month_name | year |  
SUM(order_amount) | COUNT(*) |  
+-----+-----+-----+-----+  
-----+-----+  
| Hard Disk Drive | February | 2007 |  
65000.00 | 2 |  
| Floppy Drive | February | 2007 |  
55000.00 | 2 |  
| Hard Disk Drive | February | 2007 |  
15000.00 | 3 |  
| Floppy Drive | February | 2007 |  
23000.00 | 4 |  
+-----+-----+-----+-----+  
-----+-----+  
4 rows in set (0.00 sec)
```

➔ Quarterly Sales in Mechanisburg

```
SELECT customer_city , quarter , year , SUM (order_amount) , COUNT  
(order_sk)  
FROM sales_order_fact a , customer_dim b , date_dim c  
WHERE a.customer_sk = b.customer_sk AND a.order_date_sk = c.date_sk
```

```
GROUP BY customer_city , quarter, year  
HAVING customer_city = 'Mechanicsburg'  
ORDER BY year, quarter;
```

Output:

```
+-----+-----+-----+-----+  
+-----+  
| customer_city | quarter | year |  
|SUM(order_amount) | COUNT(order_sk) |  
+-----+-----+-----+-----+  
+-----+  
| Mechanicsburg |         4 | 2007 |         177000.00  
|                10 |  
+-----+-----+-----+-----+  
+-----+  
1 row in set (0.00 sec)
```

Inside-Out Queries

➔ **Product Performer**

```
SELECT month_name , year , product_name , SUM (order_amount) ,  
COUNT(*)  
FROM sales_order_fact a , product_dim b , date_dim c  
WHERE a.product_sk = b.product_sk AND a.order_date_sk = c.date_sk  
GROUP BY month_name , year , product_name  
HAVING SUM (order_amount) >= 75000  
ORDER BY month , year , product_name ;
```

Output:

```
+-----+-----+-----+-----+  
--+-----+  
| month_name | year | product_name | SUM  
(order_amount) | COUNT(*) |  
+-----+-----+-----+-----+  
--+-----+  
| November | 2007 | LCD Panel |  
75000.00 | 2 |
```

```
+-----+-----+-----+-----+
--+-----+
1 row in set (0.00 sec)
```

→Loyal Customer

```
SELECT customer_number , year , COUNT(*)
FROM sales_order_fact a , customer_dim b , date_dim c
WHERE a.customer_sk = b.customer_sk AND a.order_date_sk = c.date_sk
GROUP BY customer_number , year
HAVING COUNT(*) > 3 AND (12 - MONTH (MAX (date))) < 7 ;
```

Output :

```
+-----+-----+-----+
| customer_number | year | COUNT(*) |
+-----+-----+-----+
|                1 | 2007 |          4 |
+-----+-----+-----+
1 row in set (0.02 sec)
```

Penutup

Pembuatan data warehouse harus memperhatikan beberapa aspek diantaranya adalah kebutuhan bisnis dan proses bisnis yang ada pada perusahaan yang kemudian model dimensional serta arsitektur yang akan digunakan. Data warehouse kemampuan analisa data yang dapat dilihat dari berbagai dimensi seperti dimensi waktu, product, customer dan order untuk mendapatkan informasi fakta penjualan.

Daftar Pustaka

Darmawikarta.2007. Dimensional Data Warehousing with MySQL: A Tutorial Brainy Software Corp
Kimball,R.; & Merz, R.1998. *The Data Warehouse Lifecycle Toolkit. Expert Methods for Designing, Developing and Deploying Data Warehouses.* Wiley Computer Publishing, Canada.

Vincent Rainardi.2008. Building a Data Warehouse: With Examples
in SQL Server. New York: après
Inmon, W.H.2002.*Building the data warehouse*.Third Edition.
John Wiley & Sons, Inc.