

# RANCANG BANGUN *DATA WAREHOUSE* UNTUK ANALISIS STRATEGI PRODUKSI PENJUALAN USULAN : PT.XYZ

Rahmad Syah\*

## Abstract

*Some companies convection that is often experienced by the owner of XYZ lies in the lack of information provided by each - each branch office ( lack of synchronization between the data center and branch office ) . This is because the human resources available in each - each branch office is not sufficient to be able to make a more detailed report to the center . Inadequate reports of each - each branch office , also led to the management does not know for sure about their finances and inventory - each branch . System to be built is a container used to store data from various branch offices . Of these containers will be carried out such an analysis process OLAP ( OnLine Analytical Processing ) and can also be used as reporting tools .Operational data of each branch will be entered into a central database . After that , we will perform an Extraction , Transformation , and Loading to filter the data that will be inserted into the data warehouse . Modeling the model used is the Star . Architecture that will be used is Two - Layer Architecture . The information generated from the processing of the data warehouse design waking with OLAP analysis in the form of a query . Information presented in the form of queries : goods , office and date*

*Keyword : Data Warehouse, OLAP, Two – Layer Architecture, Sales analysis, PT.XYZ*

## PENDAHULUAN

Di era global ini, berbagai informasi terus berkembang dan kebutuhan semakin meningkat. Sehingga menimbulkan banyaknya permintaan dalam penyajian data yang sering sekali tidak sesuai dan tidak akurat khususnya pada kebutuhan perusahaan dan perkantoran. Beberapa perusahaan yang bergerak dibidang produksi dan distribusi yang memiliki beberapa cabang yang tersebar diseluruh Indonesia sering sekali muncul

---

\* Sekolah Tinggi Teknik Harapan (STTH)

sulitnya dalam menganalisis proses transaksi penjualan. Dalam beberapa kasus juga, masih terdapat proses penjualan masih dilakukan secara manual oleh pihak perusahaan, sehingga lambatnya laporan kinerja yang dilakukan beberapa kantor cabang dapat merugikan perusahaan. Proses pengambilan data penjualan dari masing - masing cabang ke kantor pusat, sering dilakukan manual lewat *email* atau sms. Data penjualan yang diberikan oleh masing - masing cabang berisi jumlah baju yang terjual berdasarkan ukurannya, dan pemasukan cabang tersebut.

Sekarang ini, banyak perusahaan menghadapi beberapa masalah. Masalah yang pertama adalah masalah dalam produksi dan pendistribusian barang. Data penjualan yang diperoleh oleh kantor pusat, tidak memberikan informasi yang cukup tentang desain dan model mana yang banyak diminati oleh masyarakat dari masing - masing kota, sehingga pihak manajemen tidak mengetahui kebutuhan yang diminati oleh masyarakat di masing - masing kota. Analisis dapat dilakukan dengan mengumpulkan *data* penjualan yang bersifat historis atau lampau dan kemudian melihat kinerja penjualan perusahaan (Randy O.P, 2012) . Hal ini menyebabkan pihak manajemen kesulitan dalam menentukan jumlah produksi dan target distribusi sebuah produk.

Masalah lain yang dialami pihak manajemen perusahaan terletak pada kurangnya informasi yang diberikan oleh masing - masing kantor cabang (tidak adanya sinkronisasi data antara pusat dan kantor cabang). Hal disebabkan karena sumber daya manusia yang ada di masing - masing kantor cabang kurang memadai untuk dapat membuat laporan yang lebih terperinci ke pusat . Laporan yang kurang memadai dari masing - masing kantor cabang, juga menyebabkan pihak manajemen tidak mengetahui secara pasti tentang keuangan dan stok barang masing - masing kantor.

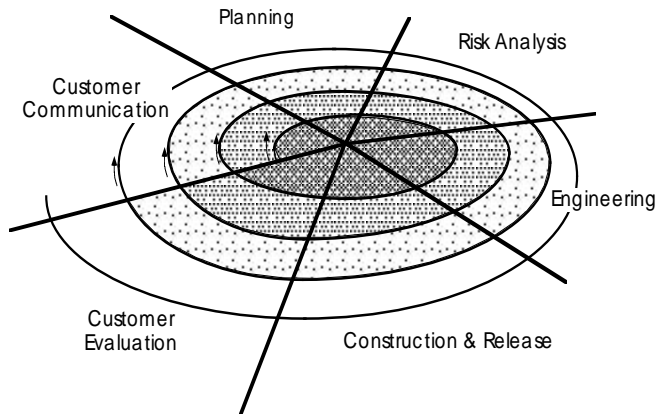
*Data warehouse* adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi, *time-variant*, dan bersifat tetap dari koleksi data dalam mendukung proses manajemen pengambilan keputusan (W.H. Inmon dan Richard D.H, 2005).

*Data warehouse* merupakan basis data yang bersifat analisis dan *read only* yang digunakan sebagai fondasi dari sistem penunjang keputusan (McLeod, et.all, 2007). *Data warehouse* merupakan basis data relasional yang didesain lebih kepada *query* dan analisa dari pada proses transaksi, biasanya

mengandung data historis dari proses transaksi dan bisa juga data dari sumber lainnya. *Data warehouse* memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan organisasi menggabung/konsolidasi data dari berbagai macam sumber (Connolly, et.all, 2005).

## METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam pembangunan sistem ini adalah model *spiral*. Metode ini dipilih karena dalam pembangunan sebuah *data warehouse* seringkali *user* tidak benar - benar tahu apa yang sebenarnya dia butuhkan sebelum dia melihat apa yang saat ini dia butuhkan (William H. Inmon, 2005).



Gambar 1. Model Spiral

1. *Planning*  
Proses perencanaan adalah proses penentuan tujuan, batasan, dan pendekatan yang akan digunakan dalam pembangunan sistem.
2. *Risk Analysis*  
Analisis resiko adalah proses paling penting dalam model *spiral*. Pada tahap ini, semua kemungkinan yang dapat membantu dalam pembangunan sistem dianalisis.
3. *Engineering*  
Pada proses ini, pembangunan sistem mulai dilakukan.

4. *Construction and Release*

Pada proses ini, sistem mulai dijalankan dan diuji coba.

5. *Customer Evaluation*

Proses ini digunakan untuk memperoleh *feedback* dari *user* berdasarkan evaluasi dari uji coba sistem.

6. *Customer Communication*

Proses ini digunakan untuk mencari kebutuhan baru dari *user* berdasarkan hasil evaluasi.

## **SNOWFLAKE**

*Snowflake* adalah skema yang memiliki dimensi yang digunakan secara bersama - sama oleh banyak tabel fakta dan masih dipecah kembali. Skema ini dipilih karena pada dimensi barang dan dimensi toko, masih perlu didetailkan untuk mempermudah dalam proses *dicing* pada *OLAP*(R.P. Aloysius, 2011).

## **OLAP (ONLINE ANALYTICAL PROCESSING)**

*OLAP* adalah salah satu cara untuk mengolah data yang ada pada sebuah *data warehouse*. *OLAP* memberikan jawaban terhadap *query analytic* untuk data yang bersifat multidimensional. Teknik dalam *OLAP* ini ada berbagai macam. Dalam pengembangan kali ini, teknik yang akan digunakan adalah *slicing* dan *dicing*(Thomas C. Hammergren, 2009).

*Slicing* adalah suatu operasi yang mengambil potongan kubus berdasarkan nilai tertentu pada suatu dimensi. Misalnya, mengambil data penjualan pada bulan Desember 2013.

*Dicing* adalah sebuah operasi untuk mengurangi kumpulan data yang akan dianalisis dengan memberikan sebuah kriteria seleksi. Misalnya, mengambil data penjualan yang terjadi pada masing - masing kota.

## **PERSPEKTIF PRODUK**

Sistem yang akan dibangun merupakan sebuah wadah yang digunakan untuk menampung data dari berbagai kantor cabang. Dari wadah inilah

nantinya bisa dilakukan proses analisis seperti OLAP dan juga bisa digunakan sebagai *reporting tools*.

Data operasional dari masing kantor cabang nantinya akan dimasukkan ke *database* pusat. Setelah itu, akan dilakukan proses *Extraction, Transformation, dan Loading* untuk menyaring data yang akan dimasukkan ke *data warehouse*.

## **FUNGSI PRODUK**

Secara umum, *data warehouse* yang akan dibangun ini akan difungsikan sebagai *OLAP tools* yang kemudian akan dibuat menjadi sebuah laporan kepada pihak manajemen PT.XYZ. *Data Warehouse* yang akan dibangun juga harus dapat menyajikan data yang diinginkan oleh pihak manajemen.

Data yang ingin disajikan nantinya adalah data penjualan, data penerimaan barang masing - masing kantor cabang, dan data pengembalian barang dari masing - masing kantor cabang. Proses *ETL* harus dapat berjalan secara otomatis ataupun dengan permintaan administrator. Sedangkan proses *reporting* harus dapat menyesuaikan dengan permintaan pengguna.

## **BATASAN - BATASAN**

*Data Warehouse* berisi data penjualan, penerimaan, dan pengembalian barang masing kantor cabang. Proses *ETL* dilakukan oleh administrator atau secara otomatis setiap waktu yang sudah ditetapkan.

## **ASUMSI DAN KEBERGANTUNGAN**

Diasumsikan data yang hendak di ekstrak ke *data warehouse* telah ada di *database* pusat. *Data Warehouse* ini juga hanya akan berisi data penjualan, penerimaan, dan pengembalian barang masing - masing kantor. Sistem ini sangat tergantung dengan data yang ada di pusat dan proses *ETL*.

## **Deskripsi Rinci Kebutuhan**

Bagian ini berisi semua kebutuhan perangkat lunak yang diuraikan secara rinci untuk keperluan perancangan perangkat lunak. Kebutuhan perangkat lunak tersebut meliputi kebutuhan antarmuka eksternal, kebutuhan fungsional dan non fungsional, kebutuhan data, atribut perancangan.

### **ANTARMUKA PEMAKAI**

Pengguna sistem ini adalah orang - orang yang tingkat kemampuan mengoperasikan komputernya cukup baik. Antarmuka sistem ini harus memberikan banyak pilihan kepada pengguna, agar nantinya data yang muncul sesuai dengan permintaan dari pengguna.

### **ANTARMUKA PERANGKAT KERAS**

Sistem ini membutuhkan komputer yang digunakan sebagai *database server* sebagai tempat penyimpanan *database*. Selain itu, sistem ini juga membutuhkan printer untuk mencetak laporan. Untuk dapat mengakses data dari komputer server melalui komputer lain, diperlukan jaringan komputer yang terhubung ke *database server*.

### **ANTARMUKA PERANGKAT LUNAK**

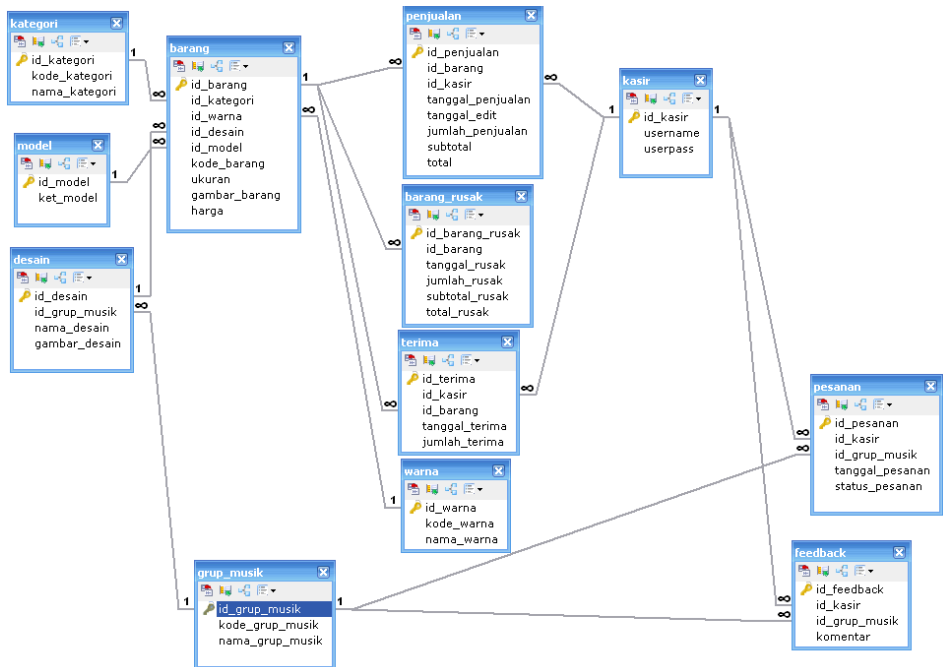
Sistem yang akan dibangun ini tidak akan langsung mengakses data operasional dari pusat. Proses ekstraksi dilakukan menggunakan *ETL tools*. *ETL tools*, *reporting tools*, dan *OLAP tools* akan dipisahkan dari aplikasi yang menangani data operasional.

### **ANTARMUKA KOMUNIKASI**

Sistem ini akan melalui banyak komputer *client*. Oleh karena dibutuhkan jaringan komputer yang menghubungkan komputer *server* dan komputer *client*. Protokol jaringan yang akan digunakan adalah TCP/IP.

## SKEMA RELASI

Berikut adalah skema relasi dari data yang ada di PT.XYZ.



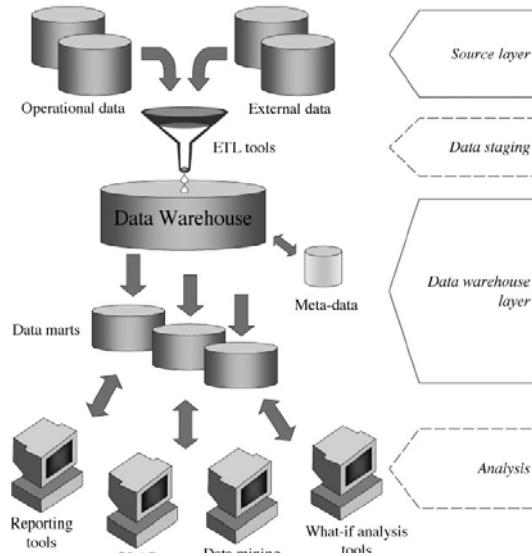
Gambar 2 Skema Relasi Data Warehouse

Ada 3 jenis laporan yang dibutuhkan oleh pihak manajemen. Laporan pertama adalah data penjualan. Data penjualan harus dapat menginformasikan jenis barang, tanggal penjualan, toko yang menjual, dan jumlah penjualan tiap barang. Laporan kedua adalah data penerimaan barang. Dalam data penerimaan barang, harus dapat menginformasikan tentang jenis barang, tanggal penerimaan, toko, dan jumlah yang diterima. Laporan yang ketiga adalah laporan pengembalian barang. Dalam laporan ini informasi yang harus ada antara lain jenis barang, tanggal pengembalian, toko, dan jumlah pengembalian. Dari gambar 2, maka nantinya akan dibutuhkan 3 tabel fakta. Tabel penjualan, penerimaan, dan pengembalian.

### *Arsitektur Pembangunan Data Warehouse*

Arsitektur yang akan digunakan adalah *Two – Layer Architecture*. Arsitektur ini terdiri dari 4 lapisan aliran data.

## Two – Layer Architecture



Gambar 3. Two – Layer Architecture

Lapisan pertama adalah *source layer*. Pada lapisan ini, data masih berupa operasional data. Data operasional yang akan digunakan pada pembangunan *data warehouse* kali ini sudah berupa data *logic* yang ada di *database server*. Lapisan kedua adalah *data staging*. Pada lapisan ini, data operasional akan diekstrak (lebih dikenal dengan proses *ETL*) ke dalam *data warehouse*.

Lapisan ketiga adalah *data warehouse layer*. Informasi akan disimpan pada sebuah penyimpanan *logic* yang tersentralisasi, yaitu *data warehouse*. *Data warehouse* dapat diakses secara langsung, dan juga bisa digunakan sebagai sumber untuk membuat *data marts* yang merupakan sebagian dari duplikasi *data warehouse* dan dirancang khusus bagian khusus.

Lapisan keempat adalah *analysis*. Analisis disini nantinya akan menggunakan OLAP sebelum dijadikan sebuah laporan khusus.

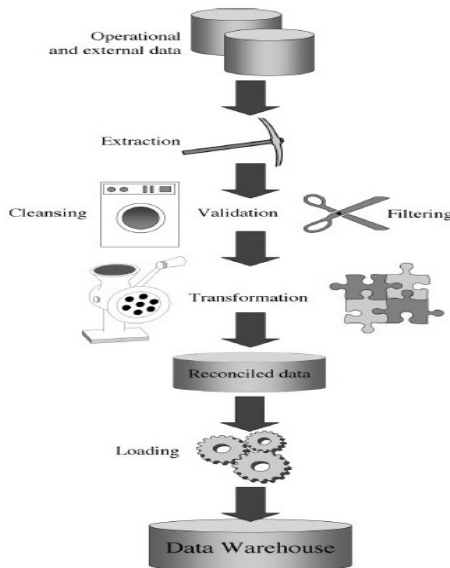
### **Source Layer**

Pada lapisan ini, data masih berupa operasional data. Data operasional yang akan digunakan pada pembangunan *data warehouse* kali ini sudah berupa data *logic* yang ada di *database server*. Skema relasinya ada pada gambar 2.



## DATA STAGING

Pada lapisan ini, data operasional akan diekstrak ( lebih dikenal dengan proses *ETL* ) ke dalam *data warehouse*. ETL terdiri dari 4 langkah.



Gambar 4. Tahapan-tahapan ETL (Ekstraksi,Transformation,Loading)

Langkah pertama adalah *Extraction*. Pada proses ini, data operasional yang diperlukan pada *data warehouse* akan dibaca untuk kemudian masuk ke proses *cleansing*. Data yang akan diambil nantinya adalah data penjualan, data penerimaan, dan data pengembalian. Data - data akan diambil dari tabel penjualan, penerimaan, pengembalian, barang, kategori, warna, desain, grup\_musik, toko, kota, dan propinsi. Untuk data penjualan, data penerimaan, dan data pengembalian tidak akan diambil keseluruhan. Untuk data - data tersebut akan dirangkum menjadi data harian.

Langkah kedua adalah *Cleansing*. Pada proses ini, data operasional yang telah dibaca akan diperbaiki dari kesalahan - kesalahan pada proses input data. Misalnya terjadi redudansi data, nilai suatu field yang tidak sesuai, ketidak konsistenan dari data, dan lain - lain. Disini juga record yang salah satu fieldnya memiliki null value akan dibersihkan. Kode barang juga akan dicocokkan kembali dengan kode kategori dan kode grup musik. Jika ada

kode barang yang tidak ada pada kode kategori ataupun kode grup musik, maka record tersebut tidak akan dimasukkan.

Langkah ketiga adalah *Transformation*. Pada proses ini, data yang sudah dibersihkan, diubah dari format data operasional menjadi format *data warehouse*. Untuk proses ini, format yang akan diubah hanya format tanggal saja, baik pada data penjualan, data penerimaan, maupun data pengembalian. Tanggal tersebut nantinya akan dipecah menjadi tanggal, bulan, dan tahun untuk dimasukkan ke tabel tanggal di *data warehouse*.

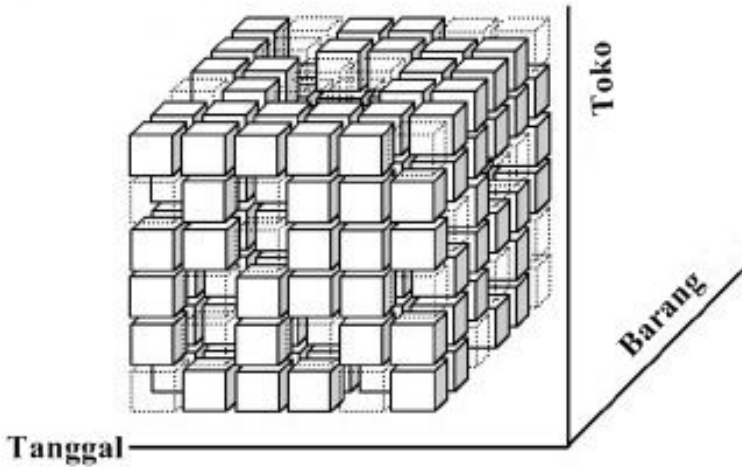
Langkah keempat adalah *Loading*. Pada proses ini, data yang sudah dibaca, dibersihkan, dan dirubah formatnya, akan disimpan pada *data warehouse*. Teknik yang akan digunakan adalah *update*. Data yang sudah ada tidak akan dihapus atau diubah karena data akan di-*update* secara berkala. Nantinya semua data yang sudah melalui proses *extraction*, *cleansing*, dan *transformation* akan langsung dimasukkan ke *data warehouse* tanpa merubah data yang sudah ada.

#### OLAP (ONLINE ANALYTICAL PROCESSING)

Setelah *data warehouse* terbentuk, langkah terakhir adalah melakukan pengambilan data dari *data warehouse*. Dalam pembangunan *data warehouse* kali ini, hasil output dari *data warehouse* akan berupa laporan dan juga digunakan untuk analisis data dengan *OLAP*.

#### SLICING DAN DICING

*Slicing* dan *dicing* adalah proses mengambil potongan kubus berdasarkan nilai tertentu pada satu dimensi atau beberapa dimensinya yang diperlihatkan gambar 5.

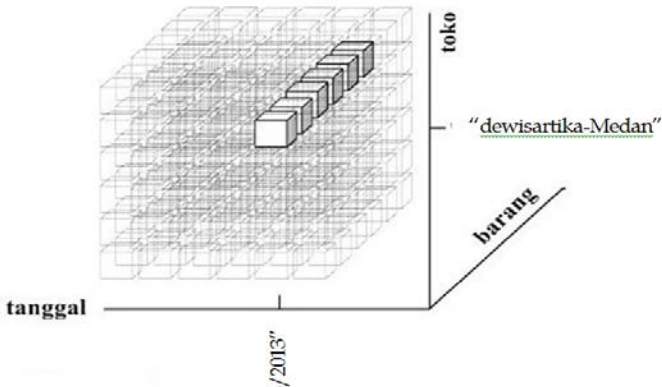


Gambar 5. Illustration Cube

Table 1. Sample Data

Kode Barang	Kategori	Grup Musik	Size	Tanggal	Toko	Jumlah Penjualan	Total Pendapatan
BJURCD001	Baju	Rancid	L	12/1/2013	dewi sartika - Medan	4	Rp 200,000.00
BJURCD001	Baju	Rancid	L	12/1/2013	dalem kaum - Medan	2	Rp 100,000.00
JUMAVS002	Jumper	Avenged Sevenfold	L	12/3/2013	dewi sartika - Medan	6	Rp 480,000.00
JUMAVS002	Jumper	Avenged Sevenfold	L	12/3/2013	dalem kaum - Medan	1	Rp 80,000.00
BJURCD003	Baju	Rancid	L	12/3/2013	dewi sartika - Medan	2	Rp 90,000.00
BJURCD003	Baju	Rancid	L	12/3/2013	dalem kaum - Medan	7	Rp 315,000.00
BJUAVS001	Baju	Avenged Sevenfold	M	12/3/2013	dewi sartika - Medan	4	Rp 200,000.00
BJUAVS001	Baju	Avenged Sevenfold	M	12/3/2013	dalem kaum - Medan	1	Rp 50,000.00
BJURCD001	Baju	Rancid	M	12/4/2013	dewi sartika - Medan	2	Rp 100,000.00
BJURCD001	Baju	Rancid	M	12/4/2013	dalem kaum - Medan	5	Rp 250,000.00

Berikut adalah sample dari *slicing*. Jika dari tabel 1, Jika ingin melihat data penjualan dari toko dewi sartika di Medan pada tanggal 3 Desember 2013, maka lakukan seleksi pada data diatas berdasarkan dimensi toko dan tanggal. Hasilnya akan diperlihatkan pada gambar 6 yaitu:

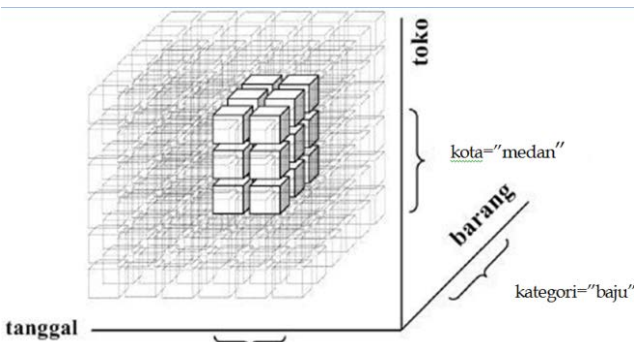


Gambar 6. Illustration Slicing

Table 2. Sample Data Slicing

Kode Barang	Kategori	Grup Musik	Size	Tanggal	Toko	Jumlah Penjualan	Total Pendapatan
JUMAVS002	Jumper	Avenged Sevenfold	XL	12/3/2013	dewi sartika - Medan	6	Rp 480,000.00
BJURCD003	Baju	Rancid	L	12/3/2013	dewi sartika - Medan	2	Rp 90,000.00
BJUAVS001	Baju	Avenged Sevenfold	XL	12/3/2013	dewi sartika - Medan	4	Rp 200,000.00

Berikut adalah *sample* dari *dicing*. Jika dari tabel diatas, kita ingin melihat data penjualan dari toko yang berada di kota Medan pada Desember 2013, maka lakukan pengelompokan dan seleksi berdasarkan turunan dari dimensi toko , tanggal dan barang. Hasilnya akan seperti gambar 7 dibawah ini.



Gambar 7. Illustration Dicing

Table 3. Sample Data Dicing

Kode Barang	Kategori	Grup Musik	Size	Tanggal	Toko	Jumlah Penjualan	Total Pendapatan
BJURCD001	Baju	Rancid	L	12/1/2013	dewi sartika - Medan	4	Rp 200,000.00
BJURCD001	Baju	Rancid	L	12/1/2013	dalem kaum - Medan	2	Rp 100,000.00
BJURCD003	Baju	Rancid	L	12/3/2013	dewi sartika - Medan	2	Rp 90,000.00
BJURCD003	Baju	Rancid	L	12/3/2013	dalem kaum - Medan	7	Rp 315,000.00
BJUAVS001	Baju	Avenged Sevenfold	XL	12/3/2013	dewi sartika - Medan	4	Rp 200,000.00
BJUAVS001	Baju	Avenged Sevenfold	XL	12/3/2013	dalem kaum - Medan	1	Rp 50,000.00
BJURCD001	Baju	Rancid	XL	12/4/2013	dewi sartika - Medan	2	Rp 100,000.00
BJURCD001	Baju	Rancid	XL	12/4/2013	dalem kaum - Medan	5	Rp 250,000.00

### Implementasi Database

Pembuatan database dilakukan dengan menggunakan aplikasi DBMS MySQL. Implementasi database dalam bahasa SQL adalah sebagai berikut:

#### Tabel kategori

```
CREATE TABLE `kategori` (
  `id_kategori` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `kode_kategori` varchar(3) DEFAULT NULL,
  `nama_kategori` varchar(20) NOT NULL,
  PRIMARY KEY (`id_kategori`),
  UNIQUE KEY `un_kategori` (`kode_kategori`)
)
```

#### Tabel warna

```
CREATE TABLE `warna` (
  `id_warna` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `kode_warna` varchar(2) NOT NULL,
  `nama_warna` varchar(20) NOT NULL,
  PRIMARY KEY (`id_warna`)
)
```

#### Tabel desain

```
CREATE TABLE `desain` (
  `id_desain` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_grup_musik` int(11) unsigned NOT NULL,
  `nama_desain` varchar(50) NOT NULL,
  `gambar_desain` varchar(50) NOT NULL,
  PRIMARY KEY (`id_desain`),
  KEY `FK_desain_grup_musik` (`id_grup_musik`),
)
```

```

    CONSTRAINT `FK_desain_grup_musik` FOREIGN KEY
    (`id_grup_musik`) REFERENCES `grup_musik`
    (`id_grup_musik`) ON UPDATE CASCADE
  )

```

#### Tabel grup\_musik

```

CREATE TABLE `grup_musik` (
  `id_grup_musik` int(11) unsigned NOT NULL
  AUTO_INCREMENT,
  `kode_grup_musik` varchar(3) NOT NULL,
  `nama_grup_musik` varchar(20) NOT NULL,
  PRIMARY KEY (`id_grup_musik`)
)

```

#### Tabel barang

```

CREATE TABLE `barang` (
  `id_barang` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_kategori` int(11) unsigned NOT NULL,
  `id_warna` int(11) unsigned NOT NULL,
  `id_desain` int(11) unsigned NOT NULL,
  `kode_barang` varchar(9) DEFAULT NULL,
  `ukuran` varchar(2) NOT NULL,
  `gambar_barang` varchar(100) NOT NULL,
  `harga` int(11) NOT NULL,
  PRIMARY KEY (`id_barang`),
  UNIQUE KEY `U_kode_barang` (`kode_barang`),
  KEY `FK_barang_kategori` (`id_kategori`),
  KEY `FK_barang_warna` (`id_warna`),
  KEY `FK_barang_desain` (`id_desain`),
  CONSTRAINT `FK_barang_desain` FOREIGN KEY (`id_desain`)
  REFERENCES `desain` (`id_desain`) ON UPDATE CASCADE,
  CONSTRAINT `FK_barang_kategori` FOREIGN KEY
  (`id_kategori`) REFERENCES `kategori` (`id_kategori`) ON
  UPDATE CASCADE,
  CONSTRAINT `FK_barang_warna` FOREIGN KEY (`id_warna`)
  REFERENCES `warna` (`id_warna`) ON UPDATE CASCADE
)

```

#### Tabel tanggal

```

CREATE TABLE `tanggal` (
  `id_tanggal` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `tanggal` int(2) unsigned NOT NULL,
  `bulan`
  enum('Januari','Pebruari','Maret','April','Mei','Juni','Ju
  li','Agustus','September','Oktober','November','Desember')
  NOT NULL,
  `tahun` int(4) unsigned NOT NULL,
  PRIMARY KEY (`id_tanggal`)
)

```

#### Tabel propinsi

```

CREATE TABLE `propinsi` (
  `id_propinsi` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `nama_propinsi` varchar(50) NOT NULL,
  PRIMARY KEY (`id_propinsi`)
)

```

**Tabel kota**

```
CREATE TABLE `kota` (  
  `id_kota` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `id_propinsi` int(11) unsigned NOT NULL,  
  `nama_kota` varchar(30) NOT NULL,  
  PRIMARY KEY (`id_kota`),  
  KEY `FK_kota_propinsi` (`id_propinsi`),  
  CONSTRAINT `FK_kota_propinsi` FOREIGN KEY  
  (`id_propinsi`) REFERENCES `propinsi` (`id_propinsi`) ON  
  UPDATE CASCADE  
)
```

**Tabel toko**

```
CREATE TABLE `toko` (  
  `id_toko` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `id_kota` int(11) unsigned NOT NULL,  
  `nama_toko` varchar(50) NOT NULL,  
  `alamat_toko` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_toko`),  
  KEY `FK_toko_kota` (`id_kota`),  
  CONSTRAINT `FK_toko_kota` FOREIGN KEY (`id_kota`)  
  REFERENCES `kota` (`id_kota`) ON UPDATE CASCADE  
)
```

**Tabel penjualan**

```
CREATE TABLE `penjualan` (  
  `id_penjualan` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `id_barang` int(11) unsigned NOT NULL,  
  `id_toko` int(11) unsigned NOT NULL,  
  `id_tanggal` int(11) unsigned NOT NULL,  
  `jumlah_penjualan` int(11) unsigned NOT NULL,  
  `subtotal` int(11) unsigned NOT NULL,  
  `total` bigint(20) unsigned NOT NULL,  
  PRIMARY KEY (`id_penjualan`),  
  KEY `FK_penjualan_barang` (`id_barang`),  
  KEY `FK_penjualan_toko` (`id_toko`),  
  KEY `FK_penjualan_tanggal` (`id_tanggal`),  
  CONSTRAINT `FK_penjualan_barang` FOREIGN KEY  
  (`id_barang`) REFERENCES `barang` (`id_barang`) ON UPDATE  
  CASCADE,  
  CONSTRAINT `FK_penjualan_tanggal` FOREIGN KEY  
  (`id_tanggal`) REFERENCES `tanggal` (`id_tanggal`) ON  
  UPDATE CASCADE,  
  CONSTRAINT `FK_penjualan_toko` FOREIGN KEY (`id_toko`)  
  REFERENCES `toko` (`id_toko`) ON UPDATE CASCADE  
)
```

**Tabel penerimaan**

```
CREATE TABLE `penerimaan` (  
  `id_penerimaan` int(11) unsigned NOT NULL  
  AUTO_INCREMENT,  
  `id_barang` int(11) unsigned NOT NULL,  
  `id_toko` int(11) unsigned NOT NULL,  
  `id_tanggal` int(11) unsigned NOT NULL,
```

```

`jumlah_penerimaan` int(11) unsigned NOT NULL,
PRIMARY KEY (`id_penerimaan`),
KEY `FK_terima_barang` (`id_barang`),
KEY `FK_penerimaan_toko` (`id_toko`),
KEY `FK_penerimaan_tanggal` (`id_tanggal`),
CONSTRAINT `FK_penerimaan_tanggal` FOREIGN KEY
(`id_tanggal`) REFERENCES `tanggal` (`id_tanggal`) ON
UPDATE CASCADE,
CONSTRAINT `FK_penerimaan_toko` FOREIGN KEY (`id_toko`)
REFERENCES `toko` (`id_toko`) ON UPDATE CASCADE,
CONSTRAINT `FK_terima_barang` FOREIGN KEY (`id_barang`)
REFERENCES `barang` (`id_barang`) ON UPDATE CASCADE
)

```

#### Tabel pengembalian

```

CREATE TABLE `pengembalian` (
  `id_pengembalian` int(11) unsigned NOT NULL
  AUTO_INCREMENT,
  `id_barang` int(11) unsigned NOT NULL,
  `id_toko` int(11) unsigned NOT NULL,
  `id_tanggal` int(11) unsigned NOT NULL,
  `jumlah_pengembalian` int(11) unsigned NOT NULL,
  PRIMARY KEY (`id_pengembalian`),
  KEY `FK_terima_barang` (`id_barang`),
  KEY `FK_pengembalian_toko` (`id_toko`),
  KEY `FK_pengembalian_tanggal` (`id_tanggal`),
  CONSTRAINT `FK_pengembalian_barang` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`) ON UPDATE
CASCADE,
CONSTRAINT `FK_pengembalian_tanggal` FOREIGN KEY
(`id_tanggal`) REFERENCES `tanggal` (`id_tanggal`) ON
UPDATE CASCADE,
CONSTRAINT `FK_pengembalian_toko` FOREIGN KEY
(`id_toko`) REFERENCES `toko` (`id_toko`) ON UPDATE
CASCADE
)

```

## KESIMPULAN

Setelah melakukan implementasi serta pengujian pada sistem yang dibangun, maka dapat diperoleh kesimpulan sebagai berikut:

1. Dapat meningkatkan strategi penjualan produk .
2. Memudahkan analisis sinkronisasi data antara pusat dan cabang.
3. Memudahkan pemberian *feedback* oleh pihak kantor cabang.
4. Membantu pihak manajemen PT.XYZ dalam menghasilkan laporan yang dapat digunakan untuk menentukan strategi produksi dan distribusi.



## SARAN

Penelitian dapat dikembangkan dengan model penambahan keamanan, optimization dan performance dalam jumlah data yang lebih besar.

## REFERENSI

- Connolly, Thomas and Carolyn Begg. (2005). *“Database System: A Practical Approach to Design, Implementation, and Management”*, 4th Edition. Addison Wesley: Harlow, England.
- McLeod, Raymond and Schell, George. (2007). *“Management Information Systems”*, 10th edition, International Edition, Pearson Prentice Hall, New Jersey.
- Randy O.P. (2012). *“Rancang bangun data warehouse untuk analisis kinerja penjualan pada industri dengan model spa-dw”* Tesis, UNDIP.
- R.P. Aloysius. (2011). *“Pembangunan Data Warehouse di CV. Saswco Perdana”*, Tesis, UNIKOM.
- Thomas C. Hammergren and Alan R. Simon (2009) *“Data Warehousing for Dummies 2nd Edition”*, Wiley Publishing, Inc.
- William H, Inmon. (2005) *“Building The Data Warehouse 4<sup>th</sup> Edition”*, John Willey & Sons, Canada.
- W.H., Inmon, Richard D.H. (2005) *“Building The Data Warehouse 4<sup>th</sup> Edition”*, Wiley Publishing, Inc.