

# Analisis Data dengan Menggunakan ERD dan Model Konseptual Data Warehouse

Doro Edi<sup>1)</sup>, Stevalin Betshani<sup>2)</sup>

Jurusan Sistem Informasi

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

Email: email : [k\\_doroedi@yahoo.com](mailto:k_doroedi@yahoo.com) <sup>1)</sup>, [vall\\_angel@yahoo.com](mailto:vall_angel@yahoo.com) <sup>2)</sup>

## Abstract

*Data is an important part in enterprise information system. So data can be used effectively, we have to analyse the data. There are many ways to analysing and modelling data, some of them are by using Entity Relationship Diagram (ERD) and conceptual model of data warehouse such as star schema, snowflake schema, dan fact constellations schema. This paper suggest a literature study about data analysis by using ERD and conceptual model of data warehouse and also a case study about mini market information system to support that explanation. Design of ERD, star schema, snowflake schema, dan fact constellations schema in that information system meant to manage point of sale, purchasing, and stock control. Design of ERD can be used to modelling transactional data. While data warehouse more used to support manager to make a decision in an enterprise.*

**Keywords:** Entity Relationship Design, Star schema, Snowflakes schema, Fact constellation schema.

## 1. Pendahuluan

Data merupakan komponen utama dari sistem informasi perusahaan karena semua informasi untuk pengambilan keputusan berasal dari data. Oleh karena itu sudah sewajarnya jika pengolahan data dipandang sebagai kebutuhan primer oleh perusahaan. Pengelolaan data yang buruk dapat mengakibatkan tidak tersedianya data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

Untuk mengolah data menjadi bentuk yang lebih bermanfaat dibutuhkan analisis yang baik dan tajam. Analisis data merupakan metode yang digunakan untuk mengetahui bagaimana menggambarkan data, hubungan data, semantik data dan batasan data yang ada pada suatu sistem informasi. Ada banyak cara dalam menganalisis dan memodelkan suatu data, beberapa diantaranya adalah dengan menggunakan *Entity Relationship Diagram (ERD)* dan *Model Konseptual Data Warehouse*.

Dalam melakukan analisis data, ERD dapat digunakan untuk menggambarkan masing-masing entitas dan relasi antar entitas dari bentuk notasi grafik menjadi sebuah diagram data sehingga segala pemrosesan data secara *transactional* dapat tergambar dengan jelas. Sedangkan model konseptual *data warehouse* dapat menunjang keputusan manajemen yang berorientasi subjek, terpadu, *time variant*, dan tidak mudah berubah. Dan merupakan penunjang pemrosesan informasi dengan menyediakan suatu *platform* yang kokoh untuk analisis data yang mengandung histori dan yang terkonsolidasi. Metode yang umum digunakan dalam

*data warehouse* untuk memodelkan dan menganalisa data adalah dengan menggunakan *star schema*, *snowflake schema*, dan *fact constellations schema*.

## 2. Konsep Data dan Analisis Data

### 2.1 Pengertian Data

Ada beberapa gagasan dalam pendefinisian istilah data, sebuah sumber menyebutkan bahwa :

*“Data adalah fakta-fakta, pemikiran, atau pendapat yang tidak atau belum memiliki arti kegunaan”* [Sco94]

Sedangkan pengertian lain dari data ialah :

*“Data didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya”* [Dav84]

Data terbentuk dari karakter yang dapat berupa alfabet, angka, maupun simbol khusus dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.

### 2.2 Hirarki Data

Secara tradisional, data di organisasikan ke dalam suatu hirarki yang terdiri atas [Kad99] :

#### 1. Elemen data

Adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Istilah lain untuk elemen data adalah *field*, *kolom*, *item*, dan *atribut*.

#### 2. Record

Adalah gabungan sejumlah elemen data yang saling terkait. Dalam sistem basis data relasional, *record* biasa disebut dengan istilah *tupel* atau *baris*.

#### 3. File

Himpunan seluruh *record* yang bertipe sama membentuk sebuah *file*. *File* dapat dikatakan sebagai kumpulan data *record* yang berkaitan dengan suatu subjek. Dalam sistem basis data relasional, *file* mewakili komponen yang disebut *tabel* atau *relationship*.

### 2.3 Pengertian Analisis Data

Analisis data merupakan metode yang digunakan untuk mengetahui bagaimana menggambarkan data, hubungan data, semantik data dan batasan data yang ada pada suatu sistem informasi.

## 3. Data Warehouse

### 3.1 Pengenalan Data Warehouse

Terdapat beberapa definisi seputar *data warehouse*. Dari sisi praktisi, Barry Devlin, IBM Consultant, menerjemahkan *data warehouse* sebagai :

*“Suatu data warehouse sederhananya adalah suatu penyimpanan data tunggal, lengkap dan konsisten, yang diperoleh dari berbagai sumber dan dibuat tersedia bagi end user dalam suatu cara yang bisa mereka pahami dan bisa mereka gunakan dalam suatu konteks bisnis.”* [Riz07]

Pendapat lain tentang pengertian *data warehouse* yang dikemukakan oleh W. H. Inmon, yang dikenal juga sebagai Bapak *data warehousing*, adalah :

“Suatu data warehouse adalah suatu koleksi data yang bisa digunakan untuk menunjang pengambilan keputusan manajemen, yang berorientasi subjek (topik), terpadu, time variant, dan tidak mudah berubah.” [Han06]

Secara garis besar, *data warehouse* adalah sebuah *database* penunjang keputusan yang mengandung data yang biasanya mewakili sejarah bisnis dari suatu perusahaan. Data Historis dari *data warehouse* digunakan di dalam aktivitas analisis yang mendukung pengambilan keputusan dalam perusahaan tersebut. Pengelolaan *data warehouse* dilakukan secara terpisah dari *database* operasional perusahaan.

### 3.2 Karakteristik Data Warehouse

Beberapa karakteristik dari *data warehouse* yaitu sebagai berikut :

#### 1. *Subject Oriented*

Yaitu mengorganisir data menurut subjek dari suatu perusahaan, misalnya konsumen, produk, dan penjualan. Difokuskan pada pemodelan dan analisis data untuk *decision maker*, bukan pada operasi harian atau pemrosesan transaksi. Menyediakan wawasan yang sederhana dan ringkas mengenai subjek dengan memisahkan data yg tidak relevan dalam proses pengambilan keputusan.

#### 2. *Integrated*

*Data warehouse* harus mengintegrasikan data dari sumber data yang beragam, seperti *relational database*, *flat files*, *on-line transaction records*.

#### 3. *Time-variant*

*Data warehouse* tetap menyimpan data-data historis. Setiap struktur *key* dalam *data warehouse* mengandung elemen waktu baik eksplisit maupun implisit.

#### 4. *Non-volatile*

Penyimpanan data transformasi dalam *data warehouse* selalu terpisah secara fisik dari lingkungan operasional. Oleh karena itu, *updatedata* operasional tidak terjadi pada lingkungan *data warehouse*, dan *data warehouse* tidak memerlukan pemrosesan transaksi, *recovery*, dan *concurrency control*. Hanya memerlukan dua operasi dalam megakses data, yaitu *initial loading of data* dan *access of data*.

### 3.3 Keuntungan Data Warehouse

Jika suatu perusahaan berhasil menerapkan *data warehouse* untuk mendukung pengambilan keputusan, maka beberapa keuntungan yang akan diberikan oleh *data warehouse* antara lain :

#### 1. Kemungkinan kembalinya nilai investasi dalam jumlah besar

Perusahaan harus menanamkan modal dalam jumlah yang cukup besar untuk menjamin penerapan *data warehouse* pada perusahaan tersebut dapat berhasil. Namun penelitian yang dilakukan oleh *International Data Corporation* (IDC) pada tahun 1996 menunjukkan bahwa dalam jangka waktu kurang lebih tiga tahun nilai *Return of Investment* (ROI), dari perusahaan yang menerapkan *data warehouse*, meningkat sebesar 401%. [Con02]

#### 2. Keunggulan bersaing (*competitive advantage*)

Besarnya ROI untuk perusahaan yang mengimplementasikan *data warehouse* merupakan bukti akan keunggulan bersaing yang luar biasa sebagai efek dari teknologi ini. Keunggulan bersaing diperoleh karena *decision maker* dapat

mengakses data yang sebelumnya tidak tampak, tidak diketahui dan belum dimanfaatkan, seperti permintaan konsumen, tren pada konsumen.

### 3. Meningkatkan produktivitas pembuat keputusan

*Data warehouse* meningkatkan produktivitas *decision maker* dengan menciptakan *database* yang konsisten, *subject-oriented*, dan mendukung data historis. Dengan kata lain, *data warehouse* mengintegrasikan data dari berbagai sistem yang bertentangan menjadi suatu bentuk yang menyediakan gambaran perusahaan yang konsisten.

## 3.4 Model Konseptual Data Warehouse

Model konseptual *data warehouse* atau *dimensional model* merupakan suatu desain *logic* yang merepresentasikan data dalam bentuk standar, dan mendukung dilakukannya akses terhadap data dengan cepat. Terdapat beberapa istilah yang berkaitan dengan model ini, yaitu *fact table*, *dimension table*, dan *hirarki*.

*Fact table* merupakan tabel utama yang berisikan kumpulan dari *primary key* tabel lain. Setiap *fact table* dalam model konseptual *data warehouse* memiliki *composite key*, begitu juga sebaliknya, setiap tabel yang memiliki *composite key* adalah *fact table*. Dengan kata lain, setiap tabel yang menunjukkan hubungan banyak ke banyak (*many to many relationship*) pastilah merupakan *fact table*.

*Dimension table* adalah tabel yang lebih sederhana dimana didalamnya terdapat *primary key* yang berhubungan dengan salah satu *composite key* yang ada pada *fact table*.

*Hirarki* mendefinisikan urutan pemetaan dari konsep level bawah ke level yang lebih tinggi, konsep yang lebih umum.

*Dimensional model* yang paling populer adalah *star schema*, *snowflakes schema*, dan *fact constellation schema*. Masing-masing model diatas akan dibahas lebih lanjut pada pembahasan berikutnya.

### 3.4.1 Star Schema

*Star schema* adalah salah satu *dimensional model* dimana *fact table* terletak di pusat dan tabel lainnya, yaitu *dimension table* terletak disekelilingnya. Kebanyakan dari *fact table* pada *star schema* merupakan normalisasi bentuk ketiga dari *database*, sedangkan *dimensional table* adalah normalisasi bentuk kedua. *Star schema* merupakan bentuk *dimensional model* yang paling sederhana.

Beberapa keuntungan dari *star schema* adalah mudah dipahami karena strukturnya yang sederhana, dan *low maintenance*. Sedangkan kekurangannya adalah performanya kurang baik dibandingkan dengan skema yang lainnya, serta jumlah *dimension table* yang ada bisa menjadi sangat banyak.

### 3.4.2 Snowflake Schema

*Snowflake schema* merupakan bentuk perluasan dari *star schema* dimana terjadi proses normalisasi dari beberapa atau seluruh *dimension table*. *Snowflake schema* biasanya digunakan pada tabel yang sangat besar dan jika *star schema* tidak mampu menggambarkan kompleksitas dari *database* tersebut.

Seperti halnya *star schema*, *snowflake schema* juga mudah untuk di-*maintain*, tidak memakan banyak ruang pada media penyimpanannya, serta mengurangi redudansi data karena skema ini menggunakan tabel yang telah dinormalisasi. Beberapa

permasalahan yang ditemui pada skema ini antara lain pada penggunaan SQL yang lebih rumit. Hal ini dikarenakan untuk menjawab *query-query* yang diinginkan, perlu dilakukan *join* dan agregasi pada banyak tabel.

### 3.4.3 Fact Constellation Schema

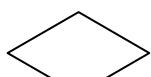
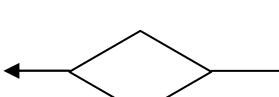
Skema ini lebih kompleks dibanding *snowflake schema* atau *star schema* karena skema ini berisi banyak *fact table*. *Fact constellation schema* memungkinkan suatu *dimension table* berhubungan dengan banyak *fact table*. *Fact constellation schema* sangat fleksibel, namun terkadang menjadi susah dalam pengaturan dan *support*. Kerugian utama skema ini adalah desain lebih rumit karena banyak varian agregasi yang harus dipertimbangkan, selain itu juga untuk menjawab *single query* mungkin butuh *multiple SQL statement*.

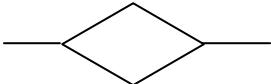
## 4. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi.

Komponen-komponen pembentuk ERD dapat di lihat pada tabel 1 di bawah ini.

**Tabel 1: Komponen-komponen ERD**

Notasi	Komponen	Keterangan
	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
	Relasi	Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
	Relasi 1 : 1	Relasi yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua
	Relasi 1 : N	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berrelasi dengan banyak entitas pada himpunan entitas yang lain

Notasi	Komponen	Keterangan
	Relasi N : N	Hubungan ini menunjukkan bahwa setiap entitas pada himpunan entitas yang pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, demikian juga sebaliknya

## 5. Perbandingan Antara ERD Dengan Dimensional Model

Seperti yang telah dijabarkan sebelumnya, ERD digunakan untuk menggambarkan *database* untuk pemrosesan transaksi *online* atau *Online Transaction Processing* (OLTP), sedangkan *dimensional model* digunakan untuk mendesain *database* dalam *data warehouse*. Berikut ini merupakan perbandingan antara OLTP dengan *data warehouse*.

**Tabel 2: Perbandingan antara OLTP dengan *Data Warehouse***

OLTP	<i>Data Warehouse</i>
Menangani data saat ini	Menangani data historis
Datanya dinamis	Datanya sebagian besar statis
Banyak menangani proses <i>update</i>	Banyak menangani proses <i>read</i>
Pola penggunaannya dapat diprediksi	Pola penggunaannya tidak dapat diprediksi
Bersifat transaksional	Bersifat analisis
Banyak transaksi kecil	<i>Query-query</i> panjang dan kompleks
Data size MB – GB	Data size GB – TB
Berorientasi aplikasi	Berorientasi subjek
Mendukung keputusan harian	Mendukung keputusan strategis
Menangani pengguna operasional dalam jumlah besar	Menangani pengguna manajerial dalam jumlah kecil

Dari tabel diatas dapat diketahui bahwa ERD digunakan untuk memodelkan data-data yang bersifat transaksional. Sedangkan *dimensional model* lebih dimanfaatkan untuk memodelkan data-data yang akan digunakan untuk menunjang pengambilan keputusan.

## 6. Studi Kasus Minimarket

### 6.1 Deskripsi Permasalahan

Sebuah *minimarket* yang menjual berbagai jenis barang kebutuhan sehari-hari memiliki sebuah sistem informasi untuk mengelola penjualan secara langsung (*point of sales*), pengadaan barang, dan *stock control*. Proses bisnis dalam penjualan barangnya dimulai pada saat *customer* memilih barang yang akan dibeli. Setelah *customer* memutuskan untuk membeli barang tersebut, maka kasir akan meminta informasi tentang identitas *customer* untuk dicatat jika *customer* yang bersangkutan terdaftar sebagai *member*. Namun jika *customer* tersebut bukanlah

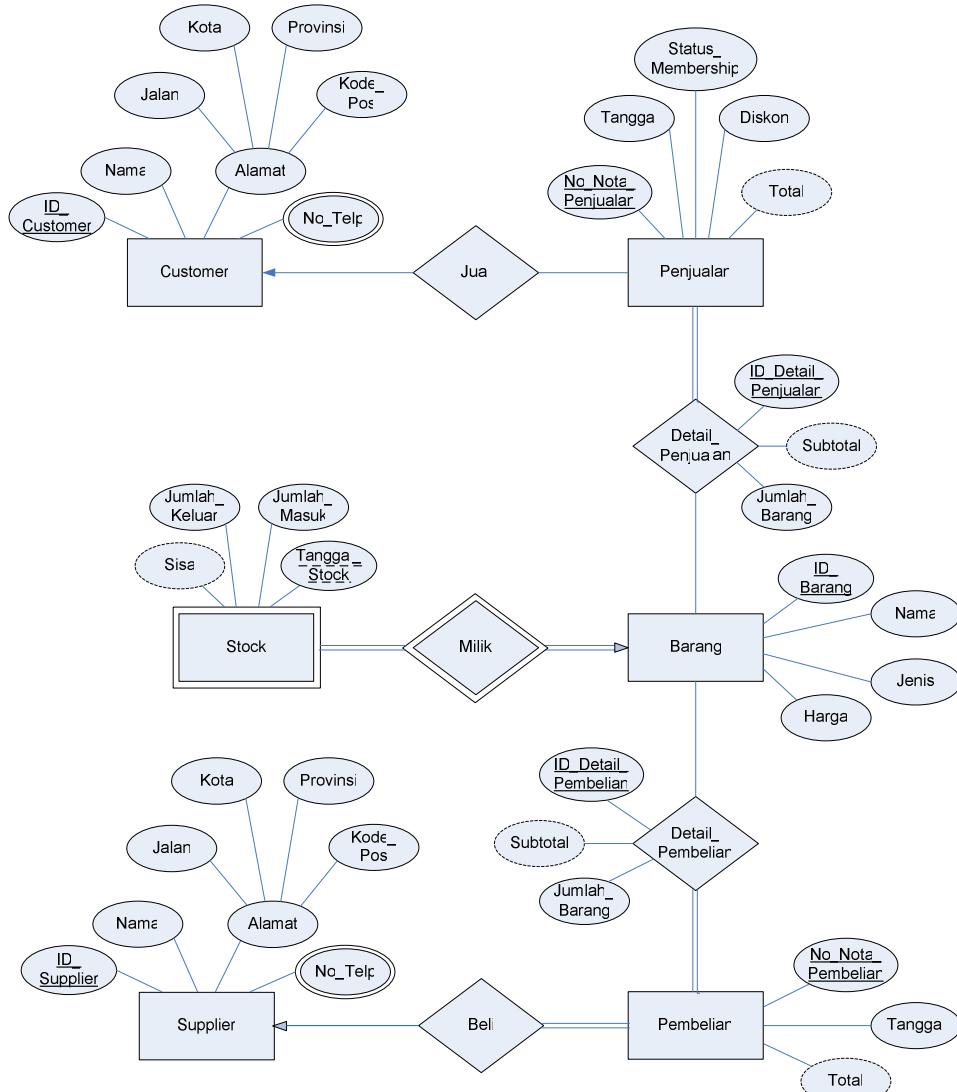
*member minimarket*, maka data-data *customer* akan diabaikan. Kemudian kasir akan membuatkan nota penjualan barang. Setelah barang diterima oleh *customer*, *customer* akan melakukan pembayaran. Proses berakhir ketika kasir memberikan bukti pembayaran kepada *customer*. Sistem informasi yang tersedia tidak melayani proses pengembalian barang dan pemesanan barang.

Proses bisnis untuk pembelian barang dari *supplier* dimulai ketika pihak *minimarket* menghubungi *supplier* dan memesan barang. *Supplier* kemudian akan membuatkan nota pembelian. Barang yang sudah dipesan lalu akan diantarkan ke *minimarket*. Jika barang sudah diterima, maka proses yang terjadi adalah pembayaran dari pihak *minimarket* ke pihak *supplier*. Setelah semua proses pembayaran selesai, *supplier* akan memberikan bukti pembayaran dan proses selesai. Seperti halnya pada proses penjualan, proses pembelian tidak menangani pengembalian barang kepada *supplier*.

Untuk proses *stock control*, dilakukan proses pencatatan terhadap barang yang *di-supply*, barang yang dibeli oleh *customer* dan sisa barang yang ada di gudang per harinya. Hal ini dimaksudkan agar setiap keluar masuknya barang yang ada dapat terawasi dan menjaga barang selalu tersedia di gudang.

## 6.2 Perancangan ERD

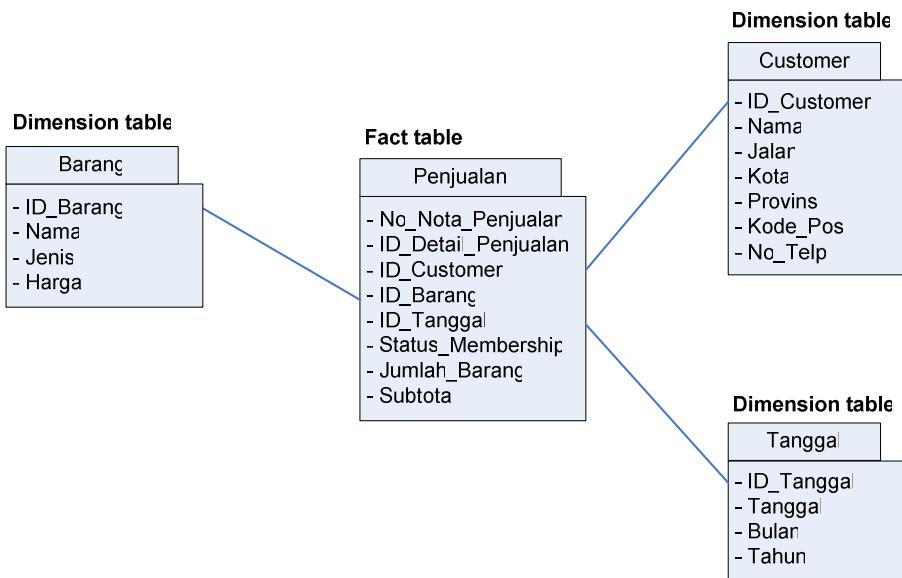
Berikut ini perancangan ERD untuk *minimarket* tersebut diatas :



Gambar 1: Entity Relationship Diagram Minimarket

### 6.3 Perancangan Star Schema

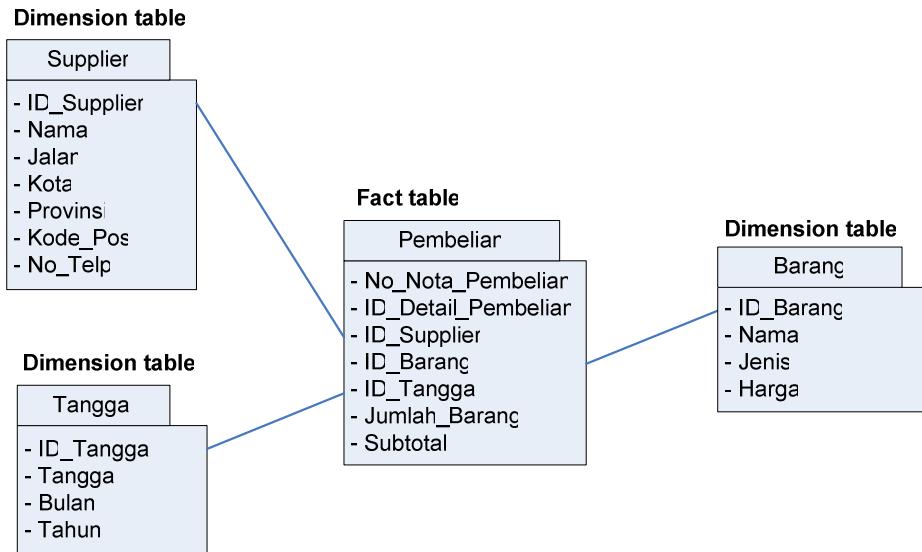
*Star schema* merupakan salah satu alat pendukung pengambilan keputusan, maka dari itu perancangan *star schema* disesuaikan dengan kebutuhan pihak manajerial dalam pengambilan suatu keputusan. Jika pihak manajerial membutuhkan data-data mengenai penjualan untuk mengambil keputusan tertentu, maka akan dirancang *star schema* untuk penjualan saja. Berikut adalah perancangan *star schema* untuk proses penjualan :



Gambar 2: *Star Schema* Penjualan

Dari perancangan *star schema* diatas, yang berperan sebagai *fact table* adalah tabel penjualan. Tabel penjualan disini merupakan penggabungan dari beberapa atribut dari entitas penjualan dengan beberapa atribut dari entitas detail\_penjualan pada *ERD* yang dirancang sebelumnya. Penggabungan ini dapat dilakukan selagi hal tersebut dapat mendukung pengambilan keputusan. Tabel penjualan dijadikan sebagai *fact table* karena tabel penjualan merupakan tabel utama yang berisikan kumpulan *primary key* dari tabel-tabel lainnya. *Dimension table* untuk skema diatas adalah tabel *customer*, tabel barang, dan tabel tanggal karena *primary key* dari tabel-tabel tersebut berhubungan dengan salah satu *composite key* yang ada pada *fact table*.

*Star schema* untuk pembelian akan dirancang jika manajer membutuhkan data-data pendukung pengambilan keputusan yang berhubungan dengan pembelian. Berikut ini merupakan perancangan *star schema* untuk proses pembelian :



Gambar 3: *Star Schema* Pembelian

Pada *star schema* diatas yang berperan sebagai *fact table* adalah tabel pembelian sedangkan *dimension table*-nya adalah tabel *supplier*, tabel barang, dan tabel tanggal. Seperti pada *star schema* penjualan, tabel pembelian dijadikan sebagai *fact table* karena tabel ini merupakan tabel utama yang berisikan kumpulan *primary key* dari tabel-tabel lainnya. Tabel pembelian disini juga merupakan penggabungan dari beberapa *field* pada tabel pembelian dan tabel detail\_pembelian pada ERD sebelumnya. Tabel *supplier*, tabel barang, dan tabel tanggal dijadikan sebagai *dimension table* karena *primary key* dari tabel-tabel tersebut berhubungan dengan salah satu *composite key* yang ada pada tabel pembelian.

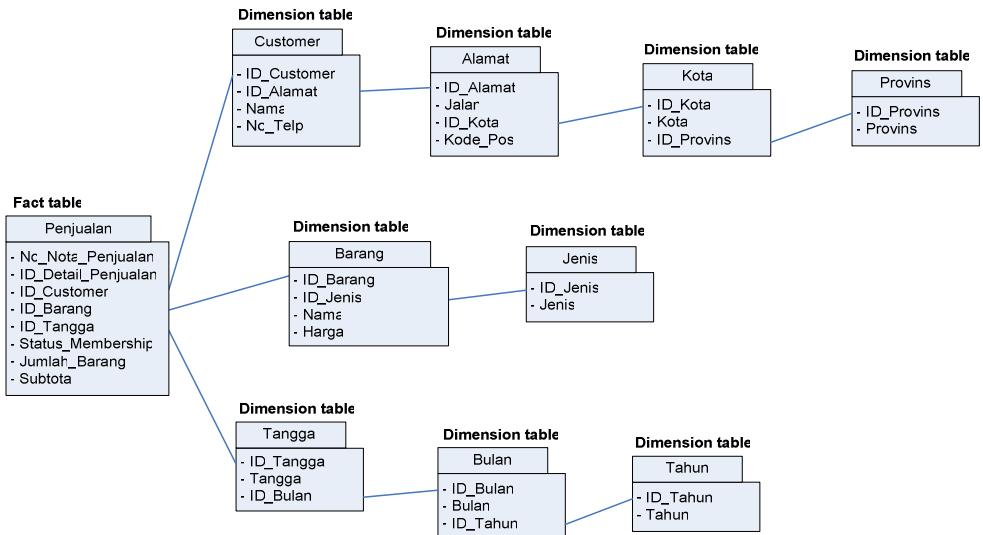
Berikut ini merupakan *star schema* untuk proses *stock control* pada *minimarket* yang telah dijabarkan sebelumnya. Dalam skema ini yang berperan sebagai *fact table* adalah tabel *stock*, sedangkan *dimension table*-nya adalah tabel tanggal dan tabel barang. Pada tabel *stock* terdapat *primary key* dari tabel barang yaitu *ID\_Tanggal* serta terdapat pula *primary key* dari tabel barang yaitu *ID\_Barang*.



Gambar 4: *Star Schema* Stock

Dari seluruh perancangan *star schema* diatas dapat diketahui bahwa perancangan tiap skema hanya terbatas berdasarkan satu proses tertentu saja, misalnya proses penjualan saja, proses pembelian saja atau proses *stock control* saja.

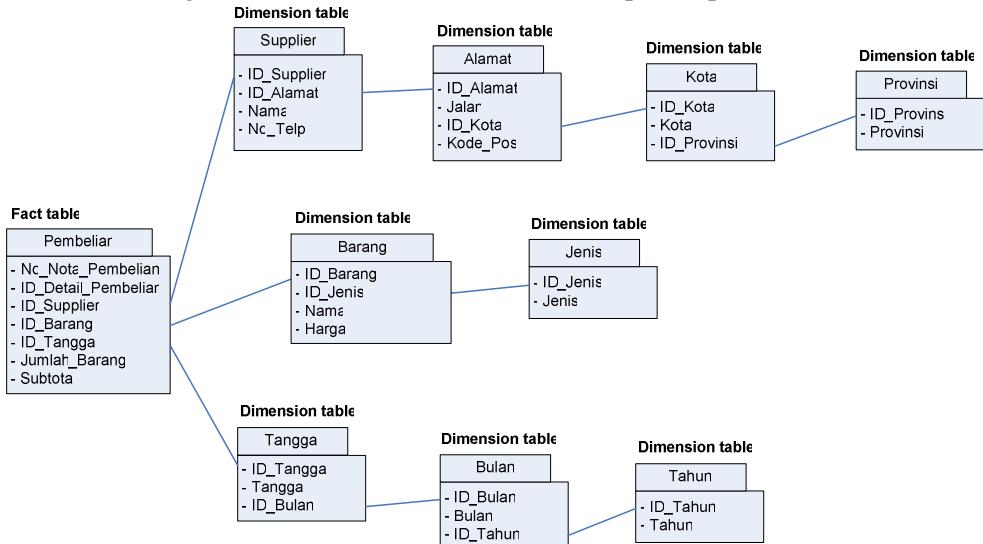
Perancangan Snowflake Schema



Gambar 5: *Snowflakes Schema Penjualan*

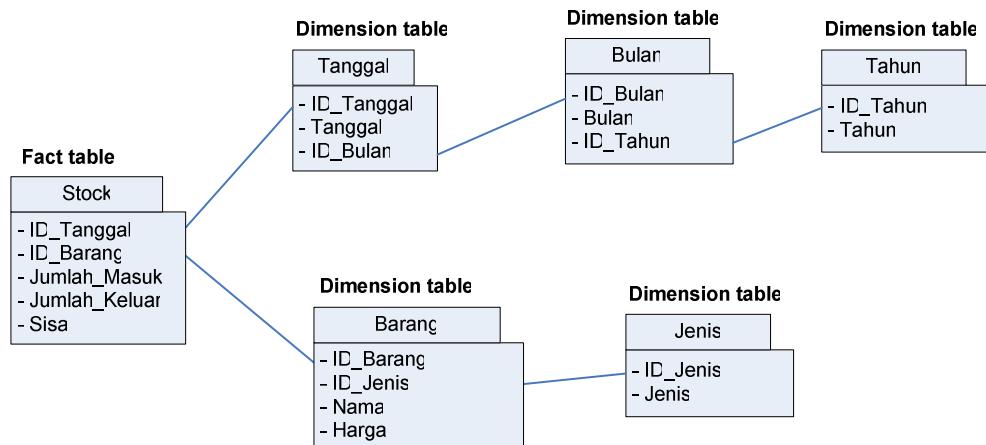
*Snowflakes schema* merupakan pengembangan dari *star schema*, dimana setiap *dimension table* yang ada dipecah kembali menjadi bentuk yang lebih sederhana sesuai dengan kebutuhan pengambilan keputusan. Gambar 5 di atas merupakan gambar *snowflakes schema* untuk proses penjualan.

*Star schema* pembelian dapat dipecah kembali menjadi bentuk yang lebih detail pada *dimension table*-nya sehingga akan menghasilkan sebuah *snowflake schema*. Berikut adalah gambar dari *snowflake schema* untuk proses pembelian :



Gambar 6: *Snowflakes Schema Pembelian*

Sedangkan gambar dibawah ini merupakan gambar *snowflakes schema* dari proses *stock control* :

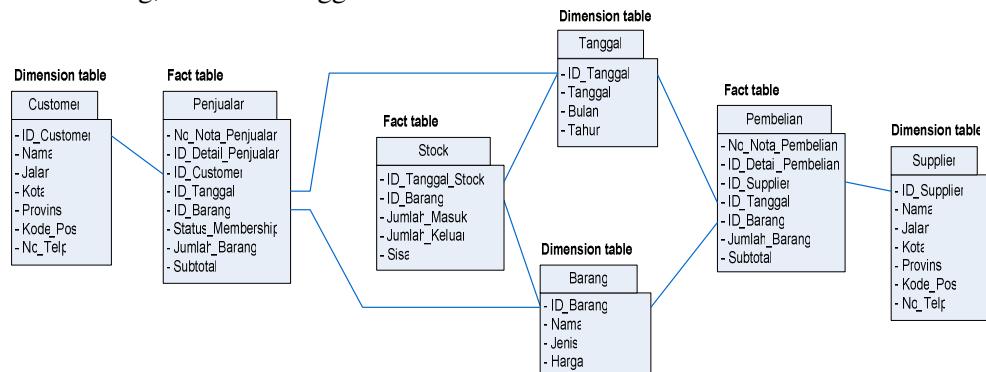


Gambar 7: Snowflakes Schema Stock

Pada ketiga *snowflakes schema* diatas diketahui bahwa tiap-tiap *dimension table* mengalami normalisasi. Hal ini bertujuan untuk mengetahui hirarki dari tabel-tabel yang ada sehingga dapat mendukung pengambilan keputusan berdasarkan hirarki tersebut.

## 6.5 Perancangan Fact Constellation Schema

Berikut ini merupakan gambar perancangan *fact constellation schema* untuk proses penjualan, pembelian, dan *stock control* pada *minimarket* yang telah dijabarkan sebelumnya. *Fact table* dari skema ini adalah tabel penjualan, tabel pembelian, dan tabel *stock* sedangkan *dimension table*-nya adalah tabel *customer*, tabel *supplier*, tabel barang, dan tabel tanggal.



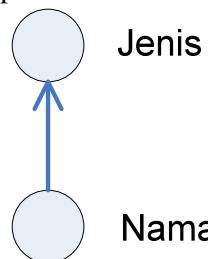
Gambar 8: Fact Constellation Schema

Sebagai *fact table*, tabel pembelian, tabel penjualan, dan tabel *stock* berisikan *primary key* dari masing-masing *dimensional table*. Sebaliknya *primary key* dari tabel *customer*, tabel *supplier*, tabel barang, dan tabel tanggal, yang berperan sebagai *dimensional table*, juga berhubungan dengan salah satu *composite key* yang ada pada *fact table*.

*Fact constellation schema* diatas menunjukkan bahwa suatu *dimension table* dapat berhubungan dengan banyak *fact table*. Misalnya pada tabel barang dan tabel tanggal. Kedua tabel diatas berhubungan dengan tiga *fact table* sekaligus, yaitu tabel penjualan, tabel pembelian, dan tabel *stock*.

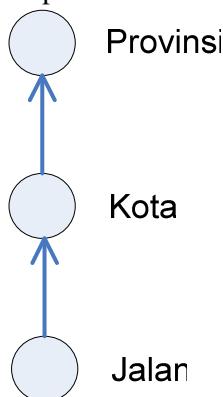
## 6.6 Hirarki dari Dimensional Table

Untuk masing-masing *dimension table* yang ada pada tiap-tiap skema yang telah dirancang sebelumnya, dapat ditarik suatu hirarki. Hirarki tersebut berguna agar tiap *dimension table* yang ada dapat di *aggregate / summarized* ke level yang lain. Berikut ini hirarki untuk tiap-tiap dimensi :



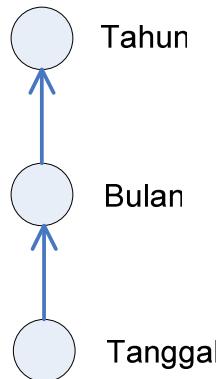
**Gambar 9: Hirarki untuk Dimensi Barang**

Gambar diatas menunjukkan hirarki untuk tabel barang. Pada tabel barang terdapat *field* Nama dan Jenis yang menunjukkan hirarki untuk dimensi barang. *Field* jenis menunjukkan adanya pengelompokan barang menjadi kelompok yang lebih *general* sedangkan *field* nama merupakan bentuk khusus dari barang itu sendiri.



**Gambar 10: Hirarki untuk Dimensi Supplier dan Customer**

Tabel *supplier* dan tabel *customer* juga memiliki hirarki yang ditunjukkan oleh gambar 10 diatas. Hirarki untuk tabel *supplier* dan tabel *customer* terletak pada lokasi dari *supplier* atau *customer* tersebut. Berdasarkan lokasinya, *supplier* atau *customer* dapat dikelompokkan berdasarkan provinsi, kota, atau jalan dimana *supplier* atau *customer* tersebut berada.



**Gambar 11: Hirarki untuk Dimensi Tanggal**

Hirarki untuk dimensi tanggal merupakan hirarki waktu. Tiap proses penjualan, pembelian, maupun pengawasan stok dapat dikelompokkan berdasarkan waktu tertentu, misalnya tiap tahun, tiap bulan, atau tiap harinya.

## 7. Kesimpulan

Setelah melakukan analisis dan perancangan data dengan menggunakan ERD dan model konseptual *data warehouse* dapat disimpulkan bahwa :

1. *Entity Relationship Diagram* (ERD) merupakan suatu metode pemodelan data yang menggambarkan entitas-entitas yang ada pada suatu *database* dan relasi atau hubungan dari masing-masing entitas tersebut.
2. *Star schema* merupakan salah satu model konseptual *data warehouse* yang paling sederhana, dimana hanya ada satu *fact table* yang dikelilingi oleh beberapa *dimension table*.
3. *Snowflake schema* merupakan pengembangan dari *star schema* yang didalamnya terjadi proses normalisasi dari beberapa *dimension table*-nya.
4. *Fact constellation schema* adalah skema yang paling kompleks dibandingkan dengan *star* dan *snowflakes schema*, dimana terdapat beberapa *fact table* dan *dimension table*-nya dapat berhubungan dengan lebih dari satu *fact table*.
5. Memodelkan data menggunakan *Entity Relationship Diagram* (ERD) digunakan untuk keperluan harian yang bersifat *transactional* sedangkan model konseptual *data warehouse*, yaitu *star schema*, *snowflakes schema*, dan *fact constellation schema*, dimanfaatkan untuk menunjang pengambilan keputusan.

## 8. Saran

Saran yang dapat disampaikan untuk kemajuan di masa mendatang, antara lain :

Untuk mengetahui secara lebih jelas tentang ERD dan model konseptual *data warehouse*, disarankan untuk melakukan uji coba secara langsung pada *database* dari suatu aplikasi yang berjalan.

Topik pembahasan sebelumnya dapat dikembangkan lagi dengan mengenakan operasi *on-line analytical processing* (OLAP), seperti *roll-up*, *drill-down*, *slice*, *dice*, *pivoting*, *ranking*, dalam menganalisis *data warehouse*.

## **9. Referensi**

- [Sco94] Scott, George M. (1994). Prinsip-prinsip Sistem Informasi Manajemen. Jakarta : PT. Raja Grafindo Persada.
- [Dav84] Davis, Gordon B. (1984). Kerangka Dasar Sistem Informasi Manajemen. Jakarta : PT. Pustaka Binaman Pressindo.
- [Con02] Connolly, T & Begg, C. (2002). Database System. UK : Addison Wesleys
- [Han06] Han, J & Kamber, M. (2006). Data Mining : Concepts and Techniques. San Francisco : Morgan Kaufmann Publishers.
- [Kad99] Kadir, A. (1999). Kosep dan Tuntunan Praktis Basis Data. Yogyakarta : Penerbit ANDI Yogyakarta.
- [Riz07] Rizky. (2007). Data Warehousing dan Decision Support. ITS. Available : [http://lecturer.eepis-its.edu/~rizky/Basdat\\_2\\_Teori/  
Day-13/13%20-%20Data%20Warehouse%20dan%20Decision  
%20Support%20System-2.pdf](http://lecturer.eepis-its.edu/~rizky/Basdat_2_Teori/Day-13/13%20-%20Data%20Warehouse%20dan%20Decision%20Support%20System-2.pdf).