

BAB II

LANDASAN TEORI

2.1 *Data Warehouse*

Menurut McLeod dan Schell (2004, p405), *data warehouse* adalah sebuah tempat penyimpanan data dimana kapasitas penyimpanannya berskala besar; datanya diakumulasikan dengan menambahkan *record* baru dan bukan dengan *update record* yang ada dengan informasi yang baru; data sangat mudah ditarik; dan data digunakan semata-mata untuk pengambilan keputusan dan bukan untuk kegiatan operasional perusahaan.^[1]

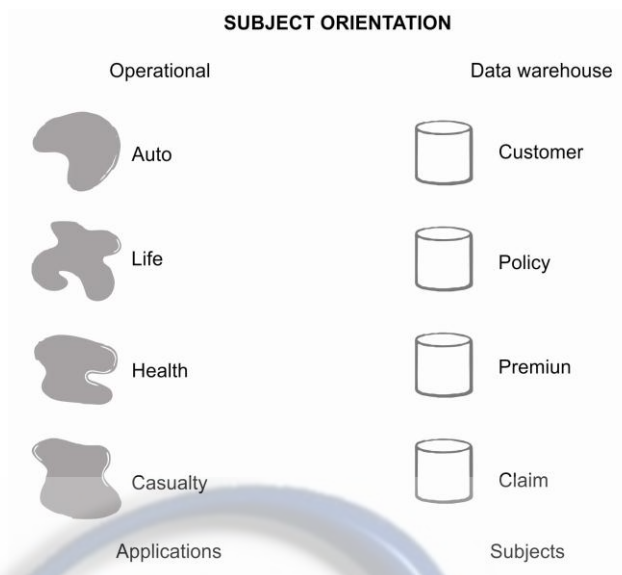
Data warehouse adalah sebuah koleksi *database* yang terintegrasi, beorientasi subjek yang dirancang untuk mendukung fungsi DSS, dimana setiap unit data relevan terhadap suatu waktu. *Data warehouse* mengandung pecahan kecil data perusahaan (Inmon, 2002, p31).^[3]

2.1.1 Karakteristik Data Warehouse

Menurut Inmon (2005, p29), karakteristik dari *data warehouse* yaitu: *subject oriented, integrated, nonvolatile* dan *time variant*.^[4]

1. *Subject Oriented* (Berorientasi Subjek)

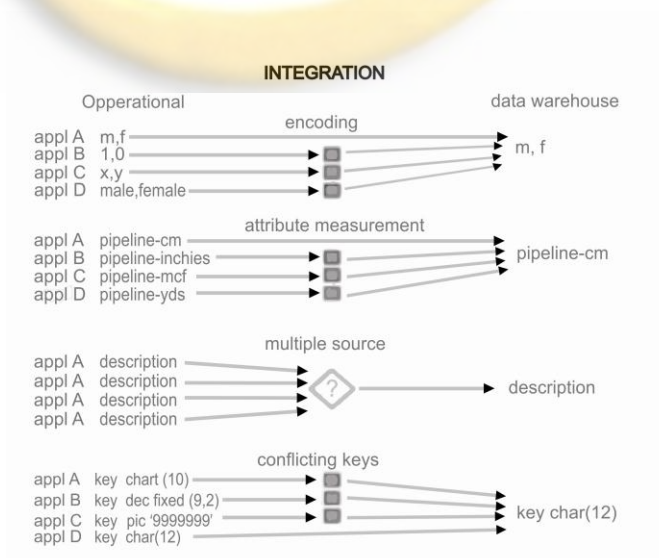
Data warehouse diorganisasikan di sekitar subjek-subjek utama dari perusahaan (seperti: pelanggan, produk, dan penjualan) dari pada area-area aplikasi utama. Hal ini tercermin dari kebutuhan untuk menyimpan data yang mendukung pengambilan keputusan dari pada data yang berorientasi aplikasi.



Gambar 2.1 Data yang Berorientasi Subjek (Inmon, 2002, p32)

2. *Integrated* (Terintegrasi)

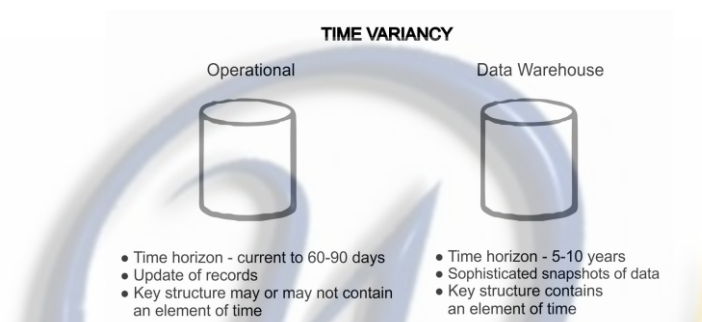
Terintegrasi karena berasal dari sumber data sistem aplikasi perusahaan yang berbeda. Sumber data seringkali tidak konsisten. Sebagai contoh, format tabel-tabel sumber data berbeda. Sumber data yang terintegrasi harus dibuat konsisten untuk menyajikan tampilan yang seragam kepada *user*.



Gambar 2.2 Data yang Terintegrasi (Inmon, 2002, p33)

3. *Time Variant* (Variansi Waktu)

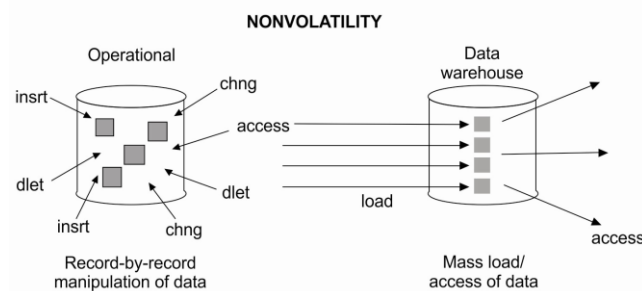
Data di dalam *warehouse* hanya akurat dan valid pada beberapa titik waktu atau pada interval waktu tertentu. Variansi waktu dari *data warehouse* juga ditunjukkan oleh perpanjangan waktu yang dimiliki oleh data, dan asosiasi implisit atau eksplisit waktu dengan semua data, serta fakta bahwa data merepresentasikan serangkaian *snapshot*.



Gambar 2.3 Variansi Waktu (Inmon, 2002, p34)

4. *Nonvolatile* (Tidak Mudah Berubah)

Data tidak *diupdate* secara *real-time* tetapi diperbaharui dari sistem operasional pada basis sehari-hari. Data baru selalu ditambahkan sebagai suplemen untuk sebuah *database*, bukan pengganti. *Database* secara terus menerus menyerap data baru, dan secara bertingkat diintegrasikan dengan data sebelumnya.



Gambar 2.4 Tidak Mudah Berubah (Inmon, 2002, p35)

Hal-hal yang berkaitan dengan *data warehouse* dalam penerapan pada sistem *business intelligence* (Inmon, 2002) adalah:^[3]

1. *Data Mart* yakni merupakan suatu bagian pada *data warehouse* yang mendukung pembuatan laporan dan analisa data pada suatu unit, bagian atau operasi pada suatu perusahaan.
2. *On-Line Analytical Processing* yakni merupakan suatu pemrosesan basis data yang menggunakan tabel fakta dan dimensi untuk dapat menampilkan berbagai macam bentuk laporan, analisis, *query* dari data yang berukuran besar.
3. *On-Line Transaction Processing* yakni merupakan suatu pemrosesan yang menyimpan data mengenai kegiatan operasional transaksi sehari-hari.
4. *Dimension Table* yakni merupakan tabel yang berisikan kategori dengan ringkasan data detail yang dapat dilaporkan. Seperti laporan laba pada tabel fakta dapat dilaporkan sebagai dimensi waktu yang berupa perbulan, perkwartal dan pertahun.
5. *Fact Table* yakni merupakan tabel yang umumnya mengandung angka dan data histori dimana *key* (kunci) yang dihasilkan sangat unik, karena *key* tersebut terdiri dari *foreign key* (kunci asing) yang merupakan *primary key* (kunci utama) dari beberapa dimensi tabel yang berhubungan.

6. *Decision Support System* yakni merupakan sistem yang menyediakan informasi kepada pengguna yang menjelaskan bagaimana sistem ini dapat menganalisa situasi dan mendukung suatu keputusan yang baik.

2.1.2 Keuntungan Data Warehouse

Menurut Conolly dan Begg (2005, p1152), *data warehouse* yang telah diimplementasikan dengan baik dapat memberikan keuntungan yang besar bagi organisasi, yaitu:^[1]

1. Potensi nilai kembali yang besar pada investasi

Sebuah organisasi harus mengeluarkan uang dan sumber daya dalam jumlah yang cukup besar untuk memastikan kalau *data warehouse* telah diimplementasikan dengan baik, biaya yang dikeluarkan tergantung dari solusi teknikal yang diinginkan. Akan tetapi, setelah *data warehouse* digunakan, maka kemungkinan didapatkannya ROI (*Return on Investment*) relatif lebih besar.

2. Keuntungan kompetitif

Keuntungan kompetitif didapatkan apabila pengambil keputusan mengakses data yang dapat mengungkapkan informasi yang sebelumnya tidak diketahui, tidak tersedia, misalnya informasi mengenai konsumen, *trend*, dan permintaan.

3. Meningkatkan produktivitas para pengambil keputusan perusahaan

Data warehouse meningkatkan produktivitas para pengambil keputusan perusahaan dengan menciptakan sebuah *database* yang terintegrasi secara

konsisten, berorientasi pada subjek, dan data historis. *Data warehouse* mengintegrasikan data dari beberapa sistem yang tidak konsisten ke dalam bentuk yang sama sehingga data tersebut dapat menjadi informasi yang berguna untuk analisis dan pengambilan keputusan.

2.1.3 Tugas Data Warehouse

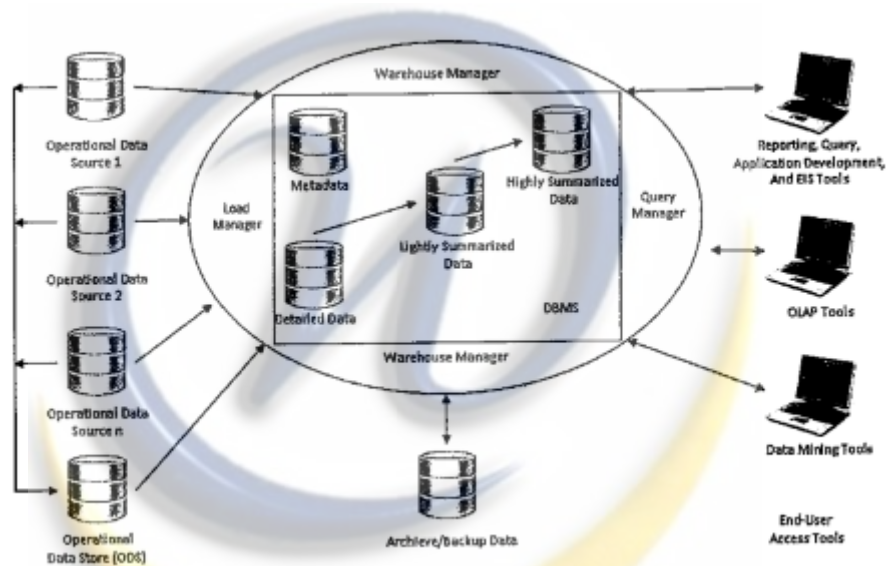
Ada empat tugas yang bisa dilakukan oleh *data warehouse* (Kimball dan Caserta, 2004) yaitu :^[1]

- a. Pembuatan Laporan yakni proses pembuatan laporan merupakan salah satu kegunaan *data warehouse* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan perhari, perbulan, pertahun atau jangka waktu kapan pun yang diinginkan.
- b. OLAP yakni dengan adanya *data warehouse*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah di dapat. OLAP mendayagunakan konsep multidimensional dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah *query*.
- c. *Data Mining* yakni merupakan proses untuk menggali pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data warehouse*, dengan menggunakan kecerdasan buatan (*artificial intelligence*), statistik dan matematika.
- d. Proses Informasi Eksekutif yakni *data warehouse* dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus

menjelajahi keseluruhan data. Dengan menggunakan *data warehouse* segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan.

2.1.4 Arsitektur Data Warehouse

Menurut Conolly dan Begg (2002, p1052) arsitektur *data warehouse* adalah sebagai berikut :^[1]



Gambar 2.5 Arsitektur *Data Warehouse* (Connolly, 2002, p1053)

Komponen-komponen dari *data warehouse* menurut Connolly dan Begg (2005, p1157), antara lain:^[1]

1. *Operational Data*

Operational Data adalah data yang digunakan dalam proses operasional harian.

2. *Operational Data Store*

Operational Data Store (ODS) adalah sebuah tempat penyimpanan data saat ini dan data operasional terintegrasi yang digunakan untuk analisis. ODS

menyediakan kemudahan untuk menggunakan *database* relasional ketika berada jauh dari fungsi *decision support* dari *data warehouse*. Dengan terlebih dahulu membangun ODS, pembuatan *data warehouse* menjadi lebih sederhana karena ODS dapat menyuplai data yang telah diekstrak dari sistem sumber.

3. *Load manager*

Load manager (disebut juga komponen *front-end* melakukan semua operasi yang berhubungan dengan ekstraksi dan *loading* data ke dalam *data warehouse*.

4. *Warehouse Manager*

Warehouse manager melakukan semua operasi yang berhubungan dengan *management data* dalam *data warehouse*. Operasi-operasi yang dilakukan oleh *warehouse manager*, meliputi:

- a. Analisis data untuk menjaga konsistensi;
- b. Transformasi dan penggabungan sumber data dari tempat penyimpanan sementara ke dalam tabel-tabel *data warehouse*;
- c. Pembuatan *index* dan *view* dalam tabel basis;
- d. Melakukan denormalisasi (jika diperlukan);
- e. Melakukan agregasi (jika diperlukan);
- f. *Backup* dan penyimpanan data.

5. *Query Manager*

Query manager (disebut juga komponen *back-end*) melakukan semua operasi yang berhubungan dengan *management* dari permintaan pengguna (*user queries*). Komponen ini biasanya dikonstruksikan menggunakan *vendor end-user*

data access tools, *data warehouse monitoring tools*, fasilitas *database*, dan program *custombuilt*. Operasi yang dilakukan pada komponen ini meliputi memproses *query* ke tabel yang tepat dan penjadwalan eksekusi *query*.

6. *Detailed Data*

Dalam *data warehouse*, area ini menyimpan *detailed data* dalam skema *database*. Umumnya, *detailed data* tidak disimpan secara *online* tetapi dapat dibuat ada dengan mengagregasi data ke tingkatan yang lebih detail.

7. *Lightly and Highly Summarized Data*

Dalam *data warehouse*, area ini menyimpan *predefined lightly* dan *highly summarized data* yang dihasilkan oleh *warehouse manager*. Tujuan dari meringkas informasi adalah untuk mempercepat pemrosesan *query*. Ringkasan data diupdate secara terus menerus ketika data baru dimasukkan ke dalam *warehouse*.

8. *Archived/Backup Data*

Dalam *data warehouse*, area ini menyimpan semua *data detailed* dan *summarized* dengan tujuan untuk penyimpanan (*archiving*) dan *backup*. Data ditransfer ke dalam *magnetic tape* atau *optical disc*.

9. *Metadata*

Dalam *data warehouse*, area ini menyimpan semua definisi *metadata* (data tentang data) yang digunakan oleh semua proses dalam *data warehouse*.

Metadata digunakan untuk berbagai tujuan termasuk :

a. Proses ekstraksi dan *loading*

Metadata digunakan untuk memetakan sumber data ke dalam *common view* dari data pada *warehouse*.

b. Proses manajemen *warehouse*

Metadata digunakan untuk otomatisasi produksi tabel ringkasan.

c. Sebagai bagian dari proses manajemen *query*

Metadata digunakan untuk mengarahkan *query* ke sumber data yang paling tepat.

10. *End-User Access Tools*

Tujuan utama dari *data warehousing* adalah menyediakan informasi kepada *business user* untuk mendukung pengambilan keputusan. Para pengguna ini berinteraksi dengan *data warehouse* menggunakan *end-user access tools*. *End-user access tools* dibagi menjadi 5 (lima) kategori utama:

a. *Reporting and Query Tools*

Reporting tools meliputi *production reporting tools* dan *report writers*. *Production reporting tools* digunakan untuk menghasilkan laporan operasional reguler. *Report writers* adalah *desktop tools* yang dirancang untuk *enduser*.

Query tools untuk *data warehouse* relasional dirancang untuk menerima SQL atau menghasilkan *statement SQL* untuk memproses data yang disimpan di dalam *warehouse*.

b. *Application Development Tools*

Application Development Tools adalah aplikasi *inhouse* yang menggunakan *data access tools* grafikal yang dirancang untuk lingkungan *client – server*. Beberapa dari *application development tools* ini terintegrasi dengan *OLAP tools*, dan dapat mengakses sistem *database* pada umumnya, termasuk *Oracle*, *Sybase*, dan *Informix*.

c. *Executive Information System (EIS) Tools*

Executive Information System lebih dikenal dengan “*everybody’s information systems*” pada awalnya dikembangkan untuk mendukung pengambilan keputusan strategi tingkat tinggi yang kemudian fokusnya meluas untuk mendukung semua tingkatan manajemen. *EIS tools* terhubung dengan *mainframe* yang memungkinkan *user* membuat aplikasi grafikal pendukung pengambilan keputusan untuk menyediakan tinjauan data perusahaan dan akses ke sumber data eksternal.

d. *Online Analytical Processing (OLAP) Tools*

Online Analytical Processing (OLAP) Tools didasarkan pada konsep *database* multidimensi dan memungkinkan pengguna untuk menganalisa data menggunakan *view* multidimensi yang kompleks. *Tools* ini mengasumsikan bahwa data diorganisasikan berdasarkan model multidimensi yang didukung oleh *database* multidimensi atau *database* relasional yang dirancang untuk memungkinkan *query* multidimensi.

e. *Data mining Tool*

Data mining adalah proses untuk menemukan korelasi, pola, dan *trend* baru yang berarti dengan menggali data dalam jumlah besar dengan menggunakan teknik statistika, matematis, dan kecerdasan buatan. *Data mining* memiliki kemampuan untuk menggantikan kemampuan dari *OLAP tools*. *Data mining* memiliki kemampuan untuk membangun model prediktif dari pada model retrospektif.

2.1.5 Perancangan Data Warehouse

Salah satu dari metodologi perancang *data warehouse* adalah *Nine-Step Methodology* oleh Kimball (1996) yang memiliki 9 (sembilan) langkah sebagai berikut:^[5]

1. Memilih Proses (*Choosing The Process*)

Proses (fungsi) mengacu pada subjek masalah dari *data mart* tertentu. *Data mart* yang pertama kali dibangun haruslah tepat waktu, sesuai dengan anggaran, dan dapat menjawab pertanyaan pertanyaan bisnis yang penting.

2. Memilih Sumber (*Choosing The Grain*)

Memilih *grain* berarti memutuskan secara pasti apa yang dinyatakan oleh *record* dari tabel fakta. Hanya dengan telah terpilihnya *grain* untuk tabel fakta maka kita dapat mengidentifikasi dimensi. Keputusan *grain* untuk tabel fakta juga menentukan *grain* untuk setiap dimensi pada tabel fakta.

3. Mengidentifikasi dan Penyesuaian Dimensi (*Identifying and Conforming The Dimensions*)

Dimensi menentukan konteks untuk memberikan pertanyaan mengenai fakta-fakta dalam tabel fakta. Kumpulan dimensi yang dibangun dengan baik membuat *data mart* dapat dimengerti dan mudah untuk digunakan. Dimensi diidentifikasi dalam detail yang cukup untuk mendeskripsikan data seperti *client* dan *properties* dari *grain* yang tepat.

4. Memilih Fakta (*Choosing The Fact*)

Grain dari tabel fakta menentukan fakta mana yang dapat digunakan pada *data mart*. Semua fakta harus dinyatakan berdasarkan tingkatan yang tersirat oleh *grain*. Fakta tambahan dapat ditambahkan ke dalam tabel fakta pada setiap waktu dengan catatan fakta tersebut konsisten dengan *grain* dari tabel.

5. Menyimpan Perhitungan Awal dalam Tabel Fakta (*Storing Pre-Calculation in The Fact Table*)

Setelah fakta dipilih, setiap fakta harus dikaji ulang untuk menentukan apakah ada kemungkinan untuk melakukan *pre-calculations*. *Pre-calculations* terjadi ketika fakta terdiri dari *statement* untung dan rugi.

6. Melihat Kembali Table Dimensi (*Rounding Out The Dimension Tables*)

Pada langkah ini, kita kembali mengkaji tabel dimensi dan menambahkan sebanyak mungkin deskripsi teks ke dimensi. Teks deskripsi haruslah seintuitif mungkin dan dapat dimengerti oleh pengguna. Kegunaan dari *data mart* ditentukan oleh cakupan dan sifat atribut pada tabel dimensi.

7. Memilih Durasi *Database* (*Choosing The Duration of Database*)

Durasi mengukur berapa lama tabel fakta dapat disimpan. Pada banyak perusahaan, ada ketentuan untuk melihat pada periode waktu yang sama satu atau dua tahun sebelumnya. Tabel fakta yang sangat besar akan mengakibatkan setidaknya dua masalah yang signifikan pada *data warehouse*. Pertama, bertambahnya kesulitan untuk menjadikan data lama yang semakin bertambah sebagai sumber. Semakin lama suatu data, semakin banyak masalah dalam

membaca dan menginterpretasikan *file* lama tersebut. Kedua, kebutuhan dimensi menggunakan versi yang lama, bukan versi yang baru.

8. Menelusuri Perubahan dari Dimensi secara Perlahan (*Tracking Slowly Changing Dimension*)

Pada tahap ini, *data warehouse* memperhatikan proses dimensi yang semakin tua seiring dengan berjalannya waktu. Untuk itu perlu dilakukan *update* agar *data warehouse* selalu konsisten. Terdapat tiga tipe dasar perubahan dimensi secara perlahan:

- a. Tipe 1: Perubahan data secara langsung atau *update* table dimensi.
- b. Tipe 2: Perubahan data membentuk *record* baru dengan *surrogate key* yang berbeda.
- c. Tipe 3: Perubahan data akan membentuk atribut atau kolom baru pada tabel dimensi.

9. Memutuskan Prioritas *Query* dan Tipe *Query* (*Deciding The Query Priorities and The Query Models*)

Pada tahap ini dipertimbangkan masalah perancangan fisik (*physical design*). Masalah utama pada perancangan fisik yang mempengaruhi persepsi pengguna akhir dari *data mart* adalah urutan penyusunan tabel fakta pada *disk* dan adanya *pre-stored summaries* dan agregasi.

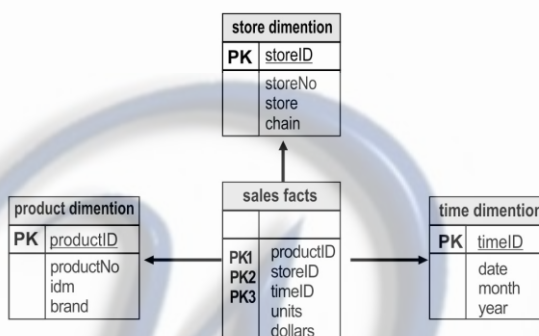
2.1.6 Pemodelan Dimensional

Menurut Connolly dan Begg (2002, p1079), pemodelan dimensional adalah sebuah teknik perancangan logical yang bertujuan untuk mempresentasikan data ke dalam sebuah standar, bentuk intuitif yang dapat diakses dengan performa yang tinggi.^[1] Setiap model dimensional terdiri dari sebuah tabel dengan sebuah *primary key* komposit yang disebut dengan tabel fakta, dan sekumpulan tabel yang lebih kecil yang disebut dengan tabel dimensi. Setiap tabel dimensi memiliki sebuah *primary key* (nonkomposit) sederhana yang berkorespondensi tepat dengan satu key komposit pada tabel fakta. Dengan kata lain, *primary key* dari tabel fakta terbuat dari dua atau lebih *foreign key*. Karakteristik dengan struktur yang seperti bintang ini disebut dengan *star schema* atau *star join*.^[1]

1. Star Schema

Star schema adalah sebuah model data dimensional yang memiliki sebuah tabel fakta di pusatnya, dikelilingi oleh tabel-tabel dimensi yang berisi referensi data yang biasanya dapat dinormalisasi. Skema bintang mengeksploitasi karakteristik dari data faktual, seperti fakta dihasilkan dari *event-event* yang terjadi di masa lampau, dan tidak akan berubah terlepas dari bagaimana tabel fakta dianalisis. Fakta yang paling berguna dalam tabel fakta adalah perhitungan *numeric*, atau “fakta” yang terjadi pada setiap *record*. Sedangkan tabel dimensi mengandung informasi tekstual yang deskriptif. Atribut pada tabel dimensi digunakan sebagai batasan-batasan dalam *query data warehouse*.^[1]

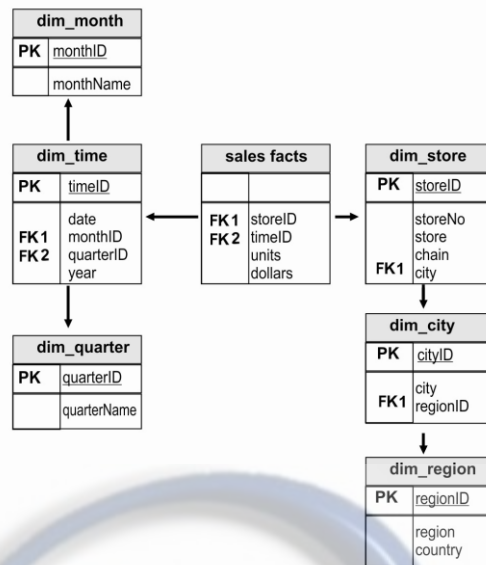
Star schema dapat digunakan untuk mempercepat kinerja *query* dengan mendenormalisasikan data referensi ke dalam sebuah tabel dimensi. Denormalisasi cocok dilakukan ketika ada beberapa entitas yang berhubungan dengan tabel dimensi sering diakses. Dengan denormalisasi maka penggabungan (*join*) tabel-tabel tambahan untuk mengakses atribut tersebut dapat dihindari. [1]



Gambar 2.6 Contoh Star Schema

2. Snowflake Schema

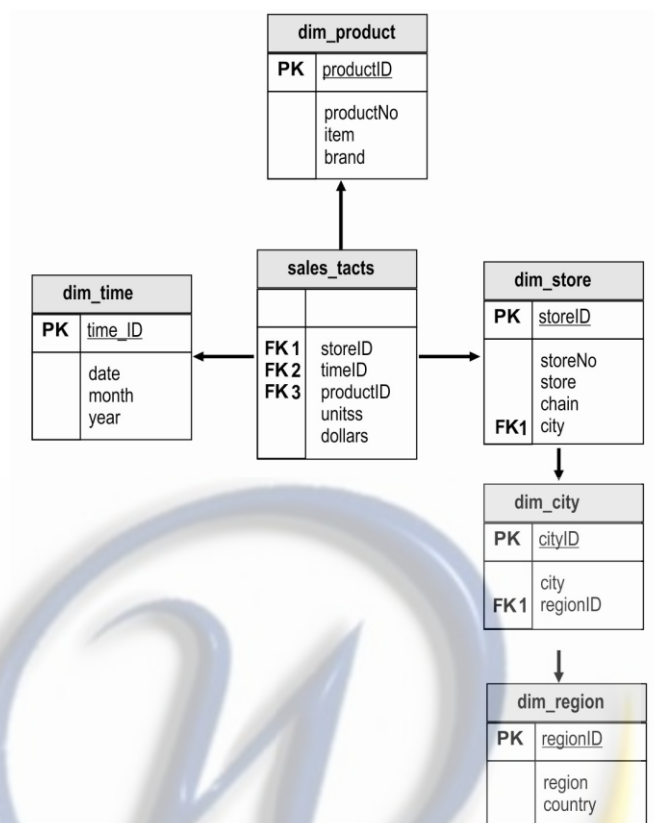
Snowflake schema adalah model data dimensional yang memiliki sebuah tabel fakta sebagai pusatnya, dikelilingi tabel-tabel dimensi yang ternormalisasi. *Snowflake schema* adalah sebuah variasi dari *star schema* dimana table dimensinya boleh memiliki dimensi. Penggunaan table dimensi pada *snowflake schema* sangatlah mendasar, sedangkan pada *star schema* tidak. *Snowflake schema* dibuat berdasarkan OLTP sehingga semua data akan termuat detail dalam setiap tabel fakta dan tabel dimensi. [1]



Gambar 2.7 Contoh Snowflake Schema

3. Starflake Schema

Starflake schema adalah model data dimensional yang memiliki sebuah tabel fakta sebagai pusatnya, dikelilingi oleh tabel-tabel dimensi yang ternormalisasi dan terdenormalisasi. Beberapa pemodelan dimensional menggunakan campuran dari *star schema* yang terdenormalisasi dan *snowflake schema* yang ternormalisasi. Kombinasi dari *star schema* dan *snowflake schema* disebut *starflake schema*.^[1]



Gambar 2.8 Contoh Starflake Schema

2.2 Business Intelligence

Menurut Niu (2009), *business intelligence* adalah proses mengekstrak, transformasi, mengelola, dan menganalisis data bisnis untuk mendukung pengambilan keputusan. Dalam proses ini pada umumnya melibatkan data set dalam jumlah besar yang tersimpan dalam *data warehouse*. Proses *business intelligence* meliputi lima tahapan :^[6]

1. Pengumpulan data.

Sistem *business intelligence* dapat mengekstrak data dari beberapa sumber data yang berasal dari berbagai unit bisnis seperti pemasaran, produksi, sumber daya manusia, dan keuangan. Data yang sudah diekstrak harus dibersihkan, transformasi, dan terintegrasi untuk dapat dianalisis.

2. Analisis data.

Pada tahapan ini, data dikonversi menjadi informasi atau pengetahuan melalui berbagai macam teknik analisis seperti laporan, visualisasi, dan *data mining*. Hasil dari proses analisis dapat membantu pihak manajemen untuk memahami situasi dan mengambil keputusan yang lebih baik.

3. Kesadaran situasi.

Kesadaran terhadap situasi dapat memberikan pemahaman yang lebih mendalam terhadap keadaan keputusan saat ini berdasarkan hasil analisis data.

4. Penilaian resiko.

Kesadaran terhadap situasi yang cukup bervariasi dapat membantu manajer untuk memprediksi masa depan, identifikasi ancaman dan peluang, dan merespon sesuai dengan kebutuhan. Saat ini bisnis beroperasi dalam kondisi lingkungan yang kompleks. Pengambilan keputusan bisnis lebih mungkin disertai resiko yang berasal dari lingkungan eksternal dan internal. Sehingga dapat disimpulkan bahwa penilaian resiko merupakan fungsi penting pada sistem *business intelligence*.

5. Dukungan pengambilan keputusan.

Tujuan utama dari *business intelligence* adalah membantu manajer mengambil keputusan dengan bijaksana berdasarkan data bisnis saat ini.

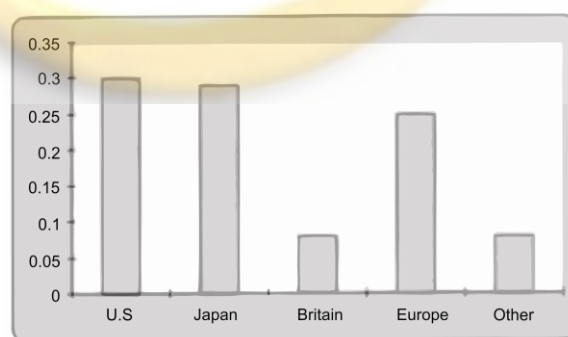
2.3 Visualisasi Data

a. Tabel

Tabel adalah kumpulan angka-angka yang disajikan dalam baris dan kolom menurut kategori-kategori tertentu sehingga dapat memudahkan dalam pembuatan analisis data. Tabel digunakan untuk menampilkan angka, tingkatan, proporsi, dan persentase kumulatif. Penyajian dengan tabel bisa memberikan angka-angka yang lebih teliti baik berupa hubungan satu arah, dua arah, ataupun lebih. ^[1]

b. *Bar Chart*

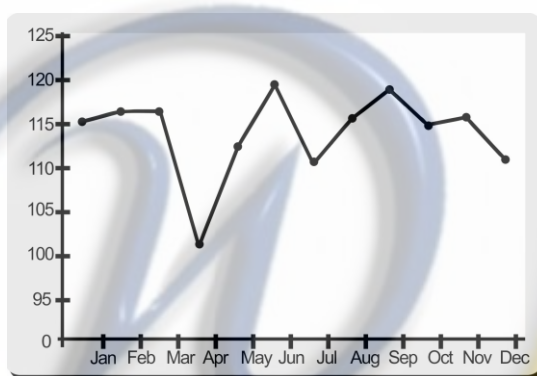
Bar chart sering digunakan untuk menunjukkan data berkategori dimana ketika tidak ada penekanan pada persentase dari total yang direpresentasikan oleh setiap kategori. *Bar chart* dapat diplot secara vertikal maupun horizontal. Skala pengukuran adalah nominal atau ordinal. *Bar chart* dapat digunakan untuk menampilkan data kontinu seperti ukuran sepatu atau warna mata dan data diskontinu seperti tinggi badan atau berat badan. ^[1]



Gambar 2.9 Contoh *Bar Chart* (Azcel, 2002, p59)

c. *Line Chart*

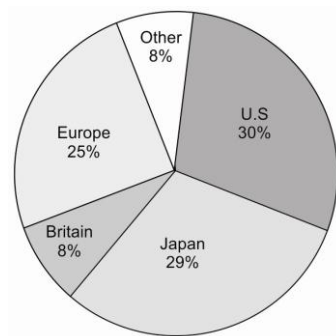
Line chart adalah sebuah tipe grafik yang digunakan untuk menampilkan informasi dalam rangkaian titik data yang dihubungkan dengan segmen garis lurus. *Line chart* sering digunakan untuk memvisualisasikan *trend* data dalam interval waktu atau dalam kurun waktu tertentu, dengan demikian *line chart* sering digambarkan secara kronologis. ^[1]



Gambar 2.10 Contoh *Line Chart* (Azcel, 2002, p60)

d. *Pie Chart*

Pie chart adalah sebuah tampilan deskriptif sederhana dari data yang merupakan jumlah dari total yang diberikan. *Pie chart* mungkin adalah sebuah cara paling ilustratif untuk menampilkan kuantitas sebagai persentase dari total yang diberikan. Total area dari sebuah *pie chart* merepresentasikan 100% dari kuantitas (jumlah dari nilai variabel pada seluruh kategori), dan ukuran dari setiap potongan adalah persentase dari total yang direpresentasikan oleh kategori yang ditunjukkan potongan. *Pie chart* biasanya digunakan untuk mempresentasikan frekuensi atau data berkategori. Skala pengukuran dapat berupa nominal atau ordinal. ^[1]



Gambar 2.11 Contoh *Pie Chart* (Azcel, 2002, p59)

2.4 Istilah Pendidikan

Menurut Peraturan Pemerintah NOMOR 60 TAHUN 1999: Pasal 6^[7]

- (1) Satuan pendidikan yang menyelenggarakan pendidikan tinggi disebut perguruan tinggi, yang dapat berbentuk akademi, politeknik, sekolah tinggi, institut atau universitas.
- (2) Akademi menyelenggarakan program pendidikan profesional dalam satu cabang atau sebagian cabang ilmu pengetahuan, teknologi, dan/atau kesenian tertentu.
- (3) Politeknik menyelenggarakan program pendidikan profesional dalam sejumlah bidang pengetahuan khusus.
- (4) Sekolah Tinggi menyelenggarakan program pendidikan akademik dan/atau profesional dalam lingkup satu disiplin ilmu tertentu.
- (5) Institut menyelenggarakan program pendidikan akademik dan/atau profesional dalam sekelompok disiplin ilmu pengetahuan, teknologi dan/atau kesenian yang sejenis.

(6) Universitas menyelenggarakan program pendidikan akademik dan/atau profesional dalam sejumlah disiplin ilmu pengetahuan, teknologi dan/atau kesenian tertentu.

