

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

Dimulai tanggal 15-01-1990 dengan membuka toko kecil yang menjual bahan kain interior dengan nama Bandung Indah Gemilang. Dikembangkan dengan membuat sebuah home industry yang memfokuskan diri pada bidang garment sprei. Perusahaan Textile lebih memproduksi bahan kain seperti sprei dan bedcover.

2.1.1 Sejarah Perusahaan

Pada tanggal 12 November 1999 didirikan sebuah perusahaan dengan nama Perusahaan Textile yang bertempat di Jl. Carinin Raya No. 439 N Bandung Jawa Barat 40224. Perusahaan Textile menempati gedung baru di Caringin 439 N, sehingga seluruh kegiatan usaha telah terintegrasi dan memulai untuk memproduksi sendiri. Pada bulan Agustus 2005 gudang kain pindah ke gedung baru di Caringin 263, menyusul cutting, quilting, sewing, finishing dan lain-lain.

2.1.2 Logo Perusahaan

Berikut ini merupakan logo perusahaan dari Perusahaan Textile dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Logo Perusahaan

2.1.3 Visi dan Misi Perusahaan

Pada sub-bab ini akan menjelaskan tentang Visi, Misi dan Tujuan dari Perusahaan Textile.

2.1.3.1 Visi

Menjadi perusahaan industry *Bed Sheets Set* terbear dan terbaik di Indonesia serta berorientasi *Export* dengan mengutamakan mutu, model, desain, komunikasi dan kepuasan pelanggan.

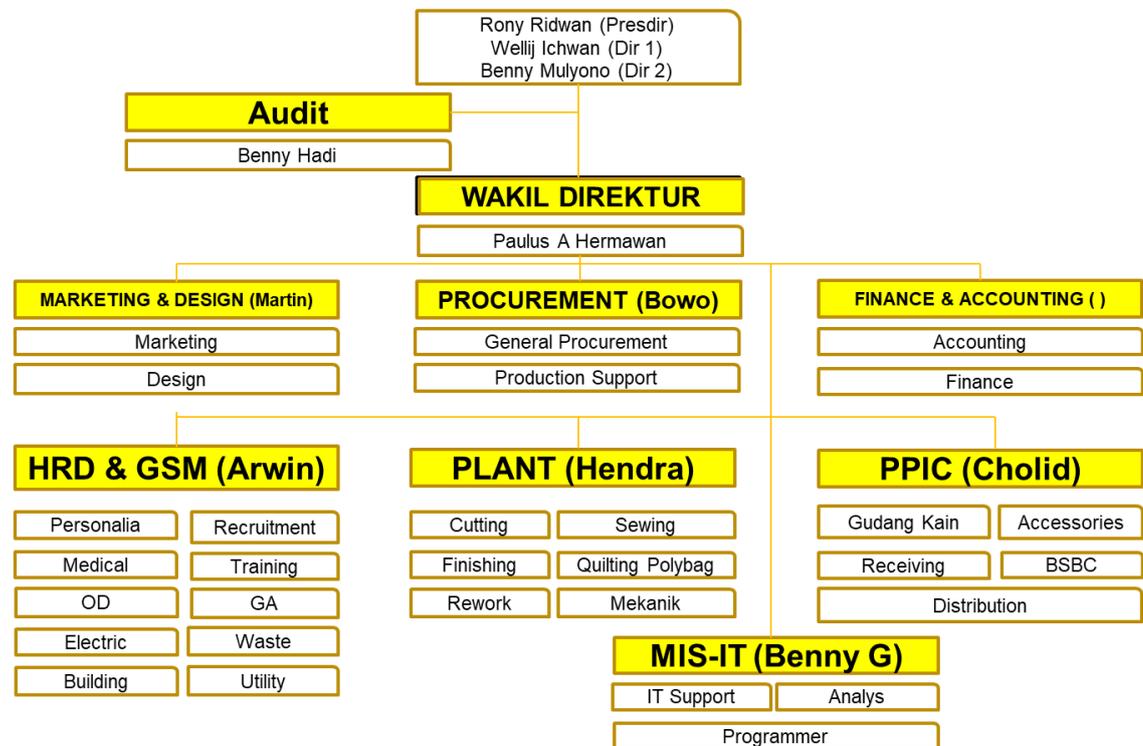
2.1.3.2 Misi

Misi dari Perusahaan Textile adalah sebagai berikut :

1. Menjadi perusahaan Industri *bed sheets set* terbesar dan terbaik di Indonesia.
2. Meningkatkan (menambah baik) mutu, model dan desain *bed sheets set*.
3. Kemampuan memperoleh (meningkatkan) kesejahteraan bagi yang mendukung bisnis ini secara professional.

2.1.4 Struktur Organisasi

Struktur organisasi adalah pola hubungan antara bagian-bagian dari instansi atau menggambarkan dengan jelas antar bagian dalam suatu instansi. Berikut struktur organisasi perusahaan textile yang diteliti dapat dilihat pada Gambar 2. 2



Gambar 2.2 Struktur Organisasi Perusahaan Textile

2.1.5 Job Desk

Dalam lingkungan perusahaan Perusahaan Textile memiliki sembilan divisi, dikarenakan yang digunakan hanya dua divisi maka yang akan di deskripsikan hanya dua divisi yaitu divisi penjualan dan divisi distribusi. Berikut deskripsi dari divisi penjualan dan divisi distribusi:

1. Divisi penjualan : Melakukan pendataan barang-barang apa saja yang akan dijual oleh perusahaan kepada toko. Melakukan pembuatan laporan penjualan mengenai barang barang yang telah dijual. Melakukan pendataan barang yang akan dijual berdasarkan tipe penjualan barang.

2. Divisi Distribusi : Mengirim barang ke toko-toko berdasarkan permintaan dari toko. Melakukan pengecekan mengenai barang dan jumlah barang yang akan didistribusikan.

2.2 Landasan Teori

Pada Sub bab ini berisi tentang teori-teori pendukung yang digunakan dalam proses analisis dan implementasi dalam tugas akhir ini.

2.2.1 Pengertian Data

Menurut Jogianto, data merupakan suatu fakta atau kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Data adalah bentuk satuan dari datum, berasal dari bahasa Latin yang berarti “sesuatu yang diberikan”. Pada penggunaan sehari-hari data merupakan suatu pernyataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra[2].

Menurut Immon, data merupakan suatu dokumentasi dari fakta-fakta, konsep-konsep atau intruksi-intruksi pada media penyimpanan (*storage*) untuk komunikasi, perolehan dan pemrosesan dengan cara otomatis dan presentasi sebagai informasi yang dapat dimengerti oleh manusia [1].

Menurut Laudon & Laudon, data adalah fakta-fakta yang mewakili kejadian-kejadian yang berlangsung dalam organisasi atau lingkungan fisik sebelum didata dan diatur kedalam bentuk yang dapat dipahami dan digunakan orang [2].

Dari dua teori tersebut dapat disimpulkan bahwa data merupakan kumpulan fakta yang secara umum tidak dapat digunakan karena belum diolah yang terdapat pada media penyimpanan untuk melalui pemrosesan sehingga dapat menyajikan informasi yang dapat dimengerti.

A. Jenis Data Menurut Cara Memperolehnya

Jenis data dapat diuraikan berdasarkan cara memperolehnya, yaitu :

1. Data Primer

Data Primer adalah data yang diambil secara langsung dari objek penelitian oleh seorang peneliti perorang maupun organisasi [5].

Contoh : Melakukan wawancara secara langsung penonton bioskop 21

untuk meneliti preferensi konsumen bioskop.

2. Data Sekunder

Data sekunder merupakan data yang diduplikasinya tidak secara langsung dari objek penelitian. Peneliti akan mendapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial [5].

3. Macam-Macam Data Berdasarkan Sumber Data

Macam-macam data dapat dibedakan berdasarkan sumber data yang diperoleh yaitu :

1. Data Internal

Data *Internal* adalah data yang menggambarkan situasi dan kondisi pada suatu organisasi secara internal [5].

Contoh : Data yang sensitive seperti data keuangan, data pegawai, data penjualan dari suatu perusahaan atau organisasi.

2. Data Eksternal

Data *eksternal* adalah data yang menggambarkan situasi serta kondisi yang ada di luar organisasi [5].

Contohnya : Data jumlah penggunaan suatu produk pada konsumen tingkat preferensi pelanggan, persebaran penduduk dan lain sebagainya.

2.2.1 Pengertian Informasi

Informasi merupakan sekumpulan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan saat ini atau saat mendatang [2]. Dapat disimpulkan bahwa :

1. Informasi bermula pada data.
2. Dapat memberikan suatu nilai tambah atau pengetahuan bagi yang menggunakannya.
3. Dapat digunakan untuk pengambilan keputusan.

2.2.3 Konsep Basis Data

Basis data dapat diartikan sebagai sekumpulan kelompok data (arsip) yang saling keterkaitan yang diorganisasi sedemikian rupa agar kelak dapat

dimanfaatkan kembali dengan cepat dan mudah. Atau dapat juga didefinisikan sebagai kumpulan data yang saling berhubungan yang disimpan secara bersamaan sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan [3data].

“ Menurut Gordon C. Everest, Database atau basis data merupakan sebuah koleksi atau kumpulan dari data yang bersifat mekanis, terbagi, terdefinisi secara formal dan terkontrol. Pemantauan dari sistem database tersebut adalah terpusat, yang biasanya dimiliki dan juga dipegang oleh suatu organisasi. “

“ Menurut Rogayah, yang terdapat di dalam modulnya menjelaskan mengenai sistem basis data, ia menyatakan bahwa sistem basis data adalah sistem yang dapat merangkai, menyusun dan mengorganisir record yang ada dengan menggunakan computer sebagai medianya sebagai media menyimpan atau mendokumentasikan serta memelihara data operasional yang lengkap dari sebuah organisasi atau perusahaan sehingga mampu menyajikan informasi yang optimal yang diperlukan pengguna untuk proses mengambil kebijakan. “

2.2.3.1 Pengertian Basis Data

Basis Data (database) adalah kumpulan data-data yang umumnya menjabarkan aktivitas-aktivitas dari suatu atau lebih dari satu organisasi yang terkait [2]. Basis Data adalah sebuah cara untuk mendokumentasikan berbagai macam data yang kemudian di manajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan. Dalam basis data, data yang ada tidak hanya disimpan begitu saja melainkan akan dikelola dengan sistem pengaturan basis data yang sering disebut *Database Managemenet System* (DBMS). Dengan adanya DBMS maka pengaksesan data dapat dilakukan dengan mudah dan cepat oleh pengguna.

Keuntungan basis data :

1. Mereduksi redudansi
2. Data dapat di-*share* antar aplikasi.
3. Standarisasi dapat dilakukan.
4. Batasan *security* dapat diterapkan.
5. Mengelola integritas data.

6. Independensi data.

2.2.3.2 Bahasa dalam Basis Data

Ada Beberapa Bahasa yang digunakan dalam basis data diantaranya adalah sebagai berikut :

1. *Data Definiton Language* (DDL)

Data Definition Language (DDL) ini adalah sekumpulan perintah-perintah yang terdapat dalam SQL yang berfungsi untuk menspesifikasikan atau menggambarkan suatu skema basis data. Dengan Bahasa ini hal-hal yang dapat dilakukan adalah membuat table baru, membuat indeks, mengubah struktur table, menentukan struktur penyimpanan table, dan lain-lain. Hal yang dasar yang dapat dilakukan DDL ini adalah menciptakan, mengubah, dan menghapus basis data.

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML) adalah sekumpulan perintah-perintah dapat dikatakan seperti Bahasa pemrograman atau sintak yang dapat digunakan untuk mengubah, memanipulasi, dan mengambil data pada basis data. Dapat melakukan aksi crud seperti menghapus, mengubah, menyisipkan dan mengambil data. DML dibagi menjadi 2 jenis yaitu :

A. Prosedural

Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkan dari data tersebut.

B. Non-Prosedural

Non-prosedural menuntut pengguna menentukan terlebih dahulu data apa saja yang diperlukan dan yang dibutuhkan, tetapi tidak perlu mendeskripsikan atau menyebutkan bagaimana cara data tersebut mendapatkannya.

1. *Transaction Control*

Transaction Control adalah bahasa pada basis data yang dapat mengatur transaksi dimana transaksi dilakukan oleh *Data Manipulation Language* (DML). *Transaction Control* ini memiliki peran yang sangat besar untuk menentukan suatu aktivitas dilakukan atau tidaknya perubahan-

perubahan data yang terdapat dalam basis data. Contoh dari *transaction control* adalah perintah *commit* dan *rollback*.

2.2.3.3 DBMS (Database Management System)

DBMS (*Database Management System*) atau disebut dengan sistem manajemen basis data, berfungsi sebagai suatu sistem dalam pengolahan basis data sehingga menjadikan sebuah data menjadi sebuah informasi baru yang berguna. Sistem ini memungkinkan untuk menyusun, mengolah dan memperbaharui data yang terdapat dalam suatu basis data. DBMS memiliki kemampuan untuk mengolah data dalam jumlah yang besar dan memanipulasi data dengan cepat dan mudah. Tujuan utama dari DBMS adalah menyediakan cara untuk menyimpan dan mengambil informasi dari *database* dengan baik, nyaman, dan efisien [4].

2.2.3.4 RDBMS (Relational Database Management System)

RDBMS (*Relational Database Management System*) merupakan sekumpulan data yang saling berhubungan yang disimpan sedemikian rupa sehingga mudah diambil informasinya oleh pengguna.

Ada Tiga prinsip dalam RDBMS.

1. *Data Definition*

Mendefinisikan data-data yang akan dibuat, seperti nama table dan *field* serta keterangan dari *field* tersebut.

2. *Data Manipulation*

Melakukan manipulasi data yang sudah dibuat, seperti mengubah na a *field* dan keterangannya, dan meghapus *record*.

3. *Data Control*

Pada bagian ini berkenaan terhadap pengendalian data kepada siapa saja yang bisa melihat isi data.

2.2.3.5 Database Administrator

Database Administrator adalah orang memiliki kemampuan dalam melakukan *control* utama terhadap keseluruhan sistem basis data (mencakup data dan program) yang mempunyai fungsi seabagai berikut :

1. Pendefinisian skema.
2. Pendefinisian struktur penyimpanan dan metode akses.

3. Modifikasi skema dan organisasi fisik.
4. Pemberian otorisasi bagi pengaksesan data
5. Mendefinisikan bagian mana basis data yang mana saja yang dapat diakses oleh seseorang.
6. Pengguna, termasuk operasi-operasi yang dapat dilakukan.
7. Spesifikasi batasan integrasi.

2.2.4 Data Warehouse

Data Warehouse adalah sekumpulan informasi *logic* yang dikumpulkan dari berbagai operasional *database* yang berfungsi untuk mendukung aktivitas para analis dalam melakukan pengambilan suatu keputusan. Penyajian *data warehouse* sendiri memiliki sifat multidimensional data, dimana *data warehouse* terdiri dari beberapa layer, kolom dan baris. Layer yang terdapat pada *data warehouse* menunjukkan suatu informasi yang ditampilkan dan disampaikan pada dimensi yang berbeda.

Data warehouse juga dapat dikatakan sebagai wadah penyimpanan data histori yang berasal dari berbagai departemen, organisasi atau perusahaan yang bertujuan untuk mengolah data, dimana data yang diolah dapat menghasilkan sebuah informasi yang efektif dan efisien dimana data tersebut dapat digunakan dalam pengambilan suatu keputusan. *Data warehouse* merupakan sekumpulan dari *database* yang berbeda, *data warehouse* mengkombinasikan informasi dengan cara merangkum dan mengelompokan. *Data warehouse* merupakan metode dalam melakukan perancangan basis data yang dapat menunjang DSS (*Decision Support System*) dan EIS (*Executive Information System*).

Data warehouse juga bisa dikatakan sebagai sekumpulan data yang berorientasi subje, terintegrasi tidak dapat di *update*, memiliki dimensi waktu yang dapat digunakan untuk pengambilan suatu keputusan yang dilakukan oleh pihak pimpinan perusahaan. Berdasarkan definisi tersebut, maka *data warehouse* memiliki karakteristik [6]. Sebagai berikut :

1. *Subject Oriented* (berorientasi Subjek)

Data warehouse memiliki entitas tertinggi dalam suatu perusahaan, contoh subjek atau entitas tertinggi di dalam suatu perusahaan manufaktur yaitu di

penjualan, marketing, dan lain-lain. *Data warehouse* pun dirancang untuk membantu pengguna dalam mengambil sebuah keputusan. Contohnya untuk mengetahui tentang data penjualan pada perusahaan. Peneliti dapat membangun sebuah data *warehouse* yang berfokus pada penjualan.

2. *Integrity* (integrasi)

Data warehouse di desain untuk mengintegrasikan sejumlah sumber data yang berbeda-beda. Data yang telah terintegrasi dapat data lebih konsisten. Sehingga lebih mudah dipahami ketika akan membuat laporan dan laporan tersebut akan dijadikan acuan sebagai pengambilan suatu keputusan.

3. *Time-variant* (variasi waktu)

Data warehouse harus bisa menyimpan histori dari suatu data atau laporan dalam jangka waktu panjang. *Data warehouse* bisa berfokus pada perubahan setiap waktunya.

4. *Non-volatile*

Data yang ada dalam data *warehouse* tidak bisa di ubah. Melainkan datanya tetap dan tidak berubah-ubah.

Data warehouse dibuat untuk melayani pengguna (*analyst* dan pengambil keputusan). Sehingga *data warehouse* wajib dirancang sesuai dengan persyaratan [7] berikut :

1. Harus bisa memberikan kepuasan kepada setiap pengguna.
2. Memiliki *function* (fungsi) sendiri tanpa menggunakan OLTP *system*.
3. Mampu menyediakan area atau tempat penyimpanan data yang konsisten.
4. Menjawab setiap *complex queries* dengan cepat.
5. Menyediakan berbagai macam analisis *tools* yang kuat, seperti OLAP dan *data mining*.

Sebagai besar *data warehouse* yang sukses selain memenuhi persyaratan di atas juga memiliki beberapa karakteristik [7] berikut:

1. Berdasarkan model dimensional.
2. Mengandung *historical data*.
3. terdiri dari *detailed* dan *summarized data*.

4. Tetap mempertahankan konsistensi data walaupun berasal dari sumber data yang berbeda.
5. Fokus dalam *single subject*, seperti penjualan, keuangan, atau inventaris.

Menurut W.H. Inmon, *data warehouse* adalah sekumpulan data yang dikoleksi, yang mempunyai sifat berorientasi subjek, *integrated*, *time-variant*, dan bersifat tetap/tidak berubah (*non-volatile*) dari data yang terkoleksi dalam mendukung proses pengambilan keputusan managerial [8].

Ada empat tingkat data di dalam arsitektur basis data yang dapat dilihat pada Gambar 2.3 yaitu sebagai berikut :

1. Tingkat Operasional

Pada tingkatan ini data yang akan diolah merupakan data yang digunakan dalam kehidupan sehari-hari, terperinci, dan berorientasi aplikasi.

2. Tingkat *Data Warehouse*

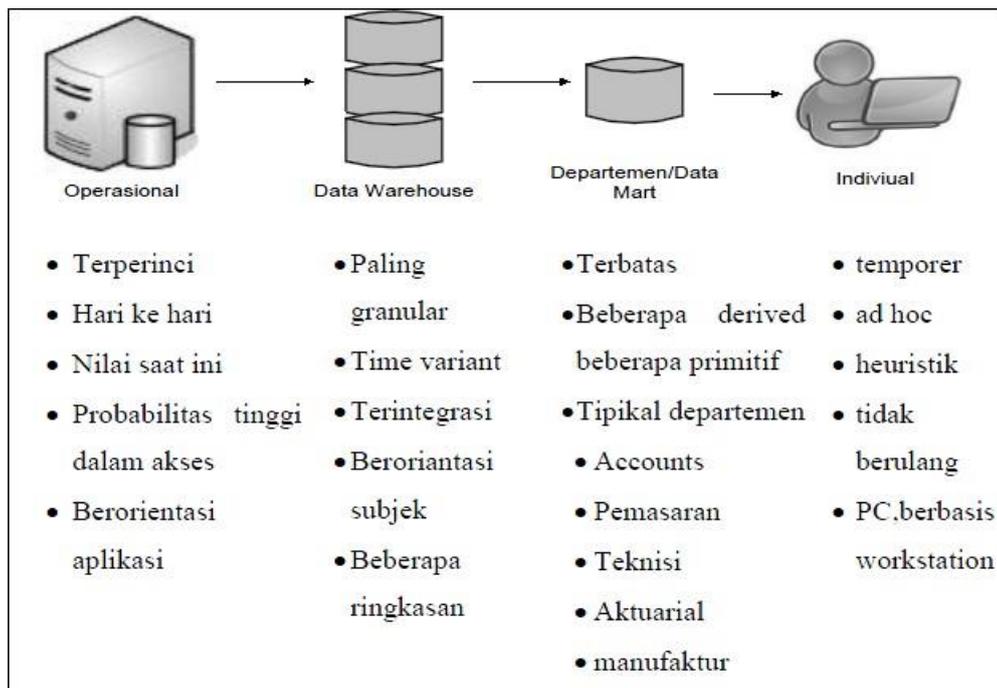
Merupakan kumpulan koleksi data yang mempunyai sifat berorientasi subjek, *integrated*, *time-variant*, dan *non-volatile*. *Data warehouse* merupakan kumpulan dari beberapa *data mart*.

3. Tingkat Departemen/*data mart*

Data mart merupakan bagian terkecil dari sebuah *data warehouse*, *data mart* hanya fokus pada sebuah subjek tunggal atau suatu area fungsional seperti bagian penjualan, produksi, keuangan dan lainnya [20].

4. Tingkat individual

Data individual biasanya memiliki sifat yang temporer dan hanya dalam ruang lingkup yang kecil. Didukung oleh perangkat komputer di mana proses EIS yang berlangsung hanya berjalan di dalam level individu.



Gambar 2.3 Arsitektur Data Warehouse

Data warehouse dapat dikatakan sebagai suatu salinan dari OLTP (*On-Line Transaction Processing*) yang mana berbentuk terstruktur dan dapat digunakan untuk memenuhi kebutuhan analisis, *reporting*, maupun *data mining*. OLTP sendiri adalah sebuah proses yang menitikberatkan pada transaksinya, seperti input data dan lain-lain. Pada Tabel 2.1 akan memperlihatkan perbedaan OLTP *system* dan *data warehouse*.

Tabel 2.1 Perbedaan OLTP dan Data Warehouse [9].

OLTP	Data Warehouse
Dirancang untuk operasi <i>real-time</i> bisnis	Dirancang untuk analisis dari suatu bisnis berdasarkan atribut dan kategori
Menangani data saai ini.	Menangani data saat ini dan data masa lalu
Data disimpan pada beberapa <i>platform</i> .	Data disimpan pada satu platform saja
Data diorganisir berdasarkan fungsi atau operasinya	Data diorganisir berdasarkan subjek.
Prosesnya bersifat berulang.	Prosesnya dilakukan setiap saat dan harus berorientasikan waktu (historical).
Untuk operasional.	Untuk managerial.
Berorientasi pada transaksi.	Berorientasi pada analisis.

2.2.4.1 Tujuan Data Warehouse

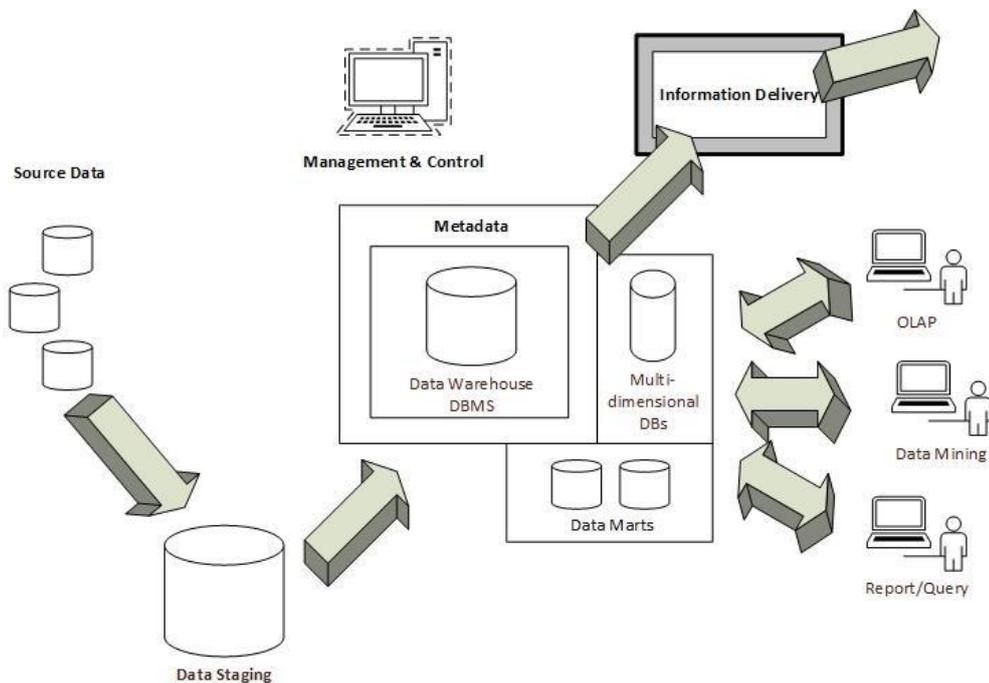
Adapun tujuan dari data *warehouse* itu sendiri adalah sebagai berikut [9] :

1. Dapat memberikan kemudahan pengguna untuk mengakses informasi yang ada. Kemudahan disini berbicara tentang efisiensi. Data *warehouse* harus efisien sehingga dengan mudah dipahami oleh pengguna bukan hanya *olive developer* saja. Selain hal tersebut, pengguna juga dapat mengkombinasikan data yang ada dalam suatu data *warehouse* dengan berbagai cara seperti (*slicing and dicing*). Untuk suatu data *warehouse* disarankan sebaiknya dilakukan dengan cara yang sederhana dan mudah dioperasikan.
2. Menyediakan informasi yang konsisten.
Data *warehouse* hanya berisi informasi-informasi yang relevan bagi kebutuhan pengguna untuk mengambil suatu keputusan. Oleh karena itu, konsistensi data yang terdapat dalam data *warehouse* harus dapat dipertanggungjawabkan.
3. Mampu beradaptasi dan tahan terhadap perubahan.
Perubahan-perubahan yang dapat terjadi dapat suatu dapat harus dapat diantisipasi oleh data *warehouse*. Dengan kata lain, data *warehouse* harus dirancang agar dapat menghadapi setiap perubahan yang terencana. Hal ini berarti perubahan yang terjadi tidak boleh merusak atau mengganggu data dan aplikasi yang telah ada sebelumnya.
4. Mampu mengamankan informasi.
Informasi yang sudah diolah oleh data *warehouse* harus tersimpan dalam data *warehouse* dengan aman. Dapat dikatakan informasi tersebut tidak boleh sampai jatuh ke tangan yang salah. Karena hal tersebut, sebuah data *warehouse* yang baik harus dapat mengendalikan setiap akses dari informasi yang ada.
5. Mampu memberikan dukungan dalam pengambilan suatu keputusan.
Ini merupakan tujuan yang paling penting dan harus ada dalam swtiap pembuatan data *warehouse*. Data *warehouse* bisa didefinisikan sebagai sekumpulan teknologi pendukung untuk mengambil suatu keputusan, dimaksudkan agar setiap pekerjaan yang berhubungan dengan sebuah informasi dapat membuat keputusan dengan cepat, tepat dan detail.
6. *User friendly*.

Seperti pada tujuan data *warehouse* pertama, data *warehouse* harus dirancang agar memudahkan pengguna ketika dioperasikan . Tidak seperti sistem operasional dimana seringkali pengguna tidak memiliki pilihan yang lain kecuali menggunakan sistem baru, akan terjadi pengguna data *warehouse* biasanya merupakan pilihan. Oleh karena itu, proses penentuan pengguna data *warehouse* merupakan factor yang sangat penting.

2.2.4.2 Arsitektur Data Warehouse

Arsitektur pada data *warehouse* dikelompokkan menjadi 5 bagian Gambar 2.4 [10].



Gambar 2.4 Arsitektur Data Warehouse [10]

1. *Source Data*

Merupakan sumber data, atau bisa dikatakan bahwa darimana data itu berasal. Untuk membangun sebuah data *warehouse* Perusahaan Textile maka *source data* berasal dari operasional *system* atau *OLTP database*.

2. *Data Staging*

Merupakan sebuah proses yang biasanya dilakukan sebelum data *source* masuk ke dalam sebuah data *warehouse*. Proses yang terjadi ini dinamakan proses ETL (*Extracting, Transformation, Loading*).

3. *Data Warehouse*

Merupakan tempat penyimpanan data yang multidimensional, dimana data yang tersimpan berupa *metadata*, *summary* data dan *row* data.

4. *Data Mart*

Merupakan bagian inti terkecil dari sebuah data *warehouse*. Seluruh data yang ada akan dibentuk menjadi sebuah data *mart*, seluruh data *mart* yang telah dibentuk apabila digabungkan akan menjadi sebuah data *warehouse* [20].

5. *User*

Merupakan pengguna yang akan menggunakan data *warehouse*. *User* bisa memiliki tugas yang berbeda-beda, ada yang untuk dilakukan analisis, ada yang melakukan *reporting*, dan ada yang untuk membentuk data *mining*. Intinya informasi yang berasal dari data *warehouse* akan diolah menjadi “sesuatu” menggunakan *tools* yang ada dan berguna untuk pengambilan suatu keputusan [20].

2.2.4.3 Dimensional Data Warehouse

Kebutuhan pengguna dan realitas data yang menjadi factor penentu untuk merancang dimensional model data *warehouse*, seperti bisnis apa yang paling diperlukan, detailnya seperti apa dan dimensi-dimensi serta fakta-fakta apa yang harus dilakukan [6].

Maka dimensional model harus sesuai dengan kebutuhan dari pengguna. Dimensional model itu sendiri harus dirancang sedemikian rupa agar dapat bertahan dan dapat beradaptasi dari segala perubahan yang akan terjadi suatu saat nanti. Desain model yang dihasilkan dibentuk menjadi sebuah *database* relasional yang nantinya dapat mendukung OLAP *cubes* untuk menyediakan secara “instan” hasil *query* dimana hasil *query* tersebut digunakan untuk analisis.

1. Tabel Dimensi (*Dimension Tables*)

Tabel dimensi biasanya menjelaskan mengenai entitas bisnis dari suatu *enterprise* atau yang dapat disebut yaitu atribut yang langsung mengacu kepada tabel fakta

[6]. Tabel dimensi biasanya berisi data keterangan dimana data tersebut jarang sekali mengalami perubahan.

2. Tabel Fakta (*Fact Tables*)

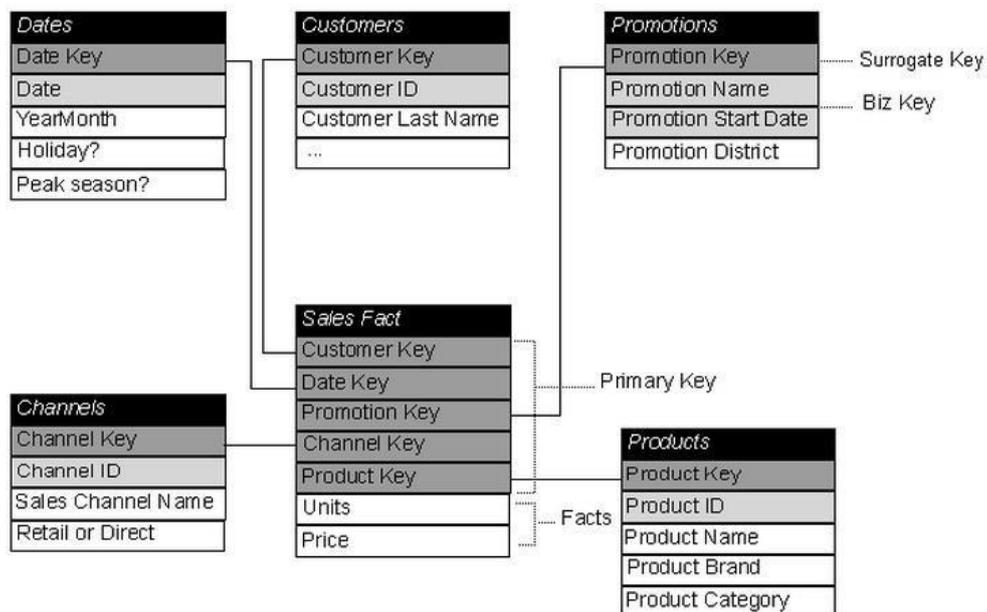
Tabel fakta merupakan sebuah table yang menjelaskan tentang transaksi bisnis dari suatu *enterprise* biasanya disebut table detail [6]. Tabel fakta umumnya berisi data yang berkaitan langsung dengan proses bisnis.

3. Skema *Dimensional Model*

Berikut ini adalah beberapa skema yang biasa digunakan dalam membangun sebuah data *warehouse* :

A) Skema *Star* (*Star Schema*)

Suatu skema disebut skema *star* jika seluruh tabel dimensi tidak dilakukan normalisasi dan setiap tabel dimensi berhubungan langsung dengan tabel fakta dan satu tabel fakta wajib memiliki relasi minimal dengan satu buah tabel dimensi [6]. Gambar 2.5 dari skema *star*.

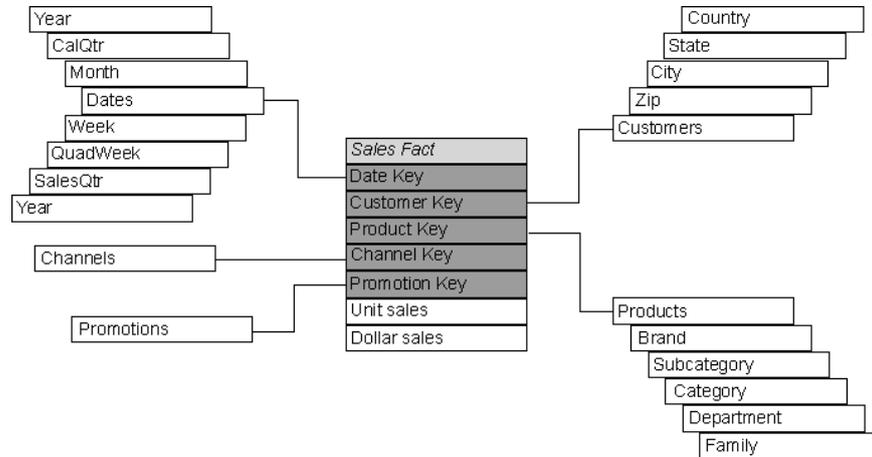


Gambar 2.5 Skema *Star* [7].

B) Skema *Snowflake*

Suatu skema disebut sebagai skema *snowflake* jika satu atau lebih tabel dimensi tidak berhubungan secara langsung dengan tabel fakta, tetapi harus

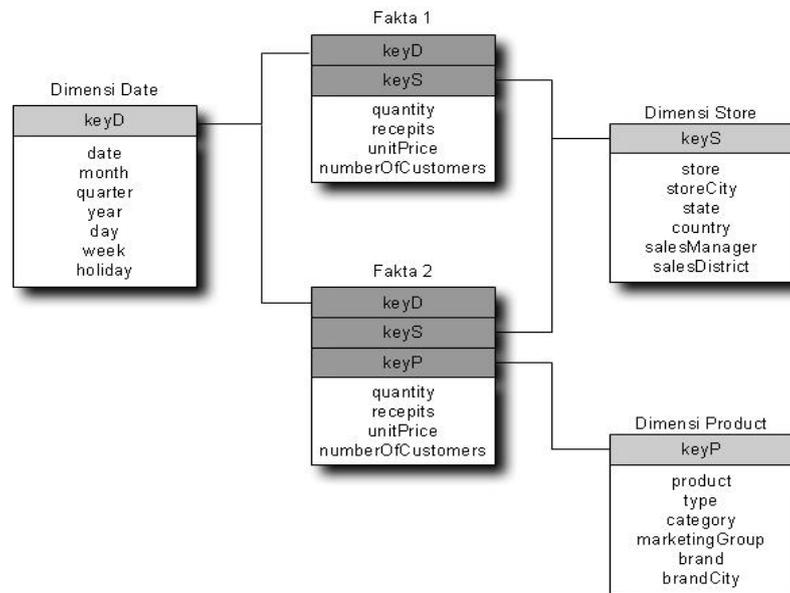
dinormalisasi dan berhubungan melalui dimensi lain baru berhubungan dengan tabel fakta[6]. Gambar 2.6 dari skema *snowflake*.



Gambar 2.6 Skema Snowflake [7]

C) Skema *Fact Constellation*

Suatu skema dapat dikatakan sebagai skema *fact constellation* jika tabel sdimensi dipakai bersamaan oleh satu atau lebih tabel fakta [11]. Gambar 2.7 dari skema *fact constellation*.



Gambar 2.7 Skema Fact Constellation [11]

2.2.5 Data Mart

Data *Mart* adalah bagian terkecil atau bagian inti dari sebuah data *warehouse* yang berfokus pada pusat keuntungan bagi perusahaan, biasanya identic dengan penjualan. Data *mart* sendiri memiliki fasilitas penyimpanan data yang berorientasi pada *Subject* tertentu atau berorientasi pada departemen tertentu dari suatu organisasi, fokus pada kebutuhan departemen tertentu seperti *Sales*, *Marketing*, *Operation* atau *Collection*. Sehingga suatu organisasi pun dapat mempunyai lebih dari satu data *mart*. Data *mart* berisi informasi yang relevan bagi pengguna yang ingin mengambil keputusan, sebuah perusahaan yang telah menginvestasikan baik uang dan waktunya dalam sebuah bisnis operasional yang telah mempunyai dan menyimpan sebuah data dari pelanggan, penjualan dan inventori [1].

2.2.5.1 Perbedaan Data *Mart* dan Data Warehouse

Tidak seperti data *mart*, data *warehouse* menangani beberapa subjek area dan biasanya diterapkan dan dikendalikan oleh organisasi terpusat seperti perusahaan yang berbasis Information Technology (IT). Seringkali hal itu disebut data *warehouse* yang terpusat atau *enterprise data warehouse*. Biasanya data *warehouse* mengumpulkan data dari beberapa sistem.

Pada bagian ini tidak ada ketentuan khusus yang membatasi ukuran dari data *mart* ataupun dari segi kompleksitas *decision-support* data yang ada pada data *mart*. Meskipun adanya begitu, data *mart* cenderung lebih kecil dan kompleksitasnya tidak serumit dan kurang dibanding data *warehouse*. Oleh karena itu data *mart* lebih mudah untuk dibangun dan dipelihara. Perbedaan mendasar antara datawarehouse dan datamart dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 Perbedaan Antara Data Warehouse dan Data Mart

Kategori	Data Warehouse	Data Mart
Cakupan	Perusahaan	Bagian Tertentu / Divisi Tertentu
Subjek	Beberapa Subjek	Subjek Tunggal
Sumber Data	Banyak	Lebih Sedikit
Ukuran	100 GB-TB+	<100 GB

Waktu Pelaksanaan	Bulan hingga Tahun	Bulan
-------------------	--------------------	-------

2.2.5.2 Dependent dan Independent Data Mart

Terdapat dua jenis data *mart* : *Dependent* dan *Independent*. Kategori yang ada tersebut diambil berdasarkan pada sumber data *mart*. *Dependent* data *mart* itu sendiri mengambil sekumpulan data dari sebuah data *warehouse* yang telah dibuat. Sedangkan *independent* data *mart* adalah sistem *standalone* yang dibangun dengan mengambil data langsung dari data operasional atau data eksternal atau keduanya [17].

Perbedaan utama antara *dependent* dan *independent* data *mart* adalah bagaimana ketika data *mart* akan diisi dan bagaimana mendapatkan data dari sumber lalu cara mengisinya ke dalam suatu data *mart*. Langkah ini disebut proses ETL (*extract, transformation, loading*). Proses ini melibatkan peindahan data dari sistem operasional, lalu memfilternya, dan memasukan ke dalam data *mart* [18].

Dalam *dependent* data *mart* prosesnya akan lebih sederhana karena data yang telah diformat dan diringkas akan dimuat ke dalam data *warehouse*. Proses ETL yang dilakukan di dalam *dependent* data *mart* sebagian besar adalah proses identifikasi subset data apa saja yang tepat dan relevan sesuai dengan subjeknya dan setelah itu *copy* dari subset data akan berpindah ke data *mart* dalam bentuk data yang sudah diringkas.

Dalam data *mart independent* akan melakukan semua aspek dari proses ETL, seperti halnya data *warehouse*. Namun jumlah sumber data akan lebih sedikit dan jumlah data yang terkait dengan data *mart* akan kurang dari data *warehouse*, mengingat fokus data *mart* adalah pada subjek tunggal.

Ini merupakan alasan di balik pembangunan kedua jenis data *mart* tersebut biasanya akan berbeda. *Dependent* data *mart* biasanya dibangun untuk peningkatan kinerja dan ketersediaan, kontrol yang lebih baik, dan biaya komunikasi yang lebih rendah karena setiap *departemen* mempunyai data *mart* masing-masing. Pembangunan *independent* data *mart* sering didorong oleh kebutuhan untuk mendapatkan solusi dalam waktu yang lebih singkat.

2.2.6 Tahapan Pembentukan Data *Mart*

2.2.6.1 Analisis Sumber Data

Terdapat beberapa data yang dapat dijadikan sebagai acuan dan sumber data masukan untuk sebuah data *mart* di antaranya :

A. Data Transaksional

Data transaksional merupakan data yang identik dengan sebuah transaksi atau penjualan yang dihasilkan dari suatu proses bisnis transaksi yang dilakukan oleh perusahaan. Contohnya : transaksi penjualan barang.

B. Data Internal

Data internal yaitu data-data yang menggambarkan situasi dan kondisi pada suatu organisasi secara intern. Misalkan data keuangan, data pegawai, data produksi, dan sebagainya.

C. Data History

Data history merupakan data yang berasal dari masa lampau yang sudah lalu dan di simpan di suatu media penyimpanan.

D. Data External

Data yang menggambarkan situasi dan kondisi yang ada di luar organisasi. Contoh persebaran pelanggan, produk, dan sebagainya.

2.2.6.2 Data Staging

Data *Staging* yaitu tempat dan area dengan sekumpulan fungsi untuk membersihkan, merubah, mengkombinasikan, penggandaan dan menyiapkan sumber data untuk penyimpanan dan digunakan pada data *mart*. Pada data *staging* terdapat suatu proses yang disebut proses ETL (*Extract, Transform dan Load*).

A. *Extraction*

Extraction atau ekstraksi merupakan proses untuk pemilihan, menentukan seluruh sumber data yang relevan dan melakukan pengambilan data dari sumber data tersebut. Penentuan pendekatan yang digunakan pada proses ekstraksi sangat terkait dengan analisis bisnis proses, pendefinisian area subjek, serta desain logik/fisik data *mart*. Adapun terhadap proses *extraction* ini meliputi proses penyaringan data yang akan digunakan dalam pembuatan data *mart*. Data *mart* yang sudah disaring dapat langsung dimasukkan ke dalam tempat penampungan sementara terlebih dahulu. Pada hakikatnya bagian dari ekstraksi melibatkan penguraian dari data yang telah diekstrak, menghasilkan suatu pengecekan jika data bertemu dengan suatu struktur atau pola yang diharapkan. Jika tidak sesuai maka data tersebut akan ditolak secara keseluruhan. Adapun fungsi ekstraksi diantaranya, yaitu.

1. Ekstraksi data secara otomatis dari sumber aplikasi awal.
2. Penyaringan atau seleksi data hasil ekstraksi.
3. Pengiriman data dari berbagai *platform* aplikasi ke sumber data yang ada.
4. Perubahan format *layout* data dari format ke aslinya.
5. Penyimpanan yang terdapat dalam *file* sementara akan dilakukan untuk proses penggabungan dengan hasil ekstraksi dari sumber data yang lain.

B. *Transformasi*

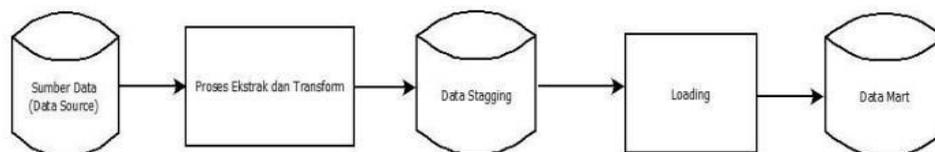
Transformasi data yang telah diekstrak ke dalam format yang diperlukan. Hal tersebut perlu dilakukan mengingat data yang diambil berasal dari sumber yang berbeda dan format data yang berbeda, kemungkinan memiliki standardisasi yang berbeda. Data dari beberapa sistem perlu ditransformasi ke dalam format umum yang disepakati dan digunakan dalam data *mart*. *Transformation* adalah proses yang memiliki peran dalam melakukan perubahan, pembagian dan mengintegrasikan skema serta struktur yang berbeda-beda ke dalam skema dan struktur yang terdefinisi dalam data *mart*.

Ada beberapa tahapan dalam proses *transformasi* diantaranya sebagai berikut:

1. Memetakan data input dari skema data aslinya ke skema data *mart*.
2. Melakukan proses konversi dari format data yang berbeda.
3. Membersihkan data yang mengandung duplikasi atau redundansi.
4. Penghitungan nilai-nilai derivat atau mula-mula.
5. Penghitungan nilai-nilai agregat atau rangkuman.
6. Pemeriksaan integritas referensi data.
7. Pengisian nilai-nilai kosong dengan nilai *default*.
8. Penggabungan data.

C. *Loading*

Loading adalah proses pemindahan data secara fisik dan sistem operasional dari sumber data awal ke dalam *database* data *mart*. Tahap load adalah men-load data ke dalam target akhir (end target), yang pada umumnya adalah data *mart*. Bergantung pada kebutuhan organisasi, proses ini bervariasi secara luas.



Gambar 2.8 ETL Proses

2.2.6.3 Online Analytical Processing (OLAP)

Online Analytical Processing (OLAP) adalah salah satu *tools* yang biasanya digunakan pada data *warehouse*/ data *mart* yang berfokus untuk mengakses suatu

informasi. Teknologi OLAP memungkinkan data mart digunakan secara aktif untuk proses *online analysis*, memberikan respon yang cepat terhadap *analytical queries* yang kompleks.

Data model multidimensional dan teknik agregasi data yang dimiliki oleh OLAP dapat mengatur dan membuat kesimpulan dari data dalam jumlah besar sehingga dapat di evaluasi secara cepat dengan menggunakan *online analysis* dan *grafical tool*.

Sistem OLAP menyediakan kecepatan dan fleksibilitas untuk men- support proses analisis secara *real time* [19].

2.2.6.4 OLAP Service

OLAP *services* adalah sesuatu yang menggambarkan beberapa perangkat yang bekerja bersama-sama untuk membantu dalam menganalisa data yang ada. Konsep dari OLAP *service* ini yaitu informasi yang sudah di dapatkan dari data *warehouse* atau data *mart* di ekstrak secara berkala lalu digunakan untuk memperbarui objek yang ada dalam OLAP *service*. Sedangkan untuk konsep OLAP *server* mengambil data dari data *warehouse* atau data *mart* dan *relational* OLAP serta memperbarui informasi yang telah disimpan dalam *multidimensional* OLAP (MOLAP). Setelah informasi tersebut dikumpulkan maka *pivot table service* yang bekerja sama dengan *excel pivot table* atau perangkat lain yang mendukung OLE DB dengan ekstensi OLAP dapat melakukan akses dan memanipulasi data yang ada [19].

2.2.6.5 Teknik penyimpanan data dalam OLAP

OLAP biasanya digunakan untuk melakukan pemrosesan sebuah informasi lalu menampilkan informasi tersebut dalam bentuk multidimensi. Terkadang data yang diolah dan ditampilkan dalam satu bentuk namun tidak menjamin data-data yang ada lalu disimpan dalam bentuk yang sama. Berikut ini ada tiga teknik dasar untuk menyimpan data OLAP [19].

A. Multidimensional OLAP (MOLAP)

Salah satu cara umum yang digunakan untuk menyimpan data yaitu di dalam basis data multidimensional. MOLAP ini bentuknya tidak seperti basis data relasional yang dapat menyimpan informasi dalam sejumlah baris dalam tabel.

Sebuah basis data multidimensional menyimpan informasi dalam sejumlah *query* multidimensional. Karena dimensi dapat diakses secara mudah, maka pengguna dapat melakukan *query* pada basis data MOLAP dengan sangat cepat. Biasanya dalam MOLAP sering terdapat data yang belum jadi, basis data MOLAP pun memiliki agregasi data sehingga dapat memberikan respon yang cepat terhadap *query*.

B. *Relational* OLAP (ROLAP)

Partisi ini menggunakan tabel *relational* dalam sebuah data *mart* untuk menyimpan suatu agregasi, sedangkan detail dari tabel fakta tersimpan dalam data *mart fact table*. Banyak orang Menganggap bahwa basis data yang dirancang secara khusus untuk sebuah keperluan analisis tidak dibutuhkan karena sebuah basis data relasional sudah cukup mampu untuk menampilkan informasi OLAP. Hal ini hanya berlaku pada tingkat tertentu saja, pada sebuah basis data yang terdiri dari ribuan atau ratusan ribu *record* maka menampilkan informasi OLAP akan menjadi sebuah masalah karena banyak data yang harus dihapus di *query*. Dan hal tersebut yang menjadi keterbatasan dari partisi ROLAP.

C. OLAP *Cube* (Kubus OLAP)

Olap *cube* merupakan objek utama yang tersimpan dalam sebuah basis data OLAP adalah sebuah *cube* (kubus). Sebuah kubus merupakan representasi yang multidimensi dari sekumpulan data yang mengandung data secara detail maupun tidak detail. Sebuah basis data OLAP dapat memiliki beberapa buah kubus sesuai dengan kebutuhan. Sebuah kubus dibangun menggunakan dua komponen utama yaitu ukuran fakta atau biasa disebut dengan (*measures*) dan dimensi. Ukuran atau measures merupakan nilai numerik atau suatu angka yang dapat dihitung dari sebuah tabel fakta dalam sebuah data *mart* seperti harga dari unit maupun kuantitas dari item. Sedangkan dimensi mengandung unsur yang berhubungan dengan tabel fakta dan identik dengan menggambarkan kategori dari ukuran yang ada. Seperti bagaimana ukuran berubah setiap waktu. Beberapa teknik yang dapat dilakukan pada kubus atau *report* multidimensi ini adalah :

2.2.6.6 Teknik Dalam OLAP

1. *Slice and Dice*

Slice and Dice memungkinkan untuk melihat kubus dari sudut pandang yang berbeda. Dengan *slice and dice* dapat ditentukan dimensi apa yang akan ditampilkan dan bagaimana mereka ditampilkan. Hal inilah yang menjadi keunggulan OLAP. Dengan cara melihat kubus dari berbagai sudut pandang yang berbeda maka akan dapat dipelajari banyak hal dari data yang diolah.

2. *Drill Down*

Sebagian besar informasi yang akan ditampilkan dalam OLAP merepresentasikan kesimpulan yang lebih detail. *Drilling Down* merupakan teknik untuk memecah sebuah informasi yang ada menjadi beberapa informasi yang lebih detail lagi. Sebagai contohnya, jika dilakukan proses *drill down* terhadap periode data tahunan maka akan dapat diperoleh data dalam periode waktu catur wulan tri wulan atau setiap bulan.

3. *Consolidation (Roll-Up)*

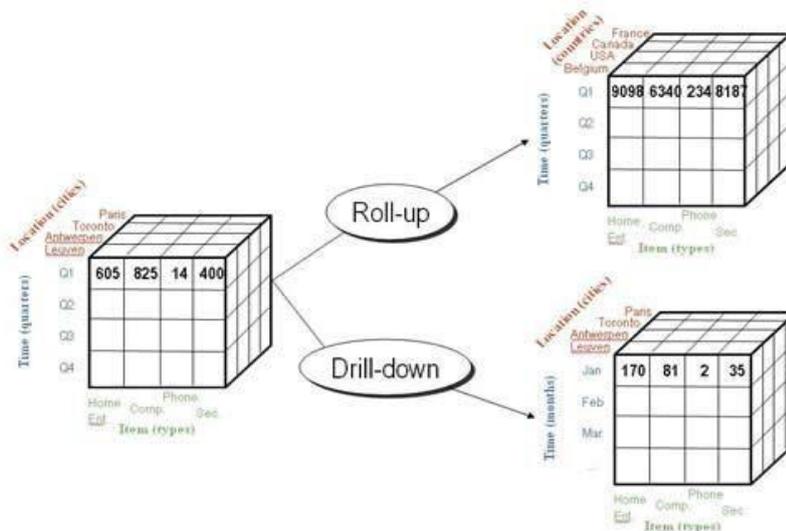
Consolidation atau lebih dikenal dengan *roll up* merupakan kebalikan dari *drill down*. Data yang ada sebelumnya dapat dilihat dalam bentuk triwulan akan dapat dilihat menjadi dalam format tahunan. Dengan *roll up* data dapat dilihat secara lebih global (*Summary*).

4. *Pivoting*

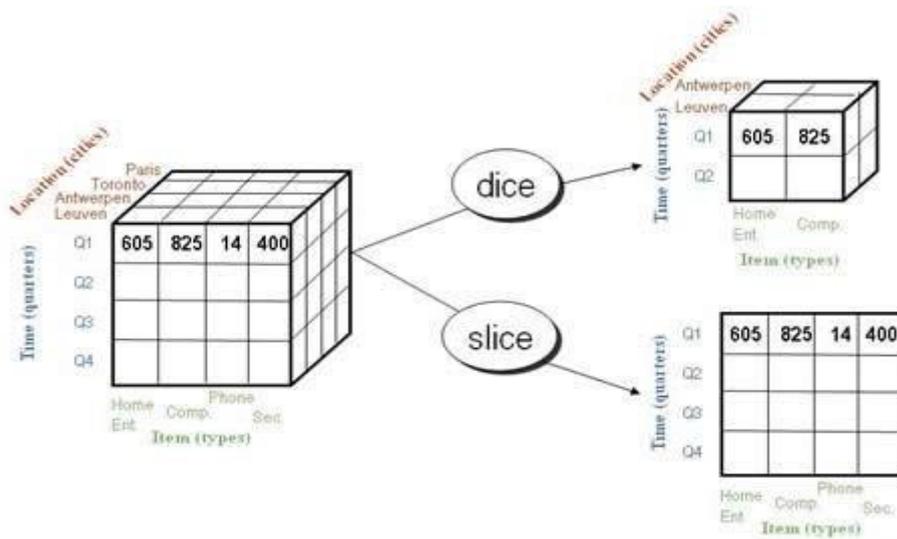
Pivoting merupakan suatu teknik untuk saling menukarkan suatu dimensi data. Dengan melakukan *pivoting*, maka dapat diamati suatu informasi atau data dari sudut pandang yang berbeda, sehingga diharapkan akan dapat memperjelas analisis yang dilakukan.

5. *Filtering*

Filtering merupakan suatu teknik untuk menyaring informasi yang dibutuhkan, sehingga para analisis tidak digabungkan oleh banyaknya informasi yang tersedia.



Gambar 2.9 Roll-Up Dan Drill Down



Gambar 2.10 Slice And Dice

2.2.7 Visualisasi Data

Bentuk visualisasi data terdapat tiga macam yaitu *bar chart*, *pie chart*, dan *line char*. Berikut ini deskripsi dari masing-masing bentuk visualisasi data :

1. *Pie Chart*

Pie Chart adalah sebuah tampilan deskriptif sederhana dari data, dimana data yang ditampilkan hanya jumlah dari total yang diberikan. *Pie Chart* merupakan sebuah cara yang paling ilustratif untuk menampilkan perbandingan kuantitas sebagai presentasi dari total yang diberikan. Total area dari sebuah *pie chart* mempresentasikan 100% dari kuantitas (jumlah dari nilai variabel pada seluruh kategori). Ukuran dari setiap potongan adalah presentasi dari total yang dipresentasikan oleh masing-masing kategori yang dapat dilihat pada Gambar 2.11 berikut:

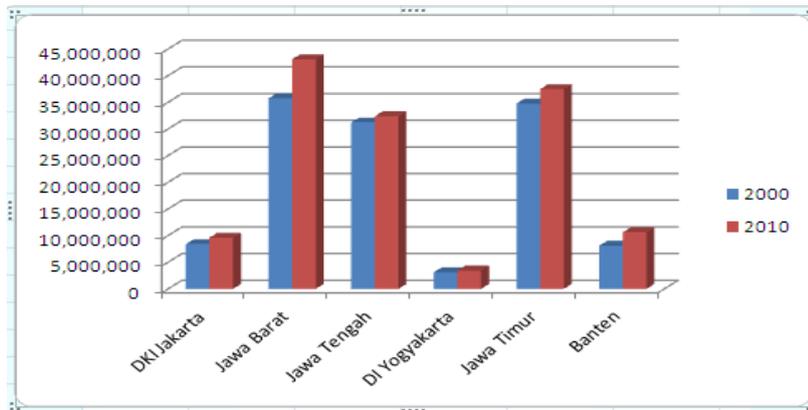


Gambar 2.11 Contoh *Pie Chart*

2. *Bar Chart*

Bar chart mayoritas digunakan untuk menunjukkan data berdasarkan kategori dan parameter tertentu dimana tidak ada total presentase yang ditekankan pada setiap kategori. *Bar chart* ini dapat disajikan macam cara baik itu secara vertikal maupun horizontal. Skala pengukuran adalah nominal atau ordinal. *Bar chart* sendiri sangat sering dipakai untuk menampilkan suatu data secara berkelanjutan seperti ukuran sendal atau warna baju dan data diskontinu seperti

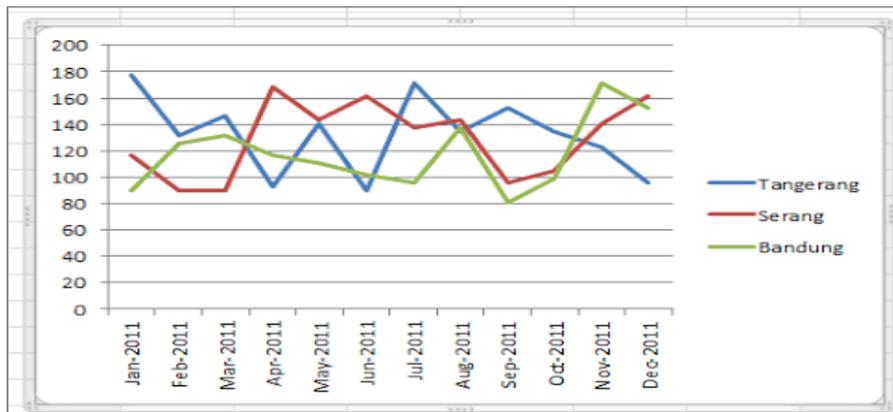
tinggi badan atau berat badan. Contoh dari bar chart dapat dilihat pada Gambar 2.12 berikut:



Gambar 2.12 Contoh Bar Chart

3. Line Chart

Line chart sering digunakan untuk menampilkan informasi yang ada dalam bentuk serangkaian titik-titik data yang dihubungkan dengan sebuah garis lurus. *Line chart* sendiri biasanya sering digunakan untuk memvisualisasikan trend data dalam periode waktu tertentu. Contoh dari *line chart* dapat dilihat pada Gambar 2.13 berikut:



Gambar 2.13 Contoh Line Chart

2.2.8 Microsoft SQL Server

Microsoft SQL Server merupakan sebuah sistem untuk melakukan manajemen basis data relasional (RDBMS) yang digunakan untuk aplikasi arsitektur dengan berbasis client-server. Bahasa *query* yang utamanya sering digunakan adalah *Transact-SQL* yang merupakan implementasi tersendiri dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan *Sybase*. Umumnya SQL Server digunakan di sunia bisnis yang memiliki basis data yang berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar. Pengguna yang menggunakan SQL Server dapat menyimpan banyak data dan dapat untuk diimplementasi untuk kepentingan bisnis dan perusahaan [7].

2.2.9 SSIS (SQL Server Integration Service)

SSIS (*SQL Server Integration Services*) adalah suatu platform untuk membangun sistem yang handal untuk integrasi data, *extraction*, *transformation*, dan *loading* yang digunakan pada data *warehousing*. SSIS sendiri menawarkan solusi dalam menghadapi permasalahan data integrasi. Selain itu, *tools* ini membantu untuk meningkatkan efisiensi waktu pembuatan [8].

Arsitektur dari *SQL Server Integration Services* secara umum dapat berisi berbagai macam komponen yang biasa digunakan, diantaranya adalah sebagai berikut :

1. *SSIS Deginer*. Salah satu *tools* yang sering digunakan untuk membuat dan mengatur paket *integration service*. Pada *SQL Server 2019*, *tools* ini sudah terintegrasi dengan *Visual Studio 2010* yang merupakan bagian *project Bussiness Intellegence*.
2. *Runtime Engine*. Komponen ini berguna untuk menjalankan semua paket yang ada SSIS yang telah dibuat.
3. *Task* dan *executable binary*.
4. *Data Flow Engine* dan *Data Flow*. Komponen data *flow* merupakan enkapsulasi data *flow engine* yang menyediakan *buffer* didalam memori dan bertugas memindahkan data dari sumber data ke tujuan data. Sedangkan data *flow* itu sendiri merupakan aliran data, tujuan data, dan transformasi.

5. *Integration Services service*. *SQL Server Management Studio* dapat digunakan dalam melakukan proses monitoring paket SSIS dan mengatur SSIS *storage* yang digunakan.
6. *SQL Server Import dan Export Wizard*. Adapun *Tools* ini kebanyakan digunakan untuk melakukan proses *copy* data dari sumber data awal ke tujuan data.

2.2.10 Diagram Konteks

Diagram konteks adalah level teratas dari diagram aliran data, yaitu diagram yang tidak detail dari sebuah sistem informasi yang menggunakan aliran-aliran data kedalam dan keluar entitas eksternal. Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada di luar sebuah sistem yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

Diagram konteks adalah sebuah diagram alir sederhana yang menggambarkan hubungan antara entitas luar, masukan dan keluaran dari suatu sistem. Diagram konteks biasanya direpresentasikan dengan sebuah lingkaran tunggal yang mewakili seluruh dari suatu sistem.

2.2.11 Data Flow Diagram

Diagram Arus Data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dari mana data awalnya disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang sudah tersimpan dan proses yang dikenakan pada data tersebut

1. Komponen Terminator

Terminator sendiri merupakan perwakilan dari entitas eksternal yang berkomunikasi dengan sistem yang sedang dibangun atau dikembangkan. Biasanya sebuah *terminator* dikenal dengan nama entitas luar (*external entity*). Terdapat dua jenis *terminator*:

- a) *Terminator* Sumber (*source*) merupakan terminator yang dijadikan sebagai sumber.
- b) *Terminator* Tujuan (*sink*) merupakan terminator yang menjadi tujuan dari data-data yang dibutuhkan.

Terminator dapat berupa subjek seperti orang, sekelompok orang, organisasi, departemen di dalam organisasi, atau perusahaan yang sama tetapi di luar kendali sistem yang sedang dibuat modelnya. *Terminator* dapat juga menjadi sebuah subjek tertinggi dari sebuah perusahaan seperti berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan. Ada tiga hal penting yang harus diperhatikan mengenai *terminator*:

- a) *Terminator* merupakan bagian/lingkungan luar sistem. Alur dari sekumpulan data yang menghubungkan *terminator* dengan berbagai proses dari suatu sistem, menunjukkan hubungan sistem dengan dunia luar.
- b) Sistem yang kompeten tidak dapat mengubah isi atau cara kerja dari organisasi, atau prosedur yang berkaitan dengan *terminator*.
- c) Rangkaian yang ada setiap *terminator* yang satu dengan *terminator* yang lain tidak digambarkan pada proses DFD.

2. Komponen Proses

Komponen proses merupakan proses memvisualkan bagian dari suatu sistem yang dapat ditransformasikan input menjadi output. Komponen proses ini diberi nama untuk mendefinisikan mengenai proses atau kegiatan apa saja yang sedang/akan dilaksanakan. Pemberian dari nama proses yang dilakukan dengan menggunakan kata kerja transitif yaitu kata kerja yang membutuhkan objek. Ada beberapa hal yang perlu diperhatikan tentang proses:

- A. Proses harus memiliki input dan output.
- B. Proses ini bisa dirangkai dengan komponen terminator, data *store* atau proses melewati alur data.
- C. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.

Umumnya kesalahan proses di DFD adalah sebagai berikut:

- A. Proses yang memiliki input tetapi tidak menghasilkan output. Kesalahan yang terjadi seperti proses ini bisa disebut dengan *black hole* (lubang hitam), hal tersebut disebut *black hole* dikarenakan data yang telah di input lalu

masuk ke dalam proses akan hilang dan tidak berbekas atau tidak ada histori seperti dimasukkan ke dalam lubang hitam.

- B. Proses menghasilkan output tetapi tidak pernah menerima input. Kesalahan ini disebut dengan miracle (ajaib), karena ajaib dihasilkan output tanpa pernah menerima input.

2.2.12 Kamus Data

Kamus data merupakan kumpulan data-data. Kamus data adalah katalog yang berisikan fakta mengenai data dan kebutuhan informasi apa saja dari suatu sistem informasi. Kamus data merupakan tempat penyimpanan definisi dari aliran-aliran data, file-file dan proses-proses dalam sebuah sistem informasi. Dengan mengimplementasikan kamus data, analisis sistem dapat lebih mudah dalam memberikan informasi mengenai definisi struktur pemakaian dari masing-masing elemen, dapat mendeskripsikan data yang mengalir di sistem dengan jelas dan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antar elemen.

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar arti yang sama mengenai masukan, keluaran, penyimpanan dan proses.

Kamus data mendefinisikan data elemen dengan cara:

1. Memaparkan arti dari alur data dan data *store* dalam DFD .
2. Memaparkan komposisi paket data yang terjadi terhadap alur data ke dalam alur yang lebih *elementary* (kecil). Contoh : alamat rumah pelanggan yang terdiri dari nama jalan, kota dan kode pos.
3. Memaparkan komposisi dari sebuah paket data di dalam data *store*.
4. Mengkhususkan nilai dan unit informasi dalam alur data dan data *store*.
5. Memaparkan hubungan yang jelas antara data *store* dalam suatu *entity relationship diagram* (ERD).