

## BAB II

### LANDASAN TEORI

#### 2.1 Data

Data adalah sesuatu yang mewakilkan objek dan peristiwa yang memiliki arti dan sangat penting bagi pemakai (Hoffer, Prescott dan McFadden ,2007 , p6).

#### 2.2 Basis Data

Basis data adalah suatu koleksi data yang saling berhubungan secara logikal, dan sebuah deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi (Connolly, 2005, p33).

#### 2.3 Data Warehouse

Menurut Inmon (2002:31), *Data Warehouse* adalah sekumpulan data yang bersifat *subject-oriented*, *integrated*, *time-variant* dan *non-volatile* untuk mendukung proses pengambilan keputusan strategis organisasi.

##### 2.3.1 Karakteristik Data Warehouse

Menurut Inmon (2002, p35), karakteristik dari *Data Warehouse* yaitu: *subject oriented*, *integrated*, *nonvolatile* dan *time variant*.

##### 1. Subject Oriented (Berorientasi Subjek)

*Data Warehouse* diorganisasikan di sekitar subjek-subjek utama dari perusahaan atau organisasi dari pada area-area aplikasi utama. Hal ini

tercermin dari kebutuhan untuk menyimpan data yang mendukung pengambilan keputusan dari pada data yang berorientasi aplikasi.

## 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 pengguna.

## 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*.

## 4. *Nonvolatile* (Tidak Mudah Berubah)

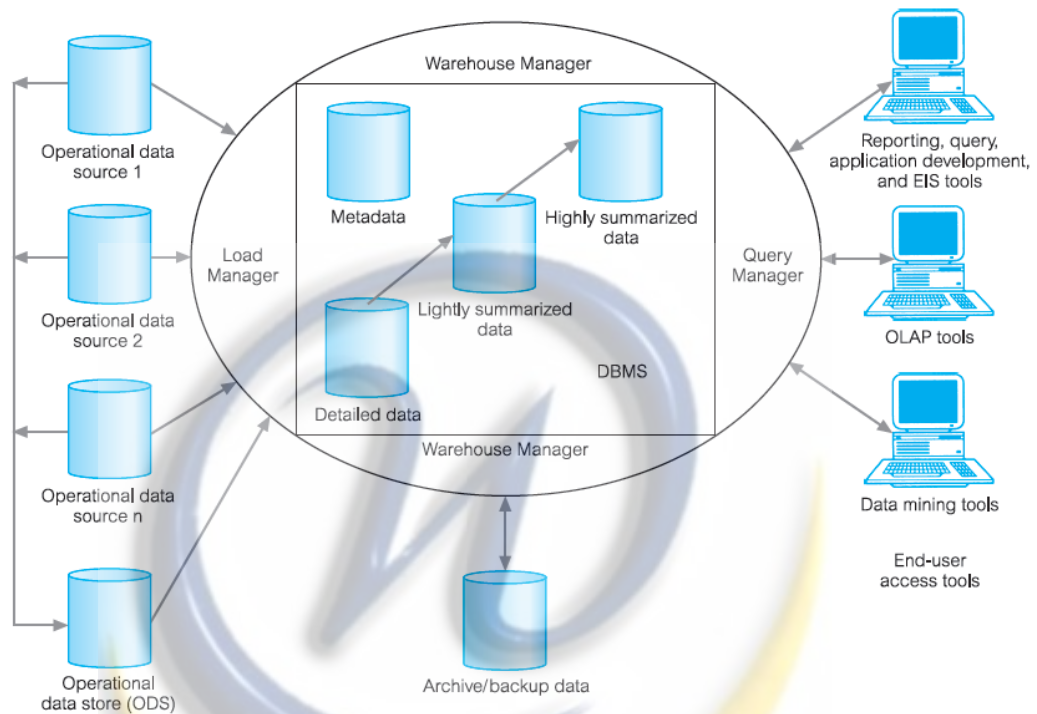
Data tidak diupdate secara *real-time* tetapi diperbaharui dari sistem operasional pada basis sehari-hari. Data baru selalu ditambahkan sebagai suplemen untuk sebuah *database*, bukan pengganti. *Database* secara terus menerus menyerap data baru, dan secara bertingkat diintegrasikan dengan data sebelumnya.

Hal-hal yang berkaitan dengan *Data Warehouse* dalam penerapan pada sistem *Business Intelligence* (Inmon, 2002) adalah:

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 history dimana *key* (kunci) yang dihasilkan sangat unik, karena *key* tersebut terdiri dari *foreignkey* (kunci asing) yang merupakan *primarykey* (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.

### 2.3.2 Arsitektur *Data Warehouse*

Menurut Conolly dan Begg (2005, p1157) arsitektur *Data Warehouse* adalah sebagai berikut:



**Gambar 2.1 Arsitektur *Data Warehouse* (Connolly, 2005, p1157)**

Komponen-komponen dari *Data Warehouse* menurut Connolly dan Begg (2005, p1157), antara lain:

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 yang telah diekstrak dari system 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 table 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 Warehouse* 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 *end user*. *Query tools* untuk *Data Warehouse*

relasional dirancang untuk menerima SQL atau menghasilkan *statement* SQL untuk memproses data yang disimpan didalam *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* memiliki kemampuan untuk menggantikan kemampuan dari *OLAP tools*. *Data Mining* memiliki kemampuan untuk membangun model prediktif dari pada model retrospektif.

### 2.3.3 Perancangan *Data Warehouse*

Salah satu dari metodologi perancang *Data Warehouse* adalah *Nine-Step Methodology* oleh Kimball (1996) yang memiliki 9 (sembilan) langkah sebagai berikut:

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 table

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. Jika ada dimensi yang muncul dalam dua *data mart*, maka kedua *data mart* tersebut harus memiliki dimensi yang sama, atau salah satu *data mart* adalah subset matematis dari *data mart* yang lain. Hanya dengan cara ini dua *data marts* berbagi satu atau lebih dimensi pada aplikasi yang sama. Ketika sebuah dimensi digunakan oleh lebih dari satu *datamart* dan dimensi tersebut tidak disinkronisasikan antar *datamart* maka seluruh *Data Warehouse* akan gagal, karena dua *data mart* tidak dapat digunakan pada saat yang bersamaan.

### 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.

### 2.3.4 Pemodelan Dimensional

Menurut Connolly dan Begg (2005, p1183), 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. 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 *starschema* atau *star join*.

### 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 didenormalisasi. 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*. *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.

## 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.

## 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*.

## 2.4 *Business Intelligence*

Menurut Niu (2009, p57), *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 *datawarehouse*. Proses *Business Intelligence* meliputi lima tahapan :

#### 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.

### 2.4.1 Arsitektur Sistem *Business Intelligence*

Menurut Niu (2009, p72), pada umumnya sistem *Business Intelligence* terdiri dari empat level komponen dan modul manajemen *metadata*. Arsitektur general dari sistem *Business Intelligence* terlampir pada gambar 1. Komponen-komponen saling berinteraksi untuk memfasilitasi fungsi dasar *Business Intelligence*: mengekstrak data dari sistem operasional perusahaan, menyimpan data yang sudah diekstrak kedalam *datawarehouse*, dan menarik data yang disimpan untuk berbagai aplikasi analisis bisnis.

- a. Level sistem operasional. : Sebagai sumber data dari sistem *Business Intelligence*, sistem operasional bisnis pada umumnya menggunakan sistem *online transaction processing (OLTP)* untuk mendukung kegiatan bisnis sehari-hari. Pada umumnya sistem OLTP adalah sistem penerimaan order pelanggan, sistem keuangan, dan sistem sumber daya manusia.
- b. Level akuisisi data : Pada level ini adalah komponen pra proses terdiri dari 3 tahapan yaitu : ekstraksi, transformasi, dan memasukkan (ETL). Sebuah perusahaan memiliki beberapa sistem OLTP yang menghasilkan jumlah data yang sangat besar. Data tersebut pertama kali diekstrak dari sistem OLTP oleh proses ETL dan kemudian ditransformasi sesuai dengan aturan



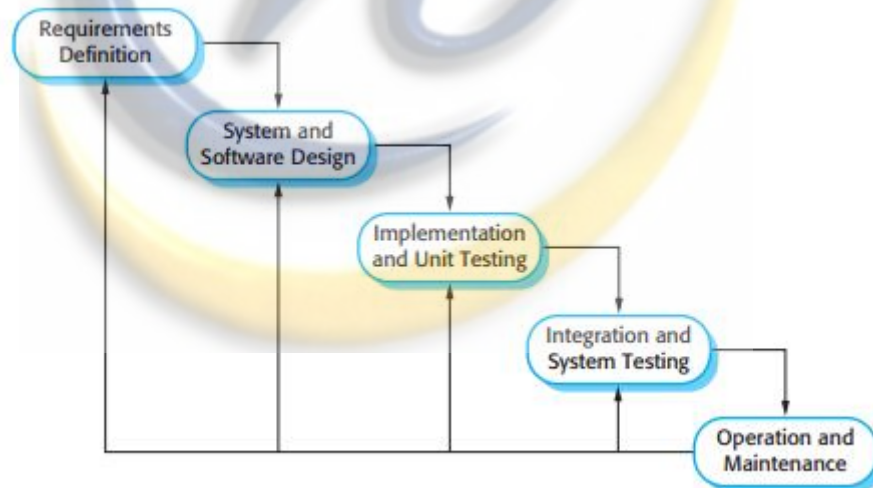
transformasi. Apabila data yang sudah ditransformasi maka data tersebut dimasukkan ke *Data Warehouse*. ETL merupakan komponen dasar dari sistem *Business Intelligence* karena kualitas data dari komponen lain tergantung pada proses ETL. Dalam perancangan dan pengembangan ETL, kualitas data, fleksibilitas sistem dan kecepatan proses adalah perhatian utama.

- c. Level penyimpanan data : Data yang telah diproses oleh komponen ETL disimpan dalam *Data Warehouse* dimana biasanya diimplementasikan dengan menggunakan tradisional sistem manajemen *database (RDMS)*. *RDMS* didesain untuk mendukung proses transaksi, sangat bertolak belakang dengan *Data Warehouse* berfokus kepada subyek, varian waktu dan disimpan secara terintegrasi. Skema *star* dan *snowflake* merupakan skema *Data Warehouse* yang paling populer. Apapun skema yang dipakai, tipe tabel pada *Data Warehouse* adalah *fact tables* dan *dimension tables*.
- d. Level analitis : Berdasarkan *Data Warehouse*, berbagai macam aplikasi analitis telah dikembangkan. Sistem *Business Intelligence* mendukung 2 tipe dasar dalam fungsi analitis: pelaporan dan *online analytical processing (OLAP)*. Fungsi pelaporan menyediakan manajer berbagai jenis laporan bisnis seperti laporan penjualan, laporan produk, dan laporan sumber daya manusia. Laporan dihasilkan dari menjalankan *queries* kedalam *Data Warehouse*. *Data Warehouse queries* pada umumnya sudah didefinisikan oleh pengembang *Data Warehouse*. Laporan yang dihasilkan

oleh sistem *Business Intelligence* biasanya memiliki format yang statis dan berisi tipe data yang pasti.

## 2.5 Metode Waterfall

Pada perancangan sistem ini, penulis menggunakan metode *waterfall*. Menurut Sommerville (2010, p29) Metode *Waterfall* meliputi kegiatan definisi kebutuhan, pengembangan, validasi dan evolusi, ada beberapa tahapan proses seperti yaitu definisi kebutuhan, perancangan perangkat lunak, implementasi, pengujian dan sebagainya. Metode *waterfall* adalah metode yang menyarankan sebuah pendekatan yang sistematis dan sekuensial melalui tahapan SDLC untuk membangun sebuah perangkat lunak. Berikut adalah gambar dari metode *waterfall*.



**Gambar 2.2 Metode Waterfall (Sommerville. 2010, p30)**

Adapun beberapa tahapan dari metode *waterfall*, diantaranya sebagai berikut :

1. Definisi Kebutuhan yaitu mendefinisikan kebutuhan secara rinci yang dibutuhkan untuk membangun program.
2. Desain perangkat lunak yaitu meliputi identifikasi dan menjelaskan penggambaran sistem perangkat lunak.
3. Implementasi dan *unit testing*, pada tahap ini desain perangkat lunak direalisasikan, setiap unit yang dilakukan pengujian harus terverifikasi dan memenuhi spesifikasi.
4. Intergrasi dan pengujian sistem, program diintegrasikan dan dilakukan pengujian sebagai satu kesatuan sistem untuk memastikan bahwa kebutuhan perangkat lunak telah dipenuhi.
5. Operasi dan pemeliharaan, ini merupakan fase terpanjang dari siklus hidup. Sistem dipasang dan dimasukkan kedalam penggunaan praktis. Pemeliharaan meliputi memperbaiki kesalahan yang tidak ditemukan sebelumnya dan meningkatkan layanan sistem.

## **2.6 Pendidikan Akademik**

Menurut Undang-Undang Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional, mendefinisikan pendidikan sebagai suatu usaha sadar dan terencana untuk mewujudkan suasana belajar dan proses pembelajaran agar peserta didik secara aktif mengembangkan potensi dirinya untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak mulia, serta keterampilan yang diperlukan dirinya, masyarakat, bangsa dan negara.

Pendidikan akademik adalah pendidikan tinggi yang diarahkan terutama pada penguasaan ilmu pengetahuan dan pengembangannya, dan lebih mengutamakan peningkatan mutu serta memperluas wawasan ilmu pengetahuan. Pendidikan akademik diselenggarakan oleh sekolah tinggi, institut, dan universitas.

## 2.7 Metode Visualisasi Data

Menurut Kirk (2012, p120), Memilih metode visualisasi yang tepat merupakan hal yang sangat penting karena akan mempengaruhi definisi dari visual yang ditampilkan. Berikut adalah metode visualisasi dan fungsinya.

Metode Visualisasi	Fungsi
<i>Bar Chart</i>	Digunakan untuk membandingkan data yang memiliki kategori tertentu, dapat juga digunakan untuk menampilkan data yang kontinu dan diskontinu.
<i>Pie Chart</i>	Digunakan untuk memberikan rincian nilai kategori, merepresentasikan 100% dari kuantitas (jumlah dari nilai variabel pada seluruh kategori), dan ukuran dari setiap potongan adalah prosentase dari total yang direpresentasikan oleh kategori yang ditunjukkan potongan. <i>Piechart</i> biasanya digunakan untuk mempresentasikan frekuensi.
<i>Line Chart</i>	Digunakan untuk menampilkan informasi dalam rangkaian titik data yang dihubungkan dengan segmen

	garis lurus. <i>Line Chart</i> Digunakan untuk memvisualisasikan tren selama jangka waktu tertentu.
Tabel	Digunakan untuk mempresentasikan data dalam baris dan kolom dengan kategori tertentu, sehingga memudahkan dalam hal analisis. Tabel digunakan untuk menampilkan angka, tingkatan, proporsi, dan persentase kumulatif.

