BAB 2

TINJAUAN PUSTAKA

2.1 Profil Instansi

2.1.1 Sejarah Instansi

PT. Sanbe Farma Unit 3 (Sanbe Farma *Sterile Preparation Plant*) terletak di Jl. Industri Cimareme No.8 Padalarang, Bandung. Pabrik ini merupakan pabrik pertama di kawasan Asia Tenggara yang menggunakan teknologi dan peralatan dari Amerika dan Eropa dengan mengikuti standar HSA Singapura. Pabrik seluas 2,9 hektar ini menggunakan teknologi terbaru yang dikhususkan untuk produksi dan pengemasan sediaan steril. Pembangunan unit 3 dimulai tahun 2000 dan selesai pada Desember 2004. Pada bulan Januari-Juli 2005 dilakukan *start-up, commisioning* mesin, validasi, kemudian trial produksi dan uji stabilitas. Peluncuran produk baru unit 3 dilaksanakan pada Oktober 2005 dan peresmian unit 3 dilakukan pada 4 November 2006 oleh Menteri Kesehatan RI.

PT. Sanbe Farma unit 3 memproduksi produk steril dengan proses aseptis dan terminal sterilization. Produk *Small Volume Parenteral* (SVP) meliputi ampul, infuse botol, tetes mata, salep mata, *fat emulsion* dan *dry injection*. Produk *Large Volume Parenteral* (LVP) meliputi sediaan infus dalam kemasan *soft bag*, larutan hemodialisis, *water for irrigation*. Produk digunakan untuk mencukupi kebutuhan pasar dalam negeri dan ekspor.

P.T. Sanbe Farma telah disertifikasi oleh Badan POM dan Badan Internasional. Sertifikat yang dimiliki antara lain:

1. Sertifikat CPOB dan Badan POM untuk:

    a. Sediaan infus antibiotik

    b. Sediaan infus non antibiotik

    c. Sediaan *injection*

    d. Sediaan *dry injection*

    e. Sediaan *sterile eye drops*

    f. Sediaan *sterile eye ointment antibiotic*

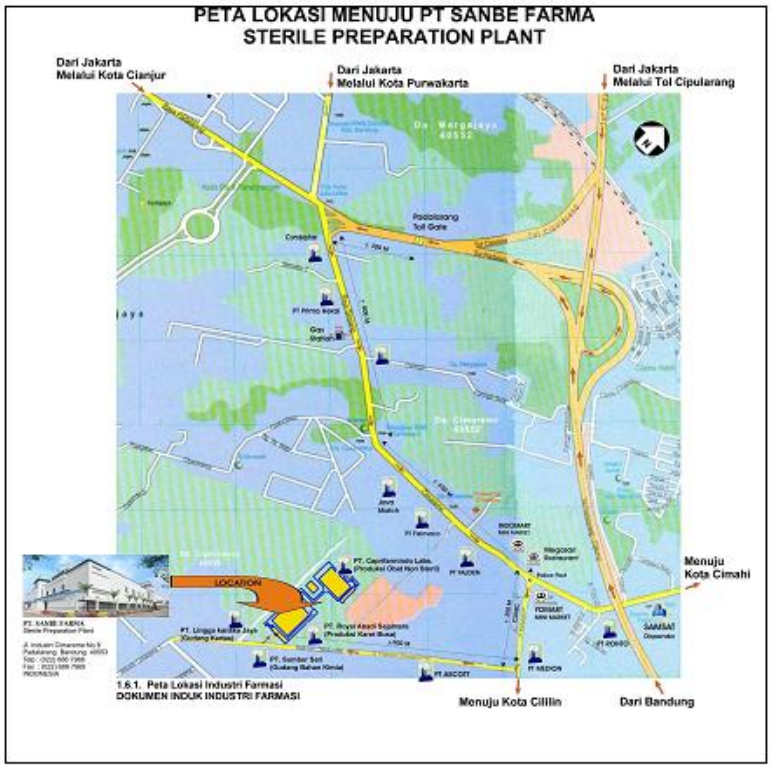
    g. Sediaan *sterile eye ointment non antibiotic*

2. Sertifikat dari HSA, Singapura

    .

2.1.2 Lokasi PT. Sanbe Farma Unit 3

PT. Sanbe Farma Unit 3 terletak di Jl. Industri Cimareme No.8, Padalarang, Bandung dengan luas bangunan 2,9 hektar dan luas tanah 3,1 hektar.



Gambar 2.1 Lokasi PT Sanbe Farma Unit 3

2.1.3 Visi dan Misi

a. Visi

Perusahaan yang berbasis ilmu pengetahuan dan teknologi yang inovatif sejalan dengan perkembangan ilmu pengetahuan dan teknologi di dunia.

b. Misi

Menjadi perusahaan farmasi terdepan di Indonesia dalam inovasi di bidang ilmu dan teknologi farmasi dan kedokteran.

2.1.4 Sarana dan Prasarana

Bangunan pabrik terdiri dari bangunan utama dan bangunan penunjang.

a. Bangunan Utama

1) Lantai Dasar

Lantai dasar bangunan utama terdiri dari ruang *water treatment system*, ruang produksi LVP dan SVP, ruang pengemasan LVP, ruang *In Process Control* (IPC), gudang bahan baku, gudang bahan pengemas primer, gudang obat jadi, mushola, laundry, klinik, loker karyawan serta lobby.

2) Lantai Satu

Gedung ini digunakan untuk kantor, pengolahan data, *Materials Management, Cost Accounting, Validation Office, Personnel and General Affairs*, departemen *Quality Assurance* (QA) serta *laboratorium Quality Control* (QC) yang dilengkapi dengan perpustakaan, ruang *retained sample*, *stability room*, ruang pengemasan SVP, gudang bahan pengemas sekunder, ruang *Information and Technology* (IT), *Centralized Documentation Area*, *Heating Ventilation Air Conditioner* (HVAC) serta kantin karyawan.

Ruang produksi di Sanbe *Sterile Preparations Plant* dicat dengan cat *epoxy* dengan tujuan agar mudah dibersihkan. Ruangan dibuat melengkung (tidak memiliki sudut) agar tidak menjadi tempat berkumpulnya debu. Ruangan dalam produksi dibagi menjadi 4 kelas yang berbeda. Perbedaan kelas ditentukan oleh: jumlah partikel di udara lingkungan, jumlah mikroba di udara lingkungan dan permukaan objek, jumlah pergantian udara (*air change*), kecepatan alir udara (*air flow*), pola

aliran udara, filter (jenis dan posisi), perbedaan tekanan antar ruang, temperatur dan kelembaban relatif (RH). Tujuan dari pembedaan kelas ini untuk menjaga sterilitas dari produk.

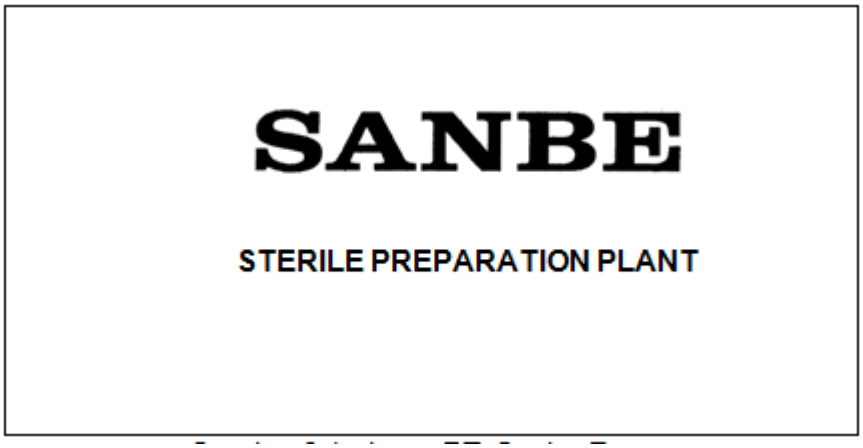
3) Lantai dua

Lantai 2 digunakan untuk ruang panel-panel sistem *HVAC*, listrik dan sistem suplai gas N_2.

b. Bangunan Penunjang

Bangunan penunjang terdiri dari bagian *engineering, energy center* (termasuk generator), *steam boiler, air compressor, chiller,* penyimpanan bahan kimia berbahaya, *nitrogen and* LPG *area*, penyimpanan bahan bakar, *workshop*, ruang penyimpanan *sparepart*, instalasi pengolahan limbah, *health care* serta pos keamanan.

2.1.5 Logo Instansi



Gambar 2.2 Logo Instansi

2.1.6 Struktur Organisasi

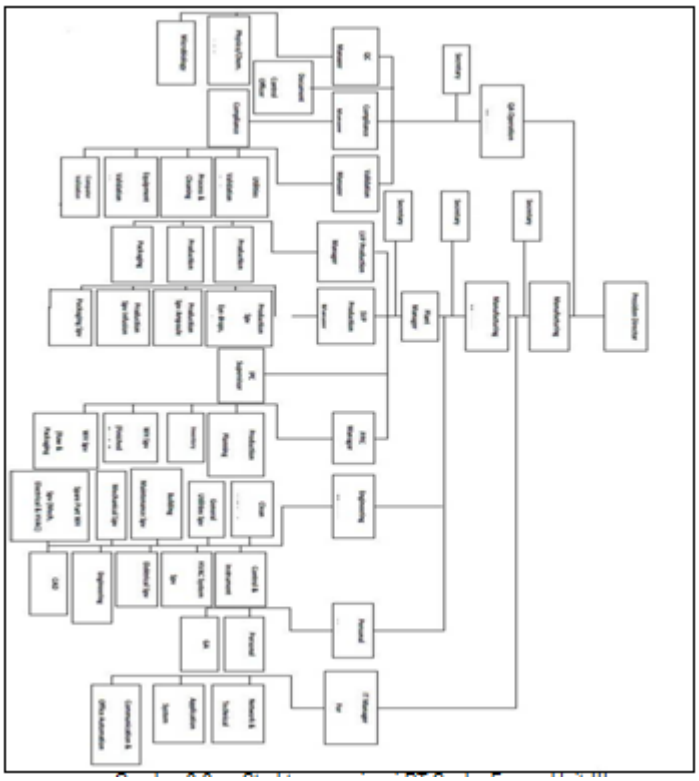
Pimpinan tertinggi P.T. Sanbe Farma adalah seorang *President director*. *President director* membawahi *Manufacturing director* dan *Head of Quality*. *Manufacturing director* bertugas memimpin perusahaan, menentukan struktur organisasi serta menetapkan peraturan dan perizinan terhadap segala sesuatu yang menyangkut kegiatan perusahaan. *Manufacturing director* membawahi *Plant manager,* IT *manager for manufacturing* dan *Technical project development manager.*

Head of Quality membawahi empat departemen yaitu QC *manager, QA manager (compliance), document control officer* dan *validation manager*. *QA manager (compliance)* bertugas menjamin bahwa sistem operasi QA benar-benar dilaksanakan, mengorganisasikan dan menjamin sistem kualitas dijalankan sesuai dengan prosedur yang telah ditetapkan, menyesuaikan seluruh pekerjaan berdasarkan cGMP, mengembangkan *corrective and preventive action and continuous improvement program*, menjamin produk sesuai dengan spesifikasi standar produksi yang ditentukan, meliputi kualitas dan kuantitas proses produksi serta material yang digunakan. *Plant manager* P.T. Sanbe Farma Unit 3 membawahi LVP *production manager,* SVP *production manager,* PPIC *manager, maintenance manager , personnel and general affair manager, product development officer* dan IPC *supervisor*. *Plant manager* bertanggung jawab pada keseluruhan operasional proses produksi.

Departemen dokumen kontrol berfungsi untuk memastikan semua kegiatan mulai dari perencanaan, pembangunan, pelaksanaan serta evaluasi

tercatat dan terdokumentasi sehingga tersedia informasi yang benar dan akurat untuk perusahaan.

BAGAN ORGANISASI PT. SANBE FARMA UNIT 3 CIMAREME



Gambar 2.3 Struktur organisasi PT. Sanbe Farma Unit 3

2.2 Landasan Teori

2.2.1 Pengertian Data

Data dan Informasi mempunyai keterkaitan antara satu dengan yang lainnya. Data adalah fakta – fakta atau pesan – pesan yang belum dievaluasi atau bahan mentah dari informasi yang melalui pengolahan tertentu dibentuk menjadi suatu informasi. Data merupakan kenyataan yang menggambarkan suatu kejadian – kejadian nyata.

“ *Menurut Steven Alter, data merupakan fakta,gambar atau suara yang mungkin atau tidak berhubungan atau berguna bagi tugas tertentu* ” [4].

“ *Menurut McLeod, data terdiri dari fakta-fakta dan angka yang secara relatif tidak berarti bagi pemakai. Sedangkan informasi adalah data yang sudah diproses atau data yang memiliki arti “*[4].

Data merupakan suatu bentuk keterangan-keterangan yang belum diolah atau dimanipulasi sehingga belum begitu berarti bagi sebagian pemakai. Sedangkan informasi merupakan data yang sudah di olah sehingga memiliki arti.

2.2.2 Basis Data (*database*)

2.2.2.1 Konsep Basis data (Database)

Basis data terdiri dari dua kata, basis dan data. Basis dapat diartikan kurang lebih sebagai markas, gudang atau tempat berkumpul [4]. Sedangkan data adalah repesentasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli dan lain – lain), barang dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasi lainnya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti [4] :

1. Himpunan kelompok data (arsip) saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembalidengan cepat dan mudah.
2. Kumpulan file / table / arsip saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Ada beberapa tahapan yang digunakan dalam proses mendesain suatu basis data yaitu:

1. Tentukan tujuan dari basis data yang akan dibuat.
2. Tentukan tabel – tabel yang sekiranya akan dperlukan.
3. Tentukan *field – field* (kolom – kolom) yang diperlukan oleh tabel.
4. Tentukan sebuah kolom yang bersifat unik untuk dijadikan sebagai kunci (*key*).
5. Tentukan relasi antar tabel.
6. Periksa ulang dan sempurnakan desain basis data.
7. Isi data dalam basis data dan buat objek – objek *database* yang sekiranya diperlukan.

2.2.2.2 Sistem Basis data (DBS)

Suatu sistem yang mengelolah data dan menyediakan data tersebut apabila dibutuhkan.

2.2.2.2.1 Komponen utama Sistem Basis data

1. Data yang disimpan dalam basis data
2. *Hardware : storage, processor, memory*
3. *Software : DBMS, Report-writer, design, arts*, dll

Pengguna :

a) Pengguna Awan (*Naïve User*)
b) Pengguna Biasa (*Casual User*)
c) Programmer
d) Administrator

2.2.3 Tujuan Database

Mereduksi redudansi yang akibatnya mengurangi inkonsistensi maka data dapat *dishare* antar aplikasi sehingga standarisasi data dapat dilakukan, batasan *security* dapat diterapkan, dimana pengelolaan *integritas* (Keterjaminan Akurasi) data dapat mudah dicapai dan dapat menyeimbangkan kebutuhan yang saling konflik [4].

2.2.3.1 Database Administrator

Database Adminstrator adalah orang yang memiliki kontrol utama terhadap keseluruhan sistem basis data (mencangkup data & program) yang mempunya fungsi yaitu [4]:

1. Pendefinisian skema
2. Pendefinisan struktur penyimpanan & metode akses
3. Modifikasi skema & organisasi fisik
4. Pemberian otorisasi bagi pengaksesan data
5. Mendefinisikan bagian basis data yang mana dapat diakses oleh seorang pemakai, termasuk operasi-operasi yang dapat dilakukan
6. Spesifikasi batasan integrasi

2.2.3.2 Bahasa/*Language* dalam Sistem Basis data

Bahasa yang digunakan di dalam basis data antara lain :

1. *Data Definition Language (DDL)*

Perintah-perintah yang digunakan oleh *database administrator* untuk mendefinisikan skema ke *DBMS*. Secara detil hal yang perlu dijabarkan pada *DBMS* :

a. Nama basis data
b. Nama seluruh berkas pada basis data
c. Nama rekaman dan medan
d. Enkripsi berkas, rekaman dan medan
e. Nama medan kunci
f. Nama Indeks dan medan yang menjadi indeks
g. Hal lain seperti ukuran basis data.

DDL juga digunakan untuk menciptakan, mengubah, dan menghapus basis data

2. *Data Manipulation Language (DML)*

DML adalah perintah-perintah yang digunakan untuk mengubah, memanipulasi dan mengambil data pada basis data. Tindakan seperti menghapus,mengubah,dan mengambil data menjadi bagian dari *DML*. *DML* dibagi atas 2 jenis :

a. Prosedural

   Prosedural menuntut pengguna menentukan data apa saja yang diperlukan dan bagaimana cara mendapatkannya

b. Nonprosedural

   Nonprosedural menuntut pengguna menentukan data apa yang diperlukan tetapi tidak perlu menyebutkan cara mendapatkannya.

2.2.4 Data Mart

Data mart adalah suatu bagian pada *data warehouse* yang mendukung pembuatan laporan dan analisis data pada suatu unit, bagian operasi pada suatu perusahaan. *Data mart* berisi informasi yang relevan bagi user 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. Data tersebut merupakan *asset* dari suatu perusahaan yang perlu di *manage* untuk keperluan *kompetitif* perusahaan. Banyak perusahaan yang telah menerapkan *DSS* (*Decision Support system)* yang telah ada di suatu data mart, sebuah *data mart* terintegrasi secara operasional dan sejarah data untuk aplikasi pengambilan keputusan.

Ada empat tugas yang bisa dilakukan dengan adanya *data mart*, keempat tugas tersebut yaitu [9]:

a. Pembuatan laporan

Pembuatan laporan merupakan salah satu kegunaan *data mart* yang paling umum dilakukan. Dengan menggunakan *query* sederhana didapatkan laporan per hari, per bulan, per tahun atau jangka waktu kapanpun yang diinginkan.

b. *On-Line Analytical Processing* (OLAP)

Dengan adanya *data mart*, semua informasi baik detail maupun hasil *summary* yang dibutuhkan dalam proses analisa mudah didapat. *OLAP* mendayagunakan konsep data multidimensi dan memungkinkan para pemakai menganalisa data sampai mendetail, tanpa mengetikkan satupun perintah *SQL*. Hal ini dimungkinkan karena pada konsep multidimensi, maka data yang berupa

fakta yang sama bisa dilihat dengan menggunakan fungsi yang berbeda. Fasilitas lain yang ada pada *software OLAP* adalah fasilitas *rool-up* dan *drill-down*. *Drill-down* adalah kemampuan untuk melihat detail dari suatu inforamsi dan *roll-up* adalah kebalikannya .

c. Data Mining

Data mining merupakan proses untuk menggali (*mining)* pengetahuan dan informasi baru dari data yang berjumlah banyak pada *data mart*, dengan menggunakan kecerdasan buatan (*Artificial Intelegence*), statistik dan matematika. *Data mining* merupakan teknologi yang diharapkan dapat menjembatani komunikasi antara data dan pelakunya.

d. Proses informasi *executive*

Data mart dapat membuat ringkasan informasi yang penting dengan tujuan membuat keputusan bisnis, tanpa harus menjelajahi keseluruhan data. Dengan menggunakan *data mart* segala laporan telah diringkas dan dapat pula mengetahui segala rinciannya secara lengkap, sehingga mempermudah proses pengambilan keputusan. Informasi dan data pada laporan *data mart* menjadi target informatif bagi pengguna.

2.2.4.1 *Multi-Dimensional Modeling*

Multi dimensional modeling adalah teknik untuk memvisualisasi model data sebagai suatu kumpulan dari ukuran yang dideskripsikan dengan aspek-aspek bisnis (Chuck Ballard, Dirk Herreman, Don Schau, Ronda Bell, Eunseang Kim, Ann Valencic; IBM Corp., 1998). Hal ini khususnya sangat berguna untuk

meringkas dan menyusun data dan memperlihatkan data untuk mendukung para analis data. *Dimensional modeling* memfokuskan pada data numerik, seperti harga, jumlah, berat, keseimbangan dan kejadian-kejadian.

Dimensional modeling mempunyai beberapa konsep :

1. Fact

Fact adalah suatu koleksi dari relasi data-data *items*, terdiri dari ukuran-ukuran dan konteks data. Setiap *fact* biasanya merepresentasikan sebuah bisnis *item*, suatu transaksi bisnis, atau sebuah kejadian yang dapat digunakan dalam analisis bisnis atau proses bisnis. Dalam *data mart*, *fact* di implementasikan dalam tabel dasar dimana semudah data *numeric* dan disimpan [5].

2. Dimensions

Dimensions adalah suatu koleksi dari anggota atau unit-unit data dengan tipe yang sama. Dalam sebuah diagram, suatu dimensi biasanya direpresentasikan dengan suatu *axis.* Dalam dimensional model, semua data menunjukan *fact table* yang diasosiasikan dengan satu dan hanya satu member sari setiap *multiple dimensions.* Jadi dimensi menunjukan latar belakang kontekstual dari *fact.* Banyak proses analisis yang digunakan untuk menghitung (*quatify*) dampak dari dimensi pada *fact* [5].

Dimensi adalah parameter dari apa yang ingin lakukan dalam *On-Line Analytical Processing (OLAP)* [6]. Sebagai contoh, dalam suatu *database* untuk menganalisa semua penjualan dari produk, dimensi berikut ini harus ada :

1) Waktu
2) Lokasi

3) Pembeli

4) Penjual

Skenario seperti kejadian, pembiayaan atau angka estimasi

Dimensi biasanya juga dapat dipetakan bukan angka numerik, *entity* yang bersifat informatif seperti merek atau karyawan.

Dimension member : suatu dimensi berisi banyak anggota-anggota. Suatu anggota dimensi adalah nama pembeda atau *indetifier* yang digunakan untuk membedakan posisi suatu data item. Sebagai contoh, semua bulan, empat bulanan dan tahunan membuat dimensi waktu. Dan semua kota, wilayah dan negara merupakan dimensi geografi [6].

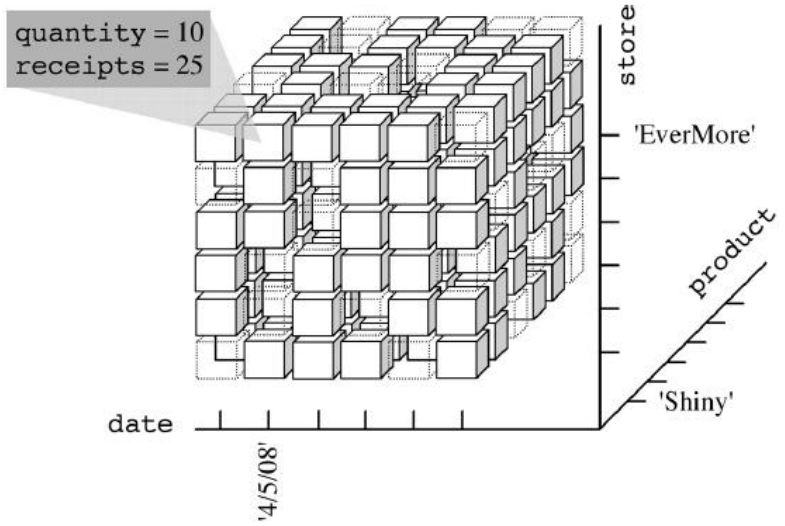
Dimension Hierarchies : dapat mengatur anggota dari suatu dimensi kedalam satu atau banyak hirarki. Setiap hirarki dapat juga mempunyai banyak level hirarki. Setiap member dari dimensi tidak boleh dilokasikan kedalam satu struktur hirarki [6].

3. Measures

Suatu *measures* (ukuran) adalah suatu besaran (angka numerik) atribut dari sebuah *fact*, yang menunjukan *performance* atau *behavior* (tingkah laku) dari bisnis secara relatif pada suatu dimensi [6]. Angka atau nomor yang ditunjukan disebut dengan *variable*. Sebagai contoh ukuran dari penjualan dalam bentuk uang, besarnya penjualan, jumlah pengadaan, biaya pengadaan, banyaknya transaksi dan lainnya. Suatu ukuran dijelaskan dengan kombinasi dari member dari suatu dimensi dan diletakkan dalam *fact.*

2.2.4.2 Visualisasi Dari Suatu Dimensional Model

Cara yang paling populer dalam memvisualisasikan suatu model dimensional adalah dengan menggambarkan sebuah *cube* data. Contohya dapat dilihat pada (Gambar 2.4) dapat menggambarkan tiga model dimensional menggunakan sebuah kubus. Biasanya suatu dimensional model terdiri dari lebih tiga dimensi dan digambarkan sebagai suatu *hypercube*, akan tetapi *hypercube* sulit untuk divisualisasikan, jadi sebuah kubus lebih biasa digunakan [6]. Gambar berikut merupakan *measurement* adalah volume dari produksi, yang mana dijelaskan dengan kombinasi dari tiga dimensi yaitu lokasi produk dan waktu. Dimensi dari lokasi dan dimensi produk mempunyai dua level hiraki. Misalnya dimensi mempunyai *level region* dan *plant*.

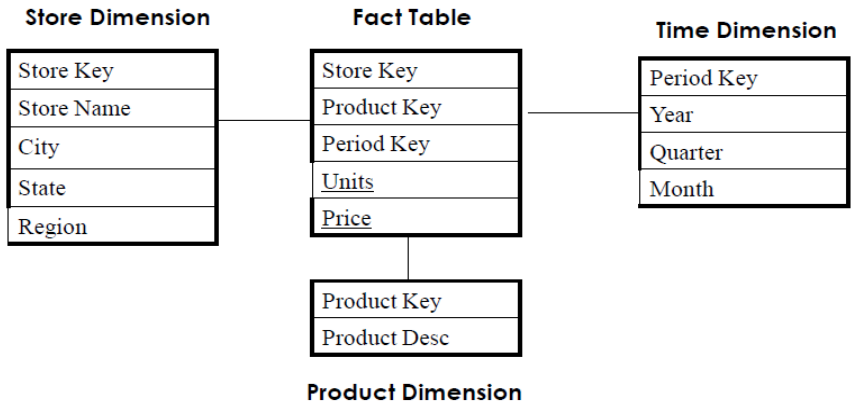


Gambar 2.4 kubus suatu visualisasi dari dimensional model.

Terdapat tiga konsep yang dapat memodelkan lebih jelas dalam penentuan *fact table* dimensi dan *measure*, yaitu :

1. Model *star*

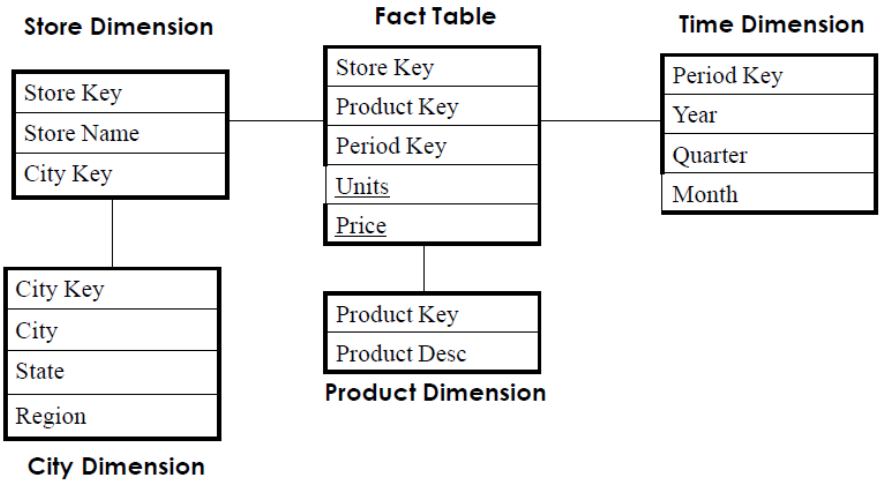
   Model *star* adalah model dasar dari pemodelan multidimensi yang dapat di lihat pada gambar 2.5. Model star memiliki satu tabel induk yang dinamakan *table fact* dan kumpulan dari tabel-tabel kecil yang disebut tabel dimensi, disusun dalam pola-pola melingkar mengelilingi *table fact* [9].



Gambar 2.5 Model *Star*

2. Model *SnowFlake*

   Selain model *star* dalam menampilkan desain sebuah *data mart* juga dapat menggunakan model *Snowflake* sebagai contoh dapat dilihat pada gambar 2.6. Dalam sebuah model *snowflake* terdapat sebuah *fact table* yang dikelilingi oleh beberapa *dimension table.* Namun *dimension table* itu sendiri dapat seolah-olah berupa sebuah *fact table* lain yang juga memiliki dimensinya sendiri [9].

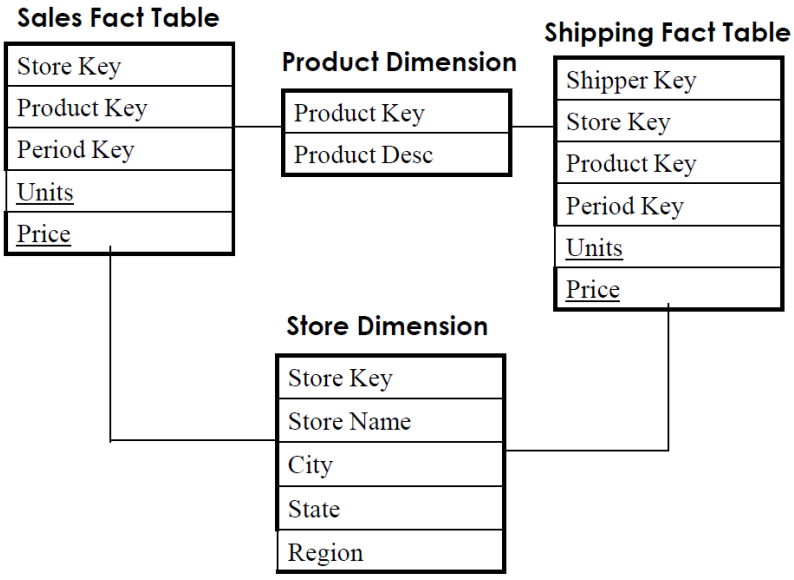


Gambar 2.6 Model *SnowFlake*

Pada model *snowflake* diatas dapat dilihat bahwa tiap-tiap tabel dimensi berhubungan dengan dimensi yang lain seolah-olah tabel dimensi tersebut merupakan tabel fakta. Hal ini dapat terjadi karena pada model *snowflake* ini telah dilakukan normalisasi. Normalisasi ini tidak terdapat pada model *star*.

3. Model *Constellation.*

   Beberapa *fact tables* berbagi tabel dimensi, sebagai contoh dapat di lihat pada gambar 2.7. Model *constellation* ditampilkan sebagai koleksi dari kumpulan skema bintang yang sering disebut sebagai skema *galaxy* [9].



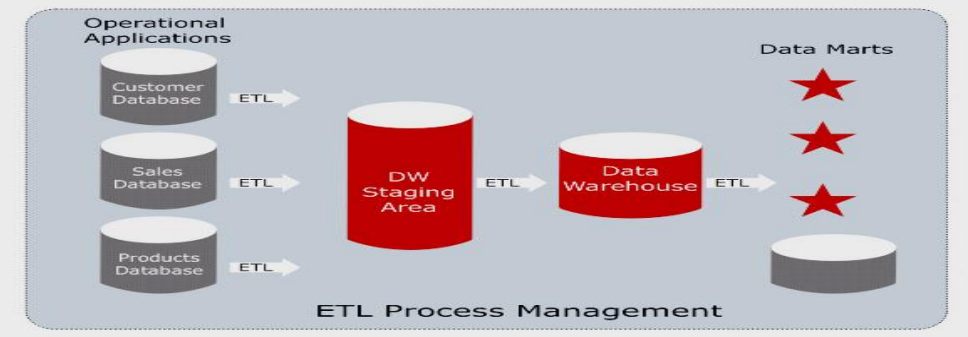
Gambar 2.7 Model *Constellation*

2.2.5 ETL (*Extract, Transform, Loading*)

Extraction merupakan proses untuk mengidentifikasi seluruh sumber data yang relevan dan mengambil data dari sumber data tersebut.Untuk lebih jelasnya dapat dilihat pada gambar 2.8. Penentuan pendekatan yang digunakan pada proses ekstrasi sangat terkait dengan analisis bisnis proses, pendefinisian area subjek , serta desain logik/fisik *data mart* [9].

Transformation merupakan proses yang mempunyai peran dalam melakukan perubahan dan integrasi skema serta struktur yang berbeda-beda kedalam skema dan struktur yang terdefinisi dalam *data mart* [9].

Loading merupakan proses pemindahan data secara fisik dan sistem operasional kedalam *data mart* [9].



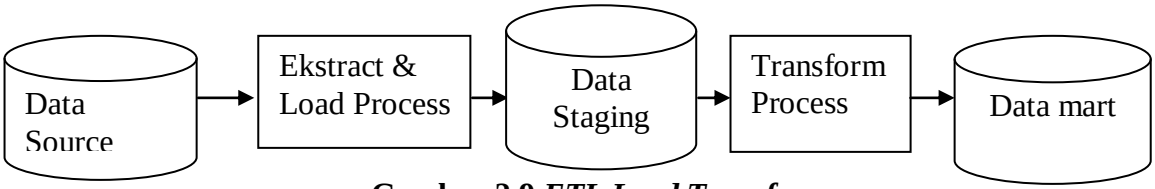
Gambar 2.8 *ETL Process Management*

Proses *ETL* meliputi 3 tahapan pemodelan yaitu model data sumber, model area bisnis dan model *data mart*. Proses *ETL* berawal dari data sumber yang berasal dari basis data operasional dan berakhir dngan proses *loading* data pada *data mart*. Proses yang dilakukan meliputi proses *Ekstraksi* dari data sumber yang akan di tempatkan pada tabel *staging* baik tabel *staging* final maupun *tabel staging temporer*.

2.2.5.1 Metode ETL

Didasarkan penggunaan tabel *staging* sebagai tabel *temporary* dari basis data sumber untuk loading pada data mart metode yang digunakan yaitu dengan pendekatan *Load Transform* yaitu menggunakan tahapan sementara dalam operasi *ETL* yang disebut dengan *staging* area/tabel [10], untuk lebih jelasnya dapat dilihat pada gambar 2.9.

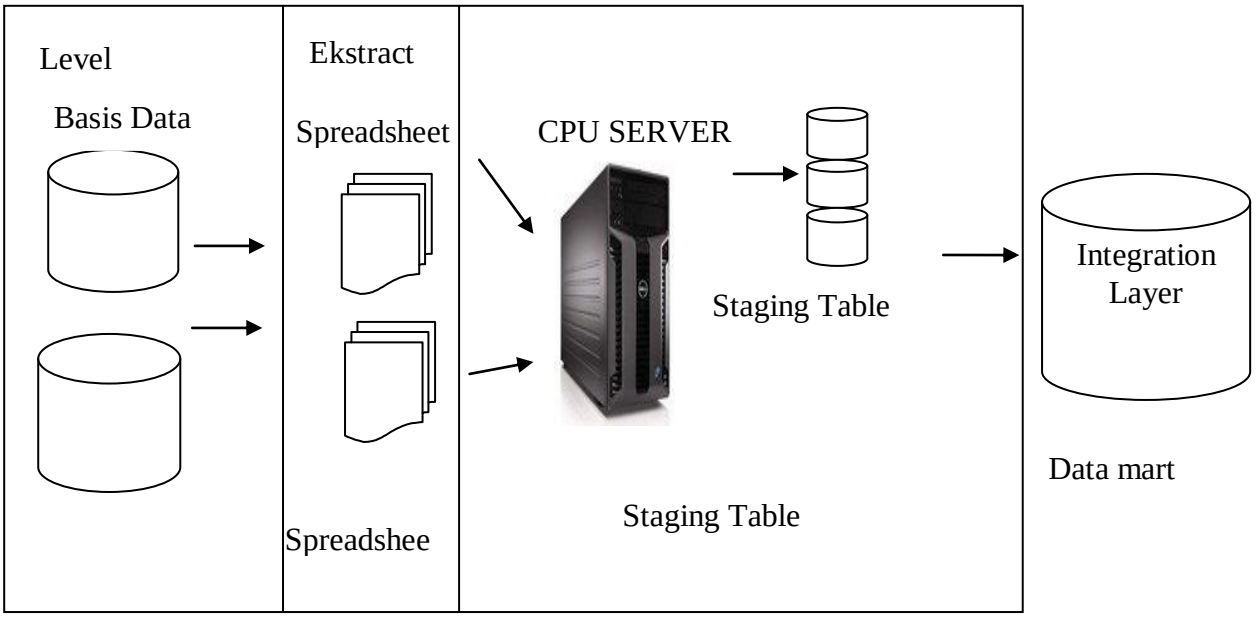
yaitu *Ekstraksi*, *Transformasi*, serta *Loading* dilakukan pemisahan secara fisik sehingga dapat dikerjakan secara terpisah/parallel.



Gambar 2.9 *ETL Load Transform*

2.2.5.2 Gambaran Proses ETL

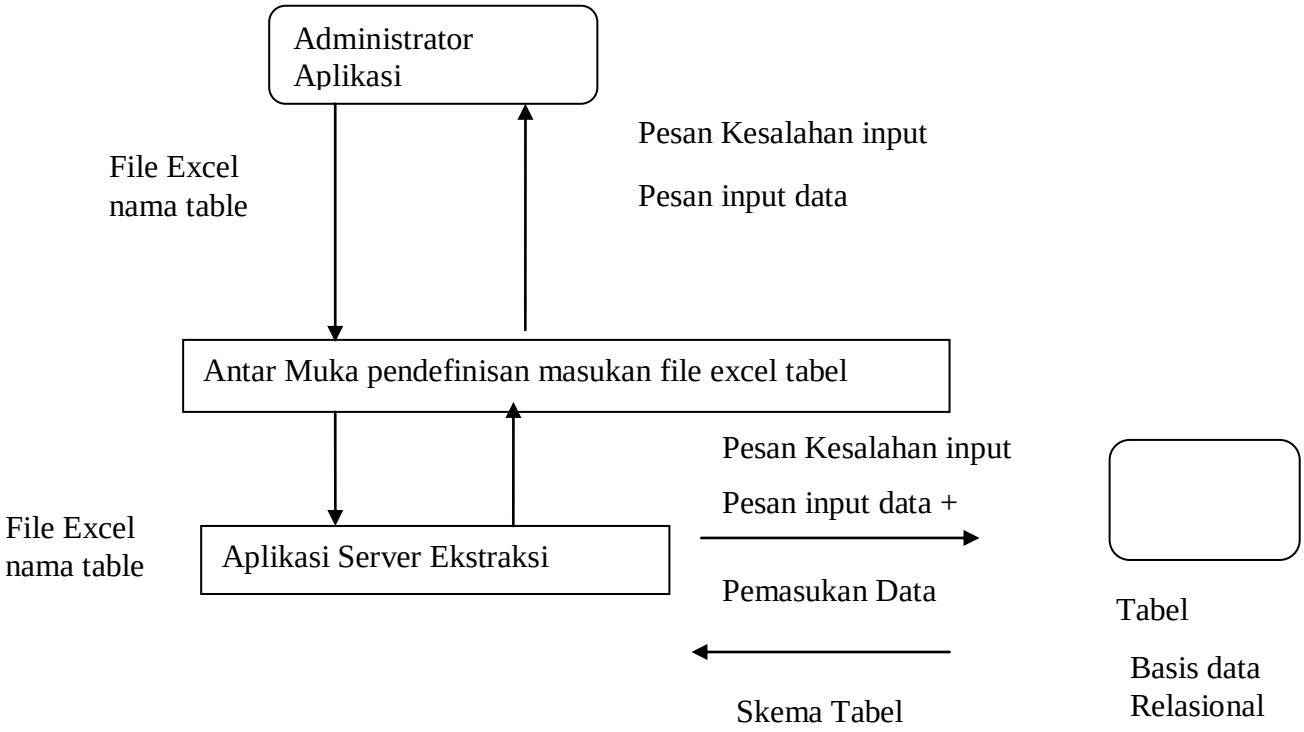
Proses pengambilan data dari sumber data merupakan sekumpulan proses yang lebih dikenal dengan *Extract*, Transfom, Load data ke system target untuk lebih jelasnya dapat di lihat pada gambar 2.10. ETL dapat digunakan untk mengambil data dari berbagai sumber [10].



Gambar 2.10 Gambaran *Process ETL*

Proses *ekstraksi* dilakukan dengan menggunakan pendekatan logic *Full Ekstraction* dan *incremental* ekstraksi disertai dengan besaran data yang di input. Hal ini dilakukan dengan menggunakan *counter* terhadap data yang di input menjadi paket-paket input dengan besaran tertentu untuk lebih jelasnya dapat di lihat pada gambar 2.11 [10]. Pengembangan selanjutnya dengan menggunakan metode *load-transform* yaitu dengan menggunakan konsep *staging* dalam proses transformasi multistage *transformation*, apabila proses transformation selesai, data yang telah bersih akan di proses pada lingkungan target yaitu *integrasi layer*.

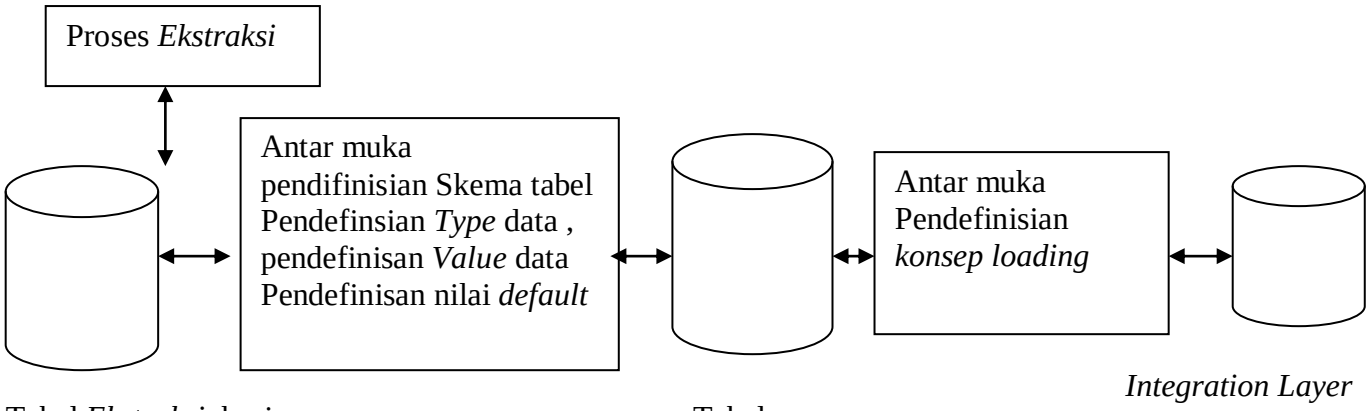
Ilustrasi Arsitektur Proses Ekstraksi



Gambar 2.11 Arsitektur *Ekstraction ETL*

2.2.5.3 Arsitektur Transformasi & Loading

Proses perpindahan data dari sumbernya (*source*) ke *data mart* untuk lebih jelasnya dapat di lihat pada gambar 2.12. Proses ini proses yang menggunakan banyak waktu dalam proyek *data mart* [10].



Gambar 2.12 Arsitektur *Transform&Loading*

ETL merupakan proses yang sangat penting dalam *data mart* [10], dengan *ETL* inilah data dari operational dapat dimasukkan ke dalam *data mart*. *ETL* dapat digunakan untuk mengintegrasikan data dengan sistem yang sudah ada sebelumnya.

2.2.6 Diagram Konteks

Diagram konteks merupakan alat pemodelan atau suatu diagram yang menggambarkan sistem berbasis komputer yang dirancang secara global dan merupakan suatu diagram alir data tingkat atas, dimana didalam diagram konteks ini menggambarkan seluruh jaringan, baik masukan maupun sebuah keluaran sebuah sistem.

Diagram konteks terdiri dari sebuah simbol proses tunggal yang menggambarkan sebuah sistem dan menunjukan data aliran utama untuk dan dari terminator. Ddiagram ini merupakan dasar yang digunakan untuk menentukan aliran data yang mengalir menuju sistem (*input system*) dan keluar dari sistem (*output system*), yang meliputi objek berupa kesatuan luar (*eksternal entity*).

Diagram konteks dapat mendefinisikan jangkauan proses penurunan sistem informasi yaitu menentukan apa yang menjadi bagian dari sistem informasi dan apa yang tidak menjadi bagian sistem informasi.

2.2.7 *Data Flow Diagram (DFD)*

Diagram aliran data atau *DFD* merupakan suatu bagan alir data yang digunakan untuk menjelaskan data yang ditransformasikan oleh suatu proses pada suatu sistem dengan menekankan pada fungsi – fungsi yang ada dalam sistem, cara menggunakan informasi yang tersimpan dan pemindahan informasi antar fungsi di dalam sistem.

Beberapa simbol yang digunakan dalam *Data Flow Diagram (DFD)* antara lain:

1. **Proses (*Process*)**

   Proses adalah simbol utama data *flow diagram*. Proses dilambangkan dengan lingkaran, dimana proses ini menunjukan bagian dari sistem yang mengubah satu atau lebih *input* dan *output*. Nama proses dituliskan dengan satu kata, singkatan atau kalimat sederhana.

2. **Aliran Dta (*Flow*)**

   Aliran Data digambarkan dengan tanda panah dari proses. Aliran data juga digunakan untuk menunjukan bagian – bagian informasi dari satu bagian ke bagian lain. Pembagian nama untuk aliran ini menunjukan sebuah arti untuk sebuah aliran. Untuk kebanyakan sistem yang dibuat,

aliran data sebenarnya menggambarkan data yakni angka, huruf, pesan, dan macam – macam informasi lainnya.

3. **Simpanan Data (*Storage*)**

   Simpanan data digunakan sebagai penyimpanan bagi paket – paket data. Notasi penyimpanan data digambarkan dengan garis horizontal yang paralel. Simpanan data merupakan simpanan data dari data yang berupa suatu file atau *database* di sistem komputer ataupun berupa arsip atau catatan manual. Nama dari simpanan data menunjukan nama filenya.

4. **Kesatuan luar (*External Entity*)**

   Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang. Organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau output dari sistem.

2.2.8 Kamus Data

Kamus data adalah suatu daftar atau kamus dari seluruh elemen-elemen data yang diperlukan oleh suatu sistem. Kamus data dibuat berdasarkan arus data yang terdapat pada DFD. Arus data dalam DFD bersifat global sehingga hanya dapat ditunjukkan nama arus datanya saja dan keterangan lebih lanjut mengenai struktur arus data dapat dilihat dari kamus data [1].

2.2.9 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram adalah suatu model atau teknik pendekatan yang dapat menyatakan atau menggambarkan hubungan *entity* didalam sebuah sistem, dimana hubungan tersebut dinyatakan sebagai *one-to-one*, *one-to-many* dan *many-to-many*. [1]

Entity relationship bertujuan untuk menentukan objek-objek data (*Entity*) dan *relationship* (hubungan) yang ada pada objek-objek tersebut.

Komponen *Entity Relationship Diagram* :

1. *Entity*

Entity adalah suatu yang dapat dibedakan atau diidentifikasikan secara unik, objek disini sifatnya berupa orang, tempat, peristiwa atau konsep yang informasinnya direkam.

2. Atribut

Atribut adalah kumpulan dari beberapa elemen data yang membentuk suatu entity. Atribut merupakan ciri sebutan atau karakteristik yang dapat mewakili suatu simbol yang digunakan untuk menggambarkan atribut pada *entity relationship diagram.*

3. *Relationship*

Relationship adalah relasi atau hubungan yang terjadi diantara beberapa *entity*. Simbol yang digambarakan untuk menggambarkan *relationship* pada teknik *entity relationship diagram* adalah berupa belah ketupat.

Setiap *relationship* mempunyai derajat *relationship*, yaitu tingkat atau hubungan yang terdiri dari tiga derajat, yaitu :

a. *Unary* (derajat 1)

Suatu *relationship* yang dihubungkan dengan suatu *entity* set, dimana penghubungnya ada dua.

b. *Binary* (derajat 2)

Relationship derajat dua adalah *relationship* yang biasa terjadi yaitu dua *entity* set dihubungkan dengan satu *entity relationship*.

c. *Ternary* (derajat 3)

Relationship derajat tiga.

Relationship diagram juga mempunyai *Cardinalitas* yaitu batasan pada jumlah *entity* yang terdiri dari:

1. Relasi satu ke satu (*one-to-one*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa satu berbanding satu.

2. Relasi satu ke banyak (*one-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak atau banyak berbanding satu.

3. Relasi Banyak ke Banyak (*many-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak.

2.2.10 Pengertian Internet

Istilah Internet berasal dari bahasa latin, *inter*, yang berarti ”*antara*”. Secara kata per kata Internet berarti jaringan antara atau penghubung. Memang

itulah fungsinya, Internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain dengan sedemikian rupa, sehingga mereka dapat berkomunikasi.

Internet adalah jaringan komputer yang saling terhubung ke seluruh dunia tanpa mengenal batas teritorial, hukum dan budaya. Secara fisik dianalogikan sebagai jaring laba-laba (*The Web*) yang menyelimuti bola dunia dan terdiri dari titik-titik (*node*) yang saling berhubungan [2].

Node bisa berupa komputer, jaringan lokal atau peralatan komunikasi, sedangkan garis penghubung antar simpul disebut tulang punggung (*back-bone*) yaitu media *terestrial* (kabel, serat optic, microwave, radio link) maupun satelit. *Node* terdiri dari pusat informasi dan database, peralatan komputer dan perangkat interkoneksi jaringan serta peralatan yang dipakai pengguna untuk mencari, menempatkan dan bertukar informasi di Internet.

Menurut Lani Sidharta (1996) walaupun secara fisik Internet adalah interkoneksi antar jaringan komputer namun secara umum Internet harus dipandang sebagai suber informasi. Isi Internet adalah informasi, dapat dibayangkan sebagai suatu database atau perpustakaan multimedia yang sangat besar dan lengkap. Bahkan Internet dipandang sebagai dunia dalam bentuk lain (maya) karena hamper seluruh aspek kehidupan di dunia nyata ada di Internet seperti bisnis, hiburan, olah raga, politik dan lain sebagainya.

Salah satu isu terpenting salah satu isu terpenting di Internet adalah penerapan *standar* komputerisasi terbuka (*open computing standart*). Karena *internetworking* dan Internet mengintegrasikan semua sistem, jenis dan tipe

komputer yang ada di dunia, maka harus ada *standar* yang menjamin komputer dapat saling berbicara satu sama lain dalam bahasa yang sama. Menurut Drew Heywood (1966): *standar* bahasa komputer universal telah dikembangkan sejak 1969, terdiri dari serangkaian protocol komunikasi disebut *Transfer Control Protocol (TCP)* yang bertugas mengendalikan transmisi paket data, koreksi kesalahan dan kompresi data dan Internet Protocol (IP) yang bertugas sebagai pengenal (*identifier*) dan pengantar paket data ke alamat yang dituju.

Protokol TCP / IP menyatukan bahasa dan kode berbagai komputer di dunia, sehingga menjadi *standar* utama jaringan komputer. TCP / IP berkembang cepat dan kaya fasilitas karena bersifat terbuka dan bebas digunakan, oleh siapa pun. Oleh karena itu, hingga saat ini jaringan-jaringan tersebut saling terhubung di seluruh belahan dunia.

Seluruh jaringa yang tersebar di setiap penjuru dunia sudah terkondisikan dengan internet. Lalu lintas data Internet ke seluruh Amerika dan penjuru dunia saat ini disebarkan oleh jaringan pusat yang disebut VBNS (*Very High Speed Backbone Network Service*). Adapun seluruh biaya yang dikeluarkan di biayai oleh badal Ilmu Pengetahuan Nasional Amerika Serikat. Dari jaringa itu, komputer-komputer yang terhubung dengan internet akan berkomunikasi dan saling mengirimkan data sehingga internet dapat dinikmati oleh seluruh umat manusia untuk berkomunikasi melalui komuputer baik di rumah maupun di kantor.

Dengan demikian, internet dapat didefinisikan dengan hubungan berbagai jenis komputer dan jaringan di dunia, yang berbeda sistem operasi maupun

aplikasinya di mana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protocol *standar* dalam berkomunikasi yaitu protocol TCP / IP. Internet berfungsi sebagai media komunikasi dan informasi modern.

2.2.11 Pengertian *World Wide Web* (WWW)

World Wide Web adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen *web* disebut *web page* dan *link* dalam *web* membuat *user* bisa pindah dari satu *page* ke *page* lain (*hyper text*), baik antar *page* yang disimpan dalam *server* yang sama atau *server* di seluruh dunia. *Pages* diakses dan dibaca melalui *web browser* seperti *netscape navigator* atau *internet explorer.*

World Wide Web sering disingkat WWW atau *web* saja, yaitu sebuah sistem dimana informasi dapat kita nikmati dalam bentuk teks, gambar, suara, dan lain-lain serta ikut dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam bentuk HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam bentuk format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). WWW dijalankan dalam *server* yang disebut HTTPD.

2.2.12 Software Pendukung

2.2.12.1 *Hypertext Markup Language* (HTML)

Dokumen HTML adalah file teks regular (disebut juga ASCII) yang diciptakan dengan menggunakan editor teks (misalnya, Emacs dalam Unix, Gnotepad pada Linux, atau Notepad pada Windows) atau dengan menggunakan word processor, tetapi harus disimpan dengan format *"text only with line breaks".* Sesuai dengan namanya, bahasa ini menggunakan tanda (MarkUp) untuk menandai perintah-perintahnya. Saat ini banyak sekali aplikasi-aplikasi yang dapat digunakan untuk membuat web page, seperti Microsof Frontpage, Adobe GoLive, Macromedia Dreamweaver, Ultradev dan lain-lainnya. Namun untuk seorang web develover kemampuan dasar perintah HTML tetap dibutuhkan.

Homepage dibangun oleh sebuah file yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama browser. Oleh browser, kode dan informasi akan disajikan sebagai sebuah halaman web yang kita kenal sebagai homepage. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti browser. Aturan tersebut dikenal dengan "bahasa pengkodean".

Untuk membuat homepage website, bahasa pengkodean yang digunakan adalah HTML. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau style dan teks yang ditandai. Simbol Mark Up yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>),

keduanya dinamai tag. Sebagai contoh akan ditampilkan teks yang tercetak tebal, maka mark up yang digunakan adalah sebagai berikut:

<B> Teks ini bercetak tebal </B>

Untuk menandai bahwa sebuah file teks merupakan file HTML, maka ciri yang terlihat jelas adalah ekstensi filenya yaitu .htm atau .html. Namun labih jauh dari pada itu didalam file tersebut harus mengandung struktur sebagai berikut:

<HTML>

…….

…….

</HTML>

Tag <HTML> harus diletakan pada bagian awal dan tag </HTML> harus diletakan pada bagian terakhir dari suatu file HTML. Tag-tag HTML tidak bersifat case sensive artinya penggunaan huruf besar maupun huruf kecil tidak menjadi masalah. Jadi tag <HTML> akan sama dengan tag <html>.

2.2.12.2 *Hypertext Preprocessor* (PHP)

PHP (Personal Home Page Tools) adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

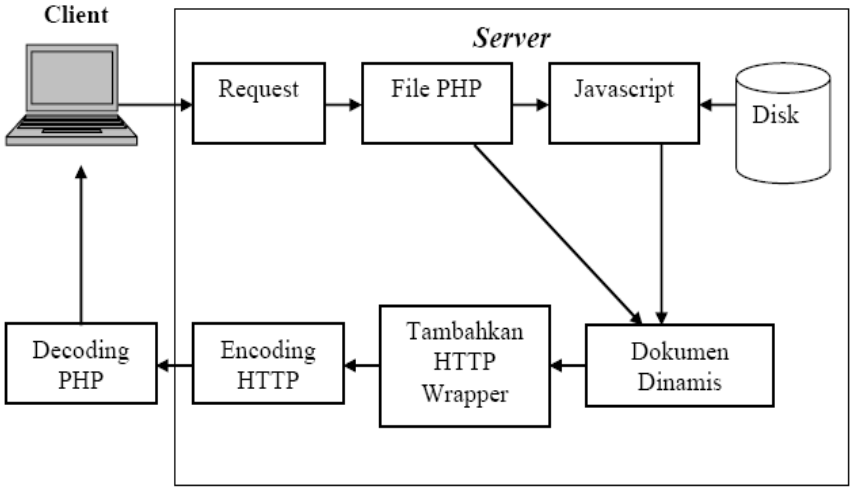
Bahasa ini memungkinkan para pembuat aplikasi web yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang di hasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah

ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl, awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang di buat dengan menggunakan PHP.

Kode program PHP menyatu dengan tag-tag HTML dalam satu file. Kode PHP diawali dengan tag <? Dan ditutup dengan tag ?>. File yang berisi tag HTML dan kode PHP ini diberi ekstensi .php atau ekstensi lainnya yang ditetapkan pada *Apache/web server*. Berdasarkan ektensi ini, pada saat file diakses, server akan tahu bahwa file ini mengandung kode PHP. Server akan menerjemahkan kode ini dan menghasilkan output dalam bentuk tag HTML yang akan dikirim ke *browser client* yang mengakses file tersebut. Contohnya dapat dituliskan seperti ini :

```
<?
echo (“<html>
<head>
<title>Pemrograman PHP</title>
</head>
<body>
<center>Selamat menggunakan PHP</center>
</body>
</html>”);
?>
```

Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah file yang terpisah seperti pada CGI. *Browser web* mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web, server web* memeriksa isi file dan menentukan apakah ada kode didalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan dimasukkan kedalam dokumen yang sama. *Server web* bekerja secara langsung terhadap file yang bersangkutan tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode di eksekusi di *server* (oleh karena itu disebut *server-side script*). Lihat gambar 2.13 dibawah ini untuk melihat kerja dari PHP dan Apache.



Gambar 2.13 Prinsip Kerja PHP dan *Apache*

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* biasa berkonsentrasi langsung pada penampilan webnya.
2. Tidak ada proses *compiling* dan *linking.*

3. Berorientasi objek *(object Oriented).*
4. Sintaksis pemrogramannya udah dipelajari, sangat menyerupai *C* dan *Perl.*
5. Integrasi yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. Database yang didukung oleh PHP : *Oralce, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePRo, Velocis, Informic, dBase, UNIX dbm.*

2.2.12.3 *Cascading Style Sheets* (CSS)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C).

CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis-suara atau pembaca layar), dan juga alat pembaca *braille*. Halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.2.12.4 *JavaScript*

JavaScript adalah bahasa *script* yang digunakan pada halaman HTML. *JavaScript* adalah varian dari *java* yang sangat terkenal dalam lingkungan pemrograman *web*. *JavaScript* mempunyai keunggulan pada sisi *client*, maupun diakses lebih cepat dari *script server*. Tetapi kelemahannya pada *script* ini belum tentu mampu diproses oleh *browser client* tergantung kompetibilitas *browser* tersebut. Cara untuk menjalankan j*avascript* hanya dengan *javascript-enabled* yaitu *browser* yang mampu menjalankan *javascript* seperti *Netscape Navigator* (versi 2.0 keatas) atau *internet ekplorer* (MSIE-versi 3.0 ke atas).

Beberapa hal mengenai JavaScript :

1) *JavaScript* didesain untuk menambah interaktif suatu *web*.
2) *JavaScript* merupakan sebuah bahasa *scripting* dengan bahasa pemrograman yang ringan.
3) *JavaScript* biasanya disisipkan (*embedded*) dalam halaman HTML.
4) *JavaScript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi.

Ada beberapa cara untuk menuliskan JavaScript yaitu :

a) *Script* pada bagian *head*

Skrip ini dieksekusi ketika dipanggil (biasanya berbentuk *function*) atau dipanggil berdasarkan *event* tertentu. Peletakan *script* di *head* akan menjamin *script* di-*load* terlebih dahulu sebelum digunakan.

```
<html>
<head>
<script type=”text/javascript”>
Function message ()
{
Alert (“Alert ini dipanggil saat even dijalankan”);
}
<script>
</head>
</html>
```

b) *Script* pada *Body*

Skrip akan dieksekusi ketika halaman di-*load* sampai dibagian <body> dan ketika ditempatkan berarti antara isi dan *javacript* dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
```

```
<script type=”text/javascript”>
Document.write(“Pesan ini akan tampil”);
</script>
</body>
</html>
```

c) *External JavaScript*

Untuk memudahkan menulis *script* yang berulang-ulang di setiap halaman maka *JavaScript* dapat ditulis di *file* secara *ekternal*. Jadi, antara HTML dan *JavaScript* dipisah, kemudian *file* tersebut dipanggil dari dokumen HTML. *File JavaScript* tersebut disimpan dengan ektensi .js.

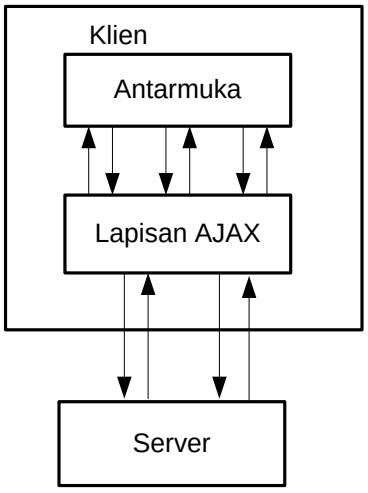
```
<html>
<head>
</head>
<body>
<script src=”xxx.js”>
</script>
</body>
</html>
```

2.2.12.5 *Asynchronous JavaScript and XML* (AJAX)

AJAX (*Asynchronous JavaScript and XML*) sesungguhnya bukanlah suatu bahasa pemrogarman, melainkan hanya suatu teknik yang memungkinkan untuk membuat aplikasi *web* yang interaktif. Dengan menggunakan AJAX, aplikasi *web*

dapat berinteraksi dengan server di latar belakang sehingga tidak mempengaruhi halaman *web* secara keseluruhan. Sebagai gambaran, Anda mungkin sering mendaftarkan diri pada suatu situs *web* sebagai anggota baru. Setelah Anda selesai mengetikan *user ID* saat registrasi, sistem bisa memberitahukan bahwa nama yang Anda pakai sudah digunakan oleh orang lain atau belum.

cBagaimana aplikasi *web* ini bisa tahu hal itu padahal Anda belum mengklik tombol apapun dalam halaman *web*. Rahasianya terletak pada pemrosesan di latar belakang yang dilakukan oleh AJAX. Gambar 2.14 Menunjukan aplikasi *web* yang berbasis AJAX.



Gambar 2.14 Model Aplikasi Web dengan AJAX

Nampak bahwa pada aplikasi *web* yang berbasis AJAX terdapat bagian yang disebut lapisan AJAX (atau terkadang disebut mesin AJAX). Lapisan inilah yang mengatur komunikasi antara *client* dan *server*.

Butir penting yang perlu diperhatikan dalam lapisan AJAX :

1. Lapisan AJAX meminta layanan ke *server* atau tidak, tergantung pada jenis layanan antarmuka. Sebagai contoh, untuk validasi terhadap objek formulir

tidak perlu melakukan permintaan ke *server*. Cukup ditangani di sisi *client*. Namun untuk pemeriksaan keabsahan *password*, mau tidak mau harus dilayangkan ke *server* karena harus berhubungan dengan database.

2. Sekalipun penambahan lapisan AJAX berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), pada kenyataanya yang terjadi adalah sebaliknya. Hal ini karena permintaan yang dilakukan oleh lapisan AJAX menghasilkan informasi yang jauh lebih sedikit dibanding kalau yang diminta adalah keseluruhan halaman.
3. Lapisan AJAX bekerja dengan pendekatan asinkron. Asinkron berarti bahwa *client* bisa meminta layanan dari *server* dan tidak perlu menunggu sampai *server* melayaninya. Dengan demikian, saat informasi dari *server* belum didapat, layanan kepada pemakai tetap bisa dilakukan.
4. Komunikasi antara *client* dan *server* tidak perlu mengubah keseluruhan antarmuka dalam halaman *web*, tetapi cukup hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi *web* yang tidak menggunakan AJAX.

AJAX pertama kali ditemukakan pada tahun 2005 oleh Jesse James Garret, presiden dan pendiri perusahaan Adaptive Path. Menurut beliau, nama AJAX diberikan untuk memudahkan dalam berkomunikasi dengan klien daripada kalau menyebutnya "*Asynchronous JavaScript + CSS + DOM + XMLHttp Request*". Teknik ini sebenarnya sudah ada sejak 1999. Pada masa itu Microsoft membuat objek bernama *XMLHttpRequest* (Objek yang kelak akan dilibatkan

dalam membuat AJAX) sebagai control *ActiveX* pada Internet Explorer 5. AJAX menjadi sangat popular baru setelah digunakan intensif oleh Yahoo! Dan Google.

Menurut Jesse James Garret, di dalam artikelnya yang berjudul "*AJAX A New Approach to Web Applications*", AJAX yang lengkap mencakup [3]:

1. XHTML (*Extensible HyperText Markup Language*), sebagai peluasan dari HTML.
2. CSS (*Cascading Style Sheets*) sebagai pemformat bagian dari dokumen.
3. DOM (*Document Object Model*) untuk mengatur interaksi dan penampilan isi HTML dan XHTML secara dinamis.
4. XML (*Extensible Markup Language*) memungkinkan untuk melakukan pertukaran data.
5. XSLT (*Extensible Stylesheet Language Transformation*) adalah bahasa berbasis XML yang ditujukan untuk melakukan transformasi dokumen XML ke bentuk yang lain.
6. *XMLHttpRequest* untuk melakukan pertukaran data secara asinkron.
7. *JavaScript* sebagai bahasa untuk membentuk lapisan AJAX.

Memang nampaknya sangat kompleks karena melibatkan banyak hal. Namun tidak pelu khawatir bahwa hal itu akan menyusahkan dalam mewujudkannya. Dalam praktik, AJAX bisa diimplementasikan tanpa menggunakan XML, XSLT, maupun CSS.

Software yang digunakan untuk membangun aplikasi *web* dengan AJAX :

1. *Browser* (misalnya *Internet Explorer* dan *Mozilla Firefox*) pada sisi klien.
2. *Web Server* (misalnya *Apache*) pada sisi server.

Tentu saja yang dicantumkan di atas adalah kebutuhan minimal untuk membuat aplikasi *web* yang melibatkan AJAX. Jika aplikasi juga melibatkan database, di sisi server juga perlu dipasang *database server.* Selain itu, agar *web server* dapat berkomunikasi dengan *database server*, perantara seperti PHP diperlukan.

2.2.12.6 MySQL

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database* [14]. Semenjak tahun 70-an bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya *Oracle, Informix dan Sybase*. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya *dBase* ataupun *Clipper* yang masih menggunakan perintah – perintah pemrograman murni.

MySQL adalah sebuah *server database* SQL *multiuser* dan *multi-threaded* [14]. SQL sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program daemon 'mysqld' dan beberapa program lain serta beberapa pustaka. Sebagaimana *database* sistem yang lain, dalam SQL juga dikenal hierarki server dengan database-database. Tiap-tiap *database* memiliki tabel-tabel. Tiap-tiap tabel memiliki *field-field*. Umumnya informasi tersimpan dalam tabel – tabel yang secara logik merupakan struktur 2 dimensi terdiri atas baris dan kolom. *Field-field* tersebut dapat berupa data seperti *int, realm char, date, time* dan lainnya. SQL tidak memiliki fasilitas pemrograman yang lengkap, tidak ada *looping*

ataupun percabangan. Sehingga untuk menutupi kelemahan ini perlu digabung dengan bahasa pemrograman semisal *Pascal*. *MySQL* sering digunakan sebagai *SQL* server karena berbagai kelebihannya, antara lain [14]:

1. Source *MySQL* dapat diperoleh dengan mudah dan gratis.
2. Sintaksnya lebih mudah dipahami dan tidak rumit.
3. Pengaksesan *database* dapat dilakukan dengan mudah.

2.2.13 Flowchart

Flowchart merupakan representasi grafik dari algritma dengan menggunakan simbol-simbol tertentu yang masing-masing mempunyai fungsi yang khusus [11], *Flowchart* menggambarkan atau langkah dari sistem (proses, operasi, fungsi atau aktifitas). *Flowchart* mempunyai beberapa simbol yang digunakan.

2.2.14 Analisis sistem yang berjalan

Analisis terhadap sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan usulan perancangan sistem. Tahap analisa sistem perlu dilakukan dahulu sebelum dilakukan perancangan sistem. Tahap analisa ini merupakan tahapan yang penting didalam merancang suatu sistem. Analisa sistem yang dilakukan untuk mengetahui masalah apa yang sedang dihadapi oleh perusahaan.

Sistem yang sedang berjalan diartikan sebagai sistem yang sedang dipakai, sedangkan analisa sistem yang sedang berjalan diartikan sebagai cara memahami terlebih dahulu masalah yang dihadapi oleh sistem, seperti mendefinisikan kebutuhan-kebutuhan fungsional dari sistem sehingga dapat diketahui apa saja kebutuhan-kebutuhan pemakai yang belum terpenuhi oleh sistem yang sedang berjalan tersebut.

Tujuan analisa sistem untuk menentukan bentuk dari rancangan sistem yang akan ditetapkan. Analisa tersebut juga dapat menentukan langkah-langkah perancangan yang akan dibuat sehingga rancangan sistem sesuai dengan kebutuhan pemakai dan sistem mempunyai unjuk kerja yang efisien dan efektif, dapat menghasilkan informasi yang cepat, tepat dan akurat.

2.2.15 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional disini membahas kendala pada pelayanan atau fungsi sistem, yaitu :

1. *Availability*: ketersediaan aplikasi, misalnya harus terus menerus beroperasi selama hari kerja tanpa adanya kegagalan pada sistem yang di bangun karena berupa *web*, maka sistem ini dapat di akses setiap hari kerja, selama komputer terhubung ke server dan internet.
2. *Reliability:* keandalan, misalnya tidak pernah boleh gagal atau kegagalan yang ditolerir dibatasi hingga berapa persen, sehingga harus dipikirkan solusi atau *tolerant fault*. Pada sistem ini, dilihat dari aspek *reliability* nya

karena berupa aplikasi *web*, kegagalan yang mungkin di tolerir adalah pada proses pengolahan – pengolahan data klien, ketika *user* akan melakukan proses pengisian data klien, perkembangan klien dan penempatan program, jika sistem rusak atau mengalamin kegagalan, akan di konfirmasikan langsung dilayar terhadap bagian yang sedang melakukan pengolahan data.

3. *Ergonomy*: kenyamanan pakai bagi pengguna.

   Sistem ini akan di bangun sedemikian rupa agar *easy use* dan *easy learning,* sehingga *user* akan sangat nyaman untuk menggunakan aplikasi ini.

4. *Portability:* kemudahan untuk dibawa dan dioperasikan ke mesin/sistem operasi/*platform* yang lain. Karena berbasis *web* dan *programming language* menggunakan bahasa *php* maka sistem ini dapat di akses dimana saja, selama mesin/sistem operasi/*platform* mendukung *programming language* ini.
5. *Safety*: yang menyangkut keselamatan manusia, misalnya untuk SW yang dipakai pada sistem kontrol di pabrik, maka aspek ini tidak diperlukan pada sistem.
6. *Security*: aspek keamanan yang harus dipenuhi, yaitu pada proses perhitungan perkembangan klien, seleksi dan perhitungan nilai akhir program untuk massase dan shiatsu, selama *user* mengikuti prosedur dengan benar, maka sistem akan menjamin keamaan tersebut dengan baik.