BAB II

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

2.1.1 Sejarah singkat Perusahaan

PT. Telkom adalah sebuah badan usaha swasta penyediaan layanan pos dan telegraf tahun 1982, badan usaha swasta penyediaan layanan pos dan telegraf dibentuk pada masa pemerintahan *colonial* Belanda. Tahun 1906 pemerintahan *colonial* Belanda membentuk sebuah jawatan layanan pos dan telekomunikasi yang diberi nama jawatan pos telegram dan telepon (Pos, Telegram dan PTT). Tahun 1961 satatus jawatan diubah menjadi perusahaan Negara pos dan telekomunikasi (PN Postel). Tahun 1965 PN Postel pecah menjadi perusahaan pos dan giro (PN pos dan giro), dan perusahaan Negara telekomunikasi (PN Telekomunikasi). Tahun 1974 PN Telekomunikasi disesuaikan menjadi perusahaan umum telekomunikasi (Perumtel) yang memberikan jasa telekomunikasi nasional maupun internasional.

Pada tahun 1980 PT. Indonesia satelit corporation (Indosat) didirikan untuk memberikan jasa telekomunikasi internasional, yang terpisah dengan perumtel.

Pada tahun 1991 perumtel berubah menjadi perusahaan perseroan (persero) telekomunikasi indonesia berdasarkan PP no25 tahun 1991. Tahun 1995 penawaran umum perdana saham Telkom (*Initial Public*

Offering / IPO) dilaksanakan pada tanggal 14 november 1995. Sejak saat itu saham Telkom tercatat dan di perdagagngkan di Bursa Efek Jakarta (BEJ), Bursa Efek Surabaya (BES), *New York Stock Exchange* (NYSE) dan *London Stock Exchange* (LSE). Saham Telkom juga di perdagangkan tanpa pencatatan (*Public Offering Without Listing* / POWL) di Tokyo *Stock Exchange* (TSE).

Pada tahun 1996 kerjasama operasi (KSO) mulai diimplementasikan pada 1 januari 1996 diwilayah divisi regional 1 Sumatra mitra PT. Pramindo Ikat Nusantara (Pramindo), divisi regional III Jawa Barat dan Banten dengan mitra PT. Aria West Internasional (Aria West), divisi regional IV Jawa tengah dan di Yogyakarta dengan mitara PT. Mitra Global Telekomunikasi Indonesia (MGTI), divisi regional VI Kalimantan dengan mitra PT. Daymitra Telkomunikasi (Dayamitra), dan divisi regional VII kawasan timur Indonesia dengan mitra PT. Bukaka Singtel. Tahun 1999 undang-undang tahun 1999 no 36/1999 tentang penghapusan monopoli penyelenggaraan telekomunikasi.

2.1.2 Sejarah Divisi (Bagian)

Sejarah Kandatel Sukabumi, Kandatel Sukabumi berada sejak PT. Telkom Sukabumi didirikan tetapi pada saat itu belum melembaga (berdiri sendiri) melainkan berada dibawah bagian SDM sampai dengan sekarang menurut keputusan No KDR. 120 / PS 150 / RE-32 / 98. Sejak adanya perubahan-perubahan struktur organisasi ini bagian kesekretariatan dan

purel (bagian *public relations*) berada dibawah bagian program manajemen.

2.1.3 Visi dan Misi

a. Visi

"To become a leading Info Com player in the region", Telkom berupaya untuk menempatkan diri sebagai perusahaan *Info Com* terkemuka di kawasan Asia Tenggara, Asia dan akan berlanjut ke kawasan Asia Pasifik.

b. Misi

Telkom mempunyai misi memberikan layanan "*One Stop InfoCom Services with Excellent Quality and Competitive Price and To Be the Role Model as the Best Managed Indonesian Corporation* " dengan jaminan bahwa pelanggan akan mendapatkan layanan terbaik, berupa kemudahan, produk dan jaringan berkualitas, dengan harga kompetitif. Telkom akan mengelola bisnis melalui praktik-praktik terbaik dengan mengoptimalisasikan sumber daya manusia yang unggul, penggunaan teknologi yang kompetitif, serta membangun kemitraan yang saling menguntungkan dan saling mendukung secara sinergis.

2.1.4 Logo Perusahaan

Logo adalah sebuah identitas perusahaan sebagai tanda / alamat yang akan mengkomunikasikan arti dan memberi makna terhadap jati diri

perusahaan. Logo juga sebagai jiwa yang menghidupkan dan memberikan semangat bagi perusahaan di dalam melaksanakan misi dan mencapai visi perusahaan. Berikut ini adalah logo dari perusahaan PT. Telkom Indonesia.



Gambar 2.1 Logo PT. Telkom Indonesia

2.1.5 Struktur Organisasi dan *Job Desk*

Struktur perusahaan, organisasi atau instansi adalah gambaran menyeluruh mengenai satuan orang beserta tanggung jawab, jabatan, tugas, dan hubungan antara satu sama lainnya untuk mencapai satu tujuan. Adanya struktur pada Kandatel Sukabumi maka setiap jabatan yang terurut dalam struktur tersebut akan menjalankan tugas dan kewajiban dengan baik.

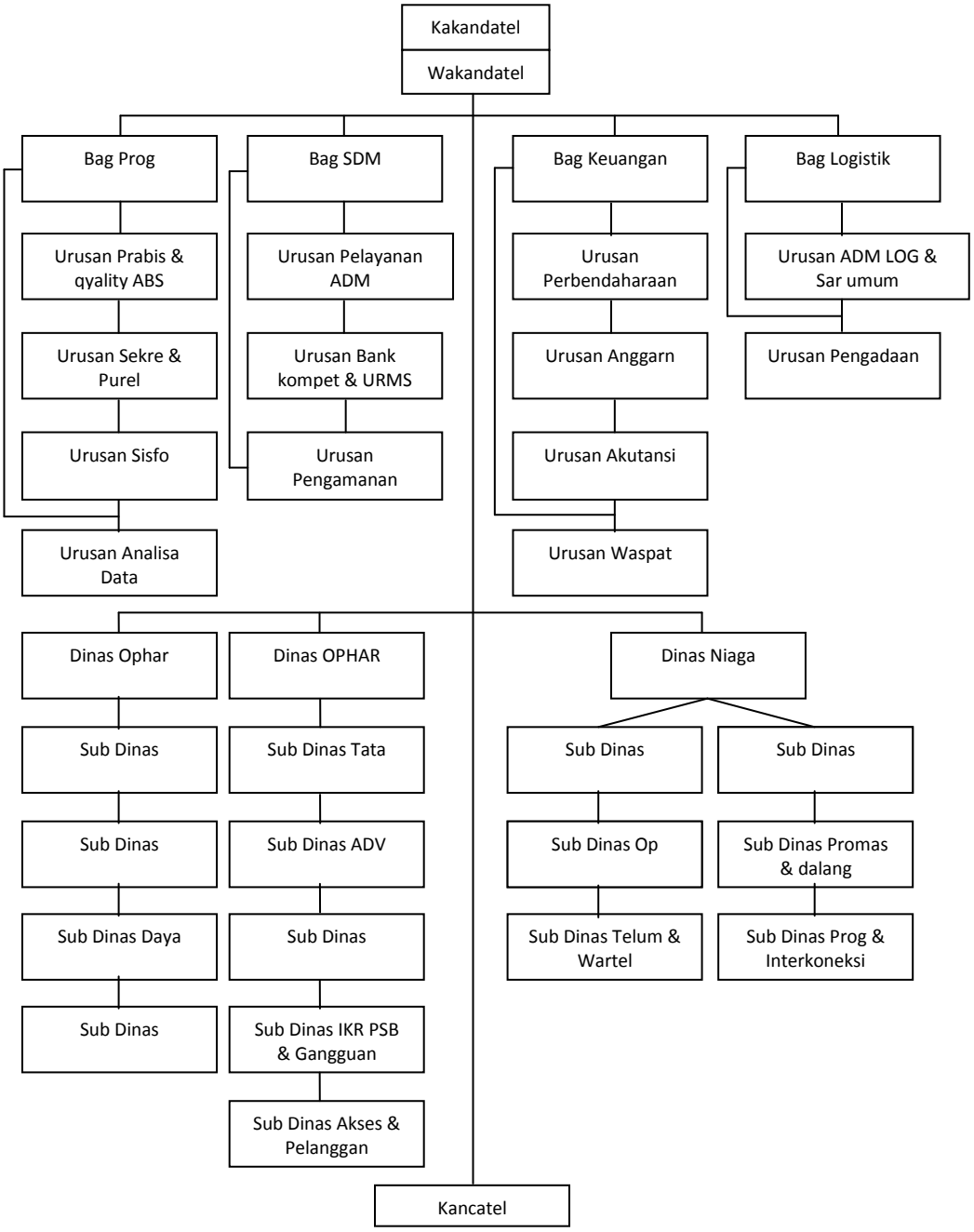
Pada tanggal 1 Desember 1998 Unit KSO Divisi Regional III Perusahaan Perseroan (Persero) PT. Telkom Indonesia menetapkan Struktur Organisasi untuk kantor daerah Telkom Sukabumi.

Berikut ini gambaran struktur Organisasi Kandatel Sukabumi.

Struktur Organisasi Kandatel Sukabumi

PT. Telkom Indonesia

Nomor : KDR. 120/PS 150/RE-32/98



Gambar 2.2 Struktur Organisasi Kandatel Sukabumi

2.1.5 *Job Deskription*

2.1.5.1 *Core Competencies*

1. *Customer Orientation* (Orientasi pelanggan atau ruang lingkup para pelanggan).
2. *Information Technology* (Teknologi Informasi yang ada di PT. Telkom Kandatel Sukabumi).
3. *Telecommunication Knowladge* (Pengetahuan pada bidang telekomunikasi).
4. *Achivement Orientation* (Ruang lingkup mengenai penghargaan-penghargaan yang telah diterima oleh PT. Telkom Kandatel Sukabumi).
5. *Communication* (Komunikasi).
6. *Innovation and Process Improvement* (Inovasi proses pengembangan yang dimiliki oleh sekretariat dan Purel di PT. Telkom Kandatel Sukabumi).
7. *Team Work* (Kerja Tim dalam bekerjasama untuk membangun kinerja yang baik dimata masyarakat).
8. *Adaptability & Diversity Management* (Kemampuan manajemen dan beradaptasi).
9. *Self Development* (Pengembangan Diri).

2.1.5.2 *Specific Competencies Personal Quality dan Skill Knowladge*

1. *Interpersonal Relationship* (Membangun hubungan interpersonal dalam tim kerja).
2. *Analitycal Thinking* (Mampu berfikir analisis dalam hubungan dengan pihak lain).
3. *Self Management* (Mengelola diri sendiri agar mampu beradaptasi dengan pihak lain).
4. *Information Seeking* (Mampu mencari informasi-informasi baru mengenai perusahaan yang dikelola).
5. *Omfort around to Management* (Mampu memberikan kenyamanan pada sekitar kita agar para karyawan maupun para tamu merasa nyaman).
6. *Learning Ability* (Kemampuan mempelajari semua pekerjaan yang diberikan oleh perusahaan).
7. *Tolerance for stress* (Bertoleransi dalam menghadapi masalah).

2.2 Landasan Teori

2.2.1 Basis Data (*Database*)

2.2.1.1 Pengertian Basis Data

Basis data dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah lemari arsip dan berwenang untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal-hal seperti : memberi map pada kumpulan-kumpulan arsip yang akan disimpan, menentukan

kelompok atau jenis arsip, memberi penomoran dengan pola tertentu yang nilainya unik pada setiap map, lalu menempatkan arsip-arsip tersebut dengan urutan tertentu di dalam lemari.

Basis Data terdiri dari dua kata yaitu Basis yang dapat diartikan sebagai markas atau gudang, sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan *file* atau *table* atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Yang sangat ditonjolkan dalam basis data adalah pengaturan atau pemilihan atau pengelompokkan atau pengorganisasian data yang akan kita simpan sesuai fungsi ataupun jenisnya. Hal ini dapat berbentuk

sejumlah *file* atau *table* terpisah atau dalam bentuk pendefinisian kolom-kolom data dalam setiap *file* atau *table.*

2.2.1.2 Sistem Basis Data

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi / tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu. Basis data hanyalah sebuah objek yang pasif atau mati.Ia ada karena ada pembuatnya. Ia tidak akan pernah berguna jika tidak ada pengelola atau penggeraknya. Yang menjadi penggeraknya secara langsung adalah program atau aplikasi (*software*).Gabungan keduanya menghasilakn sebuah system. Karena itu, secara umum sebuah system basis data merupakan system yang terdiri atas kumpulan *file* atau *table* yang saling berhubungan (dalam sebuah basis data di sebuah system komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi *file* atau *table* tersebut.

Lebih jauh lagi, dalam sebuah sistem basis data, secara lengkap akan terdapat komponen-komponen utama sebagai berikut :

a. Perangkat keras (*Hardware*)

   Perangkat keras yang biasanya terdapat dalam sebuah system basis data adalah :

1. Komputer (satu untuk sistem yang *stand-alone* atau lebih dari satu untuk sistem jaringan).
2. Memori sekunder yang *on-line* (*harddisk*).
3. Memori sekunder yang *off-line* (*Tape* atau *Removable Disk*) untuk keperluan *backup* data.
4. Media atau perangkat komunikasi (untuk sistem jaringan).

b. Sistem Operasi (*Operation System*)

Secara sederhana, sistem operasi merupakan program yang mengaktifkan atau memfungsikan sistem komputer, mengendalikan seluruh sumber daya (*resource*) dalam komputer dan melakukan operasi-operasi dasar dalam computer (operasi I/O, pengelolaan *file*, dll). Program pengelola basis data hanya dapat aktif (*running*) jika system operasi yang dikehendakinya atau sesuai telah aktif.

c. Basis Data (*Database*)

Sebuah system basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data seperti *file* atau *table*, indeks, dll. Di samping berisi atau menyimpan data, setiap basis data juga mengandung definisi struktur (baik untuk basis data maupun objek-objeknya secara detail).

d. Sistem (Aplikasi / perangkat lunak) Pengelola Basis Data (DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang

khusus. Perangkat lunak ini disebut DBMS (*Database Management System*) yang yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan atau konsistensi data, dan sebagainya.

e. Pemakai (*User*).

Ada beberapa jenis pemakai terhadap suatu system basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

1. Programmer Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML), yang disertakan (*embedded*) dalam program yang ditulis dalam bahasa pemrograman induk (seperti C, Pascal, Cobol, dan lain lain).

2. User Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan system tanpa menulis modul program. Mereka menyatakan *query* (untuk akses data) dengan bahasa *query* yang disediakan oleh suatu DBMS.

3. User Umum (*End User / Naïve User*)

   Pemakai yang berinteraksi dengan system basis data melalui pemanggilan satu program aplikasi permanen (*executable program*) yang telah ditulis atau disediakan sebelumnya.

4. User Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non-konvensional, tetapi untuk keperluan – keperluan khusus, seperti untuk aplikasi AI, Sistem Pakar, Pengolahan Citra, dll, yang bisa saja mengakses basis data dengan / tanpa DBMS yang bersangkutan.

   a. Aplikasi(*perangkat lunak*) lain (bedrsifat opsional)

      Aplikasi (perangkat lunak) lain ini bersifat opsional. Artinya, ada atau tidaknya tergantung pada kebutuhan kita. DBMS yang kita gunakan lebih berperan salam pengorganisasian data dalam basis data, sementara bagi pemakai basis data (khususnya yang menjadi *end – user* / *naive - user*) dapat dibutuhkan / disediakan program khusus / lain untuk melakukan pengisian, pengubahan dan pengambilan data. Pemrogram ini ada yang sudah disediakan bersama dengan DBMS-nya, ada juga yang harus dibuat sendiri dengan menggunakan aplikasi lain yang khusus untuk itu (*delevopment tools*).

2.2.1.3 Model Data

Model data sendiri dapat didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data. Oleh karena yang ingin ditunjukkan adalah makna dari data dan keterhubungannya dengan data lain, maka Model Data ini lebih tepat jika disebut Model Data Logik. Penyingkatannya menjadi Model Data dilakukan demi penyederhanaan penulisan.

Ada sejumlah secara dalam meprestasikan Model Data dalam perancangan basis data, yang secara umum dapat dibagi dalam 2 (dua) kelompok yaitu, yaitu :

a. Model Logik Data Berdasarkan Objek (*Objek – Based Logical Models*) yang terdiri dari :
    1. *Entity – Relationship Model*
    2. *Object – Oriented Model*
    3. *Semantic Data model*
    4. *Functional Data model*
b. Model Lojik Data Berdasarkan *Record* (*Record – Based Locical* Models) yang terdiri dari :
    1. *Relational Model*
    2. *Hierarchical Model*
    3. *Network Model*

2.2.1.3.1 Model *Entity-Relationship* (Model Keterhubungan - Entitas)

Pada model *Entity-Relationship*, semesta data yang ada di 'Dunia Nyata' diterjemahkan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umum disebut sebagai Diagram *Entity – Relationship* (Diagram E – R). sebelum kita mebahas lebih jauh tentang bagaimana Diagram E- R tersebut dapat kita gambarkan, maka yang harus lebih dulu diketahui adalah komponen – komponen pembentukan *Model Entity – relationship*. Sesuai namanya, ada dua komponen utama pembentuk Model Entity –Relationship, yaitu Entitas (*Entity*) dan Relasi (*Relation*).Kedua komponen ini di deskripsikan lebih jauh melalui sejumlah *Atribut* atau *Properti*.

2.2.1.3.1.1 Entitas (*Entity*) dan Himpunan Entitas (*Entitas Sets*)

Entitas merupakan invidu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Contoh – contoh himpunan entitas adalah :

a. Semua pelanggan atau pelanggan saja. Dengan entitas Budiman, Aminah dan seterusnya.
b. Semua mobil atau mobil saja. Dengan emtitas mobil Suzuki, mobil Toyota, mobil Honda dll.
c. Semua mahasiswa atau mahasiswa saja. Dengan entitas Ali, Budi, Iman dan seterusnya.

2.2.1.3.1.2 Atribut (Attributes / Properties)

Setiap Entitas pasti memiliki Atribut yang mendeskripsikan karakteristik (*property*) dari Entitas tersebut. Sebagaimana telah disebutkan sebelumnya, penentuan / pemilihan atribut – atribut yang relevan bagi sebuah entitas meru[akan hal penting lainnya dalam pembentukan model data. Penetapan atribut bagi sebuah entitas umunya memang didasarkan pada fakta yang ada.

Yang relevan untuk lebih diperhatikan dalam pembuatan model E-R adalah kedudukan atribut dalam entitas. Harus dapat kita bedakan / ketahui, mana atribut yang berfungsi sebagai *Primary Key* dan mana yang bukan (atribut deskriptif).

2.2.1.3.1.3 Relasi (*Relationship*)

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Misalnya, entitas seorang mahasiswa dengan nim = ‘980001’ dan *nama_mhs*=’Ali Akbar’ (yang ada di himpunan entitas Mahasiswa) mempunyai relasi dengan entitas sebuah mata kuliah dengan *kode_kul*=’IF-110’ dan *nama_kul*=’Struktur Data’. Relasi diantara kedua entitas tadi mengandung arti bahwa mahasiswa tersebut sedang mengambil atau mempelajari mata kuliah tersebut di sebuah perguruan tinggi yang kita tinjau. Kumpulan semua relasi di antara entitas – entitas yang terdapat pada himpunan entitas – himpunan entitas tersebut memebentuk

Himpunan Relasi.Istilah Himpunan Relasi jarang sekali digunakan dan lebih sering disingkat dengan istilah Relasi.

2.2.1.3.1.4 Kardinalitas / Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berlasi dengan entitas pada himpunan entitas yang lain. Sebagai contoh, dapat kita lihat bahwa entitas – entitas pada himpunan entitas Mahasiswa dapat berelasi dengan satu entitas, banyak entitas atau bahkan tidak satupun entitas dari himpunan entitas Kuliah.Begitu juga sebaliknya, entitas – entitas pada himpunan entitas Kuliah ada yang berelasi dengan beberapa entitas pada himpunan entitas Mahasiswa dan ada pula yang berelasi dengan satu entitas pada himpunan entitas Mahasiswa. Dengan demikian Kardinalitas Relasi yang terjadi diantara dua himpunan entitas :

a. Satu ke Satu

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya.

b. Satu ke Banyak

Setiap entitas pada himpunan entitas A dapat berhubungan banyak dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, di mana setiap entitas pada himpunan entitas

Bberhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

c. Banyak ke Satu

Entitas pada himpunan A berhubungan dengan paling banyak dengan satu jenis entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

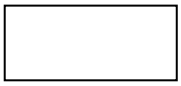
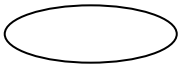
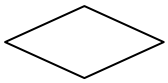
d. Banyak ke Banyak

Pada himpunan A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.2.1.3.2 *Diagram Entity – Relationship* (ERD).

Model *Entity – Relational* yang berisi komponen-komponen Himpunan Entitas dan Himpunan Relasi yang masing – masing dilengkapi dengan atribut – atribut yang mempresentasikan seluruh fakta dari '*dunia nyata'* yang di tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan ERD (*Entity Relationship Diagram*). Notasi – notasi simbolik di dalam ERD yang dapat digunakan adalah :

Tabel 2.1 Notasi – Notasi Simbolik ERD (*Entity Relationship Diagram*)

	Persegi panjang, ,menyatakan Himpunan Entitas.
	Elips menyatakan Atribut (Atribut yang berfungsi sebagai *key* digarisbawahi).
	Belah ketupat menyatakan Himpun Relasi.
	Garis sebagai penghubung antara Himpunan Relasi dengan Himpunana Entitas dan Himpunan Entitas dengan atributnya.
a. 1 dan 1 untuk relasi satu-ke-satu. b. 1 ke N untuk relasi satu-ke-banyak. c. N ke N untuk relasi banyak-ke-banyak.	Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka.

2.2.1.3.3 Varian Entitas dan Relasi

Idealnya himpunan entitas yang terlibat dalam sebuah ERD adalah himpunan entitas kuat / bebas.Himpunan entitas demikian tidak memiliki ketergantungan dengan himpunan entitas lainnya.Namun demikian, dalam pembentukan ERD kita tidak selalu dapat melibatkan himpunan entitas seperti itu. Ada kalanya kita juga melibatkan

himpunan entitas yang lemah atau merupakan bagian dari himpunan entitas lainnya.

2.2.1.3.3.1 Himpunan Entitas Lemah (*Weak Entity Sets*)

Berisi entitas – entitas yang kemunculannya tergantung pada eksistensinya dalam sebuah relasi terhadap entitas lain. Himpunan entitas demikian biasanya tidak memiliki atribut yang dapat berfungsi sebagai *key*, yang dapat menjamin keunikan entitas di dalamnya.

2.2.1.3.3.2 Sub Entitas (Subtype Entities)

Merupakan himpunan entitas yang beranggotakan entitas – entitas yang merupakan bagian dari himpunan entitas yang lebih *superior* (utama). Sub Entitas ini merupakan hasil dekomposisi (spesialisasi) himpunan entitas berdasarkan pengelompokan tertentu.

2.2.1.4 Arsitektur Sistem

Arsitektur system merujuk pada konfigurasi system secara keseluruhan yang akan menjadi "tempat hidup" dari DBMS, basis data dan aplikasi yang memanfaatkannya. Sudah seharusnya ditetapkan sejak awal sebelum memulai perancangan basis data, atau paling tidak sebelum melaksanakan tahap implementasi basis data.

Beberapa jenis arsitektur system yang dapat digunakan dan akan diuraikan pada bab ini adalah :

a. System Tunggal (*Stand - Alone*).

b. System Tersentralisasi (*Centralized System*).

c. System *Client – Server*

2.2.1.4.1 System Tunggal / Mandiri (*Stand-Alone*)

Pada arsitektur ini, DBMS, basis data dan aplikasi basis data ditempatkan pada computer yang sama dengan demikian, pemakai yang dapat menggunakannya disetiap saat juga hanya satu orang (*single user*). Arsitektur ini merupakan arsitektur system yang paling sederhana dan paling murah dan dapat menjadi sebuah pilihan jika basis data yang dikelola tidak terlalu besar dan lebih bersifat membantu mempercepat pekerjaan administrative.

2.2.1.4.2 Sistem Tersentralisasi (*Centralized System*)

Arsitektur ini terdiri atas sebuah mesin server dan sejumlah terminal (yang menjadi tempat *user* berinterkasi dengan sistem).Yang tersentralisasi dalam arsitektur ini dapat mencakup basis data, DBMS dan aplikasi basis data atau basis data saja.Untuk lingkup sentralisasi yang pertama, maka jenis *server*-nya sering disebut sebagai DBMS atau server aplikasi dan terminalnya lebih tepat disebut *dumb-terminal* (terminal pasif). Sedangkan jika yang disentralisasi hanya basis data, server yang digunakan biasa disebut *file-server* dan terminalnya disebut *work station*.

2.2.1.4.3 Sistem Client Server

Sebagai system tersentralisasi, arsitektur ketiga ini juga diterapkan pada sebuah system jaringan. System *Client – Server* ini ditujukan untuk mengatasi kelemahan – kemelamahan yang terdapat pada system tersentralisasi sebelumnya. Kelemahan pada bentuk system tersentralisasi yang pertama, yaitu beratnya beban server yang harus menangani semua proses, diatasi dengan membagi beban itu menjadi dua bagian, *client* (yang menjalankan aplikasi basis data) dan *server* (yang menjalankan DBMS dan berisi basis data) pada mesin yang berbeda. Sedang kelemahan pada bentuk sistem tersetralisasi yang kedua yaitu, padatnya lalu lintas data antara server dan *work-station* diatasi dengan mekanisme transfer data yang lebih efisien.

Sistem ini terdiri atas dua komponen utama, yaitu *client* dan *server*.*Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu didatangi oleh *client*. *Client* selanjutnya mengupayakan agar semua proses sedapat mungkin ditangani sendiri. Jika ada proses yang harus melibatkan data yang tersimpan pada basis data, barulah *client* melakukan “kontak” dengan *server*.

2.2.1.5 DBMS (Database Management System)

DBMS umunya menyediakan program khusus (*utilitas / utility*) yang dapat digunakan secara interaktif untuk melakukan berbagai operasi

terhadap basis data, seperti pembuatan *table*, penghapusan *table*, penambahan data, pengubahan data, pencarian data, penghapusan data, dll. Namun disamping adanya program khusus itu, DBMS juga umumnya menyediakan sekumpulan perintah dalam bentuk *command line*, yakni perintah yang dituliskan pemakai untuk maksud yang sama. Perintah – perintah ini dapat diberikan dan dikerjakan oleh DBMS melalui utilitas lain yang juga disediakan DBMS atau melalui program / aplikasi yang dibuat sendiri oleh pemakai. Kumpulan perintah ini dapat disebut sebagai Bahasa Basis Data (*Database Language*). Karena DBMS yang digunakan untuk mengelola basis data dan *development tools* yang digunakan untuk menulis aplikasi basis data belum tentu dibuat berasal dari perusahaan pembuat perangkat lunak (*vendor*) yang sama, maka diperlukan sebuah bahasa basis data yang sifatnya standar. Dengan bahasa yang standar, kita dapat menggunakan bahasa tersebut untuk mengakses data dimanapun tanpa memperdulikan DBMS yang digunakan. Atau bahkan sebaliknya data yang dikelola sebuah DBMS dapat diakses dari manapun, tidak peduli *development tools* yang digunakan. Dengan begitu terdapat independensi antara DBMS dan *development tools*.

2.2.2 Data Flow Diagram

DFD (*Data Flow Diagram*) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan dan kemana tujuan data yang akan keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang

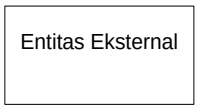
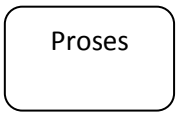
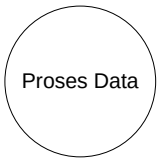
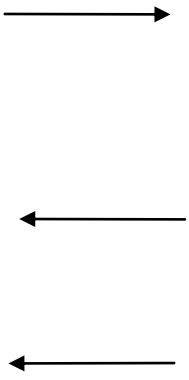
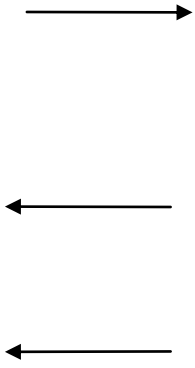
tersimapan dan proses yang dikenakan pada data tersebut. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Diagram yang menunjukkan bagaimana data berpindah / berjalan didalam sistem Informasi yang akan dikembangkan. Data flow diagram dapat menunjukkan ringkasan dari sistem yang luas / besar dari sistem input, proses dan output. Adapun keuntungan menggunakan DFD, yaitu :

1. Bebas menjalankan implementasi teknis system.
2. Komunikasi pengetahuan sistem yang ada dengan *user*.
3. Menganalisis sistem yang diajukan untuk menentukan ketepatan data dan proses yang dibutuhkan.

Dalam pembuatan DFD ini memiliki beberapa peraturan yang nantinya dapat mempermudah proses pembuatannya. Selain itu ada simbolik dalam DFD yang dapat digunakan, sebagai berikut

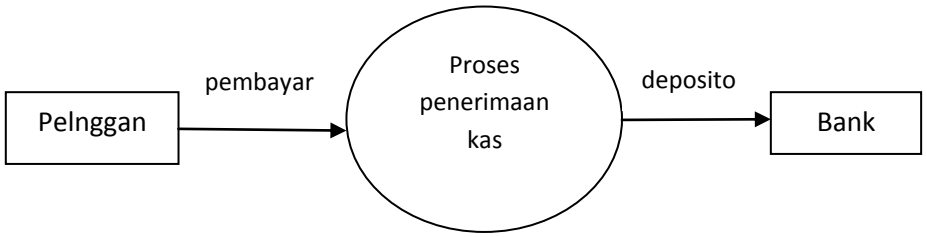
Tabel 2.2 Simbolik dalam DFD (*Data Flow Diagram*)

Gane/Sarson	**Yourdon/De Marco**	**Keterangan**
		a. Sumber / tujuan aliran data dari / ke sistem. b. Entitas eksternal, dapat berupa orang / unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		a. Fungsi dijelaskan oleh nama yang dicantumkan dalam lingkaran. b. Nama proses : Kata kerja + Kata benda. c. Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak didefinisikan.
Aliran data 	Aliran data 	a. Mengidentifikasikan satu aliran data pada waktu yang tidak dispesifikasikan. b. Aliran data dengan arah khusus dari sumber ke tujuan. c. Nama data dicantumkan pada garis.
		a. Menyatakan arsip meja, lemari, *tape*, disk atau media lainnya.

Data Store		b. Nama data / arsip diletakkan di dalam symbol. c. Penyimpanan data atau tempat data direfer oleh proses.

2.2.2.1 Type DFD (Data Flow Diagram)

DFD terdiri dari tiga tipe, *Context diagram, DFD Physical* dan *DFD Logical. Context diagram* merupakan diagram tingkat atas, yaitu diagram yang paling tidak detail, dari sebuah system informasi yang menggambarkan aliran – aliran data ke dalam dan keluar sistem.



Gambar 2.3 Contoh *Context Diagram*

DFD *Physical* merupakan representasi grafik dari sebuah sistem yang menunjukkan entitas – entitas internal dan eksternal dari system tersebut, dan aliran – aliran ke dalam dan keluar dari entitas – entitas tersebut. Entitas – entitas personal adalah personel, tempat (misal sebuah bagian organisasi), atau mesin (misalnya, komputer) dalam sistem tersebut yang mentransformasi data. Jadi dapat dikatakan tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana,

bagaimana dan oleh siapa proses – proses dalam sebuah sistem dilakukan.

DFD *Logical* merupakan grafik dari sebuah sistem yang menunjukkan proses – proses dalam sistem tersebut dan aliran – aliran data ke dalam dan keluar dari proses – proses tersebut. Kita menggunakan dokumentasi DFD *Logical* untuk membuat dokumentasi sebuah system informasi. Karena DFD *Logical* dapat mewakili logika sistem tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana dan oleh siapa proses – proses dalam sistem tersebut dilakukan. Dengan demikian, DFD *Logical* menggambarkan kegiatan – kegiatan yang dilakukan oleh sebuah system, sedangkan DFD *Physical* menggambarkan infrastruktur yang dimiliki oleh sebuah system.

DFD memiliki dekomposisi atau *Levelling* DFD yang jumlah proses yang ter-*identifikasi* dan harus digambarkan dalam DFD, mungkin saja berjumlah ratusan. Apapun cara terbaik untuk mengorganisasikan gambar DFD adalah dengan mem-*partisi* DFD secara *top-down*. Dalam penggambaran symbol DFD, terdapat beberapa peraturan yang harus diperhatikan sehingga dalam penggambarannya tidak terjadi kesalahan. Peraturan – peraturan yang ada adalah sebagai berikut :

1. Antar entity luar tidak diizinkan terjadi hubungan atau relasi.
2. Tidak boleh ada aliran data antara entitas luar dengan data *store*.

3. Tidak boleh ada aliran data antara data *store*.

2.2.3 Spesifikasi Kebutuhan Perangkat Lunak

Model analisis adalah teknik pertama untuk mempresentasikan sistem. Pemodelan analisis menggunakan kombinasi dari text dan diagram untuk mepresentasikan kebutuhan perangkat lunak (data, fungsi, dan tingkah) dengan cara yang dimengerti. Ada dua tipe pemodelan analisis yang digunakan, yaitu analisis terstruktur dan analisis berorientasi objek. Model data dimodelkan menggunakan diagram E – R untuk menggambarkan objek data, atribut dan relasi. Model fungsional dimodelkan menggunakan DFD untuk menggambarkan bagaimana data ditransformasikan kedalam system, model tingkah laku (*Behavioral modeling*) dimodelkan menggunakan *state transition diagrams* (STD) untuk menggambarkan tingkah laku kejadian diluar system. Analisis kerja suatu produk harus dikaji dalam hal *completeness*, *correctness* dan *consistency*.

Analisis juga memiliki tiga model analisis, yaitu :

a. Analisis Terstruktur (DeMarco)

Analisis produk harus sangat dipertahankan, terutama persyaratan spesifikasi perangkat lunak. Masalah ukuran harus ditangani dengan menggunakan metode efektif partisi. Sedangkan grafis harus digunakan bila memungkinkan. Membedakan antara pertimbangan logis (penting) dan fisik (implementasi), selain itu

membantu menemukan sesuatu dengan persyaratan partisi dan partisi dokumen sebelum spesifikasi. Menemukan cara untuk melacak dan mengevaluasi *user interface*. Merancang alat yang menggambarkan logika dan kebijakan yang lebih baik dari teks naratif.

b. Analisis Model Objektif

Menggambarkan apa yang *customer* butuhkan sehingga dapat dijadikan bahan dasar untuk proses *design*, selain itu dapat merencanakan sekumpulan kebutuhan yang dapat dijadikan bahan validasi pada saat perangkat lunak dibangun.

c. Analisis Model Elemen

Ada beberapa elemen – elemen yang mendukung, yaitu :

1. Kamus data : berisi deskripsi dari semua objek data yang dibutuhkan atau diproduksi oleh perangkat lunak.
2. *Entity Relationship Diagram* (ERD) : menggambarkan hubungan antara objek data.
3. *Data Flow Diagram* (DFD) : memberikan indikasi tentang bagaimana data berubah karena mereka bergerak melalui sistem. Juga menggambarkan fungsi yang mentrtansformasikan aliran data (fungsi dipresentasikan dalam DFD menggunakan spesifikasi proses atau PSPEC).

2.2.4 Desain

Desain perangkat lunak adalah suatu proses interaktif yang melalui persyaratan diterjemahkan ke dalam "*Blue Print*" untuk membangun suatu

perangkat lunak. Sepanjang proses desain, kualitas desain yang melengkapi dinilai dengan serangkaian kajian teknis formal atau desain awal. Pedoman bagi evaluasi suatu desain yang baik adalah :

a. Desain harus mengimplementasi keseluruhan persyaratan *eksplisist* yang dibebankan dalam model analisis, dan harus mengakomodasi semua persyaratan implisist yang diinginkan pelanggan.
b. Desain harus menjadi panduan yang dapat dibaca.
c. Desain harus memberikan suatu gambaran lengkap mengenai perangkat lunak.

2.2.5 Pengujian (*Blackbox*)

Pengujian Perangkat Lunak Adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidak mampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.

Meningkatnya visibilitas (kemampuan) perangkat lunak sebagai suatu elemen sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak, memotivasi dilakukannya perencanaan yang baik melalui pengujian

yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak dari pada membangun.

Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Sasaran itu berlawanan dengan pandangan yang biasanya dipegang yang menyatakan bahwa pengujian yang berhasil adalah pengujian yang tidak ada kesalahan yang ditemukan.Data yang dikumpulkan pada saat pengujian dilakukan memberikan indikasi yang baik mengenai reliabilitas perangkat lunak dan beberapa menunjukkan kualitas perangkat lunak secara keseluruhan, tetapi ada satu hal yang tidak dapat dilakukan oleh pengujian, yaitu "*pengujian tidak dapat memperlihatkan tidak adanya cacat, pengujian hanya dapat memperlihatkan bahwa ada kesalahan perangkat lunak".*

Sebelum mengaplikasikan metode untuk mendesain *test case* yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar yang menuntun pengujian perangkat lunak, yaitu:

a. Semua pengujian harus dapat ditelusuri sampai ke persyaratan pelanggan, maksudnya mengungkap kesalahan dari cacat yang menyebabkan program gagal.
b. Pengujian harus direncanakan lama sebelum pengujian itu mulai, maksudnya semua pengujian dapat direncanakan dan dirancang sebelum semua kode dijalankan.
c. Prinsip *Pareto* berlaku untuk pengujian perangkat lunak, maksudnya dari 80% kesalahan yang ditemukan selama pengujian dapat ditelusuri sampai 20% dari semua modul program.
d. Pengujian harus mulai “dari yang kecil” dan berkembang ke pengujian “yang besar”, Selagi pengujian berlangsung maju, pengujian mengubah fokus dalam usaha menemukan kesalahan pada *cluster* modul yang terintegrasi dan akhirnya pada sistem.
e. Pengujian yang mendalam tidak mungkin karena tidak mungkin mengeksekusi setiap kombinasi jalur skema pengujian dikarenakan jumlah jalur permutasi untuk program menengah pun sangat besar.
f. Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang *independent*.

Dalam lingkungan yang ideal, perekayasa perangkat lunak mendesain suatu program komputer, sebuah sistem atau produk dengan

testabilitas dalam pikirannya.Hal ini memungkinkan individu yang berurusan dengan pengujian mendesain *test case* yang efektif secara lebih mudah. Testabilitas adalah seberapa mudah sebuah program computer dapat diuji. Karena sangat sulit, perlu diketahui apa yang dapat dilakukan untuk membuatnya menjadi lebih mudah. *Procedural* dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi.

Sasaran utama desain *test case* adalah untuk mendapatkan serangkaian pengujian yang memiliki kemungkinan tertinggi di dalam pengungkapan kesalahan pada perangkat lunak.

Tehnik pengujian *blackbox* berfokus pada domain informasi dari perangkat lunak, dengan melakukan test case dengan menpartisi domain input dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Metode pengujian *graph-based* mengeksplorasi hubungan antara dan tingkah laku objek – objek program. Partisi ekivalensi membagi domain input ke dalam kelas data yang mungkin untuk melakukan fungsi perangkat lunak tertentu. Analisis nilai batas memeriksaa kemampuan program untuk menangani data pada batas yang dapat diterima.

Metode pengujian yang terspesialisasi meliputi sejumlah luas kemampuan perangkat lunak dan area aplikasi. GUI, arsitektur *client/ server*, dokumentasi dan fasilitas *help* dan sistem *real time* masing –

masing membutuhkan pedoman dan teknik khusus untuk pengujian perangkat lunak.

2.2.6 Adobe Dreamwaver CS 3

Adobe Dreamweaver CS 3 merupakan program penyunting halaman web keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak digunakan oleh pengembang web karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 9. Versi terakhir Dreamweaver keluaran Adobe Systems adalah versi 11 yang ada dalam Adobe Creative Suite 5 (sering disingkat Adobe CS5).

2.2.7 Internet

World Wide Web (disingkat sebagai ***WWW*** atau ***Web***) adalah suatu ruang informasi yang yang dipakai oleh pengenal global yang disebut *Uniform Resource Identifier* (URI) untuk mengidentifikasi sumber-sumber daya yang berguna. *WWW* sering dianggap sama dengan Internet secara keseluruhan, walaupun sebenarnya ia hanyalah bagian dari padanya.

WWW (*World Wide Web*), merupakan kumpulan *web server* dari seluruh dunia yang berfungsi menyediakan data dan informasi untuk dapat digunakan bersama. *WWW* atau biasa disebut *web* adalah bagian yang paling menarik dari Internet. Melalui *web*, dapat mengakses

informasi-informasi yang tidak hanya berupa teks tetapi bisa juga berupa gambar, suara, *video* dan animasi.

Fasilitas ini tergolong masih baru dibandingkan surel (*email*), sebenarnya *WWW* merupakan kumpulan dokumen-dokumen yang sangat banyak yang berada pada komputer *server* (*web server*), di mana server-server ini tersebar di lima benua termasuk Indonesia, dan terhubung menjadi satu melalui jaringan Internet. Dokumen-dokumen informasi ini disimpan atau dibuat dengan format *HTML* (*Hypertext Markup Language*). Suatu halaman dokumen informasi dapat terdiri atas teks yang saling terkait dengan teks lainnya atau bahkan dengan dokumen lain. Keterkaitan halaman lewat teks ini disebut *hypertext*. Dokumen infomasi ini tidak hanya terdiri dari teks tetapi dapat juga berupa gambar, mengandung suara bahkan klip *video*. Kaitan antar-dokumen yang seperti itu biasa disebut *hypermedia*.

Jadi dapat disimpulkan bahwa *WWW* adalah sekelompok dokumen multimedia yang saling terkoneksi menggunakan *hypertext link*. Dengan mengklik *hyperlink*, maka bisa berpindah dari satu dokumen ke dokumen lainnya.

WWW adalah suatu program yang ditemukan oleh Tim Berners-Lee pada tahun 1991. Awalnya Berners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya.Untuk itu, dia mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah program peranti lunak

yang diberi nama *Equire*. Dengan program itu, Berners-Lee berhasil menciptakan jaringan terkait antara berbagai arsip sehingga memudahkan informasi yang dibutuhkan. Inilah yang kemudian menjadi dasar dari sebuah revolusi yang dikenal sebagai *web*.

WWW dikembangkan pertama kali di Pusat Penelitian Fisika Partikel Eropa (CERN), Jenewa, Swiss. Pada tahun 1989 Berners-lee membuat proposal untuk proyek pembuatan *hypertext* secara global, kemudian pada bulan Oktober 1990, *''World Wide Web"* sudah bisa dijalankan dalam lingkungan CERN. Pada musim panas tahun 1991, *WWW* resmi digunakan secara luas pada jaringan Internet.

2.2.8 *Website*

Sebuah situs *web* (sering pula disingkat menjadi situs saja; web site, site) adalah sebutan bagi sekelompok halaman *web* (*web page*), yang umumnya merupakan bagian dari suatu nama domain (domain name) atau subdomain di *World Wide Web* (*WWW*) di Internet. *WWW* terdiri dari seluruh situs web yang tersedia kepada publik. Halaman-halaman sebuah situs web diakses dari sebuah URL yang menjadi "akar" (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka" atau laman *web*), dan biasanya disimpan dalam server yang sama. Tidak semua situs *web* dapat diakses dengan gratis. Beberapa situs *website* memerlukan pembayaran agar dapat

menjadi pelanggan, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surat elektronik (*e-mail*), dan lain-lain.

Secara terminologi, *website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (*WWW*) di Internet. Sebuah halaman web adalah dokumen yang ditulis dalam format *HTML* (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server* website untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari *website* akan bisa diakses melalui sebuah URL yang biasa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun, *hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka sususan keseluruhan dan bagaimana arus informasi ini berjalan.

Beberapa *website* membutuhkan subskripsi (data masukan) agar para *user* bisa mengakses sebagian atau keseluruhan isi *website* tersebut. Contohnya, ada beberapa situs-situs bisnis, situs-situs e-mail gratisan, yang membutuhkan subkripsi agar kita bisa mengakses situs tersebut.

Penemu *website* adalah Sir Timothy John ¨Tim¨ Berners-Lee, sedangkan *website* yang tersambung dengan jaringan, pertamakali muncul

pada tahun 1991.Maksud dari Tim ketika membuat website adalah untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia bekerja.Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa *WWW* dapat digunakan secara gratis oleh semua orang.

Sebuah *website* bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari sebuah organisasi, perusahaan, dan biasanya *website* itu menujukkan beberapa topik khusus, atau kepentingan tertentu. Sebuah *website* bisa berisi *hyperlink* (pranala) yang menghubungkan ke *website* lain, jadi, kadangkala perbedaan antara *website* yang dibuat oleh individu perseorangan dengan *website* yang dibuat oleh organisasi bisnis bisa saja tidak kentara.

Website ditulis, atau secara dinamik di konversi menjadi *HTML* dan diakses melalui sebuah program *software* yang biasa disebut dengan *web browser,* yang dikenal juga dengan *HTTP Client*. Halaman *web* dapat dilihat atau diakses melalui jaringan komputer dan internet, perangkatnya bisa saja berupa komputer pribadi, laptop, PDA ataupun telepon selular.

Sebuah *website* dibuat didalam sebuah sistem komputer yang dikenal dengan *server web,* juga disebut *HTTP Server,* dan pengertian ini juga bisa menunjuk pada software yang dipakai untuk menjalankan sistem ini, yang kemudian menerima lalu mengirimkan halaman-halaman yang diperlukan untuk merespon permintaan dari pengguna. *Apache* adalah

piranti lunak yang biasa digunakan dalam sebuah *web server*, kemudian setelah itu adalah *Microsoft Internet Information Services* (*IIS*).

Sebuah *website* statik, adalah salah satu bentuk *website* yang isi didalam *website* tersebut tidak dimaksudkan untuk di *update* secara berkala, dan biasanya di maintain secara manual oleh beberapa orang yang menggunakan *software editor*. Ada 3 tipe kategori *software editor* yang biasa dipakai untuk tujuan maintaining ini, mereka adalah :

1. Elemen 1 Penyunting *teks*. Contohnya adalah *Notepad* atau *TextEdit*, dimana *HTML* diubah didalam program editor tersebut.
2. Elemen 2 *WYSIWYG editor*. Contohnya Microsoft Frontpage dan Macromedia Dreamweaver, dimana situs di edit menggunakan *GUI* (*Graphical User Interface*) dan format *HTML* ini secara otomatis di generate oleh *editor* ini.
3. Elemen 3 *Editor* yang sudah memiliki *template*, contohnya *Rapidweaver* dan *iWeb*, dimana, editor ini membolehkan user untuk membuat dan mengupdate websitenya langsung ke *web server* secara cepat, tanpa harus mengetahui apapun tentang *HTML*. Mereka dapat memilih templat yang sesuai dengan keinginan mereka, menambah gambar atau obyek, mengisinya dengan tulisan, dan dengan sekejap mereka sudah dapat membuat situs *web* tanpa harus melihat sama sekali kode-kode *HTML*.

Sebuah *website dynamic* adalah *website* yang secara berkala, informasi didalamnya berubah, atau website ini bisa berhubungan dengan user dengan berbagai macam cara atau metode (*HTTP cookies* atau *Variable Database*, sejarah kunjungan, *variable* sesi dan lain-lain) bisa juga dengan cara interaksi langsung menggunakan *form* dan pergerakan mouse. Ketika *web server* menerima permintaan dari user untuk memberikan halaman tertentu, maka halaman tersebut akan secara otomatis di ambil dari media penyimpanan sebagai respon dari permintaan yang diminta oleh *user*. Sebuah situs dapat menampilkan dialog yang sedang berlangsung diantara dua *user*, memantau perubahan situasi, atau menyediakan informasi yang berkaitan dengan sang *user*.

Ada banyak jenis sistem *software* yang dapat dipakai untuk meng-generate *Dynamic Web System* dan *Situs Dynamic*, beberapa diantaranya adalah *Cold Fusion* (*CFM*), *Active Server Pages* (*ASP*), *Java Server Pages* (*JSP*) dan *PHP*, bahasa program yang mampu untuk meng-*generate Dynamic Web System* dan Situs Dinamis. Situs juga bisa termasuk didalamnya berisi informasi yang diambil dari satu atau lebih *database* atau bisa juga menggunakan teknologi berbasis *XML*, contohnya adalah *RSS*. Isi situs yang statis juga secara periodik di *generate*, atau, apabila ada keadaan dimana dia butuh untuk dikembalikan kepada keadaan semula, maka dia akan di *generate*, hal ini untuk menghindari kinerjanya supaya tetap terjaga.

Plugin tersedia untuk menambah banyaknya *feature* dan kemampuan dari *web browser*, dimana, plugin ini dipakai untuk membuka *content* yang biasanya berupa cuplikan dari gambar bergerak (*active content*) contohnya adalah *Flash*, *Shockwave* atau *applets* yang ditulis dalam bahasa *JAVA*. *Dynamic HTML* juga menyediakan untuk user supaya dia bisa secara interaktif dan *realtime*, meng-update di *web page* tersebut (catatan; halaman yang dirubah, tak perlu di load atau di *reloaded* agar perubahannya dapat dilihat), biasanya perubahan yang dilakukan mereka memakai *DOM* dan *Javascript* yang sudah tersedia pada semua *Web Browser* sekarang ini.

Seperti yang tertulis di atas, di luar sana ada beberapa perbedaan dalam penulisan dari terminologi *website*. Walaupun *Website* sudah secara umum dipakai, namun untuk Associated Press Stylebook, Reuters, Microsoft, Academia, dan kamus-kamus yang ada, penulisan yang mereka pakai adalah dengan menggunakan 2 kata, yaitu *Web site*. Hal ini karena "*Web*" bukanlah terminilogi umum, namun kependekan dari *World Wide Web*.