

BAB II

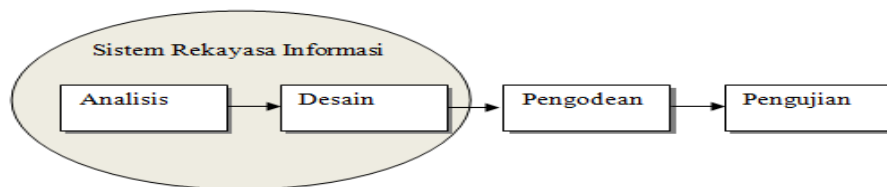
LANDASAN TEORI

2.1. Konsep Dasar Web

2.1.1. Konsep Dasar Model Pengembangan Sistem

Menurut Rosa, dkk (2013:26) menyatakan bahwa SDLC (*Software Development Life Cycle*) atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

SDLC (*System Development Life Cycle*) memiliki beberapa model dalam penerapan tahapan prosesnya. Salah satunya model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classis life cycle*). Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah. Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Gambar II.1

Ilustrasi Model *Waterfal*

Sumber : Rosa, dkk (2013: 29)

1. Analisa kebutuhan perangkat lunak

Merupakan proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak agar dapat di pahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain.

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi anatarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.



5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.2. Konsep Dasar Pemrograman

Menurut Rosa, dkk. (2013:26) menyatakan bahwa “pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer”.

Prinsip utama pemrograman terstruktur adalah jika suatu proses telah sampai pada suatu titik tertentu, maka proses selanjutnya tidak boleh kembali lagi ke baris sebelumnya, kecuali untuk proses berulang (*Loop*). Berikut ini akan diuraikan teknik pemrograman terstruktur.

1. Pemrograman Modular

Dalam pemrograman modular, program dipecah-pecah menjadi modul-modul. Setiap modul menunjukkan fungsi dan tugas tunggal. Modul-modul tersebut ditulis dan dicari kesalahannya secara terpisah. Karena ukuran dan tujuan setiap modul dibatasi, maka terjadinya kesalahan dalam program tersebut dapat dikurangi. Pemrograman modular ini diterapkan

menggunakan subroutines, yaitu sebuah kumpulan perintah yang melakukan tugas pemrosesan yang terbatas, seperti mencetak sebagian laporan, membaca masukan data, atau menghitung akar pangkat.

2. Pemrograman *Top-Down*

Pendekatan *Top-down* ini sangat berguna dalam perencanaan pemrograman modular. Dalam pemrograman *top-down* (atas-bawah), yang pertama harus kita definisikan adalah modul utama. Modul utama yang dimaksud adalah modul yang pertama kali dijalankan atau modul yang memanggil modul lainnya atau juga modul yang mengakhiri proses program tersebut.

3. Teknik *Bottom Up*

Merupakan teknik pemecahan masalah yang mulai ditinggalkan karena sulit untuk melakukan standarisasi proses dari prosedur-prosedur yang sudah ada untuk digabungkan menjadi satu kesatuan.



2.1.3. *Internet*

Menurut Irawan (2011:2) menyatakan bahwa “*internet* merupakan kependekan dari kata “*internetwork*”, yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan. Sistem computer terhubung secara global dan menggunakan TCP/IP sebagai protokol”. Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

Menurut Irawan (2011:3) menyatakan bahwa“ aplikasi *internet* adalah program yang digunakan pada jaringan *internet* untuk mengakses informasi,

berkomunikasi, dan sebagainya “. Salah satu aplikasi pokok yang digunakan saat berinternet adalah *Web Browser*.

Browser adalah aplikasi yang digunakan untuk menampilkan halaman *web* beserta kontennya. Beberapa aplikasi *browser* yang banyak digunakan antara lain *Internet Explorer*, *Firefox*, *Chrome*, dan *Opera*.

Istilah-istilah yang umum dikenal di *internet* antara lain:

a. *Web*

Merupakan fasilitas yang dapat menampilkan data-data berupa teks, gambar, bunyi, animasi dan multimedia lainnya dalam *internet*. Ada yang bilang *website* atau lazim disingkat *web* adalah halaman informasi yang disediakan melalui jalur *internet* sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan *internet*.

b. WWW (*World Wide Web*)

World Wide Web adalah sebuah sistem dari server *internet* yang mendukung dokumen khusus diformat. Dokumen-dokumen yang diformat dalam sebuah script yang disebut HTML (*HyperText Markup Language*) yang mendukung link ke dokumen lain, serta grafis, audio, dan file video.

c. *Web Pages* (*Halaman Web*)

Merupakan halaman khusus dari situs *web* tertentu yang tersimpan dalam bentuk *file*. Dalam *web pages* tersimpan berbagai informasi dan *link* yang menghubungkan suatu informasi ke informasi lain, baik itu dalam *pages* yang sama ataupun *web pages* lain pada *website* yang berbeda.

d. *Homepage*



Merupakan suatu sarana dasar untuk memperkenalkan secara singkat tentang apa yang menjadi isi dari keseluruhan *website*. Biasanya berada pada halaman pertama dari suatu *website* yang berisi tentang apa dan siapa pemilik *website* tersebut.

e. *Browser*

Merupakan *software* yang digunakan untuk menampilkan informasi dari *server web*.

f. *Universal Resource Locator (URL)*

Merupakan suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu *web server*. URL dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri dari:

- 1) Protokol yang digunakan oleh *browser* untuk mengambil informasi.
- 2) Nama komputer (*server*) dimana informasi tersebut berada.
- 3) Jalur atau *path* serta nama *file* dari suatu informasi

g. *Hypertext Transfer Protocol (HTIP)*

Merupakan suatu protokol yang di gunakan untuk mengakses *website* dan yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dari *web server*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses dokumen *HTML*.

h. *Hypertext Markup Language (HTML)*

Merupakan suatu bahasa yang digunakan untuk menulis halaman *web*. Dokumen *HTML* adalah suatu dokumen teks biasa dan disebut sebagai *markup language* karena mengandung tanda-tanda (*tag*) tertentu yang



digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen.

i. *Domain Name System (DNS)*

DNS adalah distribute database sistem yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang menggunakan TCP/IP (*Transmission Control Protocol/ Internet Protocol*). DNS biasa digunakan pada aplikasi yang terhubung ke *internet* seperti *web browser* atau *e-mail*, dimana DNS membantu memetakan *host name* sebuah komputer ke *IP address*.

2.1.4. Pengenalan PHP (*Hypertext Preprocessor*)

Pengertian *PHP* menurut Suryatiningsih (2009:141) “*PHP* adalah bahasa *scripting* yang menyatu dengan *HTML* dan dijalankan pada *serverside*. Artinya semua *sintax* yang diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja”.

Secara khusus, *PHP* dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman *web*. Pada prinsipnya *PHP* mempunyai fungsi yang sama dengan skrip-skrip seperti *ASP (Active Server Page)*, *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa *PHP* sebenarnya bisa dipakai secara *command line*. Artinya, skrip *PHP* dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam *HTML* sehingga suatu halaman *HTML* tidak lagi bersifat statis, namun menjadi bersifat



dinamis. Sifat *server side* ini membuat pengerjaan skrip tersebut dikerjakan di server sedangkan yang dikirimkan kepada *browser* adalah hasil proses dari skript tersebut yang sudah berbentuk *HTML*.

Keunggulan dari sifatnya yang *server side* adalah:

- a. Tidak diperlukan adanya *kompabilitas browser* atau harus menggunakan *browser* tertentu, karena *server*-lah yang akan mengerjakan skrip tersebut. Hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.
- b. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, misalnya hubungan ke dalam database. *Script* tidak dapat dilihat sehingga keamanan lebih terjaga.

2.1.5. Dreamweaver CS5

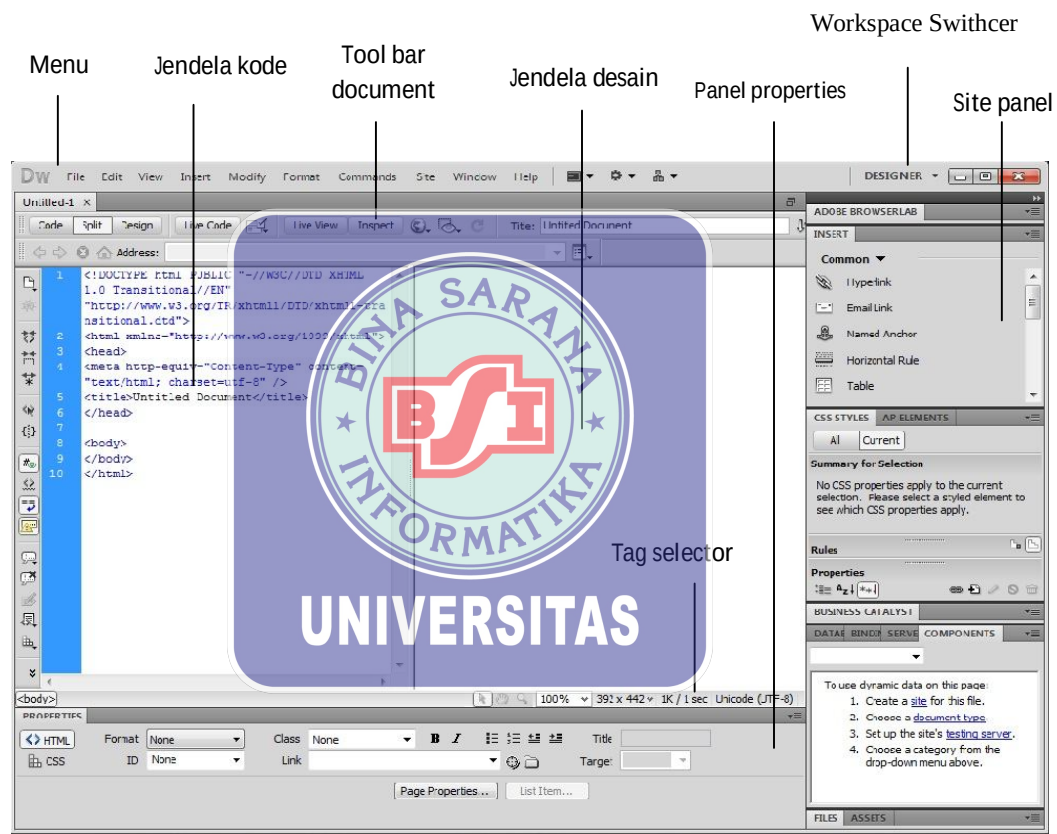
Pengertian Dreamweaver menurut MADCOMS (2011:13) Dreamweaver adalah "sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs halaman web". Dengan Program ini seorang *Web Designer* maupun *Web Programmer* akan mampu mengembangkan *situs* web karena dreamweaver mempunyai ruang kerja , fasilitas dan kemampuan yang mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu *situs web*. Versi terbaru dari Dreamweaver saat ini adalah Dreamweaver CS5.

Dreamweaver CS5 adalah editor yang lengkap yang dapat digunakan untuk membuat animasi sederhana yang berbentuk *layer*. Dengan adanya program

ini kita tidak akan susah-susah mengetik *script-script* format HTML, PHP, ASP maupun bentuk program lainnya.

Agar lebih jelas apa saja yang ada pada area kerja Adobe Dreamweaver

CS5, maka penulis mencoba menjabarkannya seperti berikut :



Gambar II.2. Tampilan Ruang kerja Dreamweaver CS5

Keterangan:

a. *Menu Bar*

Menu Bar, berada di paling atas jendela aplikasi Dreamweaver CS5, baris ini berisi tombol *workspace* (*workspace switcher*), menu dan aplikasi lainnya. *Menu Bar* ini akan memudahkan kita untuk merancang halaman web yang akan kita buat, *menu bar* memiliki pilihan perintah-perintah menu yang akan membantu kepraktisan dalam merancang halaman web.

b. *Toolbar Document*

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan Desain atau tampilan *Code*. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman web pada jendela *browser*.

c. *Jendela Desain*

Adalah lembar kerja tempat kita membuat dan mengedit desain halaman web.

d. *Workspace Swithcer*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) Dreamweaver CS5. Sebagai contoh mengubah tampilan *classic*, yaitu tampilan ruang kerja Dreamweaver versi sebelumnya.

e. *Site Panel*

Adalah kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara *default*, *Panel Group* berisi panel *Insert*, *CSS Style*, *Asset AP Element* dan *files*.



f. *Tag Selector*

Diletakkan dibagian bawah jendela dokumen, satu baris dengan status bar. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja Desain.

g. *Panel Propertis*

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela desain. Properti untuk satu objek dengan objek yang lainnya selalu berbeda-beda. Jendela ini tidak dapat diuraikan pada tampilan jendela *code*. Kita dapat mengatur properti masing-masing objek dalam panel *properties*. Untuk menampilkan atau menyembunyikan panel *properties* terdapat dua pilihan yaitu HTML dan CSS.

h. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. *Toolbar* ini hanya tampil pada jendela kode.

i. *Panel Insert*

Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media kedalam jendela dokumen. Masing-masing objek adalah suatu potongan kode HTML.

j. *Panel Siles*

Digunakan untuk mengatur *file-file* dan folder-folder yang membentuk situs web kita. Sebagai contoh mengcopy, memindah atau mengganti nama *file*.



Selain itu, panel files dapat digunakan juga untuk melakukan upload ke server, baik server *intranet* maupun *internet*. Panel *file* ditampilkan dalam panel group.

2.2. Teori Pendukung

2.2.1. Struktur Navigasi

Menurut Prihatna (2005:51) Struktur Navigasi adalah “Susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan link atau navigasi tiap halaman pada suatu situs web”. Struktur navigasi suatu situs web sangat di pengaruhi oleh tujuan dari situs web yang akan dibuat. Struktur Navigasi untuk merancang suatu website itu dibagi menjadi empat macam bentuk dasar, yaitu:

a. Struktur Navigasi *Linear*

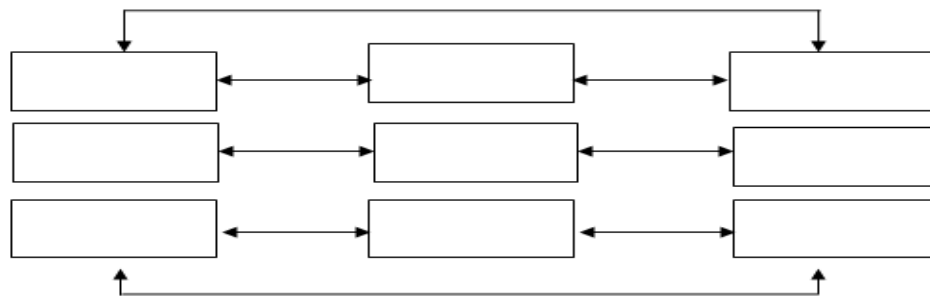
Struktur Navigasi *Linear* hanya memiliki satu rangkaian cerita yang berurut menurut ukurannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.



Gambar II.3. Struktur Navigasi Linear

b. Struktur Navigasi *Non Linear*

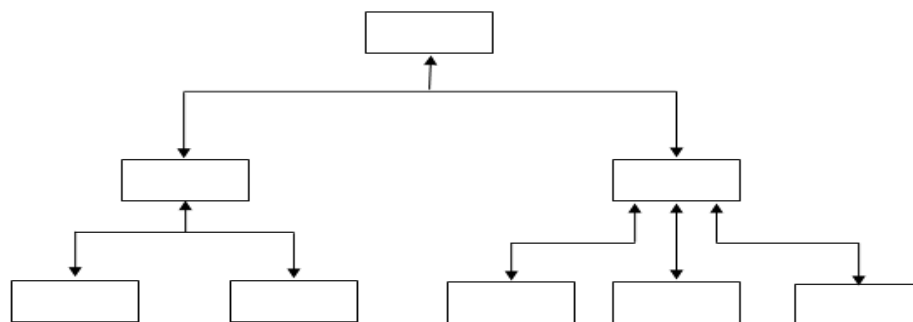
Struktur Navigasi *Non Linear* disebut juga struktur tidak berurut, ini merupakan pengembangan dari struktur *linear*



Gambar II.4. Struktur Navigasi Non Linear

c. Struktur Navigasi Hirarki

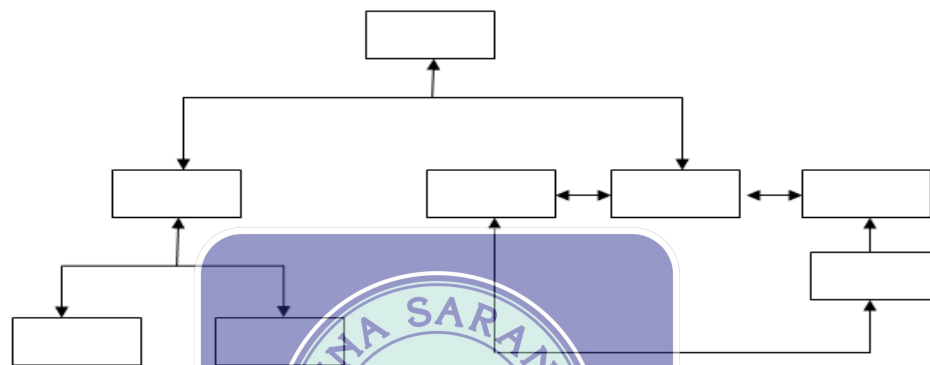
Ada yang menganggap bentuk *web pages* hirarki merupakan cara yang paling mudah serta pemahaman *logic* yang mudah diterima. Organisasi hirarki memudahkan pembaca untuk mengetahui posisi dari struktur yang dibuat. Ciri utama struktur navigasi hirarki ini adalah memiliki percabangan dengan *stavepage-stavepagenya*. Yang dapat kembali kehalaman utama. Bentuk hirarki merupakan bentuk peningkatan dari bentuk linear dengan menambahkan alternative cabang. Cabang *linear* yang pertama disebut dengan cabang lintasan utama. Bentuk ini lebih kompleks dari bentuk *linear*, struktur ini biasa disebut struktur bercabang, merupakan struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan data kriteria tertentu.



Gambar II.5. Struktur Navigasi Hirarki

d. Struktur Navigasi Campuran

Struktur Navigasi Campuran merupakan gabungan dari ketiga struktur sebelumnya. Struktur ini juga biasa disebut navigasi bebas, struktur ini banyak digunakan dalam pembuatan aplikasi multimedia sehingga dapat memberikan keinteraksian yang lebih tinggi.



Gambar 11.6. Struktur Navigasi Campuran

1. ERD (*Entity Relationship Diagram*)

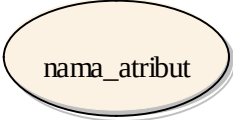
Menurut Rosa,dkk (2013:53) "ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional". Jika menggunakan OODBMS (*Object Oriented Database Management System*) maka perancangan ERD tidak perlu dilakukan. Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD).

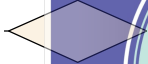
ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notaris seperti notaris Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh

Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notaris lain. Namun yang banyak digunakan adalah notaris dari Chen.

Menurut Simarmata (2007:107) menyatakan bahwa “derajat relasi adalah sejumlah entitas yang berhubungan dengan hubungan. Hubungan n -ary adalah format yang umum untuk derajat n . Kasus khusus adalah seperti *biner* dan *ternary*, dimana derajatnya adalah 2 dan 3 secara berturut-turut.

Participation Constraint menunjukkan batasan keterlibatan suatu entitas dalam suatu *relationship*. Jenisnya: *Total*, dan *Partial*. *Total* yaitu setiap entitas harus terlibat dalam suatu *relationship*. *Partial* yaitu tidak harus seluruh entitas terlibat dalam suatu *relationship*.

Simbol	Deskripsi
	ENTITAS / ENTITY Entitas merupakan data inti yang akan disimpan, baik tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penanaman entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
	ATRIBUT <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

	ATRIBUT KUNCI <i>PRIMER</i> <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa <i>id</i> ; kunci <i>primer</i> dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
	RELASI Relasi yang menghubungkan anatar entitas; biasanya diawali dengan kata kerja
	ASOSIASI / ASSOCIATION Penghubung antar relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.

Gambar II.7.

Simbol *ERD* (Entity Relationship Diagram)

Sumber : Rosa, dkk (2013:50)

2. Derajat Kardinalitas

Menurut Fathansyah (2007:77) menyatakan bahwa “Kardinalitas Relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”.

Kardinalitas diantara beberapa himpunan dapat berupa :

1. Satu Ke Satu (*One To One*)

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

2. Satu Ke Banyak (*One To Many*)

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

3. Banyak ke Satu (*Many To One*)

setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu 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.

4. Banyak ke Banyak (*Many To Many*)

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



3. Pengujian Perangkat Lunak

Menurut Rosa,dkk (2013:275) “Black-box testing yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

- a. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

