

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Menurut Mulyana (2011:5), *Web* merupakan salah satu fasilitas *hypertext* untuk menampilkan data yang dapat berupa teks, gambar, bunyi, animasi dan data multimedia lainnya dimana diantara data tersebut saling berhubungan satu sama lain”. Untuk memudahkan menampilkan data dan informasi pada *web* dapat digunakan *web browser* seperti *Microsoft Internet Explorer, Mozilla Firefox, google chrome* atau *Opera Browser*.

2.1.1. Website

1. Pengertian Website

Menurut Hidayat (2010:2), “*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Dapat disimpulkan *website* adalah sebuah tempat di internet, yang menyajikan informasi dengan berbagai macam format data seperti teks, gambar, bahkan video dan dapat diakses menggunakan berbagai aplikasi klien sehingga memungkinkan

penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi.

2. Internet

Menurut Iskandar (2010:1) "Internet (*Interconnection Networking*) adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia".

Media penghubung tersebut bisa melalui kabel, kanal satelit maupun frekuensi radio. Sehingga komputer-komputer yang terhubung tersebut dapat saling berkomunikasi. Setiap komputer yang terhubung dengan jaringan diberikan sebuah nomor yang unik dan berkomunitas satu sama lainnya dengan bahasa komunikasi yang sama. Bahasa komunikasi yang sama ini disebut *protocol*. *Protocol* yang digunakan di internet adalah TCP/IP (*Transmission Control Protocol /Internet Protocol*).

Internet awalnya merupakan suatu proyek yang dibentuk Departemen Pertahanan Amerika Serikat (*US Department of Defence*) pada tahun 1969. Dimulai dari suatu proyek yang dinamakan *Advance Research Project Agency Network* (ARPANET).

3. Web Browser

Menurut Sidik (2012:5), "Web browser adalah *software* yang digunakan untuk menampilkan informasi dari *server web*".

Program aplikasi ini mempunyai kemampuan untuk menampilkan suatu *web page* yang ditulis dalam format HTML. Dua contoh program aplikasi angkatan

pertama adalah *Lynx* dan *Mosaic*. Saat ini, *browser* yang paling banyak digunakan adalah *Internet Explorer*, *Mozilla Firefox* dan *Google Chrome*.

4. *Web Server*

Menurut Sidik (2010:6), “*Web server* adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web* dimana komputer ini akan melayani permintaan dokumen *web* dari kliennya”.

Untuk itu kita membutuhkan beberapa *server* yang mempunyai dokumen-dokumen media ke *Browser*. *Web browser* seperti, *Netscape*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* berkomunikasi melalui jaringan ke *Server web* dengan menggunakan HTTP (*Hypertext Transfer Protokol*). Halaman-halaman web yang dikirim oleh *web server* biasanya berupa file-file HTML dan CSS yang nantinya akan ditata oleh *browser* sehingga menjadi halaman-halaman web yang bagus dan mudah dibaca.

5. *WWW (World Wide Web)*

Menurut Iskandar (2012:4), ”*WWW* merupakan sebuah sistem dengan informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet *webserver* dan dipresentasikan dalam bentuk *hypertext*”.

Browser akan membuka sebuah dokumen setelah sebuah sub kalimat *hypertext* tertentu terpilih, pada suatu sistem yang menggunakan mouse, user tinggal meletakkan pointer mouse pada suatu kalimat *hypertext* dan setelah di klik maka informasi lainnya yang berhubungan dengan kalimat tersebut akan tampil. Pokok dokumen ini pun masih dimungkinkan lagi untuk disisipkan suatu *hypertext* yang menghubungkan ke dokumen lainnya.

6. HTTP (*Hypertext Transfer Protocol*)

Menurut Iskandar (2011:1) "HTTP adalah suatu protocol yang digunakan untuk mentransfer data antara web *server* ke web *browser* yang berformat HTML". Jika kita sedang menjelajahi web dan melihat tulisan seperti "*http/www.yahoo.com*", ini merupakan salah satu penggunaan *protocol* HTTP dalam web, dan dapat dipastikan bahwa kita akan dapat berhubungan banyak dengan *protocol* ini jika sedang mengakses web.

2.1.2. Bahasa Pemrograman

1. PHP (*Hypertext Preprocessor*)

Menurut Kustiyahningsih (2011:114), "PHP adalah skrip bersifat *server-side* yang di tambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat di integrasikan ke dalam HTML sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan *script* dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*".

Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis yang diambil dari database, merupakan hal yang mudah untuk diimplementasikan. Pada saat ini PHP sudah dapat berkomunikasi dengan berbagai *database* meskipun dengan kelengkapan yang berbeda-beda.

a. Tag PHP :

- 1) Menggunakan sintaks `<?php ... ?>`

Contoh:

```
<?php
```

```
echo" Ini merupakan cara pertama penulisan script PHP";
```

```
?>
```

b. Variabel PHP

Variabel digunakan untuk menyimpan data sementara dan nilainya bisa berubah-ubah setiap kali program dijalankan. Adapun aturan dalam penulisan variabel adalah sebagai berikut:

- a. Variabel di mulai dengan tanda \$
- b. Harus dimulai dengan huruf atau *underscore* (tidak boleh dimulai dengan angka).
- c. Mengenal perbedaan huruf besar dan huruf kecil (*Case sensitif*).

Contoh penulisan variabel yang benar misalnya \$buku, \$_buku, \$buku1 dan contoh penulisan variabel yang salah adalah \$1buku, \$-buku.

2. HTML (*Hyper Text Markup Language*)

Menurut Sibero (2011:19), “HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk pertukaran dokumen web”.

Menurut Sutarman (2012:163), “HTML (*Hypertext Markup Language*) adalah bahasa standar yang digunakan untuk pembuatan halaman web atau *word wide web*,

dengan *hypertext* dan informasi lain yang akan ditampilkan pada halaman web. Dokumen *hypertext* bisa berisi teks, gambar, dan tipe informasi lain seperti data file, audio, video, dan program *executeable*".

Berdasarkan pendapat para ahli yang dikemukakan diatas dapat ditarik kesimpulan bahwa HTML adalah bahasa yang sangat tepat dipakai untuk menampilkan informasi pada halaman web, karena HTML menampilkan informasi dalam bentuk *hypertext* dan juga mendukung sekumpulan perintah yang dapat digunakan untuk mengatur tampilnya informasi tersebut, sesuai dengan namanya, bahasa ini menggunakan tanda (*markup*) untuk menandai perintah-perintahnya.

2.1.3. Basis Data

1. Pengertian Database

Menurut Anhar (2010:45), "*Database* adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom. Struktur file yang menyusun sebuah database adalah *Data Record* dan *Field*".

2. MySQL

Menurut Kadir (2008:2), "MySQL adalah sebuah *software open source* yang digunakan untuk membuat sebuah *database*." Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Salah satu *database server* yang cukup dikenal saat ini adalah MySQL.

Berikut beberapa fungsi MySQL yang sering digunakan beserta deskripsi

singkat masing-masing fungsi tersebut:

a. `mysql_connect()`

Sebelum melakukan operasi apapun di MySQL, hal pertama yang harus dilakukan adalah membuka koneksi dengan MySQL. Fungsi yang digunakan untuk membuka koneksi dengan MySQL adalah `mysql_connect()`.

b. `mysql_close()`

Digunakan untuk mengakhiri koneksi ke MySQL. Fungsi yang digunakan adalah `mysql_close()`.

c. `mysql_create_db()`

Selain digunakan untuk melakukan koneksi ke MySQL, *username* dan *password* yang diberikan oleh *administrator server* adalah juga merupakan izin terbatas untuk membuat dan menggunakan *database* sendiri. Untuk membuat *database* yang hanya dapat diakses oleh *username* dan *password*, digunakan fungsi `mysql_create_db()` setelah koneksi ke MySQL dibuka.

d. `mysql_db_query()`

Perintah yang diberikan kepada MySQL untuk melakukan sesuatu disebut dengan *query*. *Query* dikirim kepada *database* yang sedang aktif.

e. `mysql_select_db()`

Jika suatu *website* menggunakan satu *database*, fungsi `mysql_select_db()` dapat digunakan memilih *database* yang sedang aktif. Dengan memilih *database* yang aktif tersebut dapat menghemat *resource*. Tidak perlu mengulang-ulang fungsi `mysql_select_db()` untuk mengirimkan *query* kepada *database* yang aktif.

f. `mysql_query()`

Fungsi `mysql_query()` digunakan untuk mengirimkan *query* kepada *database* aktif yang sudah dipilih dengan fungsi `mysql_select_db()`

g. `mysql_fetch_row()`

Fungsi `mysql_fetch_row()` digunakan untuk mengambil hasil *query* dari *database* sebagai data *numeric array*. Fungsi ini akan mengambil data baris per baris (per *record*) dari *database* secara berulang-ulang sampai nilai kondisi *query* menjadi *FALSE*. Nilai kondisi *query* diperoleh dari fungsi `mysql_query($query)`.

h. `mysql_fetch_array()`

Hampir sama dengan `mysql_fetch_row()`, `mysql_fetch_array()` digunakan untuk mengambil hasil *query* dari *database* sebagai data *associative array*.

i. `mysql_num_rows()`

Fungsi `mysql_num_row()` digunakan untuk menghitung jumlah *row* (blok data atau *record*) hasil *query*.

2.1.4. XAMPP

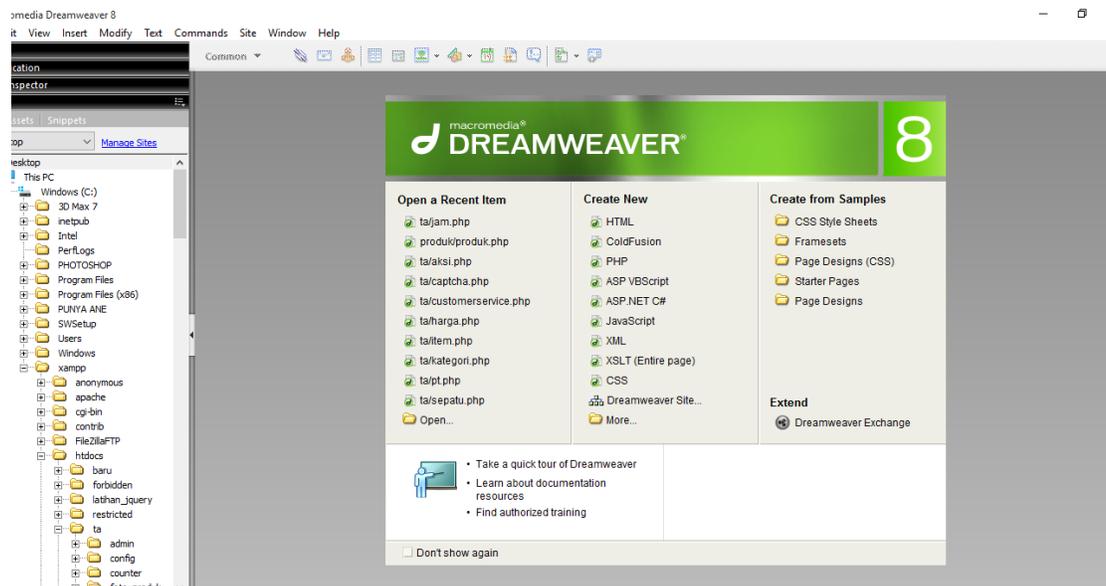
Menurut Nugroho (2011:2) “Xampp adalah suatu *bundle web server* yang populer digunakan karena mudah instalasinya. *Bundle program open source* tersebut berisi antara lain server *web Apache*, *interpreter PHP* dan basis data *MYSQL*”.

2.1.5. CSS (*Cascading Style Sheet*)

Menurut Castro (2012:119) “CSS adalah suatu bahasa yang berisi satu atau lebih tata cara, mengenai bagaimana suatu elemen ditampilkan dalam halaman web”. Terdapat pula *CSS Properties* yang berguna untuk mengontrol, seperti dasar penulisan, yaitu ukuran huruf dan warna. CSS juga memiliki beberapa properties yang dinamis, yang membuat suatu item bisa tampil atau tidak, dan digunakan untuk membuat *drop-down* dan komponen interaktif lainnya. Hal lain, CSS bisa dibuat diluar halaman web dan dijalankan pada semua halaman web dalam waktu yang bersamaan, karena CSS dibuat agar fleksibel, *powerfull* dan efisien.

2.1.6. Adobe Dreamweaver CS4

Menurut Wahana Komputer (2009:2), Adobe Dreamweaver CS4 merupakan salah satu aplikasi paling populer yang digunakan untuk membangun *website*. Dreamweaver memberikan fasilitas pengeditan HTML secara visual. Aplikasi ini menyertakan berbagai fasilitas dan teknologi pemrograman *web* terkini seperti HTML, CSS, dan *java script*. Aplikasi ini juga mendukung pemrograman *Script Server Side* seperti PHP, *Active Server Page* (ASP), ASP.NET, ASP *JavaScript*, ASP *VBScript*, *ColdFusion*, dan *Java Server Page* (JSP).

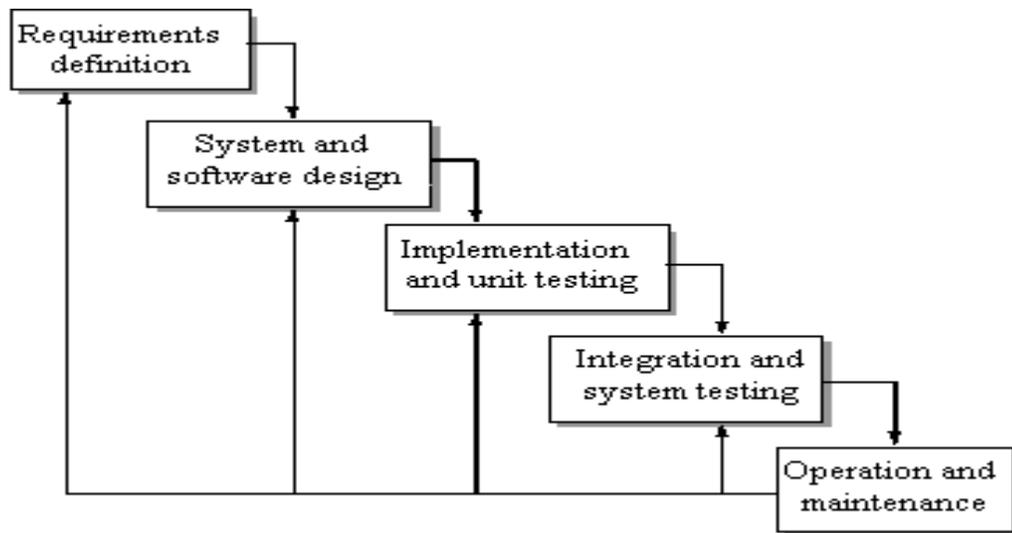


Sumber : Wahana Komputer (2009:7)

Gambar II.1 Tampilan Awal Adobe Dreamweaver CS8

2.1.7. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *waterfall*. Model *waterfall* merupakan sebuah metode pengembangan *software* yang bersifat skematis. Menurut Sommerville (2007:66) model *waterfall* terbagi menjadi beberapa tahapan yaitu:



Sumber: Sommerville (2007:66)

Gambar II.2. Model Waterfall

1. Analisis dan definisi kebutuhan: Layanan, batasan, dan tujuan dari sistem ditetapkan melalui konsultasi dengan pengguna sistem. Semua itu didefinisikan secara *detail* dan bertindak sebagai spesifikasi sistem.
2. Desain sistem dan *software*: Proses desain sistem membagi kebutuhan menjadi *hardware* atau *software*. Ini menetapkan hampir seluruh perancangan sistem. Desain *software* melibatkan pengidentifikasian dan penggambaran mengenai pemisahan dasar sistem *software* dan hubungannya.
3. *Implementation and unit testing*: Dalam tahap ini, desain *software* adalah menyadari sebagai kumpulan program atau satuan program. *Unit testing* melibatkan verifikasi bahwa setiap unit telah mencapai spesifikasinya.
4. *Integration and system testing*: Satuan program atau kumpulan program diintegrasikan dan di tes sebagai sistem yang telah selesai, untuk menjamin

bahwa kebutuhan *software* telah terpenuhi. Setelah pengetesan, sistem *software* dikirimkan kepada pelanggan.

5. *Operation and maintenance*: Biasanya, ini adalah bagian siklus hidup *software* yang paling lama. Sistem di-*install* dan dimasukkan kedalam penggunaan. Pemeliharaan melibatkan pembenaran kesalahan yang tidak ditemui dalam tahap awal siklus, meningkatkan implementasi satuan sistem, dan meningkatkan layanan sistem sehubungan ditemukannya kebutuhan baru.

2.2. Teori Pendukung

2.2.1. Struktur Navigasi

Menurut Prihatna (2009:51) “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 dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan suatu situs *web*. Bentuk dasar dari struktur navigasi adalah sebagai berikut:

1. Satu alur (*Linear*)

Linear merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut. Dengan kata lain struktur ini hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang ada di

tampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya. Salah satu yang terpenting dari struktur ini adalah tidak diperkenankan terjadinya percabangan.

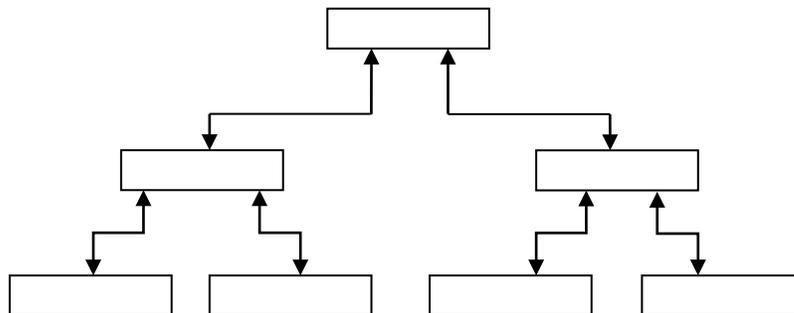


Sumber: Prihatna (2005:51)

Gambar II.3. Struktur Navigasi *Linear*

2. Hirarki (*Hierarchical*)

Struktur hirarki atau percabangan ini untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya. Yang terpenting dari struktur ini tidak diperkenankan adanya tampilan secara *linear*.

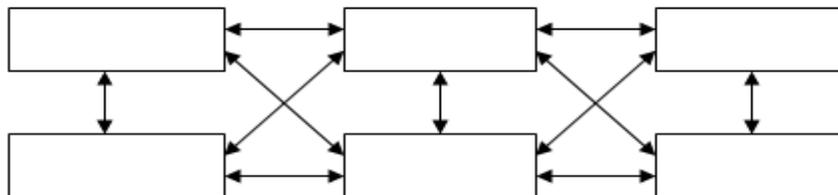


Sumber: Prihatna (2005:51)

Gambar II.4. Struktur Navigasi Hirarki

3. Struktur Navigasi *Non Linear*

Struktur *Non Linear* (tidak berurut) merupakan pengembangan dari struktur navigasi *Linear*. Pada struktur ini diperkenankan membuat struktur bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses ke semua halaman manapun. Percabangan yang dibuat pada struktur *Non Linear* ini berbeda dengan percabangan pada struktur *Hierarchi*, karena pada percabangan *Non Linear* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master Page* dan *Slave Page*.



Sumber: Prihatna (2005:51)

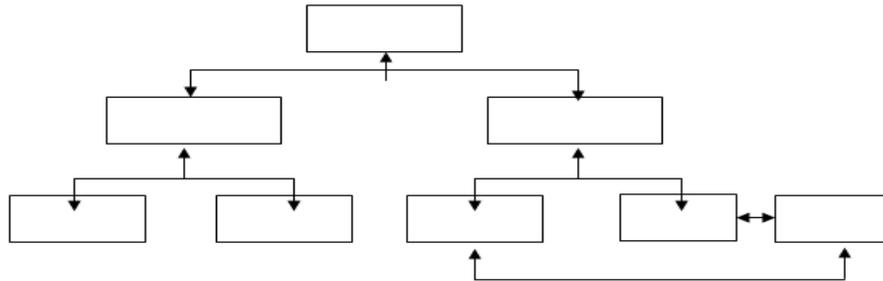
Gambar II.5. Struktur Navigasi *Non Linear*

4. Campuran (*Composite*)

Campuran atau disebut juga struktur penjejakan bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear* dan *Hierarchi*. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila dalam percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut.

Setiap struktur peta penjejakan seperti yang baru di bahas mempunyai fungsi dan tujuan tersendiri, tidak ada yang lebih baik atau lebih buruk. Penggunaan peta

penjejakan bergantung kepada kebutuhan dan tujuan dari *web* yang hendak dibuat. Semakin kompleks peta penjejakan yang digunakan, maka semakin sulit pula pembuatan page dari peta penjejakan tersebut.



Sumber: Prihatna (2005:51)

Gambar II.6. Struktur Navigasi *Composite*

2.2.2. ERD (*Entity Relationship Diagram*)

Menurut Al-Bahra (2005:142), “Diagram relasi entitas merupakan suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

Entity Relationship Diagram juga merupakan suatu model jaringan yang menggunakan susunan data yang tersimpan secara sistem atau teknik menggambar suatu skema *database* dimana setiap komponen yang terlibat dalam ERD memiliki atribut masing-masing yang mempresentasikan fakta dari dunia nyata yang sedang di tinjau.

1. Komponen ERD

a. Entitas/*Entity*

Entitas/*Entity* adalah suatu tempat atau objek untuk menyimpan data. Contoh:

Entitas buku untuk menyimpan atribut mengenai buku (judul buku, kode buku, pengarang).

- b. Relasi/*Relationship* adalah hubungan yang terjadi antara satu entitas atau lebih. Contoh: Entitas buku dan Entitas Pengarang memiliki hubungan "ditulis", yang artinya, pengarang menulis buku, dan buku ditulis pengarang. Selain itu, terdapat relasi "*one to one*", "*one to many*", dan "*many to many*" dalam *entity*.
- c. Atribut/*Attribute* adalah kumpulan elemen data yang membentuk suatu entitas. Contoh: Entitas pengarang memiliki atribut nama, alamat, no telpon, dsb. Atribut juga sering disebut dengan field atau kolom dalam suatu ERD.

Jenis-jenis atribut :

1. *Key*: Atribut yang digunakan untuk menentukan suatu *entity* secara unik.
2. Atribut *Simple*: Atribut yang bernilai tunggal.
3. Atribut *Multivalued*: Atribut yang memiliki sekelompok nilai untuk setiap instan *entity*.
4. Atribut *Composite*: Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.
5. Atribut *Derivatif* : Suatu atribut yang dihasilkan dari atribut yang lain.

2. Derajat *Relationship*

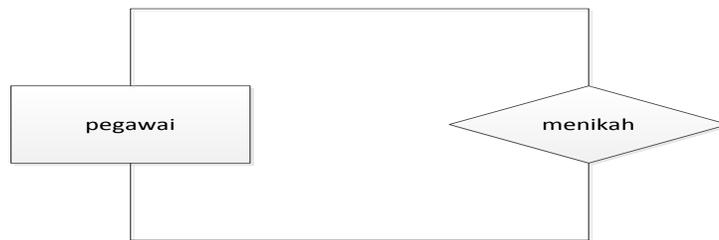
Menurut Al-Bahra (2005:144) "*Relationship Degree* atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*."

Derajat *relationship* yang sering dipakai di dalam ERD sebagai berikut :

a. *Unary Relationship*

Unary Relationship adalah model *relationship* yang terjadi antara *entity* yang berasal dari *entity* set yang sama. Model ini juga sering disebut sebagai *Recursive Relationship* atau *Reflective Relationship*.

Contoh:



Sumber: Al-Bahra (2005:142)

Gambar II.7. Diagram Relationship Unary

b. *Binary Relationship*

Binary Relationship adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

Contoh:



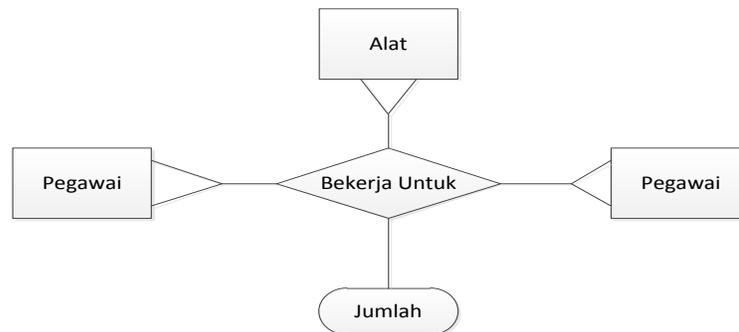
Sumber : Al-Bahra (2005:142)

Gambar II.8. Diagram Relationship Binary

c. *Ternary Relationship*

Ternary Relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara serentak.

Contoh:



Sumber : Al-Bahra (2005:142)

Gambar II.9. Diagram Ternary Relationship

3. Kardinalitas Relasi

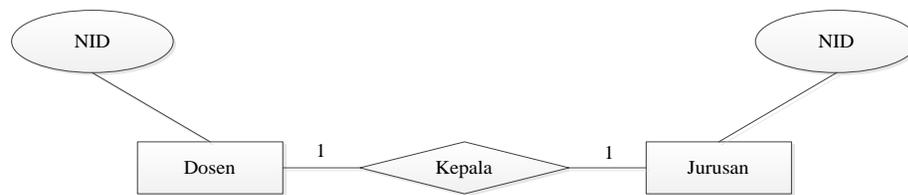
Menurut Al-Bahra (2005:147) “Kardinalitas Relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lain.”

Terdapat 3 macam kardinalitas relasi yaitu sebagai berikut:

a. Relasi Satu ke satu (*One to One*)

Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

Contoh:



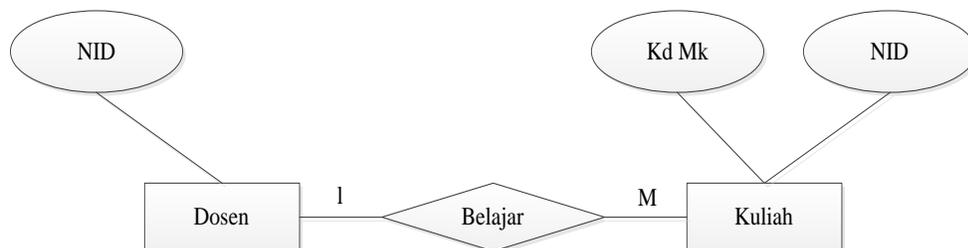
Sumber: Al-Bahra (2005:147)

Gambar II.10. Relasi *One to One*

b. Relasi Satu ke Banyak atau Banyak ke Satu (*One to Many* atau *Many to One*)

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

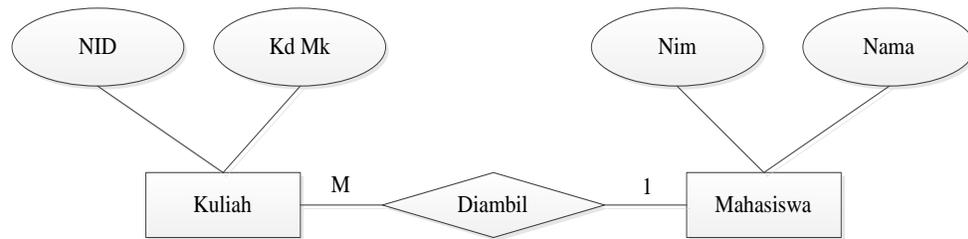
Contoh:



Sumber : Al-Bahra (2005:147)

Gambar II.11. Relasi *One to Many*

Contoh:



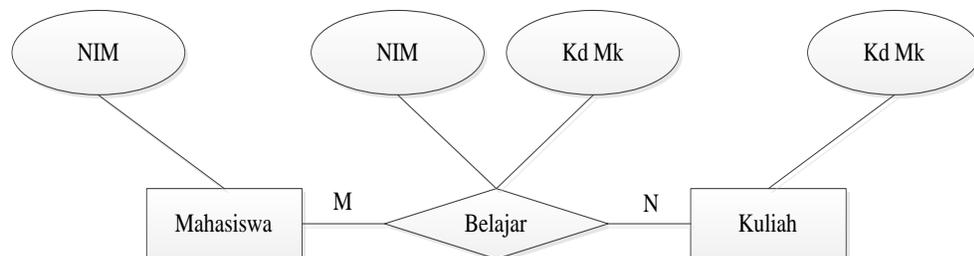
Sumber : Al-Bahra (2005:147)

Gambar II.12. Relasi Many to One

c. Relasi Banyak ke Banyak (*Many to Many*)

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

Contoh:



Sumber : Al-Bahra (2005:147)

Gambar II.13. Relasi Many to Many

4. LRS (*Logical Record Structure*)

LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tabel dan *Foreign Key* (FK).

2.2.3. Pengujian Web

Menurut Pressman (2010:10) *Black Box Testing* atau pengujian kotak hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak, artinya teknik *black box testing* memungkinkan untuk mendapat *set* kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black box testing* bukan merupakan alternatif dari pengujian *white box testing*. Tetapi merupakan pendekatan yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan dari pada metode *white box testing*.

Black box testing berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi.