

BAB II

LANDASAN TEORI

2.1. Konsep Dasar *Web*

2.1.1. *Website*

1. Pengertian *Website*

Menurut Arief (2011:7) “*Web* adalah salah satu aplikasi yang berisikan dokumen – dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan *protocol* HTTP (*Hypertext Transfer Protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*”.

Sedangkan menurut Sibero (2013:11) “Pengertian *Website World Wide Web* (W3) atau yang dikenal juga dengan istilah *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan *internet*”.

Jenis-jenis *web* berdasarkan sifatnya atau *style*-nya menurut Hidayat (2010:3), sebagai berikut:

a. *Website* Dinamis

Merupakan sebuah *website* yang menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang di gunakan antara lain PHP, ASP, .NET dan memanfaatkan database MySQL. Misalnya *website* www.artikel.com, www.detik.com, www.polinpdg.ac.id, dan lain-lain.

b. *Website Statis*

Merupakan sebuah *website* yang kontennya sangat jarang di ubah. Bahasa pemrograman yang di gunakan adalah HTML, dan belum memanfaatkan database. Misalnya *web profile* organisasi dan lain-lain.

2. *Pengertian Internet*

Menurut Shalahuddin dan Rosa (2010:3) ”*Internet* adalah sekumpulan jaringan berbeda yang saling terhubung bersama sebagai satu kesatuan dengan menggunakan berbagai macam *protocol* ”. TCP/IP adalah *protocol* yang paling banyak digunakan di internet. *Protocol* TCP/IP merupakan cara standar untuk memaketkan dan mengamatkan data komputer (sinyal elektronik) sehingga data tersebut dapat dikirim ke komputer terdekat atau keliling dunia dan tiba dalam waktu yang cepat tanpa rusak atau hilang.

Istilah-istilah umum yang sering ditemukan didalam internet antara lain adalah:

a. *Word Wide Web* (WWW)

Merupakan sebuah sistem yang *interlinked* (kumpulan *link* atau saluran yang saling terhubung), akses dokumen *hypertext* melalui internet.

b. *Hypertext Transfer Protocol* (HTTP)

Merupakan aturan pengiriman informasi yang berupa *hypertext* (teks pada komputer yang memungkinkan user saling mengirimkan (*request-request*)). HTTP adalah protokol atau aturan standar untuk mengakses dokumen HTML (*Hypertext Markup Language*) pada *web*. HTTP dijalankan dengan cara klien mengirimkan permintaan klien dengan respon yang diminta iklan.

c. *Universal Resource Locator (URL)*

Merupakan alamat lokasi dari file yang ingin diakses. Pengaksesan *web* di internet menggunakan URL dengan menggunakan protocol HTTP. *Port* standar dari *protocol* adalah 80, jika *port* yang di gunakan bukan *port standart*, maka *port* akan ditulis pada URL misalnya <http://localhost:8080/>. Dengan menggunakan URL kita dapat mengakses file yang menghasilkan tampilan *web* pada komputer klien dengan menggunakan *web browser*.

d. *Web Client (Browser)*

Merupakan salah satu perangkat lunak yang dijalankan pada komputer pemakai (*user*) yang menampilkan dokumen atau informasi *web* yang diambil dari *web server*. Contoh *web browser* adalah *Internet Explorer*, *Google Chrome*, *Mozilla firefox*.

e. *Web Server*

Merupakan salah satu perangkat lunak yang dijalankan pada komputer server dan berfungsi agar dokumen *web* yang disimpan di server dapat diakses oleh pemakai (*user*) internet seperti misalnya xampp.

f. *Homepage / Situs Web*

Merupakan informasi di *word wide web* yang disimpan dalam file yang berbeda-beda sebagai halaman *web*. *Homepage* merupakan halaman awal dari sebuah situs *web*.

g. *Web service*

Merupakan suatu sistem yang menyediakan pelayanan yang dibutuhkan oleh klien. Klien dari *web service* tidak hanya berupa aplikasi *web*, tapi juga bisa

sebuah aplikasi *enterprise*. Jadi *web service* tidak sama dengan *web server*, bahkan sebuah aplikasi *web* pada *web server* dapat menjadi klien dari *web service*. Pelayanan yang diberikan *web service* bisa berupa XML yang berisi data yang dibutuhkan klien.

h. *Web Hosting*

Merupakan organisasi yang menyediakan tempat di server-nya untuk perseorangan atau organisasi meletakkan semua file *website* nya dan menyediakan konektivitas dengan internet agar dapat di akses melalui internet.

3. Pengertian *E-Commerce*

Menurut Hidayat (2010:5) “Perdagangan Elektronik (*E-Commerce/ Electronic Commerce*) adalah bagian dari *e-lifestyle* yang memungkinkan transaksi jula beli dilakukan secara *online* dari sudut tempat mana pun”.

2.1.2. Bahasa Pemrograman

Bahasa pemrograman adalah notasi untuk mengekspresi intruksi – intruksi yang harus dilakukan computer. Bahasa pemrograman adalah media komunikasi antara manusia dan mesin. Bahasa pemrograman yang ampuh lebih dari sekedar sarana untuk memerintah komputer untuk melakukan tugas. Bahasa juga bertindak sebagai keranjang kerja dalam mengorganisasi gagasan penyelesaian ke komputer (Bambang, 2008:73).

1. CSS (*Cascading Style Sheets*)

Menurut Shalahuddin dan Rosa (2010:45) “CSS atau *Cascading style Sheets* adalah suatu fasilitas untuk mempermudah pemeliharaan sebuah halaman *web*, dengan menggunakan CSS sebuah halaman *web* dapat diubah tampilannya tanpa harus mengubah dokumen HTML-nya”.

CSS dapat ditulis dengan huruf besar maupun huruf kecil seperti halnya HTML. Karena CSS *non-case sensitive* (tidak membedakan huruf besar dan huruf kecil). CSS dapat digunakan langsung didalam tag-tag dokumen HTML dalam atribut *style*, tetapi cara ini tidak disarankan penggunaannya karena saat diperlukan pergantian tampilan *web* halaman *web*, maka dokumen HTML harus diubah pada tag-tag-nya.

2. JavaScript

Menurut Shalahuddin dan Rosa (2010:65) “JavaScript adalah bahasa pemrograman yang dapat memegang kontrol aplikasi yang berbasis pada bahasa pemrograman java, namun *JavaScript* bukanlah bagian teknologi Java dari Sun”. Java merupakan bahasa pemrograman berorientasi objek murni sedangkan *JavaScript* digunakan secara *procedural*.

JavaScript dikembangkan oleh *Netscape*, sebagai bahas pemrograman “sederhana” karena tidak dapat digunakan untuk membuat suatu aplikasi ataupun applet. Namun, dengan *JavaScript*, kita dapat membuat sebuah halaman *web* yang interaktif dengan mudah.

JavaScript merupakan bahasa yang *sensitive* seperti halnya bahasa pemrograman Java yaitu membedakan penulisan dengan huruf kecil dan huruf besar

memiliki arti yang berbeda. Pertama kali diperkenalkan oleh *Netscape*, ini bernama *LiveScript* yang berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator 2*, browser tersebut merupakan browser yang sangat populer pada saat itu. Kemudian sejalan dengan gencarnya kerjasama antara *Netscape* dan *Sun* (Pengembangan bahasa pemrograman Java) pada masa itu, maka pada tanggal 4 desember 1995 *Netscape* memberikan nama *JavaScript* kepada bahasa tersebut.

Setiap bahasa pemrograman pasti mempunyai kelebihan dan kekurangannya, begitu pun dengan *JavaScript*, adapun kelebihan pemrograman *JavaScript* antara lain sebagai berikut :

- a. Cepat, Karena diletakan di HTML, dan dapat langsung dicoba di browser.
- b. Mudah, karena memiliki sedikit *sintaks*.
- c. Terbuka (*Platform Independent*)
- d. Ukurannya sangat kecil dibandingkan dengan program-program java.

Sedangkan kekurangan dari bahasa pemrograman *JavaScript* ini antara lain seperti di bawah ini :

- a. Tidak dapat membuat program aplikasi secara sendiri.
- b. Objek yang dikelola *JavaScript* sangat terbatas.

Script dapat disalin atau di *copy* dari *web browser*, sehingga setiap orang dapat menggunakan program *JavaScript* yang telah kita buat.



3. JQuery

Menurut Hakim (2014:03) “JQuery adalah *javascript* library, kumpulan kode atau fungsi *Javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*”. Secara standar, apabila kita membuat kode Javascript, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami.

4. HTML

Menurut Sibero (2013:19) “*HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*”. Struktur dokumen HTML terdiri dari *tag* penutup. HTML versi 1.0 dibangun oleh W3C, dan terus mengalami perkembangan. Sampai saat ini HTML terakhir adalah versi 5.0. Struktur dokumen HTML sebagai berikut:

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

Struktur diatas adalah satu kesatuan yang harus ada dalam setiap dokumen HTML. Dokumen HTML terdiri dari komponen yaitu *tag*, elemen, dan atribut. Berikut adalah penjelasan dari masing – masing komponen tersebut.

Tag adalah tanda awal < dan tanda akhir > yang digunakan sebagai pengapit suatu elemen. *Tag* pada elemen pembuka diawali dengan tanda < dan diakhiri dengan tanda >.

Elemen adalah nama penanda yang diapit oleh *tag* yang memiliki fungsi dan tujuan tertentu pada dokumen HTML. Elemen dapat memiliki elemen anak dan juga nilai. Elemen anak adalah suatu elemen yang berada didalam elemen pembuka dan elemen penutup induknya. Nilai yang dimaksud adalah suatu teks atau karakter yang berada diantara elemen pembuka dan elemen penutup.

Atribut adalah properti elemen yang digunakan untuk mengkhususkan suatu elemen. Elemen dapat memiliki atribut yang berbeda pada tiap masing – masingnya.

5. PHP (*Personal Home Page*)

Menurut Kadir (2009:2) "PHP (*Personal Home Page*) adalah sebuah piranti yang biasa di pakai untuk membuat aplikasi *web*". Aplikasi *web* adalah aplikasi yang dijalankan melalui *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, anda bisa menampilkan *database* ke halaman *web*. Pada prinsipnya PHP mempunyai nilai yang sama dengan skrip – skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *comment 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* hasil proses script 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*.
- c. *Script* tidak dapat dilihat sehingga keamanan lebih terjamin.

Untuk menjalankan sistem PHP dibutuhkan tiga komponen, yaitu:

- a. *Web browser*, karena termasuk bahasa pemrograman *server-side*.
- b. Program PHP, program yang memproses *script* PHP.
- c. *Database server*, yang berfungsi untuk mengolah *database*.

Dalam membuat program PHP, yang dibutuhkan adalah memulai dengan *start tag* dan *end tag*, yaitu perintah awal dan perintah akhir ada beberapa cara bagaimana untuk menuliskan *start tag* dan *end tag*.

- 1) `<? Skrip PHP Anda ada disini ?>`
- 2) `<?PHP Skrip PHP Anda ada disini ?>`
- 3) `<% Skrip PHP Anda ada disini %>`
- 4) `<SCRIPT language =”PHP”>Skrip PHP Anda ada disini </SCRIPT>`

Proses-proses umum yang dipakai dalam PHP :

- a. Proses pencetakan pada layar pengguna

Ada 2 cara pencetakan pada PHP yaitu menggunakan perintah `echo”XXX”`

atau cara `printf(“XXX”);`

- b. Pengkondisian diartikan sebagai suatu bentuk logika yang menganalogikan dengan suatu kejadian atau beberapa kejadian. Kondisi yang digunakan, yaitu:

```

If...elseif...else
If (kondisi_1){
Pernyataan1 }
Elseif (kondisi_2){
Pernyataan 2}
Else{
Pernyataan 3}

```

- c. *Session*

Session adalah semacam register variable yang ada selama *browser* yang mengakses nilai dari *session* tersebut belum tertutup. Perintah yang sering dipakai dalam *session* adalah:

- 1) *Session_start*

Adalah perintah untuk memulai *session* juga memanggil semua variable yang didaftarkan dalam *session*. Sintaks: *session_start()*;

- 2) *Session_register*

Adalah suatu perintah untuk mendaftarkan sebuah variabel ke dalam *session* sehingga dapat membuat variabel ini dapat diakses dari halaman mana saja selama dipanggil perintah *session_start()*. Sintaks : *session_register*("nama variable");



6. My SQL

Menurut Rossa dan shalahuddin (2015:46) "SQL" (*Structure Query Language*) adalah Bahasa yang digunakan untuk mengelola data pada RDBMS. MySQL adalah *database* yang menghubungkan script PHP menggunakan perintah *query* dan *escaps character* yang sama dengan PHP. MySQL mempunyai tampilan client yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengijinkan proses yang boleh anda lakukan (kurniawan 2008:8). Untuk masuk ke dalam database disediakan *User default*, yaitu robot dengan *password maxikom* (*password* robot dibuat pada saat anda melakukan *instalasi Appserv*).

Dalam pengembangan *website* selain menggunakan beberapa bahasa pemrograman, digunakan juga beberapa tools atau software pendukung untuk membuat program, beberapa diantaranya adalah :

1. *Macromedia Dreamweaver 8*

Menrut Ramadhan (2007:2), "*Macromedia dreamweaver 8*, atau biasa di sebut *Dreamweaver 8* adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman *web*". Dengan menggunakan *Dreamweaver 8*, ketika membuat sebuah halaman web, tidak perlu mengetik kode – kode HTML atau kode – kode lainnya secara manual. Cukup melakukan klik beberapa kali, maka *simsalabim*, halaman *web* yang diinginkan sudah jadi.



Sumber: Ramadhan (2007:2)

Gambar II.1
Jendela Start Page Macromedia Dreamweaver 8



Sumber: Ramadhan (2007:2)

Gambar II.2
Tampilan Lembar Kerja Macromedia Dreamweaver 8

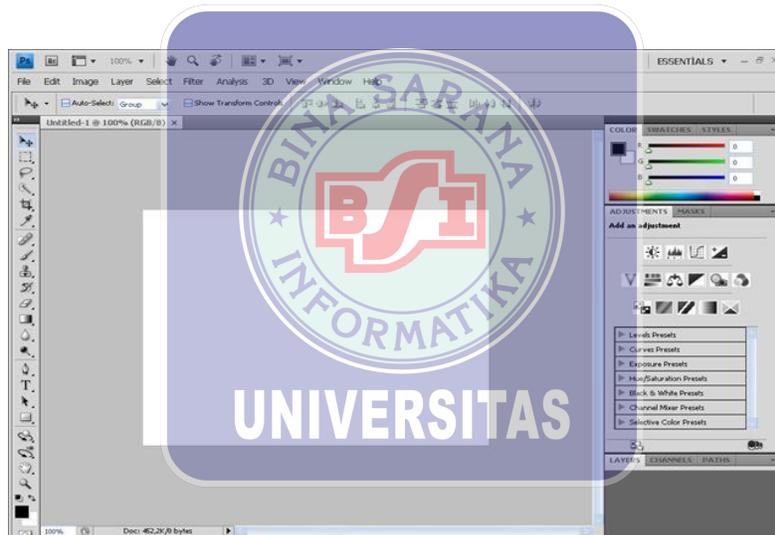
Keterangan gambar :

- a. **Insert Bar** digunakan untuk memasukan atau membuat berbagai macam objek ke dalam halaman *web* yang sedang dibuat pada *Dokumen Window*.
- b. **Menu Bar** berisi berbagai macam menu untuk mengatur halaman web yang sedang dibuat. Misalnya menu File, Edit, View, dan lain-lain.
- c. **Proverty Inspector** berisi berbagai macam atribut dari elemen yang sedang terpilih dalam *Dokumen Window*. Misalkan anda sedang mengetik sebuah table pada bagian *Dokumen Window* maka *property Inspector* akan menampilkan berbagai macam atribut dari tabel tersebut. Anda juga dapat mengubah nilai – nilai atribut pada bagian ini.
- d. **Panel groups** berisi berbagai macam *panel*. Setiap panel digunakan untuk mengatur hal – hal yang spesifik. Misalnya panel CSS digunakan untuk mengatur CSS. *Panel Files* digunakan untuk mengatur file- file, dan sebagainya.
- e. **Document Window** adalah bagian yang merupakan area kerja pembuatan *web*. Di sinilah proses mendesain dan membuat halaman *web* dilakukan. Ada 3 tampilan *Documen Window*, yaitu *Code View*, *Code and Design View* dan *Design View*.
- f. **Documen Toolbar** memiliki berbagai macam tombol yang dapat digunakan untuk mengatur tampilan *Document Window*.



2. Adobe Photoshop CS4

Menurut Hidayatulloh (2009:9) “*Adobe Photoshop CS4* adalah aplikasi *professional* untuk mengola gambar atau foto”. Aplikasi ini di lengkapi dengan berbagai macam *tool* dan *palette* untuk menampilkan dan meng-*edit* gambar. Fitur – fitur baru dalam *Adobe Photoshop CS4*, diantaranya adalah *Adjustment panel*, *Masking panel*, *Management Asset* dengan *Bridge*, proses cetakan lebih fleksibel, terintegrasi dengan *lightroom* kemampuan pada 3D dan *panning*, serta *zoom* dengan GPU akselerasi.



(Sumber:Hidayatulloh (2009:10))

Gambar II.3
Lembar Kerja *Photosop CS4*

- a. ***Document Window*** adalah tempat untuk menampilkan, meng-*edit*, dan menambah *element* gambar.
- b. ***Aplication Bar*** adalah bar pada bagian atas aplikasi *Photosop CS4* yang berisi *Workspace Switcher* (tombol pengubah tampilan), menu dan kontrol aplikasi yang lain.

- c. **Tools Panel** ketika anda menjalankan *Adobe Photoshop CS4*, *tool panel* terlihat ditempatkan di sisi sebelah kiri layar. Beberapa *tool panel* memiliki pilihan lain dalam *options bar*.
- d. **Icon collapse** adalah *icon* yang digunakan untuk mengaktifkan *dacker panel*.
- e. **Menu Bar** adalah sebaris menu untuk mengorganisir menu – menu yang lainnya.
- f. **Options Bar** dapat berupa tombol, *icon* atau menu pilihan untuk melengkapi *tool* yang ada klik pada Tools Panel.

3. PhpMyadmin

Menurut Sadeli (2013:10), “PhpMyadmin adalah sebuah *software* yang berbentuk seperti halaman situs yang terdapat pada *web server*”. Fungsi dari halaman ini adalah sebagai pengendali database MySQL sehingga Pengguna MySQL tidak perlu repot untuk menggunakan perintah – perintah SQL. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman phpMyadmin.



4. Xampp

Menurut Wicaksono (2008:7), “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan preview sehingga dapat memodifikasi *website* tanpa harus online atau terakses dengan internet.

2.1.3. Basis Data

Menurut Shalahuddin dan Rosa (2013:43) “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

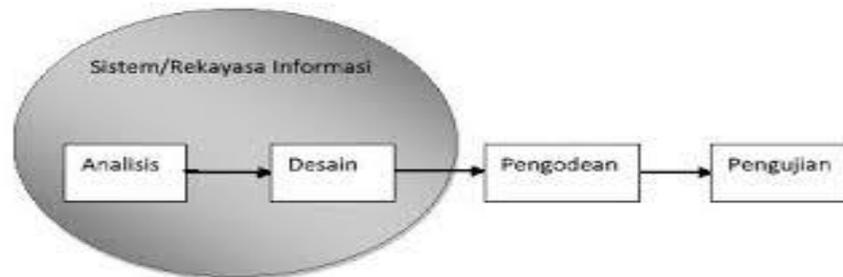
2.1.4. Model Perangkat Lunak

Menurut Shalahuddin dan Rosa (2013:2) Perangkat lunak (*software*) adalah “program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Menurut Shalahuddin dan Rosa (2013:26) menyimpulkan bahwa :

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses pengembangan atau pengubah suatu system perangkat lunak dengan menggunakan model – model dan metodologi yang digunakan orang untuk mengembangkan system – system perangkat lunak sebelumnya (berdasarkan *best practice* atau cara – cara yang sudah teruji).

Menurut Shalahuddin dan Rosa (2013:28) Model SDLC air terjun (*waterfall*) sering juga disebut “model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah model air terjun :



Sumber : Shalahuddin dan Rosa (2013:29)

Gambar II.4
Ilustrasi Model Waterfall

1. Analisa kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, 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.

2. Pembuatan kode program

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

3. Pengujian

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

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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bias 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 sfesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Pendukung

2.2.1. Struktur Navigasi

Menurut Fauzi (2008:3) “Struktur navigasi adalah struktur atau alur dari suatu program”. Sebelum membuat suatu aplikasi maupun *website* sebaiknya kita menentukan struktur navigasi terlebih dahulu. Ada empat macam bentuk dasar dari struktur navigasi yang biasa digunakan dalam pembuatan aplikasi, yaitu:

1. Struktur Navigasi Linier

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis inia dalam satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

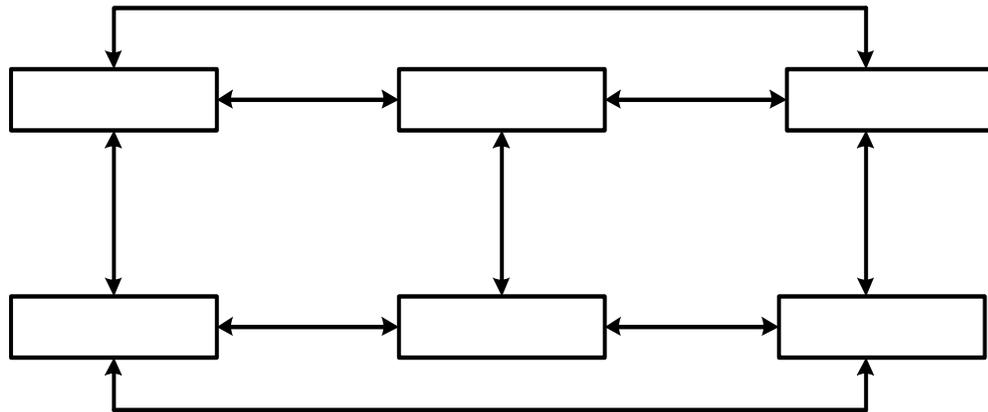


Sumber : Fauzi (2008:4)

Gambar II.5
Struktur Navigasi *Linier*

2. Struktur Navigasi Non-Linier

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*.



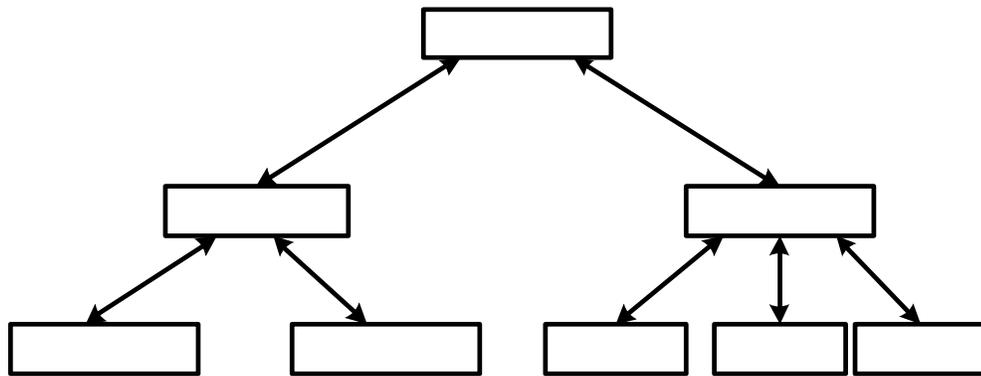
Sumber : Fauzi (2008:4)

Gambar II.6

Struktur Navigasi Non Linier

3. Struktur Navigasi Hirarki

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan criteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya.

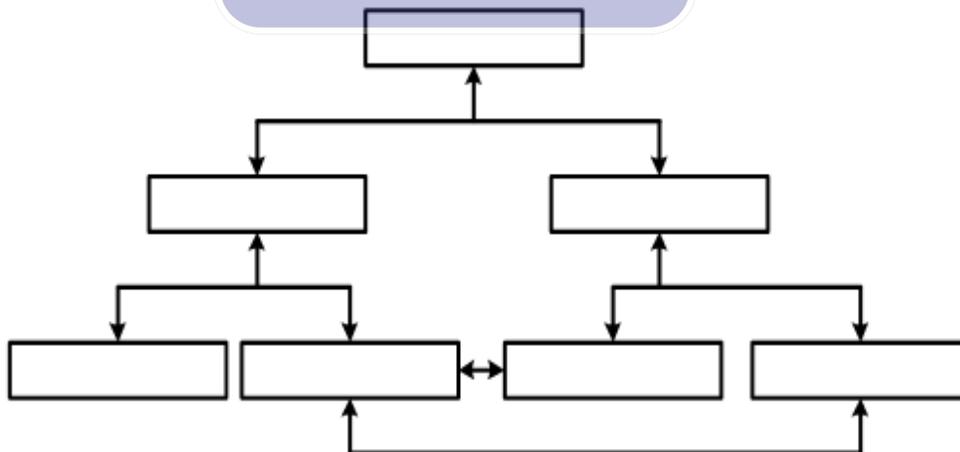


Sumber : Fauzi (2008:4)

Gambar II.7
Struktur Navigasi Hierarki

4. Struktur Navigasi Campuran

Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat digunakan dalam pembuatan website sehingga dapat memberikan interaksi yang lebih tinggi.



Sumber : Fauzi (2008:4)

Gambar II.8
Struktur Navigasi Campuran

2.2.2. ERD (Entity Relationship Diagram)

Menurut Hidayatulloh dan Kawistara (2015:148) “ERD (*Entity Relationship Diagram*) adalah salah satu pemodelan yang sering digunakan untuk merancang basis data rasional”. Dua elemen fundamental pada ER adalah entitas dan *relationship* (keterhubungan).

1. Entitas

Entitas adalah objek (baik nyata ataupun abstrak) di dunia nyata yang dapat dibedakan dari objek lain berdasarkan karakteristik yang dimilikinya.

Informasi yang ingin disampaikan dari suatu entitas disebut dengan property / atribut. Misalnya pada entitas mahasiswa, beberapa atribut yang mungkin adalah NIM, nama mahasiswa, tanggal lahir, alamat, nomor telepon, dsb. Pada ER Diagram atribut dilambangkan dengan elips.

Untuk membedakan antara data pada suatu entitas, biasanya adalah suatu atribut digunakan sebagai *identifier* yang dapat membedakan antara satu data dengan lainnya. Atribut ini disebut *primary key*. Misalnya pada entitas mahasiswa, atribut NIM digunakan sebagai *primary key*, karena NIM merupakan identitas unik dari seorang mahasiswa (tidak ada dua mahasiswa yang memiliki NIM sama). Suatu atribut dapat mengacu / merefer ke atribut pada tabel lain (atau atribut lain pada tabel itu sendiri). Atribut yang diacu oleh atribut lain disebut *foreign key*. *Foreign key* adalah suatu cara untuk menjaga integritas data.

Pertimbangan pemilihan atribut mana yang dijadikan sebagai *primary key* tergantung dari beberapa hal, antara lain:

- a. Atribut harus unik. Itu artinya untuk objek/barang yang berbeda, atribut tersebut pasti beda juga.
- b. Nilai atribut tersebut jarang berubah.
- c. Mudah diingat dan biasanya memiliki format konsisten.

2. Relasi

Relasi adalah asosiasi yang menyatakan keterhubungan antar entitas.

Misanya:

- a. Relasi **Mengajar** yang menghubungkan entitas **Dosen** dengan **Mata Kuliah**.

Relasi ini menyatakan bahwa seorang Dosen hanya boleh mengajar banyak Mata Kuliah dan satu mata kuliah bias diajar oleh beberapa orang Dosen.

- b. Relasi **Tempat** antara entitas **Mata Kuliah** dengan **Ruangan**. Relasi ini menyatakan bahwa satu mata kuliah bias diajarkan di beberapa ruang yang berbeda – beda (N) dan satu ruangan bias digunakan untuk mengajar Mata Kuliah yang berbeda – beda (N).

Pada ER Diagram, relasi dilambangkan dengan wajik/*diamond*.

3. Kardinalitas

Relasi memiliki derajat keterhubungan, derajat keterhubungan antar entitas pada suatu relasi seperti pada contoh di atas disebut dengan Kardinalitas. Terdapat tiga jenis kardinalitas :

- a. Satu ke satu (1-1), misalnya relasi **Registrasi**.
- b. Satu ke banyak/Banyak ke satu (1-N), misalnya relasi **Tempat**.
- c. Banyak ke banyak (N-N), misalnya pada relasi **Nilai**.

Contoh menentukan kardinalitas:

1. Misalnya saja di suatu kampus yang boleh menjadi admin adalah dosen.
 - a. Yang jadi pertanyaan, berapa maksimal Dosen yang boleh menggunakan satu login admin tertentu? Jawabannya 1.

Bolehkah sebuah login admin tidak dimiliki oleh Dosen (“tidak bertuan”)?

Jawabannya tidak boleh karena munculnya sebuah *login admin* adalah karena dosen yang registrasi sebagai admin (minimal 1). Itu artinya $K1 = 1..1$.

- b. Berapa maksimal login admin yang boleh dimiliki oleh seorang Dosen? Jawabannya 1.

Bolehkah dosen tidak menjadi admin? Jawabannya boleh (minimal 0). Itu artinya $K2 = 0..1$.

Kardinalitas sebuah relasi diambil dari nilai maksimalnya sehingga relasi Registrasi mempunyai kardinalitas 1-1.



Sumber : Hidayatullah dan Kawistara (2015:152)

Gambar II.9
Relasi Registrasi

2. Pada relasi Tempat

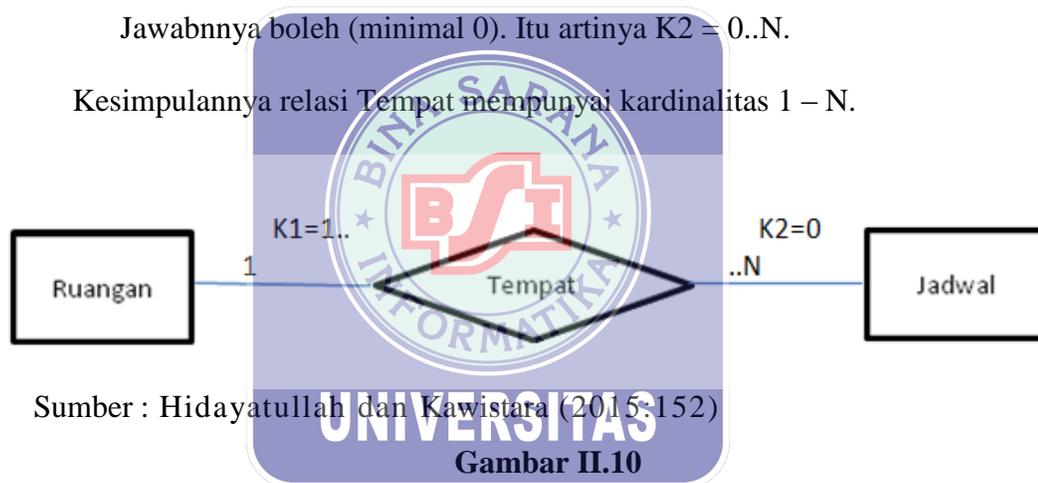
- a. Berapa maksimal ruangan yang digunakan untuk sebuah jadwal kuliah tertentu? Jawabannya 1.

Bolehkah sebuah jadwal kuliah tidak dialokasikan ruangnya? Jawabannya tidak boleh (minimal 1). Itu artinya $K1 = 1..1$.

- b. Untuk satu ruangan, bias digunakan untuk berapa jadwal? Jawabannya banyak.

Bolehkah sebuah ruangan tidak dialokasikan untuk jadwal kuliah apapun? Jawabannya boleh (minimal 0). Itu artinya $K2 = 0..N$.

Kesimpulannya relasi Tempat mempunyai kardinalitas 1 – N.



Sumber : Hidayatullah dan Kawistara (2015:152)

Gambar II.10

Relasi Tempat

3. Pada Relasi Nilai

- a. Berapa maksimal nilai mahasiswa yang boleh ada dalam satu mata kuliah tertentu? Jawabannya banyak (N). Berapa minimalnya? Realitasnya sebetulnya suatu mata kuliah akan dibuka jika yang mengambil mata kuliah tersebut memenuhi *quota*. Namun dalam konteks skema basis data, jumlah minimalnya 1. Itu artinya $K1 = 1..N$

b. Berapa jumlah maksimal nilai mata kuliah yang dimiliki seorang mahasiswa? Jawabannya banyak. Jadi misalnya Basis Data nilainya B, Pemrograman Web nilainya A, dan seterusnya.

Berapa minimalnya? Asumsinya mahasiswa diperbolehkan mengambil satu mata kuliah saja sehingga $K2 = 1..N$.

Kesimpulannya relasi Nilai memiliki kardinalitas $N - N$.



Sumber : Hidayatullah dan Kawistara (2015:153)

Gambar II.11
Relasi Nilai

Kardinalitas penting karena nantinya akan berpengaruh pada implementasi ER Diagram ke dalam basis data.

4. Langkah – langkah membuat ER Diagram

Langkah – langkah membuat ER Diagram dengan studi kasus website akademik.

a. Identifikasi dan tetapkan seluruh himpunan entitas yang akan terlibat. Misalnya studi kasus yang kita ambil adalah website akademik. Objek – objek yang terlibat dengan proses akademik antara lain : Dosen, Mahasiswa, Mata Kuliah, Ruang, Jadwal, dan Admin.

b. Tentukan atribut – atribut *key* dari masing – masing himpunan entitas. *Atribut Key* adalah atribut yang menjadi *identifier* (pembeda/identitas) dalam suatu entitas. Dosen *key*-nya adalah NIP. Mahasiswa *key*-nya adalah NIM, dll.

c. Identifikasi dan tetapkan seluruh relasi di antara entitas yang ada beserta *foreign key*-nya.

- 1) Antara entitas Dosen dan Mata Kuliah dengan relasinya adalah **Mengajar**.
- 2) Antara Dosen dengan Admin relasinya adalah **Registrasi**.
- 3) Antara Mahasiswa dengan Mata Kuliah relasinya adalah **Nilai**.
- 4) Antara Mata kuliah dengan Jadwal relasinya adalah **Alokasi**.
- 5) Antara Jadwal dengan Ruangan relasinya adalah **Tempat**.

d. Menentukan derajat kardinalitas relasi untuk setiap himpunan relasi.

- 1) Relasi **Mengajar** kardinalitasnya $N - N$.
- 2) Relasi **Registrasi** kardinalitasnya $1 - 1$.
- 3) Relasi **Nilai** kardinalitasnya $N - N$.
- 4) Relasi **Alokasi** kardinalitasnya $1 - N$.
- 5) Relasi **Tempat** kardinalitasnya $1 - N$.

e. Melengkapi entitas dan relasi dengan atribut – atribut deskripsi (*non key*).

- 1) Menambahkan atribut Nama_Dosen pada entitas Dosen.
- 2) Menambahkan atribut Nama_Mhs dan password pada entitas mahasiswa.
- 3) Menambahkan atribut password_Admin pada entitas Admin.
- 4) Menambahkan atribut Nama_Matkul, SKS_Matkul, Semester, pada entitas Mata Kuliah.
- 5) Menambahkan atribut Hari dan Jam pada entitas Jadwal.
- 6) Menambahkan atribut Nama_ruangan pada entitas Ruangan.

2.2.3. LRS (*Logical Record Structured*)

Menurut Wulandari (2013:17), “*Logical Record Structured* dibentuk dengan nomor tipe *record*”. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda LRS dengan diagram *entity relationship* diagram nama *tipe record* berada diluar kotak *field tipe record* ditempatkan. LRS terdiri dari *link-link* tentara *tipe record*. *Link* ini menunjukkan arah dari satu *tipe record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link tipe record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode lain dimulai dengan *Entity Relationship Diagram* dan langsung dikonversikan ke LRS.

1. Konversi ERD ke LRS, *Diagram Entity Relationship* harus diubah kebentuk LRS (struktur *record* secara logika). Dari bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi (tabel).
2. Konversi ERD ke LRS sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:
 - a. Setiap entitas diubah kebentuk kotak dengan nama entitas, berada diluar kotak dan atribut berada di dalam kotak.
 - b. Sebuah *relationship* kadang disatukan, dalam sebuah kotak bersama entitas, kadang sebuah kotak bersama-sama dengan entitas, kadang disatukan dalam sebuah kotak tersendiri.

3. Konversi LRS ke relasi (tabel) relasi atau tabel adalah bentuk pernyataan data secara grafis 2 (dua) dimensi, yang terdiri dari kolom dari baris. Relasi adalah bentuk visual dari sebuah *file*, dan tiap *tuple* dalam sebuah *field* atau yang dalam bentuk lingkaran *Diagram entity relationship* dikenal dengan sebutan atribut. Konversi dari *logical record structure*.

Dilakukan dengan cara:

- a. Nama *logical record structure* menjadi nama relasi.
- b. Tiap *atribut* menjadi sebuah kolom di dalam relasi.

2.2.4. Pengujian Web

Menurut Rizky (2011:237) “*Testing* adalah sebuah proses yang diejawantahkan sebagai siklus hidup dan merupakan bagian dari proses rekayasa perangkat lunak secara terintegrasi demi memastikan kualitas dari perangkat lunak serta memenuhi kebutuhan teknis yang telah disepakati dari awal”.

1. *Blackbox testing*

Menurut Rizky (2011:264), “*Blackbox testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja Internalnya”. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenal proses testing dibagian luar.

Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi *inventory* di sebuah perusahaan. Maka pada jenis *white box testing*, perangkat lunak tersebut akan berusaha listing programnya untuk kemudian dites menggunakan teknik-teknik

yang telah dijelaskan sebelumnya. Sedangkan pada jenis *black box testing*, perangkat lunak tersebut akan di eksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar listing programnya.

Menurut Rizky (2011:264), Beberapa keuntungan yang diperoleh dari jenis *testing* ini antara lain:

- a. Anggota tim tester tidak harus dari seseorang yang memiliki kemampuan teknis dibidang pemograman.
- b. Kesalahan dari perangkat lunak ataupun *bug* seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
- c. Hasil dari *black box testing* dapat memperjelas kontradiksi ataupun keracunan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
- d. Proses *testing* dapat dilakukan lebih cepat dibandingkan *white box testing*.

Menurut Rizky (2011:265), Beberapa teknik testing yang tergolong dalam tipe ini antara lain:

1) *Equivalence partitioning*

Pada teknik ini, tiap inputan data dikelompokkan ke dalam grup tertentu, yang kemudian dibandingkan *output* nya.

2) *Boundary Value Analysis*

Merupakan teknik yang sangat umum digunakan pada saat awal sebuah perangkat lunak selesai dikerjakan. Pada teknik ini, dilakukan inputan yang melebihi dari batasan sebuah data. Sebagai contoh, untuk sebuah inputan harga barang, maka dapat dilakukan testing dengan menggunakan

angka negatif (yang tidak diperbolehkan dalam sebuah harga). Jika perangkat lunak berhasil mengatasi inputan yang salah tersebut, maka dapat dikatakan teknik ini telah selesai dilakukan.

3) *Cause Effect Graph*

Dalam teknik ini, dilakukan proses testing yang menghubungkan sebab dari sebuah inputan dan akibatnya pada output yang dihasilkan. Sebagai contoh, pada sebuah inputan nilai siswa, jika diinputan angka 100, maka output nilai huruf seharusnya adalah A. Tetapi bias dilakukan testing, apakah output nilai huruf yang dikeluarkan jika ternyata inputan nilai adalah 67,5.

4) *Random Data Selection*

Seperti namanya, teknik ini berusaha melakukan proses inputan data dengan menggunakan nilai acak. Dari hasil inputan tersebut kemudian dibuat sebuah tabel yang menyatakan validitas dari *output* yang dihasilkan.

5) *Feature test*

Pada teknik ini, dilakukan proses testing terhadap spesifikasi dari perangkat lunak yang telah selesai dikerjakan.

