

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Pada pembuatan tugas akhir ini penulis menggunakan beberapa landasan teori pendukung yang dapat digunakan sebagai dasar pembuatan rancangan *website*. Landasan ini diharapkan dapat membantu dalam memahami suatu masalah dan memperkecil kemungkinan terjadinya kesalahan yang dilakukan oleh pengguna.

A. 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 atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangun yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut *Hyperlink*, sedangkan *teks* yang dijadikan media penghubung disebut *Hypertext*.

Seiringan dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokan jenis web lebih diarahkan kepada fungsi dan sifat web tersebut.

a. Berdasarkan Sifatnya

1) Website Dinamis

Website Dinamis merupakan sebuah website yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan database MySQL atau MS SQL. Misalnya website www.detik.com.

2) Website Statis

Website Statis merupakan sebuah website yang kontennya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan database. Misalnya web profil organisasi.

b. Berdasarkan Fungsinya

1) *Personal Website*

Website yang berisi informasi pribadi seseorang.

2) *Commercial Website*

Website yang dimiliki oleh sebuah perusahaan dan sifatnya untuk bisnis.

3) *Government Website*

Website yang dimiliki oleh instansi pemerintahan, pendidikan yang bertujuan memberikan pelayanan kepada pengguna.

4) *Non-Profit Organization Website*

Website yang dimiliki oleh organisasi bersifat *non-profit* atau tidak bersifat bisnis.

B. Internet

Menurut Irwansyah dan Moniaga (2014:12), “Internet adalah sebuah sistem global jaringan komputer yang saling menghubungkan antara satu dengan yang lain di seluruh penjuru dunia”. Adapun standar yang paling banyak digunakan disebut *Internet Protocol Suite* (TCP/IP).

TCP/IP adalah protokol yang paling banyak digunakan di internet. Protokol TCP/IP merupakan cara standar untuk memaketkan dan mengalamatkan data komputer (sinyal elektronik) sehingga data tersebut dapat dikirim ke komputer terdekat atau keliling dunia dan tiba dalam waktu yang cepat tanpa hilang atau rusak.

Perkembangan *internet* dimulai pada tahun 1969, lembaga riset Departemen Pertahanan Amerika atau *Defence Advance Project Agency* (DARPA) mendanai sebuah proyek untuk mengembangkan jaringan komunikasi data antar beberapa komputer. Pengembangan jaringan tersebut sukses dan melahirkan ARPANet.

Aplikasi internet yang pertama kali ditemukan adalah *File Transport Protocol* (FTP) yang merupakan sebuah cara untuk mengirimkan atau mentransferkan file. Pada tahun 1984 jumlah host di *internet* lebih dari 1000 buah. Pada tahun itu pula diperkenalkan *Domain Name System* yang digunakan untuk mengganti atau mendefinisikan suatu *Internet Protocol* (IP) agar lebih mudah diingat.

C. *Web Server*

Menurut Supardi (2010:2) *Web server* merupakan “perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan kembali ke *browser*”. Contoh *Web Server*, adalah IIS (*Internet Information Service*) produk *Microsoft Corp.*

Fungsi utama *Web Server* adalah untuk melakukan atau mentransferkan berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan

sedemikian rupa. Halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi.

D. Web Browser

Menurut Limantara (2009:1) *web browser*, adalah “aplikasi perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan *teks*, *image*, *games* dan informasi lainnya yang berlokasi pada halaman *web* pada *World Wide Web* (WWW) atau *Local Area Network* (LAN)”.

Teks dan *image* pada halaman *web* dapat berisi *hyperlink* ke halaman *web* lain pada *website* yang sama maupun berbeda. Dengan *web browser*, seorang pengguna bisa mengakses informasi yang disediakan pada banyak *website* secara cepat dan mudah. Meskipun *browser* pada umumnya digunakan untuk mengakses *World Wide Web*, *browser* juga bisa digunakan untuk mengakses informasi yang disediakan *web server* pada jaringan privat atau ini pada sistem *file*.

E. HTTP (Hypertext Transfer Protocol)

Menurut Simarmata (2010:52) HTTP adalah “ protokol komunikasi *stateless* yang berbasiskan pada TCP yang awalnya digunakan untuk mengambil kembali *file-file* HTML dari *server Web* ketika dirancang pada tahun 1991”.

Versi 1.1 (ditetapkan di RFC2616) telah mengalami peningkatan. Versi ini mengizinkan *client* dan *server* menggunakan banyak *headers* untuk menyampaikan status informasi dan deskripsi agen pengguna (*user-agent*) dan bertindak sebagai lat yang bersifat elementer untuk autentikasi pengguna. Versi awal yang diciptakan pada tahun 1991, diacu sebagai HTTP *version* 0.9, adalah protokol yang sederhana untuk komunikasi antara *client* dan *server*. Kebanyakan versi yang sekarang ini ada berasal dari protokol HTTP versi 1.1.

F. PHP (*Hypertext Preprocessor*)

Menurut Octavian (2010:31) PHP adalah akronim dari *Hypertext Preprocessor*, “ yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML”.

Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-parsing di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan dilingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *user* memilih perintah “*View Source*” pada *web browser* yang mereka gunakan. Selain menggunakan PHP, aplikasi *web* juga dapat dibangun dengan Java (*JSP- JavaServer Pages* dan *Servlet*), *Perl*, maupun ASP (*Active Server Pages*).

Meskipun PHP dapat digunakan untuk aplikasi CLI (*Command Line Interface*) dan juga aplikasi desktop, namun pada umumnya orang menggunakan PHP untuk tujuan pembuatan aplikasi *web*.

PHP adalah bahasa yang dirancang untuk mudah diletakkan didalam kode HTML. Banyak dijumpai kode PHP yang menyatu dengan kode HTML. Kode PHP diawali dengan tag **<?php** dan diakhiri dengan tag **?>**. Apabila kita melakukan konfigurasi terhadap *file* php. Ini untuk mengizinkan penggunaan tag pendek (*short tag*) dengan mengubah nilai *short_open_tag* menjadi **On**, maka *tag* tersebut dapat diganti dengan **<? Dan ?>**. Dalam PHP 5, nilai *default* dari *short_open_tag* adalah *Off*. Selain itu, PHP juga dapat menggunakan *tag* gaya

ASP, <% dan %>, dengan mengubah nilai *asp_tag* dalam *file php*, ini menjadi **On**. kita akan tetap menggunakan *tag* <?php dan ?> untuk menulis kode-kode php.

Berikut ini contoh kode PHP yang sangat sederhana:

```
<? Php
    echo "Hello World ! ";
?>
```

Perintah *echo* didalam PHP berguna untuk mencetak nilai, baik *teks*, maupun numerik, ke layar *web browser*. Selain *echo*, kita juga dapat menggunakan perintah *print*, sehingga kode diatas dapat juga ditulis seperti berikut:

```
<? Php
    print "Hello World !";
?>
```

G. HTML (*Hyper Text Markup Language*)

Menurut Raharjo dkk (2012:286) HTML adalah “singkatan dari *HyperText Markup Language*, HTML merupakan *file* teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser*”.

Setiap informasi yang tampil di *web* selalu dibuat menggunakan kode HTML. Dokumen HTML sering disebut juga sebagai *web page* (halaman *web*). Untuk membuat dokumen HTML, dapat dibuat menggunakan aplikasi *Text Editor*, bisa *Notepad* (untuk lingkungan *MS Windows*), *Emacs* atau *Vi Editor* (untuk lingkungan *Linux*), dan sebagainya.

Agar lebih mempermudah dalam pembuatan dokumen HTML, sekarang telah banyak tersedia aplikasi HTML, *Editor*, yaitu suatu aplikasi *Text Editor* yang dikhususkan untuk pembuatan kode-kode HTML.

Kode HTML dari suatu halaman *web* dapat kita lihat dari aplikasi *web browser* yang kita gunakan, yaitu dengan cara memilih menu **View-Source** (untuk Internet Explorer) dan **View-Page Source** (untuk Mozilla Firefox).

Secara umum, dokumen HTML terbagi atas dua bagian, yaitu bagian **header** (kepala) dan **body** (badan). Bagian *header* diawali dengan tag `<head>` dan diakhiri dengan tag `</head>`, sedangkan bagian *body* diawali dengan tag `<body>` dan ditutup dengan tag `</body>`. Kedua bagian tersebut diapit oleh tag `<html>` dan `</html>`, yang digunakan untuk menandai bahwa dokumen yang kita buat adalah dokumen HTML. Untuk lebih jelasnya, berikut ini kerangka dari dokumen HTML:

```
<html> <!-- Tag untuk memulai dokumen HTML-- >
<!--Bagian header -- >
<head>
  <title> Judul halaman </title>
</head>
```

```
<!--Bagian Body -- >
```

```
<body>
```

Informasi yang akan ditampilkan pada *web browser* harus ditulis dalam bagian ini.

```
</body>
```

```
</html> <!--Tag untuk mengakhiri dokumen HTML-- >
```

Seperti yang tampak diatas, *tag-tag* dalam dokumen HTML dibuat secara berpasangan (meskipun tidak semua). Sebagai contoh, dokumen HTML diawali dengan *tag* `<html>` dan ditutup atau diakhiri dengan *tag* `</html>`. Begitu juga dengan *tag* `<head>`, serta *tag* `<body>` yang ditutup menggunakan *tag* `</body>`. Bagian *teks* yang diapit oleh tanda `<!-- dan -->` merupakan komentar yaitu bagian *teks* yang tidak ikut ditampilkan ke dalam *web browser*.

H. CSS (*Cascading Style Sheet*)

Menurut Sya'ban (2010:37) CSS (*Cascading Style Sheet*) adalah “suatu bahasa yang dikhususkan untuk mengatur gaya atau *layout* sebuah halaman *web*”. CSS digunakan oleh pembuat halaman *web* dan juga pengakses halaman *web*, untuk mendefinisikan warna, huruf *layout*, dan dan aspek-aspek presentasi dokumen lainnya.

CSS memang *didesain* untuk memisahkan antara isi dokumen (yang ditulis menggunakan HTML atau bahasa lain yang sejenis) dengan bentuk presentasi dokumen (ditulis dalam CSS). Pemisahan ini memberikan keuntungan akan adanya peningkatan dalam aksesibilitas isi, menyediakan bentuk presentasi, serta mereduksi kompleksitas dan perulangan-perulangan pada struktur isi.

I. JavaScript

JavaScript diperkenalkan pertama kali oleh *Netscape* pada tahun 1995. Bahasa yang sekarang dikenal dengan nama *javascript* ini sebelumnya bernama *LiveScript* dan berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator 2*. Kemudian melalui kerjasama antara *Netscape* dan *Sun*, *Netscape* memberikan nama *JavaScript* kepada bahasa tersebut pada tanggal 4 Desember 1995. Pada waktu hampir bersamaan, pihak *microsoft* sendiri mencoba

mengadaptasi teknologi ini yang mereka sebut *Jscript* di *browser* milik mereka sendiri, yaitu *Internet Explorer 3*. Kini penggunaan dari *Jscript* sendiri masi terus dikembangkan pihak Microsoft dengan didukung oleh kebanyakan browser yang tersedia saat ini.

Menurut Wahana Komputer (2010:1) memberikan pernyataan bahwa *JavaScript* merupakan “bahasa yang terbentuk skrip yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen *web*”. Dengan kata lain, bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML (*Hypertext Markup Language*) dengan mengijinkan pengekseskuan perintah-perintah pada sisi *client*, dan bukan sisi *server* dokumen *web*. Pada hakikatnya, bahasa pemrograman *javascript* berisi skrip yang pemasangannya terselip disebuah dokumen HTML, sehingga bahasa *javascript* ini tidak memerlukan kompilator atau penerjemah khusus untuk mengekseskunya.

J. JQuery

Menurut Wahana Komputer (2012:2) menjelaskan bahwa “*JQuery* merupakan pustaka *javascript* yang dibangun untuk mempercepat dan memperingkas serta menyederhanakan manipulasi dokumen HTML, penanganan *event*, animasi, dan interaksi *ajax* untuk mempercepat pengembangan web”.

Dengan *JQuery*, *develover* akan dimanjakan dengan suatu pemrograman *javascript* yang sangat sederhana jika dibandingkan dengan native *javascript*, sebelum memulai mempelajari *JQuery*, kita harus mempunyai pengetahuan dasar mengenai HTML, CSS, dan *javascript*.

K. Dreamweaver CS6

Menurut MADCOMS (2013:2) “*Adobe Dreamweaver CS6* adalah versi terbaru dari *Adobe Dreamweaver* yang merupakan bagian dari *Adobe Creative Suite*. *Adobe Dreamweaver* sendiri merupakan aplikasi yang digunakan sebagai HTML editor professional untuk mendesain web secara visual”. Aplikasi ini juga biasa dikenal dengan istilah *What You See Is What You Get* (WYSIWYG), yang intinya adalah kita tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah site dan dapat melihat hasil desainnya secara langsung.

Dengan kemampuan fasilitas yang optimal dalam jendela desain akan memberikan kemudahan untuk mendesain *web* meskipun untuk para *web* desainer pemula sekalipun. Kemampuan *Adobe Dreamweaver CS6* untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, *javascript*, dan yang lainnya juga memberikan fasilitas maksimal kepada desainer *web* dengan menyertakan bahasa pemrograman didalamnya.

L. Database (Basis Data)

Menurut Rosa dan M. Shalahudin (2015:43) menjelaskan bahwa “Sistem basis data 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”.

Secara prinsip, dalam suatu *database* tercakup dua komponen penting, yaitu Data dan informasi. Jadi tujuan akhir bagaimana mengelola data sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan terhadap data secara mudah dan cepat (Efektif, *Efisien* dan Akurat).

M. *My Structure Query Language (MySQL)*

Menurut Kadir (2010:10) menjelaskan bahwa “*MySQL* tergolong sebagai *DataBase Management System (DBMS)*. Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat”.

Di dalam dunia internet, *MySQL* dijadikan sebuah *database* yang paling banyak digunakan selain *database* yang bersifat *share ware* seperti *Ms Access*, penggunaan *MySQL* ini biasa dipadukan dengan menggunakan program aplikasi PHP, karena dengan menggunakan keduanya program tersebut di atas telah terbukti akan kehandalannya dalam menangani permintaan data.

Pada koperasi *database* ini, *MySQL* memiliki *query* yang telah distandarkan oleh ANSI atau SIO yaitu menggunakan bahasa SQL. Kemampuan lain *MySQL* adalah mampu mendukung *Relasional Database Manajemen Sistem (RDBMS)*, sehingga dengan kemampuan ini *MySQL* akan mampu menangani data-data sebuah perusahaan yang berukuran sangat besar hingga berukuran *Giga Byte*.

N. *Xampp*

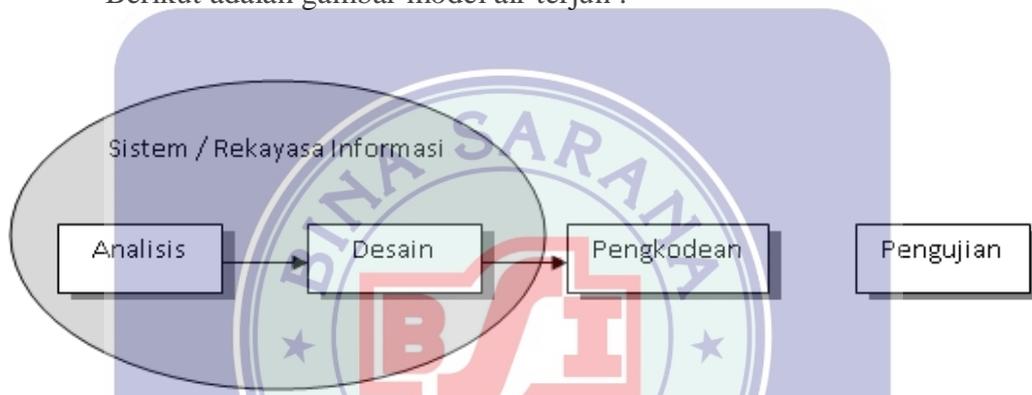
Menurut Wicaksono (2008:7) “*Xampp* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* di computer lokal”.

Xampp juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu kita melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus online atau terkases dengan internet.

O. Model Pengembangan Perangkat Lunak *Waterfall*

Rosa dan M. Shalahudin (2015:28) menjelaskan bahwa “model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasis (*classic life cycle*). Metode air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”.

Berikut adalah gambar model air terjun :



Sumber : Rosa dan M. Shalahudin (2015:28)

Gambar II.1.
Model Air Terjun

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami 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 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 didokumentasikan.

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 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 tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

2.2. Teori Pendukung (*Tools System*)

Perancangan program tidak terlepas dari penggunaan peralatan pendukung (*Tool System*), penggunaan peralatan pendukung ini menambahkan kemudahan didalam merancang halaman-halaman pada sebuah *website*. Pada perancangan *website* ini penulis menggunakan beberapa peralatan pendukung berikut:

A. Struktur Navigasi

Menurut Binanto (2010:268) mengemukakan bahwa “Setiap rencana akan dibuat desainnya dan kemudian diproduksi menjadi produk yang jadi yang bersifat sementara.”

Struktur navigasi juga dapat diartikan sebagai struktur alur dari suatu program yang merupakan rancangan hubungan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen pembuatan *website*. Ada empat macam bentuk dasar yaitu :

a. *Linear*

Menurut Binanto (2010:269) mengemukakan bahwa “Pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi yang satu



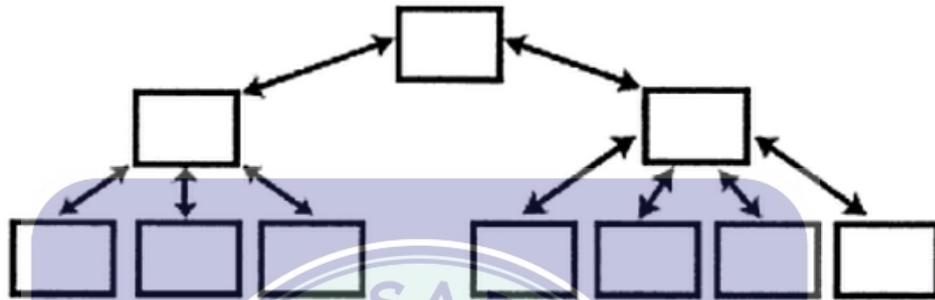
Sumber: Binanto (2010:269)

Gambar II.2.
Struktur Navigasi *Linear*

b. *Hierarchy* (Hirarki)

Struktur *Hierarchy* (bercabang) untuk menampilkan data berdasarkan kriteria tertentu. Informasi utama disebut *parent* dan informasi pada cabangnya

disebut *child*. Menurut Binanto (2010:269) memberikan pengertian bahwa “struktur dasar ini disebut juga struktur *linear* dengan percabangan karena pengguna navigasi di sepanjang cabang pohon struktur yang terbentuk oleh logika isi”.

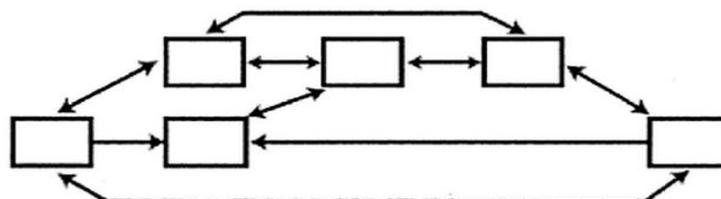


Sumber: Binanto (2010:269)

Gambar II.3.
Struktur Navigasi *Hierarkis*

c. *Non Linear*

Menurut Binanto (2010:269) Pada struktur ini “Pengguna diperkenankan membuat navigasi dengan bebas melalui isi proyek dengan jalur yang sudah ditentukan sebelumnya”. Percabangan yang dibuat pada struktur *linear* ini berbeda dengan percabangan pada struktur *hierarchy*, karena pada percabangan *non linear* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada halaman utama dan halaman pendukung.

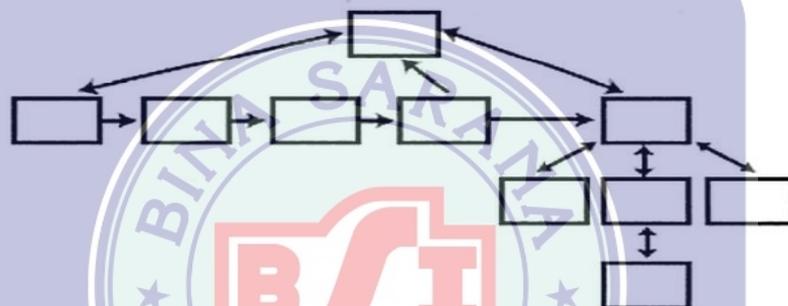


Sumber: Binanto (2010:270)

Gambar II.4.
Struktur Navigasi *Non Linear*

d. *Composite*

Struktur Navigasi *Composite* (campuran) merupakan struktur gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Hierarchy*, dan *Non Linear*. Struktur ini disebut juga struktur navigasi bebas. Menurut Binanto (2010:270) Pada struktur ini “Pengguna melakukan navigasi dengan bebas (secara *non-linear*), tetapi terkadang dibatasi informasi penting atau pada data yang paling terorganisasi secara logis pada satu *hierarchy*”.



Sumber: Binanto (2010:270)

Gambar II.5.
Struktur Navigasi *Composite*

B. *Entity Relationship Diagram*

Menurut Simarmata (2010:69) “*Entity Relationship Diagram* (ERD) adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek kedalam entitas-entitas dan menentukan hubungan antar entitas.

Entity dapat berarti sebuah *obyek* yang dapat dibedakan dengan *obyek* lainnya. *Obyek* tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari *obyek* yang lain. Dalam dunia *database* *entity* memiliki atribut yang menjelaskan karakteristik dan *entity* tersebut. Ada dua macam atribut yang dikenal dalam *entity* yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap *entity*

memiliki himpunan yang diperlukan sebuah *primary key* untuk membedakan anggota-anggota dalam himpunan tersebut.

1. Komponen ERD (*Entity Relationship Diagram*)

a. Entitas

Entitas adalah sesuatu yang nyata atau abstrak dimana kita akan menyimpan data. Ada 4 kelas entitas, yaitu misalnya pegawai, pembayaran, kampus, dan buku.

b. Relasi

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah.

c. Atribut

Atribut adalah ciri umum atau semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah property, elemen data, dan field. Misalnya nama, alamat, nomor pegawai dan gaji adalah atribut entitas pegawai.

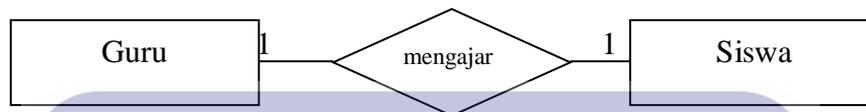
2. Pemetaan kardinalitas

Pemetaan kardinalitas menyatakan jumlah entitas dimana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi. Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himunan relasi biner R antara himpunan entitas A dan B , pemetaan kardinalitas harus salah satu dari berikut :

a. *One to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A .



Sumber: Simarmata (2010:64)

Gambar II.6.
Kardinalitas *One to One*

b. *One to Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B . sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A .



Sumber: Simarmata (2010:65)

Gambar II.7.
Kardinalitas *One to many*

c. *Many to One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B . sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A .



Sumber: Simarmata (2010:65)

Gambar II.8.
Kardinalitas *many to One*

d. *Many to Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada

A.



Sumber: Simarmata (2010:66)

Gambar II.9.
Kardinalitas *many to many*

C. LRS (*Logical Record Structure*)

Menurut Frieyadie (2007:13) “Sebelum tabel dibentuk dari field atau atribut entitas secara fisik atau level internal, maka harus dibuatkan suatu bentuk relational model yang dibuat secara logic atau level external dan konsep, dari pernyataan tersebut dibutuhkan yang disebut dengan Logical Relationship Structure (LRS).”

LRS merupakan hasil dari pemodelan Entity Relationship (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas.

D. Pengujian *Web Black Box Testing*

Menurut Rosa dan M. Salahuddin (2015:275) *Blackbox Testing* yaitu “Menguji 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”.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan perangkat lunak apakah 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 yang dibuat adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

