

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

2.1.1. Website

Menurut Sibero (2013:2) *World Wide Web (www)* adalah “suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet”.

World Wide Web Consortium (W3C) organisasi dunia yang mengatur standar *World Wide Web (Www)*. Organisasi ini berkomitmen dalam mengatur aturan pengembangan web, yang terdiri dari 320 anggota. Selain mengatur aturan pengembangan web, juga ikut melibatkan diri dalam dunia pendidikan. Beberapa standar hasil pengembangan dari *W3C* antara lain *HTML, CSS, XHTML, DOM, SOAP, XML*, dll.

Beberapa aplikasi berbasis web, yaitu:

1. *Internet*

Menurut Sibero (2011:10) “*Internet (Interconneted Network)* adalah jaringan komputer yang menghubungkan antar jaringan secara global, *internet* dapat juga disebut jaringan alam suatu jaringan yang luas”. Seperti halnya jaringan komputer lokal maupun jaringan komputer area *internet* juga menggunakan protokol komunikasi yang sama yaitu *TCP/IP (Transmission Control Protokol / Internet Protocol)*



2. *Web server*

Menurut Sibero (2013:11) “*Web server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak”. Secara bentuk fisik dan cara kerjanya, perangkat keras *web server* tidak berbeda dengan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya. Perbedaan tersebut dikarenakan *web server* bekerja sebagai penyedia layanan yang dapat diakses oleh banyak pengguna, sehingga dibutuhkan kapasitas yang besar dibandingkan PC. Dukungan perangkat lunak sangat dibutuhkan memiliki agar *web server* dapat berjalan optimal. Setiap perangkat lunak *web server* memiliki karakteristik dan teknologi yang digunakan untuk mengatur kerja sistemnya.

a. *Xampp*

Menurut Wicaksono (2012:7) “*Xampp* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolahan data *MySQL* di *computer local*”.

Beberapa paket yang telah disediakan diantaranya adalah:

- 1) *Apache*
- 2) *MySQL*
- 3) *PHP*
- 4) *FilZilla FTP Server*
- 5) *phpMyAdmin*, dan lain-lain.

Dengan adanya dukungan beberapa paket diatas maka semua kebutuhan untuk belajar *PHP* dapat terpenuhi.



b. MySQL

Menurut Arief (2011:152) “MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengolahan datanya”

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang *software* dan konsultan *database*, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

Contoh kode atau skrip dasar MySQL sebagai berikut :

1) Membuat *database*

```
create database name_database;
```

2) Masuk *database*

```
use name_database;
```

3) Membuat *tabel*

```
create database name_database(name_field1 varchar(lenght) not null,  
name_field2 varchar(lenght) not null;
```

4) Menambah *Record*

```
Insert into name_table (name_field1, name_field2, name_field3) values  
("isi", "isi", "isi");
```

5) Merubah *record*

```
update from name_table where name_field=nilai_field;
```

6) Menghapus *record*

```
delete from name_table where name_field=nilai field;
```



7) Menampilkan *record*

```
select * from name_table;
```

8) Menampilkan hanya sebagian *record*

```
Select name_table1=name_field1, name_table2=name_field2, from
name_table;
```

9) Tampilkan *record* dengan suatu syarat

```
select name_table name_field from name_table where
name_table_field_syarat=nilai field_syarat
```

10) Menampilkan dua tabel yang berhubungan

```
select name_table1. name_field1, name_table2. name_field2
name_table1.name_table2 where name_table1. name_field=
name_table2.name_field;
```

3. Web Browser

Menurut Sibero (2013:12) "*Web browser* adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi *web*".

Sumber informasi *web* diidentifikasi dengan *Uniform Resource Identifier* (URL) yang dapat terdiri dari halaman *web*, video, gambar, ataupun konten lainnya.

Sejarah *web browser* dimulai pada tahun 1991 saat Tim Berners-Lee membuat aplikasi *web browser* pertama pada komputer NeXT dengan nama *World Wide Web Browser*, kemudian di tahun 1993 NCSA (*National Center Supercomputing Application*) mengembangkan *web browser* grafis bernama *NCSA Mosaic*, yang kemudian dilanjutkan pada tahun 1994 merilis *Netscape Navigator* dan pada tahun 1998 berubah menjadi *Mozilla Firefox*. *Microsoft corp* ikut meramaikan perkembangan *web browser* dengan merilis *Internet Explorer* pada tahun 1995,

dan diikuti oleh *Opera* di tahun 1996. Tidak ketinggalan *Apple Inc* juga ikut meramaikan pasar *web browser* dengan merilis *Safari* pada tahun 2003. Dan di tahun 2008 *Google Inc* merilis *web browser* dengan nama *Google Chrome*.

2.1.2. Bahasa Pemrograman

1. *PHP (Hypertext Preprocessor)*

Menurut Andi (2014:33) “*PHP* merupakan bahasa berbentuk *script* yang ditempatkan di dalam *server* baru kemudian diproses, kemudian hasil pemrosesan dikirimkan kepada *web browser* klien”.

Untuk penulisan sintaks *PHP*, harus diawali dengan tag `<?` Dan diakhiri dengan tag `?>`. Sedangkan sintaks untuk menampilkan dalam *web browser* anda dapat menggunakan perintah *print* atau *echo*. Selain itu, *php* juga bisa anda tuliskan di dalam skrip *HTML*.

2. *CSS (Cascading Style Sheet)*

Menurut Andi (2014:5) “*CSS* merupakan bahasa pemrograman yang khusus menangani tampilan tiap elemen di dalam dokumen *HTML*. Dengan memanfaatkan *CSS*, struktur kode *HTML* kita akan terlihat rapi dan terstruktur”

CSS diperkenalkan untuk pengembangan *website* pada tahun 1996. Nama *CSS* didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan *parent child* pada setiap *style*.

3. Javascript

Menurut Andi (2014:15) “Javacript adalah bahasa pemograman *script* pada *browser*, atau biasa disebut dengan istilah *client side programming*”.

Javascript sendiri merupakan bahasa *scripting* yang bekerja disisi *client/browser* sehingga *website* bisa lebih interaktif.

4. HTML

Menurut Andi (2014:2) “*HTML* adalah singkatan dari *Hyper Text Markup Language*. *HTML* merupakan bahasa (kode) yang digunakan untuk membuat halaman web. *HTML* bukanlah bahasa prosedur seperti *C++*, atau *Pascal*. *HTML* lebih kepada bahasa yang menetapkan corak paparan dokumen pada *browser*”.

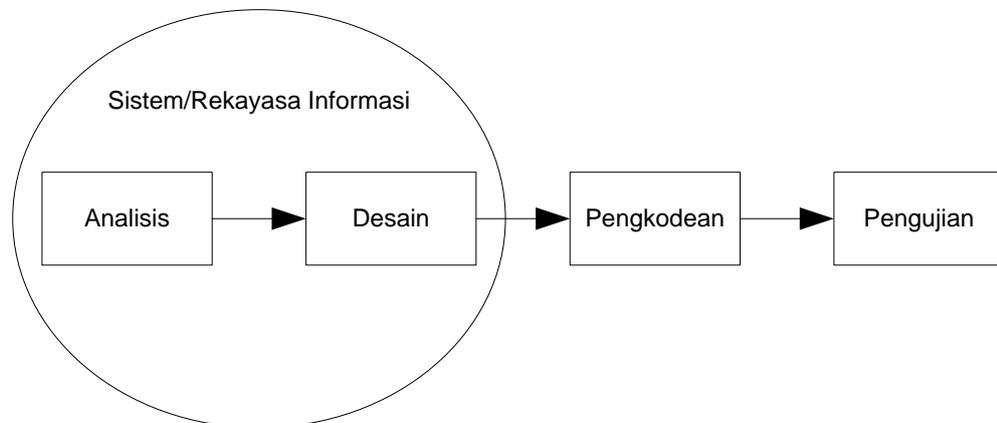
2.1.3 Basis Data

Menurut Sukamto dan Shalahuddin (2015:43) menjelaskan bahwa “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Basis data juga dapat diartikan sebagai media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

2.1.4 Model Pengembangan Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2015:28) menjelaskan bahwa “Model SDLC air terjun (*water fall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat secara sekuensial atau terurut

dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar air terjun:



Sumber: Sukanto & M.Shalahuddin (2015:28)

Gambar II.1 Model Pengembangan Perangkat Lunak *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

Tahapan ini sangat menekankan pada masalah pengumpulan kebutuhan pengguna pada tingkatan sistem dengan menentukan konsep. Seperti, kebutuhan *website* penjualan, membuat pendaftaran untuk anggota agar memudahkan anggota dalam melakukan transaksi tanpa proses yang rumit.

2. Desain

Proses perancangan sistem ini dimulai dengan pembuatan basis data seperti *entity relational diagram (ERD)* dan *logical record structured (LRS)*, pembuatan rancangan antar muka, dan pembuatan struktur navigasi yang menggunakan *composite* (campuran).

3. Pembuatan Kode Program

Pada tahapan ini adalah pembuatan kode program yang digunakan dalam mentranslasikan tahapan sebelumnya. Kode program yang digunakan adalah

seperti *HTML*, *Java Script*, *CSS* pada sisi *clientside* dan untuk *serverside* menggunakan *PHP* serta untuk pembuatan databasenya menggunakan *MySQL*.

4. Pengujian

Pada tahap ini, merupakan tahap akhir dalam pengembangan perangkat lunak, yang dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Tahap pengujian ini menggunakan *black box testing*.

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.



2.2. Teori Pendukung

2.2.1. Struktur Navigasi

Menurut Suyanto (2008:62), “Struktur navigasi dalam situs *web* melibatkan sistem navigasi situs *web* secara keseluruhan dan desain *interface* situs *web* tersebut, navigasi memudahkan jalan yang mudah ketika menjelajahi situs *web*”. 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* (Satu alur)

Linear (satu alur) 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 dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya, pengguna akan melakukan navigasi secara berurutan, dalam *frame* atau *byte* informasi satu ke yang lainnya.

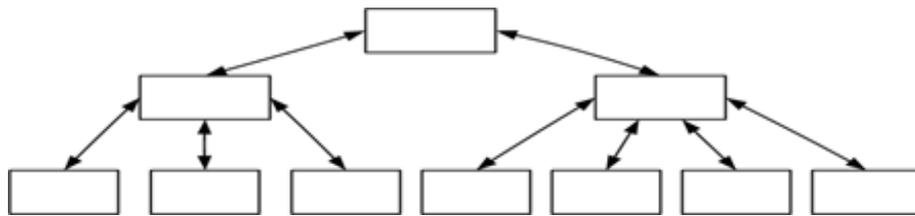


Sumber: Binanto (2010:269)

Gambar II.2 Struktur Navigasi *Linear*

b. *Hierarchical* (Hirarki)

Struktur dasar ini disebut juga struktur *linier* dengan percabangan karena pengguna melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi.

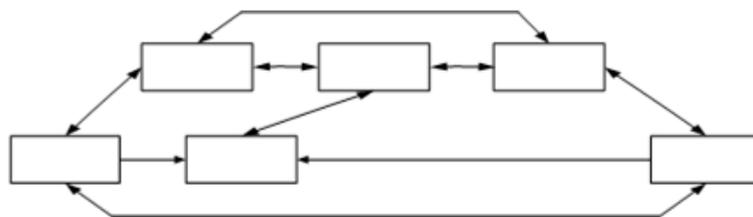


Sumber: Binanto (2010:269)

Gambar II.3 Struktur Navigasi Hirarki

c. *Non Linear* (Tidak Berurut)

Struktur navigasi *non-linier* merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur *non-linier* ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan *non-linear* ini walaupun terdapat percabangan tetap tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*, pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat dengan jalur yang sudah ditentukan sebelumnya.

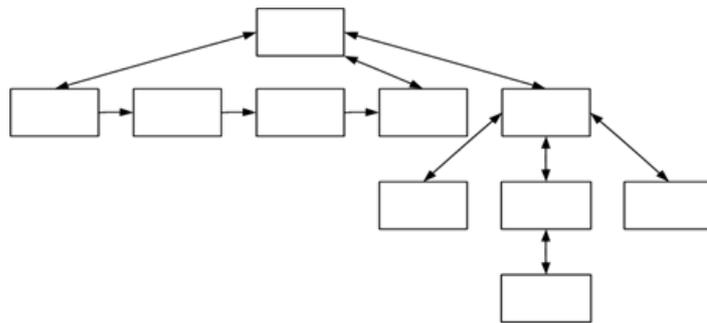


Sumber: Binanto (2010:269)

Gambar II.4 Struktur Navigasi Non Linear

d. *Composite* (Campuran)

Struktur navigasi campuran pengguna akan melakukan navigasi dengan bebas (secara *non-linier*), tetapi terkadang dibatasi presentasi *linier* film atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu hirarki.



Sumber: Binanto (2010:269)

Gambar II.5 Struktur Navigasi Campuran

2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Sutanta (2011:91) “Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.” Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. Entity Relationship Diagram (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan Entity Relationship Diagram (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan dikembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kereliasian antar data didalamnya.

1. Komponen *Entity Relationship Diagram*

Menurut Yasin (2012:276) menjelaskan “ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu:

a. *Entity*

Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.

b. *Atribut*

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar *atribut* diwakili oleh simbol *elips*.



c. Hubungan / Relasi

Relasi didefinisikan sebagai penghubung yang terjadi antar *entity*. Representasi diagram relasi adalah sebuah garis lurus yang menghubungkan dua buah *entity*. Jenis-jenis atau hubungan yang biasa terjadi ntar satu *entity* dengan *entity* lain dalam sebuah basis data, meliputi:

1) *One to One* / Satu ke Satu (1 : 1)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B. Contoh hubungan antara *entity* Mahasiswa dengan Kartu Mahasiswa. Seorang mahasiswa hanya boleh memiliki satu kartu mahasiswa, satu kartu mahasiswa hanya dimiliki oleh satu orang mahasiswa.

2) *One to many*/ Satu ke Banyak (1 : *)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A. Contohnya hubungan terjadi antara entiti konsumen dapat memiliki satu

3) *Many to Many* / Banyak ke Banyak (* : *)

setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Contohnya hubungan yang terjadi antara Mahasiswa dengan Mata Kuliah. Satu mahasiswa dapat mengikuti lebih dari satu mata kuliah dan satu mata kuliah dapat diikuti oleh lebih dari satu mahasiswa.



2. *Logical Record Structured (LRS)*

Menurut Hasugian dan Shidiq (2012:608) memberikan batasan bahwa *LRS* adalah “sebuah model sistem yang digambarkan dengan sebuah *diagram-ERD* akan mengikuti pola atau aturan permodelan tertentu dalam kaitanya dengan konvensi ke *LRS*”. Perubahan yang terjadi yaitu mengikuti aturan-aturan sebagai berikut (Hasugian dan Shidiq, 2012:608) :

- a. Setiap entitas akan diubah kebentuk kotak.
- b. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada *diagram-ER* 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi).
- c. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.



2.2.3. Pengujian Web

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Tujuan dari pengujian adalah untuk menjamin perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu mempresentasikan kajian pokok dari spesifikasi, Analisis, Perancangan, dan Pengkodean dari perangkat lunak itu sendiri.

Pengujian perangkat lunak ini menggunakan metode *black box*, pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

Black Box Testing merupakan metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja.

Menurut Sukanto dan Shalahuddin (2015:275) menjelaskan bahwa “*Black-box Testing* (pengujian kotak hitam) 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.

