

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

A. Internet dan Website

1. Internet

Menurut Rosa dan Shalahuddin (2008:3) internet atau internetwork adalah “sekumpulan jaringan berbeda yang saling terhubung bersama sebagai satu kesatuan dengan menggunakan berbagai macam *protocol*, salah satunya adalah *protocol TCP/IP (Transmission Control Protocol / Internet Protocol)*”.

TCP/IP adalah *protocol* 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 rusak atau hilang.

2. Website

Menurut Arief (2011:8) pengertian website adalah “kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (*Uniform Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya”. Hal ini dimungkinkan dengan adanya teknologi *World Wide Web (WWW)* fasilitas *hypertext* guna menampilkan data berupa teks, gambar, animasi, suara dan multimedia lainnya data tersebut dapat saling berhubungan pada web server untuk dapat di akses melalui jaringan internet. Agar data pada web dapat dibaca kita harus menggunakan web server

terlebih dahulu seperti Mozilla Firefox, Internet Explorer, Opera Mini atau yang lainnya.

Situs website di kategorikan menjadi 2 yaitu website statis dan website dinamis, website statis adalah website yang berisi atau menampilkan informasi-informasi yang sifatnya tetap (statis), sedangkan website dinamis adalah website yang menampilkan informasi serta dapat berinteraksi dengan *user*.

B. Bahasa Pemrograman

Dalam pembangunan sebuah aplikasi berbasis web dapat digunakan berbagai macam bahasa pemrograman. Program dibuat untuk menangani sesuatu yang secara terus-menerus dan berulang kali. Beberapa contoh bahasa pemrograman yang biasa digunakan dalam pembuatan sebuah website, antara lain:

1. PHP (*Perl Hypertext Preprocessor*)

Menurut Aditya (2011:1) PHP: *Hypertext Preprocessor* adalah “bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML”. PHP banyak dipakai untuk memprogram situs web dinamis.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

2. HTML (*Hyper Text Markup Language*)

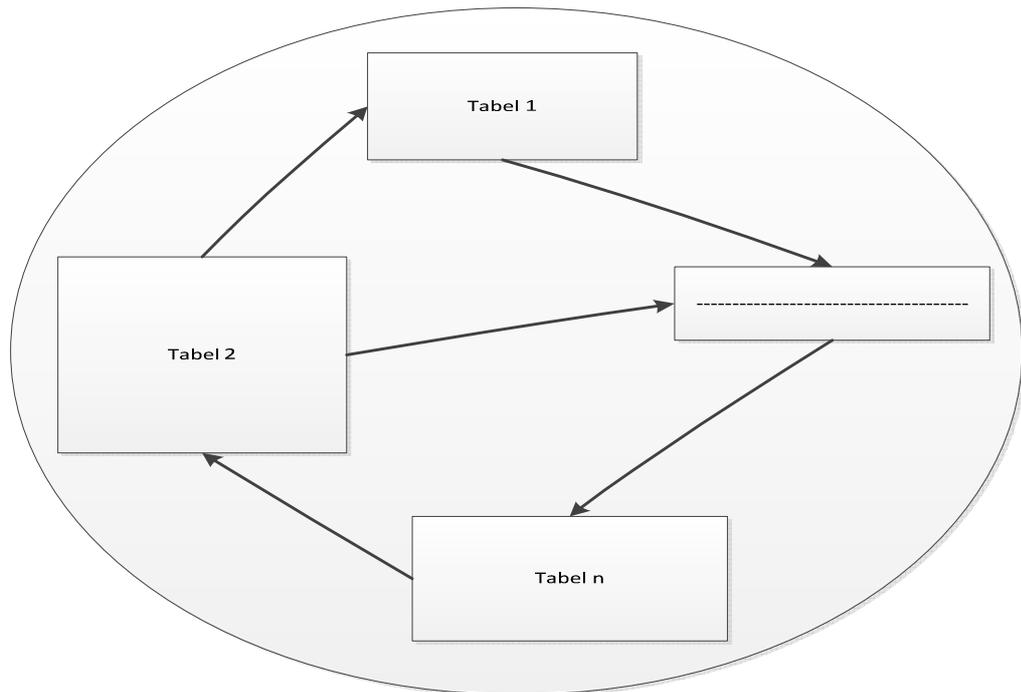
Menurut Sugiri dan Budi Kurniawan (2007:1) HTML (*Hypertext Markup Language*) adalah “sebuah *protocol* yang digunakan untuk membuat format suatu

dokumen web yang mampu dibaca dalam *browser* dari berbagai *platform* computer”. Sifat bahasa HTML ini adalah *client script*, dimana dokumen tersebut dapat dibuka dalam komputer *stand alone* yang tidak membutuhkan server untuk dapat menampilkannya di dalam *browser*. Dokumen HTML merupakan file yang pada umumnya berekstensi .html, dimana bahasa HTML tersebut tersusun atas tag berformat <isi tag>.

Secara teknis, web *browser* akan melakukan penerjemahan dokumen untuk ditampilkan dalam bentuk HTML jika dokumen tersebut terdapat diantara tag <html> dan </html>. Ada baiknya dalam penulisan HTML menggunakan huruf kecil untuk menghadapi HTML generasi berikutnya (rekomendasi dari W3C untuk HTML 4.0 dan XHTML).

C. Basis Data

Menurut Rosa dan Shalahuddin (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. Pada buku ini menggunakan basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi seperti pada gambar berikut.



Sumber: Rosa dan Shalahuddin (2013:44)

Gambar II.1.
Ilustrasi Basis Data

1. MySQL (*My Structure Query Language*)

Menurut Arief (2011:151) MySQL (*My Structure Query Language*) adalah “salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya”. MySQL bersifat *open source* dan menggunakan SQL (*Structured Query Language*). MySQL biasa dijalankan di berbagai *platform* misalnya windows Linux, dan lain sebagainya.

MySQL merupakan DBMS yang *multithread*, multi *user* yang bersifat gratis di bawah lisensi GNU *General Public Licence* (GPL). Tidak seperti Apache

yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau *open source* sehingga kita bisa menggunakannya secara gratis.

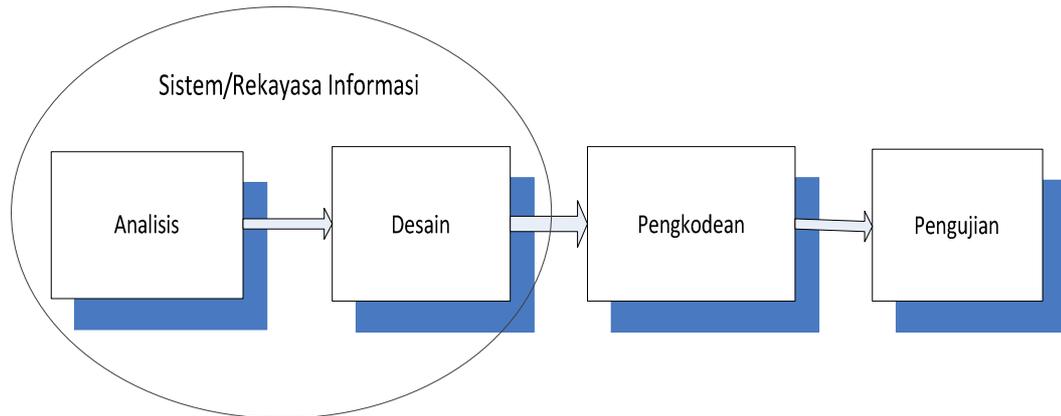
2. XAMPP

Menurut Aditya (2011:16) XAMPP adalah “perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program”. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, My SQL Database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat system operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public Licence* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

D. Model Pengembangan Perangkat Lunak

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

Berikut adalah gambar model air terjun :



Sumber: Rosa dan Shalahuddin (2013:27)

Gambar II.2.

Ilustrasi model *waterfall*

2.2 Teori Pendukung

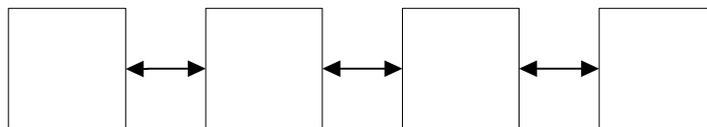
A. Struktur Navigasi

Menurut Yuhefizar (2008:67), menyimpulkan bahwa: struktur navigasi adalah "struktur alur cerita dari sebuah program". Sebelumnya menyatuhkan elemen-elemen yang digunakan dalam *aplikasi* multimedia, sebaiknya kita mendefinisikan objek-objek dan merancang tampilan. Agar semua objek yang termasuk didalam *aplikasi* tersebut tidak mengalami kekacauan informasi, dengan kata lain semua tampilan harus dapat memberikan informasi yang bulat dan utuh, sehingga dapat tercapai suatu pembentukan aplikasi multimedia.

Dalam pembuatan *aplikasi* multimedia ada empat macam struktur navigasi yang biasa digunakan, antara lain:

1. Struktur Navigasi *Linear*

Struktur Navigasi *Linear* merupakan struktur yang hanya mempunyai suatu rangkaian cerita yang berurutan. Struktur ini menampilkan satu demi satu tampilan *layer* secara berurutan menurut aturannya. 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. Pada struktur navigasi ini tidak diperkenankan adanya percabangan. Biasanya struktur ini digunakan untuk membuat multimedia persentasi karena tidak terlalu menurut keinteraksian tetapi hanya memerlukan keindahan dan kemudahan menampilkan data sebagai informasi. Berikut ini gambar struktur navigasi linier:



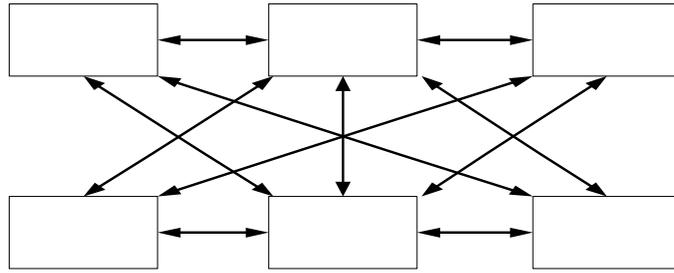
Sumber: Yuhefizar (2008:69)

Gambar II.3.

Navigasi *Linear*

2. Struktur Navigasi *Non Linear*

Struktur Navigasi *Non Linear* (tidak berurut) merupakan pengembangan dari struktur navigasi *linear*. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur *Non Linear* ini walaupun terdapat percabangan tetapi tiap-tiap tampilan mempunyai kedudukan yang sama, tidak ada *master page* dan *slave page*.



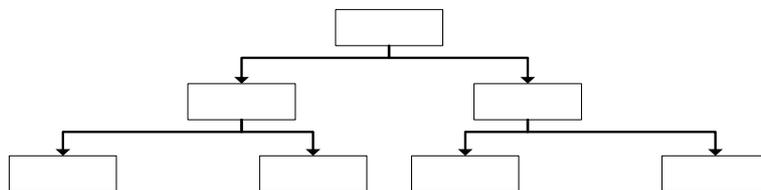
Sumber: Yuhefizar (2008:70)

Gambar II.4.

Navigasi *Non Linear*

3. Navigasi Hierarki

Struktur Hierarki (bercabang) merupakan suatu struktur yang mengandalkan percabangan 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. Pada struktur navigasi ini tidak dapat diperkenankan adanya tampilan secara *linear*.



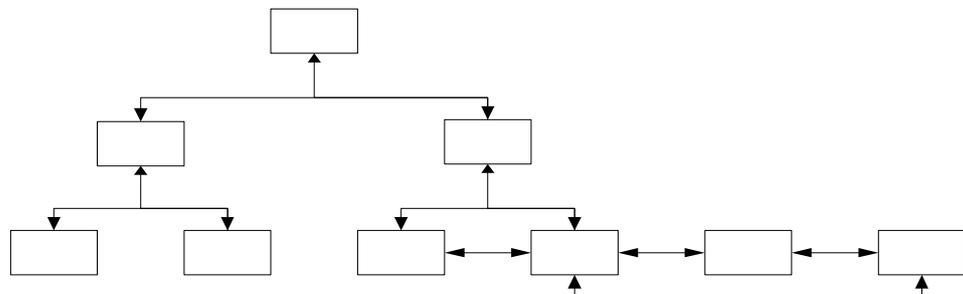
Sumber: Yuhefizar (2008:69)

Gambar II.5.

Navigasi Hierarki

4. Struktur Navigasi *Composite*

Struktur Navigasi *Composite* (campuran) merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non linear* dan *Hierarchy*. Struktur navigasi ini juga biasa disebut struktur navigasi bebas. 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. Struktur navigasi ini banyak digunakan dalam pembuatan website sebab struktur ini dapat memberikan keinteraksian yang lebih tinggi. Setiap struktur navigasi seperti yang baru dibahas mempunyai fungsi dan tujuan masing-masing. Tidak ada yang lebih baik atau lebih buruk. Penggunaan navigasi tergantung kepada kebutuhan dan tujuan dari aplikasi website yang hendak dibuat. Semakin kompleks navigasi yang digunakan, maka makin sulit pula pembuatan aplikasi website dari navigasi tersebut.



Sumber: Yuhefizar (2008:70)

Gambar II.6.

Navigasi Composite

B. *Entity Relationship Diagram*

Menurut Sutanta (2011:91), “*Entity Relationship Diagram* (ERD) adalah suatu model data yang dikembangkan berdasarkan objek, digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis”. Penggunaan ERD relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, berguna untuk memodelkan sistem yang nantinya akan dikembangkan. Diagram hubungan entitas yang biasa lebih dikenal dengan sebutan ER *diagram*, adalah notasi grafik dari sebuah model yang data atau sebuah model jaringan yang menjelaskan tentang data yang tersimpan (*storage data*) dalam sistem secara abstrak. Diagram hubungan entitas tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data. Adapun elemen-elemen diagram hubungan entitas antara lain:

1. Entitas (*entity*)

Pada E-R diagram, *entity* digambarkan dalam sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama yaitu: orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

2. *Attribute Value*

Attribute Value atau nilai atribut adalah suatu *occurrence* tertentu dari sebuah *attribute* didalam suatu *entity* atau *relationship*. Ada dua jenis *attribute* yaitu *identifier (key)* digunakan untuk menentukan suatu *entity* secara unik (*primary key*) dan *Descriptor (non key attribute)* digunakan untuk menspesifikasikan karakteristik dari suatu *entity* yang tidak unik.

3. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lain. Pada contoh sebelumnya dapat kita lihat bahwa tupel-tupel pada entitas mahasiswa dapat berelasi dengan satu tupel, banyak tupel atau bahkan tidak satupun tupel dari entitas kuliah. Begitu juga sebaliknya, entitas-entitas pada entitas kuliah ada yang berelasi dengan beberapa tupel pada entitas mahasiswa dan ada pula yang berelasi dengan satu tupel pada entitas mahasiswa.

4. Relasi

Relasi adalah hubungan antara beberapa entitas. sebagai contoh relasi antar mahasiswa dengan mata kuliah dimana setiap mahasiswa bisa mengambil beberapa mata kuliah dan setiap mata kuliah bisa diambil oleh lebih dari 1 mahasiswa. relasi tersebut memiliki hubungan banyak ke banyak. Macam-macam relasi antara lain:

a. *one-to-one*:

Sebuah entitas pada A berhubungan dengan entitas B paling banyak 1 contoh diatas relasi pegawai dan departemen dimana setiap pegawai hanya bekerja pada 1 departemen.

b. *one-to-many*:

Sebuah entitas pada A berhubungan dengan entitas B lebih dari satu contoh diatas adalah 1 departemen memiliki banyak pegawai.

c. *many-to-many*:

Sebuah entitas pada A berhubungan dengan entitas B lebih dari satu dan B berhubungan dengan A lebih dari satu juga contoh diatas adalah relasi mahasiswa dengan mata kuliah.

C. *Logical Record Structure (LRS)*

Membentuk skema database atau LRS (*Logical Record Structure*) berdasarkan diagram ERD (Simarmarta, 2007:115).

1. Jika relasinya satu-ke-satu, maka *foreign key* diletakkan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu ke banyak, maka *foreign key* diletakkan pada entitas many.
3. Jika relasinya banyak-ke-banyak, maka dibuat "file konektor" yang berisi dua *foreign key* yang berasal dari kedua entitas.

Membentuk beberapa tabel berdasarkan *primary key* yang terpilih dengan syarat sudah mencapai aturan normalisasi sekurang-kurangnya 3NF dari skema DB atau LRS yang ada.

D. *Pengujian Web*

Proses yang harus dilakukan setelah selesai membangun sebuah website ataupun perangkat lunak adalah proses pengujian (*testing*). Pengujian website adalah suatu teknik yang digunakan menguji apakah sebuah website yang dibangun telah memenuhi kebutuhan proses bisnis pengguna atau masih belum. Salah satu metode pengujian website adalah *Black-Box Testing*.

1. *Black Box Testing*

Menurut Rizky (2011:264), “*Black Box 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 dikenai proses *testing* dibagian luar. Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan.