

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

2.1.1. Website

A. Sejarah Website

Penemu situs web adalah Sir Timothy John "Tim" Berners-Lee, sedangkan situs web yang tersambung dengan jaringan pertama kali muncul pada tahun 1991. Maksud dari Tim ketika merancang situs web adalah untuk memudahkan tukar menukar dan memperbarui informasi pada sesama peneliti di tempat ia bekerja. Pada tanggal 30 April 1993, CERN (tempat di mana Tim bekerja) mengumumkan bahwa WWW dapat digunakan secara gratis oleh publik.

Sebuah situs web bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari suatu organisasi, perusahaan. biasanya pembahasan dalam sebuah situs web merujuk pada sebuah ataupun beberapa topik khusus, atau kepentingan tertentu. Sebuah situs web bisa berisi pranala yang menghubungkan ke situs web lain, demikian pula dengan situs web lainnya. Hal ini terkadang membuat perbedaan antara situs web yang dibuat oleh individu ataupun perseorangan dengan situs web yang dibuat oleh organisasi bisnis menjadi tidak begitu jelas.

Situs web biasanya ditempatkan pada *server web*. Sebuah *server web* umumnya telah dilengkapi dengan perangkat-perangkat lunak khusus untuk menangani pengaturan nama ranah, serta menangani layanan atas protokol HTTP

yang disebut sebagai Server HTTP (bahasa Inggris: *HTTP Server*) seperti *Apache HTTP Server*, atau *Internet Information Services (IIS)*.

B. Pengertian Website

Menurut Hidayat (2010:1) “*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 bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

Website atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang paling berkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

2.1.2. Web Server

Menurut Supardi (2010:2) menjelaskan bahwa “*web server* merupakan perangkat lunak yang mengelola (mengatur) permintaan *user* dari *browser* dan hasilnya dikembalikan lagi ke *browser*”. Contoh *web server* adalah, IIS (*Internet Information Services*) produk *Microsoft Corp*.

Umumnya setiap *web server* mempunyai kemampuan untuk melakukan pencatatan/*logging* terhadap informasi detail mengenai permintaan klien dan respon dari *web server* dan disimpan dalam berkas log, dengan adanya berkas log ini maka akan memudahkan *web master* untuk mendapat statistik dengan

menggunakan *tool log analyzer*. Pada penggunaan sehari-hari banyak *web server* mengimplementasikan fitur-fitur berikut:

1. Otentifikasi fitur untuk mengotorisasi suatu permintaan dari klien sebelum menggunakan sumber daya yang dimiliki oleh *web server* (biasanya *User* dan *Password*).
2. Dukungan HTTPS (dengan SSL, atau TLS) yang memungkinkan koneksi yang aman (dengan enkripsi) ke *server* pada *port* 433 berbeda dengan koneksi HTTP biasa di *port*.
3. Pengatur *Bandwith* (*Bandwith throttling*) yang berfungsi untuk membatasi kecepatan respon dengan tujuan tidak membanjiri jaringan dan menghemat pita data (*bandwith*) agar dapat melayani klien lebih banyak.

Setiap *server web* telah menentukan batas beban yang dapat ditanggung, sehingga setiap *server web* mempunyai batasan jumlah klien yang terhubung pada satu waktu (umumnya antara 2 s.d. 60.000, secara bawaan (*default*) maksimal 500 atau 1000). Jika *server web* sudah mendekati batas limit yang mampu ditangani maka *server* menjadi kelebihan beban sehingga lambat dalam memberikan respon.

2.1.3. Web Browser

Menurut Limantara (2009:1) menjelaskan bahwa *web browser* merupakan aplikasi perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan *teks*, *image*, *video*, *games*, dan informasi lainnya yang berlokasi pada halaman *web* pada *Word 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 *Word Wide Web*, *browser* juga bisa digunakan untuk mengakses informasi yang disediakan *web server* pada jaringan privat atau pada *system file*. Contoh *web browser*: *Microsoft Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*, dan lain-lain.

2.1.4. Internet

A. Sejarah Internet

Internet merupakan jaringan komputer yang dibentuk oleh Departemen Pertahanan Amerika Serikat di tahun 1969 melalui proyek ARPA yang disebut ARPANET (*Advanced Research Project Agency Network*) dimana mereka mendemonstrasikan bagaimana dengan *hardware* dan *software komputer* yang berbasis UNIX, kita bisa melakukan komunikasi dalam jarak yang tidak terhingga melalui saluran telepon.

Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan dan akhirnya semua standar yang mereka tentukan menjadi cikal bakal pembangunan protokol baru yang sekarang dikenal sebagai TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Tujuan awal dibangunnya proyek itu adalah untuk keperluan militer. Pada saat itu Departemen Pertahanan Amerika Serikakt (*US Department of Defense*) membuat sistem jaringan komputer yang tersebar dengan menghubungkan komputer di daerah-daerah vital untuk mengatasi masalah bila terjadi serangan nuklir dan untuk menghindari terjadinya informasi terpusat, yang apabila terjadi perang dapat mudah dihancurkan.

Pada mulanya ARPANET hanya menghubungkan 4 situs saja yaitu *Stanford Research Institute, University of California, Santa Barbara, University of Utah*, dimana mereka membentuk satu jaringan terpadu di tahun 1969 dan secara umum ARPANET diperkenalkan pada bulan oktober 1972. Tidak lama kemudian proyek ini berkembang pesat di seluruh daerah dan semua universitas di negara tersebut ingin bergabung, sehingga membuat ARPANET kesulitan untuk mengaturnya.

Oleh sebab itu ARPANET dipecah menjadi dua, yaitu MILNET untuk keperluan militer dan ARPANET baru yang lebih kecil untuk keperluan non-militer, seperti universitas-universitas. Gabungan kedua jaringan akhirnya dikenal dengan nama DARPA *internet*, yang kemudian disederhanakan menjadi *internet*.

B. Pengertian Internet

Menurut Irwansyah (2014:12) “*Internet* adalah sistem global jaringan komputer yang saling menghubungkan antara satu dengan yang lain diseluruh penjuru dunia”. Adapun standar yang digunakan disebut *Internet Protocol Suite* (TCP/IP). Internet digunakan diseluruh dunia untuk menghubungkan jutaan bisnis, agen pemerintahan, institusi akademis dan individu. Beberapa dari kegunaan itu adalah :

- a. Sebagai sarana komunikasi.
- b. Untuk penelitian sebagai alat pengakses informasi.
- c. Sarana untuk belanja (*online shopping*).
- d. Bank dan investasi.
- e. Hiburan.

- f. Berbagi informasi, foto, video.
- g. Mengakses aplikasi di *internet*.
- h. Alat pengajaran *online*.

2.1.5. World Wide Web (WWW)

Menurut Yuhefizar (2008a:159) *world wide web* sering disingkat *www* atau *web* adalah suatu metode untuk menampilkan informasi di *internet*, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*.

Harus diketahui bahwa WWW bukanlah *internet* demikian pula sebaliknya. Namun demikian, WWW dan *internet* sangat berkaitan satu sama lain. *Internet* adalah suatu jaringan komputer global, sedangkan WWW bukan sekedar jaringan tetapi didalamnya terdapat suatu set aplikasi komunikasi dan sistem perangkat lunak yang memiliki karakteristik sebagai berikut:

1. Umumnya terletak pada *internet host* dan *client*.
2. Umumnya menggunakan protokol TCP/IP.
3. Mengikuti model *client/server* untuk komunikasi dua arah.
4. Memungkinkan *client* untuk mengakses *server* dengan berbagai protokol seperti *HTTP*, *FTP*, *TELNET* dan *Gopher*.

2.1.6. XAMPP

Menurut Wicaksono (2008:7) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah 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*.

Fungsi XAMPP sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis.

2.1.7. HTTP (*Hypertext Transfer Protocol*)

Menurut Laudon (2007:326) “HTTP adalah standar komunikasi yang digunakan untuk *transfer* halaman web”. HTTP merupakan rangkaian huruf pertama yang terletak pada awal dari setiap alamat web, diikuti oleh alamat domain yang menentukan komputer *server* mana dalam perusahaan yang menyimpan dokumen tersebut. Kebanyakan perusahaan memiliki nama domain yang sama atau berkaitan erat dengan nama perusahaan yang resminya.

HTTP merupakan sebuah protokol yang menentukan aturan main antara *software browser (client)* dan *software server* dalam penyediaan dokumen yang diminta *browser*.

2.1.8. PHP (*Hypertext Preprocessor*)

Menurut Zaki (2008:2) “PHP adalah sebuah bahasa pemrograman *scripting* untuk membuat halaman web yang dinamis”. Walaupun dikenal sebagai bahasa untuk membuat halaman web, tapi PHP sebenarnya juga dapat digunakan untuk membuat aplikasi *command line* dan juga GUI.

Website yang dibuat menggunakan PHP memerlukan *software* bernama *webserver* tempat pemrosesan kode PHP dilakukan. *Server web* yang memiliki *software PHP Parser* akan memproses input berupa kode PHP dan menghasilkan *output* berupa halaman web. PHP bersifat terbuka dan *multiplatform*, karenanya dapat dijalankan di banyak merk *web server* (seperti Apache dan IIS). Saat ini pengguna PHP sangat banyak, diklaim ada lebih dari 20 juta *website* yang menggunakan PHP yang dijalankan dari lebih 1 juta *server*.

2.1.9. HTML (*Hypertext Markup Language*)

Menurut Solichin (2013:10) HTML merupakan singkatan dari *Hypertext Markup Language*). HTML dikembangkan pertama kali oleh Tim Berners-Lee bersamaan dengan protokol HTTP (*Hypertext Transfer Protocol*) Pada tahun 1989. Tujuan pengembangan HTML adalah untuk menghubungkan satu halaman web dengan halaman web lainnya. Tentunya pada awal perkembangannya, halaman web hanya berupa teks, tidak seperti sekarang.

Pada dasarnya, setiap halaman web ditulis dalam bentuk HTML. HTML merupakan bahasa pemrograman web yang memberitahukan peramban web (*web browser*) bagaimana menyusun dan menyajikan konten di halaman web. Dengan

kata lain, HTML adalah pondasi web. HTML disusun dengan bahasa yang sederhana, sehingga sangat mudah diimplementasikan.

2.1.10. CSS (*Cascading Style Sheet*)

Menurut Sya'ban (2010:37) “CSS adalah Suatu bahasa yang dikhususkan untuk mengatur gaya atau layout sebuah halaman web dan juga pengakses halaman web, untuk mendefinisikan warna, huruf *layout*, dan aspek-aspek presentasi dokumen lainnya”. CSS memang didesain untuk memisahkan antara isi dokumen (yang ditulis menggunakan HTML atau bahasa lain yang sejenis) dengan ini memberikan keuntungan akan adanya peningkatan dalam aksesibilitas isi, menyediakan fleksibilitas lebih, dan pengendalian terhadap spesifikasi karakteristik bentuk presentasi, serta mereduksi kompleksitas dan perulangan-perulangan pada struktur isi.

Selain HTML, suatu bentuk bahasa markup lainnya, yaitu XML juga dapat memanfaatkan CSS ini untuk mengatur gaya tampilan yang diinginkan. Dalam implementasinya, CSS didefinisikan melalui kode-kode *style sheet* yang dapat disertakan pada file HTML itu sendiri, atau terpisah dari file HTML. Jika terpisah maka *file style sheet* tersebut harus diberi ekstensi *.css* untuk menandaibahwa file tersebut adalah file CSS. Setelah itu, file CSS dapat direferensi oleh file HTML.

2.1.11. JavaScript

Menurut Sunyoto (2007:17) “JavaScript adalah bahasa *scripting* yang populer di *internet* dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer (IE)*, *Mozilla FireFox*, *Netscape* dan *Opera*. Kode JavaScript

dapat disisipkan dalam halaman web menggunakan *tag SCRIPT*". Beberapa hal tentang JavaScript :

1. *JavaScript* didesain untuk menambah interaktif suatu web.
2. *JavaScript* merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. *JavaScript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *JavaScript* biasanya disisipkan (*embedded*) dalam halaman HTML.
6. *JavaScript* adalah bahasa *interpreter* (yang berarti skrip dieksekusi tanpa proses kompilasi).
7. Setiap orang dapat menggunakan *JavaScript* tanpa membayar lisensi.

Untuk menjalankan *JavaScript*, kita hanya membutuhkan aplikasi *text editor*, dan *web browser*. *JavaScript* memiliki fitur: *high-level programming language*, *client-side*, *loosely typed*, dan berorientasi objek.

2.1.12. Basis Data (*Database*)

Menurut Yuhefizar (2008b:2) menjelaskan bahwa "*Database* diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (*relation*), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi". Secara prinsip, dalam suatu *database* tercangkup dua komponen penting, yaitu data dan informasi. Data adalah fakta, baik berupa sebuah objek, orang dan lain-lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambil keputusan.

A. MySQL

Menurut Anhar (2010:21) mengatakan bahwa “*MySQL (My Structured Query Language)* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain”. *MySQL* merupakan DBMS yang *multithread, multi-user* yang bersifat gratis dibawah 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.

MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu *MySQL AB*. *MySQL AB* memegang hak cipta kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL AB* adalah : David Axmark, Allan Larson, dan Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya, *MySQL* bersifat gratis atau *open source* sehingga kita bisa menggunakannya secara gratis. Pemograman PHP juga sangat mendukung/*support* dengan *database MySQL* sehingga apabila kita mempelajarinya dengan sungguh-sungguh kita dapat mengaplikasikan *PHP & MySQL* dalam membuat aplikasi *website* maupun dalam membuat *website*.

Beberapa kelebihan *MySQL*, antara lain :

- a. *MySQL* dapat berjalan dengan stabil pada berbagai sistem operasi, seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris*, dan masih banyak lagi.
- b. Bersifat *Open Source*, *MySQL* didistribusikan secara *open source* (gratis), dibawah lisensi *GNU General Public Licence (GPL)*.

- c. Bersifat *Multiuser*, *MySQL* dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
- d. *MySQL* memiliki kecepatan yang baik dalam menangani *query* (perintah SQL). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.
- e. Dari segi *security* atau keamanan data, *MySQL* memiliki beberapa lapisan sekuriti, seperti *level subnet mask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* yang terenkripsi.
- f. Selain *MySQL* bersifat fleksibel dengan berbagai pemrograman, *MySQL* juga memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- g. Dukungan banyak komunitas, biasanya tergabung dalam sebuah forum untuk saling berdiskusi membagi informasi tentang *MySQL*.

B. *PHPMysqlAdmin*

Menurut Rahman (2013:21) “*PHPMysqlAdmin* adalah *software* berbasis pemrograman PHP yang dipergunakan sebagai *administrator MySQL* melalui *browser* (web) yang digunakan untuk *management database*”. *PHPMysqlAdmin* mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, *table*, relasi antar *table*, dan lain sebagainya.

Pada dasarnya, penggunaan data pada *MySQL* dilakukan dengan mengetikkan perintah berupa kode-kode tertentu yang sesuai untuk tujuan tertentu, tetapi hal itu sangat menyulitkan karena penggunaannya harus mengingat setiap baris perintah agar pemrograman dapat berjalan dengan baik. Kemudian

diciptakanlah *software* yang dapat mempermudah pekerjaan pengelolaan data *MySQL*, salah satunya adalah *PHPMyadmin*.

2.1.13. Model Pengembangan Perangkat Lunak

Menurut Rosa dan M.Shalahuddin (2013:25) menjelaskan bahwa “pada awal pengembangan perangkat lunak, para pembuat program (*programmer*) langsung melakukan pengkodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak”. Dan ditemuilah kendala-kendala seiring dengan perkembangan skala sistem-sistem perangkat yang semakin besar.

A. SDLC

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

Seperti halnya proses metamorfosis pada kupu-kupu, untuk menjadi kupu-kupu yang indah maka dibutuhkan beberapa tahap untuk dilalui, sama halnya dengan membuat perangkat lunak, memiliki daur tahapan yang dilalui agar menghasilkan perangkat lunak yang berkualitas.

B. Model pengembangan air terjun (*Waterfall*)

Pada tahun 1960-an dan 1970-an, proyek pengembangan perangkat lunak merupakan pekerjaan yang sangat memakan biaya dan waktu karena

pengembangan perangkat lunak ini difokuskan pada perencanaan dan pengendalian. Kemunculan model air terjun adalah untuk membantu mengatasi kerumitan yang terjadi akibat proyek-proyek pengembangan perangkat lunak. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). (Simarmata 2010:53)

Berikut Merupakan Tahapan – tahapan Pengembangan Model *Sekuensial Linear / Waterfall Development Model* :

a. Analisis kebutuhan perangkat lunak

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

b. 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.

c. Pembuatan kode program

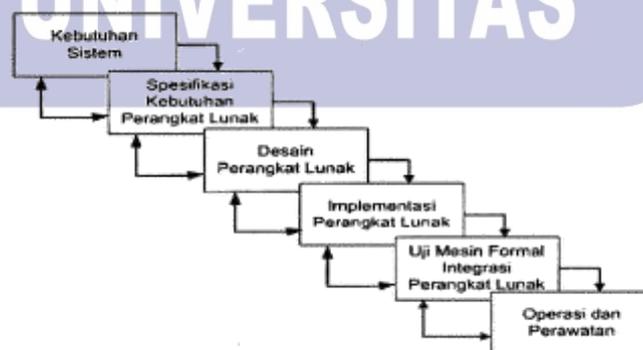
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. 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.

e. 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.



Sumber : Simarmata (2010:54)

Gambar II.1.

Penyajian Sederhana Dari Model Pengembangan Air Terjun

Model ini memungkinkan pemecahan misi pengembangan yang rumit menjadi beberapa langkah logis (desain, kode, pengujian, dan seterusnya) dengan beberapa langkah yang pada akhirnya akan menjadi produk akhir yang siap pakai. Untuk memastikan bahwa sistem bisa dijalankan, setiap langkah akan membutuhkan validasi, masukan, dan kriteria yang ada.

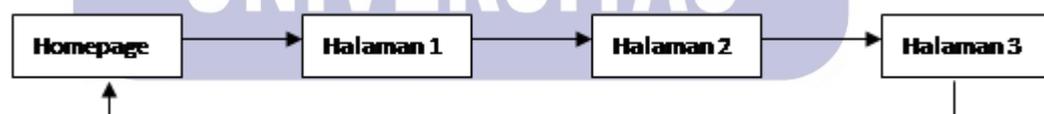
2.2 Teori Pendukung

2.2.1. Struktur Navigasi

Menurut Sutopo (2007:245) “Struktur Navigasi adalah struktur bagaimana halaman web dihubungkan dengan halaman lain”. Menentukan struktur navigasi merupakan hal yang sebaiknya dilakukan sebelum membuat *website*. Ada empat macam bentuk dasar struktur navigasi yaitu:

1. *Linear navigation model*

Digunakan oleh sebagian besar *website*. Informasi diberikan secara *sekuensial* dimulai dari satu halaman. Beberapa desainer web menggunakan satu halaman untuk masuk atau keluar dari *website*.



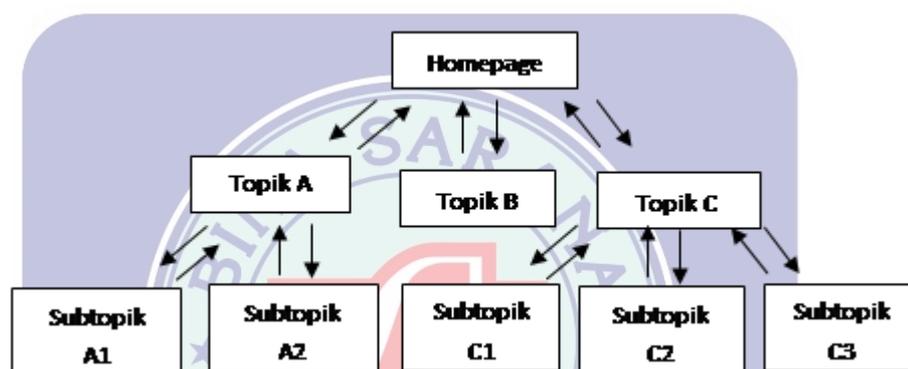
Sumber : Sutopo (2007:6)

Gambar II.2.

Struktur Navigasi *Linear*

2. Hierarchical model (*Model Hierarki*)

Diadaptasi dari *top-down design*. Konsep navigasi ini dimulai dari satu *node* yang menjadi *homepage*. Dari *homepage* dapat dibuat beberapa cabang ke halaman-halaman utama. Apabila diperlukan, dari tiap halaman utama dapat dikembangkan menjadi beberapa cabang lagi. Hal ini seperti struktur organisasi dalam perusahaan.



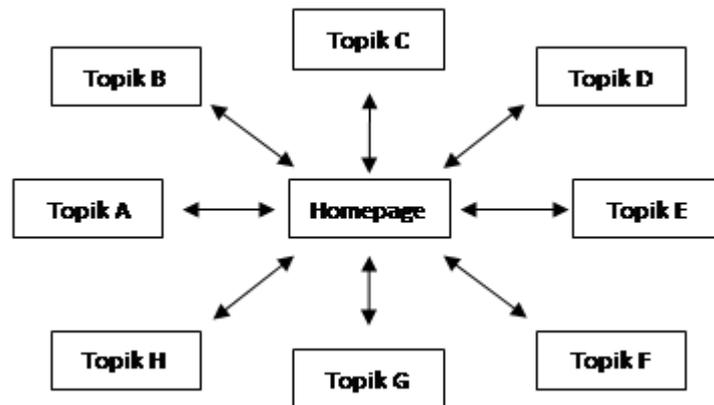
Sumber : Sutopo (2007:6)

Gambar II.3.

Struktur Navigasi Hierarki

3. Spoke-and-hub model

Hanya menggunakan satu *node* untuk berhubungan dengan *node* lain. Pada model ini hanya terdapat dua macam *link*, yaitu dari *homepage* ke halaman tertentu, dan dari halaman tersebut kembali ke *homepage*.

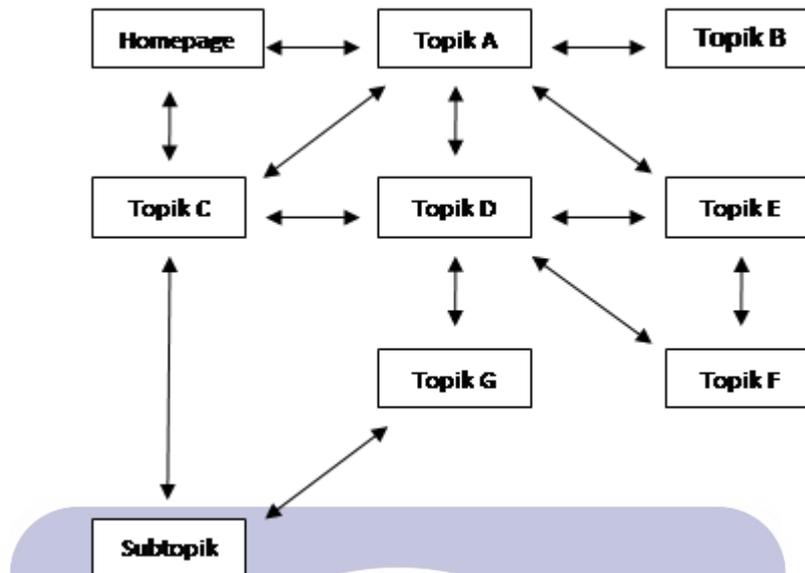


Sumber : Sutopo (2007:7)

Gambar II.4.
Struktur Navigasi Spoke-and-hub Model

4. Full web model

Memberikan kemampuan *hyperlink* yang banyak. *Full web model* banyak digunakan karena *user* dapat mengakses semua topik dengan subtopik dengan cepat. Namun kelemahan dari model ini, yaitu dapat berakibat *user* kehilangan cara untuk kembali ke topik sebelumnya.



Sumber : Sutopo (2007:8)

Gambar II.5.
Struktur Navigasi *Full Web Model*

2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Fathansyah (2012:79) *Entity Relationship Diagram (ERD)* adalah “diagram yang memperlihatkan entitas-entitas yang terlibat dalam suatu sistem serta hubungan-hubungan atau relasi antar entitas tersebut. Model *Entity-Relationship* yang berisi komponen-komponen himpunan entitas dan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari dunia Nyata yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram *Entity-Relationship*.”

Entity Relationship Diagram (ERD) menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada tiga macam komponen yang digunakan yaitu :

1. Entitas

Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain.

2. Atribut

Berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja. Setiap entiti bisa terdapat lebih dari satu atribut.

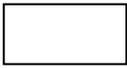
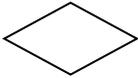
3. Hubungan atau (*Relationship*)

Sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai_ujian. Relationship disimbolkan dalam bentuk intan / *diamonds*.

Simbol-simbol yang digunakan dalam Entity Relationship Diagram dapat dilihat pada tabel II.1.

Tabel II.1

Simbol ERD

Simbol	Keterangan
	Entitas Melambangkan himpunan entitas.
	Relasi Melambangkan himpunan relasi
	Penghubung Melambangkan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

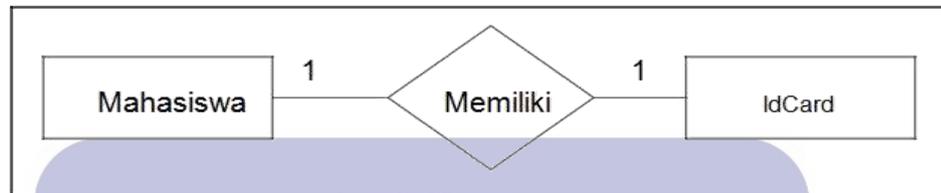
Sumber : (Fathansyah, 2012:80)

Jenis-jenis relationship (Fathansyah, 2007:80-81) :

1. Menurut Peter Chen :

a. *One – to – One*

Contoh :



Gambar II.6.

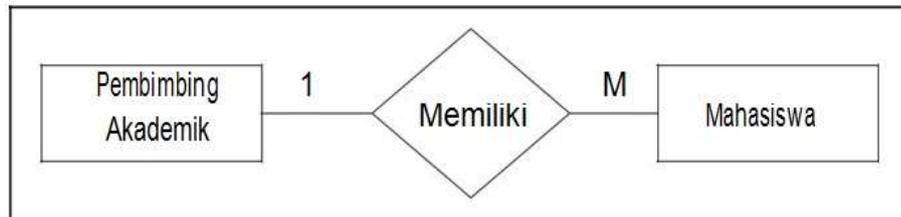
Relationship One to one

Sumber : (Fathansyah, 2012:80)

- 1) Setiap mahasiswa memiliki satu *Id Card*.
- 2) Satu *Id Card* hanya dimiliki oleh satu mahasiswa.

b One – to – Many (1:M atau M:1)

Contoh :



Gambar II.7.

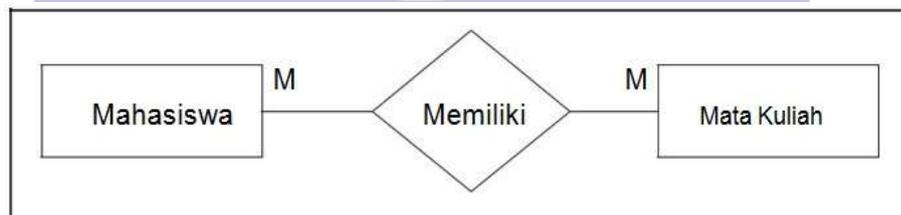
Relationship One to many

Sumber : (Fathansyah, 2012:81)

- 1) Pembimbing akademik memiliki banyak mahasiswa.
- 2) *Satu mahasiswa hanya memiliki satu pembimbing akademik.*

c Many – to – Many (M:M)

Contoh :



Gambar II.8.

Relationship Many to many

Sumber : (Fathansyah, 2012:81)

- 1) Mahasiswa memiliki banyak mata kuliah.
- 2) Mata kuliah dimiliki oleh banyak Mahasiswa.

2.2.3. Logical Record Structure (LRS)

Menurut Lestari (2013:15-16) *Logical Record Structure* dibentuk dengan nomor dari 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* diantara 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 yang lain dimulai dengan Entity Relationship diagram dan langsung dikonversikan ke LRS.

1. Konversi ERD ke LRS, Diagram *entity relationship* diagram harus diubah ke bentuk LRS (struktur *record* secara logik). 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 permodelan tertentu. Dalam kaitannya dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:
 - a. Setiap entitas diubah ke bentuk kotak dengan nama entitas, berada diluar kotak dan atribut berada didalam 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 dan 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 didalam relasi.

2.2.4. Pengujian Web

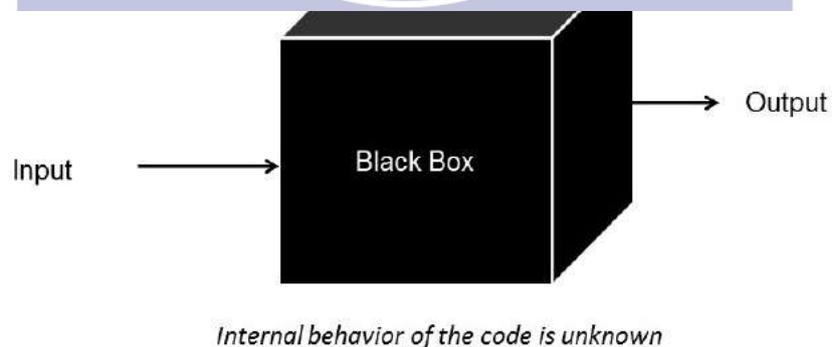
Menurut Simarmata (2010:323) “Pengujian adalah proses terhadap aplikasi. Program untuk menemukan segala kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan sesuai dengan spesifikasi perangkat lunak yang telah ditentukan sebelum aplikasi tersebut diserahkan kepada pelanggan”. Dari pengertian diatas dapat disimpulkan bahwa pengujian adalah proses terhadap aplikasi yang saling terintegrasi guna untuk menemukan kesalahan dan segala kemungkinan yang akan menimbulkan kesalahan.

Secara teoritis, *testing* dapat dilakukan dengan berbagai jenis tipe dan teknik. Namun secara garis besar, terdapat dua jenis tipe testing yang paling umum digunakan di dalam lingkup rekayasa perangkat lunak. Dua jenis tersebut adalah *Black box* dan *White box testing*.

Ada 2 jenis pengujian dalam proses aplikasi yaitu *Black Box* dan *White Box*. Namun dalam tahapan pengujian dilakukan oleh penulis saat ini menggunakan *Black Box Testing*. Penjelasan mengenai *Black Box testing* sebagai berikut:

Menurut Rizky dalam Nina Rahayu (2013) “*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 dikenai proses testing di bagian 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 *whitebox testing*, perangkat lunak tersebut akan berusaha dibongkar *listing* programnya untuk kemudian dites menggunakan teknik-teknik yang telah dijelaskan sebelumnya. Sedangkan pada jenis *blackbox testing*, perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar *listing* programnya.



Gambar II.9.

Skema Diagram *Black Box Testing*

Tujuan dilakukan *testing* pada suatu *software* yaitu: untuk mengidentifikasi dan mengungkapkan sebagai kesalahan sebanyak mungkin dalam perangkat lunak yang diuji, setelah memperbaiki kesalahan yang diidentifikasi dan melakukan pengujian ulang, pada tingkat kualitas yang memadai. Serta menyusun catatan kesalahan perangkat lunak untuk digunakan dalam pencegahan kesalahan (dengan tindakan perbaikan dan pencegahan).

Pengujian dengan metode *Black Box Testing* ini dilakukan dengan cara memberikan sejumlah input pada sistem. Input tersebut kemudian di proses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi dapat menghasilkan *output* yang sesuai dengan yang diinginkan dan sesuai pula dengan fungsi dasar dari sistem tersebut. Apabila dari input yang diberikan, proses dapat menghasilkan *output* yang sesuai dengan kebutuhan fungsionalnya, maka program yang dibuat sudah benar, tetapi apabila *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada sistem tersebut, dan selanjutnya dilakukan penelusuran perbaikan untuk memperbaiki kesalahan yang terjadi.



UNIVERSITAS