

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Untuk membangun sebuah *web* dibutuhkan sebuah bahasa pemrograman dengan sebutan *web scripting*. Dikatakan *script* karena perintah kode program tersebut akan di *interpreter* dan tidak ada kompilasi untuk menjadikannya *executable*. Berdasarkan letak proses *interpreter* maka *web scripting* dibagi menjadi dua kategori, yaitu yang bersifat *client side* dan *server side*. *Client side* dilakukan oleh *web browser* seperti *internet explorer*, *netscape*, *opera*, *firefox*. Untuk contoh bahasa *client side* adalah HTML, CSS, *Javascript*, VB *script*, dan XML. Sedangkan *server side* dilakukan oleh *web server* seperti PWS (*Personal Web Server* untuk Sistem Operasi Windows 98), IIS (untuk Sistem Operasi Windows 2000 atau Windows XP), *Apache*, *Tomcat*, *Xitami* dan *ZOPE*. Untuk contoh bahasa *server side* adalah *PHP*, *JSP*, *CFM*, dan *CGI* atau *PL*.

Web scripting yang bersifat *client side* akan menghasilkan *web page* yang statis artinya lebih menekankan pada desain format tampilan informasi dan informasi yang disajikan tidak dapat di *update* seketika, karena tidak dapat dilakukan *request* interaktif dari pengguna dan proses tidak terjadi di *server* sehingga tidak akan dapat menghasilkan *output*. Sehingga sangatlah tidak mungkin untuk menciptakan aplikasi *web* yang bersifat dinamis dengan *web*

scripting bersifat *client side* ini, tetapi harus dikombinasikan juga dengan *web scripting* yang bersifat *server side*.

Web statis adalah *web* yang berisi atau menampilkan informasi-informasi yang sifatnya statis (tetap), sedangkan *web* dinamis adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan *user* yang sifatnya dinamis. Sehingga untuk membuat *web* dinamis dibutuhkan kemampuan pemrograman *web*.

A. Website

Menurut Yuhefizar (2008:159) “*Website* adalah suatu metode untuk menampilkan informasi di internet, baik berupa *text*, *image*, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *web browser*”.

Dapat disimpulkan *Website* adalah sebuah tempat di Internet, yang menyajikan informasi dengan berbagai macam format data seperti *text*, *image*, bahkan video dan dapat diakses menggunakan berbagai *aplikasi client* sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi.

1. Internet

Menurut Yuhefizar (2008:2) “*Internet* adalah rangkaian hubungan jaringan komputer yang dapat diakses secara umum diseluruh dunia, yang mengirim data dalam bentuk paket data berdasarkan standar *Internet Protocol (IP)*”. Lebih dalam lagi, *Internet* adalah kumpulan jaringan dari jaringan-jaringan komputer dunia yang terdiri dari jutaan unit-unit kecil, seperti jaringan pendidikan, jaringan

bisnis, pemerintahan dan lain-lain, yang secara bersama menyediakan layanan informasi seperti *e-mail*, *online chat*, transfer file dan saling keterhubungan (*linked*) antara satu sama halaman *web* dengan sumber halaman lainnya.

Untuk beberapa perusahaan keuntungan yang diperoleh dari *internet* ditimbulkan dari kemampuan menjaga loyalitas konsumen, mengantisipasi kebutuhan konsumen mendatang dan memperbaiki pelayanan konsumen. Persaingan yang semakin hebat antara instusi penyedia produk belakangan ini bukan hanya disebabkan globalisasi. Tetapi disebabkan karena pelanggan semakin cerdas dan didekati oleh banyak produk. Kemajuan teknologi komunikasi juga ikut berperan akses informasi yang lebih banyak tentang berbagai macam produk yang ditawarkan.

2. Web Server

Menurut Sutisna (2007:5) “*Web Server* adalah halaman-halaman *website* yang diakses oleh *user* melalui *browser* disimpan pada *web server*”. Untuk itu, diperlukan program khusus agar *website* yang kita buat dapat diterima dengan baik oleh *client*. Ada beberapa *software* yang bisa digunakan baik gratis (*Open Source*) maupun yang komersial, diantaranya: *Netscape Server*, *Microsoft IIS*, *Xitami*, *Webstar* dan lain-lain. Contoh dari *web server* adalah *Apache*.

Menurut Kadir (2009:360) “*Apache* adalah otak dari *web server* anda. Aplikasi *web server* ini bertugas untuk mengkoordinasikan server anda dan akan menangani semuanya”. *Server HTTP Apache* atau *Server Web/WWW Apache* adalah *server web* yang dapat dijalankan di banyak sistem operasi (*Unix*, *BSD*, *Linux*, *Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang

berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web* atau *www* ini menggunakan *HTTP*.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *server* menjadi mudah.

Apache adalah perangkat lunak *web server* yang paling populer, hal ini memungkinkan komputer untuk menjadi satu *Host* atau lebih, situs yang dapat diakses melalui *internet* menggunakan *web browser*. Versi pertama *Apache* dirilis pada 1995 oleh *Grup Apache*. Pada tahun 1999 *Grup Apache* menjadi *Apache Software Foundation* sebuah organisasi nirlaba yang saat ini mempertahankan pengembangan perangkat lunak *Web server Apache*.

3. *Web Browser*

Menurut Sutisna (2007:5) “*Web Browser* adalah *software* untuk menampilkan halaman *website*”. Cara kerja *browser* adalah menerjemahkan kode program HTML ke dalam bentuk visual sesuai dengan apa yang dirancang oleh pembuat *website*.

Dapat dikatakan saat ini hanya ada beberapa *web browser* GUI yang populer: *Internet Explorer*, *Opera* dan *Mozilla Firefox*. Beberapa browser ini bersaing untuk merebut pemakainya dengan berusaha untuk mendekati standar spesifikasi dokumen *HTML* yang direkomendasikan oleh *W3C (World Wide Web Consortium)*.

4. *TCP dan IP*

TCP (Transmission Control Protocol) yang memastikan bahwa semua hubungan bekerja dengan benar, sedangkan *IP (Internet Protocol)* yang mentransmisikan data dari satu komputer ke komputer lain, *TCP* dan *IP* secara umum paspor digital yang memungkinkan perpindahan file-file dengan mudah di *internet*, sedikit mirip dengan perbedaan antara jalan biasa dengan jalan tol khusus. *TCP* dan *IP* terbuka bagi semua pengguna jalan, akhirnya dipilih sebagai standar resmi untuk *internet*.

5. *Domain Name Server (DNS)*

Nama *domain* adalah nama unik yang diberikan untuk mengidentifikasi nama *server* komputer seperti *web server* atau *email server* di jaringan komputer ataupun *internet*. Nama *domain* berfungsi untuk mempermudah pengguna *internet* pada saat melakukan akses ke *server*, selain itu juga dipakai untuk mengingat nama *server* yang dikunjungi tanpa harus mengenal deretan angka yang rumit yang dikenal sebagai alamat *IP*. Nama *domain* ini juga terkenal sebagai sebuah kesatuan dari sebuah situs *web*.

Dalam membeli *domain* untuk *website* harus membuat pertimbangan terlebih dahulu karena *domain* akan mewakili *brand website* dan atau perusahaan yang diusung. Memang tidak ada ukuran resmi mengenai *domain* yang bagus dan baik untuk membuat nama *domain* sesuai keinginan.

Nama *domain* sendiri mempunyai identifikasi ekstensi atau akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut. Contoh nama *domain* berekstensi internasional adalah *com, net, org, info, biz, name, ws*. Contoh nama *domain* berekstensi lokal negara Indonesia adalah:

- a. Korporasi (.co.id): Untuk badan usaha yang mempunyai badan hukum yang sah.
- b. Pendidikan (.ac.id): Untuk lembaga pendidikan.
- c. Pemerintahan (.go.id): Khusus untuk lembaga pemerintahan republik Indonesia.
- d. Militer (.mil.id): Khusus untuk lembaga militer republik Indonesia.

6. HTTP (*Hyper Text Transfer Protocol*)

Hyper Text Transfer Protocol (HTTP) adalah sebuah protokol untuk meminta dan menjawab antara *client* dan *server*". Sebuah *client HTTP* seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan *TCP* atau *IP* ke *port* tertentu ditempat yang jauh (biasanya *port* 80). Sebuah server *HTTP* yang mendengarkan di *port* tersebut menunggu client mengirim kode permintaan (*request*) yang akan meminta halaman yang sudah ditentukan, diikuti dengan pesan *Multipurpose Internet Mail Extension (MIME)* yang memiliki beberapa informasi kode kepala yang menjelaskan aspek permintaan tersebut diikuti dengan badan dari data tertentu.

HTTP berkomunikasi melalui *TCP* atau *IP*. *Client HTTP* terhubung ke *server HTTP* menggunakan *TCP*. Setelah membuat sambungan, *client* dapat mengirim pesan permintaan *HTTP* ke *server*. *HTTP* digunakan untuk mengirimkan permintaan dari *client web (browser)* ke *web server*, dikembalikan ke konten *web (halaman web)* dari *server* ke *client*.

B. Bahasa Pemrograman

Bahasa Pemrograman adalah instruksi standar untuk memerintah komputer melakukan suatu tindakan tertentu yang merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan mana yang data yang akan diproses oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan langkah-langkah apa yang persis jenis yang akan diambil dalam berbagai situasi.

1. *Hypertext Preprocessor (PHP)*

Menurut Anhar (2010:3) “*PHP* singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server* (*server side HTML Embedd scripting*)”.

PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script PHP* dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP adalah salah satu *server-side* programming yang paling banyak digunakan, *PHP* diciptakan oleh Rasmus Lerdorf, seorang pemrogram yang sangat handal. Pada saat ini *PHP* digunakan untuk mencatat jumlah pengunjung pada *homepage*. *PHP* mampu berhubungan dengan database dan dapat diintegrasikan dengan *HTML*. Pada tahun 1996 *HTML* dapat digunakan oleh banyak *website* di dunia, berdasarkan hasil survei dari *netcraft PHP* adalah salah satu bahasa *server-*

side yang paling populer didunia. *PHP* dapat berjalan dalam *web server* yang berbeda dan dalam menjalankan sistem operasi yang berbeda pula, akan tetapi *PHP* tidak bisa melakukan pengaturan format tampilan didalam *web browser*, misalnya mengatur warna *background* dan jenis huruf.

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya yaitu:

- a. *Script* (kode program) terintegrasi dengan *HTML*, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen *web* nya.
- b. Tidak ada proses *compiling* dan *linking*.
- c. Berorientasi objek (*Object Oriented*).
- d. Integrasi yang sangat luas ke berbagai *server database*, menulis *web* yang terhubung ke *database* menjadi sangat sederhana. *Database* yang didukung oleh *PHP* : *Oracle*, *Cybase*, *mSQL*, *MySQL*, *Solid*, *ODBC*, *PostgreSQL*, *Adabas D*, *Filepro*, *Velods*, *Informix*, *dBase*, *UNLX dbm*.

2. *Hyper Text Makrup Language (HTML)*

Menurut Kustiyahningsih (2011:13) “*Hyper Text Makrup Language (HTML)* adalah file *text* murni yang dapat dibuat dengan editor *text* sembarang, dokumen ini dikenal sebagai *web page*”.

Hyper Text Makrup Language (HTML) suatu sistem penulisan perintah dan *formating hypertext* sederhana yang ditulis kedalam dokumen teks *ASCII* agar dapat menghasilkan tampilan visual yang terintegrasi. Dengan kata lain, dokumen yang dibuat dalam aplikasi pengolahan kata dan disimpan kedalam format *ASCII* sehingga menjadi *homepage* dengan tambahan perintah-perintah *HTML*. Dokumen *HTML* disebut makrup language karena mengandung tanda-tanda

tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen.

HTML lebih menekankan pada penggambaran komponen-komponen, struktur dan *formatting* didalam halaman *web* dari pada menentukan penampilannya. Sedangkan *web browser* digunakan untuk menginterpretasikan perintah-perintah *HTML* yang disisipkan kedalam teks dan menampilkan susunan halaman ke *browser* dengan menggunakan *font*, *tab* , warna, garis, dan perataan teks yang dikehendaki ke komputer yang menampilkan halaman *web*. Struktur penulisan tag-tag yang ditempatkan pada dokumen yang terbagi kedalam area diskriptif dan fungsional.

3. *JavaScript*

Java adalah bahasa pemogramnan yang dapat dijalankan diberbagai komputer termasuk telepon genggam. Pada awalnya bahsa pemograman ini dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang ada pada 16 bahasa C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungn rutin atas bawah yang minimal. Aplikasi-aplikasi berbasis *java* pada umumnya dikompilasikan kedalam b-kode (*bytecode*) dan dapat dijalankan pada berbagai Mesin *Virtual Java (JVM)* (Wahana komputer,2010:5).

Perkembangan *java* diperkenalkan pertaman kali tahun 1995, bahasa yang sekerang dikenal dengan nama *JavaScript* ini sebelumnya bernama *LiveScript* dan fungsi sederhana untuk *browser Netscape Navigator 2* (browser yang sangat populer waktu itu). Awalnya bahasa ini memang sedikit banyak kritikan dengan

alasan kurang aman, pengembangannya yang tergesa-gesa, serta tidak ada pesan melalui kerjasama antar *Netscape* dan *Sun* (pengembangan bahasa pemrograman *Java*) pada saat itu, *Netscape* memberikan nama *JavaScript* kepada bahasa tersebut pada tanggal 4 Desember 1995. Pada waktu hampir bersamaan, pihak *Microsoft* sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai *JavaScript* sendiri masih terus dikembangkan pihak *Microsoft* dengan didukung oleh kebanyakan *browser* yang tersedia saat ini.

JavaScript merupakan bahasa yang berbentuk kumpulan skrip yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen *web*. Dengan kata lain, bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman *HTML (Hyper Text Markup Language)* dengan mengizinkan pengeksekusian 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 tidaklah memerlukan sebuah kompilator atau penerjemah khusus untuk 17 mengeksekusinya, hal tersebut juga bergantung pada *navigator* yang terdapat disetiap *browser*.

4. Adobe Dreamweaver CS6

Adobe Dreamweaver CS6 adalah sebuah editor *web* professional yang digunakan untuk mendesain dan mengelola situs *web* atau halaman *web*. *Dremweaver* paling sering digunakan oleh *web designer* atau *web programmer* dalam mengembangkan suatu situs *web*. Hal ini disebabkan area kerja, fasilitas, dan kemampuan *Dreamweaver* yang makin *powerfull* dan menunjang peningkatan

produktifitas dan efektifitas untuk mendesain atau membangun situs *web*. Dreamweaver juga dilengkapi dengan fasilitas untuk manajemen situs yang cukup

Adobe Dreamweaver CS6 mempunyai kehandalan yaitu dengan editing visualnya dapat mempermudah para pengguna untuk menciptakan halaman *web* dengan cepat dan mudah tanpa harus berlama-lama menuliskan baris-baris *Hyper Text Markup Language (HTML)*. *Adobe Dreamweaver CS6* juga memiliki kemampuan yang mendukung pemrograman *Script Server Side* seperti *Active server side (ASP)*, *Java Script Page (JSP)*, *Coldfusion*, *ASP.Net*, dan *PHP*.

C. Basis Data (Database)

Basis data atau database adalah “Mekanisme yang digunakan untuk menyimpan informasi atau data, informasi adalah suatu yang kita gunakan sehari-hari untuk berbagai alasan, dengan basis data pengguna dapat menyimpan data secara terorganisasi”.

Basis data atau Database adalah kumpulan data yang disimpan secara sistematis didalam komputer dan dapat diolah atau dimanupulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

1. SQL (Structured Query Language)

SQL adalah kumpulan perintah khusus yang digunakan untuk mengakses data dalam *database* rasional, yang merupakan bahasa komputer yang mematuhi standar ANSI (*American Nasional Standard Insistute*).

2. MYSQL

Menurut Anhar (2010:21) “*MYSQL* 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 License (GPL)*.

Seperti yang telah disebutkan sebelumnya, *MYSQL* bersifat gratis atau *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman *PHP* juga sangat mendukung dengan *database MYSQL* sehingga apabila kita mempelajari dengan sungguh-sungguh kita dapat mengaplikasikan *PHP & MYSQL* dalam aplikasi membuat *website* maupun dalam membuat *website*.

3. Php MyAdmin

“PhpMyAdmin adalah Sebuah software yang berbentuk seperti halaman situs yang terdapat pada *web server*”.

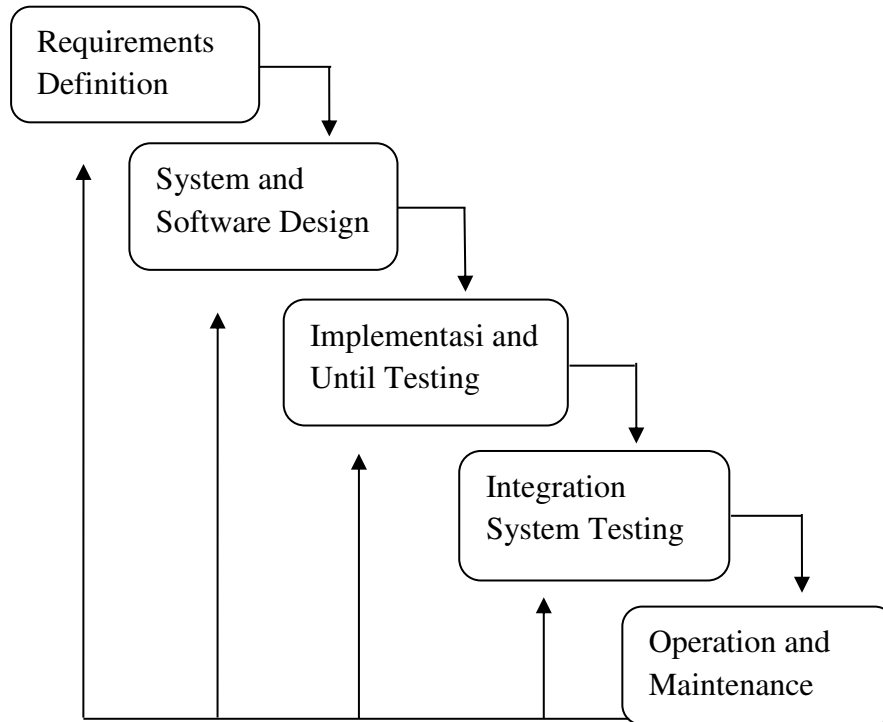
Fungsi dari halaman ini adalah sebagai pengendali *database MySQL* sehingga pengguna *MySQL* tidak perlu repot untuk menggunakan perintah-perintah *SQL*. Karena dengan adanya halaman ini semua hal tersebut dapat dilakukan hanya dengan meng-klik menu fungsi yang ada pada halaman *phpMyAdmin*.

D. Model Pengembangan Perangkat Lunak

Pada model pengembangan perangkat lunak ini menggunakan metode waterfall. Menurut Sommerville (2011:30) “ Metode *Waterfall* adalah proses pengembangan perangkat lunak dengan tahap-tahap utama dari model ini memetakan kegiatan-kegiatan pengembangan dasar”.

Gambar dibawah menjelaskan bahwa metode *waterfall* menekankan pada sebuah keterurutan dalam proses pengembangan perangkat lunak. Metode ini adalah sebuah metode yang tepat untuk membangun sebuah perangkat lunak yang

dak terlalu besar dan sumber daya manusia yang terlibat dalam jumlah yang terbatas.



Sumber :Sommerville (2011:30)

Gambar II.1.
Metode Waterfall

Berikut adalah penjelasan dari tahap-tahap yang dilakukan dalam metode *waterfall*:

1. Definisi Kebutuhan (*Requirements Definition*)

Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user* sistem. Persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem, mengumpulkan kebutuhan secara lengkap kemudian dianalisis yang harus dipenuhi oleh program yang akan dibangun.

2. Desain sistem (*System and Software Design*)

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya. Proses desain menerjemahkan syarat atau kebutuhan kedalam sebuah representasi perangkat lunak yang dapat di perkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagai persyaratan desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

3. Implementasi dan Pengujian Unit (*Implementasi and Until Testing*)

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

4. Integrasi dan Pengujian Sistem (*Integration and System Testing*)

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan. Pengujian eksternal fungsional untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

5. Operasi dan Pemeliharaan (*Operation Maintenance*)

Biasanya (walaupun tidak seharusnya), ini merupakan fase siklus hidup yang paling lama. Sistem diinstal dan dipakai. Pemeliharaan mencakup koreksi dari berbagai eror yang tidak ditemukan padatahap-tahap terdahulu,

perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan.

2.2. Teori Pendukung

Teori pendukung atau landasan teori dalam penyusunan tugas akhir ini sangat diperkukan karena sebagai referensi untuk menunjang atau memperdalam pemahaman terhadap informasi-informasi yang disajikan, teori pendukung yang berisi Struktur Navigasi, ERD dan LRS.

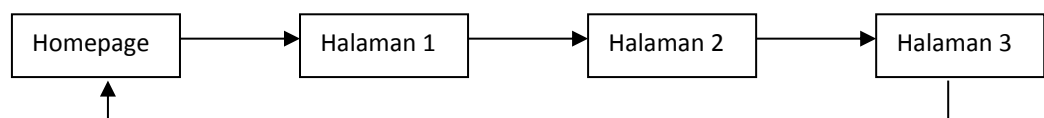
A. Struktur Navigasi

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*.

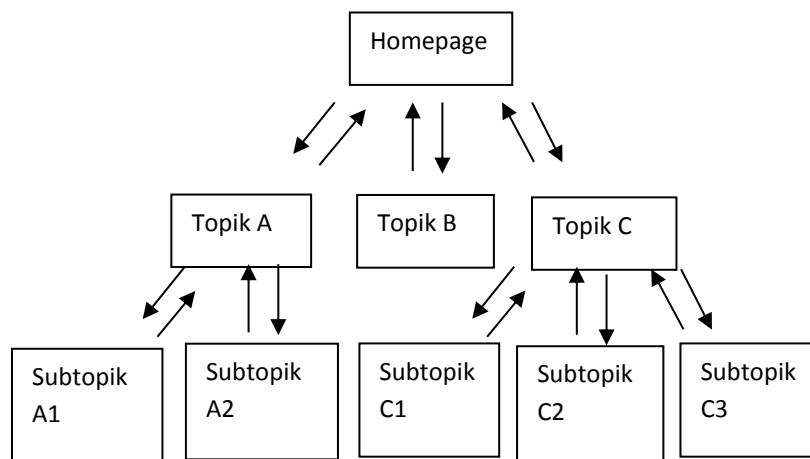


Sumber :Sutopo (2007:6)

Gambar II.2.
Struktur Navigasi Linear

2. *Hierarchical model*

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.

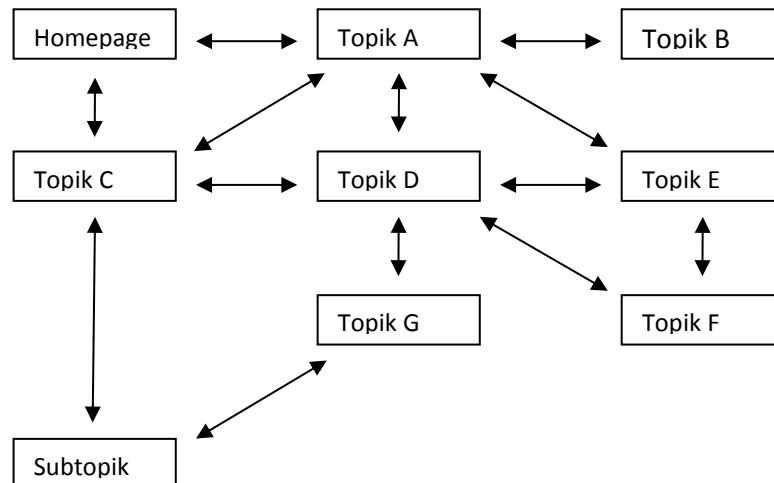


Sumber :Sutopo (2007:6)

Gambar II.3.
Struktur Navigasi *Hierarchical model*

3. *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.4.
Struktur Navigasi full web model

B. *Entity Relationship Diagram (ERD)*

“ERD (*Entity Relationship Diagram*) adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional”. Proses memungkinkan analis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien.

Adapun tujuan dari *Entity Relationship Diagram* ini adalah menunjukan objek data dan *relationship* yang ada pada objek tersebut. Selain itu model ERD merupakan salah satu alat untuk perancangan dalam basis data. Diagram hubungan data yang digunakan untuk menggambarkan hubungan antara data *store* yang ada didalam diagram alir data.

ERD juga merupakan gambaran yang menghubungkan antara objek suatu dengan objek yang lain dalam dunia nyata. Bisa dikatakan bahwa bahan yang akan di gunakan untuk membuat ERD adalah dari objek di dunia nyata. Sebagai contoh,jika akan membuat ERD dari sistem akademik suatu perguruan tinggi,

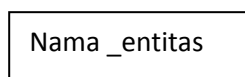
maka bahan sebagai objek ERD bisa berupa mahasiswa, dosen, ruang kelas, mata kuliah dan lain sebagainya. Secara umum ERD terdiri dari 3 komponen,yakin:

- a) Entitas (*Entity*)
- b) Atribut (*Attribute*)
- c) Relasi (*Relationship*)

1. Entitas (*Entity*)

Entitas merupakan suatu “objek nyata” yang mampu dibedakan dengan objek yang lain. Objek tersebut dapat berupa orang benda ataupun hal yang lainnya.

Penggambaran entitas dalam ERD seperti pada gambar II.6.



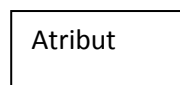
Sumber : Utami (2012:19)

Gambar II.5.
Enititas

1. Peran : Entitas berupa peran contoh adalah mahasiswa, dosen, karyawan, dan lain-lain.
2. Kejadian : Entitas berupa kejadian contohnya adalah mengejar, ambil mata kuliah dan lain-laian
3. Abstrak : Etentitas berupa abstrak contohnya adalah kurikulum, mata kuliah, pekerjaan dan lain-lain
4. Fisik : Etentitas berupa tempat contohnya adalah ruang kelas.

2. Atribut

Atribut merupakan semua informasi yang berkaitan dengan entitas. Di dalam dunia pemrograman, atribut adalah property dari suatu objek. Sebagai contoh jika entitas adalah manusia atau orang maka atributnya adalah rambut, mata, hidung, tangan, kaki, dan lain-lain. Atribut digambarkan dengan suatu lingkaran dengan nama atribut ditulis di tengahnya seperti gambar II.7.

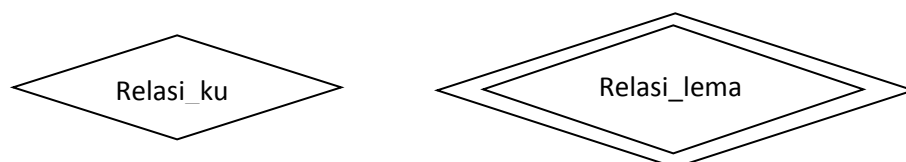


Sumber : Utami (2012:20)

Gambar II.6.
Atribut

3. Relasi

Belah ketupat merupakan penggambaran hubungan (relasi) antarentitas atau sering disebut kerelasi. Ada dua macam penggambaran relasi, yakni relasi kuat dan relasi lemah. Relasi kuat biasanya untuk menghubungkan antarentitas kuat, sedangkan relasi lemah untuk menghubungkan antara entitas kuat dengan entitas lemah, penggambaran kerelasian seperti pada gambar II.8



Sumber : Utami (2012:24)

Gambar II.7.
Relasi

Ada tiga macam relasi menurut derajatnya , yaitu unary , relasi yang menghubungkan satu entitas; binary, relasi yang menghubungkan dua entitas ternary, Relasi yang menghubungkan lebih dari dua entitas. Pada kerelasian binary terdapat kardinalitas atau derajat hubungan antarentitas. Untuk menghubungkan entitas-kerelasian-entitas digunakan garis lurus, seperti pada gambar II.9.



Sumber : Utami (2012:24)

Gambar II.8.
Kerelasian Antar Entitas

4. Kardinalitas (*Cardinality*)

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya, terdapat tiga macam kardinalitas relasi sebagai berikut:

a. *One to One*

Tingkatan hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

b. *One to Many dan Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat.

One to Many (Satu ke Banyak)

Yang berarti satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B.

Many to One (Banyak ke Satu)

Yang berarti setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, dimana setiap tupel pada entitas A berhubung dengan paling banyak satu tupel pada entitas B.

c. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

C. *Logical Record Structure (LRS)*

“LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas, menentukan jumlah tabel dan *Foreign Key (FK)*”. LRS dibentuk dengan nomor dari tipe *record*, beberapa tipe *record* digambarkan dengan kotak persegi panjang dengan nama yang unik, LRS juga terdiri dari hubungan diantara tipe *record*.

Derajat relasi atau kardinalitas rasio, menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya.

D. Pengujian Web

Menurut Simarmata (2010:316) “Pengujian adalah sebuah proses terhadap aplikasi atau 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”.

Pengujian merupakan proses eksekusi program yang telah selesai dibuat yang bertujuan untuk menemukan kesalahan. Pengujian yang baik adalah pengujian yang dilakukan dengan *probabilitas* penemuan kesalahan yang tidak diduga, sedangkan pengujian yang sukses adalah pengujian yang berhasil mengatasi penyelesaian penemuan kesalahan yang tidak diduga.

Menurut Roger (2002:551) black box testing adalah “Pengujian black box berfokus pada persyaratan fungsional perangkat lunak, dengan demikian pengujian black box memungkinkan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program”.

Klasifikasi *black box testing* mencakup beberapa pengujian, yaitu:

1. Pengujian Tegangan (*Stress Testing*)

Pengujian Tegangan berkaitan dengan kualitas aplikasi didalam lingkungan. Idennya adalah untuk menciptakan sebuah lingkungan yang lebih menuntut aplikasi, tidak seperti saat aplikasi dijalankan pada beban kerja normal. Pengujian ini adalah hal yang paling sulit, cukup kompleks dilakukan, dan memerlukan upaya bersama dari semua tim.

2. Pengujian Skenario (*Scenario Testing*)

Pengujian Skenario adalah pengujian yang *realistis*, *kredibel* dan memotivasi *stakeholder*, tantangan untuk program dan mempermudah

penguji untuk melakukan *evaluasi*. Pengujian ini menyediakan kombinasi *variabel-variabel* dan fungsi yang sangat berarti dari pada kombinasi buatan yang anda dapatkan dengan pengujian domain atau desain pengujian kombinasi.

3. Pengujian Regresi (*Regression Testing*)

Pengujian Regresi adalah gaya pengujian yang berfokus pada pengujian ulang (*Retesting*) setelah ada perubahan. Pada pengujian regresi berorientasi risiko (*Risk – Oriented Regression Testing*), daerah yang sama yang sudah diuji, akan kita uji lagi dengan pengujian yang berbeda.

4. Pengujian Mutasi (*Mutation Testing*)

Pada pengujian ini, aplikasi diuji untuk kode yang telah dimodifikasi setelah pemasangan *bug* atau cacat tertentu. Hal ini juga membantu dalam menemukan kode dan strategi pengkodean yang dapat membantu dalam mengembangkan fungsi secara efektif.

5. *Blackbox Testing*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

- a. Kelebihan Blackbox adalah dapat memilih subset test secara efektif dan efisien, dapat menemukan cacat, dan memaksimalkan testing invesmen.
- b. Kekurangan Blackbox adalah tester tidak pernah yakin dengan PL tersebut benar-bener lulus uji.