

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Pada pembuatan suatu *web*, perlu dipelajari dan dipahami dahulu teori-teori pendukung yang berhubungan erat dengan *web* yang akan dirancang. Penulis menggunakan beberapa landasan teori pendukung yang dapat digunakan sebagai dasar dalam pembuatan rancangan *web*. Adapun beberapa teori yang digunakan dalam pembuatan tugas akhir ini adalah:

A. Website

Perkembangan dunia internet telah melahirkan suatu fasilitas layanan baru, yaitu *web*, yang merupakan layanan terpenting dalam internet. Dewasa ini, fasilitas *web* mengizinkan pengakses untuk mengakses dan berinteraksi dengan teks, grafik, animasi, foto, suara, dan video. *Web* secara fisik adalah kumpulan komputer pribadi, *web browser*, koneksi ke ISP, komputer server, *router*, dan *switch* yang digunakan untuk mengalirkan informasi dan menjadi wahana pertama berbagai pihak terkait untuk saling berinteraksi satu sama lain baik dalam pertukaran suatu data maupun informasi penting lainnya.

Menurut Soetejo (2012:11), memberikan pengertian *website* sebagai berikut: Situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi, gambar gerak, suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan link-link.



Menurut Soetejo (2012:1) Mengemukakan bahwa “Internet (kependekan dari *interconnection-network-ing*) ialah sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *Internet Protocol Suite* (TCP/IP) untuk melayani miliyaran pengguna di seluruh dunia”. Manakala Internet (huruf “I” besar) ialah sistem komputer umum, yang berhubung secara global dan menggunakan TCP/IP sebagai protokol pertukaran paket (*packet switching communication protocol*). Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internetworking*.

Berikut adalah istilah-istilah dalam dunia Internet antara lain:

(Kustiyahningsih dan Anamisa: 2011:7)

a. URL (*Universal Resource Locator*)

Konsep nama *file* standar yang diperluas dengan jaringan untuk menentukan lokasi informasi pada *web server*.

b. HTTP (*Hypertext Transfer Protocol*)

Suatu protokol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan menyediakan dokumen yang diminta oleh *browser*.

c. *Web Browser*

Software yang digunakan untuk menampilkan informasi dari *server web*.

d. *Web Server*

Komputer yang digunakan untuk menyimpan dokumen-dokumen *web*, komputer ini akan melayani permintaan dokumen *web* dari kliennya.



e. WWW (*World Wide Web*)

WWW (*World Wide Web*) adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung dengan internet.

Menurut Raymon McLeod,jr., George P.Schel (2007:78) “*Word Wide Web* yang disebut pula sebagai *Web* dan WWW adalah informasi yang dapat diakses melalui internet dimana dokumen-dokumen *hypermedia* (file-file komputer) disimpan dan kemudian diambil dengan cara-cara yang menggunakan metode penentuan alamat yang unik”. Sehingga *web* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*, pemakai dituntut menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen *web* yang ditampilkan dalam *web browser*.

Situs atau *web* dapat dikategorikan menjadi dua yaitu “*web statis*” dan “*web dinamis*”. *Web statis* adalah *web* yang menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan *web* tersebut. *Web dinamis* adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Dengan demikian untuk mengetahui suatu *web* tersebut bersifat statis atau dinamis dapat dilihat dari tampilannya. Jika suatu *web* hanya terhubung dengan halaman *web* lain dan berisi suatu informasi yang tetap maka *web* tersebut disebut statis.

Sistem *web* sebenarnya merupakan aplikasi yang berarsitektur *client-server* dengan menggunakan protokol HTTP dalam komunikasi antara *client* dan *server* yang merupakan pengembangan dari arsitektur file server. Arsitektur ini adalah model konektivitas pada jaringan yang mengenal adanya *server* dan *client*, dimana

masing-masing memiliki fungsi yang berbeda satu sama lainnya. Kunci utama dalam membangun aplikasi *web* adalah meletakkan sesuai dengan proposisinya. Artinya *web developer* harus pintar dalam memilih-milih mana yang harus diproses di *server* dan mana yang di *client*.

Server merupakan komputer yang menjadi pusat pemrosesan perintah-perintah *web* yang dikirim oleh *client*. Perintah tersebut bisa berupa koneksi ke *database*, perintah untuk menampilkan suatu halaman *web* dan lain-lain. *Client* merupakan komputer yang hanya bertugas untuk menampilkan data dari *server* dan mengirim data ke *server*.

Web browser adalah sebuah program aplikasi yang dipergunakan untuk menjelajahi dunia maya atau internet. Aplikasi ini mempunyai kemampuan untuk menampilkan *web page* yang ditulis dalam format dokumen *HTML*. *Linx* dan *Mosaic* merupakan dua contoh program aplikasi *web browser* yang pertama. *Linx* adalah *web browser* yang berbasis *text*. Kekurangan dari *Linx* yaitu tidak dapat menampilkan gambar, namun karena itu pola *web browser* ini menjadi lebih cepat. Sedangkan *Mosaic* adalah *web browser* yang selain berbasis *text* juga sudah dapat menampilkan gambar.



Web server adalah sistem untuk menjalankan situs *Web* yang menempatkan semua informasi di dalam medianya, yang berupa perangkat lunak dan perangkat keras, untuk menjalankan situs *Web* yang terkandung didalamnya. Dalam Internet yang menjadi *server Web* adalah *providernya*.

B. Bahasa Pemograman

Seiring dengan perkembangan teknologi maka lahirlah PHP sebagai bahasa pemrograman open source yang digunakan secara luas terutama untuk pengembangan *web* dan dapat disimpan dalam bentuk HTML. Sehingga *web* tidak hanya memberikan informasi tetapi terjalin interaksi dan menjadikan *web* bersifat dinamis dan diintegrasikan dengan *web server* Apache, PWS dan IIS.

Kelahiran PHP bermula saat Rasmus Ledorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yaitu tahun 1994. Tahun 1995, Rasmus menciptakan PHP/FI versi 2, dimana versi tersebut dapat menempelkan kode terstruktur dalam tag HTML, dan juga PHP dapat berkomunikasi dengan *database*.

Menurut Angga (2007:2) “PHP (*Personal Home Page Tools*) adalah bahasa *scripting server-side* bagi pemrograman *web*”. Secara sederhana, PHP merupakan tool bagi pengembangan *web* dinamis. PHP sangat populer karena memiliki fungsi built-in lengkap, cepat, mudah dipelajari dan bersifat gratis. PHP dapat berjalan di berbagai *web server* dan sistem operasi yang berbeda. PHP sendiri merupakan bahasa turunan dari bahasa C. Sebagian besar perintahnya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP.

C. Basis Data

Menurut sadeli (2011:8) “*MySQL* adalah *database* yang menghubungkan *script PHP* menggunakan perintah *Query* dan *escaps character* yang sama dengan *PHP*.”

MySQL termasuk DBMS yang multithread, multi-user yang bersifat gratis di bawah lisensi GNU *general public licence* (GPL). Tidak seperti Apache yang merupakan *software* yang di kembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. *MySQL* dimiliki dan disponsori oleh perusahaan swedia, yaitu *MySQL AB*.

MySQL AB memegang hak cipta kode sumbernya. Kedua orang swedia dan satu orang finlandia yang mendirikan *MySQL AB* yaitu David Axmark, Allan Larson, dan Michael Monty Widenius. *MySQL* bersifat gratis atau *open source* sehingga bisa digunakan secara gratis. Pemrograman PHP juga sangat mendukung (*support*) dengan *database MySQL* sehingga apabila dipelajari dengan sungguh-sungguh dapat mengaplikasikan *PHP & MySQL* dalam membuat aplikasi *website*. Salah satu jenis *database* dan yang digunakan penulis pada karya ilmiah tugas akhir ini adalah *MySQL*.



Ada beberapa alasan mengapa *MySQL* menjadi *database* yang sangat populer dan digunakan oleh banyak orang, diantaranya ialah:

1. *MySQL* dapat berjalan dengan stabil pada berbagai sistem operasi, seperti windows, linux, FreeBSD, max os x server, solaris, dan masih banyak lainnya.
2. Bersifat *open source*, *MySQL* didistribusikan secara *open source* (gratis), di bawah lisensi GNU *general public licence* (GPL).
3. Bersifat *Multi user*. *MySQL* dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami suatu masalah.
4. *MySQL* memiliki kecepatan yang baik dalam menangani *query* (perintah *SQL*).

Dengan kata lain, dapat memproses lebih banyak *SQL* per satuan waktu.

5. Dari segi *security* atau keamanan data, *MySQL* memiliki beberapa lapisan *security*, seperti level *subnet mask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password yang terenkripsi.
6. 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*).
7. Dukungan banyak komunitas, biasanya tergabung dalam sebuah forum untuk saling berdiskusi, membagi informasi tentang *MySQL* misalnya, di forum <http://forums.mysql.com/>.

D. Model Pengembangan Perangkat Lunak

Dalam membuat sebuah sistem informasi diperlukan suatu model yang dapat membantu dan mempermudah proses pengembangan sistem. Dari beberapa model pengembangan sistem yang ada, penulis menggunakan model *waterfall* untuk perancangan sistem informasi berbasis *web* ini.

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

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat

lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Pada tahap ini desain di translasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.



4. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Perubahan perangkat lunak 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, tetapi tidak untuk membuat perangkat lunak baru.

Kesulitan model pengembangan sistem model *waterfall* ini terjadi apabila perubahan spesifikasi terjadi ditengah alur pengembangan, pelanggan sulit mendefinisikan semua spesifikasi di awal alur pengembangan, dan pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan diakhir alur pengembangan. Sedangkan kelebihan dari model *waterfall* ini adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan.

2.2. Teori Pendukung

A. Struktur Navigasi

Struktur navigasi digunakan sebagai penuntun alur sebuah aplikasi multimedia atau dapat pula dianalogikan sebagai diagram alur dalam perancangan bahasa pemrograman. Struktur navigasi berfungsi untuk menggambarkan dengan jelas hubungan dan rantai kerja seluruh elemen yang akan digunakan dalam aplikasi.

Dengan penggambaran struktur navigasi pembuatan sebuah aplikasi dapat sistematis dan mudah. Menurut Binanto (2010:269) struktur navigasi yang sering digunakan terdiri dari 4 (empat) jenis yaitu sebagai berikut:

1. Struktur Navigasi Linier

Pengguna akan melakukan navigasi secara berurutan, dari frame atau byte informasi yang satu ke yang lainnya. Tampilan struktur navigasi linier dapat dilihat pada gambar dibawah ini :





Sumber : Binanto

Gambar II.1 Struktur Navigasi Linier

2. Struktur Navigasi Hirarki

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

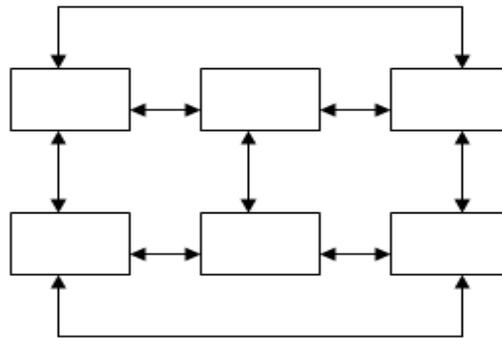


Sumber : Binanto

Gambar II.2 Struktur Navigasi Hirarki

3. Struktur Navigasi Non-linier

Pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat jalur yang sudah ditentukan sebelumnya.

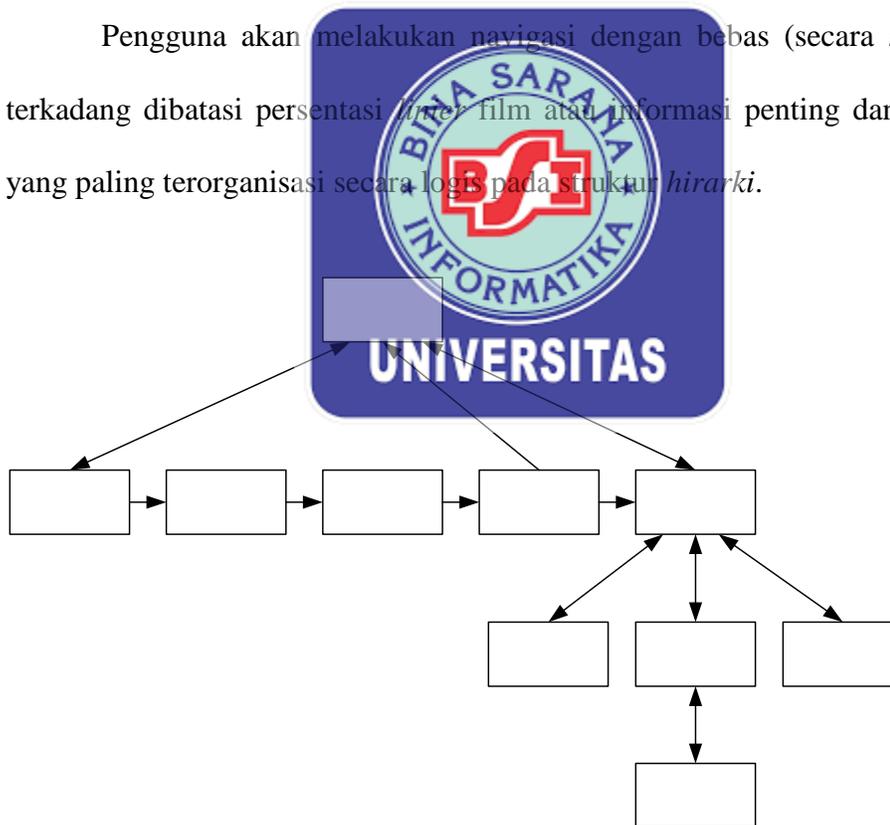


Sumber : Binanto

Gambar II.3 Struktur Navigasi Non-linier

4. Struktur Navigasi Campuran(*composite*).

Pengguna akan melakukan navigasi dengan bebas (secara *non-liner*) tetapi terkadang dibatasi persentasi *linear* film atau informasi penting dan atau pada data yang paling terorganisasi secara logis pada struktur *hirarki*.



Sumber : Binanto

Gambar II.4 Struktur Navigasi Composite.

B. Entity Relationship Diagram (ERD)

Menurut Yasin (2012:276) *Entity Relationship Diagram (ERD)* adalah suatu rancangan atau bentuk hubungan suatu kegiatan didalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relational yang didasarkan atas persepsi di dalam dunia nyata, di dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya.

Adapun komponen-komponen yang terdapat pada *entity relationship diagram* sebagai berikut:

1. Entity

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



2. 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 biasa digambarkan dengan simbol elips.

Adapun macam-macam atribut adalah sebagai berikut :

- a. *Simple Attribute* yaitu atribut yang berdiri atas satu komponen tunggal dalam keberadaannya yang independen dan tidak dapat dibagi lagi ke

komponen yang lebih kecil. *Simple attribute* dikenal juga dengan nama *atomic attribute*.

- b. *Composite attribute* yaitu atribut yang memiliki berbagai komponen dimana semua komponennya memiliki keberadaan yang independen.
- c. *Single value attribute* yaitu sebuah atribut yang mempunyai nilai tunggal induk setiap kejadian.
- d. *Multi value attribute* yaitu sebuah atribut yang mempunyai beberapa nilai untuk setiap kejadian pada sebuah entitas
- e. *Derived attribute* yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus dari satu entitas.

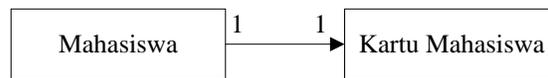
3. Hubungan atau Relasi

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

- a. *One to one* atau satu ke satu (1:1)

Hubungan relasi satu kesatu yaitu setiap entitas pada hubungan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.





Sumber : Yasin (2012:277)

Gambar II.5 Relasi *One to One*

b. *One to many* atau satu ke banyak (1:*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B hanya dapat berhubungan dengan satu entitas pada himpunan entitas A.



Sumber : Yasin (2012:278)

Gambar II.6 Relasi *One to Many*

c. *Many to many* atau banyak ke banyak (*:*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan banyak entitas pada himpunan entitas A.



Sumber : Yasin (2012:278)

Gambar II.7 Relasi *Many to Many*

C. Pengujian *Web*

Black box testing merupakan metode pengujian menggunakan suatu *software* yang dalam hal ini menggunakan internet browser dimana *software tester* tersebut tidak memiliki akses *source code* atau mengetahui implementasi dari program tersebut untuk mencari adanya kesalahan pada program dan juga tidak di haruskan memiliki pengetahuan tentang programming dan implementasinya.

