

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Menurut Sudyanto (2011:8) “situs *web* adalah dokumen-dokumen *web* yang terkumpul menjadi satu kesatuan yang memiliki URL (*Unified Resource Locator*) domain dan biasanya diterbitkan pada *internet* atau *intranet* dan dapat diakses semua pengguna dengan cara mengetikkan alamatnya”. Sedangkan URL adalah alamat-alamat sebuah halaman *web* yaitu halaman suatu dokumen atau program yang ingin ditampilkan atau digunakan. Secara umum URL perlu memasukkan tiga informasi untuk menuju ke alamat tertentu yaitu protokol, alamat *server* dan *path file*. Pada bab ini diuraikan beberapa konsep dasar *web*, diantaranya adalah sebagai berikut :



A. *Website*

Menurut Hidayat (2010:2) menyimpulkan bahwa: *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.

Menurut Kustiyahningsih (2011:113), “*Web* adalah layanan yang didapat oleh pemakai komputer yang terhubung ke internet”. 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 klien sehingga memungkinkan penyajian informasi yang lebih menarik dan dinamis dengan pengelolaan yang terorganisasi.

1. Internet

Menurut Iskandar (2009:1) “Internet (*Inter Connection Networking*) adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia”. Media penghubung tersebut bisa melalui kabel, kanal satelit maupun frekuensi radio. Sehingga komputer-komputer yang terhubung dapat saling berkomunikasi. Setiap komputer yang terhubung dengan jaringan diberikan sebuah nomor yang unik dan berkomunitas satu sama lainnya dengan komunikasi bahasa yang sama. Bahasa komunikasi yang sama ini disebut *protocol*. *protocol* yang digunakan di internet adalah TCP/IP (*Transmission Control /Internet Protocol*).



2. Web browser

Menurut Fauziah (2014:3) “istilah *browser web* digunakan untuk menerjemahkan tag-tag HTML menjadi halaman *web*”. *Browser* berfungsi sebagai penjelajah *web* untuk menampilkan dan melakukann interaksi dengan dokumen-dokumen yang disediakan oleh *web server*. *Browser* yang umum digunakan antara lain *IE (Internet Explorer)* , *Mozilla Firefox*, *Opera*, *Netscape Navigator* dan masih banyak lagi *browser* lainnya.

3. *Web Server*

Menurut Sidik (2009:6) mengemukakan bahwa “*Web server* adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web* dimana komputer ini akan melayani permintaan dokumen *web* dari kliennya”. Untuk itu kita membutuhkan beberapa *server* yang mempunyai dokumen-dokumen media ke *Browser*. *Browser web* seperti, *Netscape*, *Internet Explorer*, *Mosaic*, *Firefox*. Berkomunikasi melalui jaringan ke *Server Web* dengan menggunakan *HTTP (Hypertext Transfer Protokol)*. *Browser* mengirim suatu perintah kepada *server* yaitu meminta dokumen jika ada pada *Protocol HTTP*. *Browser* akan menerima dan mengerti isi dokumen tersebut. *Server Web* juga dapat menjalankan suatu program berdasarkan informasi yang diisi pada *form isian*, seperti menjalankan aplikasi mengakses *database* dan mengirim *e-mail*.

4. *WWW (World Wide Web)*

Menurut Iskandar (2009:4) *WWW* merupakan sebuah sistem dengan informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet *web server* dan dipresentasikan dalam bentuk *hypertext*”.

B. Bahasa Pemrograman

Menurut Hariyanto (2008:72) “Bahasa Pemrograman adalah notasi untuk mengekspresi instruksi-instruksi yang harus dilakukan komputer. Bahasa pemrograman adalah media komunikasi antara manusia dan mesin”.

1. HTML (*Hypertext Markup Language*)

Menurut Alexander F.K. Sibero (2011:19) HTML (*Hypertext Markup Language*) adalah “bahasa yang digunakan pada dokumen *web* untuk pertukaran dokumen *web*”. HTML dalam ilmu komputer merupakan bahasa pemformatan teks untuk dokumen-dokumen pada jaringan komputer yang dikenal sebagai WWW (*World Wide Web*) yang sering disebut *web*. Dokumen-dokumen HTML merupakan berkas *text* yang mengandung dua buah bagian isi, yaitu segala sesuatu yang ingin ditampilkan dan diperlihatkan dalam dokumen *Web* dan *Tag* yang merupakan informasi pemformatan, yang tersembunyi dari pandangan pengguna, yang memberitahu *browser* tentang bagaimana caranya menampilkan isi dokumen ke dalam hadapan pengguna. Beberapa *tag* dalam dokumen-dokumen HTML menentukan bagaimana teks-teks diformat. *Tag-tag* yang lain memberitahu komputer tentang bagaimana menanggapi aksi-aksi yang datang dari pengguna. Sebagai contoh, apa yang harus dilakukan komputer saat pengguna mengklikkan *mouse*-nya pada ikon tertentu. Kemudian *tag* lain yang penting adalah *link* yang mengandung URL (*Uniform Resource Locator*), yang merujuk pada dokumen lain di *server* yang sama atau komputer lain yang ada di jaringan *internet*. *Tag* adalah tanda awal < dan tanda akhir > yang digunakan sebagai pengapit suatu elemen. *Tag* pada elemen pembuka diawali dengan tanda < dan diakhiri dengan tanda >. Sedangkan untuk elemen penutup diawali dengan tanda < dan / kemudian diakhiri dengan tanda >. Untuk penulisan *tag* elemen tunggal cukup menuliskan tanda < dan sebelum tanda > ditambahkan tanda /. Elemen adalah nama penanda yang diapit oleh *tag* yang



memiliki fungsi dan tujuan tertentu pada dokumen HTML. Elemen *head* dapat digunakan sebagai tempat penulisan judul dokumen, informasi mengenai dokumen dan defenisi referensi alamat. Berikut contoh penulisan *tag* yaitu :

1. *Tag Elemen Pembuka*

Bentuk penulisannya adalah : `<head>`

2. *Tag Elemen Penutup*

Bentuk penulisannya adalah : `</head>`

3. *Tag Elemen Tunggal*

Bentuk penulisannya adalah : `<input type="text" />`

Aturan-aturan penulisan pada dokumen HTML antara lain :

- a. Setiap nama *tag* atau elemen pembuka diawali dengan tanda `<` dan diakhiri dengan tanda `>`
- b. Setiap nama *tag* atau elemen penutup diawali dengan tanda `<` dan tanda `/` kemudian diakhiri dengan tanda `>`
- c. Untuk *tag* atau elemen yang berdiri sendiri, cukup dengan menuliskan tanda `<` dan diakhiri *tag* atau elemen ditambahkan tanda `/` sebelum tanda `>`.
- d. Penulisan nama *tag*, elemen atau atribut dapat menggunakan huruf besar maupun huruf kecil (tidak *case sensitive*).
- e. Penulisan nilai pada atribut diawali dengan tanda (`"`) dan di akhiri dengan tanda (`"`).
- f. Urutan struktur dokumen setelah *tag* `<html>` sebaiknya dimulai `<head>`



kemudian `<body>`, jika *tag* `<body>` mendahului *tag* `<head>` secara aturan tidak mengubah atau menyalahi struktur dokumen HTML.

2. PHP (*Perl Hypertext Preprocessor*)

Menurut Sudyanto (2011:43) PHP (*Perl Hypertext Preprocessor*) adalah “bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis”. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dengan format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman *web* lebih terjamin. PHP dirancang untuk membentuk halaman *web* lebih dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini seperti menampilkan isi baris data ke halaman *web*. PHP termasuk dalam *open source product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas. PHP mampu lintas *platform*, artinya PHP dapat berjalan pada *web server* seperti IIS (*Internet Information Server*), Apache, Xitami. PHP juga dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya : Sistem operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dibangun sebagai modul pada *web server* Apache sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP *header*, dapat mengatur *cookies*, mengatur *authentication* dan *redirect users*. Salah satu keunggulan yang dimiliki oleh PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam



software sistem manajemen basis data atau DBMS (*Database Management System*), sehingga dapat menciptakan halaman *web* yang dinamis. PHP memiliki koneksitas yang baik dengan beberapa DBMS antara lain Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, Postgre SQL, Adabas, Filepro, Velocis, dBase, Unix dbm, dan tidak terkecuali semua database ber-*interface* ODBC. PHP juga memiliki integrasi dengan beberapa *library eksternal* yang dapat melakukan segalanya dari dokumen PDF hingga mem-*parse* XML. PHP mendukung komunikasi dengan layanan lain melalui protokol IMAP, SNMP, NNTP, POP3, atau bahkan HTTP. Bila PHP berada dalam halaman *web* anda, maka tidak lagi dibutuhkan pengembangan lingkungan khusus atau direktori khusus. Hampir seluruh aplikasi berbasis *web* yang dibuat dengan PHP. Namun kekuatan utama adalah konektivitas basis data dengan *web*.

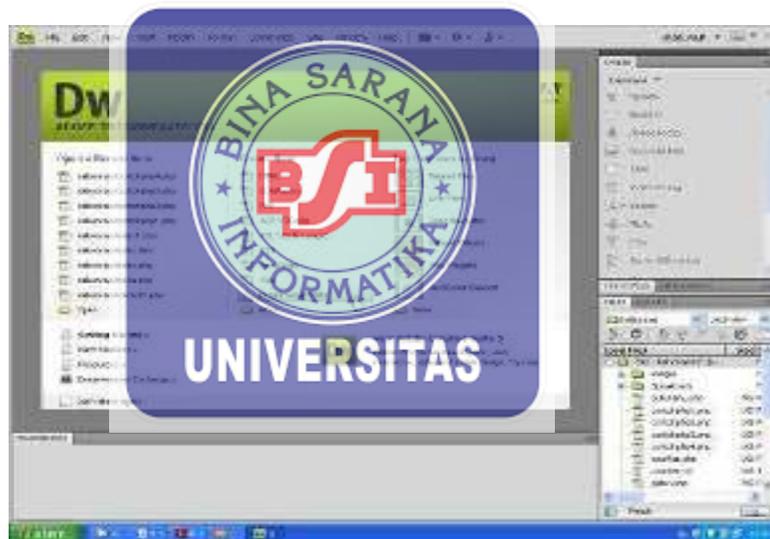
3. Java Script

Menurut Kustiyahningsih (2011:65), java script adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengeksekusian perintah-perintah di sisi *user* artinya di sisi *browser* bukan di sisi *server web*. Java script adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test* dan setiap intruksi diakhiri dengan karakter titik koma (;).”



4. Adobe Dreamweaver CS4

Menurut andi (2009:2), Adobe Dreamweaver CS4 merupakan salah satu aplikasi paling populer yang digunakan untuk membangun *website*. Dreamweaver memberikan fasilitas pengeditan HTML secara *visual*. Aplikasi ini menyertakan berbagai fasilitas dan teknologi pemrograman *web* terkini seperti HTML, CSS, dan Javascript. Selain itu, aplikasi ini juga memungkinkan pengeditan Javascript, XML, dan dokumen teks lainnya secara langsung. Aplikasi ini juga mendukung pemrograman *Script Server Side* seperti PHP, *Active Server Page* (ASP), ASP.NET, *ASP JavaScript*, *ASP VBScript*, *ColdFusion*, dan *Java Server Page* (JSP).



Sumber : andi (2009:7)

Gambar II.1 Tampilan Awal Adobe Dreamweaver CS4

C. Basis Data

Menurut Kusriani (2007:2) “Basis data adalah kumpulan data yang saling berelasi”. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data

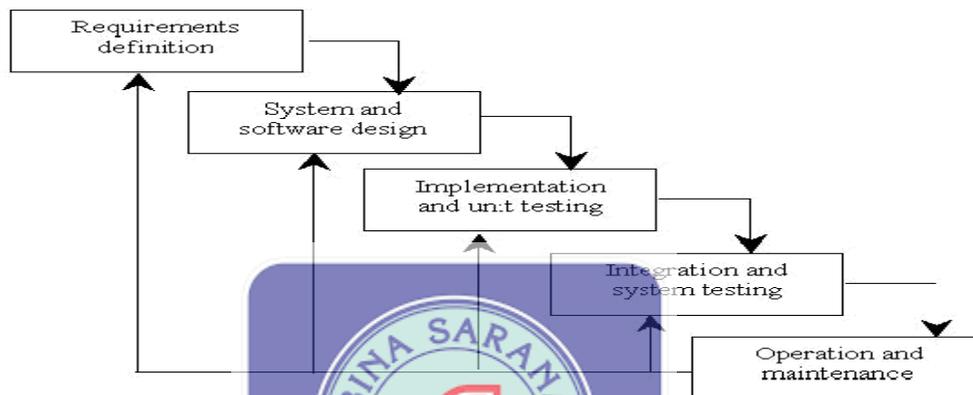
dinyatakan dengan nilai (angka, deretan karakter, atau simbol). Beberapa program *database* yang dapat digunakan untuk pembuatan website antara lain: Oracle, SQL Server, dan MySQL. Disini penulis memilih untuk menggunakan database MySQL.

Menurut Sudyanto (2011:151) “MySQL (*My Structure Query Language*) merupakan *software* yang tergolong dalam *database server* dan bersifat *open source*”. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *structure code*. MySQL termasuk jenis RDBMS (*Relational Database Management System*) yaitu hubungan antar tabel yang berisi data-data pada suatu *database*. *Database* pada MySQL terdiri dari tabel-tabel, maka setiap tabel mempunyai kolom, baris, serta *record* untuk menyimpan data. Tabel-tabel tersebut di *link* oleh suatu relasi yang memungkinkan untuk mengkombinasikan data dari beberapa tabel ketika seorang *user* menginginkan menampilkan informasi dari suatu *database*. Penggunaan MySQL biasanya dipadukan dengan menggunakan program aplikasi PHP pada *web*, tetapi aplikasi *non-web* pun dapat mengakses data dalam *database* di MySQL. MySQL sebagai *database* mempunyai banyak kelebihan. MySQL terkenal dengan pengolahan data yang cepat walaupun data *record* yang dimasukkan dalam jumlah yang banyak. Kelebihan lain dari MySQL adalah MySQL menggunakan bahasa *query* (permintaan) standar SQL (*Structured Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan oleh semua program pengakses *database* seperti Oracle, Progres SQL, SQL server. Sebuah ekspresi SQL terdiri atas 3 klausa, yaitu *select, from* and *where*.



D. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *water fall*. Model *waterwall* merupakan sebuah metode pengembangan *software* yang bersifat sekuensial. Menurut Sommerville (2007:66) model *water fall* yang terbagi menjadi lima tahapan, yaitu :



Sumber: Sommerville (2007:66)

Gambar 11.2 Model *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, rekayasa perangkat lunak (analisis) harus memahami *domain* informasi, tingkah laku, untuk kerja dan antar muka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak di dokumentasikan dan dilihat dengan pelanggan. Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun. Hal ini sangat penting, Mengingat *software* harus dapat berinteraksi

dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. Tahap ini sering disebut dengan *Project Definition*.

2. Perancangan (*Design*)

Desain perangkat lunak sebenarnya adalah proses multi langka yang berfokus pada empat atribut sebuah program yang berbeda ; struktur data, arsitektur perangkat lunak, representasi *interface* dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat atau kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

3. Pembuatan *Coding*

Desain harus diterjemahkan dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer , maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses *coding* atau Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*.

4. Pengujian (*Testing*)

Proses pengujian dilakukan pada logika internal untuk memastikan semua pernyataan sudah teruji. Pengujian eksternal fungsional untuk menemukan



kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

5. Perawatan (*Maintenance*)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*, semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.



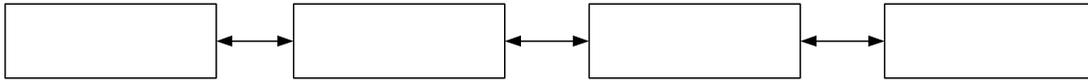
2.2. Teori Pendukung

A. Struktur Navigasi

Menurut Suyanto (2006:62) “struktur navigasi adalah struktur atau alir dari program”. Struktur navigasi merupakan hal yang sebaiknya dilakukan sebelum membuat aplikasi *homepage*. Struktur navigasi dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, keinteraktifitasannya, dan kemudahan yang membuatnya yang berpengaruh terhadap waktu pembuatan suatu situs web. Bentuk dasar dari struktur navigasi adalah sebagai berikut :

1. Struktur Navigasi Linier (*Linear*)

Menurut Suyanto (2006:62) “Struktur navigasi *Linear* (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut”. Dengan kata lain struktur ini hanya dapat menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya dan tidak dapat menampilkan dua halaman sebelumnya atau dua halaman sesudahnya.



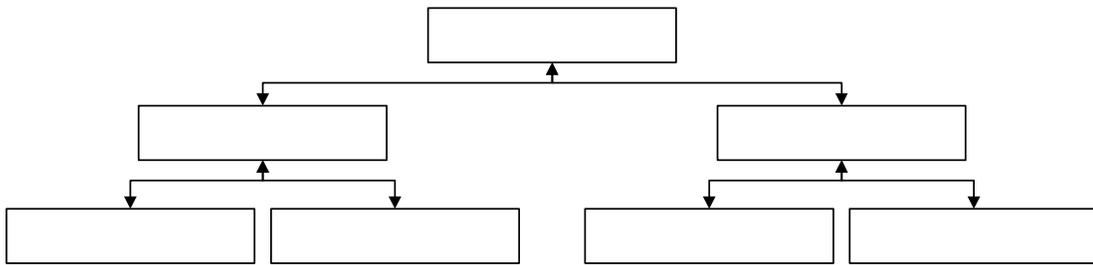
Sumber : Suyanto (2006:62)

Gambar 1.3 Struktur Navigasi Linier (*Linear*)

2. Struktur Navigasi Hirarki (*Hierarchy*)

Menurut Suyanto (2006:62) “Struktur *Hierarchy* (bercabang) ini 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 disebut sebagai *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua) dan seterusnya. Yang terpenting dari struktur penjejukan ini tidak diperkenankan tampilan secara linear.



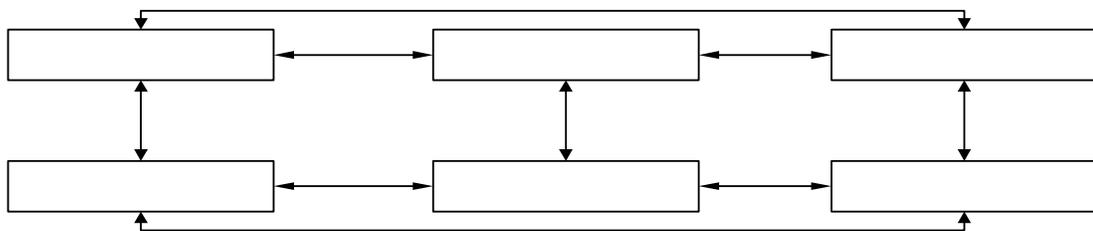


Sumber : Suyanto (2006:62)

Gambar II.4 Struktur Navigasi Hirarki (*Hierarchy*)

3. Struktur Navigasi *Non Linear*

Menurut Suyanto (2006:62) Struktur penjejakan *Non Linear* (tidak berurut) merupakan pengembangan dari struktur penjejakan *Linear*. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute dimana kontrol navigasi dapat mengakses kesemua halamana manapun. Percabangan yang dibuat pada struktur *Non Linear* ini berbeda dengan percabangan pada struktur *Hierarchy*, karena pada percabangan *Non Linear* ini walaupun terdapat percabangan, tetapi tiap-tiap percabangan mempunyai kedudukan yang sama tidak ada *Master Page* dan *Slave Page*.



Sumber : Suyanto (2006:62)

Gambar II.5 Struktur Navigasi Non Linear

4. Struktur Navigasi Campuran

Menurut Suyanto (2006:62) “Struktur navigasi Campuran atau disebut juga struktur penjejak bebas merupakan gabungan dari ketiga struktur sebelumnya yaitu *Linear, Non Linear, Hierarchi*”. Jika suatu tampilan membutuhkan percabangan, maka dapat dibuat percabangan, dan bila percabangan tersebut terdapat suatu tampilan yang sama kedudukannya maka dapat dibuat struktur *Linear* dalam percabangan tersebut. Setiap struktur peta penjejakan seperti yang baru dibahas memiliki fungsi dan tujuan tersendiri, tidak ada yang lebih baik ataupun yang lebih buruk. Penggunaan peta penjejakan bergantung kepada kebutuhan dan tujuan dari *web* yang akan dibuat. Semakin kompleks peta yang digunakan, maka semakin sulit pula pembuatan *page* dari peta penjejakan tersebut.



Sumber : Suyanto (2006:62)

Gambar II.6 Struktur Navigasi Campuran (*Composite*)

B. ERD (*Entity Relationship Diagram*)

Menurut Pratama (2014:49) “ERD (*Entity Relationship Diagram*) adalah diagram yang menggambarkan keterkaitan antartabel beserta dengan *field-field* di dalamnya pada suatu *database* sistem. Sebuah *database* memuat minimal sebuah tabel dengan sebuah atau beberapa buah *field* (kolom) di dalamnya. Namun pada kenyataannya, *database* lebih sering memiliki lebih dari satu buah tabel (dengan beberapa *field* di dalamnya). Setiap tabel umumnya memiliki keterkaitan hubungan. Keterkaitan antartabel ini biasa disebut dengan Relasi”.

Pada dasarnya ada tiga komponen yang digunakan, yaitu :

a. *Entity* (Entitas)

Entitas merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entitas ini biasanya digambarkan dengan persegi panjang.

b. *Atributte* (atribut)

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

c. *Relations* (Hubungan atau Relasi)

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Relasi Satu ke satu (*One to one*)

Hubungan yang menggambarkan bahwa antara 1 anggota *entity* A hanya dapat berhubungan dengan 1 anggota *entity* B.

Biasanya derajat relasi ini digambarkan dengan simbol 1-1.

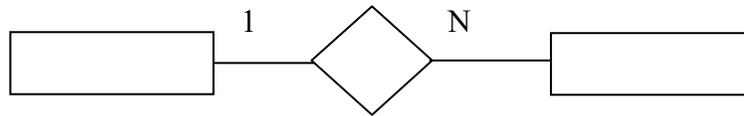


Sumber : Pratama (2014:49)

Gambar II.7 Relasi *One to one*

2) Relasi Satu ke banyak (*One to many*)

Hubungan yang menggambarkan bahwa 1 anggota *entity* A dapat memiliki hubungan dengan lebih dari 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.

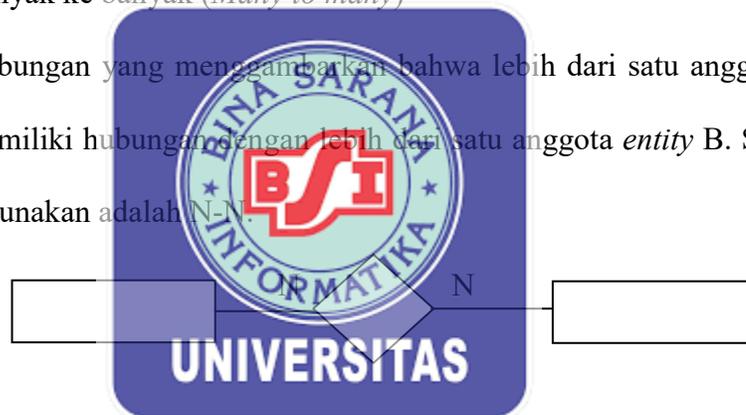


Sumber : Pratama (2014:49)

Gambar II.8 Relasi *One to many*

3) Banyak ke banyak (*Many to many*)

Hubungan yang menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan dengan lebih dari satu anggota *entity* B. Simbol yang digunakan adalah N-N.



Sumber :Pratama (2014:49)

Gambar II.9 Relasi *Many to many*

C. LRS (*Logical Record Structure*)

Menurut Priyadi (2014:40), LRS (*Logical Record Structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tabel dan *Foreign Key* (FK). *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 E-R adalah nama tipe *record* berada diluar kotak *field* tipe *record* ditempatkan. *Logical Record Structure* 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 ER-diagram dan langsung dikonversikan ke LRS.

D. Pengujian Web Menggunakan *Black Box Testing*

Menurut Pressman (2010:482) tujuan dari pengujian adalah “untuk menemukan dan memperbaiki sebanyak mungkin kesalahan dalam program sebelum menyerahkan program kepada *customer*”. Salah satu pengujian yang baik adalah pengujian yang memiliki probabilitas tinggi dalam menemukan kesalahan.

a. Pengujian Kotak Hitam (*Black Box Testing*)

Menurut Pressman (2010,495) *Black-Box testing* “berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program”. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

- 1) Fungsi yang tidak benar atau fungsi yang hilang.

- 2) Kesalahan antarmuka Kesalahan dalam struktur data atau akses *database* eksternal.
- 3) Kesalahan perilaku (*behavior*) atau kesalahan kinerja.
- 4) Inisialisasi dan pemutusan kesalahan.

Tes ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

- 1) Bagaimana validitas fungsional diuji?
- 2) Bagaimana perilaku dan kinerja sistem diuji?
- 3) Apa kelas *input* akan membuat kasus uji yang baik?
- 4) Apakah sistem *sensitive* terhadap nilai input tertentu?
- 5) Bagaimana batas-batas kelas data yang terisolasi?
- 6) Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
- 7) Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi.

b. Jenis Pengujian



Menurut Pratama (2014:51), Terdapat setidaknya empat buah jenis pengujian di sisi pengembang (*blackbox*) ini. Keempat jenis pengujian tersebut meliputi:

- 1) Pengujian *Interface* (tatap muka) aplikasi.

Pengujian *Interface* (tatap muka) aplikasi sistem informasi bertujuan untuk mengetahui fungsionalitas dari setiap elemen *interface* yang ada di setiap halaman pada aplikasi sistem informasi. Elemen ini berupa tombol (*button*) yang menjalankan aksi sesuai yang diharapkan oleh pengguna dan pengembang.

2) Pengujian fungsi dasar sistem.

Pengujian fungsi dasar sistem bertujuan untuk mengetahui sejauh mana kinerja dari setiap fungsi dasar sistem yang ada di dalam aplikasi sistem informasi. Fungsi-fungsi ini dalam penerapannya membentuk satu atau sejumlah modul. Modul ini dapat digunakan baik di sisi pengembang maupun sebagai pengguna (misal: instalasi modul melalui akun administrator).

3) Pengujian *form handle* sistem.

Pengujian *form handle* sistem bertujuan untuk mengetahui seperti apa dan sejauh mana respon oleh sistem informasi terhadap inputan yang diberikan oleh pengguna. Inputan yang diberikan oleh pengguna ke dalam sistem informasi dapat berupa inputan bernilai (misalkan: data) maupun inputan kosong.



4) Pengujian keamanan sistem.

Pengujian keamanan sistem bertujuan untuk mengetahui sejauh mana tingkat keamanan yang dimiliki oleh sistem informasi untuk dapat memberikan kenyamanan kepada para pengguna. Keamanan dicek dari sisi sistem (misalkan: *SQL Injection*), kebijakan (misalkan: ada tidaknya penanganan minimal jumlah karakter untuk *password*, otentikasi via e-mail), serta *user/pengguna* (misalkan: ada tidaknya perbedaan hak akses untuk setiap kelompok pengguna).