

BAB II

LANDASAN TEORI

2.1. Teori Dasar

2.1.1. Pengertian Sistem Informasi

Menurut Susanto (2013:22) beliau menyimpulkan bahwa: "sistem adalah kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu".

2.1.2. Pengertian Sistem Informasi

Menurut Mulyanto (2009 : 12) mengartikan bahwa: "Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya, sedangkan data merupakan sumber informasi yang menggambarkan suatu kejadian yang nyata".

2.1.3. Pengertian Sistem Informasi

Menurut Mulyanto, (2009:31) menyatakan bahwa :

Sistem informasi terdiri dari lima sumber daya yang dikenal sebagai komponen sistem informasi. Kelima sumber daya tersebut adalah manusia, *hardware*, *software*, data, dan jaringan. Kelima komponen tersebut memainkan peranan yang sangat penting dalam suatu sistem informasi. Namun dalam kenyataannya, tidak semua sistem informasi mencakup kelima komponen tersebut.

Berdasarkan uraian diatas, dapat disimpulkan sistem informasi merupakan sekumpulan komponen yang saling terkait yang berguna untuk mengumpulkan dan menyebarkan informasi dalam sebuah organisasi. Berdasarkan uraian diatas pula dapat disimpulkan bahwa sistem informasi memiliki banyak keuntungan terutama untuk menjalankan proses bisnis dengan baik.

2.1.2. Internet

Menurut Hidayatullah dan Kawistara, (2014:1) mengemukakan bahwa “Internet adalah jaringan global yang menghubungkan komputer di seluruh dunia”.

Menurut Hidayat, (2010:1) mengemukakan bahwa “Internet seolah menjelma menjadi bagian hidup masyarakat modern saat ini. Internet merupakan sumber informasi, berita, ilmu pengetahuan serta memungkinkan ketidak adanya batasan antar ruang dan waktu dalam berkomunikasi dengan berbagai orang di berbagai belahan dunia”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa Internet merupakan sebuah jaringan komputer global yang dapat mengabungkan jaringan pengguna di seluruh dunia untuk memberikan informasi yang diinginkan.

2.1.3. Website

Menurut Hidayat, (2010:2) menyimpulkan “*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”. Adapun jenis-jenis *web*:

1. Jenis – jenis *web* berdasarkan sifat atau *style*:

- a. *Website* Dinamis, merupakan sebuah *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan *database* MySQL atau MS SQL.

- b. *Website Statis*, merupakan *website* yang kontennya sangat jarang berubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*.

2. Berdasarkan pada fungsinya, *website* terbagi atas:

- a. *Personal website*, *website-website* yang berisi informasi pribadi seseorang.
- b. *Commercial website*, *website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
- c. *Government website*, *website* yang dimiliki oleh instansi pemerintah, pendidikan, yang bertujuan memberikan pelayanan kepada pengguna.
- d. *Non-profit Organization website*, dimiliki oleh organisasi yang bersifat non-profit atau tidak bersifat bisnis.

3. Ditinjau dari segi bahasa pemrograman yang digunakan, *website* terbagi atas:

- a. *Server Side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya *server*. Seperti PHP, ASP, .NET dan lain sebagainya. Jika tidak ada *server*, *website* yang dibangun menggunakan bahasa pemrograman di atas tidak akan berfungsi sebagaimana mestinya.
- b. *Client Side*, adalah *website* yang tidak membutuhkan *server* dalam menjalankannya, cukup diakses melalui *browser* saja. Misalnya, HTML.

Sedangkan Menurut Arief (2011:8) menyimpulkan bahwa :

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

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

Dari teori – teori diatas dapat disimpulkan pengertian *website* adalah halaman – halaman yang memiliki domain/URL (*Uniform Resource Locator*) berisi text, gambar, animasi, suara dan multimedia yang dapat saling terhubung melalui jaringan internet.

2.1.4. HTTP

Menurut Hidayatullah dan Kawistara (2014:5), *Hypertext Transfer Protocol* (HTTP) adalah protokol agar *client* dan *server* bisa berkomunikasi dengan gaya *request-response*. HTTP menentukan bagaimana format pesan dan bagaimana cara pengirimannya, serta bagaimana *web server* dan *browser* beraksi dan bereaksi terhadap berbagai perintah.

Dapat disimpulkan HTTP adalah protokol agar *client* dan *server* bisa berkomunikasi dengan gaya *request-response*.

2.1.5. PHP

PHP *Hypertext Preprocessor* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML (Hidayatullah dan Kawistara. 2014:53).

Berdasarkan definisi diatas, dapat disimpulkan PHP adalah bahasa pemrograman yang berfungsi untuk membantu membuat rancangan suatu tampilan *website* yang dinamis dan dapat digabungkan dalam *tag* HTML.

2.1.6. HTML

Ardhana (2012:42) “HTML merupakan suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video”.

Anhar (2010:40) “HTML adalah sekumpulan simbol-simbol atau *tag* - *tag* yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. *Tag* - *tag* HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana *x tag* HTML itu seperti *b*, *i*, *u* dll”.

Dapat disimpulkan HTML adalah sekumpulan simbol – simbol atau *tag* – *tag* yang dituliskan dalam sebuah *file* yang gunanya menampilkan informasi seperti teks, gambar, suara, animasi bahkan video.

2.1.7. CSS

Menurut Rozi & SmitDev Community (2015:69), CSS singkatan dari *Cascading Style Sheets*. CSS adalah bahasa pengkodean yang digunakan untuk menata gaya tampilan halaman *web* agar lebih cantik dan indah saat ditampilkan di *web browser*.

Cascading Style Sheets sudah didukung oleh hampir semua *web browser* karena CSS telah distandarkan oleh *World Wide Web Consortium* (W3C).

Menurut Hidayatullah dan Kawistara (2014:54) Ada 4 cara memasang kode CSS ke dalam kode HTML / halaman *web* yaitu:

1. *Inline style sheet* (Memasukan kode CSS langsung pada *tag* HTML)
2. *Internal style sheet* (Embed atau memasang kode CSS kedalam bagian `<head>`)
3. Me-link ke *external* CSS
4. *Import File* CSS.

2.1.8. Javascript

Menurut Rozi & SmitDev Community (2015: 132) didalam bukunya menyimpulkan bahwa:

Javascript adalah *script default* di HTML. Meskipun namanya berbau *Java*, Javascript dan *Java* adalah dua bahasa pemrograman yang berbeda. Bersama HTML dan CSS, Javascript menjadi tiga teknologi paling utama dalam pembuatan konten *website*. Javascript sudah didukung oleh berbagai *browser* modern, sehingga tidak membutuhkan *plugin* apapun. Melalui Javascript, anda bisa mengubah konten halaman *web*, mengubah atribut *tag* HTML, mengubah aturan *style* di CSS, atau memvalidasi *form* yang diinput pengunjung sebelum mengirimkannya ke *web server*.

Dapat disimpulkan Javascript adalah *script default* di HTML Melalui JavaScript, anda bisa mengubah konten halaman *web*, mengubah atribut *tag* HTML, mengubah aturan *style* di CSS, atau memvalidasi *form* yang diinput pengunjung sebelum mengirimkannya ke *web server*.

2.1.9. JQuery

Menurut Rozi dan SmitDev Community (2015: 132) “JQuery adalah *Library* Javascript berisi sekumpulan *script* siap pakai sehingga tidak perlu menyusun *script* program sendiri”.

Menurut Hidayatullah dan Kawistara (2014:426) “JQuery adalah kumpulan fungsi-fungsi Javascript yang sudah dibentuk sebagai suatu objek. Sehingga penggunaan JQuery ini bisa dikategorikan sebagai suatu *library* yang nantinya kita hanya perlu menggunakan fungsi-fungsi di dalam *library* tersebut”.

Sehingga dapat disimpulkan JQuery adalah *Library* Javascript berisi sekumpulan *script* siap pakai sehingga kita hanya perlu menggunakan fungsi-fungsi didalam *library* tersebut.

2.1.10. Adobe Dreamweaver CS5

Menurut MADCOMS (2010:1) "Dreamweaver merupakan *software* aplikasi yang digunakan sebagai HTML editor professional untuk mendesain *web* secara visual. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*), yang intinya adalah bahwa Anda tidak harus berurusan dengan *tag-tag* HTML untuk membuat situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada Anda untuk menggunakannya sebagai media penulisan bahasa pemrograman *web*".

Dengan kemampuan fasilitas yang optimal dalam jendela *design* membuat program ini memberikan kemudahan untuk mendesain *web* meskipun untuk para *web* desainer pemula sekalipun. Sedangkan kemampuan Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, Javascript dan lainnya juga memberikan fasilitas maksimal kepada para desainer *web* yang menyertakan bahasa pemrograman *web* di dalamnya.

Adobe kembali mengeluarkan varian terbaru dari Dreamweaver yaitu Dreamweaver CS5. Ada beberapa fitur baru yang dapat ditemukan pada versi terbaru Dreamweaver CS5 ini, diantaranya :

1. Integrasi dengan *Adobe Bussines Catalyst*.
2. Integrasi *Adobe Browser Lab*.
3. Integrasi dengan CMS.
4. Pengecekan CSS dan lain-lain.

Selain itu juga ada beberapa fitur dalam versi Dreamweaver CS4 yang tidak disertakan atau ditemukan kembali dalam Dreamweaver CS5, diantaranya:

1. Fasilitas pembuatan *web* album foto.

2. Penambahan *Flash Paper*.

3. Beberapa *behaviour javascript*, dan lain-lain.

2.1.11. *Web Browser*

Menurut Hidayatullah dan Kawistara (2014:3) mengemukakan bahwa ”*web clients* menampilkan halaman *web* dengan program yang dikenal dengan nama *Web Browser* seperti Chrome, Firefox dan Internet Explorer”. Cara *browser* menampilkan halaman *web*:

1. Semua halaman *web* mengandung instruksi – instruksi tentang bagaimana dia akan ditampilkan.
2. *Browser* menampilkan halaman tersebut dengan cara membaca instruksi – instruksi ini.
3. Instruksi yang paling umum tersebut disebut dengan *tag – tag HTML*.

2.1.12. *Web Server*

Menurut Rozi dan SmitDev Community (2015: 132) *web server* adalah program untuk melayani penyajian halaman *web* di internet. Berbeda dengan HTML dan CSS yang bisa diletakan disembarang folder di komputer lalu dibuka langsung pada *web browser*, PHP dan MySQL hanya dapat bekerja dilingkungan *server*. Pada umumnya, *server – server* di internet sudah mensupport penggunaan PHP dan MySQL. Namun untuk kebutuhan pengembangan dan simulasi di komputer lokal, maka anda harus menginstal paket *software server*.

Menurut Rozi dan SmitDev Community (2015:230) ada beberapa paket *software gratis* untuk keperluan ini. Umumnya nama paket *software*nya mengandung kata AMP (Apache MySQL PHP). Contohnya adalah WAMP (Windows Apache MySQL PHP), LAMP (Linux Apache MySQL PHP), dan

XAMPP (Cross-Platform Apache MySQL PHP Perl). Apache adalah *web server*, MySQL adalah *database server*, PHP dan Perl adalah *Script Pemrograman*. Jika menggunakan komputer Windows, maka bisa digunakan WAMPP atau XAMPP. Jika menggunakan komputer LINUX, bisa menggunakan LAMPP atau XAMPP. Adapun pilihan paling ideal adalah XAMPP.

1. Install XAMPP

Download XAMPP di www.apachefriends.org/download.html. Pilih versi yang sesuai dengan sistem operasi di komputer. Setelah terdownload, jalankan *file installer* XAMPP dan selesaikan *wizard* instalasi XAMPP di komputer. Tempatkan *folder* instalasi secara default di C:\ProgramFiles\Files\xampp\ atau di lokasi lain, misalnya di D:\xampp\ agar diproyek – proyek *web* anda lebih mudah dikelola.

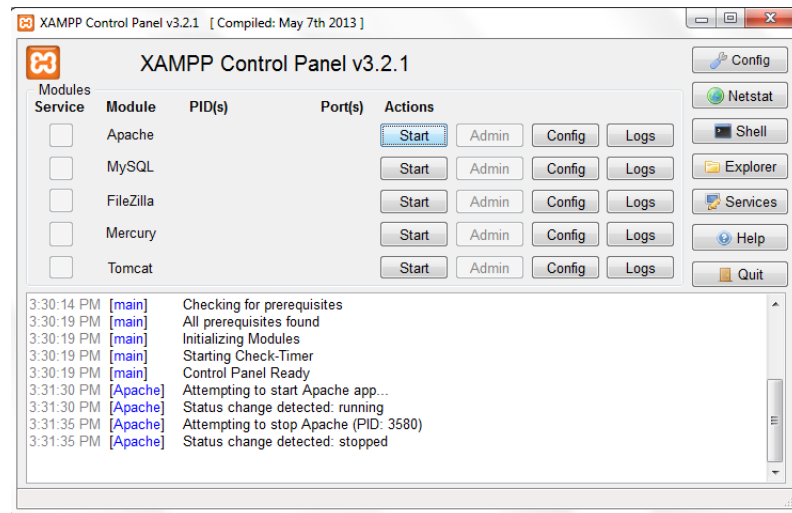


Sumber : XAMPP Rozi dan SmitDev Community (2015:231)

Gambar II.1 Tampilan Install XAMPP

2. Aktifkan *web server* Apache dan *database server* MySQL

Untuk mengaktifkan *server* anda, jalankan *XAMPP Control Panel*. Klik tombol *Start* pada modul Apache dan MySQL.



Sumber : XAMPP Rozi dan SmitDev Community (2015:231)

Gambar II.2 Tampilan XAMPP Control Panel

3. Halaman Default XAMPP

Buka *browser* dan ketikkan <http://localhost>, maka halaman *default* XAMPP akan ditampilkan sebagai tanda *server* lokal telah bekerja dengan baik dan siap digunakan.



Sumber : XAMPP Rozi dan SmitDev Community (2015:232)

Gambar II.3 Tampilan Halaman Default XAMPP

2.1.13. Basis Data

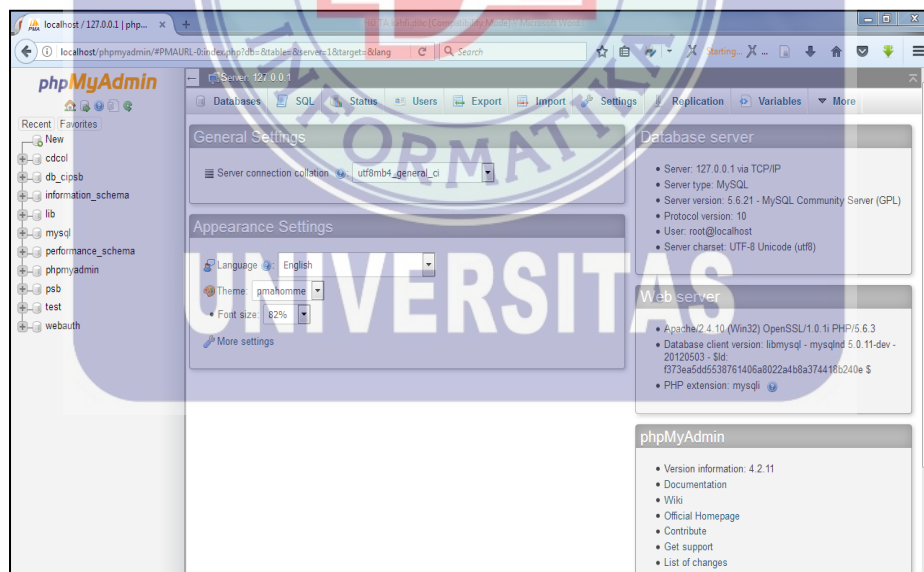
Menurut Rosa dan Shalahuddin (2014:43) Menyimpulkan bahwa “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah

memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

2.1.14. MySQL

Menurut Hidayatullah dan Kawistara (2014:186) MySQL adalah salah satu aplikasi *Database Management System* yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*.

Menurut Rozi dan SmitDev Community (2015:238) untuk membuat database di MySQL, anda dapat memanfaatkan aplikasi bantu yang bernama phpMyAdmin. Aplikasi ini disertakan pada XAMPP dan umunya juga disediakan oleh penyedia *hosting* di *server* internet. Untuk mengakses phpMyAdmin, buka <http://localhost/phpmyadmin>.



Sumber : XAMPP Rozi dan SmitDev Community (2015:232)

Gambar II.4 Tampilan phpMyAdmin

Dari teori – teori diatas dapat disimpulkan bahwa MySQL adalah salah satu aplikasi *Database Management System* yang paling banyak digunakan oleh para pemrograman aplikasi *web*.

2.1.15. Model Pengembangan Perangkat Lunak

Menurut Sukamto dkk, (2013:28) memberikan batasan bahwa “Model SDLC (*Software Develoment Life Cyle*) air terjun (*Waterfall*) sering juga disebut model skuensi linier (*sequential liniar*) 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*). Badalah gambar model air terjun (*waterfall*):



Gambar II.5 Ilustrasi Model Water Fall

1. Analisis Kebuthan Perangkat Lunak.

Proses kebutuhan dilakukan secara insentif mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang di butuhkan oleh *user*. Spesipikasi kebututhan perangkat lunak pada tahapan ini perlu untuk di dokumentasi.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi perangkat lunak, dan prosedur pengkodean. Tahapan ini mentranslasi kebutuhan perangkat lunak dari tahapan analisa kebutuhan ke representasi desain agar dapat di implementasikan menjadi program tahap selanjutnya. Desain perangkat lunak yang di hasilkan pada tahapan ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

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

4. Pengujian

Pengujian focus pada perangkat lunak secara dari segi logik dan *fungsi*ional 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*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika dikirimkan ke *user*. Perubahan bias terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengurangi proses pengembangan mulai dari *analisis spesifikasi* untuk

perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Pendukung

2.2.1. *Entity Relationship Diagram*

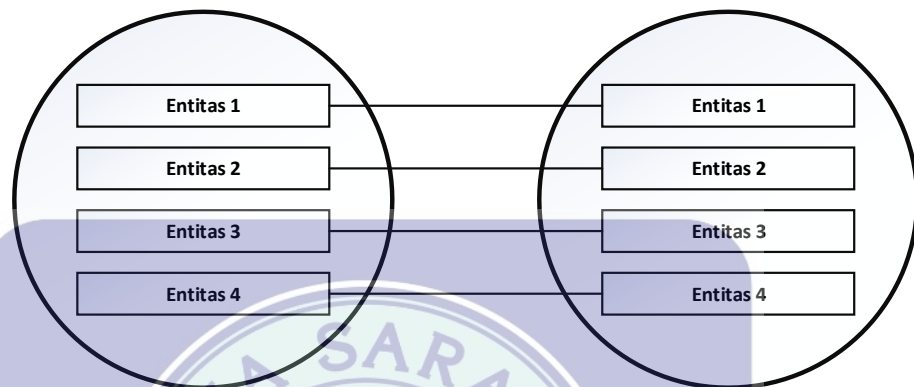
Menurut Fathansyah (2012:81), “*Entity Relationship Diagram* adalah diagram yang berisi komponen-komponen himpunan entitas dan relasi yang masing-masing dilengkapi dengan atribut yang mempersentasikan seluruh fakta dari dunia nyata yang kita tinjau, dapat di gambarkan dengan lebih sistematis dengan mengguakan Diagram *Entity-Relationship* (Diagram-ER)”. Komponen yang sering di gunakan dalam pembuatan Diagram-ER adalah:

- a) Persegi Panjang , menyatakan himpunan entitas.
- b) Lingkaran/Elip, menyatakan Atribut (Atribut yang berfungsi sebagai key digaris bawah)
- c) Belah Ketupat, menyatakan himpunan relasi.
- d) Garis, sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.
- e) Kardinalitas Relasi, dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu ke satu, dan N untuk relasi satu ke banyak atau N dan N untuk relasi banyak ke banyak).

Menurut Fathansyah (2012:78), “Kardinalitas Relasi menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”. Kardinalitas Relasi yang terejadi antara dua himpunan entitas (misalnya A dan B) dapat berupa:

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

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan B, dan begitu juga sebaliknya

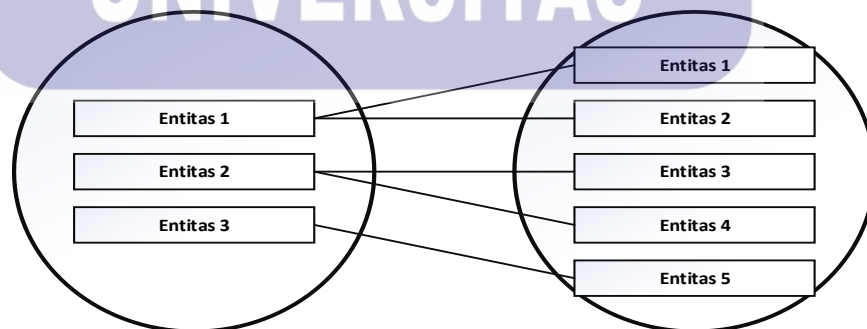


Sumber: Fathansyah (2012:79)

Gambar II.6 Kardinalitas Relasi Satu ke Satu

2) Satu ke Banyak (*One to Many*)

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

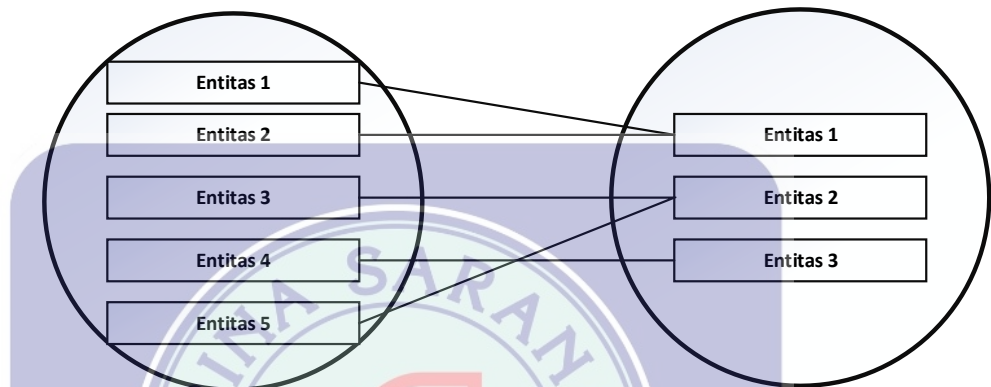


Sumber: Fathansyah (2012:80)

Gambar II.7 Kardinalitas Relasi Satu ke Banyak

3) Banyak ke Satu (*Many to One*)

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

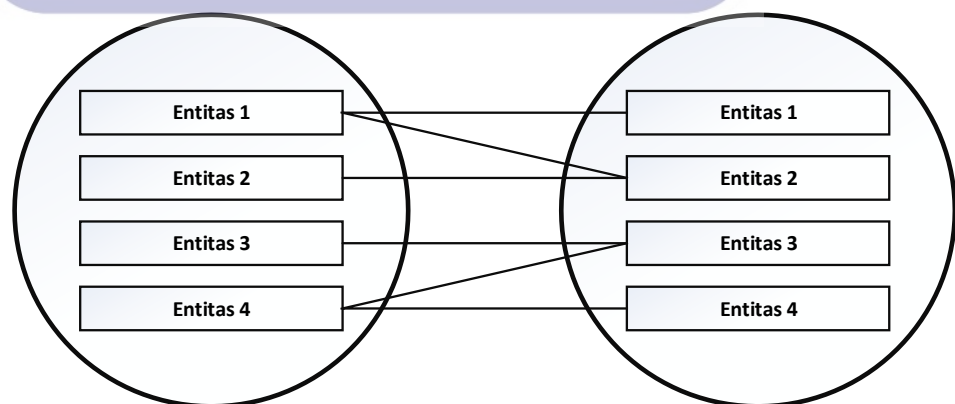


Sumber: Fathansyah (2012:80)

Gambar II.8 Kardinalitas Relasi Banyak ke Satu

4) Banyak ke Banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyaknya entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Sumber: Fathansyah (2012:81)

Gambar II.9 Kardinalitas Relasi Banyak ke banyak

2.2.2. Logical Relationship Structure

Menurut Fathansyah (2012:30), “LRS dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dengan nama yang unik LRS juga terdiri dari hubungan antar *tipe record*. Dua metode yang dapat di konversikan ke LRS. Metode lain dimulai dengan ERD dan langsung dikonversikan ke LRS”.

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* adalah nama tipe *record* berada diluar kotak *field tipe record* ditempatkan. LRS terdiri dari *link-link* diantara tipe *record*. *Link* ini menunjukkan arah dari suatu tipe *record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*.

Penggambaran LRS dimulai dengan menggunakan model yang dimengerti. Dua metode yang 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 mentranformasikan ke bentuk relasi (*table*).
2. Konversi ERD ke LRS sebuah model sistem yang di gambarkan dengan sebuah ERD akan mengikuti pola permodelan tertentu. Dalam kaitannya dengan konversi ke LRS, 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 (*table*) relasi atau *table* adalah bentuk pernyataan data secara grafis 2 (dua) dimensi, yang terdiri kolom dan baris.

Realasi adalah bentuk visual dari sebuah *file*, dan tiap *tuple* dalam sebuah *field* atau yang dalam bentuk lingkaran Diagram *entity relationship* dikenal dengan sebutan atribut.

- a) Nama *logical record structure* menjadi nama relasi.
- b) Tiap atribut menjadi sebuah kolom didalam relasi.

2.2.3. Struktur Navigasi

Menurut Kurniawan (2008:213), “pada pengembangan aplikasi berbasis *web*, tentunya dalam membuat lebih dari satu halaman *web*, bahkan bisa ratusan jika aplikasi yang di bangun kompleks oleh karenanya navigasi adalah fitur yang harus disediakan”. Navigasi berfungsi untuk berpindah dari satu halaman ke halaman yang lain pada suatu aplikasi berbasis *web*, yang juga di gunakan untuk memberikan informasi lokasi halaman yang sedang di buka.

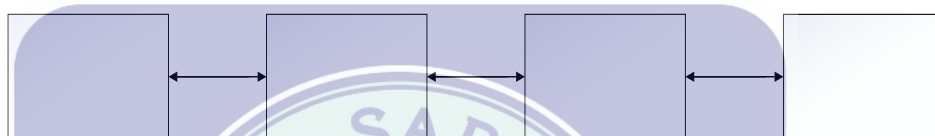
Ada 4 macam bentuk dasar dari peta navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia:

1. Struktur navigasi *linear*

Linier (satu alur) merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurutan. Dengan kata lain struktur ini hanya dapat menampilkan

satu demi satu tampilan layar secara berurut menurut urutan nya. Tampilan yang dapat di tampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya. Salah satu yang terpenting dari struktur ini adalah tidak di perkenankan terjadinya percabangan.

Menurut Binanto (2010:269) mengemukakan bahwa “pengguna akan melakukan navigasi secara berurutan, dari *frame* atau *byte* informasi yang satu ke yang lain”.



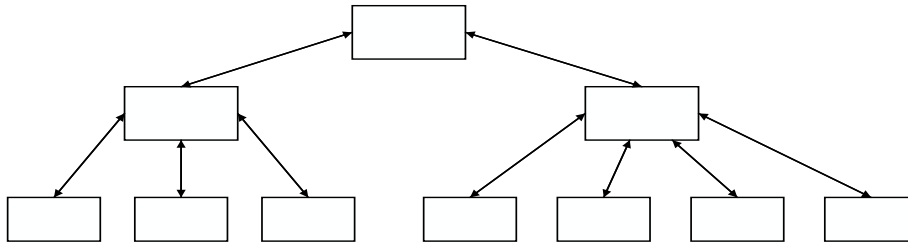
Sumber: Binanto (2010:269)

Gambar II.10 Struktur Navigasi Linear

2. Struktur navigasi *hirarki*

Struktur navigasi hirarki (bercabang) merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan informasi yang berdasarkan kriteria tertentu. Tampilan pada menu pertama akan di sebut *Master Page* (halaman utama kesatu), halaman utama ini akan mempunyai halaman percabangan yang dikatakan *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung di pilih akan di aktifkan maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan *linear*.

Menurut Binanto (2010:269) mengemukakan bahwa “struktur dasar ini di sebut juga struktur *linear* dengan percabangan karena pengguna melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi”.



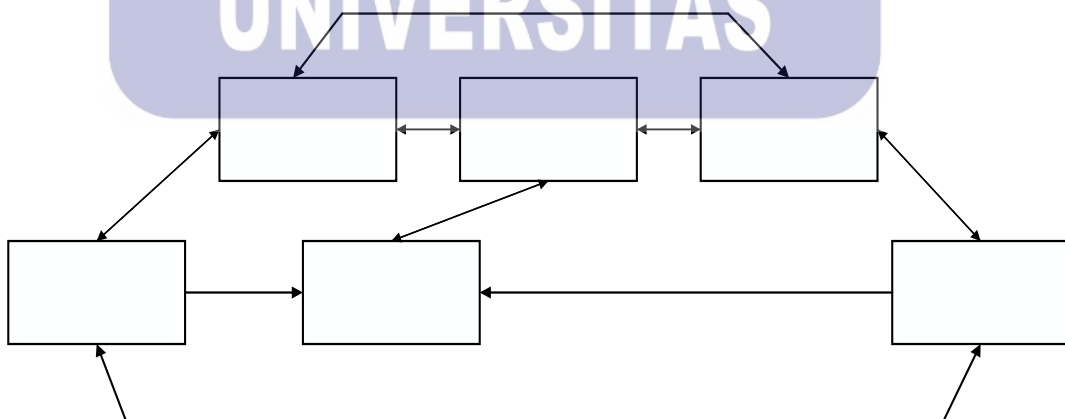
Sumber: Binanto (2010:269)

Gambar II.11 Struktur Navigasi Hirarki

3. Struktur navigasi *non linear*

Struktur navigasi *non linear* (tidak berurutan) merupakan pengembangan dari struktur navigasi *linear*. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur linear ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan *non linear* ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *master page* dan *slave page*.

Menurut Binanto (2010:270) mengemukakan bahwa “pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat dengan jalur yang sudah ditentukan sebelumnya”.



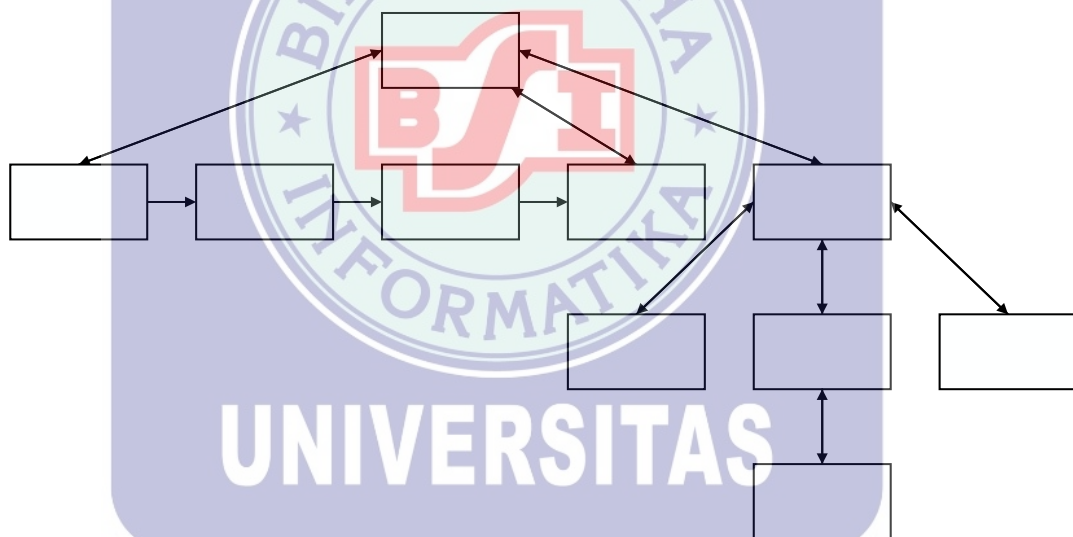
Sumber: Binanto (2010:270)

Gambar II.12 Struktur Navigasi *Non Linear*

4. Struktur Navigasi Campuran

Struktur Navigasi *Composite* (campuran) merupakan struktur gabungan dari ketiga struktur sebelumnya yaitu *Linear*, *Non Linear*, *Hirarki*. Struktur navigasi ini disebut juga struktur navigasi bebas. Kelebihan dengan menggunakan struktur navigasi ini adalah suatu aplikasi mampu memberikan keterkaitan informasinya lebih baik.

Menurut Binanto (2010:270) mengemukakan bahwa “pengguna akan melakukan navigasi dengan bebas (secara *non linear*), tetapi terkadang dibatasi persentasi *linear* film atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu *hirarki*”



Sumber: Binanto (2010:270)

Gambar II.13 Struktur Navigasi Composite

2.2.4. Pengujian Web

Menurut Sukamto dkk, (2013:275) menyatakan bahwa “*Black-Box Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian yang dimaksud untuk

mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibuat”. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus yang benar dan kasus salah, misalnya untuk kasus proses login maka kasus uji yang dibuat adalah:

1. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukan nama pemakai (*username*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya.

