

BAB II

LANDASAN TEORI

2.1. Konsep Dasar *Web*

Semakin canggih teknologi dan ilmu pengetahuan, maka komputer dan internet sudah menjadi infrastruktur untuk membantu menjalankan aktivitas kehidupan manusia di jaman sekarang ini. Perkembangan aplikasi *web* yang semakin pesat sejak munculnya teknologi *internet* sangat membantu dalam kemudahan serta kecepatan pengiriman, penyampaian dan penerimaan informasi. Mulai dari perusahaan-perusahaan, sekolah-sekolah, perguruan tinggi, dan lembaga atau organisasi lainnya telah banyak memanfaatkan aplikasi *web* dalam kegiatan lainnya. *Internet* memiliki banyak keunggulan bagi para konsumen maupun organisasi, misalnya dalam hal kenyamanan akses 24 jam sehari, jangkauan *global*, sumber informasi potensial dan masih banyak lainnya.

2.1.1. *Website*

A. Sejarah *Website*

Dunia internet berkembang dengan sangat pesat seakan-akan telah menjadi bagian hidup masyarakat *modern* ini. Dengan internet seolah-olah tidak ada lagi batasan antar ruang dan waktu dalam berkomunikasi dengan berbagai orang di belahan dunia. Misalnya saja untuk bertransaksi jual beli di jaman sekarang tidak harus bertemu antara pedagang atau pembeli, kita hanya mengakses *website e-commerce*. Penemu *website* adalah Sir Timothy John “Tim” Berners-Lee, sedangkan situs *website* yang tersambung dengan jaringan pertamakali muncul

pada tahun 1991. Maksud dari Tim ketika merancang situs *website* adalah untuk memudahkan tukar menukar dan memperbaharui informasi pada sesama peneliti di tempat ia bekerja.

Pada tanggal 30 april 1993, CERN (tempat dimana Tim bekerja) mengumumkan bahwa WWW dapat digunakan secara gratis oleh publik yang dapat diakses melalui sebuah *software* yang disebut *browser*, seperti internet explorer, mozilla firefox, opra dan lain-lain. Sebuah situs *website* bisa berupa hasil kerja dari perorangan atau individu, atau menunjukkan kepemilikan dari suatu organisasi, perusahaan. Sebuah server website umumnya telah dilengkapi dengan perangkat-perangkat lunak khusus untuk mengenai pengaturan nama ranah, serta mengenai layanan atas protokol HTTP yang disebut sebagai *Server HTTP* seperti *Apache HTTP Server*, atau *InternetInformation Services (IIS)*.

B. Pengertian Website

Menurut Arief (2011:7) memberikan pengertian bahwa “*website* adalah Salah satu aplikasi yang berisikan dokumen-dokumen multi media (*teks*, gambar, suara, animasi, *video*) di dalamnya yang menggunakan protokol HTTP (*Hypertext Transfer Protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*”.

Browser (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi browser yang biasa disebut *web engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan.

2.1.2. Web Server

Menurut Anhar (2010:4) memberikan pengertian bahwa “*Web Server* adalah aplikasi yang berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui *web browser* di mana *webservice* mengirimkan kembali informasi yang diminta tersebut melalui HTTP (*HyperText Transfer Protocol*) untuk ditampilkan ke layar monitor”.

Bisa dikatakan bahwa *web server* ini adalah penyedia informasi yang berbasis *web*. Jika alamat yang dipanggil dalam *web browser* terdapat informasi, maka akan dikirimkan untuk kemudian ditampilkan dalam *web browser*.

2.1.3. Web Browser

Menurut Arief (2011:19) memberikan definisi bahwa “*web browser* merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format HTML”. Bagaimana halaman *web* yang dibuat ditampilkan sangat tergantung pada *web engine* yang digunakan oleh masing-masing *browser*”

Semua jenis *web browser* yang ada saat ini mengikuti standarisasi yang dibuat oleh *World Wide Web consortium* (W3C) yang merupakan badan independen yang mengurus semua hal yang berkaitan dengan *web* di dunia. Berikut adalah contoh *web browser* yang cukup populer beserta *web engine* yang digunakan:

1. *Web Engine Webkit: Safari, Google Chrome*
2. *Web Engine Trident: Microsoft Internet Explorer, Maxthon.*
3. *Web Engine Gecko: Mozilla Firefox.*
4. *Web Engine Presto: Opera.*

2.1.4. *Internet*

A. *Sejarah Internet*

Menurut Simarmata (2010:50) memberikan definisi bahwa *internet* pertama kali digunakan sebagai proyek penelitian yang ditemukan oleh *Advanced Research Project Agency (ARPA) Department of Defense (DOD)* di amerika serikat. Pada dasarnya, *internet* digunakan untuk menghubungkan komputer. Versi yang pertama disebut ARPANET. Pada tahun 1972, ARPA berubah menjadi DARPA dengan tetap mempromosikan proyek ARPANET.

Pengembangan *internet* dengan jenis peralatan yang berbeda, namun bisa saling berhubungan satu sama lain merupakan tantangan yang besar pada saat itu. Pada tahun 1973-1974, peneliti merancang sebuah *Transmission Control Protocol/internet Protocol (TCP/IP)*. Pada awalnya, TCIP/IP dimaksudkan untuk menyediakan dukungan untuk kebutuhan berikut:

1. *Interoperabilitas* antarsistem *heterogen*,
2. Komunikasi *end-to-end* berbagai jaringan yang berbeda,dan
3. Operasi otomatis dan sempurna di dalam menghadapi terjadinya kegagalan hubungan data

Pada saat itu, aplikasi yang digunakan masih sangat sederhana daripada yang digunakan saat ini. Aplikasi yang paling banyak yang digunakan mungkin adalah telnet untuk *loginremote* dan FTP (*File Transfer Protocol*) untuk perpindahan *file* dan *email*.

Pada awal tahun 1980-an, ARPANET di pecah menjadi dua bagian, yaitu MILNET dan ARPANET karena pertimbangan keamanan. Pihak militer berjalan terus dengan MILNET, sedangkan penelitian, pengembangan, dan sektor lain

tetap memakai ARPANET. Pada pertengahan tahun 1980-an, *National Science Foundation* (NSF) di Washington, D.C. mendistribusikan teknologi *internet* kepada beberapa Universitas (Berkeley, MIT, Stanford, dan UCLA). Selanjutnya, *internet* pun mulai menyebar di dunia.

Pada tahun 1990, DOD memutuskan untuk membubarkan ARPANET dan menggantikannya dengan pendukung (*backbone*) NSFNET, bekerja sama dengan agen jaringan lain. Hal inilah yang kemudian menjadi prinsip pendukung jaringan *internet*. Tabel berikut adalah tabel statistik populasi dan penggunaan *Internet* dunia.

B. Pengertian *Internet*

Menurut Dhanta (2008:15) mengemukakan bahwa "*Internet* merupakan suatu jaringan informasi yang tersaji pada sebuah organisasi yang bersifat bebas dan untuk saling bertukar informasi yang ada". *Internet* menyediakan akses untuk layanan telekomunikasi dan sumber daya informasi untuk jutaan pemakainya yang tersebar di seluruh dunia. Layanan *internet* meliputi komunikasi langsung (*email, chat*), diskusi (*email*) dan sumber daya informasi yang terdistribusi (*World Wide Web*).

Secara etimologis, *Internet* berasal dari bahasa Inggris, yaitu *inter* yang berarti *antar* dan *net* yang berarti jaringan. Jadi, dapat diartikan *internet* merupakan hubungan antar jaringan. *Internet* merupakan suatu media informasi yang berjalan dalam suatu komputer. Akan tetapi, tidak semua komputer yang ada bisa berhubungan karena suatu komputer dapat dikatakan sebagai *internet* jika

sudah terhubung dengan suatu jaringan. *Website* dapat digunakan sebagai media penyampaian informasi di *internet*.

2.1.5. WWW (*Word Wide Web*)

Menurut Fathansyah (2012:464) menyatakan bahwa “*Word Wide Web* (WWW atau *Web*) merupakan sistem informasi terdistribusi yang berbasis *hypertext*”. Dokumen-dokumen yang dikelola dalam *web* bisa beraneka jenis (pengolah kata, lembar kerja, tabel basis data, presentasi, *hypertext* dan lain-lain) dan beragam format (.doc, .pdf, .xls, .dbf, .ppt, .htm dan lain-lain). Jenis dokumen yang paling umum adalah dokumen *hypertext* yang dibentuk berdasarkan format HTML (*Hypertext Markup Language*). HTML sendiri sesungguhnya merupakan varian dari SGML (*Standard Generalized Markup Language*). Dokumen HTML umumnya berisi *teks* dengan spesifikasi *font* dan berbagai instruksi format lainnya. *Link* ke dokumen yang lain (baik lokal maupun jarak jauh atau *remote*) dapat dilekatkan sebagai bagian dari *teks*. Begitu juga dengan objek gambar yang dapat pula diacu dan tampilan melalui perintah format yang khusus.

Dengan demikian, *hypertext* yang berhubungan dengan *web* membentuk antar muka penayangan (*browsing*) yang sangat atraktif secara *visual*. Program peramban yang populer digunakan saat ini adalah MS *Internet Explorer*, *Mozilla Firefox* dan *Google Chrome*. Program perambah web juga mendukung berbagai bahasa pemrograman, seperti Java, PHP atau Perl, yang memungkinkan dokumen-dokumen dilengkapi dengan program yang dapat dieksekusi di lokasi pemakai. Dengan begitu, dokumen tidak hanya bersifat pasif, tetapi juga dapat menjadi aktif.

2.1.6. Dreamweaver CS6

Menurut Sadeli (2013:2) memberikan definisi bahwa “*Dreamweaver CS6* merupakan suatu perangkat lunak *web editor* keluaran *Adobe System* yang digunakan untuk membangun dan mendesain suatu *website* dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya”. *Dreamweaver CS6* tergabung dalam paket *Adobe Creative Suite (CS)* yang didalamnya terdapat paket *desain grafis, video editing, dan pengembangan web* aplikasi.

Pada *Dreamweaver CS6* terdapat beberapa fitur baru dari versi sebelumnya seperti *Fluid Grid Layout* yang dapat mengatur multi halaman *web* dengan mudah, *Web Fonts Manager* manajemen yang memungkinkan anda menggunakan *font* yang tersedia di *web server*, *CSS Transition* spesial efek pada *element*, *Phonen Gap* memungkinkan anda untuk membangun aplikasi *smratphone* menjadi lebih mudah dari sebelumnya.

2.1.7. Adobe Photoshop CS6

Menurut Wahana Komputer(2012:2) memberikan definisi bahwa “*Adobe Photoshop CS6* merupakan kumpulan efek-efek yang dapat anda gunakan untuk memberikan tampilan yang berbeda dan menarik pada *image*”. Versi terbaru dari *adobe Photoshop CS6*.

Adobe Photoshop CS6 sebagai “*The Propessional Standart In Desktop Digital Emaging*”. merupakan pengembangan dari *Adobe Photoshop* versi sebelumnya yang dikonsentrasikan agar program aplikasi ini lebih mudah dipakai, lengkap, andal, fleksibel, dan lebih efisien. *Adobe photosop CS6* merupakan program aplikasi pengolah *image* atau gambar *bitmap*.

2.1.8. HTTP (*Hyper Text Transfer Protocol*)

Menurut Simarmata (2010:52) memberikan definisi bahwa “HTTP adalah *protocol* komunikasi *state less* yang berbasiskan pada TCP yang awalnya digunakan untuk mengambil kendali *file-file HTML* dari *server web* ketika dirancang pada tahun 1991”.

Versi 1.1 (ditetapkan di RFC2616) telah mengalami peningkatan versi ini mengijinkan *elient* dan *server* menggunakan banyak *leaders* untuk menyampaikan status informasi dan deskripsi agen pengguna (*user-agent*) dan bertindak sebagai alat yang bersifat *elementer* untuk *autentikasi* pengguna. Versi awal yang diciptakan pada awal tahun 1991, di pacu sebagai *HTPP* version 0.9, adalah *protokol* yang sederhana untuk komunikasi antar *elient* dan *server*. Kebanyakan versi yang sekarang ini ada berasal dari *protocol* HTTP versi 1.1.

2.1.9. PHP (*Hypertext Preprocessor*)

Menurut Arief (2011:43) memberikan definisi bahwa “PHP (*hypertextPreprocessor*) dengan HTML untuk membuat halaman *web* yang dinamis”. Karena PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman *web* lebih terjamin. PHP di rancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data kehalaman *web*.

PHP termasuk dalam *open source product*, sehingga *source code* php dapat diubah dan didistribusikan secara bebas. PHP juga mampu lintas *platform*.

Artinya PHP dapat berjalan di banyak sistem operasi yang beredar saat ini, di antaranya: Sistem Operasi *Microsoft Windows* (semua versi), *Linux*, *Mac Os*, *Solaris*.

Salah satu keunggulan yang dimiliki oleh PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data/*Database Management System* (DBMS), sehingga dapat menciptakan suatu halaman *web* yang dinamis.

2.1.10. HTML (*HyperText Markup Language*)

Menurut Supriansyah (2008:59) mengatakan bahwa “*HyperText Markup Language* (HTML) adalah format *file* standard untuk membuat sebuah *web*”. Format ini akan ditemukan ketika *browsing* di *internet*. Saat melakukan *save* suatu halaman *web* ke dalam *harddisk*, *file-file* tersebut akan tersimpan dalam bentuk *file* htm.

HTML memiliki sintak atau aturan, struktur dalam menuliskan *script* atau kode-kode yang biasa dinamakan tag HTML. Tag HTML berupa tanda-tanda bahasa baku dalam *web* yang menentukan tampilan suatu teks atau tingkatkepentingan dari *teks* tersebut dalam suatu dokumen.

2.1.11. CCS (*Cascading Style Sheet*)

Menurut Saputra (2012:108) memberikan definisi bahwa “CSS (*Cascading Style Sheet*) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”.sama halnya *style* dalam aplikasi pengolahan kata

seperti *Microsoft Word* yang dapat mengatur beberapa *style* , misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *stlye* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file).

Menurut Wiswakarma (2012:2) “CSS (*Cascading Style Sheet*) merupakan salah satu bahasa *styleweb* yang sekarang sedang populer dipakai pada berbagai aplikasi *web*”. CSS di buat untuk membuat *style* pada *web* bervariasi dan mudah digunakan. Seirig dengan pertumbuhan bahasa HTML di *internet*, tampilan yang ditawarkan CSS makin bervariasi dan makin memudahkan para pengembang *web* untuk memperindah tampilan *web*.

2.1.12. *Jquery*

Menurut Sigit (2011:1) memberikan definisi bahwa “*Jquery* adalah *library* atau kumpulan kode *JavaScript* siap pakai”. Keunggulan menggunakan *jquery* dibandingkan dengan *javascript* standar, yaitu menyederhanakan kode oleh *jquery*. *Javascript* sendiri merupakan bahasa *scripting* yang bekerja di sisi *client/browser* sehingga *website* bisa lebih *interaktif*.

Menurut Hakim (2010:3) memberikan definisi bahwa “*Jquery* adalah *library* kumpulan kode/fungsi *JavasRipt* yang siap dipakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*”. Secara standar, apabila kita membuat kode *JavaScript*, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami, nah disinilah peran *Jquery* sebagai *JavaScript library*, dimana kita bisa langsung memanggil fungsi yang terdapat didalam *Library* tersebut.

Kesimpulannya *Jquery* menyederhanakan kode *Javascript*. Hal ini sesuai dengan slogannya yaitu “*Write Less Do More*”.

2.1.13. Database (Basis Data)

Menurut Anhar (2010:45) Memberikan definisi bahwa “*Database* adalah sekumpulan dari *field* atau kolom”. Struktur *file* yang menyusun sebuah *database* terdiri dari data, *record* dan *field*. Data adalah satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu *file database*.

Record adalah data yang isinya merupakan satu kesatuan seperti *NamaUser* dan *Password*. Setiap keterangan yang mencakup *Nama User* dan *Password* dinamakan satu *record*. Setiap *record* diberi nomor urut yang disebut nomor *record* (*Record Number*). *Field* adalah sub bagian dari *record*. Dari contoh isi *record* di atas, maka terdiri dari dua *field*, yaitu : *field NamaUser* dan *Password*.

A. Xampp

Xampp adalah suatu paket yang berisi kumpulan *software* (*Apache*, *MySQL*, *PHP*) yang digunakan untuk dan dapat berjalan pada *platform* yang berbeda.

Menurut Sibero (2013:2). XAMPP dikembangkan oleh *Apache Friends*, sebuah organisasi nirlaba yang bekerja pada proyek-proyek pengembangan sistem.

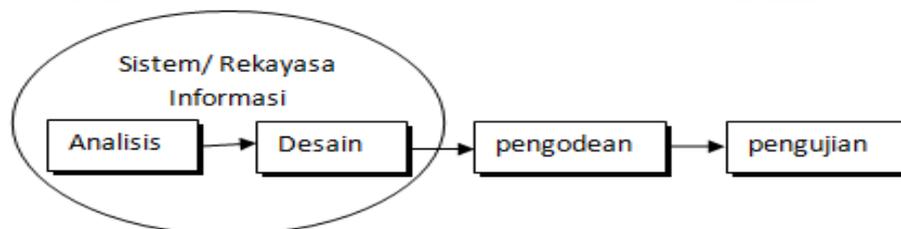
B. MySQL (*My Structure Query Language*)

Menurut Arief (2011:151) memberikan definisi bahwa “*MySQL* adalah salah satu jenis *database server* yang sangat yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya”.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). *MySQL* dan PHP dianggap sebagai pasangan *software* pengembangan aplikasi *web* yang *ideal*. *MySQL* sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP.

2.1.14. Model Pengembangan Perangkat Lunak

Menurut Rosa dan M. Shalahudin (2013:28) model air terjun (*Waterfall*) sering juga disebut model sekuensial *linier* (*Sequential Linier*) atau alur hidup klasik (*classic lifecycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).



Sumber : Rosa dan M. Shalahudin (2013:28)

Gambar II.1.

Waterfall Method

1. Analisis Kebutuhan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspezifikasikan 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 pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

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

4. Pengujian

Pengujian fokus pada perangkat lunak secara 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*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami

perubahanketika 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 adat pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk membuat perangkat lunak baru.

2.2. Teori Pendukung

2.2.1. Struktur Navigasi

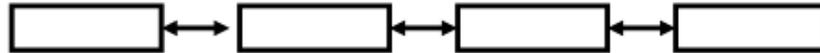
Menurut Simarmata (2010:309) menyatakan bahwa “Struktur Navigasi Adalah alur dari suatu program yang merupakan rancangan hubungan dan rantai kerja dan beberapa area yang berbeda, juga dapat membantu mengorganisasikan seluruh elemen pembuatan *website* dengan memberikan perintah atau pesan”.

Ada empat macam bentuk struktur navigasi linier, struktur navigasi *non linier*, struktur Navigasi Hirarki, dan struktur Navigasi *Composite*.

A. Struktur Navigasi *Linier*

Struktur navigasi *linier* hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan layar secara berurut urutannya.

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



Sumber: Binanto (2010:269)

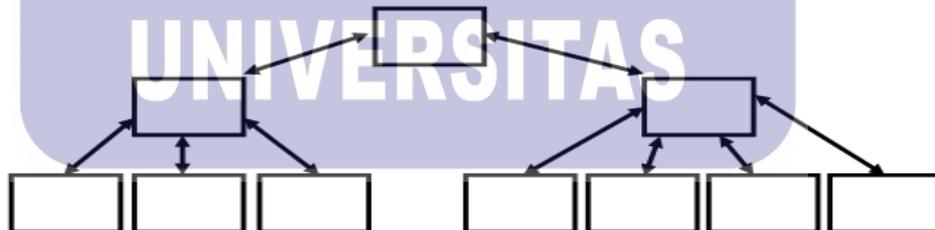
Gambar II.2

Struktur Navigasi Linier

B. *Hierarchi* (Hirarki)

Struktur *Hierarchi* (bercabang) untuk menampilkan data berdasarkan kriteria tertentu. Informasi pada halaman utama disebut *parent* dan informasi pada cabangnya disebut *child*.

Menurut Binanto (2010:269) memberikan pengertian bahwa “struktur dasar ini disebut juga struktur *linier* dengan percabangan karena pengguna navigasi di sepanjang cabang pohon struktur yang terbentuk oleh logika isi”.



Sumber : Binanto (2010:269)

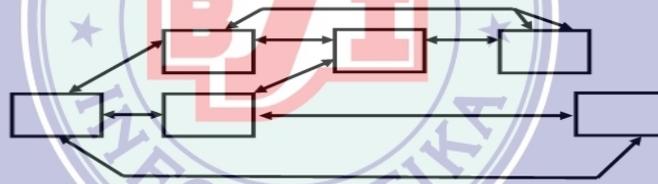
Gambar II.3.

Stuktur Navigasi Hierarki

C. *Non Linear*

Pada struktur ini diperkenankan membuat navigasi bercabang tidak terikat pada jalur yang ditentukan sebelumnya. 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 halaman utama dan halaman pendukung.

Menurut Binanto (2010:269) memberikan pengertian bahwa pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat dengan jalur yang sudah ditentukan sebelumnya.



Sumber: Binanto (2010:270)

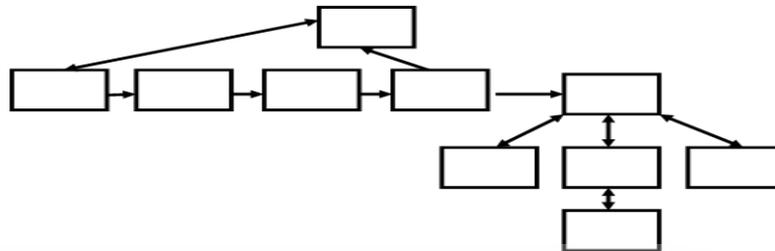
Gambar II.4.
Struktur Navigasi *Non Linear*

D. *Composite*

Struktur Navigasi *Composite* (campuran) merupakan struktur gabungan dari ketiga struktur sebelumnya yaitu *Linier*, *Hirarchi*, dan *Non Liniear*. Struktur ini disebut juga struktur navigasi bebas.

Menurut Binanto (2010:270) memberi pengertian bahwa pengguna akan melakukan navigasi dengan bebas (secara *non-linier*), tetapi terkadang dibatasi

linier film atau informasi penting atau pada data yang paling terorganisasi secara logis pada suatu hirarki.



Sumber: Binanto (2010:270)

Gambar II.5.
Struktur Navigasi Composite

2.2.2. *Entity Relation Diagram* (ERD)

Menurut Fathansyah (2012:81) memberikan definisi bahwa “*EntityRelation Diagram* (ERD) merupakan model *Entity-Relationship* yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari ‘dunia nyata’ yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity-Relationship* (*Diagram E-R*)”.

Notasi-notasi simbolik di dalam Diagram E-R yang dapat kita gunakan adalah:

- Persigi panjang, menyatakan himpunan Entitas.
- Lingkar/*Elip*, menyatakan Atribut (Atribut yang berfungsi sebagai *Key* digarisbawahi).
- Belah ketupat, menyatakan Himpunan Relasi.
- Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.

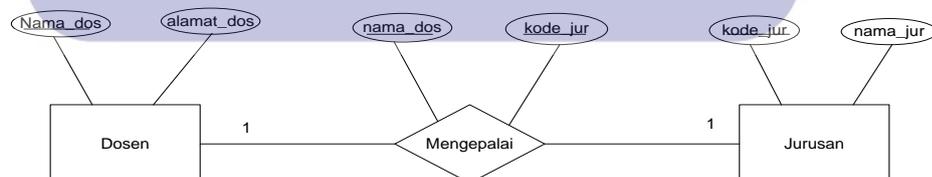
- e. Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).

A. Kardinalitas Relasi

Menurut Fathansyah (2012:82) menyatakan bahwa “Kardinalitas Relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak)”.

1. Relasi satu-ke-satu (*One-to-one*)

Misalnya adanya relasi antara himpunan entitas Dosen dengan himpunan entitas jurusan. Himpunan relasi kita beri nama ‘mengepalai’. Para relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan pasti dikepalai oleh paling banyak satu orang dosen. Maka penggambarannya adalah :



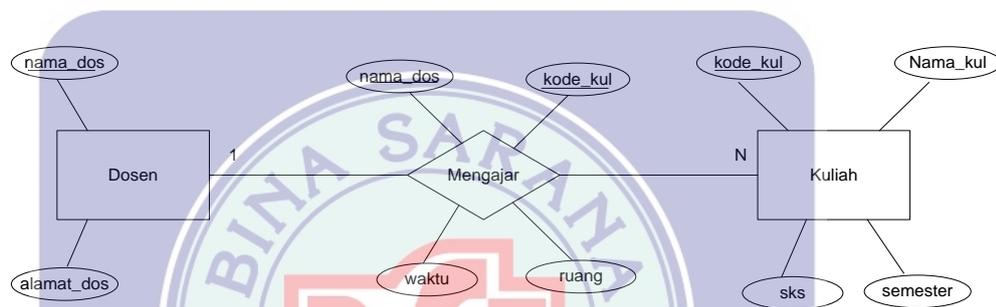
Sumber :Fathansyah (2012:82)

Gambar II.6.

Relasi Satu ke satu (*one-to-one*)

2. Relasi satu-ke-banyak (*One-to-many*)

Misalnya adanya relasi antara himpunan entitas dosen dengan himpunan entitas kuliah. Himpunan relasinya kita beri nama 'Mengajar'. Pada relasi ini, setiap dosen dapat mengajar lebih dari satu mata kuliah, sedangkan setiap matakuliah diajar hanya oleh paling banyak satu orang dosen. Maka penggambarannya adalah:



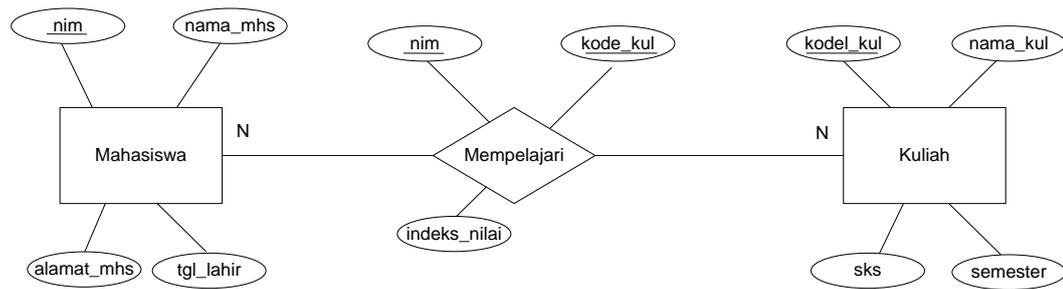
Sumber : Fathansyah (2012:83)

Gambar II.7.

Relasi satu ke banyak (*one-to-many*)

3. Relasi banyak-ke-banyak (*many-to-many*)

Adanya relasi antara himpunan entitas Mahasiswa dengan himpunan entitas Kuliah. Himpunan relasinya kita beri nama 'Mempelajari'. Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap matakuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Maka penggambarannya adalah:



Sumber : Fathansyah (2012:83)

Gambar II.8.

Relasi Banyak ke Banyak

B. LRS (*logical Relationship Structure*)

Menurut Hasugian dan Shidiq (2012:608) memberikan batasan bahwa LRS adalah “sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola atau aturan pemodelan tertentu dalam kaitannya dengan konvensi ke LRS”.

Perubahan yang terjadi yaitu mengikuti aturan-aturan sebagai berikut (Hasugian dan Shidiq, 2012:608):

1. Setiap entitas akan diubah kebentuk kotak.
2. Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan cardinality M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi).
3. Sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungan M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

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 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 di mengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS Metode yang lain di mulai dengan ER-diagram dan langsung dikonversikan ke LRS.

2.2.3. Pengujian Web

Menurut Rosa dan M. Shalahudin (2013:275) “*Black – Box testing* (penguji kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan 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*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

