

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar *Web*

Pembuatan Tugas Akhir ini tidak terlepas dari teori-teori yang mendukung dalam mempelajari serta merancang *website* yang diharapkan berfungsi secara maksimal. Kemudahan dalam melihat suatu *website* bagi setiap pengunjung akan sangat membantu dalam menyelesaikan setiap pencarian katalog yang ada.

##### A. *website*

Menurut Pramono dan Syaifi (2005:31), "*WebSite* adalah sekumpulan halaman *web* dimana satu halaman terkait dengan halaman yang lain". *Web Page* merupakan halaman khusus dari sebuah situs *web*. Untuk memudahkan penjajakan (navigasi) berbagai data dan informasi pada *internet* maka kita memerlukan program aplikasi *browser*. *Browser* yang populer dan paling banyak digunakan saat ini seperti *Microsoft Internet Explorer, Mozilla, Opera*. Untuk menggunakan fasilitas *internet*, maka selain komputer, modem, program *browser* dan jaringan telepon, ISP (*Internet Service Provider*) juga dibutuhkan untuk melakukan sambungan *internet* melalui telepon, HTTP (*Hypertext Transfer Protocol*), yaitu untuk mengirim sebuah teks ke dalam protokol, HTML (*Hypertext Mark Up Language*), yaitu suatu bahasa yang digunakan untuk menulis dihalaman *web*.



## B. *Internet*

Menurut Daryanto (2005:22) “*Internet* adalah dua komputer atau lebih yang saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer didunia (*Internasional*), yang saling berinteraksi dan bertukar informasi”.

*Internet* yang kita kenal saat ini pertama kali dikembangkan tahun 1969 dengan nama ARPAnet (*US Defense Advanced Research Project Agency*) oleh Departemen Pertahanan Amerika Serikat. Kejadian ini berlangsung dua bulan setelah Neil Armstrong melangkah ke bulan. ARPAnet dibangun dengan tujuan untuk membuat jaringan komputer yang tersebar dan untuk menghindari pemusatan informasi di satu titik yang dipandang rawan untuk dihancurkan apabila terjadi peperangan. Di awal 1980-an ARPAnet terpecah menjadi dua jaringan, yaitu ARPAnet dan Milnet (sebuah jaringan militer), akan tetapi keduanya mempunyai hubungan sehingga komunikasi antar jaringan tetap dapat dilakukan. Pada mulanya jaringan interkoneksi ini disebut *DARPA Internet*, tapi lama-kelamaan disebut *internet* saja.



Berikut beberapa istilah yang ada pada umumnya sering digunakan dalam hubungannya dengan *internet*, diantaranya:

### 1. WWW (*World Wide Web*)

Menurut Yuhefizar (2009:10) mengatakan “*Website* atau *World Wide Web* adalah suatu metode untuk menampilkan informasi di *internet*, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*”. Berbagai informasi dapat Anda temukan pada

WWW, seperti informasi politik, ekonomi, sosial, budaya, sastra, sejarah, teknologi, pendidikan dan sebagainya. Kita dapat mengumpamakan WWW ini merupakan perpustakaan besar yang menyediakan berbagai informasi yang dibutuhkan. TCP/IP (*Transmission Control Protocol/Internet Protocol*) yang memastikan bahwa semua hubungan bekerja dengan benar, sedangkan IP (*Internet Protocol*) yang mentransmisikan data dari satu komputer ke komputer lain.

## 2. *Web Site* (Situs *Web*)

Merupakan tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. Diumpamakan situs *Web* ini adalah sebuah buku yang berisi topik tertentu.

## 3. *Web Pages* (Halaman *Web*)

Merupakan sebuah halaman khusus dari situs *Web* tertentu. Diumpamakan halaman *Web* ini adalah sebuah halaman khusus buku dari situs *Web* tertentu.

## 4. *Homepage*

Merupakan sampul halaman yang berisi daftar isi atau menu dari sebuah situs *Website*.

## 5. *Browser*

Merupakan program aplikasi yang digunakan untuk memudahkan Anda melakukan navigasi berbagai data dan informasi pada WWW

## C. *E-Library*

Menurut Suwarno (2010:27) “Sebuah perpustakaan digital adalah suatu sistem yang menyediakan suatu komunitas pengguna dengan akses terpadu yang menjangkau keluasan informasi dan ilmu pengetahuan yang telah tersimpan dan terorganisasi dengan baik”.



Perpustakaan digital (*digital library* atau *electronic library* atau *virtual library*) adalah perpustakaan yang mempunyai koleksi buku sebagian besar dalam bentuk format digital dan yang bisa diakses dengan komputer. Jenis perpustakaan ini berbeda dengan jenis perpustakaan konvensional yang berupa kumpulan buku tercetak, film mikro ataupun kumpulan kaset audio, video, dan lain-lain. Isi dari perpustakaan digital berada dalam suatu komputer *server* yang bisa ditempatkan secara lokal, maupun di lokasi yang jauh, namun dapat diakses dengan cepat dan mudah lewat jaringan komputer

#### D. PHP (*Personal Home Page*)

Menurut Wahyono (2009:35) “PHP (*Personal Home Page*) adalah *script* untuk membuat suatu aplikasi yang dapat terintegrasi ke dalam halaman HTML, sehingga suatu halaman *website* tidak lagi bersifat statis, namun menjadi bersifat dinamis”.

PHP merupakan *software yang open source* (gratis) dan mampu lintas *platform*, yaitu dapat digunakan dengan menggunakan sistem operasi dan *web server* apapun. PHP mampu berjalan di *windows* dan beberapa versi *Linux*. PHP juga dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

PHP pertama kali dibuat oleh Rasmus Leodorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengelola data formulir dari *web*.



### E. HTML (*Hypertext Markup Language*)

Menurut Sugiri dan Kurniawan (2007:1) “HTML (*Hypertxt Markup Language*) adalah sebuah protokol yang digunakan untuk membuat *format* suatu dokumen *web* yang mampu dibaca dalam *browser* dari berbagai *platform* komputer”.

Sifat bahasa HTML ini merupakan *client script*, dimana dokumen tersebut dapat dibuka dalam komputer *stand alone* yang tidak membutuhkan *server* untuk dapat menampilkan didalam *browser*. Dokumen HTML merupakan *file* yang pada umumnya berekstensi, htm, atau html dimana bahasa HTML tersebut tersusun atas *tag* yang berformat <isi tag>.

### F. CSS (*Cascading Style Sheets*)

Menurut Castro (2007:20) “CSS (*Cascading Style Sheets*) adalah *system* yang di gunakan untuk mengatur tampilan halaman sebuah web”. Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

### G. Javascript

Menurut Hakim (2014:4) “*JavaScript* merupakan pemrograman *web* yang berjalan di sisi klien (*browser*), sehingga *JavaScript* dapat membuat *website* lebih hidup (interaktif dan responsif)”. *JavaScript* bekerja pada sisi *browser*, artinya untuk menampilkan halaman *web*, *user* menuliskan alamat *web* di *address bar url*. Setelah itu, *browser* mengambil *file* HTML (dengan *file JavaScript* yang melekat padanya

jika memang ada) ke *server* yang beralamat di URL yang diketikkan oleh *user*. Selesai *file* diambil, *file* ditampilkan pada *browser*. Setelah *file JavaScript* berada pada *browser*, barulah skrip *JavaScript* tersebut bekerja.

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses skrip pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan *JavaScript* adalah membuka halaman *pop up*, fungsi validasi pada *form* sebelum data dikirimkan ke *server*, merubah gambar kursor ketika melewati objek tertentu, dan lain-lain. Yang harus diperhatikan dalam pengelolaan pemrograman *JavaScript* diantaranya *JavaScript* adalah bahasa pemrograman yang *case sensitive*, yang artinya *JavaScript* membedakan huruf kecil dan huruf besar. Hal ini sama seperti bahasa pemrograman Turbo C atau C++ dimana huruf "A" tidak sama dengan huruf "a".

#### H. JQuery

Menurut Hakim (2014:3), *jQuery* adalah *JavaScript Library*: kumpulan kode/fungsi *JavaScript* siap pakai, 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. Disinilah peran *jQuery* sebagai *JavaScript Library* menyederhanakan kode *JavaScript*. Hal ini sesuai dengan slogannya "*Write less, do more*".

#### I. Basis Data

Menurut Stephens dan Plew (2005) "Basis Data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan". Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis

data (*database management system* (DBMS)). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya yang mana penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data yang mana ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system* (DBMS)). Jika konteksnya

sudah jelas, banyak *administrator* dan *programmer* menggunakan istilah basis data untuk kedua arti tersebut.

### 1. Lingkungan Basis data

Lingkungan basis data adalah sebuah habitat dimana terdapat basis data untuk bisnis. Dalam lingkungan basis data, pengguna memiliki alat untuk mengakses data. Pengguna melakukan semua tipe pekerjaan dan keperluan mereka bervariasi seperti menggali data (*data mining*), memodifikasi data, atau berusaha membuat data baru. Masih dalam lingkungan basis data, pengguna tertentu tidak diperbolehkan mengakses data, baik secara fisik maupun logis. (Koh dalam Janner Simarmata dan Imam Paryudi, 2006: 33).

### 2. Tahapan Perancangan Basis Data

Perancangan basis data merupakan upaya untuk membangun sebuah basis data dalam suatu lingkungan bisnis. Untuk membangun sebuah basis data terdapat tahapan-tahapan yang perlu kita lalui yaitu:

- a. Perencanaan basis data
- b. Mendefinisikan sistem
- c. Analisa dan mengumpulkan kebutuhan
- d. Perancangan basis data
- e. Perancangan aplikasi
- f. Membuat prototipe
- g. Implementasi
- h. Konversi data
- i. Pengujian



j. Pemeliharaan operasional

### 3. Bahasa pada Basis Data

Terdapat dua jenis bahasa komputer yang digunakan saat kita ingin membangun dan memanipulasi sebuah basis data, yaitu:

- a. *Data Definition Language* (DDL)
- b. *Data Manipulation Language* (DML)

#### J. MySQL (*My Structured Query Language*)

Menurut Kadir (2008:348) mengatakan bahwa “MySQL merupakan salah satu jenis *database server* yang sangat terkenal yang bersifat *open source* (anda tidak perlu membayar untuk menggunakannya) dengan menggunakan bahasa SQL untuk mengakses *databasenya*.”

MySQL merupakan *software* sistem manajemen *database* (*Database Management System* (DBMS)) yang sangat populer di kalangan pemrograman *web*, terutama di lingkungan Linux dengan menggunakan *script* PHP dan Perl.

Fungsi MySQL dapat dikatakan sebagai *interpreter query*, karena setiap kita menggunakan *query* SQL (perintah SQL) kita harus meletakkannya di dalam fungsi ini. Dengan kata lain, SQL tidak dapat dijadikan tanpa adanya fungsi MySQL. MySQL termasuk jenis *relational database management system* (RDBMS). Sehingga istilah seperti tabel, baris dan kolom tetap digunakan dalam MySQL.s

SQL (dibaca “ess-que-el”) merupakan kependekan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *Database*. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima,



memelihara dan mengatur akses-akses ke basis data serta digunakan untuk memanipulasi dan menampilkan data dari *database*.

### K. *Waterfall*

Menurut Pressman (2010:75), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini seharusnya adalah “*Linear Sequential Model*”. Model ini sering disebut dengan “*classic life cycle*” atau model *waterfall*. Model ini termasuk kedalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Berikut adalah gambar pengembangan perangkat lunak berurutan/ linear



#### 1. Tahapan Metode *Waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: *requirement* (analisis kebutuhan), *system design* (desain sistem), *Coding & Testing*, Penerapan Program, pemeliharaan.

##### a. *Requirement* (analisis kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study* literatur. Seseorang sistem analisis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan

menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

b. *Design System* (desain sistem)

Proses desain akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, *representasi interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

c. *Coding & Testing* (penulisan kode program / *implementation*)

*Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

d. *Integration & Testing* (Penerapan / Pengujian Program)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

### e. *Operation & Maintenance* (Pemeliharaan)

Perangkat lunak yang susah 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.

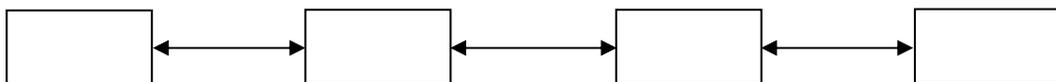
## 2.2. Teori Pendukung

### 2.2.1. Struktur Navigasi

Menurut Prihatna (2005:51) “Struktur navigasi adalah susunan menu atau hirarki dari suatu situs yang menggambarkan isi setiap halaman dan *link* atau navigasi tiap halaman pada suatu situs web”. Struktur navigasi yang biasa digunakan, yaitu:

#### 1. Struktur Navigasi *Linier*

Struktur navigasi *linier* merupakan suatu struktur dengan satu alur atau satu rangkaian cerita dan menampilkan satu demi satu tampilan layar secara berurutan. Tampilan yang dapat ditampilkan pada jenis struktur ini adalah satu halaman sebelumnya atau satu halaman sesudahnya. Pada jenis struktur navigasi ini tidak diperkenalkan adanya percabangan.



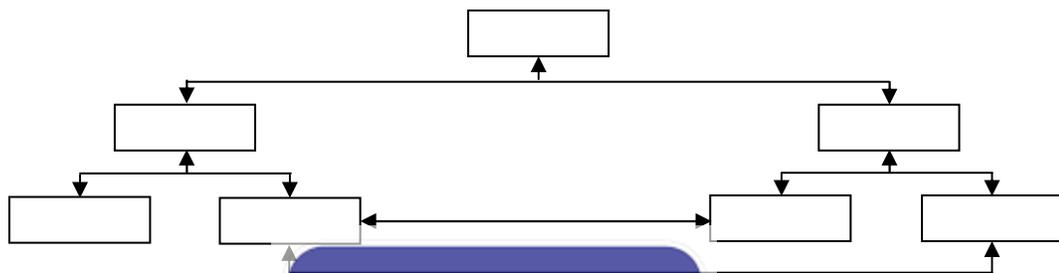
Sumber: Prihatna (2005:51)

**Gambar II.1.**

**Struktur Navigasi *Linier***

## 2. Struktur Navigasi *Non-Linier*

Struktur navigasi ini disebut juga struktur tidak berurutan yang merupakan pengembangan dari struktur navigasi *linier*. Pada struktur ini diperbolehkan membuat halaman percabangan tetapi berbeda dengan struktur hirarki. Walaupun terdapat percabangan tetapi tidak ada *Master Page* dan *Slave Page*.

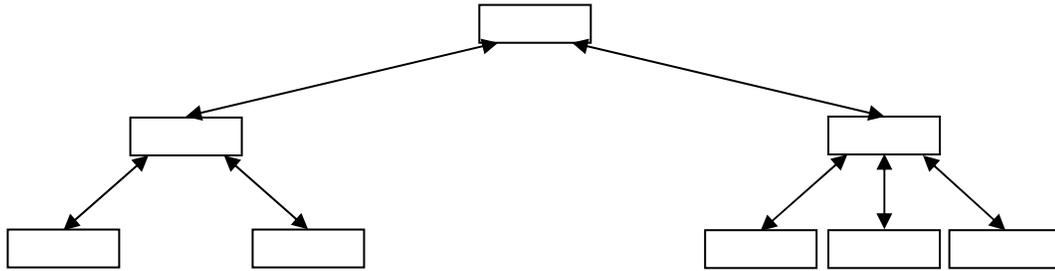


Sumber: Prihatna (2005:52)



## 3. Struktur Navigasi Hirarki

Struktur Hirarki (bercabang) merupakan struktur halaman yang berbentuk seperti pohon *binary*. Tampilan pada menu utama disebut *master page*. Menu utama itu memiliki percabangan yang dinamakan *slave page* atau halaman pendukung.



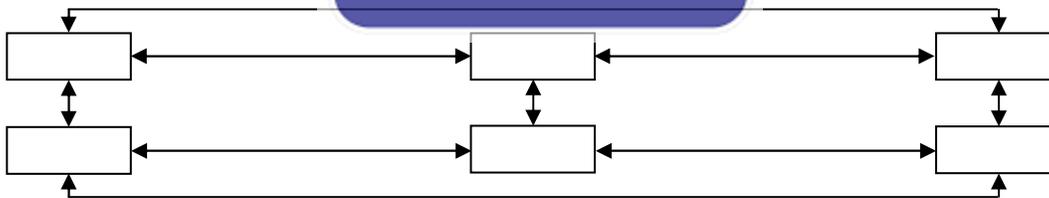
Sumber: Prihatna (2005:53)

**Gambar II.3.**

### Struktur Navigasi Hirarki

#### 4. Struktur Navigasi Campuran

Struktur navigasi ini merupakan gabungan dari ketiga struktur navigasi sebelumnya yaitu *Linier*, *Non-Linier*, *Hirarki*. Struktur navigasi ini biasa disebut struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan aplikasi sebab struktur ini dapat memberikan keinteraksian yang lebih tinggi.



Sumber: Prihatna (2005:54).

**Gamabar II.4.**

### Struktur Navigasi Campuran

### 2.2.2. ERD (*Entity Relationship Diagram*)

Pengertian ERD menurut Conolly (2005 :3354) “ERD adalah penggambaran dari sebuah kebutuhan penyimpanan data dengan cara kerja dari suatu pekerjaan atau organisasi yang bebas dari ambiguitas, ERD digunakan untuk mengidentifikasi data yang akan disimpan diolah dan di ubah untuk mendukung aktifitas bisnis suatu organisasi”.

Tiap relasi disimpan sebagai sebuah file tersendiri. Perancangan basis data merupakan suatu kegiatan yang setidaknya bertujuan sebagai berikut:

1. Menghilangkan redundansi data
2. Meminimumkan jumlah relasi di dalam basis data
3. Membuat relasi berada dalam bentuk normal, sehingga dapat meminimumkan permasalahan berkenaan dengan penambahan, pembaharuan dan penghapusan

Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Menjadi mentalist Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya. Contoh: *entity* Mahasiswa, mempunyai atribut nama, umur, alamat, dan nim. Diagram E-R terdiri dari:

1. Kotak persegi panjang, menggambarkan himpunan entitas
2. Elip, menggambarkan atribut-atribut entitas
3. Diamon, menggambarkan hubungan antara himpunan entitas
4. Garis, yang menghubungkan antar objek dalam diagram E-R



### 2.2.3. 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 dan tipe *record*. Beberapa tipe *record* digunakan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda LRS dan ER-Diagram nama dan tipe *record* berada diluar kotak *field 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-filed* yang kelihatan pada kedua *link* tipe *record*. Penggambaran LRS mulai dengan menggunakan mode yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode lain yang dimulai dengan ER-Diagram dan langsung dikonversikan ke LRS.



### 2.2.4. *Black Box*

Pada siklus hidup pengembangan perangkat lunak (*System Development Life Cycle (SDLC)*) salah satu proses yang harus dilakukan adalah proses pengujian (*testing*). Salah satu metode pengujian perangkat lunak adalah *Black Box Testing*. Menurut Pressman (2010:482) ”*Black-box testing* atau biasa dikenal sebagai

*behavioral testing* adalah proses pengujian yang berfokus *requirement* (persyaratan) fungsional dari perangkat lunak”.

Fokus dari pengujian menggunakan metode *Black Box* adalah pada pengujian fungsionalitas dan output dihasilkan aplikasi. Pengujian *black box* didesain untuk mengungkap kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme *internal* atau komponen dari suatu program..

Salah satu dari pengujian *Black Box* yang dapat dilakukan oleh seorang penguji independen adalah *Functional testing*. Basis uji dari *functional testing* ini adalah pada spesifikasi dari komponen perangkat lunak yang akan diuji. *Functional testing* memastikan bahwa semua kebutuhan telah dipenuhi dalam sistem aplikasi. Dengan demikian fungsinya adalah tugas-tugas yang didesain untuk dilaksanakan sistem. *Functional testing* berkonsentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi.

