

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Pada 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 fasilitas hotel dan pemesanan kamar hotel.

2.1.1. Website

Website atau yang disingkat dengan *web* halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Secara etimologi *website* adalah kumpulan dari halaman-halaman situs yang biasanya terangkum dalam domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di internet, situs tersebut terdiri dari seluruh situs web yang tersedia kepada publik.

Menurut Sunarto (2009:52), “Web adalah fasilitas yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh”. Untuk melakukan *browsing* dibutuhkan sebuah *software* umumnya disebut dengan *browser* yang mendukung *software* tersebut seperti : *Internet Explorer*, *Netscape Opera*, *Mozilla Firefox* dan sebagainya. Program ini menyediakan menu untuk menulis alamat-alamat *web* yang disebut *address bar*.



Sumber : Sunarto (2009:52)

Gambar II.1

Software Pendukung *Browsing*

A. *Web Browser*

Menurut EMS (E-Media Komputindo) (2009:1) “*Web Browser* adalah aplikasi perangkat lunak untuk berinteraksi dengan *teks, image, video, games* dan informasi lainnya yang berlokasi pada halaman *web* pada *World Wide Web* atau *Local Area Network* ”. Kedua, *web browser* menginterpretasikan tag *HTML* di halaman untuk menampilkan halaman *web* sebagaimana yang dikehendaki oleh pembuat halaman *web* itu sendiri.

B. *Web Server*

Web server merupakan *software* yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. *Web server* merupakan mesin dimana tempat aplikasi atau *software* beroperasi dalam mendistribusikan *webpage* ke *user* tentu saja sesuai dengan permintaan *user*.

C. Internet (*Interconnection Networking*)

Menurut Sibero (2011:10) “*Interconnection Networking* (Internet) adalah jaringan computer yang menghubungkan antar jaringan secara *global*, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas”. *Internet* juga menggunakan *protocol* komunikasi *Transmission Control/Internet Protocol Suite* (TCP/IP). *Protocol* adalah spesifikasi sederhana mengenai bagaimana komputer saling bertukar informasi.

Transmission Control Protocol (TCP) bertugas untuk memastikan bahwa semua hubungan bekerja dengan benar, sedangkan *Internet Protocol Suite* (IP) yang mentransmisikan data dari suatu komputer ke komputer yang lain.

D. Sistem Informasi

Sistem informasi merupakan kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi manajemen. Sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Sistem informasi tersebut tidak hanya pada penggunaan organisasi Teknologi Informasi dan Komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

Data yang diolah menjadi bentuk yang berguna bagi para pemakainya. Data yang diolah saja tidak cukup dapat dikatakan sebagai suatu informasi. Untuk dapat berguna, maka informasi harus didukung oleh tiga pilar sebagai berikut: tepat kepada orangnya atau relevan (*relevance*), tepat waktu (*timeliness*), dan tepat nilainya atau

akurat (*accurate*). Keluaran yang tidak didukung oleh tiga pilar ini tidak dapat dikatakan sebagai informasi yang berguna, tetapi merupakan sampah (*garbage*).

E. Sistem Informasi Reservasi

Reservasi atau pemesanan dalam bahasa Inggris *Reservation* yang berasal dari kata *to reserve* yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation* yaitu pemesanan suatu tempat fasilitas. Jadi secara umum Reservasi yaitu pemesanan fasilitas yang diantaranya akomodasi, *meal*, *seat* pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club* dan sebagainya. Sehingga sistem informasi reservasi tersebut dapat menyajikan informasi fasilitas pada waktu tertentu.

Perancangan Sistem Informasi Reservasi Kamar Hotel Berbasis Web merupakan sistem informasi yang akan membantu proses pengolahan data pemesanan kamar hotel sesuai dengan tanggal *check-in* serta *check-out*, pembayaran dan transaksi pemesanan kamar jarak jauh yang dinamis agar memudahkan pelanggan.

2.1.2. Bahasa Pemrograman

A. *Personal Home Page (PHP)*

Awalnya PHP itu merupakan kepanjangan dari *Personal Home Page* yang dapat diartikan sebagai “situs personal”. Namun, semakin berjalannya waktu kini PHP telah berubah wujud menjadi *Hypertext Preprocessor*.

Menurut Saputra (2012:2), “PHP merupakan suatu bahasa pemrograman yang hanya dapat berjalan pada sisi server (*Server Side Scripting*)”. Artinya proses yang dibuat dengan PHP tidak akan berjalan tanpa menggunakan *web server*. PHP digunakan untuk membangun aplikasi berbasis *web* agar *web* tersebut dapat digunakan secara dinamis, seperti menambah, mengubah, membaca, serta menghapus suatu konten.

B. CSS (*Cascading Style Sheet*)

Menurut Saputra (2012:5), “CSS atau yang memiliki kepanjangan *Cascading Style Sheet* merupakan suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapih, terstruktur dan seragam”.

CSS merupakan pemrograman wajib yang harus dikuasai oleh setiap pembuat program (*web programming*), terlebih lagi adalah pendesain *web* (*web designer*). CSS saat ini dikembangkan oleh *world wide web consortium* (W3C) dan menjadi bahasa standar dalam pembuat *web*. CSS difungsikan sebagai penopang atau pendukung dan pelengkap dari *file* html yang berperan dalam penataan kerangka dan *layout*. CSS lintas *platform*, maksudnya dapat dijalankan pada berbagai macam sistem operasi dan *web browser*. Secara umum, yang dilakukan oleh CSS adalah pengaturan *layout*, teks, gambar, warna, tebal, spasi dan lain sebagainya.

C. HTML (*Hyper Text Markup Language*)

Menurut Saputra (2012:1), “HTML adalah bahasa paling dasar dan sangat penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*”.

HTML (*Hyper Text Markup Language*) merupakan suatu metode untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen. HTML sendiri bukan tergolong pada suatu bahasa pemrograman karena sifatnya yang hanya memberikan tanda (*marking up*) pada suatu naskah teks dan bukan sebagai program.

Berdasarkan kata-kata penyusunnya HTML dapat diartikan lebih dalam lagi menjadi :

1. *Hypertext*

Link hypertext adalah kata atau *frase* yang dapat menunjukkan hubungan suatu naskah dokumen atau naskah-naskah lainnya. Jika kita klik pada kata atau *frase* untuk mengikuti *link* ini maka *web browser* akan memindahkan tampilan pada bagian lain dari naskah atau dokumen yang kita tuju.

2. *Markup*

Pada pengertiannya disini *markup* menunjukkan bahwa pada *file* HTML berisi suatu instruksi tertentu yang dapat memberikan suatu format pada dokumen yang akan ditampilkan pada *World Wide Web*.

3. *Language*

Meski HTML sendiri bukan merupakan bahasa pemrograman, HTML merupakan kumpulan dari beberapa instruksi yang dapat digunakan untuk mengubah-ubah format naskah atau dokumen.

D. *JQuery*

Menurut Wahana Komputer (2012:8) “*JQuery* merupakan salah satu dari sekian banyak *framework* yang digunakan dan memiliki pengguna yang paling banyak. *JQuery* merupakan *JavaScript* yang dibangun untuk mempercepat dan memperingkas serta menyederhanakan manipulasi dokumen *HTML*, penanganan *event*, animasi, dan interaksi *Ajax* untuk mempercepat pengembangan *web*. Dengan *JQuery*, *developer* akan dimanjakan dengan suatu pemrograman *JavaScript* yang sangat sederhana jika dibandingkan dengan *native Java Script*. Sebelum anda memulai mempelajari *JQuery*, anda harus mempunyai pengetahuan dasar mengenai *HTML*, *CSS*, dan *JavaScript*.”

E. *JavaScript*

Javascript menurut Sunyoto (2007:17) adalah “Bahasa *scripting* yang populer di *internet* dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer (IE)*, *Mozilla Firefox*, *Google*, *Netscape* dan *Opera*”. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *tag Script*.”

2.1.3. Basis Data

A. Pengenalan *Database*

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta adalah representasi dari semua fakta yang ada pada dunia nyata. Definisi lebih rinci diberikan oleh *Fathansyah* (2012:1) “Basis data (*database*) dapat dibayangkan sebagai sebuah lemari jika kita memiliki sebuah lemari arsip dan bertugas untuk mengelolanya”.

Ada tiga basis data yang penulis gunakan, yaitu:

1. XAMPP

Menurut Winarno (2014:1) “*XAMPP* adalah *software web server* yang bisa dipakai untuk mengakomodasikan *system* operasi yang anda Pakai (X), *Apache* (A), *MySQL* (M), *PHP* (P), dan *Perl* (P).

Kelebihan lain dari XAMPP adalah dapat berjalan pada beberapa Sistem Operasi seperti Windows, Linux, Mac dan lain-lain. Progtammer dan Developer biasa menggunakan XAMPP untuk membuat server local atau biasa disebut localhost.

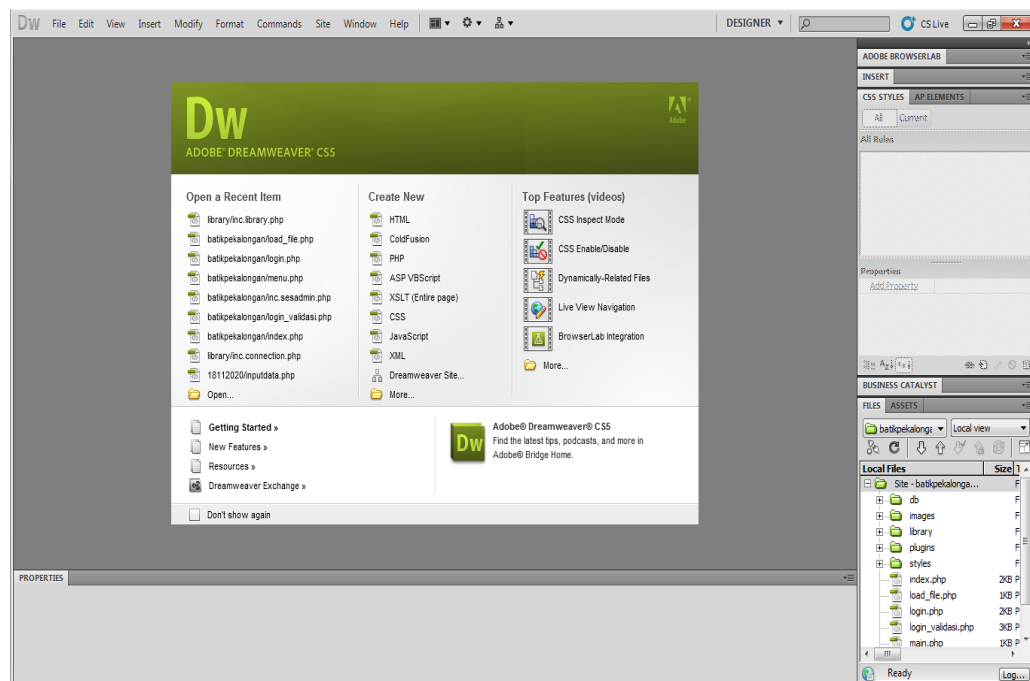
2. *MySQL*

Menurut Anhar (2010:6) “*MySQL* adalah suatu perangkat lunak sistem manajemen basis data (*Database Management System*) atau DBMS yang menggunakan perintah standar *SQL* (*Structured Query Language*)”. Dimana *MySQL* mampu untuk melakukan banyak eksekusi perintah *query* dalam satu permintaan (*multithread*), baik itu menerima dan mengirimkan data. Menu *MySQL* digunakan untuk manajemen *database MySQL* yang ada di *server hosting/web server*. *MySQL*

adalah pengolah *database* yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur dan menampilkan data.

3. *Adobe Dreamweaver CS5*

Menurut Sadeli (2011:2) “*Dreamweaver CS5* merupakan perangkat lunak *web editor* keluaran *Adobe System* yang digunakan untuk membangun dan mendesain suatu *website* dengan fitur-fitur yang menarik dan mudah untuk penggunaannya.”



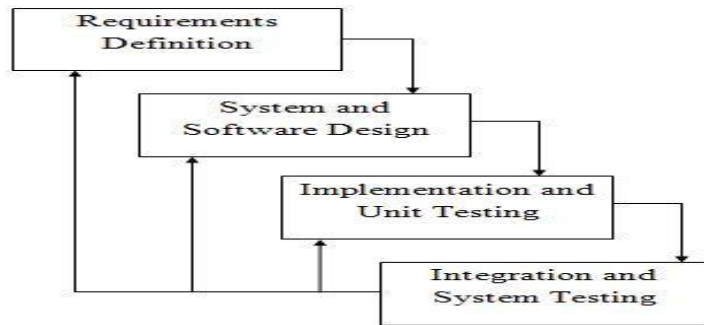
Sumber : Sadeli (2011:2)

Gambar II.2

Tampilan Adobe Dreamweaver CS5

2.1.4. Model Pengembangan Sistem Perangkat Lunak

Model ini merupakan yang paling pertama dipublikasikan. Model ini berasal dari proses sistem dari *engineering* yang lebih umum, model ini dikenal sebagai *waterfall model* atau siklus hidup *software* (Sommerville 2007:66). Beberapa tahapan dalam *waterfall model* menurut Sommerville (2007:66).



Sumber: Sommerville (2007:66)

Gambar II.3

Waterfall Model

Tahap utama model ini dibagi kedalam 4 bagian berdasarkan pengembangan kegiatannya :

- A. *Requirements Definition* : Layanan, batasan dan tujuan dari sistem ditetapkan melalui konsultasi dengan penggunaan sistem. Semua itu didefinisikan secara *detail* dan bertindak sebagai spesifikasi sistem.
- B. *System and Software Design* : Proses desain sistem membagi kebutuhan menjadi *hardware* dan *software*. Ini menetapkan hampir seluruh perancangan sistem. Desain *software* melibatkan pengidentifikasian dan penggambaran mengenai pemisahan dasar sistem *software* dan hubungannya.

- C. *Implementation and Unit Testing* : Dalam tahap ini, desain *software* adalah menyadari sebagai kumpulan program atau suatu program. *Unit testing* melibatkan verifikasi bahwa setiap *unit* telah mencapai spesiikasinya.
- D. *Integration and Sistem Testing* : Suatu program atau kumpulan program diintegrasikan dan dites sebagai sistem yang telah selesai, untuk menjamin bahwa kebutuhan *software* telah terpenuhi. Setelah pengetesan, sistem *software* dikirimkan kepada pelanggan.

Menurut Sommerville (2007:67-68) “kelebihan *waterfall* model adalah dokumentasi dihasilkan dalam setiap tahap, dan ini cocok dengan model proses *engineering*”. Komitmen harus dibuat dalam tahap awal dari proses ini, sehingga sulit menanggapi perubahan permintaan pelanggan. Oleh karena itu, *waterfall* model seharusnya hanya digunakan saat kebutuhan sangat dimengerti dan tidak mungkin berubah sama sekali dalam pengembangan sistem.

2.2. Teori Pendukung

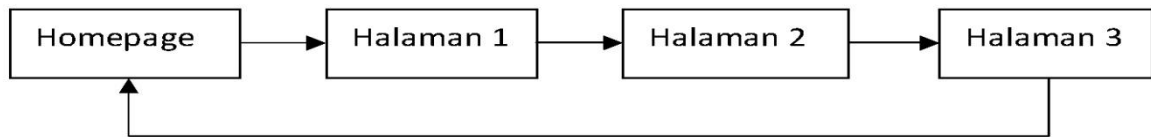
2.2.1. Struktur Navigasi

Menurut Prihatna (2010:51) “Struktur Navigasi adalah susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi tiap halaman pada suatu situs *web*”.

Struktur navigasi dapat digolongkan menurut kebutuhan objek, kemudahan pemakaian dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan situs *web*. Bentuk dasar struktur navigasi adalah sebagai berikut :

A. Struktur Navigasi *Linear* (Satu alur)

Linear atau 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. Salah satu yang terpenting dari struktur ini adalah tidak diperkenankan terjadinya percabangan.



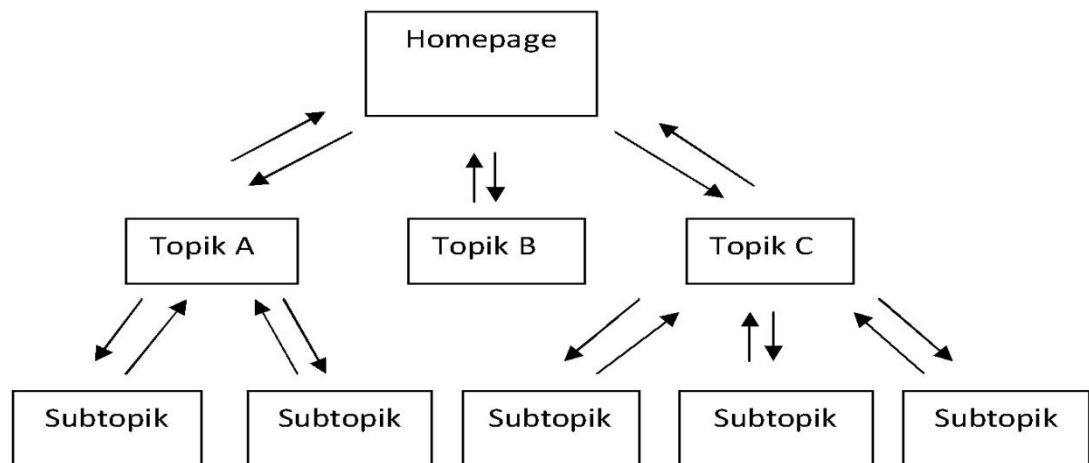
Sumber : Sutopo (2007:6)

Gambar II.4

Struktur Navigasi *Linear*

B. Struktur Navigasi *Hierarchical* (Hirarki)

Menurut Sutopo (2007:7), “*Hierarchical model* diadaptasi oleh top-down design. Konsep navigasi ini dimulai dari satu *node* yang menjadi *homepage*. Dari *homepage* dapat dibuat beberapa cabang ke halaman-halaman utama”.



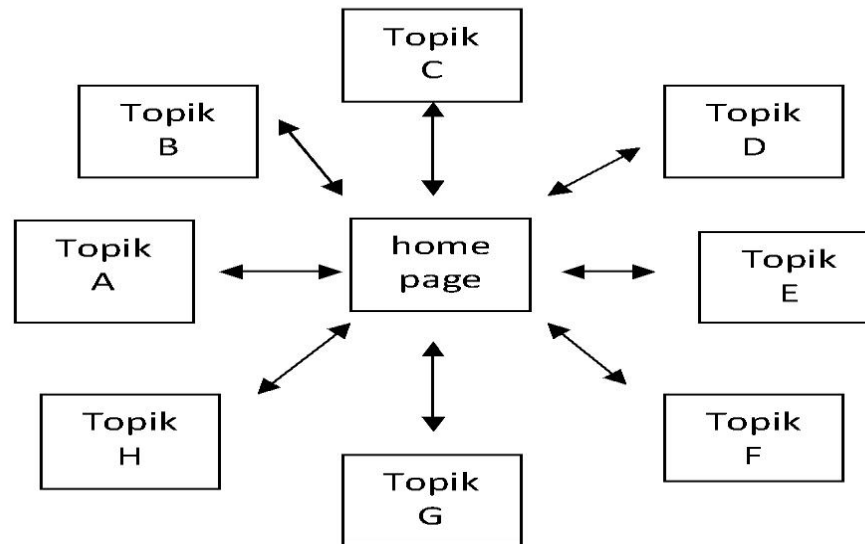
Sumber : Sutopo (2007:6)

Gambar II.5

Struktur Navigasi *Hierarchical* (Hirarki)

C. Struktur Navigasi *Non Linear* (Tidak Berurut)

Dalam *spoke-and-hub model* hanya ada dua macam *link* yaitu dari *homepage* ke halaman tertentu dan dari halaman tertentu ke halaman *homepage*. *Spoke-and-hub model* hanya menggunakan satu *node* untuk menghubungkan satu *node* ke *node* yang lain.



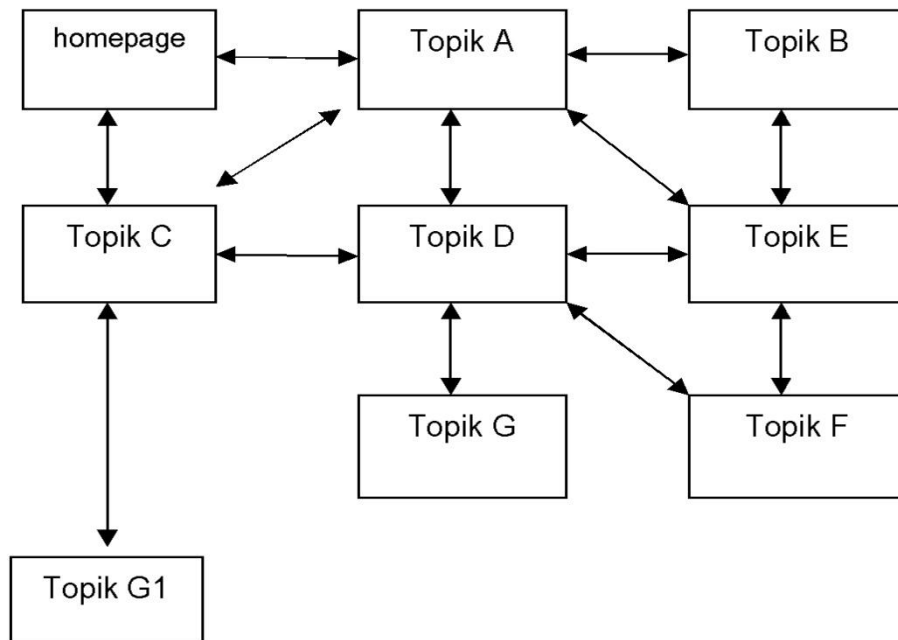
Sumber : Sutopo (2007:7)

Gambar II.6

Struktur *Non Linear*

D. Struktur Navigasi *Full Web Model*

Menurut Sutopo (2007:8) menerangkan bahwa “*Full web model* memberikan kemampuan *hyperlink* yang banyak. Full web model banyak digunakan karena user dapat mengakses semua topik dengan subtopik dengan cepat. Namun kelemahan dari model ini, yaitu dapat berakibat *user* kehilangan cara untuk kembali ke topik sebelumnya”.



Sumber : Sutopo (2007:8)

Gambar II.7

Struktur Navigasi *Composite*

2.2.2. *Entity Relationship Diagram (ERD)*

Menurut Brady dan Loonam (2010:67), “*Entity Relationship diagram (ERD)* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh Sistem Analis dalam tahap analisis persyaratan proyek pengembangan sistem”. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan *detail* pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

Terdapat 3 komponen yang akan dibentuk yaitu :

A. Entitas

Pada *post* sebelumnya mengenai basis data telah dijelaskan sedikit tentang pengertian *entity* (entitas) yaitu suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Pengertian lainnya menurut Brady dan Loonan (2010:67),” Entitas adalah objek yang menari di bidang organisasi yang dimodelkan”.

Contoh : Mahasiswa, Kartu Anggota Perpustakaan (KAP), dan Buku.

B. Relasi

Relasi adalah hubungan antara dua jenis entitas dan dipresentasikan sebagai garis lurus yang menghubungkan dua entitas.

Contoh : Mahasiswa mendaftar sebagai anggota perpustakaan (KAP), relasinya adalah mendaftar.

C. Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.

Derajat Relasi atau kardinalitas rasio

Menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya. Beberapa relasi dalam *ERD* adalah :

1. *One to One* (1:1)

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitupun sebaliknya.

2. *One to Many* (1:M)

Setiap anggota entitas A dapat berhubungan lebih dari satu anggota entitas B tetapi tidak sebaliknya

3. *Many to Many* (M:M)

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

2.2.3. *Logical Relational Structure* (LRS)

Menurut Simarmata (2008:12), “LRS adalah representasi dari struktur *record-record* pada tabel-tabel yang berbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah tabel dan *Foreign Key* pada LRS dibentuk dengan nomer dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik.

Perbedaan LRS dengan ERD dan tipe *record* berada diluar *field* tipe *record* ditempatkan. LRS 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 type 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 *Entity Relationship Diagram* dan langsung dikonversikan ke LRS.

Berikut tahapan transformasi ERD ke LRS menurut Wulandari (2013:15-16)

- A. Konversi *ERD* ke *LRS*, *Entity Relationship Diagram* harus diubah ke bentuk *LRS* (struktur *record* secara *logic*). Dari bentuk *LRS* inilah yang nantinya dapat ditransformasikan ke bentuk relasiable.
- B. Konversi *ERD* ke *LRS* sebuah model 24able24 yang digambarkan dengan sebuah model 24able24 yang digambarkan dengan sebuah *ERD* akan mengikuti pola pemodelan tertentu. Dalam kaitannya dengan konversi ke *LRS*, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:
 - 1. Setiap entitas diubah ke bentuk kotak dengan nama entitas, berada diluar kotak dan atribut berada didalam kotak.
 - 2. 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 tabel atau tabel adalah bentuk pernyataan data secara grafis dimensi, yang terdiri dari kolom dan baris. Relasi adalah bentuk visual bentuk lingkaran *Entity Relationship Diagram* dikenal dengan sebutan atribut. Konversi dari *logical structure*, dilakukan dengan cara:
 - A. Nama *logical record structure* menjadi nama relasi.
 - B. Tiap atribut menjadi sebuah kolom didalam relasi.

2.2.4. Pengujian Web (*Black Box Testing*)

Menurut *Pressman* (2010:14) “*Black Box Testing* (pengujian kotak hitam) atau disebut juga *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak”. Artinya teknik *Black Box Testing* memungkinkan untuk mendapat set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black Box Testing* bukan merupakan *alternatif* dari pengujian *White Box Testing*. Sebaliknya, *Black Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan metode *White Box Testing*.

Uji coba *Black Box Testing* untuk menemukan kesalahan dalam kategori berikut :

- A. Fungsi tidak benar atau hilang.
- B. Kesalahan *interface* atau antarmuka.
- C. Kesalahan dalam struktur data atau akses *database eksternal*.
- D. Kesalahan kinerja atau perilaku.
- E. Kesalahan inisialisasi dan terminasi.

Tidak seperti *White Box Testing* yang dilakukan pada awal proses pengujian, *Black Box Testing* cenderung diterapkan pada tahap selanjutnya dari pengujian. Karena *Black Box Testing* sengaja mengabaikan struktur control perhatian difokuskan pada domain informasi.