

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Web memiliki konsep-konsep dasar yang menjadi acuan dalam mempelajari suatu *web*. Pada bab ini penulis akan membahas beberapa teori, adapun beberapa teori yang akan dijelaskan untuk membantu dalam pembuatan *web* sebagai berikut:

A. Pengertian Internet

Menurut (Krisianto, 2014:1) “Internet adalah salah satu bentuk media komunikasi dan informasi interaktif.” Wujud internet adalah jaringan komputer yang terhubung di seluruh dunia. Internet digunakan untuk mengirim informasi antar komputer di seluruh dunia. Sehingga, melalui internet kita bisa mengakses dan bertukar informasi secara cepat. Mengakses internet berarti mengakses informasi. Alat untuk mengakses internet bisa menggunakan komputer/laptop, *handphone*, dan berbagai *smartphone/gadget*.

B. Pengertian Website

Menurut Helianthusonfri (2013:5) “*Website* adalah halaman informasi yang ada di internet, di mana halaman tersebut merupakan kumpulan komponen yang terdiri dari teks, gambar, dan atau suara animasi.”

Dikenal tiga jenis *website*, yaitu:

1. *Website Statis*, adalah jenis *website* yang isi didalam *website* tersebut tidak dimaksudkan untuk diupdate secara berkala. Jadi update (pembaruan) di *website* statis ini jarang sekali terjadi, dan biasanya jika ada update, akan dilakukan oleh orang-orang tertentu saja.
2. *Website dinamis*, adalah *website* yang isinya bisa diupdate secara cepat dan sering. Berbeda dengan *web* statis yang terkesan sederhana, di *web* dinamis ada banyak hal yang bisa kita lakukan. Misalnya kita bisa menampilkan *flash*, mengomentari *web* tersebut, dan banyak lagi. Contoh dari sebuah *web* dinamis adalah *web* jenis blog.
3. *Website interaktif*, adalah jenis *website* yang memungkinkan pengguna *web* tersebut untuk langsung berinteraksi, baik itu dengan pengguna lain ataupun dengan *website* itu sendiri.

Adapun fungsi dari *website*, yaitu:

1. Untuk kegiatan publikasi informasi
2. Untuk kegiatan promosi
3. Untuk kegiatan jual beli
4. Untuk membangun komunitas
5. Untuk menciptakan kesan profesional dan keren

Beberapa istilah yang ada dalam *website* :

1. *Web Browser*

Menurut Setiawan (2017:4) “*Web Browser* adalah sebuah aplikasi yang digunakan untuk menjelajahi situs-situs didunia maya atau yang biasa disebut dengan *website*.”

2. *Web Server*

Menurut Marisa (2017:142) “*Web server* merupakan *software* dalam *server* yang berfungsi untuk menerima permintaan *request* berupa halaman *web* melalui HTTP atau HTTPS dari klien yang dikenal dengan *web Browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML.”

C. **Bahasa Pemrograman**

Dalam penulisan tugas akhir ini penulis menggunakan bahasa pemrograman. Bahasa pemrograman ini akan memanjang dan dapat menentukan secara persis data apa saja yang akan diolah oleh komputer. Adapun bahasa pemrograman yang dipakai sebagai berikut:

1. *JavaScript*

Menurut Setiawan (2017:194) “*Java Script* adalah bahasa *scripting* yang populer disebagian besar *Browser*. *JavaScript* disisipkan pada halaman *web* menggunakan tag `<script>`”. Kegunaan *JavaScript* yaitu untuk menambah interaktif suatu *website*

Beberapa hal tentang *JavaScript*:

- a. Open source(semua orang dapat menggunakan secara gratis)
- b. *JavaScript* merupakan bahasa *scripting* yang ringan

- c. *JavaScript embedded* (disisipkan) dalam html
- d. Dalam *JavaScript*, *script* akan langsung dieksekusi tanpa kompilasi

2. PHP

Menurut Setiawan (2017:54) “PHP sendiri sebenarnya merupakan singkatan dari *Hypertext Preprocessor*, yang merupakan sebuah bahasa *scripting* tingkat tinggi yang dipasang pada dokumen HTML.”

Sebagian besar sintaks dalam PHP mirip dengan bahasa C, *Java* dan *Perl*, namun pada PHP ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancang *web* yang dinamis dan dapat bekerja secara otomatis.

3. HTML

Menurut Setiawan (2017:4) “HTML atau *Hyper Text Markup Language* merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *Web Browser*”. HTML sendiri secara resmi lahir pada tahun 1989 oleh Tim Berners Lee dan dikembangkan oleh *World Wide Web Consortium*(W3C), yang kemudian pada tahun 2004 dibentuklah *Web Hypertext Application technology Working Group* (WHATG) yang hingga kini telah mengembangkan HTML 5, sebuah versi terbaru dari HTML yang mendukung tidak hanya gambar dan teks, namun juga menu interaktif, audio, video dan lain sebagainya.

4. JQuery

Menurut Setiawan (2017:210) “*Jquery* adalah kumpulan fungsi-fungsi *JavaScript* yang memudahkan penulisan kode *JavaScript*”.

Jquery dikembangkan pertama kali oleh John Resig di tahun 2006. Sejak saat itu, *Jquery* berkembang menjadi proyek *opensource* dan menjadi *library JavaScript* paling populer di dunia. Situs *Jquery* beralamat di jquery.com.

Fitur lain dari *Jquery* (dan mungkin menjadi alasan paling penting), yaitu menyediakan berbagai efek animasi menarik. Beberapa diantaranya sangat mudah digunakan dan hanya membutuhkan 1 atau 2 baris kode program. Khusus untuk membuat efek animasi ini, *jQuery* juga memiliki *jQuery UI* yang fokus ke animasi dan interaksi *user*. *jQuery* juga memiliki berbagai plugin atau kode tambahan untuk fungsi-fungsi lanjutan, seperti membuat efek *slideshow*, *image carousel*, hingga *jQuery mobile*. Efek-efek seperti ini dibuat berdasarkan kode dasar *jQuery*.

5. CSS

Menurut Setiawan (2017:116) “CSS adalah kependekan dari *Cascading Style Sheet* merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman *web* supaya lebih elegan dan menarik”.

CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Awalnya CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa *markup* seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (*XML Interface Language*). CSS digunakan oleh *web programmer* dan juga *web designer* untuk menentukan warna, tata letak *font*, dan semua aspek lain dari presentasi dokumen di situs mereka. Saat ini, hampir tidak ada situs *web* yang dibangun tanpa kode CSS.

D. Basis Data

Menurut Anhar dalam (Kurniadi, 2016) “Database atau basis data adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom”.

Struktur file yang menyusun sebuah database adalah data record dan field. Database sering digunakan untuk melakukan proses terhadap data-data tersebut dan menghasilkan informasi tertentu. Tujuan utama pembuatan database adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan, diubah, dihapus dan dibaca dengan mudah dan cepat. Saat ini banyak tersedia perangkat lunak yang ditujukan untuk mengelola database. Perangkat seperti itu biasa dinamakan DBMS (Database Management System). MySQL merupakan software yang tergolong database server dan bersifat open source. MySQL juga bersifat multiplatform yaitu dapat dijalankan pada berbagai sistem operasi.

Menurut Yanto (2016:10) :

“Basis data terdiri dari dua kata, yaitu basis dan data. Basis data dapat diartikan sebagai markas, gudang, tempat berkumpul. Sedangkan data adalah fakta yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, keadaan dan sebagainya, yang direkam dalam bentuk, angka, huruf, simbol, teks gambar atau kombinasinya.”

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan untuk memenuhi
3. Kumpulan *file* yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data tidak sekedar penyimpanan secara elektronik, karena :

1. Pada penyimpanan dokumen berisi data dan *file* teks, *file Spreat sheet*, tidak ada pemilihan dan pengelompokan data sesuai jenis atau fungsi sehingga akan menyulitkan pencarian data.
2. Keutamaan Basis data adalah pengeturan, pemilihan, pengelompokan, pengoprasian data yang akan disimpan sesuai fungsi dan jenisnya.

E. MySQL

Menurut Nugroho (2013:26) “MySQL adalah *software* atau program *database server*.”

Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan(*query*) dalam *database server* termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam *software database server* lain, seperti SQL server, Oracle, PostgreSQL dan lainnya.

Menurut Kadir (2014:15) “SQL(*Structured Query Language*) adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional. Sesungguhnya SQL tidak terbatas hanya untuk mengambil data(*query*), tetapi juga dapat dipakai untuk menciptakan tabel(*CREATE*), menghapus tabel(*DELETE*), menambahkan data ke tabel(*INSERT*), menghapus data di tabel(*DROP*), mengganti data di tabel(*UPDATE*), dan berbagai operasi lain.

MySQL merupakan *database* yang dikembangkan dari bahasa SQL (*Structural Query Language*). SQL merupakan bahasa terstruktur yang digunakan untuk interaksi antara *script* program dengan *database server* dalam hal pengolahan

data. Sebuah model SQL terdiri dari tabel, baris, *field* dan elemen. MySQL merupakan sistem manajemen *database* yang bersifat open source atau gratis.

Keunggulan yang dapat kita peroleh dari MySQL antara lain:

1. Kecepatan.
2. Kemudahan user dalam penggunaannya.
3. Support dalam bahasa query.
4. User dapat mengakses lebih dari satu dalam satu waktu.
5. Akses dapat dilakukan diberbagai tempat dengan fasilitas internet.
6. MySQL mudah didapatkan karena *source kode* yang dapat disebarluaskan.

Berdasarkan pendapat yang dikemukakan dapat ditarik kesimpulan bahwa MySQL adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*.

F. XAMPP

Menurut Nugroho (2013:1) “XAMPP adalah paket program *web* lengkap yang dapat anda pakai untuk belajar pemrograman *web*, khususnya PHP dan MySQL”.

Didalam folder utama Xampp terdapat beberapa folder penting yang perlu diketahui. Untuk memahami setiap fungsinya, dapat dilihat penjelasannya sebagai berikut:

Table II.1.
Folder Penting Xampp

Folder	Keterangan
<i>Apache</i>	Folder utama dari <i>Apache Webserver</i>
<i>Htdocs</i>	Folder utama untuk menyimpan data-data latihan <i>web</i> , baik <i>PHP</i> maupun <i>HTML</i> biasa.
<i>Manual</i>	Berisi <i>subfolder</i> yang di dalamnya terdapat manual program dan <i>database</i> , termasuk manual <i>PHP</i> dan <i>MySQL</i> .
<i>MySQL</i>	Folder utama untuk <i>database MySQL Server</i> .
<i>PHP</i>	Folder utama untuk program <i>PHP</i> .

Sumber : Nugroho (2013:7)

G. Adobe Dreamweaver CS6

Menurut Bekti (2015:3) “*Adobe Dreamweaver* adalah sebuah aplikasi yang dipergunakan untuk mendisain *website* keluaran *Adobe System* yang dulu dikenal dengan sebagai *Macromedia Dreamweaver* keluaran *Macromedia*”.

Program ini banyak digunakan oleh pengembang *web* karena fitur-fiturnya yang menarik dan kemudahan penggunaanya.

Menurut Mandar (2017:1)“*Adobe Dreamweaver* merupakan aplikasi pengembang yang berfungsi untuk mendesain *web* yang dibuat, dikembangkan, dan diproduksi oleh *Adobe System*”. Sampai saat versi terbaru dari *Adobe Dreamweaver* adalah *Adobe Dreamweaver CS6* yang dirilis pada tanggal 21 April 2012 dengan harga \$399 USD.

Aplikasi pengembang *web* saat ini digemari oleh *web* desainer dalam merancang *web* sebab perangkat lunak komputer ini memiliki kelebihan dan kemudahan

dalam penggunaannya. Dengan menggunakan aplikasi ini, pengembangan *web* dapat dilakukan secara visual sehingga hasil perancangan *web* dapat langsung terlihat tanpa harus menggunakan aplikasi bantu peramban seperti *Google Chrome*, *Firefox*, atau *Internet Explore*. Teknologi *web* yang didukung oleh *Adobe Dreamweaver* sangat beragam, salah satunya adalah teknologi untuk kebutuhan pengembangan *web* berbasis *mobile*.

H. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *waterfall* Sukamto & Salahuddin (2013:28) yang terbagi menjadi lima tahapan, yaitu:



Sumber : Sukamto & Salahuddin (2013:29)

Gambar II.1
Ilustrasi Model Waterfall

1. Analisa kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memespesifikasikan 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 disain yang telah dibuat pada tahap disain.

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 perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa teradaptasi dengan lingkungan baru. Tahan pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Pendukung (*Tools Program*)

Teori Pendukung (*Tools Program*) merupakan teori yang digunakan untuk membantu mempermudah dalam pembuatan, pembacaan logika dan algoritma program, serta membantu untuk mengetahui alur program yang dibuat dari mulai masukan, proses dan keluaran yang dihasilkan. Adapun teori pendukung yang digunakan penulis, antara lain:

A. Struktur Navigasi

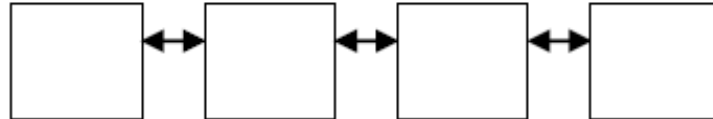
Menurut Prihatna dalam (Riyadi, Arkhadius, & Yoraeni, n.d.) “Struktur Navigasi adalah Susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan link atau navigasi tiap halaman pada suatu web”.

Ada empat macam bentuk dasar dari peta navigasi yang biasa digunakan dalam proses pembuatan aplikasi, yaitu:

1. Struktur Navigasi *Linear*

Struktur navigasi *linear* merupakan struktur yang hanya mempunyai satu rangkaian cerita yang berurut untuk menampilkan satu demi satu tampilan

layer secara berurut menurut urutannya dan tidak diperbolehkan adanya percabangan. Tampilan yang dapat ditampilkan adalah salah satu halaman sebelumnya atau salah satu halaman sesudahnya.

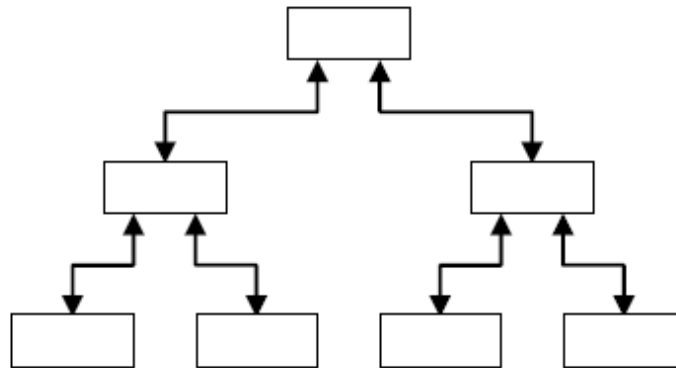


Sumber : (Riyadi et al., n.d.)

Gambar II.2
Struktur Navigasi Linear

2. Struktur navigasi *Hirarki*

Struktur navigasi *hirarki* merupakan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama kesatu), halaman utama ini mempunyai halaman percabangan yang dikatakan *slave page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya yang terpenting dari struktur penjejakan ini tidak diperkenankan adanya tampilan secara *linear*.

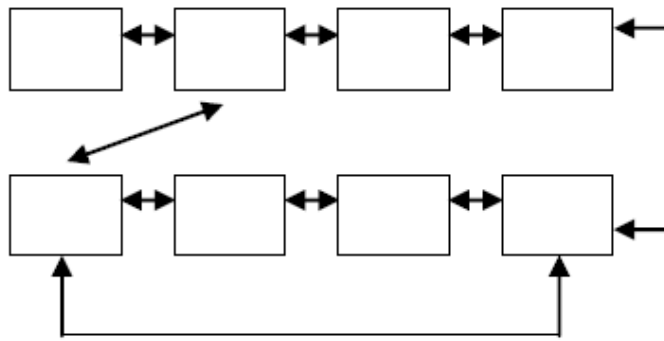


Sumber : (Riyadi et al., n.d.)

Gambar II.3
Struktur Navigasi Hirarki

3. Struktur Navigasi Non Linier

Struktur navigasi non linear merupakan pengembangan dari struktur linier. Pada struktur ini diperkenankan membuat penjejakan bercabang. Pemakai bebas menelusuri *website* tanpa dibatasi oleh suatu rute. Percabangan yang dibuat pada struktur non linier ini berbeda dengan percabangan hirarki, karena pada percabangan non linier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama tidak ada *Master Page* dan *Slave Page*.

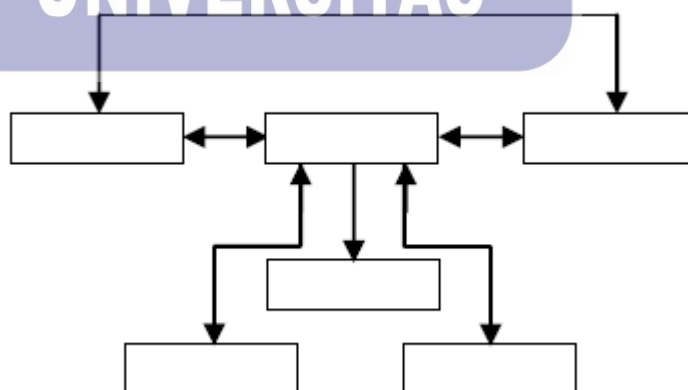


Sumber : (Riyadi et al., n.d.)

Gambar II.4
Struktur Navigasi Non Linear

4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi campuran (*composite*) merupakan gabungan dari ketiga struktur sebelumnya. Struktur navigasi ini paling banyak digunakan dalam aplikasi multimedia sebab dapat memberikan keinteraksian yang lebih tinggi.



Sumber : (Riyadi et al., n.d.)

Gambar II.5
Struktur Navigasi Campuran

B. *Entity Relationship Diagram (ERD)*

Menurut Yuhefizar dalam (Sagita & Sugiarto, 2016) “Diagram E-R digunakan untuk menggambarkan secara sistematis hubungan antara entity-entity yang ada dalam satu sistem database menggunakan simbol-simbol sehingga mudah dipahami”.

Menurut Sukamto & Salahuddin (2013:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data dan relasional.” ERD merupakan singkatan dari *Entity Relationship Diagram*, dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS(*Object Oriented Database Management System*) maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi *Chen* (dikembangkan oleh Peter Chen), *Barker* (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Adapun Komponen-komponen *ERD* adalah sebagai berikut:

1. Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. sederhananya entitas menunjuk pada individu suatu objek, sedang himpunan entitas menunjuk pada rumpun (*family*) dari individu tersebut.

Dilihat dari jenisnya entitas terbagi atas dua yaitu :

a. Entitas Kuat (*Strong Entity*)

Entitas kuat adalah entitas yang dapat berdiri sendiri tidak bergantung pada entitas lainnya, entitas kuat memiliki *atribut key* dan entitas kuat digambarkan sebagai kotak persegi panjang bergaris tunggal. Contoh entitas kuat adalah entitas pegawai.

b. Entitas Lemah (*Weak Entity*)

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri. Entitas lemah merupakan hasil dari pembentukan entitas kuat, entitas lemah tidak memiliki *atribut key* dan entitas lemah digambarkan sebagai kotak persegi panjang bergaris ganda. Jika entitas kuat yang membentuk entitas lemah dihapus maka secara otomatis entitas lemah akan terhapus. Contoh entitas lemah adalah entitas pegawai kontrak, pegawai tetap.

2. Atribut

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*Property*) dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal yang penting lainnya dalam pembentukan model data.

3. Relasi

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

4. Garis

Sebagai penghubung antara himpunan relasi dengan himpunan entitas dengan himpunan atributnya.

Adapaun Kardinalitas Relasi pada ERD adalah sebagai berikut:

a. Satu ke satu (*One to One*)

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

b. Satu ke banyak (*One to Many*)

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

c. Banyak ke banyak (*Many to Many*)

Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak 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.

C. *Logical Record Structure (LRS)*

Menurut Frieyadie dalam (Supriyanta & Tina, 2017) “LRS merupakan suatu bentuk relational model yang dibuat secara logic atau external level dan terkonsep sebelum tabel terbentuk dari field atau atribut entitas secara fisik atau internal level”.

Logical Record Structure dibentuk dengan nomer dari tipe record. Beberapa tipe record digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe record berada diluar kotak *field* tipe record ditempatkan.

Logical Record Structure terdiri dari *Link-Link* diantara tipe record,. *Link* ini menunjukan arah dari satu tipe record lainnya. Banyak *Link* dari LRS yang diberi tanda *field-field* yang keliatan pada kedua *Link* tipe record. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonfersikan ke LRS, metode yang lain dimulai dengan E-R diagram dan langsung dikonversikan ke LRS.

D. Implementasi dan Pengujian Web

Sebuah perangkat lunak perlu dijaga kualitasnya bahwa bergantung pada kepuasan pelanggan(*customer*). Sering perangkat lunak mengandung kesalahan atau error pada proses-proses tertentu pada saat perangkat lunak sudah berada pengujian perangkat lunak sebelum perangkat lunak diberikan pelanggan atau selama perangkat lunak masih terus dikembangkan.

“*Black Box Testing* (pengujian kotak hitam) adalah 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 lunak 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.

Pengujian *unit* fokus pada usaha verifikasi pada *unit* yang terkecil pada desain perangkat lunak(komponen atau modul perangkat lunak). Setiap *unit* perangkat lunak diuji agar dapat diperiksa apakah aliran masukan(input) dan keluaran(output) dari *unit* sudah sesuai dengan yang diinginkan. Pengujian *unit* biasanya dilakukan saat kode program dibuat. Karena dalam sebuah perangkat lunak banyak memiliki *unit-unit* kecil maka untuk menguji *unit-unit* kecil ini biasanya dibuat program kecil(main program) untuk menguji perangkat lunak.

Unit disini secara fiksi dapat berupa prosedur atau fungsi, sekumpulan prosedur atau fungsi yang ada dalam satu berkas(file) jika dalam pemrograman terstruktur, atau kelas, bisa juga kumpulan kelas dalam satu package dalam pemrograman berorientasi objek. Setiap *unit* diuji menggunakan sebuah program pengujian yang khusus dibuat untuk menguji sebuah *unit* menggunakan kumpulan kasus uji yang didefinisikan.



UNIVERSITAS