

BAB II

LANDASAN TEORI

2.1. Konsep Dasar *Web*

Pada bab ini dijelaskan beberapa konsep dan dasar teori yang berkaitan dengan permasalahan yang akan dibahas penulis sebagai pemahaman dalam mengimplementasikan konsep-konsep tersebut semua kegiatan perancangan *website* ini dan apa saja yang digunakan dalam pembahasan tersebut yaitu:

2.1.1 *Website*

Menurut (Muhammad, 2014) menyimpulkan bahwa:

Website merupakan halaman yang menampilkan informasi data *teks*, gambar, suara, video atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis. Halaman pertama sebuah *website* disebut *homepage*, sedangkan halaman demi halaman sendiri disebut *webpage*.

Website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi. Secara garis besar *website* bisa digolongkan menjadi tiga jenis yaitu:

1. *Website* statis adalah *website* yang berisi informasi-informasi yang bersifat statis (tetap) sehingga informasi didalamnya tidak *up to date*.
2. *Website* dinamis adalah *website* yang menampilkan informasi yang bersifat dinamis (berubah-ubah) dan dapat berinteraksi dengan *user*. *Website* dinamis biasanya dilengkapi dengan animasi gambar yang membuat tampilan lebih menarik dan juga berinteraksi dengan *database*.

3. *Website* interaktif adalah *website* yang saat ini memang sedang *booming*, contoh *website* interaktif adalah *blog* dan *forum*.

Dalam sebuah *website* ada beberapa komponen penghubung agar dapat diakses oleh pemakai, diantaranya:

1. *Internet*

Menurut (Supriyanta, 2015) menyimpulkan bahwa:

Internet adalah jaringan komunikasi global yang terbuka dan menghubungkan ribuan jaringan komputer, melalui sambungan telepon umum maupun pribadi (pemerintah dan swasta). Secara individual, jaringan komponennya dikelola oleh agen-agen pemerintah, universitas, organisasi komersial maupun sukarelawan.

Secara teknis, *internet* merupakan dua komputer atau lebih yang saling berhubungan membentuk suatu jaringan komputer hingga meliputi jutaan komputer di dunia, yang saling berinteraksi dan bertukar informasi. Dilihat dari segi komunikasi, *internet* merupakan sarana yang efisien dan efektif untuk melakukan pertukaran informasi jarak jauh maupun didalam lingkungan perkantoran.

Dari beberapa definisi diatas, dapat disimpulkan bahwa pengertian *internet* adalah sebuah jaringan komputer yang menghubungkan jutaan komputer yang tersebar diseluruh dunia.

2. *WebServer*

Menurut (Yuhdo & Prasetyo, 2018) menyimpulkan bahwa "*Webserver*" adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser*, kemudian memberikan tanggapan permintaan dalam bentuk halaman situs *web* atau lebih umumnya dalam dokumen *HTML*. namun, *webserver* dapat mempunyai dua pengertian berbeda, yaitu sebagai bagian dari perangkat keras (*hardware*), maupun sebagai bagian dari perangkat lunak (*software*).

Jika merujuk pada *hardware*, *webserver* digunakan untuk menyimpan semua data seperti *HTML* dokumen, gambar *file*, *CSS stylesheets*, dan *file javascript*. sedangkan pada sisi *software*, fungsi *webserver* adalah sebagai pusat kontrol untuk memproses permintaan yang diterima dari *browser*. Jadi, sebenarnya semua proses yang berhubungan dengan *website* biasanya juga berhubungan dengan *webserver* karena tugas *webserver* adalah mengatur semua komunikasi yang terjadi antara *browser* dengan *server* untuk memproses sebuah *website*.”

3. *Web browser*

Menurut (Edy, et al, 2015) menyimpulkan bahwa ”*Web browser* adalah perangkat lunak yang fungsinya menerima dan menyajikan informasi diinternet. Sumber informasi bisa dilihat dengan menggunakan *web browser*, baik yang berupa halaman *web*, gambar, video atau jenis konten lainnya.”

Walaupun tujuan utamanya mengakses konten diinternet, sebuah *browser* sebenarnya juga dapat dipakai untuk mengakses informasi yang disediakan oleh *web server* jaringan pribadi atau untuk melihat struktur *folder* disistem komputer. Contoh *web browser* yang populer sekarang ini adalah *Google chrome*, *Mozilla Firefox*, *Internet Explorer*, *Opera* dan *Safari*.

2.1.2 Bahasa Pemrograman

Menurut (Abdul, 2017) menyimpulkan bahwa:

Bahasa Pemrograman adalah proses untuk menyelesaikan masalah-masalah dalam bentuk langkah-langkah penyelesaian yang dapat dikerjakan oleh komputer (yang disebut algoritma) hingga ke penerjemahan kode dalam suatu bahasa pemrograman, sehingga masalah tersebut benar-benar dieksekusi oleh komputer.

Berikut ini adalah contoh bahasa pemrograman yang digunakan:

1. *PHP HyperText Preprocessor (PHP)*

Menurut (Yuhdo & Prasetyo, 2018) menyimpulkan bahwa: *PHP* atau *HyperText Preprocessor* adalah bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web. Bahasa pemrograman ini memang dirancang untuk para pengembang *web* agar dapat menciptakan suatu halaman *web* yang bersifat dinamis.

2. *HyperText Markup Language (HTML)*

Menurut (Yuhdo & Prasetyo, 2018) menyimpulkan bahwa "*HTML* atau sering disebut dengan *HyperText Markup Language*, awalnya dulu pernah disebut sebagai pemrograman. dari kepanjangannya, bias kita simpulkan bahwa *HTML* adalah sebuah penkita atau mark. Jadi, kita luruskan mindset kita dulu bahwa *HTML* bukan bahasa pemrograman, tetapi sebuah penkita".

3. *CSS (Cascading Style Sheet)*

Menurut (Suryana Taryana & Koesharyatin, 2014) menyimpulkan bahwa: *CSS (Cascading Style Sheet)* adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan. pada umumnya *CSS* digunakan untuk memformat halaman *web* yang ditulis dengan *HTML* atau *XHTML*.

Ada dua cara yang bisa diterapkan untuk menggunakan *CSS* pada *web*. Cara yang pertama dengan membuat *CSS* langsung didalam satu *file HTML (internal/inline style sheet)*. Cara yang kedua dengan memanggil *CSS* tersebut dari *file CSS* tersendiri (*eksternal*).

Cara pemakaian *CSS* ada dua cara. Cara yang pertama dengan menghubungkan *CSS* langsung ke dalam satu *file markus (internal)*. Cara yang kedua memanggil *CSS* tersebut (*eksternal*).

4. *JavaScript*

Menurut (Yuhdo & Prasetyo, 2018) menyimpulkan bahwa “*Java script* atau sering disingkat dengan *JS*. *JS* digunakan untuk membuat *website* kita lebih interaktif kepada *user* yang menggunakan *website* tersebut. Misal, kita ingin mengubah suatu nilai pada *HTML* dengan mengklik suatu tombol, kita bias pakai *java script* ini.”

5. *Jquery*

Menurut (Yuniar, 2018) menyimpulkan bahwa:

JQuery merupakan *library* (kumpulan program yang dipakai oleh pengembangan program) yang menggunakan *interface javascript* yang dapat menyederhanakan penelusuran dokumen, penanganan kejadian (*event*), animasi dan interaksi dengan *AJAX*. Pada awal pengembangannya, *jquery* ditujukan untuk kebutuhan *web desktop*, tetapi seiring dengan berkembangnya *web mobile* maka dibuat atau disesuaikan untuk kebutuhan *web mobile* melalui *jquery mobile*.

2.1.3 Basis Data

Menurut James Martin dalam (Subandi & Syahidi, 2018) menyimpulkan bahwa:

Basis data (*database*) merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada satu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Berikut ini adalah yang berkaitan dengan basis data diantaranya:

1. *MySQL*

Menurut (Yuniar, 2015) menyimpulkan bahwa:

MySQL adalah perangkat lunak *database server* atau kita sebut saja *database smart*. *Database* ini semakin lama semakin populer. Dengan menggunakan *database* ini, data semakin aman dan berdaya guna. *Database* ini juga banyak dipakai pada *web database* sehingga data kita mungkin terintegrasi antara *database desktop* dengan *database web*. Untuk menggunakan *database MySQL* kita harus menginstalasinya dahulu ke komputer kita. Karena kita

menggunakan *interface* atau tampilan program *visual basic*, maka kita perlu menginstalasi *driver ODBC* untuk *MySQL*.

2. *PhpMyAdmin*

Menurut (M.Hilmi & Java, 2015) menyimpulkan bahwa:

PHPMYAdmin adalah sebuah aplikasi perangkat lunak bebas (*open source*) yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk mengenai administrasi *database MySQL* melalui jaringan lokal maupun internet. *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, bidang (*field*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain.”

3. *Xampp*

Menurut (Aryanto, 2016) menyimpulkan bahwa “XAMPP adalah sebuah aplikasi perangkat lunak pemrograman dan *database* yang didalamnya terdapat berbagai macam aplikasi pemrograman seperti: *Apache HTTP Server*, *MySQL database*, bahasa pemrograman *PHP* dan *Pearl*.”

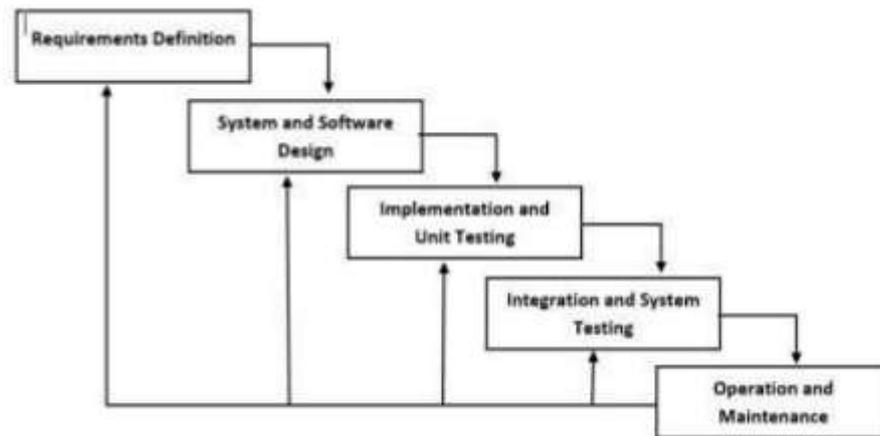
4. *Sublime Text*

Menurut (Ricky, 2017) menyimpulkan bahwa:

Sublime Text merupakan *text editor* kedua yang juga sering digunakan. Selain digunakan untuk belajar, *sublime text* terkadang digunakan untuk penulisan *coding theme*. Sama seperti *Notepad++*, *sublime text* juga memiliki ragam fitur pendukung untuk para *programmer*. *Text editor* ini tersedia untuk sistem operasi *windows*, *mac* dan *linux*.

2.1.4 Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *Waterfall*. Menurut Sommerville dalam (Kusno, et al, 2019) Model ini adalah klasik yang bersifat sistematis berurutan dalam membangun software yang terbagi menjadi lima tahapan, yaitu:



Sumber: (Kusno et al., 2019)

Gambar II.1. Model Waterfall

1. *Requirements Definition*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan *desain* yang lengkap.

2. *System Software Design*

Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

3. *Implementation and Unit Testing*

Desain program diterjemahkan kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

4. *Integration and System Testing*

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*System Testing*).

5. *Operation and Maintenance*

Mengoperasikan program-program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi yang sebenarnya.

2.2. *Tools Program*

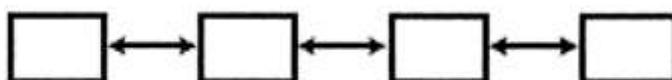
Dalam penulisan Tugas ini penulis menggunakan beberapa teori pendukung yang digunakan yaitu:

2.2.1 Struktur Navigasi

Menurut (Iwan, 2016) menjelaskan struktur navigasi dapat diartikan sebagai alur suatu program yang menggambarkan rancangan hubungan antar area yang berbeda sehingga memudahkan proses pengorganisasian seluruh elemen *website*.

1. Linier

Pengguna akan melakukan navigasi secara berurutan, dari frame atau byte informasi yang satu ke yang lainnya, seperti yang terlihat pada gambar.

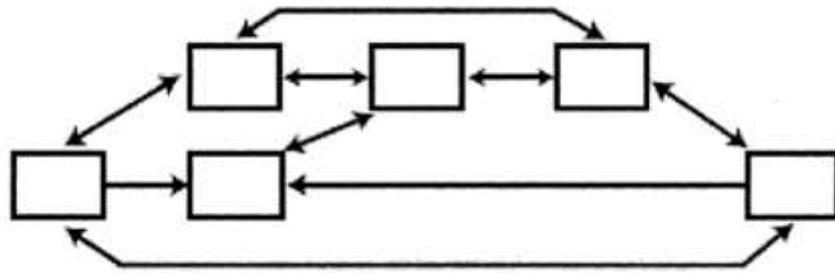


Sumber: (Iwan, 2016)

Gambar II.2
Struktur Navigasi Linier

2. Non-linier

Pengguna akan melakukan navigasi dengan bebas melalui si proyek dengan tidak terikat dengan jalur yang sudah ditentukan sebelumnya, seperti yang terlihat pada gambar.

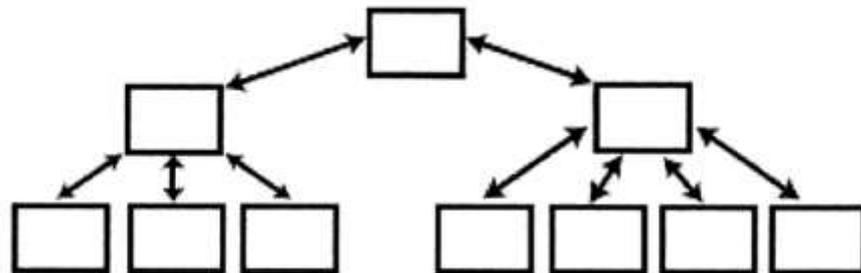


Sumber: (Iwan, 2016)

Gambar II.3
Struktur Navigasi Non-Linier

3. Hirarki

Struktur dasar ini disebut juga struktur linier dengan percabangan karena melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi, seperti yang terlihat pada gambar tersebut.

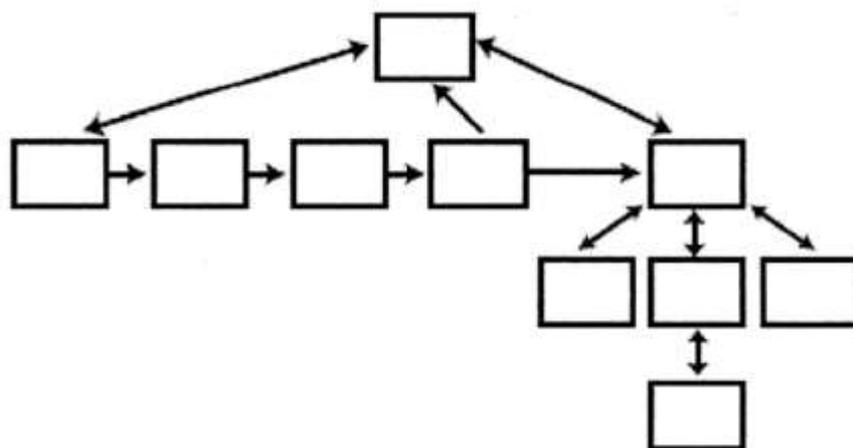


Sumber: (Iwan, 2016)

Gambar II.4
Struktur Navigasi Hirarki

4. Campuran

Pengguna akan melakukan navigasi dengan bebas (secara nonlinier), tetapi terkadang dibatasi presentasi linier film atau informasi penting pada data yang paling terorganisasi secara logis pada suatu hirarki, seperti yang terlihat pada gambar.



Sumber: (Iwan, 2016)

GambarII.5
Stuktur Navigasi Composite

2.2.2 *Entity Relationship Diagram (ERD)*

Menurut Sukanto Ariani Rosa dan M. Salahuddin dalam (Subandi & Syahidi, 2018) mengutarakan bahwa “pemodelan awal basis data paling banyak digunakan adalah dengan *Entity Relationship Diagram (ERD)*. *ERD* dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *Object Oriented Database Management Systems (OODMS)* maka perancangan basis data tidak perlu menggunakan *ERD*”.

ERD memiliki beberapa aliran notasi seperti notasi *Chen*, *Barker*, *Crow’s Foot* dan beberapa notasi lainnya. namun paling banyak yang digunakan adalah *Chen*. *Chen* mengajukan *ERD* yang merupakan *tool* analisis sistem pertama yang memusatkan pada data dan keterkaitan antar data serta pengorganisasian data. Tujuan *ERD* adalah untuk menunjukkan objek data dan *relationship* yang ada pada objek tersebut. Disamping itu *ERD* ini merupakan salah satu alat untuk perancangan dalam basis data.

1. Komponen ERD

Komponen-komponen *Entity Relationship Diagram* (ERD) dengan notasi chen.

a. Entitas (*entity*)

Entitas merupakan data inti yang disimpan, sebagai bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas merupakan suatu objek yang dapat dibedakan atau dapat diidentifikasi secara unik dengan objek yang lainnya, dimana semua informasi yang berkaitan dengannya dikumpulkan.

b. Atribut (*Atributte*)

Karakteristik dari entitas atau dari relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Atribut sebagai field atau kolom data yang perlu disimpan dalam suatu entitas.

c. Atribut kunci primer

Sebagai *field* atau kolom data yang perlu disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* (*Primary Key*) yang diinginkan biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

d. Atribut multivalai/*multivalue*

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memilikinilai lebih dari satue. Relasi-Relasi yang menghubungkan antar entitas biasanya diawali dengan kata kerja.

2. Kardinalitas

Untuk menjelaskan jumlah *entity* yang berpartisipasi dalam suatu relasi, kita dapat menggunakan derajat. Dengan menggunakan derajat *relation* pengguna dapat menentukan hubungan antar entitas yang telah dibuat. Adapun gambar kardinalitas atau derajat relasi menurut Subandy (2018) adalah sebagai berikut:

a. Satu ke Satu (*one to one*)

Perbandingan antara entitas pertama dengan entitas kedua berbanding satu berbanding satu.



Sumber : (Subandi & Syahidi, 2018)

Gambar II.6.
Derajat Relasi *One to One*

a. Satu ke Banyak (*one to many*)

Perbandingan antara entitas pertama dengan entitas kedua berbanding satu berbanding banyak.



Sumber : (Subandi & Syahidi, 2018)

Gambar II.7.
Derajat Relasi *One to Many*

b. Banyak ke Satu (*many to one*)

Perbandingan antara entitas pertama dengan entitas kedua berbanding banyak berbanding satu.



Sumber : (Subandi & Syahidi, 2018)

Gambar II.8.

Derajat Relasi *Many to One*

- c. Banyak ke Banyak (*many to many*)

Perbandingan antara entitas pertama dengan entitas kedua berbanding banyak berbanding banyak, jarang terjadi untuk hubungan M:M ini.



Sumber : (Subandi & Syahidi, 2018)

Gambar II.9.

Derajat Relasi *Many to Many*

3. Derajat Relasi (*Relationship Degree*)

- a. Binary Degree (Derajat Dua)

Satu buah relasi menghubungkan dua buah entitas.

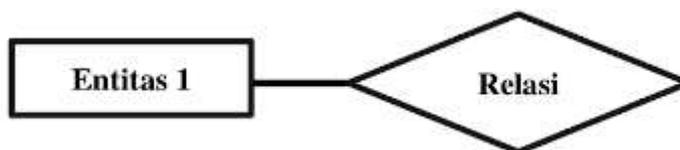


Sumber: (Subandi & Syahidi, 2018)

Gambar II.10.
Binary degree

b. Unary Degree (Derajat Satu)

Satu buah relasi menghubungkan satu buah entitas

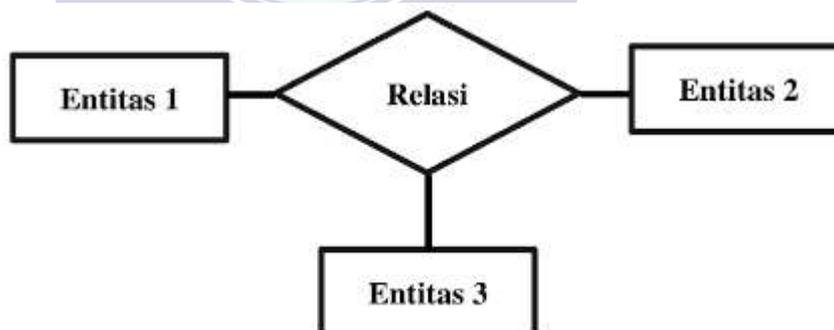


Sumber: (Subandi & Syahidi, 2018)

Gambar II.11.
Unary degree

c. Ternary Degree

Satu buah relasi menghubungkan tiga buah entitas



Sumber: (Subandi & Syahidi, 2018)

Gambar II.12.
Ternary degree

4. Macam-macam kunci

Kunci-kunci yang digunakan memiliki fungsi masing-masing dalam mengidentifikasi yang berbeda-beda. Berikut ini macam-macam kunci pada sebuah model Relasional:

a. Kunci Kandidat (*Candidate Key*)

Kunci Kandidat adalah suatu kunci yang mengidentifikasi secara unik suatu spesifik dari *entity*. Jika suatu kunci kandidat berisi lebih dari suatu atribut, disebut dengan *composite key* atau kunci gabungan. Setiap kunci kandidat berpeluang menjadi kunci utama.

b. Kunci Utama (*Primary Key*)

Kunci Utama adalah suatu kunci yang mengidentifikasi secara unik dari suatu kejadian spesifik dan mewakili setiap kejadian dari suatu *entity*. Kunci utama bersifat unik dan nilai dari kunci utama tidak sama antara nilai yang satu dengan nilai yang lainnya. Akan tetapi, kunci gabungan dapat juga dijadikan *Primary Key*.

c. Kunci Alternatif (*Alternate Key*)

Kunci Alternatif adalah suatu kunci kandidat yang tidak digunakan sebagai kunci utama. Kunci Alternatif ini biasa digunakan untuk pencarian suatu data atau sebagai kunci pengurutan suatu data Kunci.

d. Kunci Tamu (*Foreign Key*)

Kunci Tamu adalah suatu kunci yang ditempatkan pada tabel lain yang berguna untuk hubungan ke tabel induk.

2.2.3 Logical Record Structure (LRS)

Menurut (Muhamad, 2014) menyimpulkan bahwa “*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 dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonverensikan ke *LRS*, metode yang lain dimulai dengan *ER-diagram* dan langsung konversikan ke *LRS*”.

2.2.4 Pengujian Program (*Black Box Testing*)

Menurut Rizky dalam (Uus, 2019) menyimpulkan bahwa: *Black Box Testing* adalah tipe testing yang memerlukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “Kotak Hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses testing dibagian luar”.

Beberapa keuntungan yang diperoleh dari jenis testing ini antara lain:

1. Anggota tim *tester* tidak harus dari seseorang yang dimiliki kemampuan teknis dibidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun seringkali ditemukan oleh komponen tester yang berasal dari pengguna.
3. Hasil dari *blackbox testing* dapat memperjelaskan kontradiksi ataupun kerancuan yang mungkin ditimbulkan dari eksekusi perangkat lunak.
4. Proses testing dapat dilakukan lebih cepat dibandingkan *whitebox testing*.