

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen–komponen sistem dapat berupa suatu subsistem atau bagian–bagian dari sistem meliputi masukan (*input*), pengolahan (*process*), dan keluaran (*output*).

##### **2.1.1. Pengertian Sistem**

Menurut (Hutahaean, 2014:2) mengemukakan bahwa “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berkaitan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Menurut (Fauzi, 2017:2) mengemukakan bahwa, “Sistem adalah suatu kesatuan yang terdiri dari interaksi subsistem untuk mencapai tujuan yang sama”.

Sedangkan menurut Dr. Ir. Harijono Djodjodihardjo dalam (Hutahaean, 2014:2) menyatakan bahwa “Suatu sistem adalah sekumpulan objek yang mencakup hubungan fungsional antara tiap-tiap objek dan hubungan antara ciri tiap objek, dan yang secara keseluruhan merupakan suatu kesatuan secara fungsional”.

Berdasarkan pendapat yang dikemukakan diatas dapat disimpulkan bahwa, Sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu.

### 2.1.2. Pengertian Informasi

Menurut (Tyoso, 2016:21), “Informasi adalah suatu penambahan dalam ilmu pengetahuan yang menyumbangkan kepada konsep kerangka kerja yang umum dan fakta-fakta yang diketahui”.

Menurut (Hutahaean, 2014:10), “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Dari beberapa pendapat diatas dapat disimpulkan bahwa, Informasi adalah data yang diolah agar bermanfaat dalam pengambil keputusan bagi penggunanya.

### 2.1.3. Sistem Informasi

Menurut (Hutahaean, 2014:15) mengemukakan bahwa, “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Sistem Informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok Masukkan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. Blok Model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.

3. Blok Keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem

4. Blok Teknologi (*technology block*)

Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsur utama :

- a. Teknisi (*human ware* atau *brain ware*)
- b. Perangkat lunak (*software*)
- c. Perangkat keras (*hardware*)

5. Blok Basis Data (*database block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan, ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan

bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

#### 2.1.4. Pengolahan Data

Menurut (Lubis, 2016:1), “Data adalah fakta-fakta yang menggambarkan suatu kejadian yang sebenarnya pada waktu tertentu”. Menurut George R. Terry Ph.D dalam (Hutahaean, 2014:9) menyatakan bahwa “Pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan”.

8 unsur pokok pengolahan data, yaitu :

- a. Membaca
- b. Menulis, Mengetik
- c. Mencatat atau mencetak
- d. Menyortir
- e. Menyampaikan atau memindahkan
- f. Menghitung
- g. Membandingkan
- h. Menyimpan



#### 2.1.5. Website

Menurut (Abdulloh, 2016:1), “Website atau disingkat *web* dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.”

### 1. *Web Server*

Menurut Fathansyah dalam (Prayitno & Safitri, 2015) menerangkan bahwa “*Server Web (Web Server)* merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.

### 2. *Web Browser*

Menurut (Husda & Wangdra, 2016) menyimpulkan bahwa : *Web Browser* atau diringkas menjadi *browser* adalah program atau *software* yang dirancang untuk mencari dan menampilkan dokumen *web* dalam format HTML. Dengan *Browser*, para pengguna komputer dapat mencari dan menelusuri serta melihat isi dari dokumen *web* dan berpindah dari sebuah tempat ketempat yang lain di *web*. Contoh program *browser* yang populer misalnya *Internet Explorer, Netscape, Opera, Mozilla* dan lain-lain.

### 3. *Web Page*

Menurut (Husda & Wangdra, 2016), “*Web Page* atau halaman *web* adalah sebuah *file* atau dokumen HTML yang disimpan oleh sebuah *server* komputer dan merupakan bagian dari kumpulan *file* dan dokumen yang dimiliki oleh sebuah *website*.”

#### 2.1.6. Bahasa Pemrograman

Dalam rancang bangun *website* ini digunakan beberapa bahasa pemrograman. Bahasa pemrograman ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi. Adapun bahasa pemrograman yang di pakai sebagai berikut :

## 1. HTML

Menurut (Jumardi, 2019:15), “*HyperText Markup Language (HTML)* adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi didalam sebuah penjelajah web internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi”.

## 2. CSS

Menurut (Abdulloh, 2016:2), “CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih rapi dan elegan.”

## 3. JavaScript

Menurut (Suryana & Koesharyatin, 2014:181), “JavaScript adalah bahasa *script* yang berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML”.

## 4. Java

Menurut (Supardi, 2015:1) menyimpulkan bahwa :  
Java merupakan perangkat lunak produksi **Sun Microsystem Inc.**, Java merupakan perangkat lunak untuk pemrograman beberapa tujuan (*multi purpose*), *multiplatform* (dapat berjalan di beberapa sistem operasi), mudah dipelajari dan *powerful*. Aplikasi-aplikasi yang dapat dibuat dengan Java, meliputi *Web Programming* (pemrograman web), *Desktop Programming* (pemrograman desktop), *Mobile Programming* (pemrograman *mobile/handphone*).

### a. JSP

Menurut (TIM EMS, 2015:118), “JSP (*Java Server Pages*) adalah teknologi Java yang memungkinkan developer men-*generate* HTML dan XML

secara dinamis. Atau, dengan kata lain, merupakan bahasa pemrograman *scripting* yang bersifat *server-side*”

b. Servlet

Menurut (Syarief, 2017), “Servlet merupakan bahasa pemrograman Java yang digunakan untuk memperluas kemampuan *server* melalui *request-response programming model*, sehingga dapat menambahkan konten dinamis ke *server web* baik itu HTML ataupun XML”

### 2.1.7. Basis Data

Menurut (Lubis, 2016:3) menyimpulkan bahwa :  
Basis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap pada sebuah organisasi/perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan.

1. MySQL (*My Structure Query Language*)

Menurut Budi Raharjo dalam (Hidayat, 2017) menjelaskan bahwa, “MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*) dan dapat melakukan suatu proses secara sinkron atau bersamaan (*multi-threaded*).”

### 2.1.8. Software Pendukung

Beberapa *software* pendukung yang penulis gunakan yaitu:

1. XAMPP

Menurut (Dadan & Kerendi Developers, 2015:28), “XAMPP adalah salah satu aplikasi *web server* apache yang terintegrasi dengan MySQL dan PhpMyAdmin.

XAMPP adalah singkatan dari X, Apache Server, MySQL, PhpMyAdmin, dan Phyton.”

## 2. PhpMyAdmin

Menurut (Haqi & Setiawan, 2019:9), “PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Untuk membukanya, buka *browser* lalu ketikkan alamat <http://localhost/phpmyadmin>, akan muncul halaman PhpMyAdmin.”

## 3. NetBeans

Menurut (Jubilee Enterprise, 2015:8), “NetBeans merupakan IDE (*Integrated Development Environment*) untuk membuat aplikasi dengan Java, PHP, C, C++, dan HTML5.”

### 2.1.9. Bootstrap

Menurut (Jubilee Enterprise, 2016:1), “Bootstrap adalah *framework front-end* yang intuitif dan *powerful* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah”. Sedangkan menurut (Asyikin, 2019:34), “Bootstrap adalah *toolkit open source* untuk dikembangkan dengan HTML, CSS, dan JS. Dengan kata lain, Bootstrap merupakan *framework* untuk mempercantik *user interface (UI)*.”

### 2.1.10. Back End dan Front End

Menurut (Arhandi, 2016), “*Back end* atau sering di sebut *server side* pada dasarnya adalah tempat dimana proses suatu aplikasi atau sistem berjalan di *back end* ini data di proses ditambahkan, diubah atau dihapus”.

Menurut (Arhandi, 2016), “*Front end* adalah segala sesuatu yang menghubungkan antara user dengan sistem *back end*. Biasanya merupakan sebuah *user interface* dimana *user* akan berinteraksi dengan sistem”.

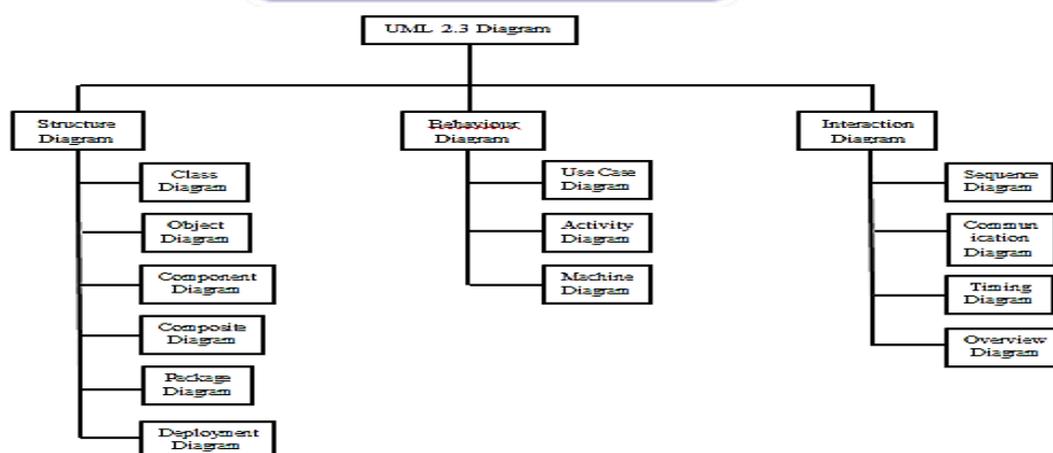
## 2.2. Peralatan Pendukung

Adapun peralatan pendukung yang mendukung kemudahan dalam mempelajari dan merancang program *web* ini sebagai berikut :

### 2.2.1. Unified Modelling Language (UML)

Menurut (Sukamto & Shalahuddin, 2018:140), “*Unified Modelling Language* (UML) merupakan sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek”.

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



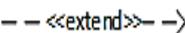
Sumber : (Sukamto & Shalahuddin, 2018:140)

**Gambar II.1.**  
**UML 2.3. Diagram**

## 1. Use Case Diagram

Menurut (Sukamto & Shalahuddin, 2018:140), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. Berikut ini adalah simbol-simbol yang ada pada *use case diagram*:

**Tabel II.1.**  
*Use Case Diagram*

	<p><b>USECASE</b> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.</p>
	<p><b>ACTOR</b> Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
	<p><b>ASSOCIATION</b> Komunikasi antara <i>actor</i> dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki Interaksi dengan aktor.</p>
	<p><b>INCLUDE</b> Relasi <i>usecase</i> dimana proses bersangkutan akan dilanjutkan keproses yang dituju.</p>
	<p><b>EXTEND</b> Relasi <i>usecase</i> tambahan sebuah <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu.</p>

Sumber : (Sukamto & Shalahuddin, 2018:140)

## 2. Activity Diagram

Menurut (Sukamto & Shalahuddin, 2018:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Berikut ini adalah simbol-simbol yang ada pada *activity diagram*:

**Tabel II.2.**  
**Activity Diagram**

	<b>INITIAL NODE</b> Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
	<b>ACTION</b> Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<b>DECISION</b> Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<b>JOIN</b> Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<b>ACTIVITY FINAL</b> Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<b>SWIMLANE</b> Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

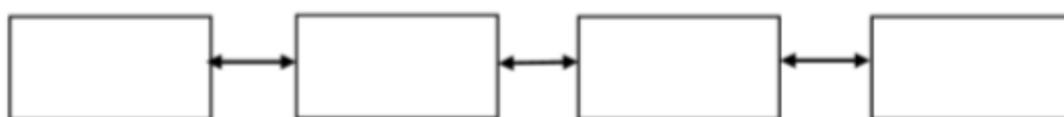
Sumber : (Sukamto & Shalahuddin, 2018:161)

### 2.2.2. Struktur Navigasi

Menurut (Andriansyah, 2016:61), “Struktur Navigasi dapat diartikan sebagai alur dari suatu program yang menggambarkan rancangan hubungan antar area yang berbeda sehingga memudahkan proses pengorganisasian seluruh *element website*”. Bentuk dasar dari struktur navigasi sebagai berikut :

### 1. Struktur Navigasi *Linear*

Struktur navigasi *linear* hanya mempunyai rangkaian cerita yang berurut yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya, pengguna akan melakukan navigasi secara berurutan, dalam *frame* atau *byte* informasi satu ke yang lainnya.

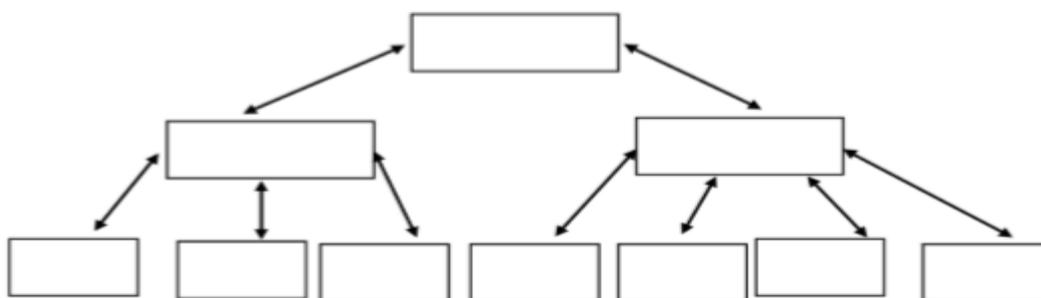


Sumber : (Andriansyah, 2016:61)

**Gambar II.2.**  
**Struktur Navigasi *Linear***

### 2. Struktur Navigasi Hierarki

Struktur dasar ini disebut juga struktur *linear* dengan percabangan karena pengguna melakukan navigasi disepanjang cabang pohon struktur yang terbentuk oleh logika isi.

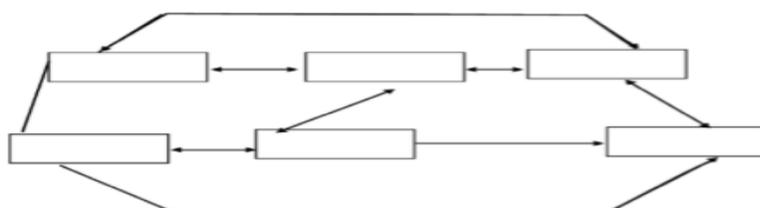


Sumber : (Andriansyah, 2016:61)

**Gambar II.3.**  
**Struktur Navigasi Hierarki**

### 3. Struktur Navigasi Tidak Berurutan (*Non-Linear*)

Struktur navigasi *non-linear* merupakan pengembangan dari struktur navigasi *linear*. Pada struktur ini diperkenankan membuat navigasi bercabang percabangan yang dibuat pada struktur *non-linear* ini berbeda dengan percabangan pada struktur hierarki, karena ada percabangan *non-linear* ini walaupun terdapat percabangan tetap tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*, pengguna akan melakukan navigasi dengan bebas melalui isi proyek dengan tidak terikat jalur yang sudah ditentukan sebelumnya.

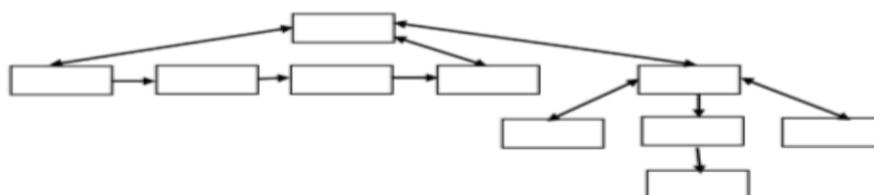


Sumber : (Andriansyah, 2016:61)

**Gambar II.4.**  
**Struktur Navigasi Tidak Berurutan (*Non-Linear*)**

### 4. Struktur Navigasi Campuran (*Composite*)

Struktur navigasi jenis ini pengguna akan melakukan navigasi dengan bebas (secara *non-linear*) tetapi terkadang dibatasi presentasi *linear* film atau informasi penting dan pada data yang paling terorganisasi secara logis pada suatu hierarki.



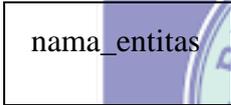
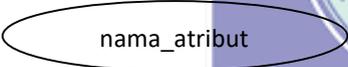
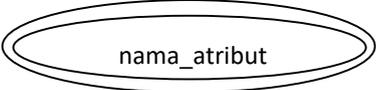
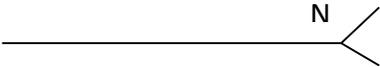
Sumber : (Andriansyah, 2016:61)

**Gambar II.5.**  
**Struktur Navigasi Campuran (*Composite*)**

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

Menurut (Hutahaean, 2014:109), “*Entity Relationship (E-R) Diagram* adalah *high level conceptual data model* yang dikembangkan oleh Chen (1976) untuk memfasilitasi perancangan *database*”. Sedangkan menurut (Sukamto & Shalahuddin, 2018:50), “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*. ERD digunakan untuk pemodelan basis data relasional”.

**Tabel II.3.**  
***Entity Relationship Diagram***

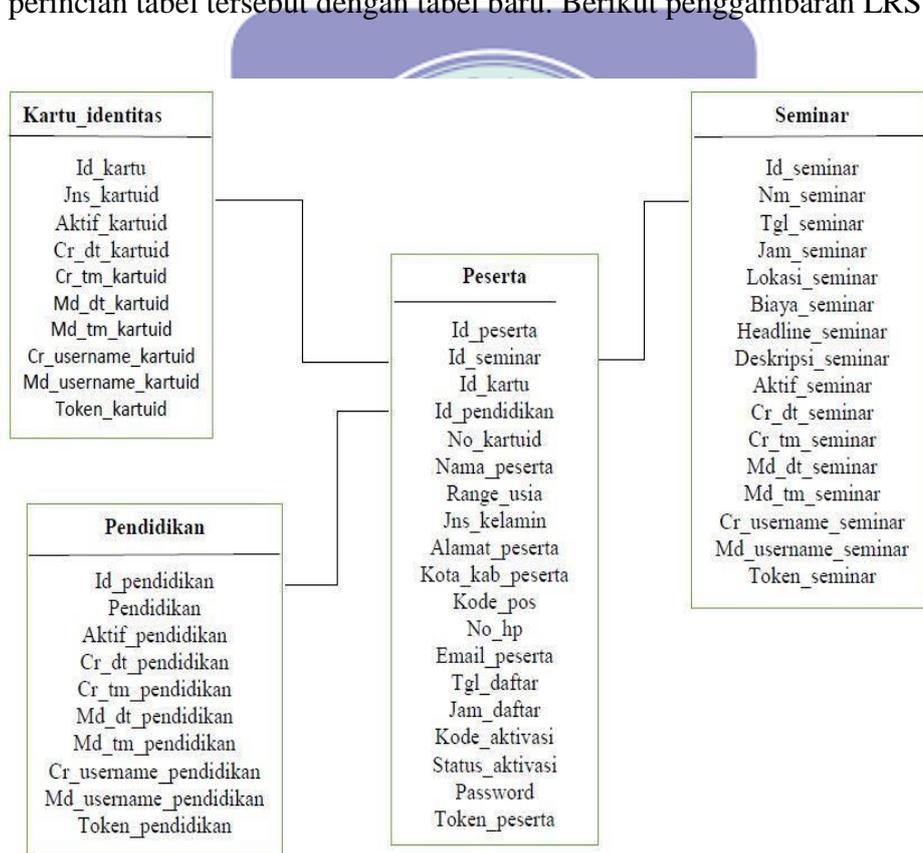
Simbol	Deskripsi
Entitas 	Entitas merupakan data inti yang akan disimpan
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan
Atribut multivalai 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas
Asosiasi 	Penghubung antar relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

Sumber : (Sukamto & Shalahuddin, 2018:50),

### 2.2.4. LRS (*Logical Record Structure*)

Menurut (Andriansyah, 2016:53) memberikan batasan bahwa, “*Logical Record Structure* (LRS) merupakan transformasi penggambaran ERD dalam bentuk lebih jelas dan mudah untuk dipahami”.

Penggambaran LRS hampir mirip dengan penggambaran normalisasi file, hanya saja tidak digambarkan simbol *asterix* sebagai simbol *primary key* (kunci utama) dan *foreign key* (kunci tamu). Apabila dalam penggambaran ERD pada point sebelumnya terdapat tabel yang didetail-kan, maka pada LRS dapat digambarkan secara jelas perincian tabel tersebut dengan tabel baru. Berikut penggambaran LRS :



Sumber : (Andriansyah, 2016:53)

**Gambar II.6.**  
**Contoh Rancangan LRS Pendaftaran Peserta**

### 2.2.5. *Black Box Testing*

Menurut (Kusniyati & Sitanggang, 2016), “Pengujian *Black-Box* merupakan metode yang menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Sedangkan menurut Rizky dalam (Rini, Iqbal, & Astuti, 2016) mengemukakan bahwa, “*Testing* adalah sebuah proses yang disebut sebagai siklus hidup dan merupakan bagian dari proses rekayasa perangkat lunak secara terintegrasi demi memastikan kualitas dari perangkat lunak serta memenuhi kebutuhan teknis yang telah disepakati dari awal.”

*Black box testing* ditujukan untuk berusaha menemukan kesalahan dalam beberapa kategori yaitu:

1. Fungsi-fungsi yang tidak sesuai atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.

### 2.2.6. *User Interface*

Menurut (Handayani, 2015:63), “Pengertian *User Interface* (UI) adalah penghubung antara manusia sebagai pengguna (*user*) dengan *software* pada perangkat (*interface*)”.