



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Data

Menurut Sutabri (2014:122), data adalah fakta-fakta yang akan dibuat menjadi informasi yang bermanfaat. Data inilah yang akan diklasifikasikan, dimodifikasi atau diolah oleh program-program supaya dapat menjadi informasi yang tepat guna, tepat waktu, dan akurat.

Sedangkan menurut Setyaningrum (2013:1), data adalah catatan atas sekumpulan fakta yang belum mempunyai arti bagi penerimanya dan maish memerlukan suatu pengolahan. Data dapat dinyatakan dalam bentuk karakter, angka, simbol, suara, atau dalam bentuk simbol lainnya yang bias kita gunakan sebagai bahan untuk melihat lingkungan obyek, kejadian ataupun suatu konsep.

Usman dan Akbar (dikutip Setiawan 2013:12) mengungkapkan bahwa, “data ialah suatu bahan mentah yang jika diolah dengan baik melalui berbagai analisis dapat melahirkan berbagai informasi”. Sedangkan menurut Agresti dan Finlay (dikutip Setiawan 2013:12), “data terdiri dari pengukuran-pengukuran pada karakteristik yang menarik”.

2.1.2 Pengertian Pengolahan Data

Yakub (2012:75) mengemukakan bahwa, “pengolahan data adalah manipulasi atau transformasi simbol-simbol seperti; angka dan abjad untuk tujuan meningkatkan kegunaannya”.

Manullang (2016:313) mengemukakan bahwa, “pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data”.

Dari beberapa pendapat diatas dapat disimpulkan bahwa pengolahan data terdiri dari kegiatan-kegiatan memanipulasi data yang lebih spesifik atau berupa informasi.

2.1.3 Pengertian *Internet*

Nugraha (2013:1) mengemukakan bahwa, *internet* (kependekan dari *interconnected-networking*) adalah sistem global dari seluruh jaringan komputer



yang saling terhubung menggunakan standar *Internet Protocol (TCP/IP)* untuk melayani miliaran pengguna diseluruh dunia.

Sujatmiko (2012:138) mengemukakan bahwa, *internet* merupakan jaringan global yang menghubungkan berjuta-juta komputer diseluruh dunia melalui jalur telepon kabel maupun satelit.

Dari beberapa pendapat diatas dapat disimpulkan *internet* adalah jaringan komputer yang saling terhubung untuk memberikan informasi kepada penggunanya.

2.1.4 Pengertian Perangkat Keras (*Hardware*)

Menurut Sutabri (2014:122), istilah perangkat keras merujuk pada perkakas mesin. Karena itu, perangkat keras terdiri dari komputer itu sendiri yang terkadang disebut sebagai *CPU* beserta semua perangkat pendukungnya. Perangkat pendukung yang dimaksud adalah peranti keluaran, peranti penyimpan, dan peranti komunikasi.

Sedangkan Puspitosari (2013:7) mengemukakan bahwa, "*hardware* merupakan salah satu elemen dari sistem komputer suatu alat yang bisa dilihat dan diraba oleh manusia secara langsung yang mendukung proses komputerisasi". Menurut Manullang (2016:317), perangkat keras adalah suatu aspek dasar secara teknis di dalam suatu proses komputer yang merupakan keseluruhan dari komponen peralatan yang membentuk suatu sistem komputer yang harus dipenuhi agar proses komputerisasi dapat berjalan. Perangkat keras menyediakan 5 (lima) fungsi utama, yakni:

1. Penyimpanan data.
2. Masukan pada komputer.
3. Komputas, pengendalian dan penyimpanan utama.
4. Penyimpanan sekunder.
5. Keluaran dari komputer.

2.1.5 Pengertian Perangkat Lunak (*Software*)

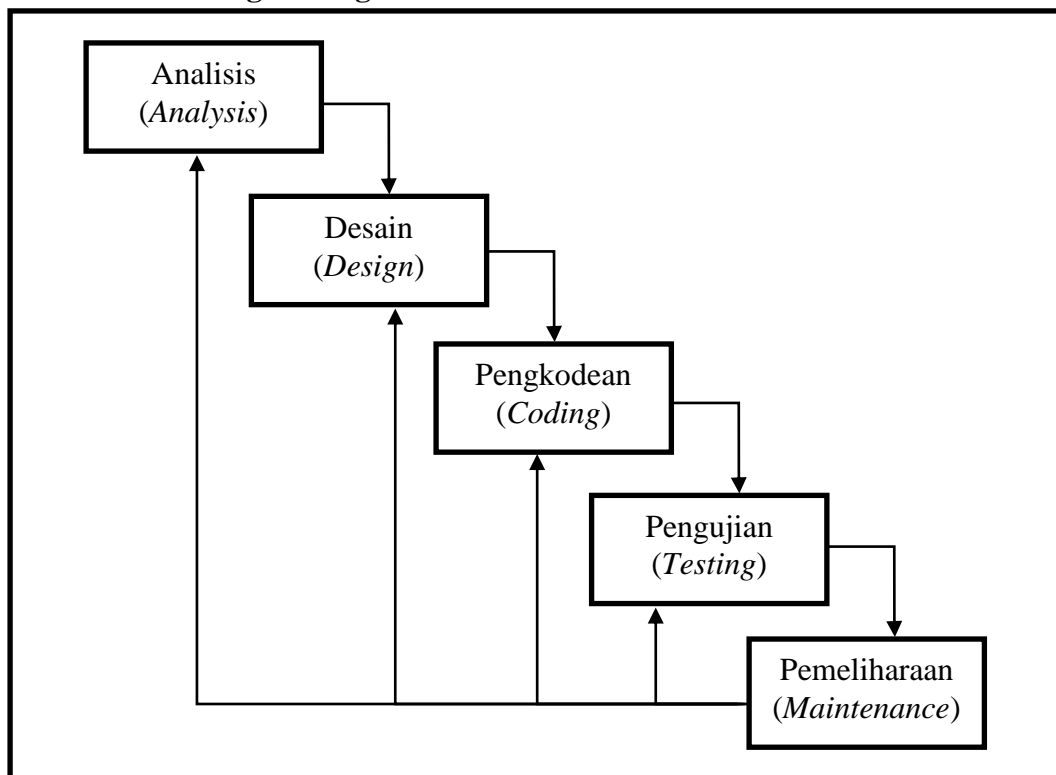
Manullang (2016:317) mengemukakan bahwa, perangkat lunak merupakan suatu prosedur dari sistem pengolahan data di luar mesin komputer, yang akan mengarahkan atau memberikan kemudahan pengoperasian dengan



peraturan yang berlaku, standard yang digunakan dan lain sebagainya. Inti dari fasilitas perangkat lunak adalah apa yang disebut dengan *system design* dan *programs*.

Sedangkan menurut Sutabri (2014:122), istilah perangkat lunak merujuk pada program-program komputer beserta manual pendukungnya. Yang disebut program komputer adalah instruksi-instruksi yang dapat dibaca oleh mesin yang memerintahkan bagian-bagian perangkat keras SIM berbasis komputer untuk berfungsi sedemikian rupa sehingga dapat menghasilkan informasi yang bermanfaat dari data yang tersedia.

2.1.6 Metode Pengembangan Sistem



Gambar 2.1 Ilustrasi Model *Waterfall*

Sukanto dan M.Shalahudin (2013:28-30) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut dengan model sekuensial linier (*sequesntial linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup terurut mulai dari analisis, desain, pengkodean, pengujian, dan pemeliharaan.



1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan kode program

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang dibuat pada tahap desain.

4. Pengujian

Tahap pengujian fokus pada perangkat lunak dari segi logika 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 terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem yang baru.



2.2 Teori Khusus

2.2.1 Pengertian *Data Flow Diagram (DFD)*

Sunyoto (2014:129) mengemukakan bahwa, *Data Flow Diagram (DFD)* adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas. Di dalam *data flow diagram* terdapat 3 level yaitu :

1. Diagram Konteks: menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi dalam *data flow diagram* dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.
2. Diagram Nol (diagram *level-1*): merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram konteks ke diagram nol. Didalam diagram ini memuat penyimpanan data.
3. Diagram Rinci: merupakan diagram yang menguraikan proses apa yang ada dalam diagram nol.

Sukamto dan M. Shalahuddin (2013:70) mengemukakan bahwa, *Data Flow Diagram (DFD)* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan *DFD*:

1. Membuat *DFD Level 0* atau sering disebut juga *Context Diagram*. *DFD Level 0* menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. *DFD Level 0* digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.



2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

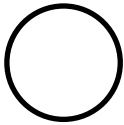


Modul-modul pada DFD Level 2 dapat di-*breakdown* menjadi DFD Level Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.


Adapun simbol-simbol *Data Flow Diagram* adalah sebagai berikut:

Tabel 2.1 Simbol-Simbol *Data Flow Diagram*

No.	Notasi	Keterangan
1.		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya di dalam kode program
2.		File atau basisdata atau penyimpanan; pada pemodelan perangkat lunak yang akan di implementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan.
3.		Entitas luar (<i>external entity</i>) orang yang berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan



Lanjutan Tabel 2.1 Simbol-Simbol *Data Flow Diagram*

4.		Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan.
----	---	--



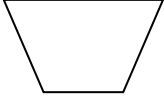
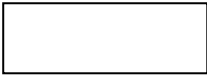
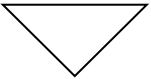

(Sumber: Sukamto dan M. Shalahuddin, 2013:71)

2.2.2 Pengertian *Block Chart*


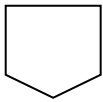
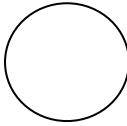
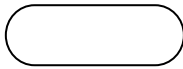
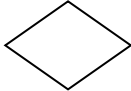

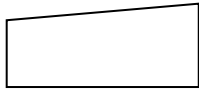
Kristanto (2008:68) mengemukakan bahwa, *block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol *block chart* adalah sebagai berikut:

Tabel 2.2 Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>data storage</i>)

Lanjutan Tabel 2.2 Simbol-Simbol *Block Chart*




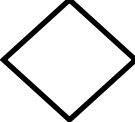


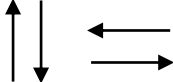



7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (monitor).
13.		Pemasukan data secara manual

(Sumber: Kristanto, 2008:68)

2.2.3 Bagan Alir Sistem (*System Flowchart*)

Jogiyanto (2009:455), bagan alir sistem (*system flowchart*) digunakan untuk menggambarkan proses dari sistem yang lama atau sistem baru yang diusulkan. Bagan alir sistem juga menunjukkan arus dari dokumen-dokumen yang ada di organisasi, sehingga disebut juga dengan nama bagan alir dokumen (*document flow chart*). Adapun simbol-simbol *flowchart* adalah sebagai berikut:

Tabel 2.3 Simbol-Simbol *FlowChart*

No.	Simbol	Arti
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
3.		Simbol <i>Input/Output</i> yang mendefinisikan masukan dan keluaran proses
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen

Lanjutan Tabel 2.3 Simbol-Simbol *FlowChart*

11.		Simbol yang menyatakan bagian dari program (subprogram)
12.		Simbol masukan atau keluaran dari atau ke sebuah pita <i>magnetic</i>
13.		Simbol <i>database</i> atau basis data

(Sumber: Ladjamudin, 2013:266-268)

2.2.4 Pengertian *Entity Relationship Diagram (ERD)*

McLeod dan Schell (2004:429), diagram hubungan entitas (*entity relationship diagram*) atau *ERD*, mendominasi data perusahaan dengan mengidentifikasi jenis entitas dan hubungannya. *ERD* disiapkan pada suatu titik dalam proses pengembangan sistem saat “gambaran besar” data ditentukan.





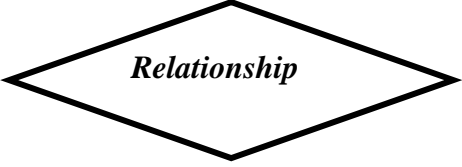
Simarmata (2007:111-114), *Entity Relationship Diagrams (ERD)* mengilustrasikan struktur logis dari basis data. Peter Chen mengembangkan *ERD* pada tahun 1976. Kemudian, Charles Bachman dan James Martin menambahkan beberapa perbaikan dalam prinsip-prinsip dasar *ERD*.

Tabel 2.4 Simbol-Simbol *ERD*

No	Simbol	Keterangan
1		Entity Suatu <i>entity</i> merupakan suatu objek atau konsep mengenai tempat yang Anda inginkan untuk menyimpan informasi
2		Weak Entity Suatu <i>weak entity</i> tergantung pada entitas lainnya <i>to exist</i>

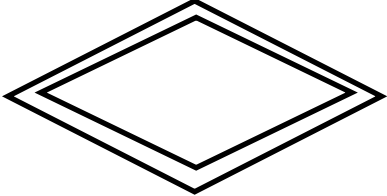
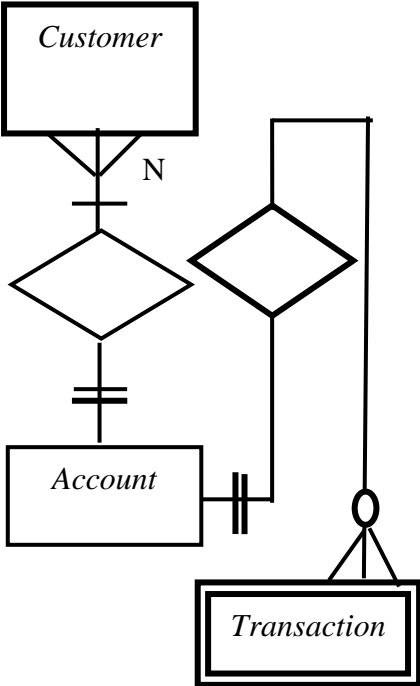
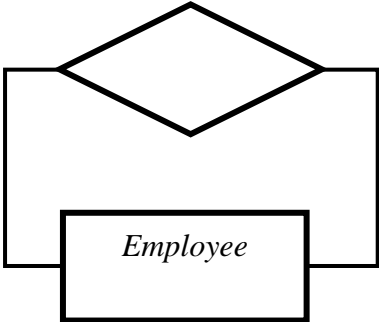


Lanjutan Tabel 2.4 Simbol-Simbol ERD

3		<p>Attributes</p> <p><i>Attributes</i> adalah sifat-sifat atau karakteristik dari suatu entitas</p>
4		<p>Key attribute</p> <p>Suatu <i>key attribute</i> adalah unik (<i>unique</i>), dan memiliki karakteristik pembeda dari entitas. Sebagai contoh, nomor mahasiswa mungkin menjadi atribut <i>key</i> mahasiswa</p>
5		<p>Multivalued attribute</p> <p>Suatu <i>multivalued attribute</i> memiliki lebih dari satu nilai. Sebagai contoh, suatu entitas pegawai bisa memiliki nilai pada berbagai keahlian</p>
6		<p>Derived attribute</p> <p>Suatu <i>derived attribute</i> didasarkan pada atribut lainnya. Sebagai contoh, gaji bulanan seorang pegawai berdasarkan pada gaji bulanan karyawan lain yang berdasarkan pada gaji tahunan</p>
7		<p>Relationships</p> <p><i>Relationships</i> mengilustrasikan bagaimana dua entitas berbagai informasi di dalam struktur basis data. Cara menggambar relasi adalah menghubungkan dua entitas terlebih dahulu, baru kemudian menge-<i>drop</i> notasi relasi pada garis</p>



Lanjutan Tabel 2.4 Simbol-Simbol ERD

8		<p>Weak relationship</p> <p>Untuk menghubungkan <i>weak entity</i> dengan yang lainnya, Anda perlu menggunakan notasi <i>weak relationship</i></p>
9		<p>Cardinality</p> <p>Penetapan kardinalitas atas jumlah kejadian dari suatu entitas pada suatu kejadian dari entitas lainnya. Ordinalitas juga dihubungkan ke kardinalitas. Sementara kardinalitas menetapkan kejadian pada suatu hubungan. Ordinalitas menguraikan hubungan seperti mandatori maupun opsional. Dengan kata lain, kardinalitas menetapkan jumlah maksimum hubungan dan ordinalitas menetapkan jumlah minimum</p>
10		<p>Recursive relationship</p> <p>Dalam banyak kasus, entitas bisa dihubungkan dengan dirinya sendiri. Sebagai contoh, karyawan bisa mengawasi karyawan yang lain.</p>

(Sumber: Simarmata , 2007:112-114)



2.2.5 Pengertian Kamus Data (*Data Dictionary*)

Yakub (2012:168) mengemukakan bahwa, “kamus data (*data dictionary*) merupakan daftar elemen data yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem sehingga user dan analisis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan *data storage*”.

Sukamto dan M. shalahuddin (2013:73) mengemukakan bahwa kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

- a. Nama, nama dari data
- b. Digunakan pada, merupakan proses-proses yang terkait data
- c. Deskripsi, merupakan deskripsi data
- d. Informasi tambahan, seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Adapun simbol-simbol yang sering digunakan dalam kamus data adalah sebagai berikut:

Tabel 2.5 Simbol-Simbol Kamus Data

No.	Simbol	Arti
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik ...atau...
4.	{ ⁿ }	N kali diulang / bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

(Sumber : Sukamto dan M. Shalahuddin, 2013:74)



2.2.6 Daftar Kejadian (*Event List*)

Kristanto (2008:64) menjelaskan tentang daftar kejadian (*event list*) sebagai berikut:

Daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari sistem. Suatu kejadian mewakili satu aliran data atau proses dalam diagram konteks serta deskripsi penyimpanan yang digunakan untuk memodelkan data harus diperhatikan dalam kaitannya dengan daftar kejadian.

Adapun cara-cara mendeskripsikan daftar kejadian adalah sebagai berikut:

1. Pelaku adalah entitas luar, jadi bukan sistem.
2. Menguji setiap entitas luar dan mencoba mengevaluasi setiap entitas luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan entiti luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaiknya merupakan respon dari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.

Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.3 Teori Judul

2.3.1 Pengertian Sistem

Menurut Sutabri (2012:3) mengemukakan bahwa, “sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung sama lain dan terpadu”.



Sedangkan menurut Jogiyanto (2009:34), sistem (*system*) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu.

Dari beberapa pendapat diatas dapat disimpulkan sistem adalah sekumpulan komponen yang dihubungkan bersama untuk memudahkan aliran informasi.

2.3.2 Pengertian Informasi

Sutabri (2014:2) mengemukakan bahwa, informasi adalah hasil pemrosesan, manipulasi dan perorganisasian atau penataan dari sekelompok data yang mempunyai nilai pengetahuan bagi penggunanya.

Sedangkan menurut Jogiyanto (2009:36), informasi (*information*) adalah data yang diolah menjadi bentuk berguna oleh para pemakainya. Jadi, dari dua pendapat di atas dapat disimpulkan informasi adalah data penting yang diolah menjadi bentuk yang berguna sebagai bahan pengambilan keputusan.

Dari beberapa pendapat diatas dapat disimpulkan informasi adalah pengumpulan atau pengolahan data yang penting dan mempunyai nilai untuk memberikan pengetahuan.

2.3.3 Tipe Informasi

Menurut Jogiyanto (2009:69), Sistem informasi dapat menyediakan tiga macam tipe informasi, masing-masing mempunyai arti yang berbeda untuk tingkatan manajemen yang berbeda yaitu:

1. Informasi Pengumpulan Data (*scorekeeping information*).

Informasi Pengumpulan Data (*scorekeeping information*) merupakan informasi yang berupa akumulasi atau pengumpulan data untuk menjawab pertanyaan: "Am I doing well or badly?" (Apakah saya sudah mengerjakannya dengan baik atau belum?). Informasi ini berguna bagi manajer bahwa untuk mengevaluasi kinerja personil-personilnya.



2. Informasi Pengarahan Perhatian (*attention directing information*).

Informasi Pengarahan Perhatian (*attention directing information*) merupakan informasi untuk membantu manajemen memusatkan perhatian pada masalah-masalah yang menyimpang, ketidakberesan, ketidakefisienan, dan kesempatan-kesempatan yang dapat dilakukan. Informasi ini untuk menjawab pertanyaan: "What problem should I look into?" (Permasalahan apakah yang seharusnya saya amati?). Informasi ini akan membantu manajemen menengah untuk melihat penyimpangan-penyimpangan yang terjadi.

3. Informasi Pemecahan Masalah (*problem solving information*).

Informasi Pemecahan Masalah (*problem solving information*) merupakan informasi untuk membantu manajer atas mengambil keputusan memecahkan permasalahan yang dihadapinya. Informasi ini untuk menjawab pertanyaan: "Of the several ways of doing the job which is the best?" (Manakah yang terbaik dari beberapa cara melakukan pekerjaan?). *Problem Solving* biasanya dihubungkan dengan keputusan-keputusan yang tidak berulang-ulang serta situasi yang membutuhkan analisis yang dilakukan oleh manajemen tingkat atas.

2.3.4 Pengertian Sistem Informasi

Menurut Pratama (2014:10) mengemukakan bahwa sistem informasi merupakan gabungan dari empat bagian utama mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih.

Alter (dikutip Kadir dan Triwahyuni 2013:546), "sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi".

Dari beberapa pendapat diatas dapat disimpulkan sistem informasi adalah suatu sistem didalam suatu organisasi yang mengumpulkan informasi untuk pengambilan keputusan.

2.3.5 Pengertian Website

Utomo (2013:1) mengemukakan bahwa *web* merupakan representasi sebuah halaman di *internet* yang menyajikan berbagai macam informasi.



Halaman-halaman tersebut merupakan dokumen yang tersebar di beberapa komputer di seluruh dunia yang terhubung satu sama lain.

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

2.3.6 Sistem Informasi Kepegawaian pada Kantor Dinas Perdagangan Kota Palembang Berbasis Web

Sistem Informasi Kepegawaian pada Kantor Dinas Perdagangan Kota Palembang Berbasis *Web* merupakan sebuah sistem informasi yang dibuat dengan menggunakan bahasa pemrograman *PHP* dan *database MySQL* untuk memberikan informasi mengenai kepegawaian mulai dari data pegawai, data jabatan, data golongan, data cuti, data pensiun, serta daftar urut kepangkatan (DUK) pada Kantor Dinas Perdagangan Kota Palembang.

2.4 Teori Program

2.4.1 Pengertian Basis Data (*Database*)

Menurut Kadir dan Triwahyuni (2013:484), “basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi”.

Sedangkan menurut Setyaningrum (2013:2), basis data merupakan kumpulan data yang terdiri dari atribut, *entity* dan *relationship* dari informasi suatu instansi atau perusahaan yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi.

Fathansyah (2015:2), Basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.



- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.4.1.1 Data Definition Language

Menurut Yakub (2012:65-67), *DDL* merupakan singkatan dari *data definition language* yang juga bagian dari *structured query language (SQL)*. *DDL* berfungsi lebih ke dalam memanipulasi struktur dari *database*. *DDL* digunakan untuk membuat tabel atau menghapus tabel, membuat *key* atau indeks, membuat relasi antartabel. Beberapa statement atau sintaks yang sering dijumpai adalah sebagai berikut:

- a. *CREATE TABLE*, bertugas untuk membuat tabel.
- b. *CREATE INDEX*, bertugas untuk membuat suatu indeks dalam tabel.
- c. *DROP TABLE*, bertugas untuk menghapus suatu tabel.
- d. *ALTER TABLE*, bertugas untuk mengubah struktur suatu tabel.

DDL merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel, atribut serta hubungan antartabel dijelaskan sebagai berikut:

a. *CREATE*

Pernyataan ini digunakan untuk menciptakan tabel, indeks, sintaks *SQL* untuk melakukan pembuatan tabel dan indeks di dalam basis data adalah sebagai berikut:

CREATE TABLE T (A1, D1, A2, D2, ..., An, Dn)

Dimana:

- T : nama tabel yang akan dibuat.
 A1, A2, ..., An : nama-nama atribut yang terdapat di dalam tabel T.
 D1, D2, ..., Dn : domain nilai masing-masing atribut yang ditentukan berdasarkan tipe datanya.

Pada perintah *CREATE TABLE*, minimal harus mendefinisikan nama tabel, kolom-kolom dan tipe datanya.



```
CREATE TABEL tblpengarang(
    Kd_pengarang INTEGER,
    Nama_pengarang CHAR (15),
    Alamat_pengarang CHAR (30),
    Kota CHAR (15)).
```

b. *DROP*

Pernyataan drop ini digunakan untuk menghapus tabel dan indeks, sedangkan sintaks *SQL* untuk melakukan penghapusan tabel dan indeks adalah sebagai berikut:

```
DROP TABLE Nama_tabel
```

Dimana:

Nama_Tabel adalah nama dari tabel yang akan dihapus.

```
DROP TABLE tblpengarang.
```

Perintah ini merupakan pernyataan untuk menghapus tabel pengarang.

c. *ALTER TABLE*

Pernyataan *alter table* ini digunakan untuk mengubah struktur tabel, contoh *SQL* untuk melakukan perubahan struktur tabel adalah sebagai berikut.

```
ALTER TABLE tblpengarang
    ADD Kelamin CHAR (1).
```

Perintah ini merupakan pernyataan untuk menambah kolom dengan nama kolom kelamin dan bertipe *CHAR* (1).

2.4.1.2 *Data Manipulation Language*

Menurut Yakub (2012:67-69), *Structured query language (SQL)* adalah sekumpulan sintaks-sintaks atau statement untuk mengakses data dalam *database*, tetapi *SQL* sendiri juga bisa digunakan untuk melakukan proses *insert*, *update* atau *delete* ke dalam suatu *database*. Sintaks-sintaks ini yang disebut dengan *Data Manipulation Language (DML)* yang merupakan bagian dari *SQL*. Berikut ini adalah penjelasan singkat dari sintaks-sintaks *SQL* sebagai berikut:

- a. *INSERT*, bertugas untuk menambahkan data ke dalam suatu tabel dalam *database*.



- b. *UPDATE*, bertugas untuk mengupdate (mengubah) data dalam suatu tabel pada *database*.
- c. *SELECT*, bertugas untuk mengakses data dari suatu tabel dalam *database*.
- d. *DELETE*, bertugas untuk menghapus data dari suatu tabel dalam *database*.

Data Manipulation Language (DML) merupakan kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data, dan kelompok *DML* dijelaskan sebagai berikut:

a. *INSERT*

Pernyataan *insert* ini digunakan untuk menambah sebuah baris pada suatu tabel, sintaks *SQL* untuk melakukan penambahan baris adalah sebagai berikut:

INSERT [INTO] Nama Tabel [Daftar_Kolom] Value Daftar_Nilai

Dimana:

INSERT : Klausa ini menspesifikasikan nama tabel dimana data bisa ditambahkan.

VALUE : Klausa ini menspesifikasikan nilai data yang akan disisipkan ke dalam kolom pada suatu tabel.

Daftar_Kolom : Merupakan daftar kolom yang dipisahkan oleh tanda koma menyatakan kolom-kolom yang akan diisi data. Jika tidak ada kolom yang dinyatakan, berarti semua kolom di dalam akan di isi data. Jika hanya sebagian daftar yang dinyatakan, nilai null atau nilai default akan diisikan ke kolom yang tidak disebutkan dalam daftar kolom.

Daftar_Nilai : Daftar nilai untuk kolom tabel yang akan disisipkan sebagai sebuah baris data dalam tabel. Data yang diberikan pada daftar nilai harus sesuai dengan daftar kolom. Banyak data harus sama dengan banyak kolom, tipe data, presisi, dan skala dari setiap data harus sesuai dengan kolomnya.



INSERT INTO tblpengarang

VALUES (1,'Asnadi','Jl. Beo 34','Yogyakarta','P')

Perintah ini merupakan pernyataan untuk menambah data dalam 1 (satu) baris.

INSERT INTO tblpengarang (Kd_Pengarang, Nama_pengarang)

VALUES (11,'Hendi')

b. *UPDATE*

Pernyataan *update* ini digunakan untuk mengubah sebuah data pada suatu tabel, contoh *SQL* untuk melakukan pengubahan data adalah sebagai berikut:

UPDATE tblpengarang

SET Nama_pengarang='Hendi'

WHERE Kd_pengarang=11

Pernyataan di atas:

- 1) *SET*, untuk menentukan kolom yang akan diubah dan nilai penggantinya.
- 2) *WHERE*, untuk menentukan kondisi dari baris-baris yang akan diganti.

c. *SELECT*

Pernyataan ini digunakan untuk memilih sebuah baris dan kolom pada tabel, sintaks *SQL* untuk melakukan pemilihan baris dan kolom adalah sebagai berikut:

SELECT *

FROM Nama_Tabel

Dimana:

* : Memilih semua kolom.

Nama_Tabel : Tabel yang akan diambil datanya.

SELECT juga memungkinkan untuk menampilkan baris tertentu saja, tergantung apa baris dan kolom yang diinginkan, misalnya untuk menampilkan nama pengarang yang tinggal di Yogya dan ingin menampilkan nama pengarang yang mempunyai kode pengarang 5.

SELECT Nama_pengarang, Kota

FROM tblpengarang

WHERE Kota='Yogya'



Contoh lain, misalnya untuk menampilkan nama pengarang yang mempunyai kode pengarang 5.

```
SELECT Kd_pengarang, nama_pengarang
FROM tblpengarang
WHERE Kd_pengarang=5
```

2.4.1.3 Data Control Language

Menurut Yakub (2012:69-70), *Data control language (DCL)* merupakan kelompok perintah yang berisi untuk mengendalikan pengaksesan data *DCL* digunakan untuk menangani masalah keamanan dalam *database server*. Pengelompokkan dalam perintah *DCL* ini adalah *GRANT* dan *REVOKE*.

a. *GRANT*

Pernyataan *GRANT* ini digunakan untuk memberikan atau mengizinkan seorang *user* untuk mengakses tabel dalam *database* tertentu, sintaks yang digunakan adalah sebagai berikut:

```
GRANT hak ON tabel_atau_hak TO pengguna
```

Hak dapat berupa *INSERT*, *DELETE*, *UPDATE*, dan *SELECT*, dalam hal ini hak dapat lebih dari satu macam. Contoh pernyataan *GRANT* akan diberikan dengan didasarkan pada tabel pengarang dan tabel gaji.

```
GRANT UPDATE ON tblgaji(gaji) TO Widya
```

Pernyataan di atas menyatakan bahwa pengguna Widya diberi hak *UPDATE* (mengubah data) terhadap kolom gaji pada tabel *tblgaji*.

```
GRANT INSERT, DELETE ON tblpengarang, tblgaji TO Dahlan
```

Pernyataan di atas merupakan pernyataan untuk memberikan hak *INSERT* (menambah baris baru) dan *DELETE* (menghapus baris) terhadap tabel *tblpengarang* dan *tblgaji* kepada pengguna bernama Dahlan.

```
GRANT SELECT ON tblpengarang TO Sinta, Wawan
```

Pernyataan di atas menyatakan bahwa pengguna bernama Sinta dan Wawan diberi hak *SELECT* (melihat data) terhadap tabel *tblpengarang*.

b. *REVOKE*

Pernyataan *REVOKE* ini digunakan untuk mencabut suatu hak akses dalam *database* tertentu, contoh pernyataan yang digunakan adalah sebagai berikut:



REVOKE SELECT ON tblpengarang *FROM* Sinta

Pernyataan di atas menyatakan bahwa, Sinta tidak mempunyai hak *SELECT* lagi terhadap tabel tblpengarang.

2.4.2 Pengertian XAMPP



Gambar 2.2 Logo XAMPP

Riyanto (2015:1) mengemukakan bahwa, XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa perangkat lunak berbeda ke dalam satu paket.

Madcoms (2016:186) mengemukakan bahwa, XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla dan lain-lain. XAMPP berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan pengembangan *web* memerlukan PHP, Apache, MySQL dan PhpMyAdmin serta *software-software* yang terkait dengan pengembangan *web*.

Menurut Nugroho (2013:1-5), XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal. Pada saat buku ini dibuat, XAMPP sudah merilis versi Xampp 1.7.7. Berikut cara mendapatkannya:

1. Cari koneksi internet, atau dapat dari Warung *Internet* (Warnet).
2. Buka alamat; www.apachefriends.org, lalu klik ikon XAMPP, sehingga masuk pada halaman; <http://www.apachefriends.org/en/xampp.html>.
3. Gulung layar ke bawah, lalu pilih versi XAMPP yang mau di ambil, misalnya saja versi XAMPP for Windows.
4. Pada halaman versi, pilih saja yang *Installer* supaya gampang menginstalnya.
5. Akan muncul jendela penyimpanannya, klik saja tombol *Save File*, lalu simpan pada komputer Anda.



2.4.3 Pengertian MySQL (*My Structure Query Language*)



Gambar 2.3 Logo MySQL

Menurut Hendry (2015:7), *MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi *GPL (General Public License)*. Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

Pramono (2015:19) menyatakan bahwa, *MySQL* adalah suatu perangkat lunak untuk relasi *database (Relation Database Management System/RDMS)* seperti halnya *Oracle, PostgreSQL, Microsoft SQL*. Sebuah *database* dapat terdiri dari beberapa *table*.

Beberapa tipe data dalam *MySQL* yang sering dipakai sebagai berikut :

Tabel 2.6 Tipe Data MySQL

Tipe Data	Keterangan
INT (M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT (M,D)	Angka Pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH-MM-SS
CHAR (M)	String dengan panjang tetap sesuai dengan Yang ditentukan. Panjangnya 1-255 Karakter



Lanjutan Tabel 2.6 Tipe Data *MySQL*

VARCHAR (M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1-255 Karakter
BLOB	Teks dengan Panjang Maksimum 65535 Karakter
LONGBLOB	Teks dengan Panjang Maksimum 294967295 Karakter

(Sumber: Pramono, 2015:20)

2.4.4 *HTML (Hypertext Markup Language)*

Faizal dan Irnawati (2015:1) mengemukakan bahwa *Hypertext Markup Language (HTML)* adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web internet* dan pemformatan *hiperteks* sederhana yang di tulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi.

Abdulloh (2016:2) menyatakan bahwa *HTML* singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa *tag-tag* untuk membuat dan mengatur struktur *website*. Beberapa tugas utama *HTML* dalam membangun *website*, di antaranya sebagai berikut:

- a. Menentukan layout *website*.
- b. Mem-*format* teks dasar, seperti pengaturan paragraf dan *format font*.
- c. Membuat *list*.
- d. Membuat tabel.
- e. Menyisipkan gambar, video, dan audio.
- f. Membuat *link*.
- g. Membuat formulir, berikut ini merupakan skrip sederhana *html*:

```
<html>
  <head>
    <title>judul file html</title>
  </head>
  <body> isi dari file html </body>
</html>
```



Penjelasan script *html*

1. Setiap skrip *HTML* selalu diawali dengan tag **<html>** dan ditutup dengan **</html>**, di dalam tag ini dibagi menjadi dua bagian yaitu tag **<head></head>** dan tag **<body></body>**. Skrip yang ditulis antara **<head>** dan **</head>** merupakan skrip yang digunakan untuk pengaturan dokumen *HTML* menuliskan skrip *CSS* dan *Javascript* atau menuliskan skrip untuk memanggil kedua file tersebut. Skrip yang berbeda diantara tag **<body>** dan **</body>** adalah skrip yang mengatur isi dokumen *HTML* yang akan ditampilkan dihalaman *website*.
2. Tag **<title></title>** digunakan untuk membuat judul pada bagian *title bar browser*.
3. Tag **<link>** digunakan untuk memanggil *file CSS* yang mengatur desain dari dokumen *HTML* ini. Tag ini tidak memiliki tag penutup, seperti tag-tag lain. Atribut *rel* harus diisi “**stylesheet**”, sedangkan **href** diisi letak *file CSS* yang dipanggil karena *file CSS* berada di *folder lib* yang berada diluar *folder*, maka ditambah **../** untuk keluar dari *folder* diikuti nama *folder*, yaitu **lib/**, kemudian nama *file CSS*, yaitu misal **admin.css**, jika digabung menjadi “**../lib/admin.css**”.
4. Tag **<div></div>** digunakan untuk membuat bagian-bagian *layout website* yang terpisahkan dalam blok-blok tersendiri.
5. Tag **<p></p>** digunakan untuk membuat sebuah paragraf. Skrip **©** akan membuat simbol *copyright*. Untuk menyisipkan simbol-simbol lain dapat ditemukan dengan mudah di *internet*.

2.4.5 *PHP (Hypertext Preprocessor)*



Gambar 2.4 Logo *PHP*

Raharjo (2016:38) menyatakan bahwa, *PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. Ketika



dipanggil dari *web browser*, program yang ditulis dengan *PHP* akan di-*parsing* didalam *web server* oleh *interpreter PHP* dan diterjemahkan ke dalam dokumen *HTML*, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program *PHP* dilakukan di lingkungan *web server*, *PHP* dikatakan sebagai bahasa sisi *server (server-side)*. Kode *PHP* tidak akan terlihat pada saat *user* memilih perintah “*View Source*” pada *web browser* yang mereka gunakan. Kode *php* diawali dengan *tag <?php* dan diakhiri dengan *tag <?*.

Sedangkan menurut Nugroho (2009:114), *PHP* merupakan bahasa standar yang digunakan dalam dunia *website*, *PHP* adalah bahasa program yang berbentuk skrip yang diletakkan di dalam *server web*. Sedangkan menurut Subagia (2018:1), *PHP* merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah *web server (server side)*.

2.4.6 Skrip PHP Sederhana

Nugroho (2009:120), contoh sederhana, pada program berikut kita hendak menampilkan satu kalimat ke dalam *browser*. Berikut contohnya:

```
<?php
  echo "Selamat pagi DARMAJAYA . . !";
  echo "Terimakasih kesempatannya";
?>
```

2.4.7 PhpMyAdmin

Menurut Madcoms (2016:12), *PhpMyAdmin* adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *phpMyAdmin*, Anda dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng-*update* data dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara manual.

Sedangkan menurut Riyanto (2011:17), *PhpMyAdmin* merupakan aplikasi *web* yang telah banyak digunakan untuk administrasi *database MySQL*. Setelah paket *XAMPP* terinstal di komputer, Anda dapat mengakses *PhpMyAdmin* melalui <http://localhost/phpmyadmin>. Karena kehandalannya, fitur *PhpMyAdmin* tidak hanya diikutkan untuk paket *web server* maya seperti *XAMPP*, tetapi juga banyak digunakan para penyedia *domain-hosting* untuk memudahkan pelanggannya melakukan administrasi *database MySQL*.



2.4.8 CSS (*Cascading Style Sheets*)

Menurut Abdullah (2016:2), CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun *HTML* mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan *HTML* terlihat lebih rapi dan elegan.

2.4.9 Metode Pengujian *Black Box Testing*

Sukanto dan Shalahudin (2013:275), *black box testing* (pengujian kotak hitam), yaitu 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, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.