



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Kadir (2019:2) “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Menurut Wahyudin dan Munir (2018:1) “Komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah input, memberikan informasi, menggunakan suatu program yang tersimpan di memori komputer, dapat menyimpan program dan hasil pengolahan, serta bekerja secara otomatis”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasar urutan instruksi ataupun program yang diberikan kepadanya.

2.1.2 Pengertian Aplikasi

Menurut Asropudin (2013:6) “*Application* atau aplikasi merupakan *Software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*”.

Menurut Sutarman(2013:10) “program aplikasi adalah program-program yang dibuat oleh suatu perusahaan komputer untuk para pemakai yang beroperasi dalam bidang-bidang umum, seperti toko, penerbitan, komunikasi, penerbangan, perdagangan, dan sebagainya”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa aplikasi adalah *Software* atau program-program yang dibuat untuk membantu para pemakai atau *user* dalam mengerjakan tugas-tugas tertentu.



2.1.3 Pengertian Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2018:2) “Perangkat Lunak adalah Program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Menurut Kadir (2019:2) “Perangkat Lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa perangkat lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.

2.1.4 Pengertian Sistem

Menurut Sujatmiko (2012:263) “Sistem adalah Kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu”.

Menurut Arif (2016:11) “Sistem adalah kumpulan dari komponen apapun yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa sistem adalah sekumpulan komponen dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran dan maksud.

2.1.5 Metode Pengembangan Sistem

Sukamto dan Shalahuddin (2018:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut :



- a. Rekayasa dan pemodelan sistem/informasi. Karena perangkat lunak selalu merupakan bagian dari sebuah sistem (bisnis) yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak, manusia dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.
- b. Analisis kebutuhan perangkat lunak. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perancang perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antarmuka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.
- c. Desain. Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi interface dan detail (algoritma) procedural. Proses desain menerjemahkan syarat/kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.
- d. Generasi Kode. Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis
- e. Pengujian. Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional – yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa



input yang dibatasi akan memberikan hasil actual yang sesuai dengan hasil yang dibutuhkan.

- f. Pemeliharaan. Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah perangkat lunak yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasikan perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.2 Teori Judul

2.2.1 Pengertian Sistem

Menurut Sujatmiko (2012:263) “Sistem adalah Kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu”.

Menurut Arif (2016:11) “Sistem adalah kumpulan dari komponen apapun yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa sistem adalah sekumpulan komponen dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran dan maksud.

2.2.2 Pengertian Informasi

Menurut Bodnar (2000:1) “Pengertian informasi adalah sebuah data yang diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat”.

Menurut Melino (1990:331) “Informasi merupakan data yang telah diproses dan diolah untuk tujuan tertentu. Adapun tujuan tersebut untuk menghasilkan sebuah keputusan”.



Berdasarkan definisi di atas maka penulis menyimpulkan bahwa informasi adalah sebuah data yang telah diproses dan diolah yang tujuannya adalah untuk mengambil suatu keputusan

2.2.3 Pengertian Kompensasi

Menurut Nawawi (2011:134) “Kompensasi adalah penghargaan/ganjaran pada para pekerja yang telah memberikan kontribusi dalam mewujudkan tujuannya, melalui kegiatan yang disebut bekerja”.

Menurut Handoko (2014:155) “Kompensasi adalah segala sesuatu yang diterima para karya karyawan sebagai balas jasa untuk kerja mereka”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa kompensasi adalah sesuatu yang diterima oleh seseorang sebagai balasan dari pekerjaan yang sudah dilakukannya.

2.2.4 Pengertian Peraturan

Menurut E. Utrecht, “Aturan ialah sekumpulan petunjuk hidup yang isinya wacana aturan tata tertib dalam kehidupan bermasyarakat serta mempunyai sifat mengikat dan harus ditaati oleh masyarakat yang bersangkutan”.

Menurut Apeldoorn, “Sulit atau tidak memungkinkan untuk memperlihatkan definisi aturan secara niscaya dan juga sesuai dengan kenyataan, lantaran aturan sangat luas aturan yang dicakupnya. Hanya pada tujuan aturan yang mengatur pergaulan hidup secara damai”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa peraturan adalah sekumpulan petunjuk yang di dalamnya berisikan tentang sebuah sifat yang mengikat dan harus ditaati oleh masyarakat.

2.2.5 Pengertian Tata Tertib

Menurut Adwimarta (1992:134) “Tata tertib adalah peraturan-peraturan yang harus ditaati dan dilaksanakan”.



Menurut Suryobroto (2004:81) “Tata tertib adalah ketentuan-ketentuan yang mengatur kehidupan sekolah sehari-hari dan mengandung sanksi terhadap pelanggarannya”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa tata tertib adalah sebuah peraturan yang harus ditaati dan dilaksanakan.

2.2.6 Pengertian Mahasiswa

Menurut Budiman (2006) “Mahasiswa adalah orang yang belajar di sekolah tingkat perguruan tinggi untuk mempersiapkan dirinya bagi suatu keahlian tingkat sarjana”.

Menurut Daldiyono (2009) “Mahasiswa adalah seorang yang sudah lulus dari Sekolah Lanjutan Tingkat Atas (SLTA) dan sedang menempuh pendidikan tinggi”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa mahasiswa adalah orang yang sedang menjalani pendidikan tinggi di perguruan tinggi.

2.2.7 Pengertian Judul Secara Keseluruhan

Aplikasi Sistem Informasi Pelaksanaan Kompensasi Terhadap Peraturan dan Tata Tertib Mahasiswa Politeknik Negeri Sriwijaya (Studi Kasus: Jurusan Manajemen Informatika) adalah sebuah aplikasi atau perangkat lunak berbasis *website* yang berfungsi sebagai akses bagi mahasiswa Jurusan Manajemen Informatika untuk mengakses kompensasi yang di dapatkannya selama 1 semester penuh tanpa harus datang ke admin jurusan untuk meminta lembar kompensasi.

2.3 Teori Khusus

2.3.1 Pengertian *Data Flow Diagram* (DFD)

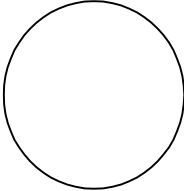
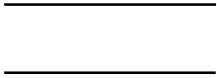
Sukamto dan Shalahuddin (2018:69) menyatakan “*Data Flow Diagram* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.



Adapun simbol–simbol atau notasi–notasi yang menggambarkan *Data Flow Diagram* (DFD), sebagai berikut:



1. Teknik Edward Yourdon dan Tom DeMarco

Tabel 2.1 Simbol–simbol *Data Flow Diagram* (DFD) menurut Edward Yourdon dan Tom DeMarco

No.	Notasi	Keterangan
1.		<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harus menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p>File atau basis data atau penyimpanan (<i>storage</i>); 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, tabel–tabel ini juga harus sesuai dengan perancangan tabel–tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)). Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>



Lanjutan Tabel 2.1 Simbol–simbol *Data Flow Diagram* (DFD)
menurut Edward Yourdon dan Tom DeMarco

No.	Notasi	Keterangan
3.		<p>Entitas luar (<i>External Entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
4.		<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.</p>

(Sumber: Sukamto dan Shalahuddin, 2018:71-72)



Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membentuk 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 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 dan seterusnya.

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.

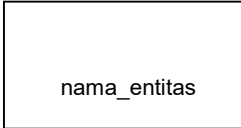
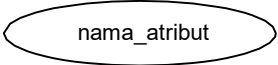
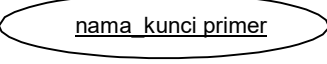
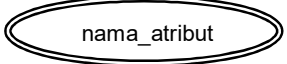
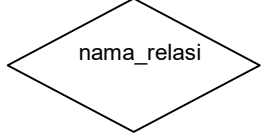
2.3.2 Pengertian *Entity Relationship Diagram (ERD)*

Menurut Sukamto dan Shalahuddin (2018:53) “*Entity Relationship Diagram (ERD)* adalah bentuk paling awal dalam melakukan perancangan basis data relasional”.

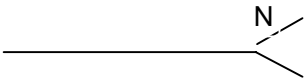


Adapun simbol-simbol yang digunakan dalam *Entity Relationship Diagram* (ERD), yaitu:

Tabel 2.2 Simbol–Simbol *Entity Relationship Diagram* (ERD)

No.	Simbol	Deskripsi
1.	Entitas/ <i>Entity</i> 	Entitas merupakan data inti yang akan tersimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	AtributKunciPrimer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat
4.	AtributMultinilai/ <i>Multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

Lanjutan Tabel 2.2 Simbol–Simbol *Entity Relationship Diagram* (ERD)

No.	Notasi	Keterangan
6.	Asosiasi/ <i>Association</i> 	Penghubung antara relasi dan entitas di mana dikedua ujungnya punya <i>multiplicity</i> ke-mungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut dengan <i>one</i>

(Sumber: Sukanto dan Shalahuddin, 2018:50-51)

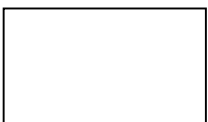

2.3.3 Pengertian Flow Chart

Murhada dan Giap (2013:112-113), “Flowchart adalah bagian-bagian yang mempunyai arus menggambarkan langkah-langkah penyelesaian suatu masalah, merupakan cara penyajian dari suatu algoritma”.

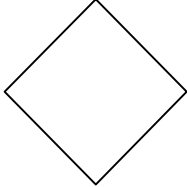



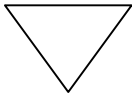

Menurut Kadir (2017:36), “Diagram Air (*Flowchart*) merupakan cara lain untuk menuangkan algoritma”.

Adapun simbol–simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

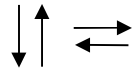

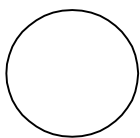
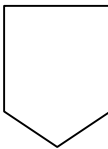

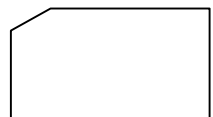
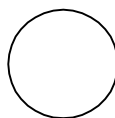
Tabel 2.3 Simbol–Simbol *Flow Chart*

No	Simbol	Nama	Fungsi
Processing Symbols			
1.		Simbol Offline Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		Simbol Manual	Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).

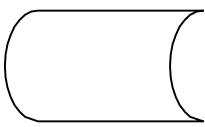


Lanjutan Tabel 2.3 Simbol–Simbol *Flow Chart*

No	Simbol	Nama	Fungsi
Processing Symbols			
3.		Simbol Decision/logika	Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/ tidak.
4.		Simbol Predefined Proses	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		Simbol Terminal	Untuk menyatakan permulaan atau akhir suatu program.
6.		Simbol Keying Operation	Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
7.		Simbol Offline Storage	Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8		Simbol Manual Input	Untuk memasukkan data secara manual dengan menggunakan online keyboard.

Lanjutan Tabel 2.3 Simbol–Simbol *Flow Chart*

No	Simbol	Nama	Fungsi
Flow Direction Symbols			
1.		Simbol arus / flow	Untuk menyatakan jalannya arus suatu proses.
2.		Simbol Communication link	Untuk menyatakan bahwa adanya transisi suatu data/informasi dari suatu lokasi ke lokasi lainnya.
3.		Simbol Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4.		Simbol Offline Connector	Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
Input / Output Symbols			
1.		Simbol Input-output	Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
2.		Simbol Punched Card	Untuk menyatakan input berasal dari kartu atau output dituliskan ke kartu.
3.		Simbol Magnetic-tape Unit	Untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic.

Lanjutan Tabel 2.3 Simbol–Simbol *Flow Chart*

No	Simbol	Nama	Fungsi
Input / Output Symbols			
4.		Simbol Disk Storage	Untuk menyatakan input berasal dari disk atau output disimpan ke disk.
5.		Simbol Document	Untuk mencetak laporan ke printer.
6.		Simbol Display	Untuk menyatakan peralatan output yang digunakan berupa layar (video, komputer).



(Sumber: Murhada dan Giap, 2013:268)

2.3.4 Pengertian *Block Chart*

Kristanto (2018:75), “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol–simbol tertentu”. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

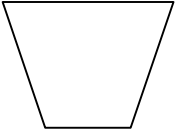

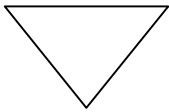
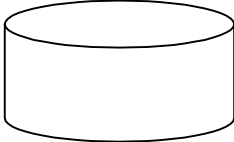

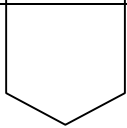
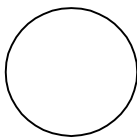

Adapun simbol–simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

Tabel 2.4 Simbol-simbol dalam *Blockchart*

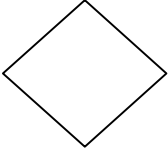


No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen



Lanjutan Tabel 2.4 Simbol-simbol dalam *Blockchart*

No	Simbol	Keterangan
3.		Proses manual
4.		Proses yang dilakukan oleh computer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (data storage)
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

Lanjutan Tabel 2.4 Simbol-simbol dalam *Blockchart*

No	Simbol	Keterangan
11.		Pengambilan keputusan (<i>decision</i>)
12.		Layar peraga (<i>monitor</i>)
13.		Pemasukkan data secara manual

(Sumber: Kristanto, 2018:75-77)

2.3.5 Pengertian Kamus Data (*Data Dictionary*)

Sukamto dan Shalahuddin (2018:69) menyatakan “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 biasanya berisi:

1. Nama, nama dari data
2. Digunakan pada, merupakan proses-proses yang terkait data
3. Deskripsi, merupakan deskripsi data
4. Informasi tambahan, seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.5 Simbol-simbol Kamus Data (*Data Dictionary*)

No.	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan



Lanjutan Tabel 2.5 Simbol-simbol Kamus Data (*Data Dictionary*)

3.	[]	Baik ... atau...
4.	{ } ⁿ	n kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

(Sumber : Sukamto dan Shalahuddin, 2018:74)

2.4 Teori Program

2.4.1 Sekilas Tentang *PHP*

Madcoms (2016:2), “*PHP (Hypertext Preprocessor)* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. *PHP* banyak dipakai untuk membuat program situs web dinamis”.

Nugroho (2013:153), “*PHP* itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis web (website, blog, atau aplikasi web)”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa *PHP* merupakan salah satu pemrograman web *server-side* yang digunakan untuk membuat suatu aplikasi dengan menggunakan editor teks atau editor HTML.

2.4.2 Sintak Dasar *PHP*

PHP merupakan salah satu bahasa yang harus di kuasai *PHP Hypertext Preprocessor* atau di singkat dengan *PHP* ini adalah suatu bahasa *scripting* khususnya di gunakan untuk *web development*.

Berikut ini contoh sintaks dasar penggunaan *PHP* :

```
<!doctype html>
<HTML>
  <HEAD>
    <META charset="utf-8">
    <TITLE>Testing</TITLE>
  </HEAD>
  <?php
```

Echo “Sintaks dasar php”;



```
?>
<BODY>
</BODY>
</HTML>
```

2.4.3 Pengertian XAMPP

Menurut Riyanto (2011:1) “XAMPP merupakan paket PHP dan MySQL berb asis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP”.

Menurut Madcoms (2016:186) “XAMPP adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, Php MyAdmin, PHP, Perl, Filezilla dan lain-lain”.

Berdasarkan definisi di atas maka penulis menyimpulkan bahwa XAMPP adalah sebuah paket PHP berbasis *Open Source* yang digunakan untuk pengembangan aplikasi berbasis PHP.

2.4.4 Pengertian MySQL

Menurut Rusmawan (2019:97) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Databas Management System*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia”.

Menurut Hidayatullah dan Kawistara (2017:175) “MySql Adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Contog DBMS lainnya adalah : PostgreSQL (freeware), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro dsb”.

2.4.4.1 Kelebihan MySQL

1. *Open Source*
2. Handal
3. Selalu *Update*
4. Banyak forum yang memfasilitasi pengguna