



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Erzha (2014:7), “Komputer adalah suatu alat untuk pengolahan informasi sesuai prosedur yang telah dirumuskan.”

Irwansyah, *et.al* (2014:2), “Komputer adalah perangkat elektronik yang beroperasi di bawah kendali instruksi yang memiliki memori, dapat menerima dan mengolah data sesuai dengan aturan yang ditetapkan, menghasilkan hasil atau informasi dan menyimpan hasil tersebut untuk digunakan di masa depan atau di kemudian hari.”

Wahyudi (2012:3), “Komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*).”

Kesimpulannya, komputer adalah alat elektronik yang beroperasi di bawah kendali instruksi yang menerima data dan mengolah data untuk menghasilkan informasi sesuai prosedur yang telah dirumuskan.

2.1.2. Pengertian Sistem

Kristanto (2011:1), “Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.”

Hutahaeen (2014:2), “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu.”

Menurut Jogianto dalam Hutahaeen (2014:1), “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.”

Kesimpulannya, sistem adalah suatu kumpulan dari jaringan, elemen atau prosedur yang saling terhubung yang secara bersama-sama mencapai atau menyelesaikan tujuan tertentu.



2.1.3. Karakteristik Sistem

Hutahaean (2014:3), menjelaskan karakteristik sistem adalah sebagai berikut:

1. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, artinya saling berkerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenace input*), dan masukan sinyal (*signal input*). *Maintenace input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal*



input adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem *computer program* adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem (*Process*)

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, system akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran Sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.1.4. Klasifikasi Sistem

Hutahaean (2014:6), menjelaskan, sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

1. Sistem diklasifikasi sebagai :

a. Sistem abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik.

2. Sistem diklasifikasi sebagai :

a. Sistem alamiah (*natural system*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi.

b. Sistem buatan manusia (*human machine sistem*)



Sistem buatan manusia merupakan sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine sistem*).

3. Sistem diklasifikasi sebagai :

a. Sistem tertentu (*deterministic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan.

b. Sistem tak tentu (*probabilistic system*)

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

4. Sistem diklasifikasi sebagai :

a. Sistem tertutup (*close system*)

Sistem tertutup adalah sistem yang tidak terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja otomatis tanpa ada turut campur lingkungan luar.

b. Sistem terbuka (*open system*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima *input* dan *output* dari lingkungan luar atau subsistem lainnya.

2.1.5. Metode Pengembangan Sistem

Mulyani (2009:1), menjelaskan *SDLC* adalah langkah-langkah dalam pengembangan sistem informasi. *SDLC* menyediakan *framework* yang lengkap untuk aktivitas rekayasa bentuk dan pembangunan sistem informasi yang formal. Langkah-langkah *SLDC* meliputi antara lain:

1. Perencanaan sistem Informasi

Perencanaan sistem informasi akan memberikan manfaat dalam bentuk:

- a. Pendefinisian ruang lingkup pengembangan sistem informasi.
- b. Identifikasi potensial masalah.
- c. Pengaturan urutan tugas pengembangan sistem informasi.
- d. Pengendalian.



2. Analisa sistem Informasi

Analisa sistem informasi adalah telaah atas sistem berjalan dengan tujuan untuk mendesain sistem baru atau menyempurnakan sistem lama. Rincian tujuan dari tahapan analisis sistem informasi adalah untuk:

- a. Membuat keputusan apabila sistem saat ini mempunyai masalah atau sudah tidak berfungsi secara baik dan hasil analisisnya digunakan sebagai dasar untuk memperbaiki sistem.
- b. Mengetahui ruang lingkup pekerjaannya yang akan ditanganinya.
- c. Memahami sistem yang sedang berjalan saat ini.
- d. Mengidentifikasi masalah dan mencari solusinya.

3. Desain Sistem Informasi

Desain sistem informasi adalah penentuan proses dan kebutuhan data dari sistem yang baru. Tahap-tahapan desain sistem informasi meliputi:

- a. Penyiapan Rincian Desain Sistem.
- b. Mengidentifikasi Konfigurasi Alternatif Sistem.
- c. Mengevaluasi Konfigurasi Alternatif Sistem. Tujuan dari tahapan ini adalah membandingkan alternatif-alternatif sistem dengan menggunakan metodologi terstruktur. Hasil dari tahapan ini adalah hasil-hasil dari studi sistem.
- d. Memilih Konfigurasi Sistem yang Terbaik.
- e. Menyiapkan Proposal Implementasi.
- f. Menerima atau Menolak Implementasi Sistem.

4. Implementasi Sistem Informasi

Implementasi sistem informasi adalah penentuan proses dan kebutuhan data dari sistem yang baru. Tahap-tahapan implementasi sistem informasi meliputi:

- a. Perencanaan Implementasi.
- b. Mengkomunikasikan Implementasi.
- c. Memperoleh Sumberdaya *Hardware*.
- d. Memperoleh Sumberdaya *Software*.
- e. Menyiapkan *Database*.
- f. Menyiapkan Fasilitas Fisik.



g. Pelatihan.

Pelatihan terdiri dari 2 (dua) jenis, yaitu:

- 1) Pelatihan Kelas.
- 2) Asistensi.

h. Menyiapkan Proposal *Cutover*.

Cutover sistem lama oleh sistem baru harus jelas. Tujuan dari *cutover* adalah merubah pemakaian sistem lama ke sistem baru dari sistem informasi yang berhasil dibangun. Perubahan sistem merupakan tanggungjawab tim designer ke pemakai. Hasil dari *cutover* adalah rencana (jadwal dan metode) perubahan sistem.

i. Menerima atau Menolak *Cutover* Sistem.

j. Melaksanakan *Cutover* Sistem.

5. Pemakaian Sistem Informasi.

Tahap-tahapan pemakaian sistem informasi meliputi:

a. Penggunaan Sistem.

b. Audit Sistem.

c. Pemeliharaan Sistem. Meliputi:

- 1) Perbaikan *Errors*.
- 2) Modifikasi.
- 3) Penyempurnaan Sistem.

d. Menyiapkan Proposal Rekrayasa Sistem.

e. Menerima atau Menolak Rekrayasa Sistem.

2.1.6. Pengertian *Database*

Badiyanto (2013:57), “*Database* adalah suatu kumpulan dari data yang tersimpan dalam tabel dan diatur atau diorganisasikan sehingga data tersebut bias diambil atau dicari dengan mudah dan efisien.”

Tim EMS (2014:54), “*Database* adalah tempat untuk menyimpan data dan nantinya data ini bias diambil lagi.”



2.1.7. Pengertian Kependudukan

Faqih (2010:3), “Kependudukan atau demografi adalah ilmu yang mempelajari persoalan dan keadaan dinamika kependudukan manusia, meliputi di dalamnya ukuran, struktur, dan distribusi penduduk, serta bagaimana jumlah penduduk berubah setiap waktu akibat kelahiran, kematian, perkawinan, migrasi dan mobilitas sosial.

2.1.8. Pengertian *Online*

Online adalah keadaan komputer yang terkoneksi/ terhubung ke jaringan Internet. Sehingga apabila komputer kita online maka dapat mengakses internet/ browsing, mencari informasi-informasi di internet.

<http://www.temukanpengertian.com/2013/06/pengertian-online-online-adalah-online.html>/ diakses pada tanggal 11 Juni 2016.

2.1.9. Pengertian Sistem Kependudukan Online pada Kelurahan 20 Ilir Daerah IV Palembang

Sistem Kependudukan Online pada Kelurahan 20 Ilir Daerah IV Palembang adalah sebuah website yang berisi tentang informasi kependudukan pada Kelurahan 20 Ilir Daerah IV Palembang, yang dapat digunakan masyarakat untuk mengetahui informasi mengenai angka kelahiran, angka kematian, jumlah pindah dan jumlah datang pada setiap bulannya, serta melayani kebutuhan-kebutuhan masyarakat dimulai dari pembuatan Surat Keterangan kematian, kelahiran, kepindahan, dan pendatang.



2.2. Teori Khusus

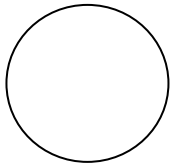
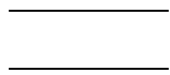
2.2.1. Pengertian *Data Flow Diagram* (DFD)

Sutabri (2012:117) menjelaskan “Data Flow Diagram adalah network yang menggambarkan suatu sistem automat/ komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.”


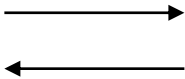
DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

Rosa, et.al (2013:71), menjelaskan notasi pada DFD adalah sebagai berikut :

Tabel 2.1 Simbol-simbol dalam *Data Flow Diagram*

No.	Simbol	Keterangan
1		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja
2		<i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan 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

Lanjutan Tabel : **Tabel 2.1** Simbol-simbol dalam *Data Flow Diagram*

No.	Simbol	Keterangan
		<p>basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM))</p> <p>Catatan :</p> <p>Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
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 :</p> <p>Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) 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 :</p> <p>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 : Rosa, et.al (2013:71))



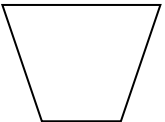

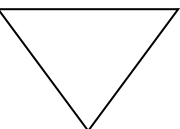
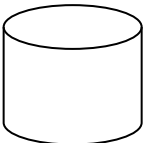

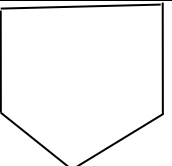


2.2.2. Pengertian *Blockchart*

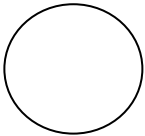
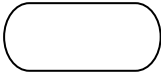
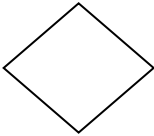


Kristanto (2011:68) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Adapun simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel 2.2. berikut ini:

Tabel 2.2. Simbol-simbol dalam *Blockchart*

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

Lanjutan Tabel: **Tabel 2.2.** Simbol-simbol dalam *Blockchart*

No	Simbol	Keterangan
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukkan data secara manual.

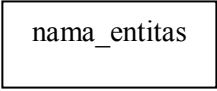
(Sumber: Kristanto (2011:68))

2.2.3. Pengertian *Entity Relational Diagram (ERD)*

Rosa, et.al (2013:50), “*ERD* adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. *ERD* digunakan untuk pemodelan basis data relasional.”

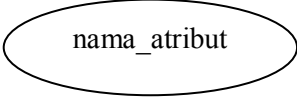
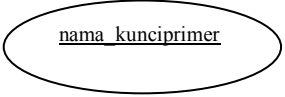
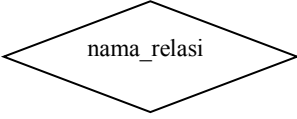

Simbol-simbol yang digunakan dalam *ERD* dapat dilihat di dalam table 2.3 berikut:

Tabel 2.3. Simbol-simbol dalam *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya adar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.



Lanjutan Tabel: **Tabel 2.3.** Simbol-simbol dalam *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	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; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
4.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
5.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

(Sumber : Rosa, et.al (2008 :75))



2.2.4. Pengertian *Flowchart*



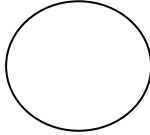
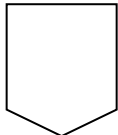
Ladjamudin (2013:263), ”*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma”.

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagimenjadi 3 kelompok adalah sebagai berikut :

1. *Flow Direction Symbols* (Simbol penghubung/ alur)
2. *Processing Symbols* (Simbol proses)
3. *Input-output Symbols* (Simbol Input-Output)



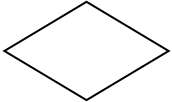
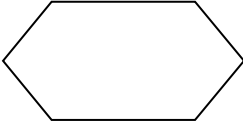

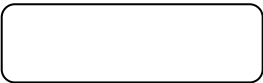
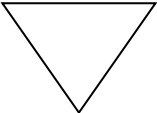
Simbol-simbol yang digunakan *flowchart* dapat dilihat di dalam table 2.4 berikut:

Tabel 2.4. Simbol-simbol Penghubung/Alur(*Flow Direction Symbols*)


Simbol	Deskripsi
	<p>Simbol Arus/ <i>Flow</i> Menyatakan jalannya arus suatu proses</p>
	<p>Simbol <i>Communication Link</i> Menyatakan bahwa adanya transisi suatu data/ informasi dari satu lokasi ke lokasi lainnya</p>
	<p>Simbol <i>Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama</p>
	<p>Simbol <i>Offline Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman /lembar yang berbeda</p>

Sumber : Ladjamudin (2013:266)


Tabel 2.5. Simbol-simbol Proses (*Processing Symbols*)

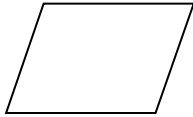
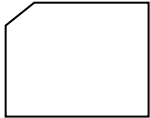
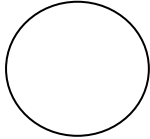


Simbol	Deskripsi
	Simbol <i>Offline Connector</i> Menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda
	Simbol <i>Manual</i> Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual)
	Simbol <i>Decision/ Logika</i> Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak
	Simbol <i>Predifined Proses</i> Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
	Simbol <i>Terminal</i> Meyatakan permulaan atau akhir suatu program
	Simbol <i>Keying Operation</i> Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
	Simbol <i>Off-line Storage</i> Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu

Lanjutan Tabel 2.5. Simbol-simbol Proses (*Processing Symbols*)

Simbol	Deskripsi
	<p>Simbol Manual Input</p> <p>Memasukkan data secara manual dengan menggunakan online keyboard</p>

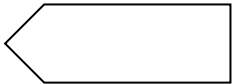
Sumber : Ladjamudin (2013:267)

Tabel 2.6. Simbol-simbol Input-Output (*Input-Output Symbols*)

Simbol	Deskripsi
	<p>Simbol <i>Input-Output</i></p> <p>Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya</p>
	<p>Simbol <i>Punched Card</i></p> <p>Menyatakan input berasal dari kartu atau output ditulis ke kartu</p>
	<p>Simbol <i>Magnetic-tape unit</i></p> <p>Menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic</p>
	<p>Simbol <i>Disk Storage</i></p> <p>Menyatakan input berasal dari disk atau output disimpan ke disk</p>
	<p>Simbol <i>Document</i></p> <p>Untuk mencetak laporan ke printer</p>



Lanjutan Tabel 2.6. Simbol-simbol Input-Output (*Input-Output Symbols*)

Simbol	Deskripsi
	<p>Simbol <i>Display</i> Menyatakan peralatan output yang digunakan berupa layar (video, komputer)</p>

Sumber : Ladjamudin (2013:268)

2.2.5. Pengertian Kamus Data

Rosa, et.al (2013:73), “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).

Fungsi kamus data adalah untuk mengetahui *field* dari tabel-table yang ada di dalam database secara lebih detail termasuk tipe *field* dan panjang karakter tulisan yang ada pada masing-masing *field*.

Simbol-simbol yang digunakan kamus data dapat dilihat di dalam table 2.5 berikut:

Table 2.7. Simbol-simbol dalam Kamus Data

No.	Simbol	Arti
1.	=	disusun atau terdiri atas
2.	+	Dan
3.	[]	baik ...atau...
4.	{ ⁿ }	n kali diulang/ bernilai banyak
5.	()	data operasional
6.	*...*	batas komentar

(Sumber : Rosa dan Shalahuddin (2013:74))



2.3. Teori Program

2.3.1. Sekilas Tentang PHP

Utomo (2014:2), *PHP* dirancang oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1994, *PHP* digunakan untuk mendeteksi *user* yang berkunjung pada situs. Selain sebagai bahasa pemrograman *web* yang dapat dieksekusi sendiri, *PHP* juga dikenal sebagai *embedded language*. Artinya, anda dapat memasukkan kode *PHP* ke dalam bahasa *HTML* yang sebelumnya dikenal sebagai bahasa pembentuk halaman *website*.

2.3.1.1. Pengertian PHP (*Hypertext Preprocessor*)

Abdulloh (2015:3), “*PHP* singkatan dari *Hypertext Preprocessor* merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi server”.

Badiyanto (2013:32), “*PHP (Hypertext Preprocessor)* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML/ PHP* banyak dipakai untuk membuat situs *web* dinamis.”

Tim EMS (2014:56), “*PHP* adalah singkatan dari *PHP Hypertext Preprocessor* merupakan bahasa scripting seperti *HTML* yang lazim digunakan di halaman *web*.”

2.3.1.2. Tipe Data PHP

Tim EMS (2014:72) menjelaskan tipe data yang dapat diakomodasi di PHP, yaitu :

1. Integer

Integer adalah tipe data paling standar, ini hanya menyatakan kebenaran, apakah TRUE (benar) atau FALSE (salah). Untuk menyatakan sebuah boolean, anda bisa menggunakan kata kunci TRUE atau FALSE.

2. Integer

Sebuah integer adalah bilangan bulat (bukan pecahan), baik negatif atau positif. Misalnya {...,-2, -1, 0, 1, 2, ...}.



Integer bisa dituliskan dalam satuan desimal (berbasis 10), heksadesimal (berbasis 16), atau oktal (berbasis 8) dan bisa juga ditambahi tanda plus atau minus (- atau +) tapi penandaan ini opsional saja.

3. Floating Point

Floating point atau nomor pecahan atau juga bilangan real, bisa didefinisikan dengan syntax berikut ini :

```
<?php
$a = 3.652;
$b = 3.2.e3;
$c = 7E - 10;
?>
```

4. String

String adalah untaian karakter, sehingga karakter sama seperti byte. Ada 256 karakter yang bisa dijadikan string. Sebuah string bisa didefinisikan dengan berbagai cara, tapi yang paling mudah adalah dengan satu petik.

5. Null

Nilai NULL mempresentasikan variabel tanpa value. NULL adalah nilai yang mungkin untuk NULL. Sebuah variabel akan menjadi null jika :

- a. Di alokasikan konstanta NULL
- b. Variabel yang belum diset ke nilai apapun.
- c. Menerima unset().

2.3.1.3. Skrip *PHP*

Utomo (2014:18), “*PHP* merupakan *embedded language* pada *HTML*. *PHP* dapat dimasukkan pada kode *HTML* dalam satu *file*. Berikut ini contoh sintaks dasar penggunaan *PHP* :

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

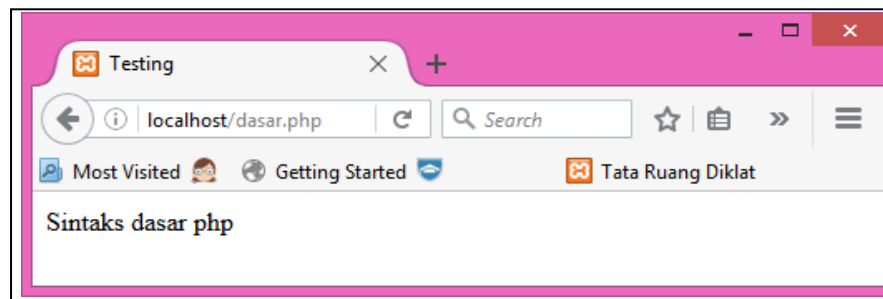


```

</HEAD>
    <?php
        Echo "Sintaks dasar php";
    ?>
<BODY>
</BODY>
</HTML>

```

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.1. Tampilan *PHP* Sederhana

2.3.2. Pengertian *HTML*

Abdulloh (2015:2), “*HTML* singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat mengatur struktur website”.

Kadir (2013:20), “*HTML (Hypertext Markup Language)* adalah bahasa skrip yang digunakan untuk menyusun halaman *web*.”

Tim EMS (2014:2), “*HTML* adalah singkatan dari *Hypertext Markup Language*. Bahasa pemograman ini terdiri dari *tag* dan aturan-aturan yang memungkinkan membuat dokumen *hypertext*.”

2.3.3. Pengertian *MySQL*

Faizal, *et.al* (2015:4), “*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa inggris :*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 (enam) juta instalasi di seluruh dunia”.



Prasetio (2014:29), “*MySQL* adalah sebuah *database*, dan dapat anda bayangkan seperti sebuah tempat penyimpanan.”

Utdirartatmo (2002:3), “*MySQL* adalah sistem manajemen *database* relational. Suatu *database* relational menyimpan data dalam tabel-tabel terpisah.”

2.3.4. *JavaScript*

Kadir (2013:58), “*Java Script* adalah bahasa skrip yang digunakan untuk mengontrol tindakan-tindakan yang diperlukan di halaman web.

Wahana Komputer (2012:2), “*Java Script* adalah salah satu bahasa *script website* yang paling banyak digunakan untuk menambah manipulasi *script HTML* dan *CSS* pada sisi *client/browser*.

Javascript merupakan bahasa pemrograman client side. Kode Javascript dituliskan dalam bentuk fungsi yang ditaruh di tag <head> yang dibuka dengan tag <script type=”teks/javascript” src=”alamat.js”> dan diakhir dengan tag </script>.

3.3.4.1. Variabel dan Operator *JavaScript*

Wahana Komputer (2012:2) menjelaskan variabel dan operator yang biasa digunakan *Java Script*:

1. Variabel

Wahana Komputer (2012:48), “Variabel adalah sebuah *identifier/* pengenal/ nama yang mempunyai nilai yang dapat berubah-ubah. Variabel merupakan bagian dalam sebuah bahasa pemrograman yang digunakan untuk menyimpan data sementara waktu.”

Variabel dapat dideklarasikan dengan *keyword var* atau tanpa *keyword*. Pendeklarasian variabel dengan *keyword var* pada suatu blok akan menyebabkan *variabel* itu hanya dapat diakses pada blok itu atau disebut variabel lokal. Sedangkan pendeklarasian variabel tanpa *var* dimana pun.



2. Operator

Operator adalah simbol yang digunakan untuk melakukan operasi dua nilai atau lebih.”

- a. Operator *Assignment* adalah operator yang digunakan untuk memberikan nilai pada suatu variabel. Operator *assignment* asli disimbolkan dengan karakter = (sama dengan). Operator tersebut dapat digabungkan dengan operator aritmatik sehingga dapat melakukan operasi *arithmetic*.
- b. Operator *Aritmatic* adalah operator yang digunakan untuk melakukan operasi matematika, baik penjumlahan, pengurangan, perkalian, pembagian, dan modulo.
- c. Operator Logika adalah operator yang digunakan untuk melakukan operasi *boolean* (*true* dan *false*) atau (1 dan 0). Pada *Java Script* terdapat 3 buah operator logika, yaitu *AND* (&& dan &), *OR* (|| dan |), dan *NOT* (!).
- d. Operator *Conditional* adalah operator yang mempunyai 3 buah ekspresi. Ekspresi pertama adalah nilai *boolean*. Ekspresi kedua dijalankan apabila ekspresi pertama bernilai *true*. Ekspresi ketiga akan dijalankan apabila ekspresi pertama bernilai *false*.
- e. Operator *String* adalah operator yang digunakan untuk melakukan penggabungan *string*. Hanya ada satu operator, yaitu +. Apabila terjadi operasi antara tipe *int* dengan *string* maka dilakukan operasi string dan hasilnya dalam bentuk *string*.
- f. Operator Perbandingan adalah operator yang digunakan untuk membandingkan sebuah nilai. Jika dua buah nilai memenuhi operasi perbandingan maka akan menghasilkan nilai *true*. Sebaliknya, jika tidak memenuhi akan menghasilkan nilai *false*.