



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Sujatmiko (2012:156), “Komputer merupakan mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program.”

Fahmi, dkk (2016:2), “Komputer *to computer* (bahasa Inggris) atau *computare* (bahasa Latin) yang artinya menghitung adalah sekumpulan alat *logic* yang dapat menerima data, mengolah data dan menyimpan data dengan menggunakan program yang terdapat pada memori sistem komputer kemudian memberikan hasil pengolahan tersebut dalam bentuk *output*.”

##### 2.1.2. Pengertian Aplikasi

Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu.”

Sutabri (2012:148), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.”

##### 2.1.3. Pengertian Sistem Informasi

Sutabri (2012:46), sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Ladjamudin (2013:13), sistem Informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan untuk menyajikan informasi.

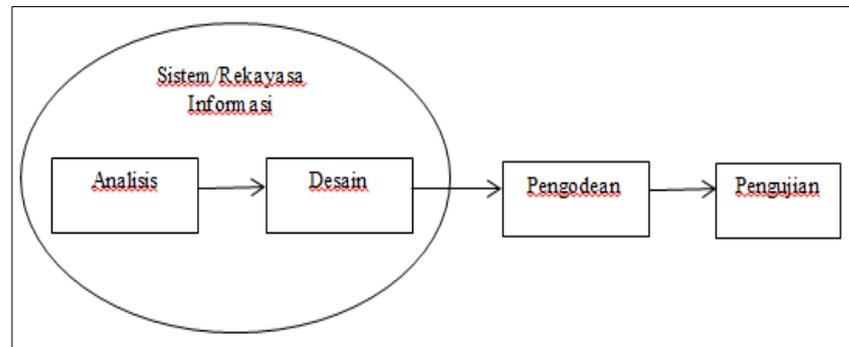


Dari kedua pengertian di atas, penulis dapat menyimpulkan bahwa sistem informasi adalah suatu sistem yang dihasilkan dari berbagai komponen pendukung yang saling berhubungan satu sama lain guna menghasilkan suatu informasi yang bermanfaat dalam pengambilan keputusan.

#### 2.1.4. Metode Pengembangan Sistem

Soekamto Salahuddin (2014:28), menjelaskan metode pengembangan sistem yang sering digunakan dalam tahapan pengembangan sistem yaitu model SDLC air terjun (waterfall) sering juga disebut model sekuensial atau alur hidup klasik (*classic life cycle*) dengan tahapan sebagai berikut:

- a. Analisis kebutuhan perangkat lunak  
Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
- b. Desain  
Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan orogram perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.
- c. Pembuatan kode program  
Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
- d. Pengujian  
Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji.
- e. Pendukung (*support*) atau pemeliharaan (*maintenance*)  
Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.



**Gambar 2.1** Ilustrasi Model Waterfall (Sumber: Rosa dan Salahuddin, 2014:29)

### 2.1.5. Pengertian Pelayanan

Suparlan (2000:35), “Pelayanan adalah usaha pemberian bantuan atau pertolongan kepada orang lain, baik berupa materi maupun non materi agar orang yang itu dapat mengatasi masalahnya sendiri.”

Moenir (2005:47), “Pelayanan adalah proses pemenuhan kebutuhan melalui aktivitas orang lain secara langsung.”

### 2.1.6. Pengertian Publik

Jefkins (2003), “Publik adalah sebagai kelompok atau orang yang berkomunikasi dengan organisasi, baik secara internal maupun eksternal.”

Kriyantono (2012), “Publik adalah sebagai sekumpulan orang/kelompok dalam masyarakat yang memiliki kepentingan atau perhatian yang sama terhadap suatu hal.”

### 2.1.7. Pengertian Balai Besar Perikanan Budidaya Laut (BBPBL) Lampung

Pelayanan Publik pada Balai Besar Perikanan Budidaya Laut (BBPBL) Lampung adalah salah satu instansi pemerintahan yang bergerak di bidang kelautan dan perikanan yang bertugas pengelolaan produksi, pengujian laboratorium, mutu pakan ikan, kesehatan ikan, dan lingkungan kehidupan ikan, serta bimbingan teknis perikanan budidaya laut.

### 2.1.8. Pengertian Aplikasi Pelayanan Publik pada Balai Besar Perikanan Budidaya Laut (BBPBL) Lampung

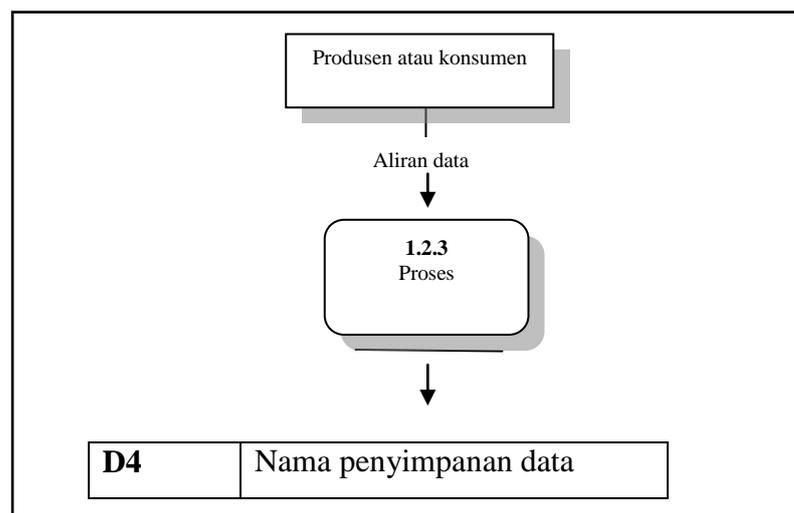
Aplikasi Pelayanan Publik pada Balai Besar Perikanan Budidaya Laut (BBPBL) Lampung adalah suatu aplikasi yang memberi kemudahan kepada masyarakat untuk menerima informasi di Besar Perikanan Budidaya Laut (BBPBL) Lampung.

## 2.2. Teori Khusus

### 2.2.1. Pengertian *Data Flow Diagram (DFD)*

Ladjamudin (2013:64), “*Data Flow Diagram (DFD)* merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil.”

Sukanto Shalahuddin (2014:69), “*Data Flow Diagram (DFD)* awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology (SSADM)* yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah system.” Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson:



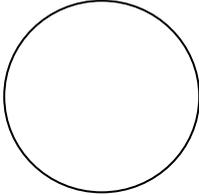
**Gambar 2.2** Contoh DFD yang dikembangkan Chris Gane dan Trish Sarson



Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

Sukanto Shalahuddin (2014:71) menjelaskan, “Notasi- notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

**Tabel 2.1** Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>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.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau <i>basisdata</i> 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 basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>)</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang</p>

	<p>memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>
	<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 Shalahuddin (2014:71)

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 system lain. *DFD Level 0* digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat *DFD Level 1*

*DFD Level 1* digunakan untuk menggambarkan modul-modul yang ada dalam system 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 dua*. 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 di-*breakdown* lagi. Untuk sebuah sistem, jumlah *DFD Level 2* sama dengan jumlah modul pada *DFD Level 1* yang di-*breakdown*.

#### 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.

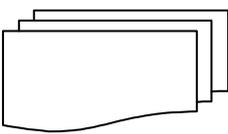
Pada satu diagram *DFD* sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

### 2.2.2. Pengertian *Block Chart*

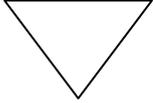
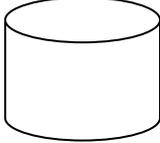
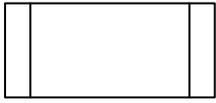
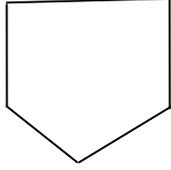
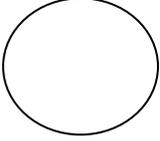
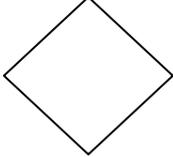
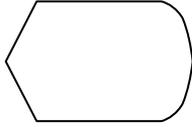
Kristanto (2011:65), "*BlockChart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *BlockChart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi."

Kristanto (2011:68) menjelaskan, "Simbol-simbol yang sering digunakan dalam *BlockChart* dapat dilihat pada tabel berikut ini:

**Tabel 2.2** Simbol-simbol dalam *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> )
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas 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 ( <i>monitor</i> ).

13.		Pemasukan data secara manual.
-----	-----------------------------------------------------------------------------------	-------------------------------

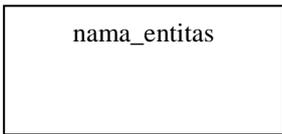
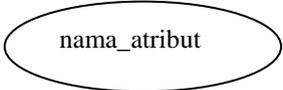
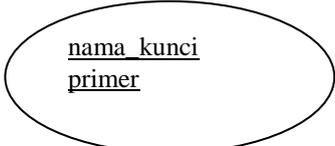
Sumber : Kristanto (2011:68)

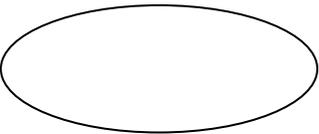
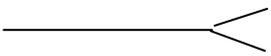
### 2.2.3. Pengertian ERD (Entity Relationship Diagram)

Sukanto Shalahuddin (2014:50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan menggunakan *Entity Relationship Diagram (ERD)*. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

Sukanto Shalahuddin (2014:71) menjelaskan, berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel 2.3** Simbol-simbol ERD dengan Notasi Chen

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada

	yang sama).
Atribut multinilai / <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
Asosiasi / <i>association</i> N 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

Sumber : Sukamto dan Shalahuddin (2014:71)

#### 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."

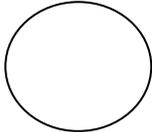
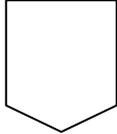


### 2.2.4.1. Simbol-simbol *Flowchart*

*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)

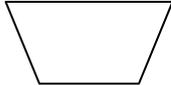
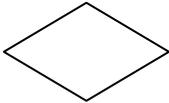
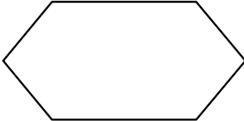
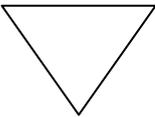
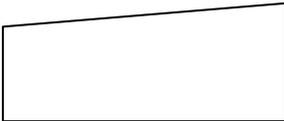
**Tabel 2.4** Simbol-simbol Penghubung/Alur(*Flow Direction Symbols*)

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

Sumber : Ladjamudin (2013:263)

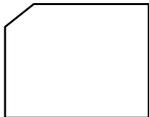
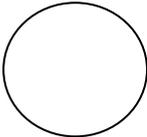
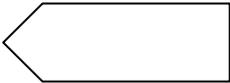
**Tabel 2.5** Simbol-simbol Proses (*Processing Symbols*)

Simbol	Deskripsi
	<b>Simbol Offline Connector</b> Menyatakan sambungan dari satu proses

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

Sumber : Ladjamudin (2013:263)


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

Simbol	Deskripsi
	<p><b>Simbol <i>Input-Output</i></b> Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya</p>
	<p><b>Simbol <i>Punched Card</i></b> Menyatakan input berasal dari kartu atau output ditulis ke kartu</p>
	<p><b>Simbol <i>Magnetic-tape unit</i></b> Menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic</p>
	<p><b>Simbol <i>Disk Storage</i></b> Menyatakan input berasal dari disk atau output disimpan ke disk</p>
	<p><b>Simbol <i>Document</i></b> Untuk mencetak laporan ke printer</p>
	<p><b>Simbol <i>Display</i></b> Menyatakan peralatan output yang digunakan berupa layar (video, komputer)</p>

Sumber : Ladjamudin (2013:263)

### 2.2.5. Pengertian Kamus Data

Sukamto Shalahuddin (2014:73), “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada *DFD*. Kamus data adalah kumpulan daftar elemen data yang mengalir pada system 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

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

**Tabel 2.7** Simbol-simbol dalam Kamus Data

Simbol	Keterangan
=	disusun atau terdiri dari
+	Dan
[]	baik ...atau...
{ <sup>n</sup> }	n kali diulang/ bernilai banyak
()	data opsional
*...*	batas komentar

**Sumber :** Sukamto dan Shalahuddin(2014:74)

## 2.3. Teori Program

### 2.3.1. Pengertian Basis Data (*Database*)

Raharjo (2015:2), “Basis Data didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara tepat.”

Kristanto (2011:72), “Basis Data adalah kumpulan data yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.”

Sukanto Shalahuddin (2014:43), “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

### 2.3.2. Pengertian *HTML* (*Hypertext Markup Language*)

Tim Ems (2016:2), “*Hyper Text Markup Language (HTML)* adalah bahasa pemrograman yang terdiri dari tag dan aturan-aturan yang memungkinkan anda membuat dokumen *hypertext*.”

Faizal, dkk (2015:1), “*HTML* adalah sebuah bahasa markah yang digunakan untuk membuat halaman web, menampilkan berbagai informasi didalam sebuah penjelajah web internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi.”

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

### 2.3.3. Pengertian *PHP* (*Hypertext Preprocessor*)

Madcoms (2016:17), “*PHP* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam *HTML* dan bahasa pemrograman script *server-side*.”

Tim Ems (2016:2), “*PHP* merupakan jenis bahasa scripting yang lazim digunakan di halaman *web*. Artinya, kode ini langsung dimasukkan ke dalam kode *html*.”

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



#### **2.3.4. Pengertian *MySQL* (*My Structure Query Language*)**

Raharjo (2015:16),”*MySQL* merupakan software RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*).”

Faizal, dkk (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.”

Madcoms (2016:17),”*MySQL* adalah sistem manajemen database *SQL* yang bersifat *Open Source* dan paling populer saat ini.”

#### **2.3.5. Pengertian *Xampp***

Winarno, dkk (2014:1),”*Xampp* adalah *software web server* yang bisa dipakai untuk mengakomodasi sistem operasi yang anda pakai (X), *Apache* (A), *MySQL* (M), *PHP* (P), dan *Perl* (P).”

Madcoms (2016:148), “*Xampp* adalah sebuah paket kumpulan *software* yang terdiri *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla* dan lain-lain.”