



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Kadir (2017:2) mengemukakan bahwa, komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia.

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

Dari beberapa pendapat diatas dapat disimpulkan komputer adalah alat yang dipakai untuk mengolah data berdasarkan prosedur yang telah dirumuskan.

##### 2.1.2 Pengertian Internet

Sujatmiko (2012:138), “*Internet* adalah jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit.”

Yugianto dan Rachman (2012:36), “*Internet* adalah suatu sistem jaringan komunikasi beberapa komputer yang terhubung tanpa batas waktu maupun tempat, sehingga dapat dikatakan sebagai komunitas jaringan global.”

Kesimpulannya, Internet adalah jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit yang terhubung tanpa batas waktu maupun tempat.

##### 2.1.3 Pengertian Perangkat Lunak (*Software*)

Menurut Ladjamuddin (2013:20), “*Software* merupakan kumpulan dari perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu”.

Rosa dan Shalahuddin (2014:02), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumantasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Kesimpulannya, Perangkat Lunak adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer yang terasosiasi dengan dokumantasi perangkat lunak seperti dokumentasi kebutuhan.



### 2.1.4 Pengertian Perangkat Keras (*Hardware*)

Kadir (2017:2) mengemukakan bahwa, perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik. Monitor, *hard disk*, dan mouse adalah contoh perangkat keras.

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.

Dari beberapa pendapat di atas dapat disimpulkan perangkat Keras (*Hardware*) adalah elemen atau perangkat fisik komputer yang beroperasi di dalamnya.

### 2.1.5 Pengertian Sistem

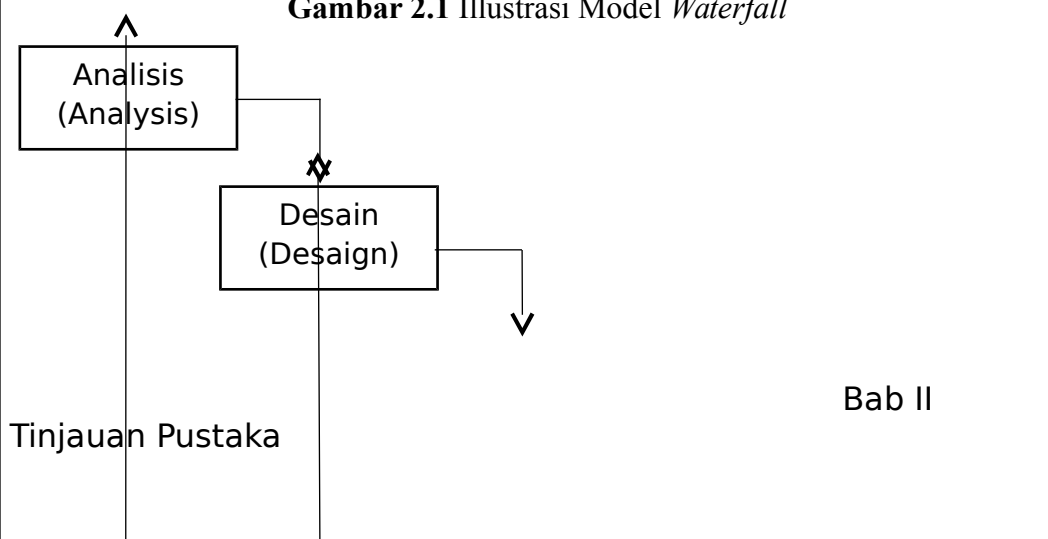
Indrajani (2015:69) “Sistem adalah sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur.”

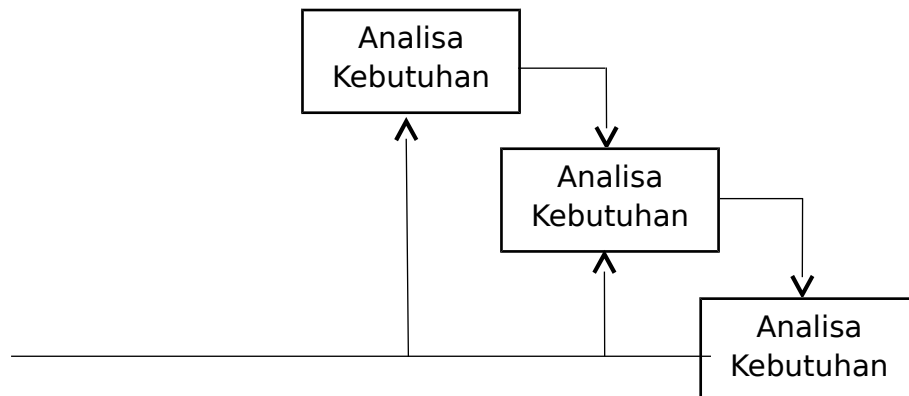
Jeperson (2015:2) “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melaksanakan kegiatan atau untuk melakukan sasaran yang tertentu.”

Kesimpulannya, sistem adalah kumpulan-kumpulan dari berbagai bagian elemen yang saling berhubungan berdasarkan fungsinya agar menjadi satu kesatuan.

### 2.1.6 Metode Pengembangan Sistem

Gambar 2.1 Ilustrasi Model *Waterfall*





Menurut Sukamto 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, pengodean, pengujian, dan pemeliharaan.

#### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami pperangkat 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 deain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

#### 3. Pembuatan kode program

Pada tahap pengodean, desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desaian 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.

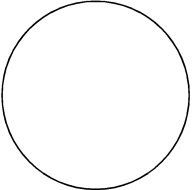
## 2.2 Teori Khusus

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

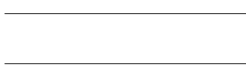

Rosa dan Shalahuddin (2012:70) menjelaskan, “*Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*ouput*).

Adapun notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :


**Tabel 2.1** Simbol-simbol *Data Flow Diagram*

Notasi	Keterangan
	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
	<p>File basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harusnya 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))</p> <p>Catatan : Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
<b>Notasi</b>	<b>Keterangan</b>
	<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>



	<p>Aliran data: merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses 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 : Rosa dan Shalahuddin, 2012:71-72)

Rosa dan Shalahuddin (2012:72-73) menjelaskan, Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 (nol) atau sering disebut juga Context Diagram  
DFD Level 0 (nol) menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 (nol) 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 (nol) yang sebelumnya sudah dibuat.
3. Membuat DFD Level 2  
Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2 (dua) Modul mana saja yang harus di-*breakdown* lebih 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 ada DFD Level 1 (satu) yang di-*breakdown*.
4. Membuat DFD Level 3 dan seterusnya  
DFD Level 3,4,5 dan seterusnya merupakan *breakdown* dari modul pada DFD di atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.



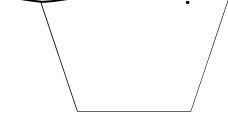

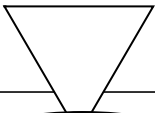




### 2.2.2 Pengertian *Block chart*

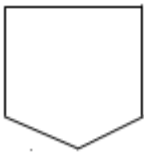
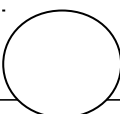
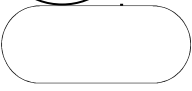
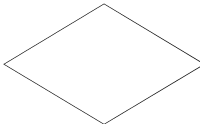


Kristanto (2008:68) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu”. Pembuatan blockchart harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol yang sering digunakan dalam block chart dapat dilihat pada tabel berikut ini:

**Tabel 2.2** Simbol-simbol *Block chart*

Simbol	Arti
	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/berkas atau cetakan.
	Multi Dokumen
	Proses Manual
	Proses yang dilakukan oleh komputer
	Menandakan dokumen yang diarsipkan (arsip manual)
	Data penyimpanan ( <i>data storage</i> )
	Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik



	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
	Terminasi yang menandakan awal dan akhir dari suatu aliran
	Pengambilan keputusan ( <i>decision</i> )
	Layar peraga ( <i>monitor</i> ).
	Pemasukkan data secara manual.

(Sumber : Kristanto, 2008:68-70)

### 2.2.3 Pengertian ERD (*Entity Relational Diagram*)

Rosa dan Shalahuddin (2013:50) menjelaskan, “*Entity Relational Diagram* (ERD) digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lain.

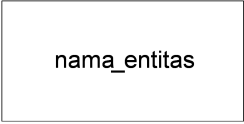
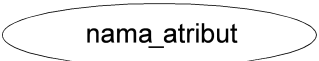

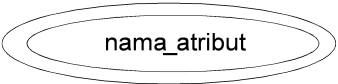
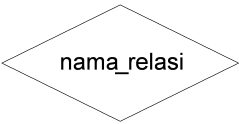
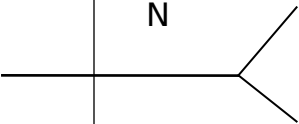
Adapun simbol-simbol yang digunakan pada ERD dengan notasi Chen adalah sebagai berikut:

**Tabel 2.3** Simbol-simbol *Entity Relational Diagram*

Simbol	Deskripsi
Entitas / <i>entity</i>	Entitas merupakan data inti yang





	<p>akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimoan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel</p>
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas</p>
<p>Atribut kunci primer</p> 	<p><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)</p>
<p>Atribut multivalai / <i>multivalue</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</p>
<p>Asosiasi / <i>association</i></p> 	<p>Penghubung antar relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu</p>



	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 maka
--	---

(Sumber: Rosa dan Shalahuddin , 2012:50-51)

### 2.2.4 Pengertian *Flowchart*

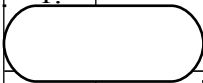
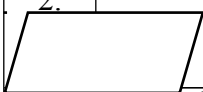
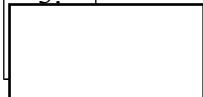
Menurut Saputra (2013:120) menyatakan bahwa, “*Flowchart* merupakan suatu diagram yang menggambarkan alur kerja suatu sistem.”

Menurut Indrajani (2015:36), “*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.”

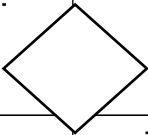
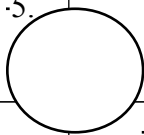
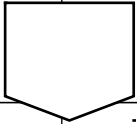
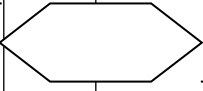

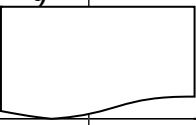
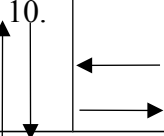
Menurut Ladjamudin (2013:263) mengemukakan bahwa, “*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.”

Siallagan (2009:6), menjelaskan simbol-simbol yang digunakan dalam *Flowchart*, yaitu:

**Tabel 3.4** Simbol-Simbol dalam *Flowchart*

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program
2.		<i>Input / Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
3.		<i>Proses</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh computer



4. 	<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya / tidak.
5. 	<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6. 	<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
7. 	<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8. 	<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9. 	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer)
10. 	<i>Flow</i>	Menyatakan jalannya arus suatu proses

(Sumber. Siallagan, 2009:6)

### 2.2.5 Pengertian Kamus Data



Menurut Ladjamudin (2013:70), “Kamus data sering disebut juga dengan *sistem data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.”

Rosa dan Shalahuddin (2012:73) menjelaskan, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*ouput*) dapat dipahami secara umum (memiliki standar cara penulisan).”

Mnurut Rosa dan Shalahuddin (2012:74), Kamus data memiliki beberapa simbol sebagai berikut :

**Tabel 2.5** Simbol-simbol Kamus Data

<b>Simbol</b>	<b>Keterangan</b>
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau...
{ }n	N kali diulang/bernilai banyak
()	Data opsional
*...*	Batas komentar

(Sumber: Rosa dan Shalahuddin, 2012:74)

## 2.3 Teori Judul

### 2.3.1 Pengertian Aplikasi

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

Asropudin (2013:6), “Aplikasi adalah software yang dibuat oleh suatu perusahaan untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word, Ms-Excel*”

Kesimpulannya, Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari user (pengguna).

### 2.3.2 Pengertian Penjadwalan

Menurut KBBI (Kamus Besar Bahasa Indonesia), penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal.



### 2.3.3 Pengertian Pengamanan

Menurut di Buku Asistensi Sistem Pengamanan Obvit / Nas (2017:4), Pengamanan adalah segala usaha, pekerjaan dan kegiatan dalam rangka pencegahan, penangkalan dan penyalahgunaan serta penegakan hukum terhadap setiap ancaman dan gangguan keamanan.

### 2.3.4 Pengertian Obyek Vital

Menurut di Buku Asistensi Sistem Pengamanan Obvit / Nas (2017:5), Obyek Vital adalah kawasan/lokasi, bangunan/instalasi atau usaha yang menyangkut hajat hidup orang banyak, kepentingan negara dan sumber pendapatan negara yang bersifat strategis.

### 2.3.5 Pengertian Aplikasi Penjadwalan Pengamanan Obyek Vital pada Direktorat Pam Obvit Polda Sumatera Selatan berbasis Web.

Aplikasi Penjadwalan Pengaman Obyek Vital pada Direktorat Pam Obvit Sumatera Selatan berbasis *Web* adalah suatu aplikasi yang berisi informasi patrol/pengamanan obyek vital dibawah naungan Polda Sumatera Selatan dan akan diberikan kepada Anggota Polisi untuk mengetahui jadwal dan lokasi pengamanan obyek vital yang datanya terkoneksi pada internet sehingga memudahkan Anggota mendapatkan informasi secara tepat dan akurat.

## 2.4 Teori Program

### 2.4.1 Pengertian Basis Data (*Database*)

Menurut Indrajani (2015:70) mengemukakan bahwa, “Sebuah Basis Data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi.”

### 2.4.2 Pengertian *XAMPP*



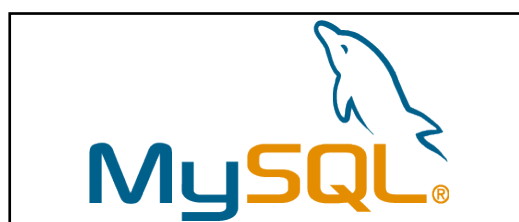


### Gambar 2.2 Logo XAMPP

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*.

### 2.4.3 Pengertian MySQL



Gambar 2.3 Logo MySQL

Sunafrihantono (2003:28), “*MySQL* adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). SQL adalah bahasa standar yang digunakan untuk mengakses *database server*.”

Agus Saputra (2013:14), “*MySQL* merupakan *database storage* yang paling banyak digunakan oleh *Web developer* karena sifatnya yang *free*, alias gratis.”



Kesimpulannya, *MySQL* adalah bahasa yang digunakan untuk mengelola data pada RDBMS bersifat *open source* sehingga banyak digunakan di dunia.

#### 2.4.4 Pengertian *PHP*



Sunafrihantono (2004:140), “*PHP* adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat halaman *Web* yang dinamis.”

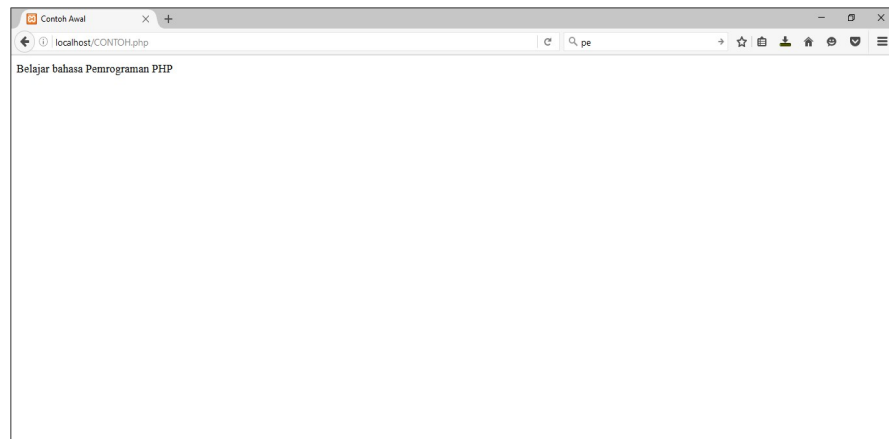
Saputra Agus (2013:4), “*PHP* merupakan bahasa *Server Side Scripting*, dimana *php* selalu membutuhkan *web server* dalam menjalankan aksinya.”

Kesimpulannya, *PHP* adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada *HTML*, yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi *desktop*.

#### 1. Sintaks Dasar *PHP*

Kode (Script) *PHP* yang sering disebut dengan istilah *embedded script* yaitu script *PHP* yang disisipkan di antara script *HTML*. Jadi dapat dikatakan script *PHP* hanya ditulis atau disisipkan ketika dibutuhkan saja, seperti menampilkan data dari database meng-upload file, delete data, edit data dan lain sebagainya. Contoh script :

```
<HTML>
  <HEAD>
    <TITLE>Contoh Awal</TITLE>
  </HEAD>
<BODY>
  <?php
    echo “Belajar bahasa Pemrograman PHP”;
```



Gambar 2.5 Contoh Script PHP

## 2. Tipe Data PHP

Tipe data *PHP* digunakan untuk menentukan jenis data yang akan disimpan dalam suatu variabel. Risnandar,dkk (2013:60-61), menjelaskan *PHP* mempunyai empat tipe data dasar sebagai berikut :

1. **Integer** merupakan tipe data yang mencakup semua bilangan bulat. Range bilangan integer adalah antara -2.147.4833.647 sampai dengan 2.147.483.647
2. **Floating** point merupakan tipe dataa yang mencakup semua bilangan desimal (bilangan yang memiliki angka dibelakang koma). Range bilangan floating point antara 1e308 sampai dengan 1e308.
3. **Character** merupakan tipe yang digunakan untuk menyimpan data-data yang berupa karakter (satu huruf). Penulisannya biasanya diapit dengan tanda kutip satu ('...').
4. **String** merupakan tipe data tersendiri dan tidak dapat dikelompokkan menjadi tipe data dasar. Penulisannya biasa diapit dengan tanda kutip dua ("...")."

### 2.4.5 Pengertian *PHPMYAdmin*



Gambar 2.6 Logo *PhpMyAdmin*





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

Madcoms (2016:12) *PhpMyAdmin* adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *PhpMyAdmin*, 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.