



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Sistem

Pratama (2014:7) sistem merupakan sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama.

Hutahaean (2015:2) sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu.

Dari uraian tersebut didapat bahwa definisi sistem adalah serangkaian dua atau lebih yang saling berhubungan satu dengan lainnya dalam melakukan tugas secara bersama-sama untuk mencapai tujuan tertentu.

2.1.2 Pengertian Informasi

Rochaety, E., *et al.* (2013:6) mengemukakan bahwa, informasi merupakan kumpulan data yang telah diolah baik bersifat kualitatif maupun kuantitatif dan memiliki arti lebih luas.

Djahir dan Pratita (2014:10) informasi adalah merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian – kejadian nyata yang dapat digunakan sebagai alat bantu untuk mengambil suatu keputusan.

Dari uraian-uraian tersebut didapat bahwa definisi dari informasi adalah sekumpulan data yang di olah menjadi sebuah informasi yang akan berguna atau berfungsi bagi penerima informasi tersebut.

2.1.3 Pengertian Sistem Informasi

Pratama (2014:10) sistem informasi merupakan gabungan dari empat bagian utama keempat bagian utaa tersebut mencakup perangkat lunak (*software*),



perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia (SDM) yang terlatih.

Hutahaean (2014:13) sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Dari uraian diatas dapat disimpulkan bahwa definisi dari sistem informasi adalah sekumpulan komponen atau elemen yang saling berhubungan dalam mencapai tujuan untuk mengelolah dan menyampaikan informasi yang akan berguna bagi penerimanya.

2.1.4 Pengertian Sistem Informasi Geografis

Gillavry (dalam Riyanto, 2010 : 26) SIG adalah sistem informasi yang didistribusikan diseluruh lingkungan jaringan komputer untuk mengintegrasikan, menyebarkan dan mengkomunikasikan informasi geografis secara visual di World Wide Web melalui internet.

2.1.5 Pengertian Pemetaan

Sendow dan Longdong (2012:37) pada umumnya peta adalah sarana guna memperoleh gambaran data ilmiah yang terdapat di atas permukaan bumi dengan cara menggambarkan berbagai tandatanda dan keterangketerangan, sehingga mudah dibaca dan dimengerti.

Peta yang memberikan gambaran mengenai kondisi permukaan suatu areal tertentu pada permukaan bumi yang dinyatakan dengan simbol-simbol, tandatanda, serta keterangan dalam skala tertentu disebut peta topografi.

2.1.6 Pengertian Basis Data (*Database*)

Raharjo (2011:3) *database* adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.



Oktavian (2010:62) *database* adalah sekumpulan data dan prosedur yang memiliki struktur sedemikian rupa sehingga mudah dalam menyimpan, mengatur, dan menampilkan data. Banyak program *database* yang tersedia, diantaranya adalah: *Oracle, MySQL, MSSQL, PostgreSQL, Paradox, Foxpro*, dan lain-lain. *Database* terbentuk dari beberapa komponen. Berikut adalah komponen pembentuk *database* yaitu *table, Record, Field*.

Dari uraian diatas dapat disimpulkan bahwa definisi dari Basis data merupakan sekumpulan data yang saling berelasi atau berhubungan yang memungkinkan beberapa pemakai untuk mengakses dan memanipulasinya.

2.1.7 Pengertian Kebakaran

Kebakaran adalah suatu reaksi oksidasi eksotermis yang berlangsung dengan cepat dari suatu bahan bakar yang disertai dengan timbulnya api/penyalaan.

(<https://pemadamapi.wordpress.com/definisi-pengertian-kebakaran/>)

2.1.8 Pengertian Sistem Informasi Geografis Pemetaan Persebaran Pos Pemadam Kebakaran Kota Palembang Berbasis *Android*

Sistem informasi geografis ini adalah sebuah sistem dalam mencari pos persebaran tempat pemadam kebakaran yang dibangun dengan menggunakan *web* sebagai sistem yang dipakai dalam penggunaanya.

2.2 Teori Khusus

2.2.1 *Data Flow Diagram (DFD)*

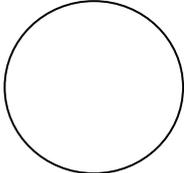
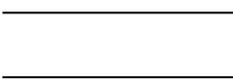
Sukamto dan Shalahudin (2013:70) *data flow diagram (DFD)* atau dalam Bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi



beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi oleh karena itu, DFD lebih sesuai untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur, karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

Tabel 2.1 Simbol DFD (*Data Flow Diagram*)

No.	Simbol	Keterangan
1.		<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 :</p> <p>Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p>File atau basis data atau penyimpanan (storage); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi table-table pada basis data yang dibutuhkan, table-table ini juga harus sesuai dengan perancangan table-table pada basis data (Entity Relationship Diagram (ERD), Conceptual Data Model (CDM), Physical Data Model (PDM)).</p> <p>Catatan:</p>



		Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.
3.		Entitas luar (external entity) atau masukan (input) atau keluaran (output) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan Catatan: Nama yang digunakan pada masukan (input) atau keluaran (output) biasanya kata benda
4.		Aluran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (input) atau keluaran (output). Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata benda, dapat diawali dengan kata data misalnya "data siswa" atau tanpa kata data misalnya "data siswa" atau tanpa kata data misalnya "siswa"

(Sumber : Sukanto dan Shalahuddin 2013:71-72)

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD :



1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan datang 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 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

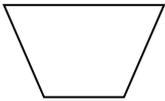
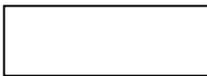
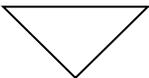
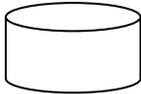
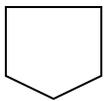
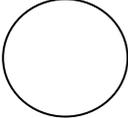
DFD Level 3, 4, 5, dan seterusnya merupakan *breakdown* dan modul pada DFD Level di-atasnya. *Breakdown* pada Level 3,4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

2.2.2 *Block Chart*

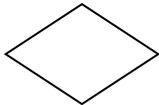
Kristanto (2011:68) mengemukakan bahwa, *block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol *Block Chart* adalah sebagai berikut:

Tabel 2.2 Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran



11.		Pengambilan keputusan (decision)
12.		Layar peraga (monitor)
13.		Pemasukan data secara manual

(Sumber: Kristanto, 2011:68)

2.2.3 Kamus Data (*Data Dictionary*)

Sukanto dan Shalahuddin (2013:73) kamus data (*data dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem sehingga masukan dan keluaran dapat dipahami secara umum (memiliki standar cara penulisan).

Adapun simbol-simbol dalam kamus data adalah sebagai berikut :

Tabel 2.3 Simbol-simbol Kamus Data (*Data Dictionary*)

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

(Sumber: Sukanto dan Shalahuddin 2013:74)

2.2.4 Daftar Kejadian (*Event List*)

Kristanto (2011:64) menjelaskan tentang daftar kejadian (*event list*) sebagai berikut:



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

Adapun cara-cara mendeskripsikan daftar kejadian adalah sebagai berikut:

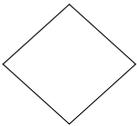
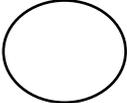
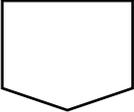
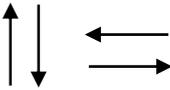
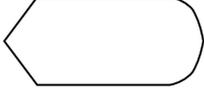
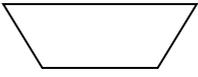
1. Pelaku adalah entiti luar, jadi bukan sistem.
2. Menguji setiap entiti luar dan mencoba mengevaluasi setiap entiti luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya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

2.2.5 Pengertian *Flowchart*

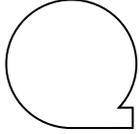
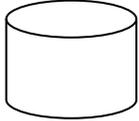
Ladjamudin (2013:211) menyatakan bahwa, *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

Adapun simbol-simbol *Flowchart* adalah sebagai berikut:

Tabel 2.4 Simbol-Simbol *FlowChart*

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



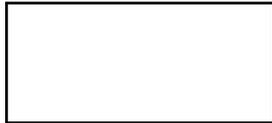
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (subprogram)
12.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic
13.		Simbol database atau basis data

(Sumber: Ladjamudin, 2013:211)

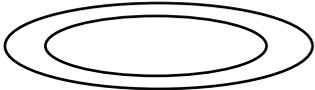
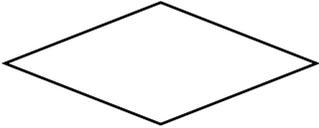
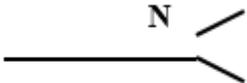
2.2.6 Entity Relational Diagram (ERD)

Hidayatullah dan Kawistara (2014:148) entitas adalah suatu objek (baik nyata maupun abstrak) didunia nyata yang dapat dibedakan dari objek lain berdasarkan karakteristik yang dimilikinya. Relasi adalah asosiasi yang menyatakan keterhubungan antar entitas. Jadi, *Entity Relationship Diagram* (ERD) adalah diagram yang merepresentasikan hubungan antar entitas dalam pemodelan yang digunakan dalam merancang basis data.

Tabel 2.5 Simbol ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas / entity 	Entitas merupakan data inti yang akan disimpan, bakal table pada basis data, benda yang memiliki data dan harus disimpan datanya agar diakses oleh aplikasi <i>computer</i> , penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table.



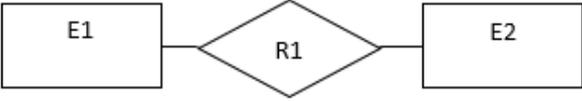
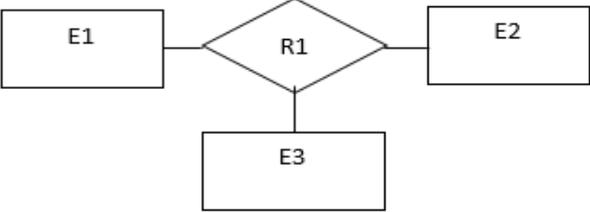
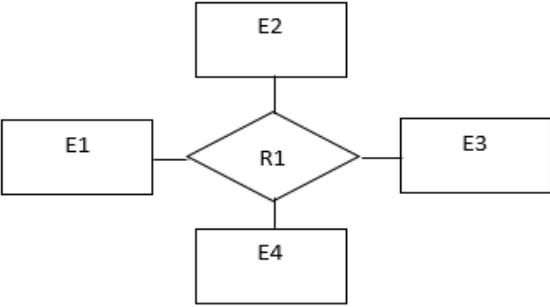
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.</p>
<p>Atribut kunci primer</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat <i>unik</i> (berbeda tanpa ada yang sama)</p>
<p>Atribut multivalai / multivalue</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 / association</p> 	<p>Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antar 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 dengan entitas B.</p>

(Sumber: Sukamto dan Shalahudin, 2013:50)



ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*. Berikut adalah contoh bentuk hubungan relasi dalam ERD:

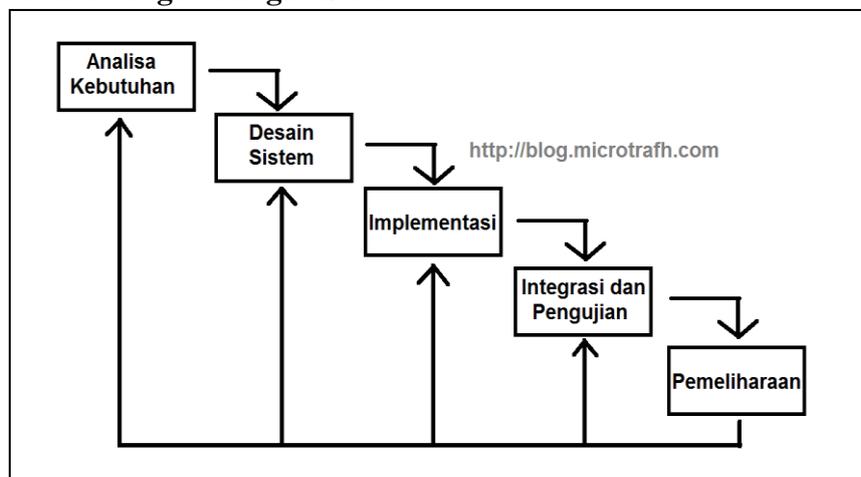
Tabel 2.6 Simbol Relasi *Entity Relationship Diagram* (ERD)

Nama	Gambar
<i>Binary</i>	
<i>Ternary</i>	
<i>N-ary</i>	

(Sumber: Sukamto dan Shalahudin, 2013:52)



2.2.7 Metode Pengembangan Sistem



Gambar 2.1 Model Waterfall

Sukanto dan Shalahuddin (2013:28) 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*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

a. Analisis

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.

b. Desain

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

c. Pengodean

Pada tahap pengodean, desain harus ditranlasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain



d. Pengujian

Tahap pengujian fokus pada sistem 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.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah sistem mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem baru.

2.3 Teori Program

2.3.1 Sekilas Tentang *PHP*

2.3.1.1 Pengertian *PHP*

Saputra (2012:2) menyatakan bahwa *PHP* memiliki kepanjangan *Hypertext Preprocessor*, merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. Sehingga *PHP* disebut juga sebagai bahasa *Server Side Scripting*.

Naista (2016:4) berpendapat *Hypertext Preprocessor* atau lebih singkatnya *PHP* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* dapat digunakan untuk membangun *CMS (Content Management System)*.

Sarwandi (2017:41) *php* merupakan bahasa pemrograman web. Bahasa pemrograman *PHP* merupakan bahasa pemrograman untuk membuat web yang bersifat *server side scripting*.

Berdasarkan pendapat di atas dapat disimpulkan bahwa, *PHP* merupakan salah satu pemrograman web *server-side* yang digunakan untuk membuat suatu aplikasi dengan menggunakan editor teks atau editor *HTML*.



2.3.1.2 Script PHP

Menurut Murya (2017:18), “*Script PHP* berkedudukan sebagai tag dalam bahasa *HTML*. Sebagaimana diketahui, *HTML (Hypertext Markup Language)* adalah bahasa standar untuk membuat halaman-halaman *web*.”

Sebagai contoh, berikut adalah kode *HTML* (disimpan dengan ekstensi *.htm* atau *.html*) :

```
<?php
// membuat koneksi
$con=mysqli_connect("localhost","root","","latihan_php");
// Melakukan pengecekan koneksi
If (mysqli_connect_errno()) {
echo "Koneksi Gagal!:".mysqli_connect_error();
}
?>
```

2.3.2 Sekilas Tentang MySQL

2.3.2.1 Pengertian MySQL

Raharjo (2011:21) *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*).

Hidayatullah dan Kawistara (2014:180) *MySQL* adalah suatu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrograman aplikasi *web*. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sring *dibundling* dengan *web server* sehingga proses instalasinya jadi lebih mudah.

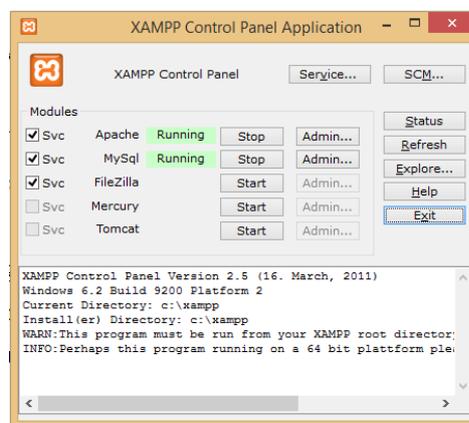
Saputra (2012:7) menyatakan bahwa *MySQL* bekerja menggunakan bahasa SQL Language (Structure Query Language). Itu dapat diartikan bahwa *MySQL* merupakan standar penggunaan database di dunia untuk pengolahan data.



Dari pendapat di atas dapat disimpulkan bahwa MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database.

2.3.3 Pengertian XAMPP

Nugroho (2013:1) menjelaskan XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal.



Gambar 2.2 Tampilan XAMPP

2.3.4 Pengertian WebServer

Hidayatullah dan Kawistara (2014:125) *web server* adalah tempat dimana anda dapat menyimpan aplikasi web kemudian mengaksesnya melalui internet. Setiap perubahan, kecil maupun besar, Anda *upload* ke *web server* baru setelah itu anda periksa apakah perubahan itu sudah sesuai dengan yang Anda inginkan atau belum.

Setiawan (2015:2) mengemukakan bahwa, Web Server adalah yang berfungsi untuk menerima informasi yang diminta oleh browser, mengolah informasi dan mengirimkan output dengan format HTML.

Dari beberapa pendapat di atas maka dapat disimpulkan bahwa web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada Klien yang dikenal dan biasanya kita kenal dengan nama web browser (Mozilla Firefox dan Google



Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML.

2.3.5 Pengertian Android

Safaat (2015:1) android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi.

2.3.6 Pengertian Javascript

Sidik (2011:2) javascript adalah salah satu pengetahuan yang penting dikuasai oleh pengembang program berbasis web karena dengan *javascript* ini dapat membuat dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif dengan memberikan beberapa fungsionalitas ke dalam halaman web dengan menggunakan *interface web*.

Winarno., et al. (2014:129) bahasa scripting client side yang sangat populer. Hampir semua programmer web menggunakan javascript untuk memberi efek pemrograman di halaman.

Dari beberapa pendapat diatas maka dapat disimpulkan bahwa javascript adalah JavaScript adalah bahasa pemrograman yang bisa disisipkan ke HTML seperti halnya PHP akan tetapi javascript berjalan di sisi Client.

2.3.7 Pengertian phpMyAdmin

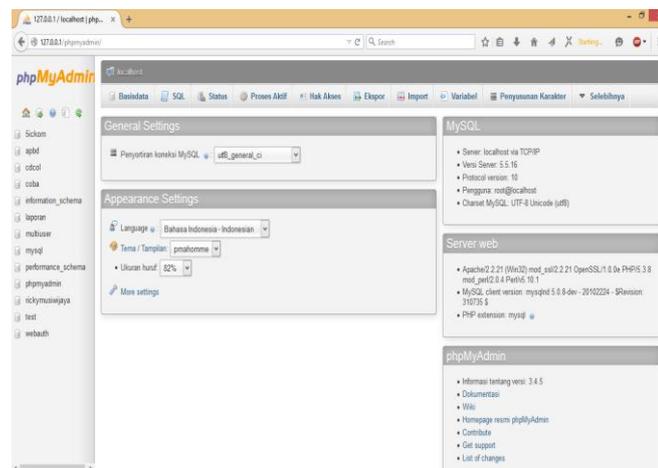
Nugroho (2013:71) menjelaskan phpMyAdmin adalah tools *MySQL Client* berlisensi *Freeware*, Anda dapat mengunduhnya secara gratis pada situs resminya di www.phpmyadmin.net.

Riyanto (2011:17) menyatakan bahwa *PHPMYAdmin* merupakan aplikasi web berbasis PHP yang telah banyak digunakan untuk administrasi database MySQL.

Sarwandi (2017:41) *phpMyAdmin* merupakan bagian untuk mengelola basisdata MySQL yang ada di komputer. Untuk membukanya, buka browser kemudian ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman PHPMyAdmin.



Dari beberapa pendapat diatas maka dapat disimpulkan bahwa *phpMyAdmin* adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (*World Wide Web*). *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perizinan (*permissions*), dan lain-lain)



Gambar 2.3 Tampilan phpMyAdmin