



## BAB II TINJAUAN PUSTAKA

### 2.1. Teori Umum

#### 2.1.1. Pengertian Aplikasi

Sutabri (2012:147), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.”

Hendrayudi (2009:143), “Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).”

Jogiyanto (1999:12), “Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.” (Diakses di situs <http://pdianaekadara.blogspot.com/2013/10/pengertian-aplikasi.html> pada tanggal 09 Juni 2015)

Dari pengertian di atas, dapat disimpulkan bahwa aplikasi adalah alat yang dibuat untuk melaksanakan tugas tertentu sesuai dengan fungsi, kemampuan dan yang dibutuhkan oleh pengguna (*user*).

#### 2.1.2. Pengertian Pengolahan

Jogiyanto (2006:9), “Pengolahan (*processing*) adalah proses data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.”

“Pengolahan adalah sebuah proses mengusahakan atau mengerjakan sesuatu (barang dsb) supaya menjadi lebih sempurna.” (Tim Penyusun Kamus Pusat Pembinaan dan Pengembangan Bahasa: 1988). (Diakses di situs <http://kumpulantugaskuliahpertanian.blogspot.com/2012/03/pengertian-pengolahan.html> pada tanggal 09 Juni 2015)



Dari pengertian di atas, dapat disimpulkan bahwa pengolahan adalah proses mengolah data menjadi suatu informasi yang menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.

### 2.1.3. Pengertian Data

Sutabri (2012:22), “Data adalah kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata, dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.”

Wahyudi (2012:3), “Data adalah suatu *file* ataupun *field* yang berupa karakter atau tulisan dan gambar.”

Utami dan Sukrisno (2005:8), “Data merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.”

Ladjamudin (2013:9), “Data adalah deskripsi dari sesuatu dan kejadian yang kita hadapi (*the description of things and events that we face*).”

Mulyanto (2009:16), “Data merupakan material atau bahan baku yang belum mempunyai makna atau belum berpengaruh langsung kepada pengguna sehingga perlu diolah untuk dihasilkan sesuatu yang lebih bermakna.”

Dari pengertian di atas, dapat disimpulkan bahwa data adalah suatu *file* yang dapat berupa tulisan ataupun gambar yang belum berpengaruh karena belum diolah menjadi sesuatu yang lebih berguna sehingga belum dapat disajikan.

### 2.1.4. Pengertian Pengolahan Data

Ladjamudin (2013:9), “Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Rajab (2009:66), “Pengolahan data adalah suatu kegiatan mengubah data awal menjadi data yang memberikan informasi ke tingkat yang lebih tinggi.”



Kristanto (2008:8), “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Supranto (2000:27), “Pengolahan data adalah proses untuk memperoleh data/angka atau ringkasan berdasarkan suatu kelompok data mentah.”

Dari pengertian di atas, dapat disimpulkan bahwa pengolahan data adalah waktu yang digunakan untuk memperoleh data/angka untuk dapat memberikan informasi ke tingkat yang lebih tinggi atau berguna.

#### **2.1.5. Pengertian Pelatihan (*Training*)**

Mardiyatmo (2008:49), “Pelatihan atau *training* adalah suatu kegiatan dari perusahaan yang bermaksud untuk dapat memperbaiki dan memperkembangkan sifat tingkah laku, keterampilan dan pengetahuan dari para karyawannya, sesuai dengan keinginan dari perusahaan yang bersangkutan.”

Tim Pengembang Ilmu Pendidikan FIP – UPI (2007:463), “Pelatihan adalah kegiatan profesional yang memerlukan pengalaman khusus dan pengakuan (sertifikasi).”

Tim Pengembang Ilmu Pendidikan FIP – UPI (2007:463), “Pelatihan adalah suatu disiplin akademik, yaitu kegiatan terorganisasi untuk mempelajari proses, produk, dan profesi pelatihan dengan menggunakan kajian sejarah, filsafat, dan ilmu pengetahuan tentang manusia, atau kajian keilmuan tentang manusia yang bermasyarakat (*the sciences of social man*).”

Dari pengertian di atas, dapat disimpulkan bahwa pelatihan adalah suatu upaya untuk menumbuhkembangkan perubahan yang digunakan untuk meningkatkan keterampilan, pengetahuan, dan nilai untuk mendapatkan sertifikasi.



### **2.1.6. Pengertian Karyawan**

Karyawan adalah semua pihak yang memperoleh penghasilan secara teratur dan bekerja untuk sebuah perusahaan dan penghasilannya telah dipotong dan dihitung PPh Pasal 21 oleh perusahaan melalui laporan bulanan perusahaan.

Karyawan yaitu mereka yang bekerja pada suatu badan usaha atau perusahaan baik swasta maupun pemerintahan dan diberikan imbalan kerja sesuai dengan peraturan perundang-undangan yang berlaku baik yang bersifat harian, mingguan, maupun bulanan yang biasanya imbalan tersebut diberikan secara mingguan. (Diakses di situs <http://hujau.blogspot.com/2010/06/pengertian-buruh-karyawan-dan-pegawai.HTML> pada tanggal 1 Juni 2015).

Dari pengertian di atas, dapat disimpulkan bahwa karyawan adalah mereka yang bekerja pada suatu badan usaha yang memperoleh penghasilan atau diberi imbalan kerja sesuai dengan peraturan perundang-undangan yang berlaku secara teratur.

### **2.1.7. Pengertian Aplikasi Pengolahan Data Pelatihan Karyawan pada Pertamina *Health, Safety, and Environment Training Center* Sungai Gerong Palembang**

Aplikasi pengolahan data pelatihan karyawan pada Pertamina *Health, Safety, and Environment Training Center* (HSE TC) Sungai Gerong Palembang adalah suatu aplikasi yang dibuat untuk mempermudah dalam proses pengolahan data pelatihan karyawan dan informasi laporan kepada pimpinan dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

### **2.1.8. Metode Pengembangan Sistem**

Sukanto dan Shalahudin (2014:28), menjelaskan tentang metode pengembangan sistem model *waterfall*, sebagai berikut:



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 dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

#### 1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

#### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentransaksi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

#### 3. Pembuatan Kode Program

Desain harus ditranslasikan ke program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

#### 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi dan fungsional dan memastikan bahwa bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

#### 5. Pendukung (*Support*)

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 perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## 2.2. Teori Khusus

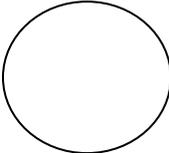
### 2.2.1. Pengertian *Data Flow Diagram (DFD)*

Kristanto (2008:61), “*Data Flow Diagram* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.”

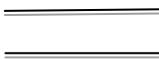
Sukanto dan Shalahuddin (2014:288), “*Data Flow Diagram* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan (*input*) dan keluaran (*output*). *DFD* tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek.”

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

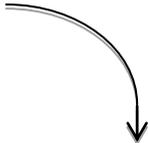
**Tabel 2.1** Simbol-simbol *Data Flow Diagram*

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

Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*

NOTASI	KETERANGAN
	<p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>
	<p><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 basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CMD), <i>Physical Data Model</i> (PDM))</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai atau 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>

Lanjutan Tabel 2.1 Simbol-simbol *Data Flow Diagram*

NOTASI	KETERANGAN
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>)</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

Sumber: Sukamto dan Shalahuddin (2014:71)

Menurut Sukamto dan Shalahuddin (2014:72), 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 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 kedetilan 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.

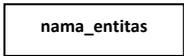
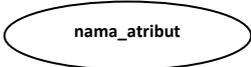
*DFD Level 3, 4, 5* dan seterusnya merupakan *breakdown* dari modul pada *DFD Level* di atasnya. *Breakdown* pada *level 3, 4* dan *5* dan seterusnya aturannya sama persis dengan *DFD Level 1* atau *Level 2*.

### 2.2.2. Pengertian *Entity Relationship Diagram (ERD)*

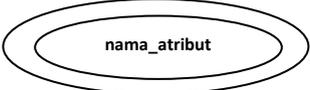
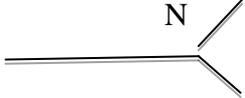
Sukanto dan Shalahuddin (2014:289), “*Entity Relationship Diagram (ERD)* adalah pemodelan awal basis data yang akan dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional”.

Sukanto dan Shalahuddin (2014:50), 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. Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

**Tabel 2.2** Simbol-simbol *Entity Relationship Diagram (ERD)*

SIMBOL	DESKRIPSI
Entitas / Entity 	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 komputer; 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

Lanjutan Tabel 2.2 Simbol-simbol *Entity Relationship Diagram* (ERD)

SIMBOL	DESKRIPSI
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 multivalu/multivalue 	<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/association 	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. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B

Sumber: Sukamto dan Shalahuddin (2014:50)



### 2.2.3. Pengertian *Flow Chart*

EWolf (2012:16), “*Flowchart* adalah simbol-simbol yang pekerjaan yang menunjukkan bagian aliran proses yang saling terhubung.”

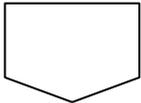
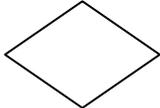
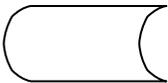
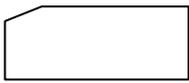
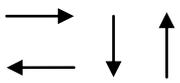
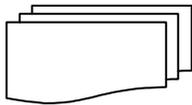
Siallagan (2009:6), “*Flow chart* adalah suatu diagram alir yang mempergunakan simbol atau tanda untuk menyelesaikan masalah.”

Siallagan (2009:6) juga menjelaskan simbol-simbol pada *flow chart* sebagai berikut:

**Tabel 2.3** Simbol-simbol *Flow Chart*

NO.	SIMBOL	FUNGSI
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
3.		Proses yang terdefinisi atau sub program.
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Menyatakan masukan dan keluaran ( <i>input/output</i> ).
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.

Lanjutan Tabel 2.3 Simbol-simbol *Flow Chart*

NO.	SIMBOL	KETERANGAN
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>decision</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetic.
11.		Menyatakan input/output menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan input/output dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		Multidocument (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

Sumber: Siallagan (2009:6)

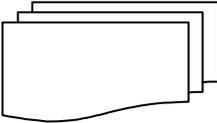
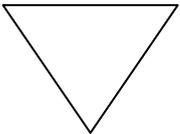
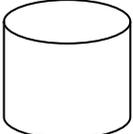


### 2.2.4 Pengertian *Block Chart*

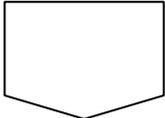
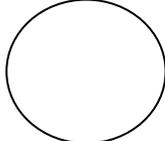
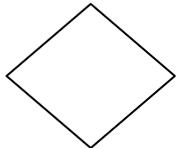
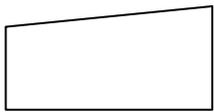
Kristanto (2008:75), “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu.”

Kristanto (2008:75) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

**Tabel 2.4** Simbol-simbol *Block Chart*

NO.	SIMBOL	ARTI
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh computer.
5.		Menandakan dokumen yang diarsipkan (arsip manual).
6.		Data penyimpanan (data storage).
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.

Lanjutan Tabel 2.4 Simbol-simbol *Block Chart*

NO.	SIMBOL	ARTI
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 (2008:75)

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

Kristanto (2008:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.”

Jogiyanto (2008:46), “Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.”

Menurut Jogiyanto (2008:46), dari definisi ini, terdapat tiga hal yang berhubungan dengan basis data, yaitu sebagai berikut ini.



1. Data itu sendiri yang diorganisasikan dalam bentuk basis data (*database*).
2. Simpanan permanen (*storage*) untuk menyimpan basis data tersebut. Simpanan ini merupakan bagian dari teknologi perangkat keras yang digunakan di sistem informasi. Simpanan permanen yang umumnya digunakan berupa *hard disk*.
3. Perangkat lunak untuk memanipulasi basis datanya. Perangkat lunak ini dapat dibuat sendiri dengan menggunakan bahasa pemrograman komputer atau dibeli dalam bentuk suatu paket. Banyak paket perangkat lunak yang disediakan untuk memanipulasi basis data. Paket perangkat lunak ini disebut dengan DBMS (*Data Base Management System*). Contoh DBMS yang terkenal misalnya *dBASE*, *Fox Base*, *Microsoft Access*, *Oracle* dan lain-lain.

#### 2.2.6. Pengertian Kamus Data

Sukamto dan Shalahuddin (2014:73), “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).”

Sukamto dan Shalahuddin (2014:73), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu :

**Tabel 2.5** Simbol-simbol Kamus Data

NO.	SIMBOL	ARTI
1.	=	disusun atau terdiri atas
2.	+	Dan
3.	[ ]	baik ...atau...
4.	{ } <sup>n</sup>	n kali diulang/ bernilai banyak
5.	( )	data operasional
6.	*...*	batas komentar

Sumber: Sukamto dan Shalahuddin (2014:73),



Kamus data pada *DFD* nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan *DFD* masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan *DFD*-nya, atau keduanya.

### 2.3. Teori Program

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

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

Saputra (2013:4), “PHP merupakan bahasa *server side scripting*, di mana php selalu membutuhkan server dalam menjalankan aksinya.”

Kadir (2013:120), “PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi *web*. Ditinjau dari pemrosesannya, PHP tergolong berbasis *server side*. Artinya, pemrosesan dilakukan di *server*. Hal ini berkebalikan dengan bahasa seperti *Java Script* yang pemrosesannya dilakukan di sisi klien (*client side*).”

Sidik (2014:4) menjelaskan, PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemograman *server side*.

Dari beberapa pengertian diatas, dapat disimpulkan bahwa PHP adalah bahasa pemograman yang digunakan untuk aplikasi *web* selalu membutuhkan *server* dalam menjalankan aksinya.

---



### 2.3.2. Pengertian MySQL

Wahana Komputer (2010:2), “MySQL sendiri adalah sebuah database relasional. Database yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Pada setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefinisikan jenis informasi apa yang harus disimpan.”

Kadir (2013:15), “MySQL adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan MySQL, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.”

Pratama (2010:10), “MySQL adalah suatu sistem manajemen basis data rasional (*RDBMS-Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh, dan mudah digunakan.”

Sidik (2014:333), “MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi *query* dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah. MySQL telah tersedia juga di lingkungan Windows.”

Dari beberapa pengertian diatas, dapat disimpulkan bahwa MySQL adalah software database relasional tujuan memudahkan penyimpanan dan pengaksesan data dan dapat diakses dengan cara yang mudah dan cepat.

### 2.3.3. Pengertian XAMPP

Sidik (2014:72), “XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *server web* PHP dan *database* MySQL yang paling populer dikalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai databasenya.”



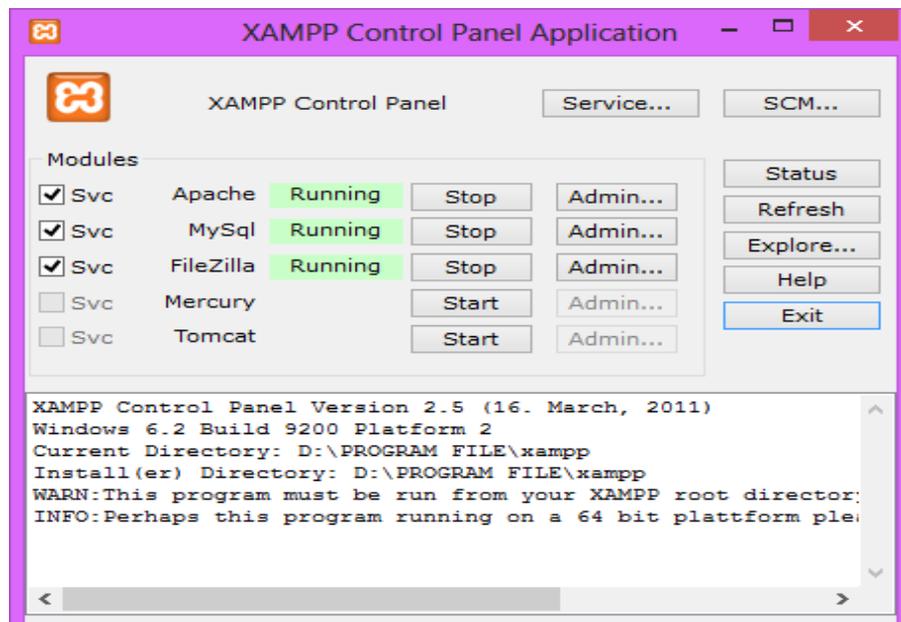
Nugroho (2013:1), “XAMPP adalah paket program *web* lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya PHP dan MySQL, paket ini dapat di *download* secara gratis dan legal.”

Menurut Nugroho (2013:7), dibawah folder utama XAMPP, terdapat beberapa folder penting yang perlu diketahui. Penjelasan fungsinya sebagai berikut:

**Tabel 2.6** Folder-folder penting dalam XAMPP

NO.	NAMA FOLDER	FUNGSI
1.	Apache	Folder utama dari <i>Apache Webserver</i> .
2.	Htdocs	Folder utama untuk menyimpan data-data latihan <i>web</i> , baik PHP maupun HTML biasa. Pada folder ini, anda dapat membuat subfolder sendiri untuk mengelompokkan file latihannya. Semua folder dan file program di htdocs bisa diakses dengan mengetikkan alamat <a href="http://localhost/">http://localhost/</a> di <i>browser</i> .
3.	Manual	Berisi subfolder yang di dalam terdapat manual program dan <i>database</i> , termasuk manual PHP dan MYSQL.
4.	MySQL	Folder utama untuk <i>database</i> MYSQL <i>server</i> . di dalamnya terdapat subfolder data (lengkapnya: C:\MySQL\MySQL\data) untuk merekam semua nama <i>database</i> , serta subfolder bin yang berisi <i>tools klien</i> dan <i>server</i> MYSQL.
5.	PHP	Folder utama untuk program PHP.

Sumber: Nugroho (2013:7)



Gambar 2.1 XAMPP Control Panel Application