



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

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

Menurut Kadir (2017:2) “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Menurut Siallagan (2009:1) “Komputer adalah sebagai sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (input), mengolah data (*process*), memberikan informasi (*output*), dan terkoordinasi di bawah control program yang tersimpan dalam memorinya”.

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

2.1.2 Pengertian Data

Menurut Asropudin (2013:22) “Data adalah kumpulan dari angka – angka maupun karakter – karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

Menurut Setyaningrum (2015:1) “Data adalah catatan atas sekumpulan fakta yang belum mempunyai arti bagi penerimanya dan maish memerlukan suatu pengolahan”.

Menurut Sutabri (2012:22) “Data merupakan kenyataan yang menggambarkan suau kejadian serta merupakan suatu bentuk yang masih belum dapat bercerita banyak sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi”.

Dari beberapa pendapat diatas Data adalah kenyataan terhadap fakta tentang suatu kejadian yang tidak berarti bagi pemakai.



2.1.3 Pengertian Pengolahan Data

Menurut Yakub (2012:75) “Pengolahan data adalah manipulasi atau transformasi simbol-simbol seperti; angka dan abjad untuk tujuan meningkatkan kegunaannya”.

Menurut Manullang (2013:313) “Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data”.

Dari beberapa pendapat diatas dapat disimpulkan pengolahan data adalah manipulasi data ke bentuk yang lebih informatif berupa informasi.

2.1.4 Pengertian Perangkat Keras (*Hardware*)

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

Menurut Puspitosari (2013:7) “*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 diatas dapat disimpulkan perangkat Keras (*Hardware*) adalah elemen atau perangkat fisik komputer yang beroperasi di dalamnya.

2.1.5 Pengertian Perangkat Lunak (*Software*)

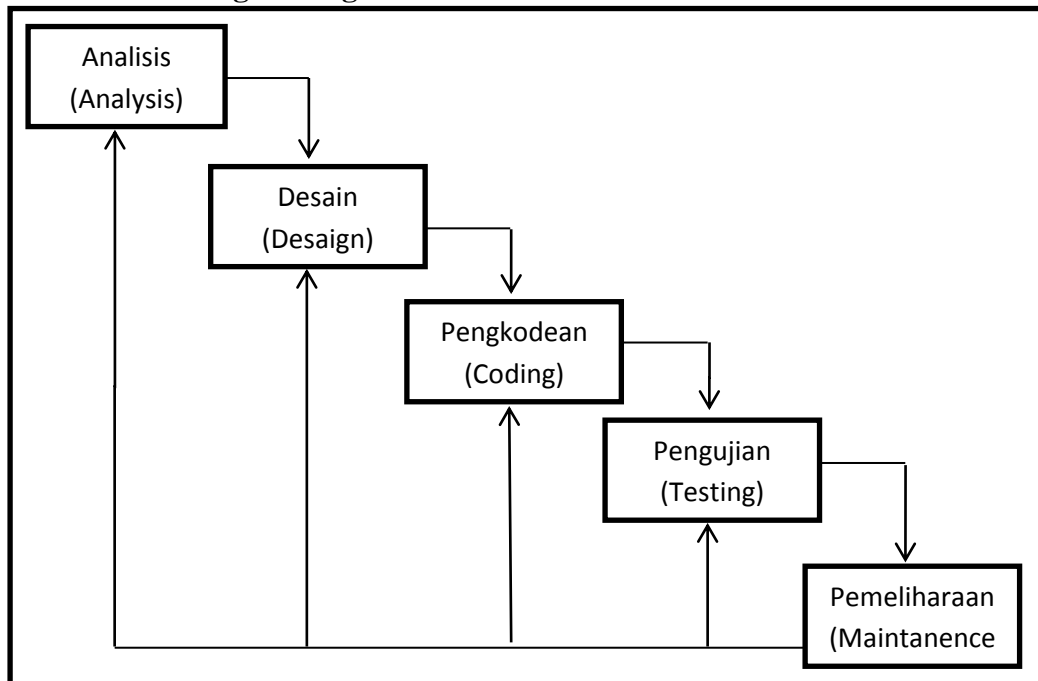
Menurut Kadir (2017:2) “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Blm di print difoto

Menurut Sukamto dan Shalahuddin (2013:2) “Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual)”.

Dari beberapa pendapat diatas dapat disimpulkan perangkat lunak adalah program untuk memasuk data, mengecek data, menyimpan data, memperoleh data hasil, dan presentasi serta manipulasi data.



2.1.6 Metode Pengembangan Sistem



Gambar 2.1 Ilustrasi Model *Waterfall*

Menurut Sukanto dan 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 perangkat 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 desain 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 desain 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 Data Flow Diagram (DFD)

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

Menurut Sukamto dan Shalahuddin (2013:70) *DFD* dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail.

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 2

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 Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut.

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.

Adapun simbol-simbol *Data Flow Diagram* adalah sebagai berikut:

Tabel 2.1. Simbol-Simbol *Data Flow Diagram*

No.	Nama	Simbol	Keterangan
1.	Proses		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya di dalam kode program
2.	Berkas atau Tempat Penyimpanan		File atau basisdata atau penyimpanan; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang

Lanjutan **Tabel 2.1.** Simbol-Simbol *Data Flow Diagram (DFD)*

			harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan.
3.	Entitas Luar		Entitas luar (external entity) orang yang berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan
4.	Aliran data		Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan.



(Sumber: Sukamto dan Shalahuddin, 2013:71)

2.2.2 Block Chart

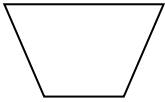

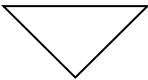
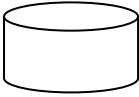

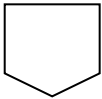
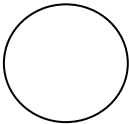
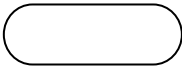
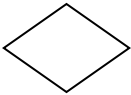

Menurut Kristanto (2008:68) “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

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

Lanjutan Tabel 2.2. Simbol-Simbol *Blockchart*

3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (data storage)
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)

Lanjutan Tabel 2.2. Simbol-Simbol *Blockchart*

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

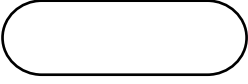

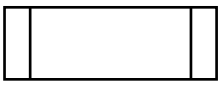

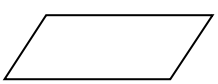
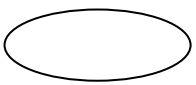
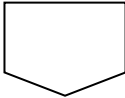

(Sumber: Kristanto, 2008:68)

2.2.3 Flowchart

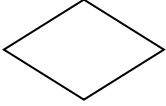
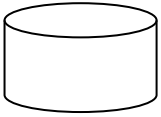
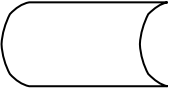
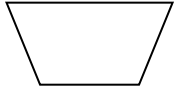
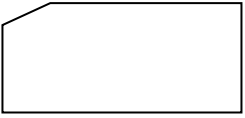
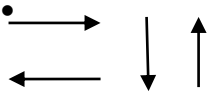

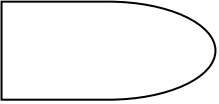
Menurut Ladjamudin (2013:211) “*Flowchart* adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah – langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma”.

Menurut Siallagan (2009:6) menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:

Tabel 2.3. Simbol-Simbol dalam *Flowchart*

No.	Simbol	Keterangan
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 yang menyatakan bagian dari program (sub program).
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses.
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.

Lanjutan **Tabel 2.3.** Simbol-Simbol *Flowchart*

9.		Menyatakan desicion (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
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.		Delay (penundaan atau kelambatan).

(Sumber : Siallagan, 2009:6)




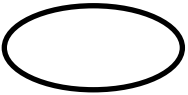
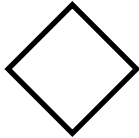

2.2.4 Entity Relationship Diagram (ERD)

Menurut Sukanto dan Shalahuddin (2013:50) “*Entity relationship diagram* digunakan untuk permodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *Object Oriented Database Management System (OODBMS)* maka perancangan basis data tidak perlu menggunakan *ERD*”.

Menurut Yakub (2012:60) “*Entity relationship diagram (ERD)* untuk mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas (*entity*) dan hubungannya *ERD* merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak”.

Adapun simbol-simbol *Entity Relationship Diagram* adalah sebagai berikut:

Tabel 2.4. Simbol-Simbol *Entity Relationship Diagram (ERD)*

No.	Nama	Simbol	Keterangan
1.	Entitas		Berupa orang, kejadian, atau benda di mana data akan dikumpulkan
2.	Atribute		Merupakan properti dari entitas. Nama atribut harus merupakan kata benda
3.	Relationship		Menunjukkan hubungan antar 2 entitas. Dideskripsikan dengan kata kerja
4.	Link		Sebagai penghubung antara entitas dan relationship serta entitas dan atribut

(Sumber: Yakub, 2012:60)

2.2.5 Kamus Data (*Data Dictionary*)

Menurut Yakub (2012:168) “Kamus data (*data dictionary*) merupakan daftar elemen data yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem sehingga user dan analisis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan data *storage*”.



Menurut Sukamto dan Shalahuddin (2013:73) “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada *DFD*. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) data keluaran (*output*) dapat dipahami secara umum”.

Adapun simbol-simbol yang sering digunakan dalam kamus data adalah sebagai berikut:

Tabel 2.5. Simbol-Simbol Kamus Data

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 : Sukamto dan Shalahuddin, 2013:74)

2.2.6 Daftar Kejadian (*Event List*)

Menurut Kristanto (2008:64) 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.

Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.2.7 Metode Pengujian *Black Box Testing*

Menurut Yakub (2012:150) “*Black box testing* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program”.

Menurut Sukamto dan Shalahudin (2013:275) *Black box testing* (pengujian kotak hitam), yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.



2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Sujatmiko (2012:23) “Aplikasi adalah program komputer yang dibuat oleh perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugasnya tertentu. Misalnya *Ms-Word*, *Ms-Excel*. *Application* berbeda dengan sistem operasi (yang menjalankan komputer), *utiliy* (yang melaksanakan perawatan atau tugas-tugas umum) dan bahasa.

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

Dari beberapa pendapat diatas dapat disimpulkan bahwa aplikasi adalah sebuah *software* yang dijadikan alat untuk melakukan tugas tertentu sesuai dengan kemampuan yang dimilikinya.

2.3.2 Pengertian Pembayaran

Menurut Tim Pustaka Phoenix (dalam Septania,2014:8) “Pembayaran adalah perbuatan membayarkan atau membayar”.

Menurut Kamus Besar Bahasa Indonesia definisi pembayaran adalah dari suku kata “*pem.ba.yar.an*” yang berarti proses, cara, perbuatan, membayar.

Sistem pembayaran merupakan sistem yang berkaitan dengan pemindahan sejumlah nilai uang dari satu pihak ke pihak lain. Media yang digunakan untuk pemindahan nilai uang tersebut sangat beragam. Mulai dari penggunaan alat pembayaran yang sederhana sampai pada penggunaan sistem yang kompleks dan melibatkan berbagai lembaga.

2.3.3 Pengertian Biaya Studi

Menurut Alwi (dalam Septania,2014:8) “Biaya studi adalah uang yang dikeluarkan untuk mengadakan (mendirikan, melakukan, dsb) sesuatu”.



2.3.4 Pengertian Mahasiswa

Menurut Kamus Bahasa Indonesia (dalam landasan teori(2008)) menyatakan bahwa, “Mahasiswa adalah orang yang belajar di perguruan tinggi. Setelah menyelesaikan pendidikan di bangku sekolah, sebagian siswa ada yang menganggur, mencari pekerjaan, atau melanjutkan pendidikan ke tingkat perguruan tinggi”.

2.3.5 Aplikasi Pembayaran Biaya Studi Mahasiswa pada Sekolah Tinggi Ilmu Ekonomi (STIE) Aprin Palembang

Aplikasi Pembayaran Biaya Studi Mahasiswa pada Sekolah Tinggi Ilmu Ekonomi (STIE) Aprin Palembang merupakan aplikasi yang berguna untuk mempermudah pegawai keuangan mengelola pembayaran biaya studi mahasiswa. Mengenai pembayaran biaya Pendaftaran, Pembangunan, Registrasi masuk, SPP (Pagi, Sore dan Sabtu), Konversi, Her-registrasi, Bimbingan , Biaya ujian per semester dan Biaya ujian akhir dan wisuda.

2.4 Teori Program

2.4.1 Pengertian PHP

Menurut Prasetio (2015:130) “*PHP (PHP:Hypertext Preprocessor)* adalah bahasa *script* yang ditanam di sisi sever. Proses *PHP* dijalankan di server sat sebuah halaman dibuka dan mengandung kode *PHP*, prosesor itu akan menerjemahkan dan mengeksekusikan semua perintah dalam halaman tersebut, dan kemudian menampilkan hasilnya ke browser sebagai halaman html biasa.

Menurut Raharjo (2016:38) *PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari web browser, program yang ditulis dengan *PHP* akan di-*parsing* didalam web server oleh *interpreter PHP* dan diterjemahkan ke dalam dokumen *HTML*, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program *PHP* dilakukan di lingkungan web server, *PHP* dikatakan sebagai bahasa sisi server (*server-side*). kode *PHP* tidak akan terlihat pada saat user memilih



perintah “*View Source*” pada *web browser* yang mereka gunakan. Kode *php* diawali dengan tag `<?php` dan diakhiri dengan tag `<?>`.



Gambar 2.2 Logo PHP

2.4.2 Script PHP Sederhana

Menurut Prasetio (2015:142) Perintah “*echo*” dalam kode ini artinya tampilan pada layar *web browser* saat membuka file tersebut.

```
<?php
```

```
Echo (“Anda berada di situs Prothelon”);
```

```
?>
```

2.4.3 Pengertian MySQL

Menurut Puspitosari (2013:3) “*MySQL* adalah salah satu dari sekian banyak *database Managemen System (DBMS)* yang menganut atau mengimplementasikan *database relasional* yang disebut juga *relasional database Management System (RDBMS)*”.

Menurut Setiawan (2015:19) “*MySQL* merupakan aplikasi pengolah *database* yang bersifat *open source*. Dikembangkan oleh *Oracle* (sebelumnya *Sun* dan *MySQL AB*). Merupakan pengolah *database* yang paling banyak digunakan di dunia dan lazim diterapkan untuk aplikasi web”.



Gambar 2.3 Logo Mysql



2.4.4 Pengertian Xampp

Menurut Riyanto (2015:1) “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”.

Menurut Madcoms (2016:186) 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*.



Gambar 2.4 Logo Xampp

2.4.5 Pengertian Database

Menurut Puspitosari (2013:3) *Database* adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari, secara cepat. Selain berisi data, database juga berisi metadata. Metadata memiliki definisi data yang menjelaskan tentang struktur dari data itu sendiri. Sebagai contoh kita dapat memperoleh informasi tentang nama kolom dan tipe data pada sebuah tabel.

Menurut Fathansyah (2015:2) Basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.



- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis. Untuk selanjutnya didalam buku ini, kita akan menggunakan istilah **Tabel (Table)**, sebagai komponen utama pembangunan Basis Data.

2.4.6 Pengertian phpMyAdmin

Menurut 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.

Menurut Nugroho (2013:15) *PhpMyAdmin* adalah aplikasi manajemen *database server MYSQL berbasis web*. Dengan aplikasi *phpMyAdmin* kita bisa mengelola *database* sebagai root (pemilik server) atau juga sebagai user biasa, kita bisa membuat *database* baru, mengelola *database* dan melakukan operasi perintah-perintah *database* secara lengkap seperti saat kita di *MySQL Prompt (versi DOS)*



Gambar 2.5 Logo *phpMyAdmin*

Fungsi dalam halaman ini adalah sebagai pengendali *database MySQL* sehingga pengguna *MySQL* tidak perlu report untuk menggunakan perintah-perintah *SQL*. Karena dengan adanya halaman ini semua yang ada pada halaman *phpMyAdmin*.



2.4.7 Pengertian HTML

Menurut Setiawan (2015:33) “*HTML* adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai di dalam sebuah browser internet”.

Menurut Abdulloh (2016:2) *HTML* singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website*.

Beberapa tugas utama *HTML* dalam membangun *website*, di antaranya sebagai berikut :

- a. Menentukan layout *website*.
- b. Memformat teks dasar, seperti pengaturan paragraf dan format font.
- c. Membuat list.
- d. Membuat tabel.
- e. Menyisipkan gambar, video, dan audio.
- f. Membuat link.
- g. Membuat formulir, berikut ini merupakan skrip html :

```
<html>
  <head>
    <title>judul file html</title>
  </head>
  <body> isi dari file html </body>
</html>
```

Penjelasan *script html*

1. Setiap skrip HTML selalu diawali dengan tag **<html>** dan ditutup dengan **</html>**, di dalam tag ini dibagi menjadi dua bagian yaitu tag **<head></head>** dan tag **<body></body>**. Skrip yang ditulis antara **<head>** dan **</head>** merupakan skrip yang digunakan untuk pengaturan dokumen HTML menuliskan skrip CSS dan Javascript atau menuliskan skrip untuk memanggil kedua file tersebut. Skrip yang berbeda diantara tag **<body>** dan **</body>**



adalah skrip yang mengatur isi dokumen *HTML* yang akan ditampilkan di halaman website.

2. Tag `<title></title>` digunakan untuk membuat judul pada bagian *title bar browser*.
3. Tag `<link>` digunakan untuk memanggil file CSS yang mengatur desain dari dokumen *HTML* ini. Tag ini tidak memiliki tag penutup, seperti tag-tag lain. Atribut `rel` harus diisi “**stylesheet**”, sedangkan `href` diisi letak file CSS yang dipanggil karena file CSS berada di folder `lib` yang berada diluar folder, maka ditambah `../` untuk keluar dari folder diikuti nama folder, yaitu `lib/`, kemudian nama file CSS, yaitu misal `admin.css`, jika digabung menjadi “**../lib/admin.css**”.
4. Tag `<div></div>` digunakan untuk membuat bagian-bagian layout website yang terpisahkan dalam blok-blok tersendiri.
5. Tag `<p></p>` digunakan untuk membuat sebuah paragraf. Skrip `©` akan membuat simbol copyright. Untuk menyisipkan simbol-sibol lain dapat ditemukan dengan mudah di internet .