



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Sujatmiko (2012:156), komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program.

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

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

2.1.2. Pengertian Data

Yogiyanto (2005:8), data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut.

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

Dari beberapa pendapat diatas dapat disimpulkan Data adalah kumpulan informasi yang diperoleh dari suatu pengamatan, dapat berupa angka, lambing atau sifat.

2.1.3. Pengertian Pengolahan Data

Yakub (2012:75), pengolahan data adalah manipulasi atau transformasi simbol-simbol seperti; angka dan abjad untuk tujuan meningkatkan kegunaannya.’

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 yang lebih spesifik atau berupa informasi.



2.1.4. Pengertian Perangkat Keras

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”

Puspitosari (2013:7), hardware merupakan salah satu elemen dari sistem komputer suatu alat yang bisa dilihat dan diraba oleh manusia secara langsung mendukung proses komputerisasi.

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

2.1.5. Pengertian Perangkat Lunak

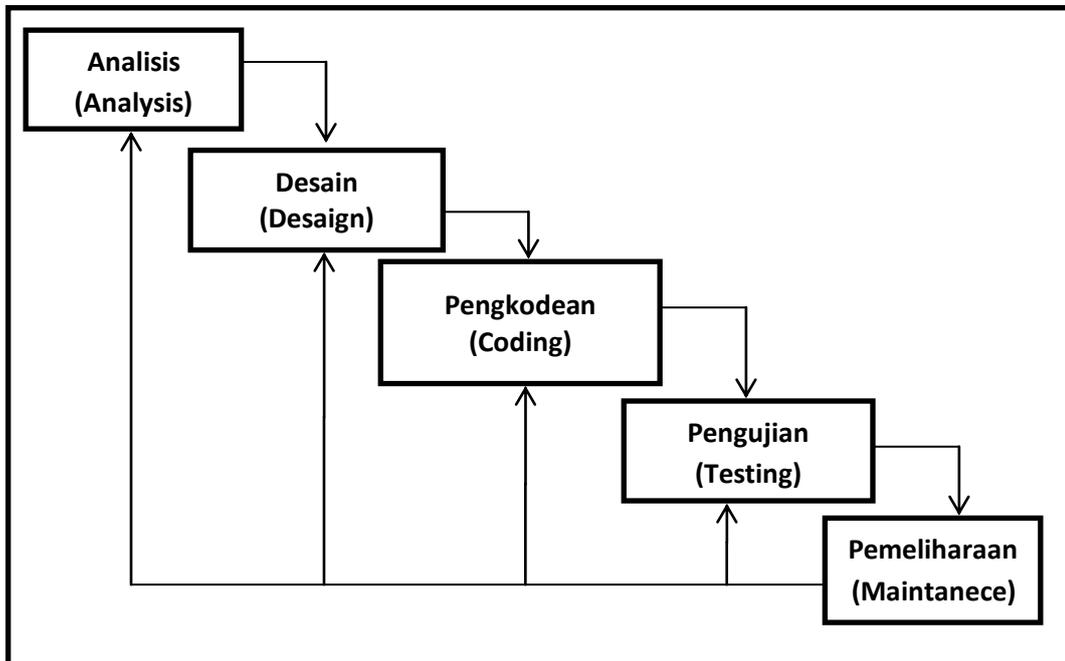
Kadir (2017:2), perangkat lunak adalah intruksi-intruksi yang ditunjukkan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai.

Sukanto 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 di atas dapat disimpulkan perangkat lunak adalah program untuk memasukan 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*

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 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 kedalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain-desain yang dibuat pada tahap desain.

4. Pengujian

Tahap pengujian focus 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 dikirmkan keuser. 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)

Sunyoto (2014:129), *Data Flow Diagram (DFD)* adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas.”

Sukamto, dan M. Shalahuddin (2013:70), *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 (output).

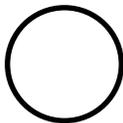


Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD :

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 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. *Breakdwon* pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

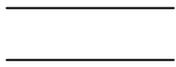
Adapun simbol-simbol *Data Flow Diagram* adalah sebagai berikut:

Tabel 2.1. Simbol-Simbol *Data Flow Diagram*

| No. | Notasi | Keterangan |
|-----|---|--|
| 1. |  | Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah |



Lanjutan 2.1. Simbol-Simbol *Data Flow Diagram*

| | | |
|----|---|--|
| | | yang harusnya di dalam kode program |
| 2. |  | File atau basisdata atau penyimpanan; pafa pemodelan perangkat lunak yang akan di implementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan. |
| 3. |  | Entitas luar (<i>external entity</i>) orang yang berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan |
| 4. |  | Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukkan. |

(Sumber:Sukamto, Ariani Rosadan M. Shalahuddin, 2013:71)

2.2.2. Pengertian ERD

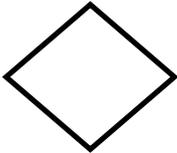
Setiawan (2015:53), *Entity Relationship Diagram* (ERD) merupakan model data berupa notasi grafis dalam pemodelan data koseptual yang menggambarkan hubungan antar penyimpanan. Entity Relationship Diagram melengkapi penggambaran grafik dari struktur logika. Dengan kata lain E-R Diagram menggambarkan arti dari aspek data seperti bagaimana entitas-entitas, atribut-atribut, dan relationship-relationship disajikan. Sebelum membuat E-R Diagram, tentunya kita harus memahami benar data yang diperlukan dan ruang lingkupnya. Di dalam pembuatan E-R Diagram perlu diperhatikan penentuan sesuatu konsep apakah merupakan suatu entitas, atribut ,atau *relationship*.



Yakub (2012:60), *Entity Relationship Diagram* (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi.

Berikut ini simbol-simbol dari ERD (*Entity Relationship Diagram*) :

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

| Simbol | Keterangan |
|---|--|
|  | Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik |
|  | Relasi, yaitu hubungan yang terjadi secara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu; satu ke banyak; dan banyak ke banyak |
|  | Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas. |
|  | Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasinya. |

(Sumber: Yakub, 2012:60)

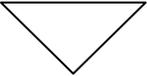
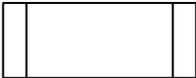
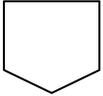
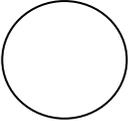
2.2.3. Blockchart

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.



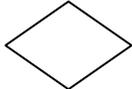
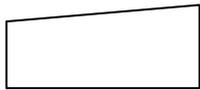
Adapun simbol-simbol *Block Chart* adalah sebagai berikut:

Tabel 2.3 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 |



Lanjutan 2.3 Simbol-Simbol *Block Chart*

| | | |
|-----|--|--|
| 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, 2008:68)

2.2.4. *Flowchart*

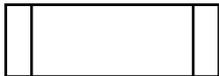
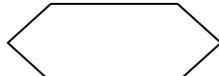
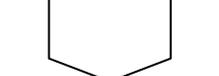
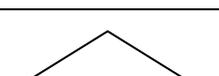
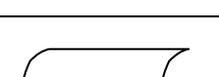
Ladjamudin (2013:211) mengemukakan bahwa, *flowchart* adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah – langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

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. |
| 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)

2.2.5. Kamus Data (*Data Dictionary*)

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*.

Sukanto dan shalahuddin (2012:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (outpput) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data dalam implementasi program dapat menjadi parameter masukkan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

- a. Nama-nama dari data
- b. Digunakan pada – merupakan proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan- seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.



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

Tabel 2.4 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 Ariani Rosa, dan M. Shalahuddin, 2012:74)

2.2.6. Daftar Kejadian (Even List)

Kristanto (2011:64) menjelaskan tentang daftar kejadian (Even List) 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 antar sistem dengan entiti luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaiknya merupakan respon dari 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 memasukan

2.2.7. Metode Pengujian Black Box Testing

Yakub (2012:150), black box testing berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkain kondisi input yang sesuai dengan persyaratan fungsional suatu program

Sukamto dan shalahudin (2013:275), blak 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 coba yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus besar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dibuat adalah :

1. Jika user memasukan nama pemakai (username) dan kata sandi (password) yang benar.
2. Jika user memasukan nama pemakai (username) dan kata sandi (password) yang salah, misalnya nama pemakai benar tetapi kata sandi salah, sebaliknya, atau keduanya salah.

2.3. Teori Judul

2.3.1. Pengertian Aplikasi

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



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 Data

Yogiyanto (2005:8), data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut.

Yakub (2012:75), pengolahan data adalah manipulasi atau transformasi simbol-simbol seperti; angka dan abjad untuk tujuan meningkatkan

Dari beberapa pendapat diatas dapat disimpulkan Data adalah kumpulan informasi yang diperoleh dari suatu pengamatan, dapat berupa angka, lambing atau sifat

2.3.3. Pengertian Pengolahan Data

Yakub (2012:75), pengolahan data adalah manipulasi atau transformasi simbol-simbol seperti; angka dan abjad untuk tujuan meningkatkan kegunaannya.

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 yang lebih spesifik atau berupa informasi.

2.3.4. Pengertian Peminjaman

Supryanto (dalam Ratika 2015:9), mengemukakan bahwa, simpan pinjam adalah koperasi dari orang-orang yang mempunyai kepentingan langsung dalam hal simpan pinjam.

2.3.5. Pengertian Pembayaran

Alwi (dalam Septiani 2014:8) mengemukakan bahwa, pembayaran adalah perbuatan membayarkan atau di bayarkan.



2.3.6. Pengertian Pengolahan Data Peminjaman Uang Dan Proses Pembayaran Pada Koperasi Zidam II/Sriwijaya

Aplikasi Pengolahan Data Peminjaman Uang Dan Proses Pembayaran Pada Koperasi Zidam II/Sriwijaya adalah suatu aplikasi yang dibuat dengan menggunakan bahasa pemrograman *PHP* dan Database *MySQL* dengan tujuan untuk membantu proses pengolahan data peminjaman dan pembayaran yang terjadi pada Koperasi Zidam II/Sriwijaya.

2.4. Teori Program

2.4.1. Pengertian PHP



Gambar 2.2. Logo PHP

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

```
<? PHP
```

```
Echo (“Anda berada disitus Prothelon”);
```

```
?>
```

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 `<?.`”

2.4.2. Pengertian PHP MyAdmin



Gambar 2.3. Logo PhpMyAdmin

Riyanto (2011:17), *PHPMyAdmin* merupakan aplikasi *web* berbasis *PHP* yang telah banyak digunakan untuk administrasi *database MySQL*. Setelah paket *XAMPP* terinstal di komputer, Anda dapat mengakses *PHPMyAdmin* melalui <http://localhost/phpmyadmin>. Karena kehandalannya, fitur *PHPMyAdmin* 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.



2.4.3. Pengertian XAMPP



Gambar 2.4. Logo XAMPP

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.

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*.



2.4.4. Pengertian *MYSQL*

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).

Setiawan (2016:19), “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 2.5. Logo Mysql

2.4.5. Pengertian HTML

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

Abdulloh (2016:2) menyatakan bahwa *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 dihalaman 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 .

