



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Judul

##### 2.1.1 Pengertian Aplikasi

Aplikasi merupakan perangkat lunak yang digunakan untuk mengerjakan tugas-tugas tertentu seperti *Ms. Word* dan *Ms. Excel* yang dibuat oleh perusahaan computer (Asropudin dalam Maiyana, 2018:57). Sedangkan menurut Indrajani (2018:3), Aplikasi merupakan program pemrosesan informasi yang digunakan untuk menyelesaikan tugas khusus dari pemakai komputer.

Selain itu menurut Juansyah (2015:2), Aplikasi merupakan suatu program yang siap digunakan oleh pengguna aplikasi untuk melaksanakan sasaran yang akan dituju. Dari definisi diatas penulis menyimpulkan bahwa aplikasi adalah *software* atau program yang dibuat untuk mengerjakan tugas-tugas khusus dari pemakai komputer.

##### 2.1.2 Pengertian *Monitoring*

*Monitoring* merupakan proses pengumpulan informasi secara berkala agar dapat menganalisa kegiatan program sesuai dengan ketentuan dan menyempurnakan program kegiatan selanjutnya (Siahaan dan Gusrianty, 2019:149). Sedangkan menurut Gustfriyanto dkk. (2019:266), Upaya dalam memantau pelaksanaan program sesuai dengan perencanaan dan hasil yang diperoleh dari suatu program disebut *monitoring* dan evaluasi.

Selain itu menurut Apriani dkk. (2019:110), Suatu aktivitas yang dilakukan untuk memantau atau mengamati sesuatu disebut dengan *monitoring*. Berdasarkan pengertian diatas penulis menyimpulkan bahwa *monitoring* atau pengawasan merupakan suatu aktivitas untuk memantau pelaksanaan program atau suatu kegiatan agar program kegiatan selanjutnya sesuai dengan ketentuan.



### 2.1.3 Pengertian Kantor Cabang

Kantor merupakan satuan unit organisasi untuk membantu pimpinan organisasi yang terdiri dari tempat, personel dan operasi ketatausahaan (Asriel dkk., 2016:2). (<https://bookwormaisyah.blogspot.com> diakses 10 Mei 2020) mengemukakan bahwa, “Kantor Cabang adalah kantor perwakilan yang didirikan oleh kantor pusat dimana cabang didirikan di lokasi yang berbeda dengan lokasi kantor pusat”.

Jadi dapat penulis simpulkan pengertian kantor cabang adalah kantor yang mengurus kepentingan suatu perusahaan ditempat lain yang kedudukannya berada dibawah kantor pusat.

### 2.1.4 Pengertian *Android*

Yudhanto dan Wijayanto (2017:1), *Android* merupakan sebuah sistem operasi Linux untuk perangkat layar sentuh seperti telepon pintar dan computer tablet. Menurut Safaat H. (2015:18), *Android* merupakan sistem operasi yang terdiri dari sitem operasi, *middleware* dan aplikasi yang digunakan untuk perangkat mobile berbasis Linux.

Selain itu menurut Maiyana (2018:57), *Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux. Dari pendapat di atas dapat disimpulkan bahwa *Android* adalah sistem operasi untuk perangkat *mobile* yang berbasis Linux.

### 2.1.5 Pengertian Aplikasi *Monitoring Kantor Cabang pada Kantor Regional III PT Pos Indonesia Berbasis Android*

Aplikasi *Monitoring Kantor Cabang pada Kantor Regional III PT Pos Indonesia* adalah sistem yang dibuat untuk membantu admin kantor pusat dalam memonitor kantor – kantor cabang yang dapat diakses oleh admin kantor pusat dan admin kantor cabang serta pimpinan agar dapat mengontrol kantor cabang dengan mudah dan lebih efisien.



## 2.2 Teori Umum

### 2.2.1 Pengertian Komputer

Kadir (2017:2) mengemukakan bahwa komputer merupakan perangkat elektronik yang digunakan untuk memudahkan pengguna dalam melakukan pekerjaan. Sedangkan menurut Abdurrohman (2017:2) komputer atau *computer* yang berasal kata dari *compute* yang berarti menghitung merupakan kata ejaan dari bahasa Inggris.

Selain itu menurut Husda dan Wangdra (2016:2) komputer adalah alat untuk data yang akan diolah sesuai dengan arahan yang telah ditentukan. Dari pendapat di atas dapat disimpulkan bahwa komputer adalah peralatan elektronik untuk memudahkan pengguna dalam menyelesaikan tugasnya.

Aplikasi merupakan perangkat lunak yang digunakan untuk mengerjakan tugas-tugas tertentu seperti *Ms*.

### 2.2.2 Pengertian Perangkat Lunak

Menurut Kadir (2017:2), perangkat lunak adalah arahan-arahan yang diarahkan kepada komputer agar tugas dapat dijalankan sesuai dengan kebutuhan pengguna. Menurut Sukamto dan Shalahuddin (2018:2) perangkat lunak (*software*) adalah program komputer yang tergabung dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).

Sedangkan menurut Husda dan Wangdra (2016:37) *Software* adalah kumpulan arahan yang memiliki fungsi untuk melaksanakan perintah seperti memberikan informasi dan menentukan fungsi *hardware* serta menjalankan sistem. Jadi perangkat lunak adalah program komputer yang berfungsi untuk melaksanakan perintah pengguna dalam menyelesaikan tugas sesuai dengan kebutuhan.

### 2.2.3 Pengertian Basis Data

Abdulloh (2018:103) mengemukakan bahwa database atau basis data merupakan gabungan informasi yang disimpan dalam komputer secara teratur



hingga bisa dicek dengan suatu program komputer untuk mendapatkan informasi. Sedangkan menurut Winarno, dkk. (2014:102) Database adalah suatu tempat penyimpanan data yang beraneka ragam. Selain itu menurut Jubilee Enterprise (2017:1) database merupakan aplikasi yang digunakan untuk menyimpan kumpulan data.

Dalam membangun Aplikasi *Monitoring* Kantor Cabang, penulis menggunakan sistem *database* MySQL sebagai tempat untuk menyimpan dan mengolah data-data kantor cabang.

#### **2.2.4 Metode Pengembangan Sistem**

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan *Waterfall*. Menurut Sukanto dan Shalahudin (2018:28) 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 terurut mulai dari analisis, desain, pengkodean, pengujian dan pemeliharaan.

##### **a. Analisis Kebutuhan Perangkat Lunak**

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

##### **b. Desain**

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean.

##### **c. Pembuatan Kode Program**

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



#### d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik 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. Dalam penelitian ini pengujian sistem akan menggunakan pengujian *Black-Box*. Pengujian *Black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian metode ini memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

#### e. Pendukung (*support*) dan pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

### 2.3 Teori Khusus

#### 2.3.1 Pengertian *Unified Modeling Language* (UML)

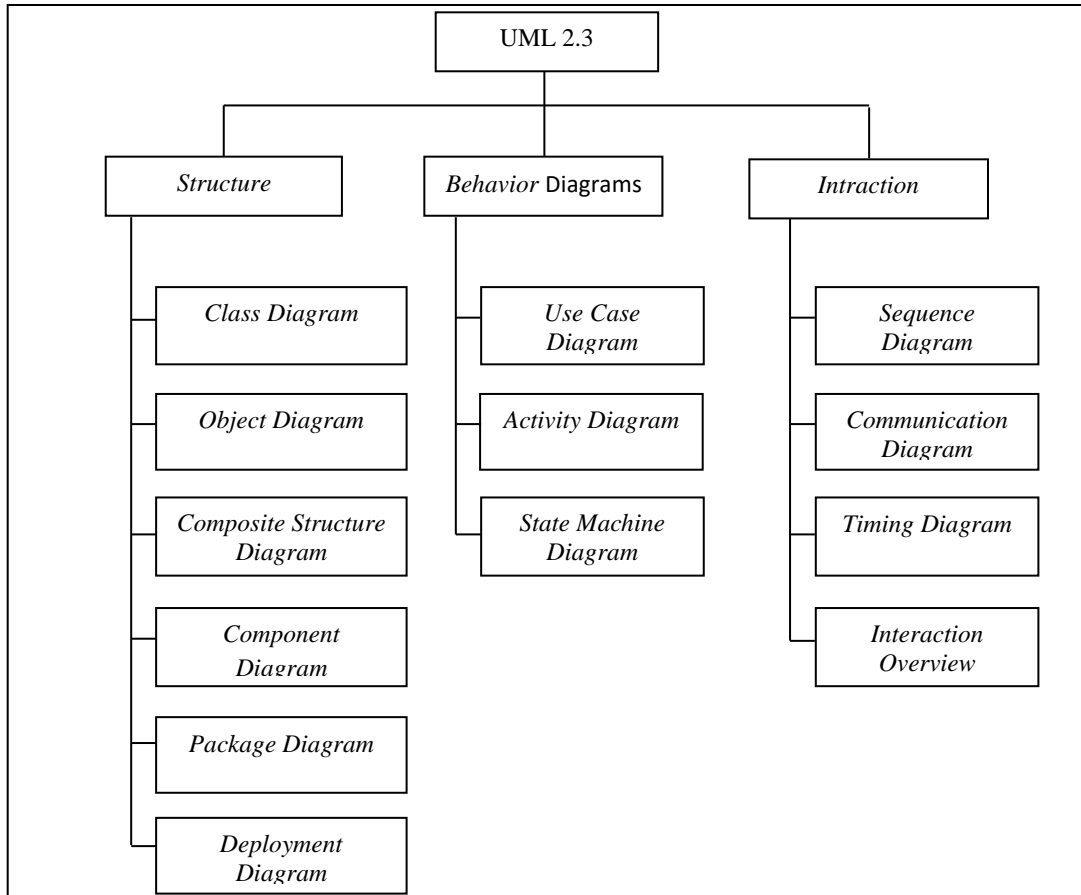
Menurut Sukanto dan Salahuddin (2018:133), *Unified Modeling Language* (UML) merupakan suatu standar bahasa yang banyak dipakai di dunia industri untuk mendeskripsikan *requirement*, membuat analisis dan desain. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan merupakan alat untuk mendukung pengembangan sistem.



**Gambar 2.1.** Tampilan Logo UML



Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



**Gambar 2.2.** Kategori dan Macam-macam *Diagram*

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141) :

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.






## 2.3.2 Jenis-Jenis Diagram UML

### 2.3.2.1 Pengertian *Use case Diagram*

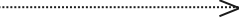
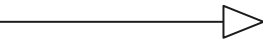
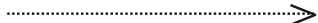
Menurut Sukamto dan Shalahuddin (2018:155), *use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Adapun simbol-simbol yang digunakan dalam *Use case* adalah sebagai berikut:

**Tabel 2.1.** Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
<p><b>Use Case</b></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.</p>
<p><b>Aktor / actor</b></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p><b>Asosiasi / association</b></p> 	<p>Komunitas antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.</p>

Lanjutan Tabel 2.1 Simbol–simbol *Use Case Diagram*

Simbol	Deskripsi
<p><b>Ekstensi / extend</b></p> <p style="text-align: center;">&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tamhanan itu; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
<p><b>Generalisasi / Generalization</b></p> 	<p>Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><b>Include</b></p> <p style="text-align: center;">&lt;&lt;include&gt;&gt;</p> 	<p>Relasi use case tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini; <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</p>

Sumber: Sukamto dan Shalahuddin (2018:156)

### 2.3.2.2 Pengertian *Class Diagram*

Menurut Sukamto dan Shalahuddin (2018:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan operasi:

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.





Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

**Tabel 2.2** Simbol-simbol *Class Diagram*

Simbol	Deskripsi
<p><b>Kelas</b></p>	Kelas pada struktur sistem.
<p><b>Antar muka/ <i>interface</i></b></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p><b>Asosiasi / <i>association</i></b></p>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p><b>Asosiasi berarah / <i>directed association</i></b></p>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p><b>Generalisasi</b></p>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
<p><b>Kebergantungan / <i>dependency</i></b></p>	Relasi antarkelas dengan makna kebergantungan antarkelas.
<p><b>Agregasi / <i>aggregation</i></b></p>	Relasi antarkelas dengan makna semua-bagian (whole-part).

Sumber: Sukanto dan Shalahuddin (2018:146)


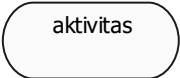





### 2.3.2.3 Pengertian Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

**Tabel 2.3** Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
<b>Status awal</b> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
<b>Aktivitas</b> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
<b>Percabangan / decision</b> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
<b>Penggabungan / join</b> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<b>Status akhir</b> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan Tabel 2.3 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
<p><b>Swimlane</b></p> <p>atau</p>	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

Sumber: Sukanto dan Shalahuddin (2018:164)

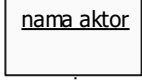

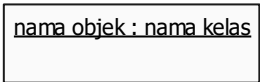

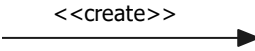
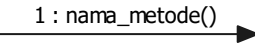
### 2.3.2.4 Pengertian *Sequence Diagram*

Menurut Sukanto dan Shalahuddin (2018:165), *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek tersebut. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :


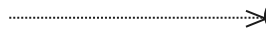
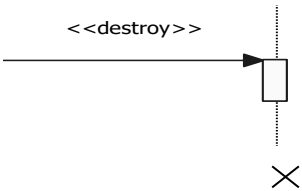
Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p><b>Aktor</b></p> <p>atau</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol</p>

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
 <b>tanpa waktu aktif</b>	dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
<b>Garis hidup / lifeline</b> 	menyatakan kehidupan suatu objek.
<b>Objek</b> 	Menyatakan objek yang berinteraksi pesan.
<b>Waktu aktif</b> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
<b>Pesan tipe create</b> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<b>Pesan tipe call</b> 	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p><b>Pesan tipe send</b></p> <p>1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
<p><b>Pesan tipe return</b></p> <p>1 : keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
<p><b>Pesan tipe destroy</b></p> <p>&lt;&lt;destroy&gt;&gt;</p> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Sukamto dan Shalahuddin (2018:165)

## 2.4 Teori Program

### 2.4.1 Pengertian XAMPP

Menurut Pendapat Winarno, dkk (2014:1) menyatakan bahwa XAMPP merupakan software web yang digunakan untuk akomodasi sistem operasi yang dipakai (X), Apache (A), MySQL (M), PHP (P) dan Perl (P). Sedangkan menurut Jubilee Enterprise (2017:94) bahwa XAMPP adalah server yang sering digunakan dan diminati karena bisa mengubah komputer menjadi MySQL Server, modul Apache-nya juga dapat digunakan untuk membantu pemrograman PHP.

Jadi XAMPP merupakan software yang digunakan untuk membantu pemrograman PHP untuk mengakomodasi sistem operasi.



### 2.4.2 Pengertian MySQL

Menurut Winarno, dkk. (2014:102) menyatakan bahwa MySQL adalah tipe data relasional yang menyimpan datanya dalam bentuk table-tabel yang saling berelasi. Selain itu menurut Raharjo (2016:241) MySQL adalah sistem database yang dipakai untuk pengembangan aplikasi *web*. Sedangkan menurut Kadir (2018:170) MySQL dibaca “Mai-es-kyu-el” adalah suatu sistem manajemen *database* yang dimiliki oleh Oracle dan MySQL Community Server yang bersifat *Open Source* merupakan salah satu produknya

Dari beberapa pendapat di atas dapat disimpulkan bahwa *MySQL* adalah sebuah program untuk membuat database yang siapa saja boleh menggunakannya.

### 2.4.3 Pengertian PHP

Abdulloh (2018:127) PHP atau *PHP Hypertext Processor* merupakan bahasa pemrograman web yang bisa ditambahkan dalam script HTML dan bekerja sama dengan server. Selain itu, menurut Jubilee Enterprise (2018:1) PHP adalah bahasa pemrograman yang dipakai untuk membangun website dinamis dan interaktif. Menurut Raharjo (2016:38) PHP merupakan salah satu bahasa pemrograman skrip yang di desain untuk membuat aplikasi web.

Dari beberapa pendapat di atas dapat disimpulkan bahwa *PHP* adalah salah satu bahasa pemrograman script yang digunakan untuk membangun website.



Gambar 2.3. Tampilan Logo PHP

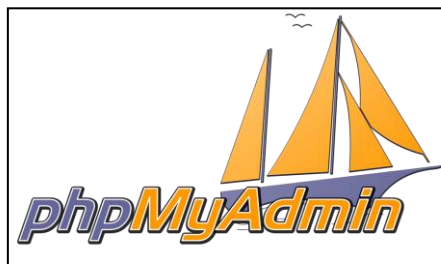
### 2.4.4 Pengertian PHPMYAdmin

Haqi dan Setiawan (2019:9), PHPMYadmin adalah salah satu bagian dalam mengelola database MySQL yang ada di computer. Sedangkan menurut



Chen (2017:163), PHPMyAdmin merupakan alat untuk administrasi database MySQL seperti database, table, indeks, trigger, user, hak akses dan lain-lain yang dibuat oleh PHP.

Jadi, penulis simpulkan pengertian *PHPMyAdmin* adalah aplikasi *PHP* sebagai administrator *MySQL* yang digunakan untuk membuat *database*, mengelola tabel, mengelola data, relasi antar table.



**Gambar 2.4.** Tampilan logo PHPMyAdmin

#### 2.4.5 Pengertian Android Studio

Android studio merupakan software IDE (*Integrated Development Environment*) yang digunakan untuk platform Android (Yudhanto dan Wijayanto, 2017:1). Selain itu menurut Fikri dkk. (2016:48), “Android Studio merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk membangun aplikasi yang berjalan pada platform android”.

Jadi, penulis simpulkan pengertian Android Studio merupakan IDE (*Integrated Development Environment*) *software* yang digunakan untuk membangaun aplikasi platform android.



**Gambar 2.5.** Tampilan logo Android Studio