



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Menurut Irma (2016:1), “Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi dibawah kontrol program yang tersimpan dimemorinya.”

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

Jadi penulis dapat menyimpulkan bahwa Komputer adalah alat bantu pemrosesan data secara elektronik mulai dari *input*, *proses*, dan *output* yang bermanfaat untuk melaksanakan berbagai pekerjaan manusia.

2.1.2 Pengertian Perangkat Lunak

Utami dan Asnawati (2015:2), “*Software* adalah perangkat lunak yang berisikan sebuah intruksi yang diperintahkan dan diproses dengan bantuan perangkat keras sehingga tanpa perangkat lunak, perangkat keras tidak bisa dipakai sehingga *software* dan *hardware* tidak bisa dipisahkan.”

Sukanto dan Shalahuddin (2018:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Jadi didapatkan kesimpulan bahwa Perangkat Lunak adalah program komputer yang diperintahkan dan diproses dengan bantuan perangkat keras yang terasosiasi dengan dokumentasi perangkat lunak untuk memudahkan pekerjaan manusia.



2.1.3 Pengertian Data

Menurut Kristanto (2018:8) menyatakan bahwa, “Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi”.

Selain itu menurut Rusdiana dan Irfan (2016:68), “Data adalah fakta-fakta mentah yang harus dikelola untuk menghasilkan informasi yang memiliki arti bagi suatu organisasi atau perusahaan.

Jadi penulis dapat menyimpulkan bahwa definisi dari data adalah fakta yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model untuk menghasilkan informasi yang memiliki arti bagi suatu organisasi atau perusahaan.

2.1.4 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2018:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).



3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut menurut Kadir (2017:3), “Perangkat lunak adalah kumpulan intruksi yang ditujukan kepada komputer. Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa intruksi-intruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi”.

Sedangkan menurut Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu”.

Dari definisi diatas penulis menyimpulkan bahwa Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya serta merupakan suatu perangkat komputer yang siap pakai bagi user untuk mengerjakan tugas tertentu.



2.2.2 Pengertian Aplikasi Persebaran Daerah Rawan Bencana Berbasis *Website* Pada Badan Penanggulangan Bencana Daerah Kota Pagar Alam.

“Aplikasi Persebaran Daerah Rawan Bencana Berbasis *Website* Pada Badan Penanggulangan Bencana Daerah Kota Pagar Alam adalah sebuah aplikasi pemetaan daerah rawan bencana yang dibuat berdasarkan *history* kejadian bencana di Kota Pagar Alam yang dapat memberikan informasi yang akurat kepada masyarakat khususnya di daerah rawan bencana agar dapat waspada terhadap bencana.”

2.3 Teori Khusus

2.3.1 Pengertian *Unified Modeling Language* (UML)

Menurut Mustakim et.al (2016:4), “UML adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.”

Menurut Nugroho (2005:16), ”UML adalah notasi-notasi yang digunakan sebagai pemodelan visual untuk sistem informasi atau perangkat lunak. Pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati”

Dari definisi di atas penulis dapat menyimpulkan bahwa pengertian UML adalah bahasa spesifikasi standar yang dapat digunakan sebagai pemodelan visual untuk sistem informasi atau perangkat lunak”.



Gambar 2.1 Tampilan Logo UML



2.3.2 Jenis-Jenis Diagram UML

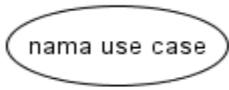
2.3.2.1 Pengertian *Use case* Diagram

Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut :

“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.”

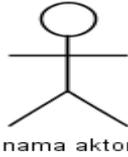
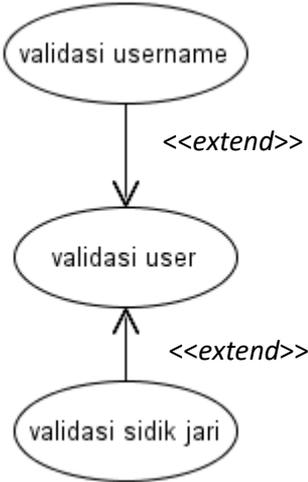
Adapun simbol-simbol yang digunakan dalam *Use case* adalah sebagai berikut:

Tabel 2.1. Simbol-simbol *Use case* Diagram

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>

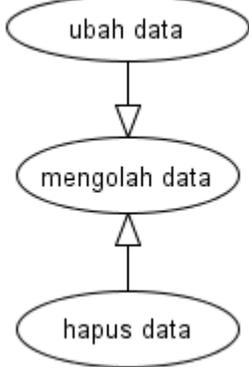
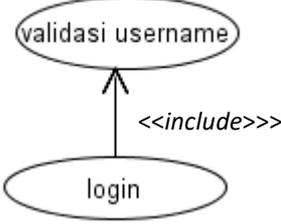


Lanjutan Tabel 2.1. Simbol-simbol Use case Diagram

Simbol	Deskripsi
<p>Aktor / <i>actor</i></p>  <p>nama aktor</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 dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
<p>Asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>



Lanjutan Tabel 2.1. Simbol-simbol Use case Diagram

Simbol	Deskripsi
<p>Generalisasi / <i>generalization</i></p> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><code><<include>></code></p>  <p><code><<uses>></code></p> 	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut: 



Lanjutan Tabel 2.1. Simbol-simbol Use case Diagram

Simbol	Deskripsi
	<ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus . berikut: <div style="text-align: center;"> <pre> graph BT A(ubah data) -- "<<include>>" --> B(validasi user) </pre> </div> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Sukamto dan Shalahuddin (2018:156)

Ada dua hal utama pada *use case* yaitu:

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2.3.2.2 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut :

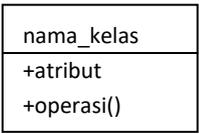
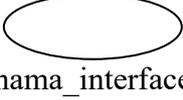
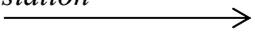
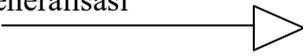
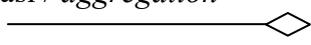
“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di



dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.”

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

Tabel 2.2 Simbol-simbol *Class Diagram*

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

Sumber: Sukamto dan Shalahuddin (2018:146)

2.3.2.3 Pengertian *Activity Diagram*

Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

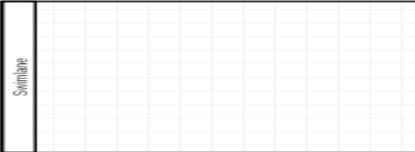
“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan



aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

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

Tabel 2.3 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Sukamto dan Shalahuddin (2018:164)



2.3.2.4 Pengertian *Sequence Diagram*

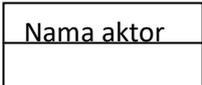
(Sukamto dan Shalahuddin, 2018:165) “Diagram sekuen menggambarkan kelakuan objek pada *Use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *Use case*”

Banyaknya diagram sekuen yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

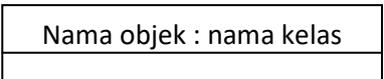
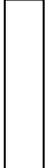
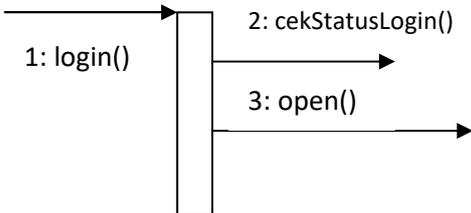
Dapat penulis simpulkan bahwa *Sequence diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada *Sequence Diagram* :

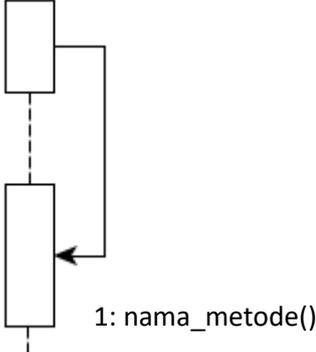
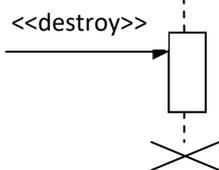
Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p>Actor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Garis hidup / <i>lifeline</i> 	menyatakan kehidupan suatu objek
Objek 	menyatakan objek yang berinteraksi pesan
Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  maka <code>cekStatusLogin ()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p>Pesan tipe <i>call</i></p> <p style="text-align: center;"><<create>></p> 	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>1: nama_metode()</p> <p>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</p>
<p>Pesan tipe <i>send</i></p> <p style="text-align: center;">1: masukan</p> 	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> <p style="text-align: center;">1: keluaran</p> 	<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>
<p>Pesan tipe <i>destroy</i></p> <p style="text-align: center;"><<destroy>></p> 	<p>menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

Sumber: Sukamto dan Shalahuddin (2018:165-167)



2.4 Teori Program

2.4.1 Pengertian XAMPP

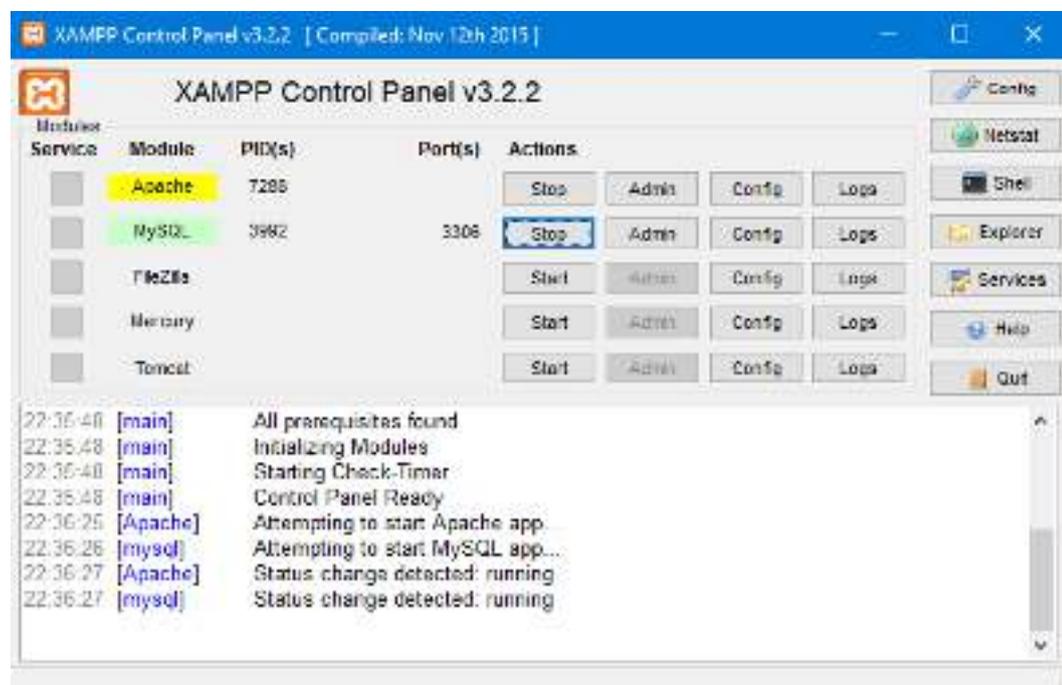
Menurut Dadan dan Kerendi (2015:28), “*XAMPP* adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *mysql* dan *phpmyadmin*.”

Sedangkan menurut Madcoms (2016:186), “*XAMPP* adalah sebuah paket kumpulan software yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *Perl*, *Filezilla* dan lain-lain.”

Dari definisi di atas penulis dapat menyimpulkan bahwa pengertian *XAMPP* adalah sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *Perl*, *Filezilla* dan lain-lain.

2.4.2 Cara Menjalankan XAMPP

Untuk menjalankan XAMPP, pertama buka aplikasi XAMPInstall. Lalu klik Start pada module Apache dan MySQL.



Gambar 2.2 Tampilan Aplikasi XAMPP



Setelah keduanya berjalan tanpa error, silahkan akses localhost menggunakan link berikut:

http://localhost

Atau

127.0.0.1

Maka Anda akan diarahkan ke halaman dashboard XAMPP.

2.4.3 Pengertian Data Base (*Basisdata*)

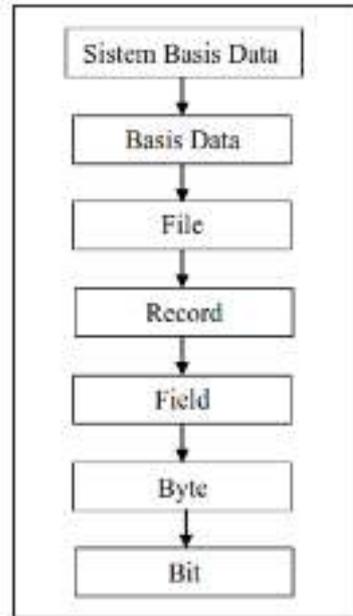
Menurut Abdullah (2015:42), “Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai(angka, deretan karakter, atau symbol)”.Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan secara cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Sedangkan Menurut Rahmad dan Dedi Setiadi (2014:1333) “Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada.” Dapat disimpulkan bahwa basis data atau database adalah sekumpulan data yang berisi deskripsi dari data yang dinyatakan dalam bentuk angka, karakter dan symbol yang dibutuhkan dalam sebuah organisasi yang diatur sedemikian rupa sehingga data tersebut dapat memenuhi kebutuhan informasi suatu organisasi.



Swara dan Pembriadi (2016:32) mengemukakan bahwa bentuk dari hierarki sebuah database dapat dilihat sebagai berikut :



Gambar 2.3 Hierarki Data Dalam Database

Hierarki data dalam data base mulai dari yang terbesar ke yang terkecil yaitu :

a. Database

Suatu database menggambarkan data yang saling berhubungan antara satu dengan yang lainnya.

b. File

Merupakan kumpulan dari record-record.

c. Record

Suatu record menggambarkan suatu atribut dari record, dimana kumpulan field membentuk suatu record.

d. Field

Suatu field menggambarkan suatu atribut dari record, dimana kumpulan field membentuk suatu record.

e. Byte

Atribut dari field berupa huruf yang membentuk nilai dari sebuah field

f. Bit

Merupakan bagian terkecil dari data secASCII (American Standar Code Form



2.4.4 Pengertian MySQL

Menurut Abdullah (2015:42), “MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user. MySQL sangat populer pengembang web (web developers) karena memiliki kecepatan dan ukuran yang kecil membuat MySQL lebih ideal untuk website ditambah lagi dengan fakta bahwa MySQL adalah open source yang berarti gratis.”

Dalam pengertian lain oleh Yuliansyah (2014:826) “MySQL adalah sebuah database manajemen system (DBMS) populer yang memiliki fungsi sebagai relational database manajemen system (RDBMS).”

Jadi, dapat disimpulkan, MySQL adalah sebuah perangkat lunak basis data yang memiliki fungsi sebagai relational database manajemen system RDBMS yang multithread dan multi-user. Ada beberapa jenis database yang diketahui yaitu Oracle, MySQL, Microsoft Access, Microsoft SQL server, IBM DB2. Dari beberapa jenis diatas penulis menggunakan MySQL karena :

1. MySQL terkenal akan keamanan datanya.
2. Tidak membutuhkan RAM besar
3. Mendukung multi user
4. Bersifat open source
5. Struktur tabel yang fleksibel
6. Keamanan yang terjamin
7. Tipe data yang bervariasi.



Contoh penulisan dalam MySQL :

```
TABLE IF NOT EXISTS `user` (
  Kode` int(5) NOT NULL AUTO_INCREMENT,
  Nama` varchar(30) NOT NULL,
  Username` varchar(20) DEFAULT NULL,
  Password` varchar(20) DEFAULT NULL,
  Level` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`Kode`)
)ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;
```

Gambar 2.4 Contoh Penulisan dalam MySQL

2.4.5 Pengertian PHP

Menurut Yudhanto dan Prasetyo (2018:7), “PHP adalah bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web. Bahasa pemrograman ini dirancang untuk para pengembang web agar dapat menciptakan suatu halaman web yang bersifat dinamis.

Sedangkan menurut Winarno (2014:49), “PHP adalah sebuah bahasa pemrograman web berbasis *server (server-side)* yang mampu mem-*parsing* kode PHP dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan *website* yang dinamis disisi *client (browser)*.

Dari definisi di atas penulis dapat menyimpulkan bahwa pengertian PHP adalah bahasa pemrograman web berbasis *server side* yang digunakan untuk membuat *website* yang bersifat dinamis dan tampilan kontennya sesuai kondisi tertentu.



Gambar 2.5 Tampilan Logo PHP



2.4.5.1 Sintaks Dasar PHP

Contoh :

File latihan1.html

```
<HTML>
  <HEAD>
    <TITLE> Latihan HTML </TITLE>
  </HEAD>
  <BODY>
    Mari Belajar Membuat Web
  </BODY>
</HTML>
```

Contoh diatas bisa ditulis dengan menggunakan PHP sebagai berikut yang kodenya disimpan dengan latihan1.php

```
<HTML>
  <HEAD>
    <TITLE>Latihan HTML</TITLE>
  </HEAD>
  <BODY>
    <?php
      Printf (“Mari Belajar Membuat Web”);
    // atau
    Echo “<br>”;
    Echo “Mari Belajar Membuat Web”;
  ?>
  </BODY>
</HTML>
```



2.4.6 Pengertian HTML

Menurut Abdulloh (2015:2) “HTML singkatan dari Hyper Text Markup Language yaitu script yang berupa tag – tag untuk membuat dan mengatur struktur website.” Menurut Harison dan Ahmad Syarif (2016:43) “HyperText Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan formating hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.”

Berdasarkan pendapat diatas, (HyperText Markup Language) HTML adalah bahasa markup berupa tag – tag yang digunakan untuk membuat halaman web dan menampilkan informasi didalam sebuah web internet.

2.4.7 Pengertian CSS

CSS merupakan sebuah pemrograman yang mempunyai kontrol terhadap semua tampilan dan desain dari sebuah halaman website (Johan dkk, 2015:134).

Menurut Rohi Abdulloh (2015:2), “CSS singkatan dari cascading style sheets, yaitu skrip yang digunakan untuk mengatur desain website. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih rapi dan indah”.

Jadi CSS atau Cascading Style Sheets adalah skrip yang digunakan untuk mengatur tampilan dan desain dari halaman website sehingga halaman website terlihat lebih rapi dan indah.



2.4.8 Pengertian *Sublime Text*



Gambar 2.6 Logo *Sublime Text*

Sublime Text Editor adalah *text editor* yang di buat untuk mempermudah pekerjaan *programmer*. *Sublime* merupakan *text editor* yang digunakan untuk banyak sekali bahasa pemrograman dan bahasa *markup*. *Sublime text editor* juga mendukung penambahan *plugin*. *Sublime* dibangun dengan menggunakan *python*.

Sublime Text memiliki banyak kelebihan diantaranya:

1. *Multiple Selection*, mempunyai fungsi untuk melakukan perubahan pada sebuah kode dalam waktu yang sama dan dalam baris yang berbeda.
2. *Command Pallete*, mempunyai fungsi yang berguna untuk mengakses file *shortcut* dengan mudah, untuk mencari file tersebut dengan menekan CTRL+SHIFT+P.
3. *Distraction free mode*, fitur ini sangat dibutuhkan oleh pengguna yang sedang fokus dalam pekerjaan, yaitu dapat merubah tampilan layar menjadi penuh dengan menekan SHIFT+F11.
4. *Find in project*, kita dapat mencari dan memiih file dalam *project* dengan mudah, dengan menekan SHIFT+P.
5. *Multi platform*, *Sublime Text* sudah tersedia dalam berbagai *platform* sistem operasi seperti Windows, Linux, Mac os.

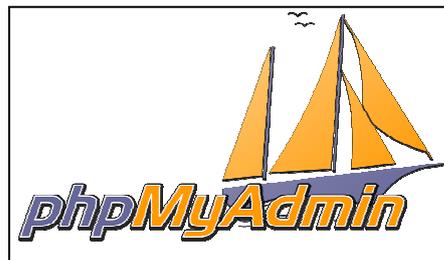


2.4.9 Pengertian PHPMyAdmin

Hikmah, dkk (2015:2), “PHPMyAdmin merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim database secara cepat dan mudah tanpa harus menggunakan perintah (*command*) *SQL* .”

Rahman (2014:12), “PHPMyAdmin adalah aplikasi PHP sebagai administrator *MySQL*, PHPMyAdmin mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, *table*, relasi antar *table* dan lain sebagainya.”

Jadi, penulis simpulkan pengertian PHPMyAdmin adalah aplikasi PHP sebagai administrator *MySQL* yang digunakan untuk membuat *database*, mengelola tabel, mengelola data, relasi antar tabel, dan mengirim database secara praktis tanpa harus menggunakan perintah (*command*) *SQL*.”



Gambar 2.7 Tampilan Logo PHPMyAdmin