



BAB 2

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Komputer

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

Sindu dan paramartha (2019:2) mengemukakan bahwa, “Komputer merupakan sebuah mesin penghitung elektronik yang cepat dapat menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan dimemorinya (stored program) dan menghasilkan output informasi.”

Jadi, dapat penulis simpulkan bahwa komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya, dan menghasilkan output program yang tersimpan dimemori.

2.1.2 Perangkat Lunak

Kadir (2017:2) mengemukakan bahwa, “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai.”

Rosa 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, dari beberapa pendapat diatas dapat disimpulkan pengertian perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak agar dapat melaksanakan tugas sesuai kehendak pemakai.

2.1.3 Data

Indrajani (2018:2), “Data adalah fakta atau observasi mentah yang biasanya mengenai fenomena fisik atau transaksi data.”



Fathansyah (2018:2), “Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.”

Dari beberapa pendapat diatas dapat disimpulkan bahwa Data adalah kenyataan terhadap fakta tentang suatu kejadian yang belum memiliki arti bagi pemakai dan perlu diolah kembali agar menghasilkan informasi yang berguna.

2.1.4 Basis Data

Yanto (2016:11), “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan membuat informasi tersedia untuk memenuhi berbagai kebutuhan.”

Rosa dan Shalahuddin (2018:43), “Basis data adalah sistem komputarisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang di olah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Jadi, penulis menyimpulkan bahwa Basis data adalah suatu kelompok data yang di kelola dan disimpan secara terkomputerisasi sehingga dapat kita gunakan sesuai keperluan.

2.1.5 Framework

Murya (2018:2) mengemukakan bahwa, “Framework secara sederhana dapat diartikan sebagai kumpulan dari fungsi atau prosedur dan class untuk tujuan tertentu yang sudah siap untuk digunakan sehingga mempermudah dan mempercepat programmer dalam membuat program tanpa harus membuat fungsi atau class dari awal.”

2.1.6 CodeIgniter

Murya (2018:1) mengemukakan bahwa, “CodeIgniter dapat disebut sebagai framework pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP dengan kerangka kerja sehingga menjadi sistematis.”



Sidik (2019:3) “CodeIgniter (CI) adalah framework PHP yang populer dan memiliki peringkat pengguna framework terbanyak di dunia, masuk ke dalam hitungan 5 besar framework PHP di dunia.

2.1.7 Metode Pengembangan Aplikasi

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Rosa 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 Rosa dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. Inception (*permulaan*)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business 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 Khusus

2.2.1 Pemrograman Berorientasi Objek Oriented (OOP)

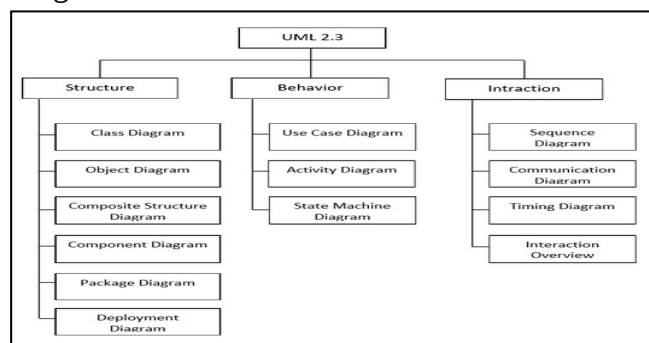
Subagia (2018:1) dalam bukunya mengatakan bahwa, “Pemrograman berorientasi objek atau object-oriented programming (OOP) merupakan suatu pendekatan pemrograman menggunakan objek dan class.”

2.2.2 UML (Unified Modeling Language)

Munawar (2018:49) mengemukakan bahwa, “UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan karena UML menyediakan Bahasa Pemodelan Visual yang memungkinkan bagi pengemban system untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.”

2.2.2.1 Macam-macam Diagram 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 2.1 Macam-macam Diagram UML



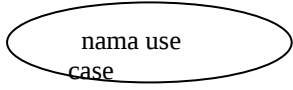

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141):

- 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.



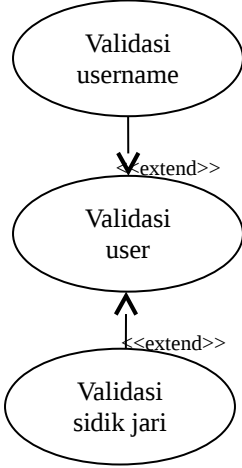
2.2.2.2 Use Case Diagram

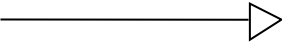
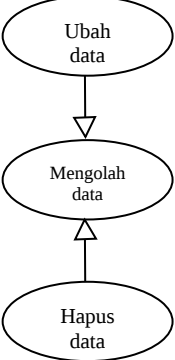
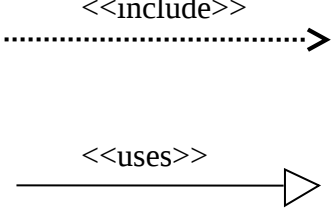
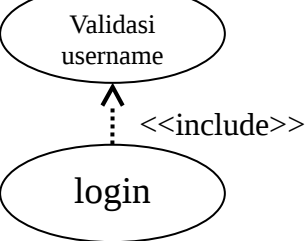
Munawar (2018:89) mengemukakan bahwa, "Use Case adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai."

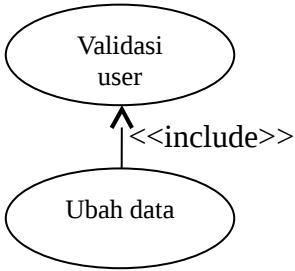
Tabel 2.1 Simbol-simbol pada Use Case Diagram

No.	Simbol	Deskripsi
1		<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awalawal frase nama use case.</p>
2		<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 aktor.</p>



No.	Simbol	Deskripsi
3	<i>asosiasi / association</i> 	komunikasi antar aktor dan use case yang berpartisipasi pada use case.
4	<i>ekstensi / extend</i> <<extend>> 	<p>relasi use case tambahan ke sebuah use case dimana use case yang di tambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misalnya</p> <div style="text-align: center;">  <pre> graph TD A([Validasi user]) -- "<<extend>>" --> B([Validasi username]) C([Validasi sidik jari]) -- "<<extend>>" --> A </pre> </div> <p>arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>

No.	Simbol	Deskripsi
5	<p><i>Generalisasi / generalization</i></p> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p>  <p>misalnya: arah panah mengarah pada use case yang menjadi generalisasinya (umum)</p>
6	<p><i>menggunakan / include / uses</i></p> 	<p>relasi tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini ada dua sudut pandang yang cukup besar mengenai include di use case:</p> <ul style="list-style-type: none"> • Include berarti use case yang ditambahkan akan selalu di panggil saat use case tambahan dijalankan, misalnya pada kasus berikut: 

No	Simbol	Deskripsi
		<p>Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah dijalankan sebelum use case tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT A([Ubah data]) -- "<<include>>" --> B([Validasi user]) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

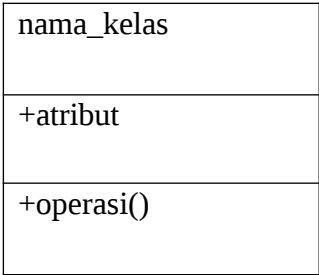
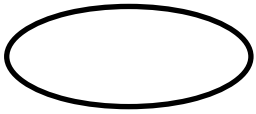
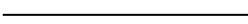
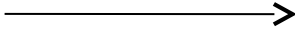
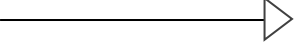
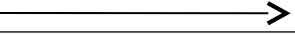
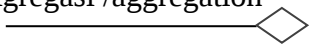
Sumber : Sukamto dan Shalahuddin (2018:155)

2.2.2.3 Class Diagram

Munawar (2018:101) mengemukakan bahwa, “Class Diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek system, tetapi juga untuk membangun kode eksekusi (executable code) dari aplikasi perangkat lunak.”

Tabel 2.2 Simbol-simbol pada *Class Diagram*

No	Simbol	Deskripsi
----	--------	-----------





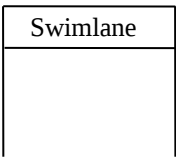
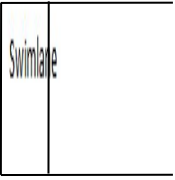
1	<p>Kelas</p> 	Kelas pada struktur sistem.
No	Simbol	Deskripsi
2	<p>antarmuka / interface</p>  <p>nama_interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	<p>Asosiasi / association</p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai multiplicity
4	<p>Asosiasi berarah / directed association</p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
5	<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6	<p>Ketergantungan / dependency</p> 	Relasi antarkelas dengan makna ketergantungan antar kelas
7	<p>Agregasi /aggregation</p> 	Relasi antarkelas dengan makna semua-bagian (whole-part)

Sumber : Sukamto dan Shalahuddin (2018:141)

2.2.2.4 Activity Diagram

Munawar (2018:127) mengemukakan bahwa, “Activity Diagram adalah bagian terpenting dari UML yang menggambarkan aspek dinamis dari system. Logika prosudural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam activity diagram. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku parallel sedangkan flowchart tidak bisa”.

Tabel 2.3 Simbol-simbol pada Activity Diagram


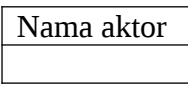

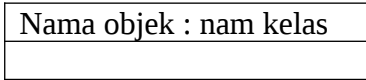

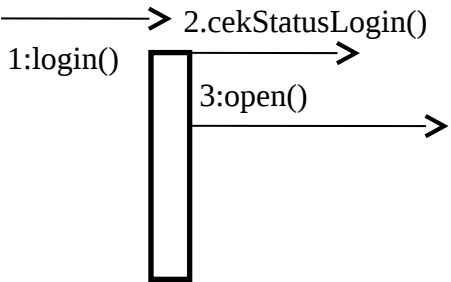
No.	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
No	Simbol	Deskripsi
3	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

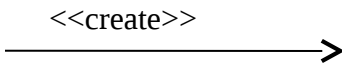
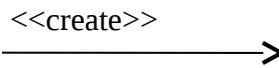
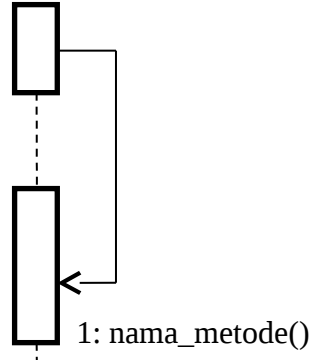
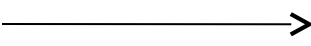
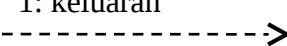
Sumber : Sukamto dan Shalahuddin (2018:161)

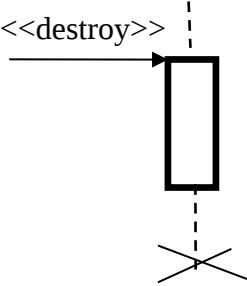
2.2.2.5 Sequence Diagram

Munawar (2018:137) mengemukakan bahwa, “Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan diantara obyek-obyek ini di dalam usecase.”

Tabel 2.4 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
1	<p>Actor</p>  <p>nama 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>
2	<p>Garis hidup /lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
3	 <p>Objek</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
4	<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cekStatusLogin () dan open()</p>

		dilakukan di dalam metode login() aktor tidak memiliki waktu aktif
5	<p>Pesan tipe create</p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
No	Simbol	Deskripsi
6	<p>Pesan tipe call</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>
7	<p>Pesan tipe send</p> <p>1: masukan</p> 	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
8	<p>Pesan tipe return</p> <p>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>

9	Pesan tipe destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy
---	---	---

Sumber : Sukamto dan Shalahuddin (2018:141)

2.3 Pengertian Judul

2.3.1 Aplikasi

Indrajani (2018:3), “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer.”

Solichin (2016:1), “Aplikasi atau perangkat lunak (*software*) merupakan bagian yang tidak terpisahkan dari suatu sistem komputer, disamping keberadaan, pengguna (*brainware*), perangkat keras (*hardware*) dan jaringan (*networking*).”

Berdasarkan pendapat diatas, maka dapat disimpulkan bahwa pengertian aplikasi adalah program yang menentukan aktivitas pemrosesan informasi untuk menyelesaikan tugas-tugas khusus dari pemakai komputer.

2.3.2 Penyewaan

Badan Pengembangan Bahasa dan Perbukuan (2016), “Penyewaan adalah proses, cara, perbuatan menyewa atau menyewakan”

2.3.3 Ruangan

Badan Pengembangan Bahasa dan Perbukuan (2016), “Ruangan adalah tempat yang lega; kamar (besar); bilik (dalam rumah); kelas (tempat belajar); tempat dalam kapal (perahu); tempat muatan; palka;”

2.3.4 Fasilitas



Badan Pengembangan Bahasa dan Perbukuan (2016), “Fasilitas adalah sarana untuk melancarkan pelaksanaan fungsi; kemudahan”

2.3.5 Website

Abdulloh (2018:1), “Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia.”

2.3.6 Aplikasi Penyewaan Ruangan beserta Fasilitas Berbasis Web pada Sudirman City Center Office Tower Beston Hotel Palembang

Aplikasi Penyewaan Ruangan beserta Fasilitas Berbasis Web pada Sudirman City Center Office Tower Beston Hotel Palembang adalah suatu aplikasi yang berfungsi untuk mempermudah dalam penyewaan dan pengolahan data pelanggan dengan cara yang lebih efektif dan efisien.

2.4 Teori Program

2.4.1 PhpMyAdmin

Madcoms (2016:186) mengemukakan bahwa, “PhpMyAdmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PhpMyAdmin, Anda dapat membuat database, membuat tabel, meng-insert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.”

2.4.2 Html

Abdulloh (2018:7), “HTML merupakan singkatan dari Hypertext Markup Language yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (World Wide Web Consortium) berupa tag-tag yang menyusun setiap elemen dari website.”

2.4.3 Xampp



Hidayatullah dan Kawistara (2017:125) dalam bukunya mengatakan bahwa, “*XAMPP support* untuk banyak sistem operasi seperti *Windows, Linux, Mac OS* dan *Solaris* sehingga tidak terdapat masalah ketika melakukan perpindahan sistem operasi.”

2.4.4 **Css**

Sidik (2018:486), “Cascading Style Sheet (CSS) adalah suatu aturan yang digunakan untuk melakukan pemformatan suatu elemen HTML, sehingga setiap elemen akan memiliki gaya (style) sendiri.”

Abdulloh (2018:45), “CSS adalah singkatan dari Cascading Style Sheet yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan.

2.4.5 **Mysql**

Menurut Hidayatullah dan Kawistara (2017:175), “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi web.”

Sementara Santoso (2017:86), “MySQL merupakan software yang tergolong sebagai DBMS (Database Management System) yang bersifat open source. Open source menyatakan bahwa software ini dilengkapi dengan source code (code yang dipakai untuk membuat MySQL). Selain tentu saja bentuk executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi dan bisa diperoleh secara gratis dengan mendownload di internet.”

2.4.6 **Sublime text**

Menurut Faridl (2015:3), “*Sublime text* adalah teks editor berbasis *Python*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal dikalangan *developer* (pengembang), penulis dan desainer. Para programmer biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan.”



Sementara menurut Murya (2017:9) mengatakan bahwa, “*sublime* atau *sublime text* merupakan text editor sama halnya dengan *Notepad++* hanya saja fitur dan *user interface* yang lengkap yang membuat editor ini banyak dilirik oleh kalangan *programmer*.”

2.4.7 Php

2.4.7.1 Pengertian PHP

Menurut Anton Subagia (2018:1) dalam bukunya mengatakan bahwa, “PHP merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah web server (server side).”

Madcoms (2016:2), “PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau di sisipkan ke dalam HTML.”

2.4.7.2 Script PHP

Yuana (2015:2), menjelaskan kode-kode PHP dituliskan diantara tanda berikut ini:

```
<?php
```

```
...
```

```
...
```

```
...
```

```
?>
```

Atau

```
<?
```

```
...
```

```
...
```

```
...
```

```
?>
```

Apabila membuat kode php dan berencana akan mendistribusikan ke pihak/orang lain, maka usahakan menggunakan sintaks `<?php ... ?>`. Hal ini dikarenakan untuk penggunaan kode yang menggunakan `<? ... ?>` terkadang tidak bisa dijalankan dalam server tertentu.

2.4.7.3 Tipe Data PHP



Tipe data merupakan jenis dari suatu data yang akan di proses oleh bahasa pemrograman. Murya (2014:26), menjelaskan beberapa tipe data dalam PHP, sebagai berikut:

1. **Integer** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. Range bilangan integer adalah antara -2.147.483.647 sampai dengan 2.147.483.647
2. **Double Floating** adalah tipe data yang berguna untuk menyimpan bilangan desimal. Range bilangan floating point antara $1e308$ sampai dengan $1e308$.
3. **Boolean** adalah tipe data yang paling sederhana, hanya berupa **TRUE** dan **FALSE**.
4. **String** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan string harus diapit dengan tanda petik, baik berupa petik tunggal (' ... ') maupun petik ganda (" ... ").
5. **Objek** adalah tipe data dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data ini bisa berupa bilangan.
6. **Array** merupakan **Tipe Compound Primitif**, terdapat pada bahasa pemrograman lain.
7. **Null** adalah tipe data yang tidak memuat apapun. Setiap variable yang diset menjadi tipe data Null, ini akan menjadikan variabel tersebut kosong.
8. **Resources** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.