



BAB II

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*).”

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.”



2.1.3 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.

2.1.4 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.”

2.1.5 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.



2.1.6 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 (*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 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.”

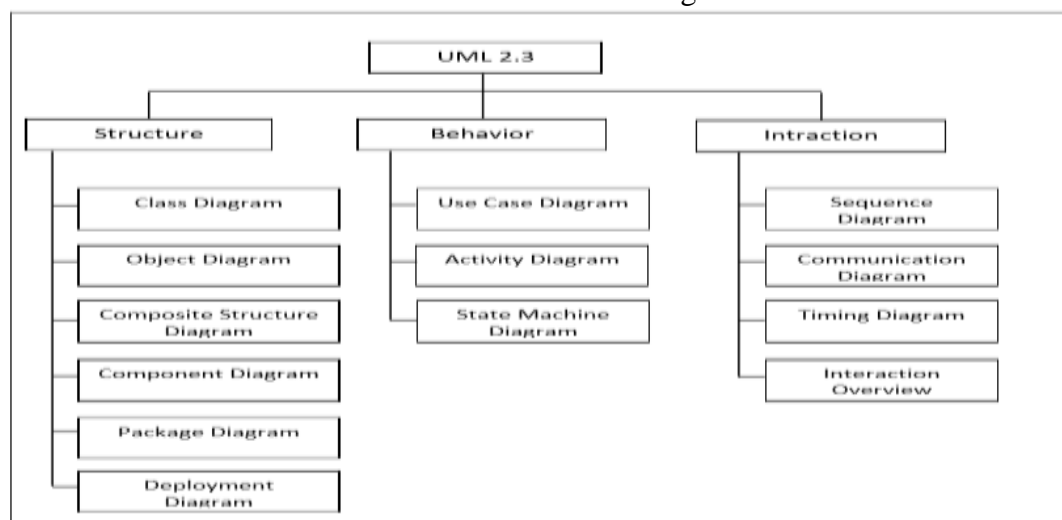
2.2.2 *Unified Modeling Language*(UML)

Munawar (2018:49) mengemukakan bahwa, “*Unified Modeling Language* (UML) 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



Sumber : Sukamto dan Shalahuddin (2018:155)



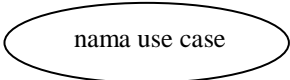
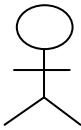
Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141):

- 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


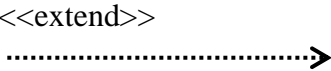
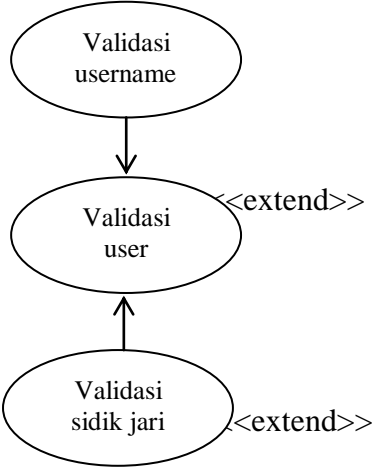
Munawar (2018:89) mengemukakan bahwa, "Use Case adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai."

Tabel 2.1 Simbol-simbol pada Use Case Diagram

No.	Simbol	Deskripsi
1	 nama use case	fungsi 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.
2	 nama aktor	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.

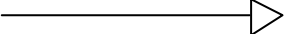
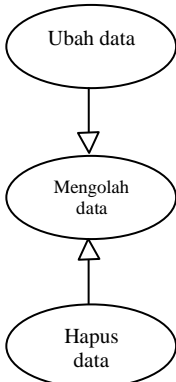


Lanjutan Tabel 2.1 Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Deskripsi
3	<i>asosiasi / association</i> 	komunikasi antar aktor dan use case yang berpartisipasi pada use case.
4	<i>ekstensi / extend</i> 	<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>  <pre>graph TD; A([Validasi username]) --> B([Validasi user]); C([Validasi sidik jari]) --> B; B --- B_ext("<<extend>>"); C --- C_ext("<<extend>>");</pre> <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>

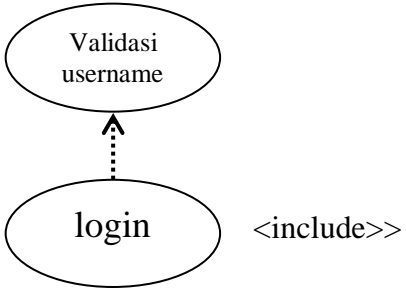
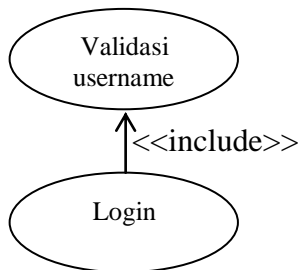


Lanjutan Tabel 2.1 Simbol-simbol pada Use Case Diagram

No.	Simbol	Deskripsi
5	<p><i>Generalisasi / generalization</i></p>  <p>Ubah data</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>.....></p> <p><<include>></p> <p>————></p> <p><<uses>></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:



Lanjutan Tabel 2.1 Simbol-simbol pada *Use Case Diagram*

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>  <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukanto 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	Kelas 	Kelas pada struktur sistem.
2	antarmuka / interface nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3	Asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai multiplicity
4	Asosiasi berarah / directed association 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6	Ketergantungan / dependency 	Relasi antarkelas dengan makna kebergantungan antar kelas
7	Agregasi /aggregation 	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
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



Lanjutan Tabel 2.3 Simbol-simbol pada *Activity Diagram*

No.	Simbol	Deskripsi
6	<p>Swimlane</p> <p>atau</p> <p>swimlane</p>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Sukamto dan Shalahuddin (2018:161)


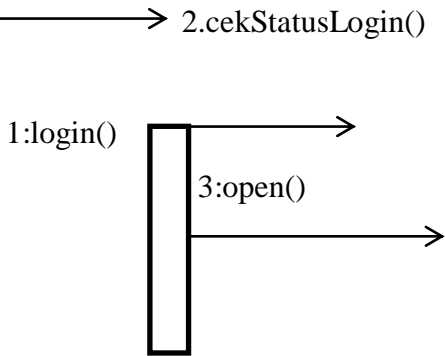
2.2.2.5 Sequence Diagram

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*



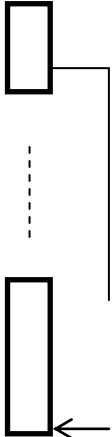

No	Simbol	Deskripsi
1	<p>Actor</p> <p>nama actor</p> <p>Atau</p> <p>tanpa waktu aktif</p>	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
2	<p>Garis hidup /lifeline</p>	Menyatakan kehidupan suatu objek

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No.	Simbol	Deskripsi
3	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Nama objek : nam kelas</div> <div style="border: 1px solid black; height: 20px; width: 100%;"></div> Objek	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya,</p> <p>misalnya</p> <p>misalnya</p>  <p>maka cekStatusLogin () dan open()</p> <p>dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>

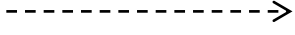
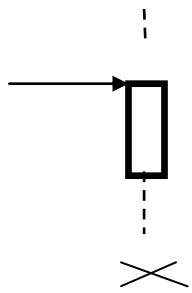


Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No.	Simbol	Deskripsi
5	<p>Pesan tipe create</p> <p><<create>></p> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6	<p>Pesan tipe call</p> <p><<create>></p>  	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 1: nama_metode() 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
7	<p>Pesan tipe send</p>  1: masukan	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim



Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No.	Simbol	Deskripsi
8	<p>Pesan tipe return</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
9	<p>Pesan tipe destroy</p> <p><<destroy>></p> 	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.”

Pane dkk (2020:132) menjelaskan bahwa “Pengertian aplikasi adalah program yang dibuat dengan tujuan untuk melaksanakan fungsi sesuai dengan kegunaan aplikasinya, penggunaanya, dan jenis aplikasi itu sendiri”.

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 Elektronik Vote (*E-Vote*)

Marlindawati (2018:24), “Pemilihan adalah suatu proses memilih dan menentukan wakil bagi suatu kelompok untuk mengisi jabatan-jabatan tertentu. Pemilihan dapat dilakukan dengan berbagai cara antara lain musyawarah mufakat untuk mencapai suatu kesepakatan atau dengan cara mengumpulkan suara terbanyak dari tiap peserta yang biasa disebut dengan *Vote*.”

2.3.3 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.”

2.3.4 Aplikasi Elektronik Vote (*E-Vote*) Bujang Gadis pada Dinas Pariwisata Kabupaten Empat Lawang Berbasis *Web Mobile View*

Aplikasi Elektronik *Vote (E-Vote)* Bujang Gadis pada Dinas Pariwisata Kabupaten Empat Lawang Berbasis *Web Mobile View* adalah suatu aplikasi yang berfungsi untuk mempermudah peserta, petugas, dan masyarakat dalam hal pendaftaran dan melakukan vote untuk menentukan pemenang *favorite* 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 *Hypertext Markup Language (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 *Cascading Style Sheet (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 *Hypertext Preprocessor (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.