



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Fauzi (2018:1), “Komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah input (processing), memberikan informasi dengan menggunakan suatu program yang tersimpan di memori komputer dan dapat menyimpan program dari hasil pengolahan yang bekerja secara otomatis”.

Menurut Krisbiantoro (2018:1), “Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan. Komputer berasal dari kata to compute yang artinya menghitung.”

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian komputer adalah alat elektronik yang dapat menerima informasi input dan menghasilkan informasi output sesuai dengan program yang tersimpan.

2.1.2 Pengertian Internet

Menurut Anhar (2016:7), “Internet adalah jaringan atau system pada jaringan computer yang saling berhubungan (terhubung) dengan menggunakan Sistem *Global Transmission Control Protocol/Internet Protocol Suite* (TCP/IP) sebagai protocol pertukaran paket (paket switching communication protocol) untuk melayani miliaran pengguna di seluruh dunia.”

Selain itu, Menurut Ismail (2020:224), “Internet (*Interconnected Network*) merupakan jaringan global yang menghubungkan computer yang satu dengan lainnya diseluruh dunia”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian internet adalah sebuah jaringan komputer yang saling berhubungan sama lain dimana dapat memberikan layanan informasi bagi pengguna.



2.1.3 Pengertian Basis Data

Menurut Yudhanto dan Prasetyo (2019:16), “Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis didalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.”

Sukanto dan Salahudin (2018:43) menjelaskan bahwa “Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Berdasarkan definisi diatas bisa disimpulkan bahwa basis data adalah sebuah kumpulan data yang disimpan secara sistematis dengan tujuan untuk memelihara data agar bisa tersedia saat data tersebut dibutuhkan.

2.1.4 Pengertian Sistem

Menurut Habibi dan Karnovi (2020:1), “Sistem adalah suatu jaringan dari prosedur-prosedur yang saling berhubungan, berkumpul dan bersama-sama untuk menyelesaikan suatu sasaran tertentu.

Dan Menurut Prehanto (2020:3), menjelaskan bahwa “Sistem merupakan bagian-bagian komponen dikumpulkan yang memiliki hubungan satu sama lain baik fisik maupun non fisik yang bersama-sama dalam bekerja demi tujuan yang dituju secara harmonis.”

Dari definisi diatas dapat disimpulkan bahwa sistem adalah bagian dari berbagai komponen atau prosedur yang terhubung antara satu dengan yang lain untuk mencapai tujuan tertentu.

2.1.5 Metode Pengembangan Sistem

Penulisan ini menggunakan metode pengembangan perangkat lunak dengan XP (*eXtreme Programming*). Menurut Suryantara (2017:23), menjelaskan bahwa “XP (*eXtreme Programming*) merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk mengembangkan aplikasi oleh para developer”. Adapun tahap-tahap (*fase*) dalam metode pengembangan XP menurut Suryantara (2017:24-25) adalah sebagai berikut :

1. Perencanaan (*Planning*)

Tahap ini dimulai dengan pemahaman konteks bisnis dari aplikasi,



mendefinisikan keluaran (*output*), fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, penentuan waktu dan biaya pengembangan aplikasi, serta alur pengembangan aplikasi.

2. Perancangan (*Design*)

Tahap ini menekankan pada desain aplikasi secara sederhana. Alat untuk mendesain pada tahap ini dapat menggunakan kartu CRC (*Class Responsibility Collaborator*). CRC digunakan untuk pemetaan (membangun) kelas-kelas yang akan digunakan pada diagram *use case*, diagram kelas, dan diagram objek. CRC diperkenalkan oleh Kent Beck dan Ward Cunningham sekitar tahun 1989 sebagai kelengkapan pemrograman berorientasi objek. CRC sebagai cikal bakal yang menjadi kelas pada saat tahap analisis.

3. Pengkodean (*Coding*)

Hal utama dalam pengembangan aplikasi dengan menggunakan XP adalah *pair programming* (dalam membuat program melibatkan 2 atau lebih *programmer*).

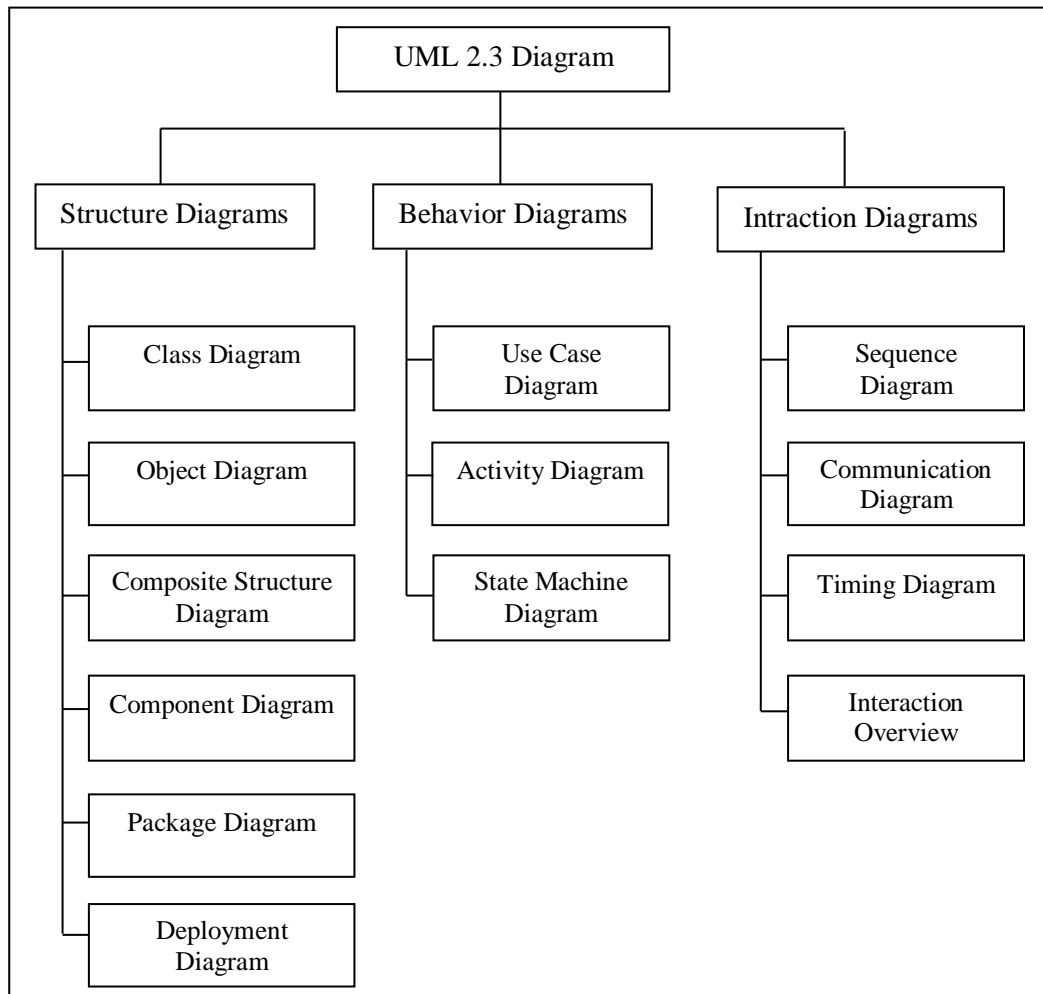
4. Pengujian (*Testing*)

Tahap ini memfokuskan pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak ada kesalahan (*error*) dan aplikasi yang dibuat sesuai dengan proses bisnis pada klien (pelanggan).

2.2 Teori Khusus

2.2.1 Pengertian UML (Unified Modeling Language)

Menurut Sukanto 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 Sukanto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah.



Gambar 2.1 Macam-macam Diagram UML

Menurut Nugroho (dalam Alda, 2021:56), “UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek”.

Dari Pengertian diatas penulis menyimpulkan bahwa *Unified Modeling Language* (UML) merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak.

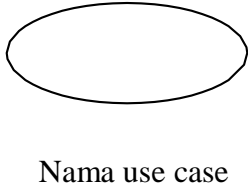
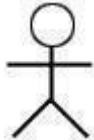
2.2.2 Pengertian Use Case Diagram

Menurut Nugroho (2017:31), “*Use Case Diagram* pada dasarnya digunakan untuk mendeskripsikan bagaimana entitas eksternal akan menggunakan sistem atau perangkat lunak. entitas eksternal itu bias saja berupa manusia atau sistem yang lain. dalam diagram *use case*, entitas eksternal ini sering dinamakan aktor.”


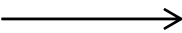
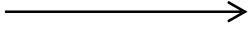
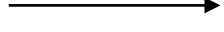
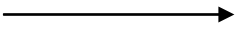
“*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan.” (Mamed Rofendy Manalu, 2015)

Sukanto, Rosa A, dan Shalahuddin (2018:155) menjelaskan bahwa *Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol-simbol Pada *Usecase Diagram*

No	Simbol	Keterangan
1.	 <p><i>Use Case</i></p> <p>Nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anatar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case
2.	 <p>Aktor/<i>actor</i></p>	Orang , proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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 diawal frase nama.

Lanjutan Table 2.1 Simbol-Simbol pada *Usecase Diagram*

3.	Asosiasi/ <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Ekstensi/ <i>extend</i>  <i><<extend>></i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	Generalisasi/  <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan/ <i>include/use</i>  <i><<include>></i>  <i><<uses>></i>	Relasi <i>use case</i> 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.

Sumber :Sukamto dan Shalahuddin (2018:156-15)

2.2.3 Pengertian Class Diagram

Menurut Nugroho (2017:34), “*Class* sesungguhnya mempresentasikan suatu konsep diskret didalam aplikasi yang dimodelkan. ini merupakan sesuatu yang bersifat fisik seperti sepeda motor, mobil, dan sebagainya. sesungguhnya kelas merupakan himpunan dari objek-objek yang memiliki struktur-struktur yang serupa, serta memiliki perilaku dan relasi yang serupa”.

Sukamto, Rosa A, dan Shalahuddin (2018:75) menjelaskan bahwa Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem.

Berikut adalah simbol- simbol yang ada pada diagram kelas:

Tabel 2.2 Simbol-simbol *Class Diagram*

No	Simbol	Keterangan
1.	<p>Kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>nama_kelas</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>+atribut</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>+operasi()</p> </div>	Kelas pada struktur sistem
2.	<p>Antarmuka/ <i>Interface</i></p> <p style="text-align: center;">○</p> <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi/</p> <p style="text-align: center;">_____</p> <p><i>association</i></p>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p>Asosiasi berarah/</p> <p style="text-align: center;">_____→</p> <p><i>directed association</i></p>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	<p>Generalisasi</p>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	<p>Kebergantungan/</p> <p style="text-align: center;">_____</p> <p><i>Dependency</i></p>	Kebergantungan antarkelas
7.	<p style="text-align: center;">_____→</p> <p>Agregasi/ <i>aggregation</i></p>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)



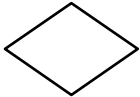

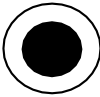
Sumber :Sukanto dan Shalahuddin (2018:146-147)

2.2.4 Pengertian Activity Diagram

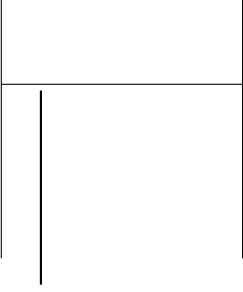
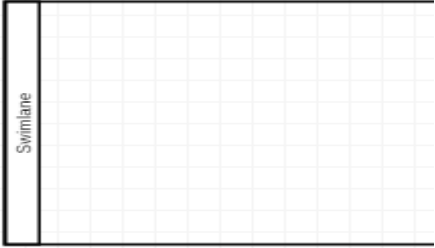
Menurut Nugroho (2017:55), “*Activity diagram*, yang prinsip dasarnya serupa dengan diagram alir (flowchart) pada paradigma pemrograman terstruktur, menggambarkan aliran aktivitas-aktivitas yang harus terjadi selama berjalannya suatu operasi atau proses. *Activity diagram* selalu memuat titik awal operasi (starting point) yang digambarkan dengan lingkaran kecil penuh dan panah-panah yang menggambarkan aliran kerja dari suatu aktivitas ke aktivitas lainnya”.

Sukanto, Rosa A, dan Shalahuddin (2018:161) menjelaskan bahwa, Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol *activity diagram* :

Tabel 2.3 Simbol-simbol *Activity Diagram*

No	Simbol	Keterangan
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan/ <i>decision</i> 	Percabangan/ <i>decision</i>
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 Table 2.3 Simbol-Symbol *Activity Diagram*

6.	<p><i>Swimlane</i> nama swimlane</p>  <p>Atau</p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
----	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

Sumber : Shalahuddin (2014 : 162-163)


2.2.5 Pengertian Sequence Diagram

Menurut Nugroho (2017:50), "Sequence Diagram merupakan model visual yang bertipe dua dimensi dan menggambarkan bagaimana interaksi dalam sistem berjalan pada rentang waktu tertentu. Pada umumnya, *sequence diagram* ini berbasis pada skenario"

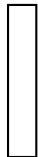

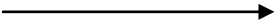
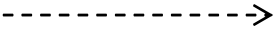

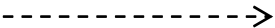
"Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara object-oriented untuk menampilkan interaksi antar objek." (Nofriyadi Jurdam, 2014)

Sukanto, Rosa A, dan Shalahuddin (2018:165) menjelaskan bahwa, Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada sekuen diagram:

Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Keterangan		
1.	Aktor  Atau <u>nama aktor</u>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi 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.		
2.	Garis hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek		
3.	Objek <table border="1" data-bbox="403 1794 673 1917"><tr><td><u>nama objek</u> :</td></tr><tr><td><u>nama kelas</u></td></tr></table>	<u>nama objek</u> :	<u>nama kelas</u>	Menyatakan objek yang berinteraksi pesan
<u>nama objek</u> :				
<u>nama kelas</u>				

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Keterangan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
5.	Pesan tipe create <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe call  1: nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.	Pesan tipe send  1: masukkan	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek yang lainnya, arah panah mengarah pada objek yang dikirim
8.	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
9.	Pesan tipe return  1: keluaran	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
10.	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: Shalahuddin (2018:165-167)

2.2.6 Kamus Data

Menurut Kristanto (2018:72), “Kamus Data (*Data Dictionary*) adalah kumpulan elemen-elemen atau symbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file di dalam system.”

Dan menurut Santi (2020:58), menjelaskan bahwa “kamus data merupakan bagian yang sangat penting agar setiap data yang ada pada model lainnya dapat diterjemahkan dengan tepat, oleh sebab itu kamus data jarang digunakan untuk sistem yang berukuran besar.”

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel 2.5 Simbol-simbol pada Kamus Data

No.	Simbol	Keterangan
1	=	Terdiri atas
2	+	Dan
3	()	Opsional
4	[]	Memilih salah satu alternative
5	**	Komentar
6	@	Identifikasi atribut kunci
7		Pemisah alternatif simbol []
8	{ }	Iterasi (Pengulangan proses)

Sumber: Kristanto (2018:72)

2.2.7 Pengertian Flowchart


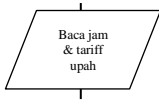

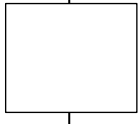

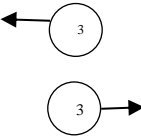

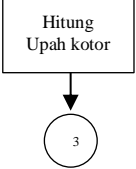
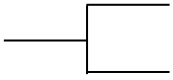
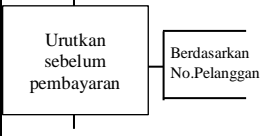
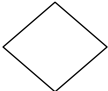
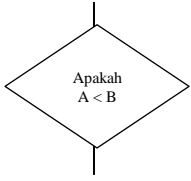

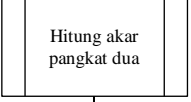
Menurut Sukamto (2018:843), "*Flowchart* dapat dimengerti dan asal katanya adalah sebuah flow atau aliran dan chart atau bagian, sehingga didapat jika dari asal katanya *flowchart* adalah sebuah bagan aliran dari sesuatu, dan sesuatu itu juga berupa aliran proses".

Adapun pendapat lain dari Sari (2017:54), mengatakan bahwa “Sebuah diagram alur adalah gambar yang menunjukkan setiap tindakan yang mungkin dapat terjadi dalam program”.


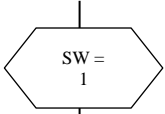
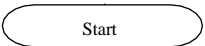
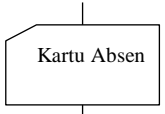

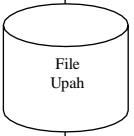

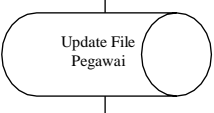

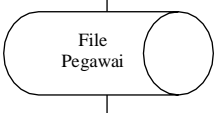

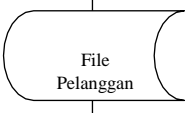



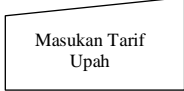
Berdasarkan pengertian di atas dapat disimpulkan bahwa pengertian *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

Simbol-simbol flowchart yang biasanya dipakai adalah simbol-simbol *flowchart* standar yang dikeluarkan oleh ANSI dan ISO, yaitu :


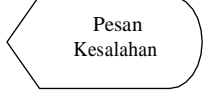
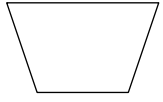
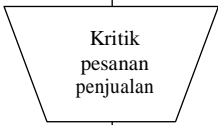
Tabel 2.6 Simbol-simbol pada *Flowchart*

No	Simbol	Arti	Contoh
1	Input / Output 	Merepresentasikan input data atau output data yang diproses atau informasi.	
2	Proses 	Merepresentasikan operasi	
3	Penghubung 	Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama	
4	Anak panah 	Merepresentasikan alur kerja	
5	Penjelasan 	Digunakan untuk komentar tambahan	
6	Keputusan 	Keputusan dalam program	
7	Predefined Process 	Rincian operasi berada ditempat lain	

Lanjutan Table 2.6 Simbol-Simbol Pada *Flowchart*

8	Preparation 	Pemberian harga awal	
9	Terminal Points	Awal / akhir Flowchart	
10	Punched Card	Input / Output yang menggunakan kartu berlubang	
11	Dokumen	I/O dalam format dicetak	
12	Magnetic Tape	I/O yang menggunakan pita magnetik	
13	Magnetik Disk 	I/O yang menggunakan disk magnetik	
14	Magnetik Drum 	I/O yang menggunakan drum magnetik	
15	Online Storage 	I/O yang menggunakan penyimpanan akses langsung	
16	Punched Tape 	I/O yang menggunakan pita kertas berlubang	
17	Manual Input 	Input yang dimasukkan secara manual dari keyboard	

Lanjutan Table 2.6 Simbol-simbol pada *Flowchart*

NO	Simbol	Arti	Contoh
18	Display 	Output yang ditampilkan oleh terminal	
19	Manual Operation 	Operasi manual	

Sumber :Sari (2017:54-57)

2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Pane, dkk (2020:8), “Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu.”

Menurut Habibi, dkk (2020:4), menjelaskan bahwa “Aplikasi adalah suatu program komputer yang bertujuan untuk mengerjakan tugas dari *user*.”

Dan menurut Habibi dan Karnovi (2020:14), “Aplikasi adalah sebuah program siap pakai yang bisa dipakai untuk menjalankan sejumlah perintah dari pengguna itu sendiri.”

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian aplikasi adalah suatu program atau perangkat lunak yang dibuat untuk memudahkan dalam melakukan tugas-tugas tertentu.

2.3.2 Pengertian Pendataan

Menurut Kuncoro dkk (2019:595), “Pendataan maupun pencatatan merupakan suatu kegiatan penghimpunan data dengan cara mencatat atau menginputkan data yang ada, baik yang lama maupun yang baru kedalam suatu media seperti buku dan catatan, maupun kedalam media elektronik seperti komputer, laptop ataupun sejenisnya sehingga menghasilkan satu kesatuan informasi.”

Menurut Ariansyah dkk (2017:26), “Pendataan merupakan proses dari pengolahan data yang dimanipulasi agar kegunaan dari data tersebut dapat menghasilkan suatu informasi yang bermanfaat”.



Dari kedua pendapat diatas, dapat disimpulkan bahwa pendataan adalah suatu kegiatan mengolah data yang mana data tersebut akan menghasilkan sebuah informasi yang bermanfaat.

2.3.3 Pengertian Website

Menurut Susilowati (2019:36), “*Website* merupakan sejumlah halaman web yang memiliki topik saling terkait antara halaman yang satu dengan halaman yang lain, terkadang disertai pula dengan gambar, *video*, *animasi*, atau jenis-jenis objek lainnya.”

Menurut Abdullah (2016:1), “*Website* adalah halaman-halaman yang berisi informasi yang ditampilkan oleh browser, seperti Mozilla Firefox, Google Chrome, atau yang lainnya.”

Berdasarkan pendapat diatas, maka penulis dapat simpulkan bahwa website adalah rangkaian atau sejumlah halaman di internet dan terdapat sebuah domain yang mengandung informasi.

2.4 Teori Program

2.4.1 Pengertian HTML

Menurut Sulistiono (2018:2), “*HTML (Hypertext Markup Language)* adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi seperti gambar, teks, video dan suara pada penjelajah web internet, yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.”

Menurut Saputra (2019:2), “*HTML* adalah sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat laman website yang dapat diakses atau ditampilkan menggunakan *web browser* (peramban web).

Menurut kedua pendapat diatas, maka penulis dapat simpulkan bahwa HTML adalah sebuah bahasa pemrograman yang mempunyai struktur tertentu yang digunakan untuk mendesain dan membuat laman website.



2.4.2 Penulisan HTML

Penulisan HTML Sebagai Berikut :

```
<html>
<head>
<title>Selamat Datang</title>
</head>
<body>
...
</body>
</html>
```

2.4.3 Pengertian Javascript

Menurut Sulistiono (2018:3), “*Javascript* adalah bahasa pemrograman tingkat tinggi untuk membuat website dinamis, dan *Jquery* adalah *library* atau pustaka dari javascript yang dirancang untuk memudahkan penerapan *client side scripting* dan menyajikan sebuah paradigma baru pada penanganan event pada Javascript.”

Dan menurut “*Javascript* adalah bahas skript dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-halaman HTML.”

Berdasarkan definisi diatas bisa disimpulkan bahwa Javascript adalah bahasa pemrograman tingkat tinggi yang dibangun untuk membuat sebuah halaman atau website dinamis dan interaktif.

2.4.4 Pengertian CSS

Enterprise (2016:94), “CSS adalah kumpulan kode untuk mendesain atau mempercantik tampilan halaman website.”

Dan menurut Sulistiono (2018:2), “CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.”

Menurut kedua pendapat diatas, maka penulis dapat simpulkan bahwa CSS adalah sebuah kode pemrograman yang dimasukkan dalam kode *HTML* yang berfungsi untuk mempercantik tampilan halaman website.

2.4.5 Penulisan CSS

2.4.5.1 Penulisan dengan Inline CSS

```
<h1 style="color: red;">
H1 dengan CSS
</h1>
```



2.4.5.2 Penulisan dengan Embedded CSS

```
<head>
<style type="text/css">
  h1 {color : red;}
</style>
</head>
```

2.4.5.3 Penulisan dengan External CSS

Buat sebuah file css, misal style.css, lalu tulis kode berikut:

```
p {font-family: arial; font-size: small;} h1 {color: red; }
```

2.4.6 Pengertian PHP

Menurut Sulistiono (2018:5), “PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat website atau situs dinamis dan menangani rangkaian bahasa pemrograman antara *client side scripting* dan *server side scripting*.”

Adi (2020:2), “PHP merupakan program *Server Side Scripting* yaitu program yang dapat dikompilasi atau diterjemahkan ke dalam server, sehingga dapat menghasilkan aplikasi web dinamis.”

2.4.7 Penulisan Program PHP

Cara penulisan skrip PHP ada dua macam, yaitu *Embedded Script* dan *Non Embedded Script*, contoh :

Embedded Script

```
<html>
  <head>
  <body>

  <?php
    Echo "Hallo Dunia";
  ?>
</body>
</ head >
</html>
```

Not Embedded Script

```
<?php
  Echo "<html>";
  Echo "<head>";
  Echo "<title>"Mengenal PHP </title>"
  Echo "<head>";
  Echo "<body>";
```



```
Echo"<p>PHP cukup menyenangkan</p>";  
Echo"<body>";  
Echo"<html>";  
?>
```

2.1.1 Pengertian MySQL

Menurut Yudhanto dan Prasetyo (2019:16), "MySQL yang merupakan singkatan dari *My Structured Query Language* adalah database yang paling favorit saat ini, program ini berjalan sebagai server yang menyediakan multi-user, mengakses ke sejumlah database baik multithread maupun multi-user, dan telah diinstal oleh sekitar 6 juta kali diseluruh dunia."

Dan menurut Fitri (2020:2), "MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License (GPL)* sehingga dapat anda pakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada."

Menurut kedua pendapat diatas bisa ditarik kesimpulan bahwa MySQL adalah sebuah *database server* yang menyediakan banyak fitur dan bersifat *General Public License*.

2.1.2 Pengertian XAMPP

Menurut Yudhanto dan Prasetyo (2019:17), "XAMPP adalah kompilasi program aplikasi gratis terfavorit di kalangan developer/programmer yang berguna untuk pengembangan website berbasis *PHP* dan *MySQL*."

Dan menurut Fitri (2020:4), "XAMPP adalah sebuah aplikasi *open source* terkait pengelolaan server yang dikembangkan oleh *Apache Friends*".

Berdasarkan kedua pendapat diatas bisa disimpulkan bahwa XAMPP adalah sebuah aplikasi yang berguna untuk pengelolaan server dan bersifat *open source*.

2.1.3 Pengertian PhpMyAdmin

Menurut Yudhanto dan Prasetyo (2019:19), "PhpMyAdmin adalah aplikasi web untuk mengelola database *MySQL* dan database *MariaDB* dengan mudah melalui antarmuka (*interface*) grafis."

Menurut Kadir (2018:172), "PhpMyAdmin adalah perangkat yang digunakan untuk memudahkan dalam mengakses *database*."

Berdasarkan kedua pendapat diatas dapat disimpulkan bahwa PhpMyAdmin adalah sebuah aplikasi web yang digunakan untuk memudahkan



dalam mengakses dan mengelola database.

2.1.4 Pengertian Laravel

Menurut Yudhanto dan Prasetyo (2019:21), “Laravel adalah salah satu *framework* PHP terbaik yang dikembangkan oleh *Taylor Otwell*, proyek Laravel dimulai pada April 2011”.

Menurut Habibi dkk (2019:8), “Laravel adalah PHP *open source framework* yang dibangun dengan model view controller dan dibekali dengan berbagai macam sintaks.”

Menurut kedua pendapat diatas, maka penulis dapat simpulkan, CodeIgniter adalah sebuah *framework* PHP yang dikembangkan oleh *Taylor Otwell* yang mana memiliki konsep model view controller didalamnya.

2.1.5 Pengertian Composer

Menurut Iksanudin (2019:257), “Composer adalah sebuah tool yang bertujuan untuk memudahkan developer dalam manage *dependency* pada proyek berbasis PHP.”

Dan menurut Habibi dkk (2019:6), “Composer merupakan tool yang didalamnya terdapat *dependency* dan kumpulan library.”

Berdasarkan kedua pendapat diatas dapat ditarik kesimpulan bahwa composer adalah sebuah alat atau tools yang didalamnya terdapat kumpulan library sehingga memudahkan kerja dari para developer.

2.1.6 Pengertian Visual Studio Code

Menurut Roza dkk (2020:78), “*Visual Studio Code* (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows.”

Dan menurut Saputra dan Aprilian (2020:101), “*Visual Studio Code* merupakan salah satu *text editor* yang paling populer dikalangan *Web text editor* diseluruh dunia.”

Berdasarkan kedua pendapat diatas dapat disimpulkan bahwa *Visual Studio Code* adalah sebuah program *text editor* yang sangat populer dikalangan para developer, yang mana *text editor* ini sendiri dibuat oleh Microsoft dan bersifat multiplatform.



2.1.7 Pengertian Bootstrap

Menurut Enterprise (2016:1), “Bootstrap adalah *framework front-end* yang intuitif dan powerful untuk pengembangan aplikasi web yang lebih cepat dan mudah.”

Menurut Sulistiono (2018:17), “Bootstrap adalah sebuah pustaka *open source* yang merupakan *framework* CSS dan *javascript* untuk membuat website yang responsive”.

Menurut kedua pendapat diatas, maka penulis dapat disimpulkan, Bootstrap adalah framework yang berisi kumpulan tool yang gratis untuk membuat layout web yang fleksibel dan responsif serta responsive secara tepat dan mudah.

