



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Wahyudin dan Munir (2018:1), “Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima input, memproses input sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan output dalam bentuk informasi”.

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

Madcoms (2014:2), “Komputer didefinisikan sebagai alat elektronik yang dapat menerima input data, mengolah data dan memberikan hasil dalam bentuk informasi dengan menggunakan suatu program yang tersimpan di memori komputer dan juga dapat menyimpan program dan hasil pengolahan yang bekerja secara otomatis”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa komputer adalah peralatan elektronik yang melaksanakan beberapa pekerjaan seperti mengolah input data menjadi hasil output bentuk informasi.

2.1.2 Pengertian Data

Rusmawan (2019:34), “Data adalah catatan atas kumpulan fakta. Data merupakan bentuk jamak dari datum, berasal dari bahasa latin yang berarti sesuatu yang diberikan”.

Hutahaean (2015:8), “Data adalah bahan mentah bagi informasi, dirumuskan sebagai kelompok lambing-lambang tidak acak menunjukkan jumlah-jumlah, tindakan-tindakan, hal-hal dan sebagainya”.

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

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa data adalah bahan mentah berupa catatan atau kumpulan fakta yang harus



melalui proses pengolahan sehingga dapat menghasilkan informasi yang dapat digunakan oleh pengguna.

2.1.3 Pengertian Informasi

Rusmawan (dalam Gaol, 2019:31), “Informasi adalah data yang telah diproses atau diolah ke dalam bentuk yang sangat berarti untuk penerimnya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya”.

Hutahaean (2015:9), “Informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi penerimanya”.

Berdasarkan dari pengertian-pengertian di atas maka dapat disimpulkan bahwa informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya, dan nilai yang dapat dipahami dalam tindakan atau keputusan.

2.1.4 Pengertian Web

Sidik (2017:1), Situs Web (*web site*) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di Internet) untuk mendapatkan informasi, dengan cukup mengklik suatu *link* berupa teks atau gambar, maka informasi dari teks atau gambar akan ditampilkan secara lebih rinci (detail).

Abdulloh (2017:32), “Web adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”.

Berdasarkan dari pengertian-pengertian di atas maka dapat disimpulkan bahwa web adalah suatu kumpulan halaman yang berisi informasi dalam bentuk data digital berupa *text*, gambar, video, audio, dan gabungan dari semuanya yang menggunakan konsep *hyperlink* yang memudahkan penelusuran informasi dengan cukup mengklik suatu *link*.



2.1.5 Pengertian Perangkat Lunak

Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti Windows, Mac OS, dan Linux, dan aplikasi seperti Microsoft Word dan Microsoft Excel adalah contoh perangkat lunak”.

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*)”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa perangkat lunak (*software*) adalah instruksi-instruksi yang ditujukan kepada komputer supaya bisa melaksanakan tugas sesuai pemakai serta merupakan program komputer yang terasosiasi dengan program perangkat lunak.

2.1.6 Pengertian Perangkat Keras

Kadir (2017:16), “Perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik”.

Mushlihudin dan Oktafianto (2016:12), “Perangkat keras yaitu komponen untuk melengkapi kegiatan memasukkan data, memproses data, dan keluaran data”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa perangkat keras (*hardware*) adalah perangkat untuk melengkapi kegiatan pada komputer yang terlihat secara fisik.

2.2 Teori Khusus

2.2.1 Pengertian *Object Oriented Programming* (OOP)

Abdullah (2017:30), OOP (*Object Oriented Program*) merupakan teknik pemrograman dengan menggunakan konsep objek. Tujuan dari OOP adalah untuk memudahkan *programmer* dalam pembuatan program dengan menggunakan konsep objek yang ada dalam kehidupan sehari-hari. Jadi setiap bagian permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.



Sebuah objek pada OOP memiliki data atau disebut *property* yang menjelaskan tentang sifat-sifat objek tersebut. Seperti sebuah *handpone* dapat memiliki data warna, merek, ukuran layar, dan sebagainya. Begitu juga dengan objek-objek yang ada di dalamnya seperti layar memiliki data berupa lebar, tinggi dan sebagainya. Selain memiliki *property*, sebuah objek dalam OOP memiliki *method* berupa fungsi yang dapat dipanggil untuk melakukan tindakan atau merubah nilai dari *property* yang ada di dalamnya. Seperti *handphone* dapat melakukan tindakan merekam, *restrart*, memanggil, mengirim pesan dan sebagainya. *Handphone* juga dapat diganti *casing* untuk mengubah warnanya (mengubah nilai *property*).

2.2.2 Pengertian *Unified Modelling Language* (UML)

Sukanto dan Shalahuddin (2018:133), *Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek

Sukanto dan Shalahuddin (2018:140), Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *Class Diagram*, *Object Diagram*, *Component Diagram*, *Composite Structure Diagram*, *Package Diagram* dan *Deployment Diagram*.
2. *Behavior Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada

sebuah sistem. *Behavior Diagram* terdiri dari *Use Case Diagram*, *Activity Diagram*, *State Machine System*.

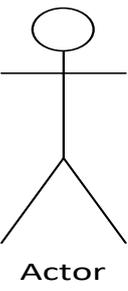
3. *Interaction Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction Diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

2.2.2.1 Pengertian Diagram Use Case

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

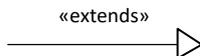
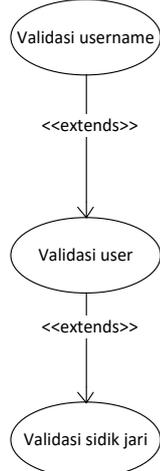
Berikut ini adalah simbol-simbol yang terdapat pada Diagram *Use Case*.

Tabel 2.1 Simbol-simbol Diagram *Use Case*

No.	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.		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 Diagram *Use Case*

No.	Simbol	Deskripsi
3.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Exstensi/ <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, 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:  <pre>graph TD; A([Validasi username]) -- <<extends>> B([Validasi user]); B -- <<extends>> C([Validasi sidik jari]);</pre> <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>
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.



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

No.	Simbol	Deskripsi
6.	Menggunakan / <i>include / uses</i> <<include>> —————> <<uses>> —————>	<p>Relasi <i>use case</i> 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 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, misal pada kasus berikut: <pre>graph BT; Login((Login)) -- <<include>> --> Validasi_username((Validasi username));</pre> <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 di jalankan, misal pada kasus berikut: <pre>graph BT; ubah_data((ubah data)) -- <<include>> --> Validasi_user((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 (2014:156).

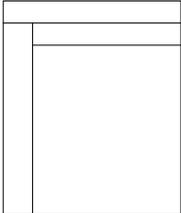


2.2.2.2 Pengertian Diagram Activity

Sukamto dan Shalahuddin (2018:161), Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang terdapat pada Diagram Activity.

Tabel 2.2. Simbol-simbol pada Diagram Activity

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas/ <i>activity</i> 	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 di mana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukamto dan Shalahuddin (2018:162).

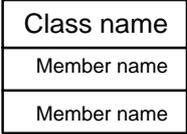
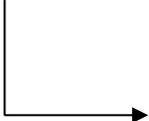


2.2.2.3 Pengertian Diagram Class

Sukanto dan Shalahuddin (2018:141), “Diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Berikut ini adalah simbol-simbol yang terdapat pada Diagram *Class*.

Tabel 2.3 Simbol-simbol pada Diagram *Class*

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antar muka/ <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum khusus).
6.	Kebergantungan/ dependensi 	Relasi antar kelas dengan makna kebergantungan antar kelas.

Lanjutan Tabel 2.3 Simbol-simbol pada Diagram *Class*

No.	Simbol	Deskripsi
7	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

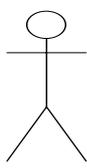
Sumber: Sukanto dan Shalahuddin (2018:147).

2.2.2.4 Pengertian Diagram *Sequence*

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

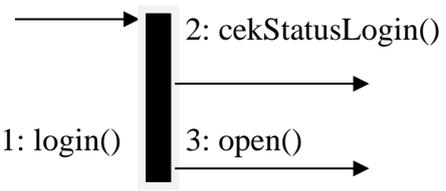
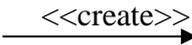
Berikut ini adalah simbol-simbol yang terdapat pada Diagram *Sequence*.

Tabel 2.4. Simbol-simbol pada Diagram *Sequence*

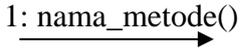
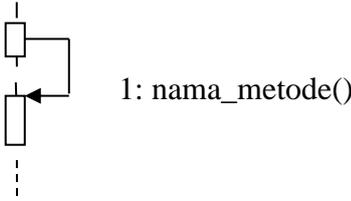
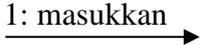
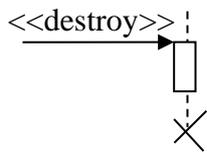
No.	Simbol	Deskripsi
1.	Aktor  Actor Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.



Lanjutan Tabel 2.4. Simbol-simbol pada Diagram *Sequence*

No.	Simbol	Deskripsi
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	Objek 	Menyatakan objek yang berinteraksi pesan.
4.	Waktuaktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:  Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Lanjutan Tabel 2.4. Simbol-simbol pada Diagram *Sequence*

No.	Simbol	Deskripsi
6.	<p>Pesan tipe <i>call</i></p> <p>1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</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 <i>send</i></p> <p>1: masukkan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
8	<p>Pesan tipe <i>return</i></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	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

Sumber: Sukanto dan Shalahuddin (2018:165).



2.3 Teori Program

2.3.1 Pengertian Basis Data

Abdulloh (2018:103), “*Database* atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Hidayatullah dan Khairul (2017:147), “Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa, basis data atau *database* adalah sekumpulan data yang disusun dalam beberapa tabel untuk menyimpan data saling berhubungan yang agar dapat diakses dengan mudah dan cepat.

2.3.2 Pengertian XAMPP

Sidik (2017:67), XAMPP secara *default* menyatukan antara HTML dan PHP, dengan direktori penyimpanan *script* terletak di c:\xampp\htdocs. Direktori c:\xampp\htdocs ini disebut sebagai direktori *root* (*root directory*), isi dari direktori ini yang akan ditampilkan jika kita mengakses lewat *browser* web, dengan URL <http://localhost>, jika kita mengakses *server* web dari komputer yang sama.

Madcoms (2016:148), “XAMPP adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyadmin, PHP, Perl, Filezilla, dan lain-lain. XAMPP berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, MySQL dan PhpMyAdmin serta *software-software* yang terkait dengan pengembangan web.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa XAMPP adalah suatu aplikasi pemrograman perangkat lunak yang dapat menyatukan antara banyak bahasa pemrograman, beberapa di antaranya ialah HTML dan PHP.



2.3.3 Pengertian MySQL

Hidayatullah dan Khairul (2017: 180), MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak *form* yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di-*bundling* dengan web *server* sehingga proses instalasinya jadi lebih mudah.

Sidik (2017:301), “MySQL merupakan *software database* yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi *query* dari *database*-nya yang saat itu bisa dikatakan paling cepat, dan jarang bermasalah. MySQL telah tersedia juga di lingkungan Windows”.

Madcoms (2016:17), “MySQL adalah sistem manajemen *database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system* (DBMS)”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa MySQL adalah sebuah aplikasi *database* yang banyak digunakan oleh para *programmer* aplikasi web dan merupakan aplikasi *database* yang paling populer pada saat ini.

2.3.4 Pengertian PHP

Hidayatullah dan Khairul (2017: 231), “*PHP Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk web *development*. Karena sifatnya yang *server-side scripting*, maka untuk menjalankan PHP harus menggunakan web *server*”.

Sidik (2017:5), “PHP: *HyperText Preprocesssor* merupakan bahasa utama *script server-side* yang disisipkan pada HTML yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi desktop”.

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



pemrograman *script server-side* yang didesain untuk pengembangan web. Disebut bahasa pemrograman *server-side* karena PHP diproses pada *computer server*".

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa PHP adalah bahasa pemrograman *script* yang bisa digabungkan ke dalam HTML yang digunakan untuk pengembangan web.

Struktur dasar penulisan PHP ialah sebagai berikut:

```
<?php           // awal tag
.....
.....           // skrip PHP
.....
?>             // akhir tag
```

2.3.5 Pengertian HTML

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

Hidayatullah dan Khairul (2017:13), *Hypertext Markup Language* (HTML) adalah bahasa standar yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan HTML yaitu

1. Mengatur tampilan dari halaman web dan isinya.
2. Membuat tabel dalam halaman web.
3. Mempublikasikan halaman web secara online.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via web.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman web.
6. Menampilkan area gambar (*canvas*) di *browser*.



Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa HTML adalah bahasa standar yang digunakan untuk halaman web, isinya berupa *tag-tag* yang menyusun elemen *website*.

Struktur dasar pada HTML seperti berikut:

```
<html>
    deskripsi
    <head>
        .....
    </head>
    Isi
    <body>
        .....
    </body>
</html>
```

2.3.6 Pengertian PhpMyAdmin

Madcoms (2016:186), “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 meng-*update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetik perintah SQL secara manual”.

Abdulloh (2016:6), “PhpMyAdmin merupakan aplikasi berbasis web yang digunakan untuk membuat *database* MySQL, sebagai tempat untuk menyimpan data-data *website*, keduanya biasanya sudah disediakan dalam satu paket aplikasi seperti Appserv atau Xampp”.

Berdasarkan dari pengertian-pengertian di atas maka dapat disimpulkan bahwa PhpMyAdmin merupakan sebuah perangkat lunak yang ditulis dalam bahasa pemrograman PHP yang berfungsi untuk memudahkan manajemen MySQL



2.3.7 Pengertian Notepad++

Madcoms (2016:15), “Notepad++ adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. Beberapa daftar bahasa program yang didukung oleh Notepad++ adalah c, c++, Java, c#, XML, PHP, Javascript dan masih banyak lagi bahasa program yang didukung.

2.3.8 Pengertian CSS

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”.

2.3.9 Pengertian Bootstrap

Abdulloh (2018:261), “Bootstrap merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. Bootstrap memungkinkan desain sebuah web menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik”.

2.4 Teori Judul

2.4.1 Pengertian Aplikasi

Kadir (2017:3), “Aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa instruksi-instruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi”.

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

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa aplikasi adalah *software* atau program yang dibuat untuk membantu para pemakai dalam mengerjakan tugas-tugas tertentu.



2.4.2 Pengertian Klaim

Suparmin (2019:209), “Klaim adalah suatu tuntutan atas suatu hak, yang timbul karena persyaratan dalam perjanjian yang ditentukan sebelumnya telah dipenuhi”.

Ilyas (2011), “Klaim adalah suatu permintaan salah satu dari dua pihak yang mempunyai ikatan, agar haknya terpenuhi. satu dari dua pihak yang melakukan ikatan tersebut akan mengajukan klaimnya kepada pihak lainnya sesuai dengan perjanjian yang disepakati bersama oleh kedua pihak”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa klaim adalah suatu tuntutan atau permintaan dari satu pihak ke pihak yang lain dalam perjanjian yang ditentukan sebelumnya.

2.4.3 Pengertian Garansi

Tjiptono (2014), “Garansi dirancang untuk meminimalisir kerugian pelanggan mengenai ketidakpuasan pelanggan setelah membeli produk. Garansi memiliki fungsi utama sebagai pengurangan risiko kerugian, sebelum dan setelah pelanggan membeli produk, juga mengharuskan perusahaan untuk menyediakan kinerja terbaik guna mendapatkan loyalitas pelanggan”.

Utami (2010), “Beberapa manfaat yang diperoleh dari garansi yaitu dapat lebih fokus dengan kebutuhan dan keinginan pelanggan, faktor mendorong perbaikan atas kesalahan, dan mewujudkan *feedback*”.

Berdasarkan dari pengertian-pengertian di atas, maka dapat disimpulkan bahwa garansi adalah salah satu komponen yang penting bagi perusahaan untuk membuat konsumen menjadi lebih percaya akan produk yang dijual oleh perusahaan tersebut. Garansi juga dapat dijadikan alat untuk meningkatkan kinerja perusahaan guna mewujudkan loyalitas pelanggan.

2.4.4 Pengertian Produk

Abdullah dan Tantri (2013:207), “Produk adalah segala sesuatu yang dapat ditawarkan ke pasar untuk mendapatkan perhatian, dibeli, dipergunakan, atau



dikonsumsi dan dapat memuaskan keinginan atau kebutuhan. Jika didefinisikan secara luas, produk meliputi objek secara fisik, pelayanan, orang, tempat, organisasi, dan gagasan”.

Sudaryono (2016: 207), “Produk merupakan sesuatu yang ditawarkan ke pasar untuk diperhatikan, dimiliki, dipakai, atau dikonsumsi sehingga dapat memuaskan keinginan dan kebutuhan”.

Berdasarkan dari pengertian-pengertian di atas, dapat disimpulkan bahwa produk adalah segala sesuatu yang ditawarkan kepada pasar baik terlihat secara fisik maupun tidak guna memenuhi kebutuhan serta keinginan dari konsumen.

2.5 Metode Pengembangan Aplikasi

Penelitian Laporan Akhir ini menggunakan metode pengembangan perangkat lunak RUP (*Rational Unified Process*). 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-fase) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. *Inception* (Pemulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modelling*) 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. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.