



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Perangkat Lunak

Kadir (2017:2), “Instruksi-instruksi yang di tunjukan 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”.

Sukanto dan Shalahuddin (2016:2), “Perangkat lunak (Software) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual).”

Berdasarkan beberapa definisi perangkat lunak diatas, dapat disimpulkan bahwa perangkat lunak adalah instruksi-instruksi atau data yang diformat secara digital, yang bisa dibaca dan ditulis oleh komputer sesuai kehendak pemakai.

2.1.2 Pengertian Internet

Fauziah (2014:1), “Internet merupakan jaringan global yang menghubungkan suatu jaringan yang satu dengan jaringan yang lainnya di seluruh dunia”. Maka dapat disimpulkan dari pengertian diatas, bahwa internet adalah jaringan global yang menghubungkan seluruh jaringan di dunia melalui jalur satelit.

2.1.3 Pengertian Komputer

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

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



Disimpulkan bahwa komputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

2.1.4 Pengertian Software

Kadir (2017:2) menyatakan, “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”.

2.1.5 Pengertian Basis Data (*Database*)

Menurut Fathansyah (2015:1), “Basis Data (*database*) adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”. Sedangkan, menurut Jubilee Enterprise (2017:40), “Basis Data (*database*) adalah “Suatu aplikasi yang menyimpan sekumpulan data, Setiap database mempunyai perintah tertentu untuk membuat akses, mengatur, mencari, dan menyalin data yang ada didalamnya”. Dan menurut Rosa dan Shalahuddin (2016:2) “Basis Data (*database*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan suatu program komputer agar dapat dimanipulasi dan diperoleh dengan cepat dan mudah.



2.1.6 Pengertian Data

Secara umum menurut Kamus Umum Bahasa Indoensia (KUBI), Data adalah bukti yang ditemukan dari hasil penelitian yang dapat dijadikan dasar kajian atau pendapat.

Betha sidik (2017:36), “data adalah catatan atas kumpulan fakta. Data merupakan bentuk jamak dari datum, berasal dari Bahasa Latin yang berarti “suatu yang diberikan”. Dan menurut Sarosa (2017:2),” Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, dan sebagainya yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasi.”.

Dari definisi diatas penulis menyimpulkan bahwa data adalah sekumpulan fakta dapat berupa angka maupun karakter yang harus diolah terlebih dahulu agar dapat menghasilkan suatu informasi yang berguna.

2.1.7 Pengertian Informasi

Sutabri (2014:2), Informasi adalah hasil pemrosesan, manipulasi, dan pengorganisasian / penataan dari sekelompok data yang mempunyai nilai pengetahuan bagi penggunanya.

2.2 Teori Khusus

2.2.1 Pengertian *Unified Modeling Language* (UML)

Rosa dan Shalahuddin (2018:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*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 menyediakan serangkaian gambar dan diagram yang sangat baik.



Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *teamprogrammer* maupun dengan pengguna.



Gambar 2.1 Logo UML

Tabel 2.1 Tipe-Tipe Diagram UML

No	Diagram	Tujuan
1.	<i>Class</i>	Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.
2.	<i>Object</i>	Menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.
3.	<i>Component</i>	Menunjukkan organisasi dan ketergantungan di antaranya kumpulan komponen dalam sebuah sistem.
4.	<i>Composite Structure</i>	Menggambarkan struktur di dalam kelas atau kolaborasi.
5.	<i>Pakage</i>	Menyediakan cara mengumpulkan elemen yang saling terkait dalam diagram UML.



6.	<i>Deployment</i>	Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.
7.	<i>Use Case</i>	Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.
8.	<i>Acitivity</i>	Menggambarkan <i>workflow</i> (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.
9.	<i>State Machine</i>	Menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.
10.	<i>Squence</i>	Menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.
11.	<i>Comunication</i>	Menggambarkan interaksi antara objek/bagian dalam bentuk urutan pengiriman pesan.
12.	<i>Timing</i>	Mendeskripsikan operasi dari alat digital kerana penggambaran secara visual akan lebih mudah dipahami dari pada kata-kata.
13	<i>Interaction Overview</i>	Menggambarkan sekumpulan urutan aktivitas.

Sumber Rosa dan Shalahuddin (2018:141-171)

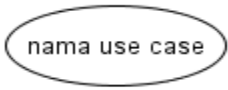
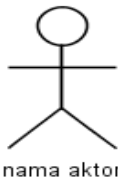
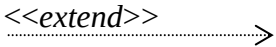
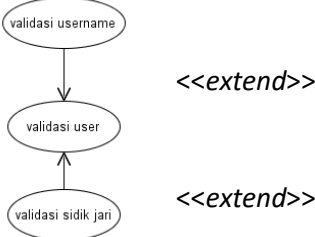
2.2.2 Jenis-Jenis Diagram UML

2.2.2.1 Use case Diagram

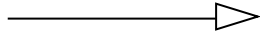
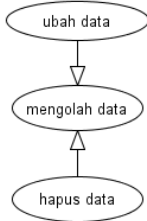
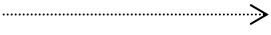
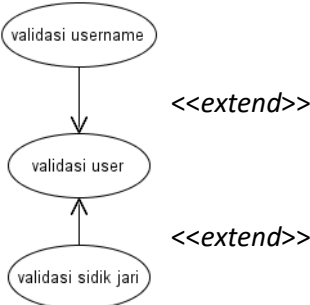
Rosa dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut :

“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut :

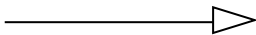
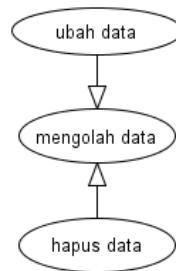
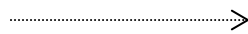
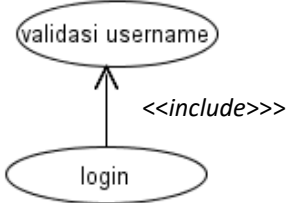
Tabel 2.2 Simbol-simbol *Use case Diagram*

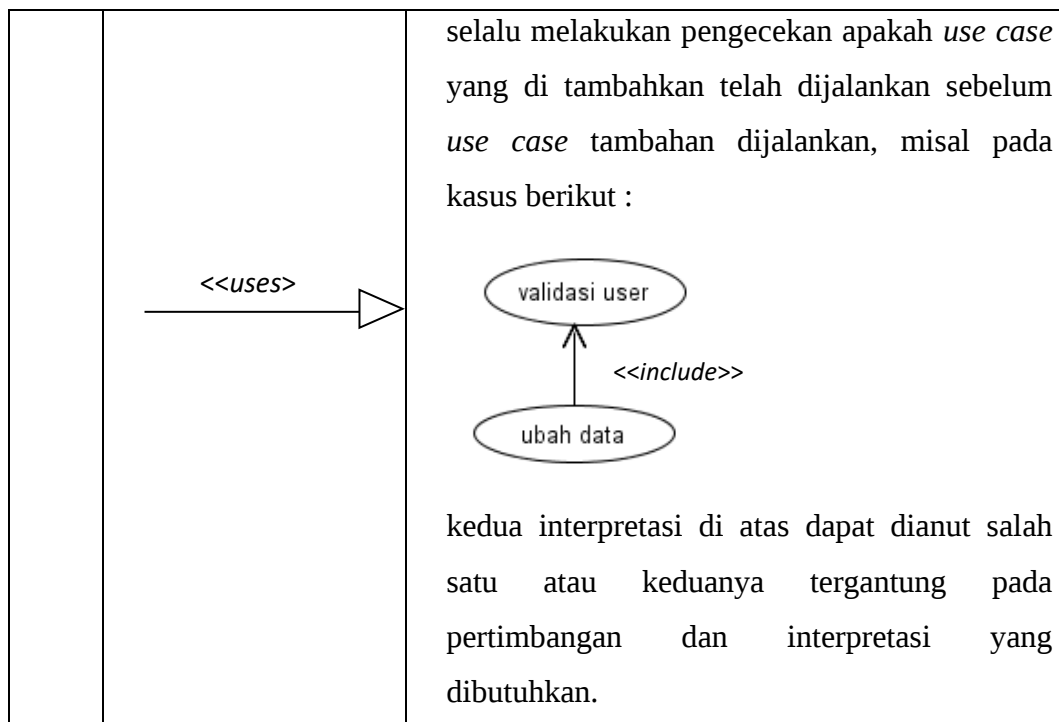
No	Simbol	Deskripsi
1.	<i>Use case</i> 	fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.
2.	aktor / <i>actor</i> 	orang, proses, atau sistem lain yang berinteraksi 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.
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> 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya 



		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.
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,  misalnya : arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	ekstensi / <i>extend</i> <<extend>> 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya  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.
7.	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,  misalnya : arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
8.	menggunakan / <i>include / uses</i>  <<include>>	relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  • <i>Include</i> berarti <i>use case</i> yang tambahan akan



Sumber: Rosa dan Shalahuddin (2018:156)

Ada dua hal utama pada *use case* yaitu :

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

2.2.2.2 Class Diagram

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

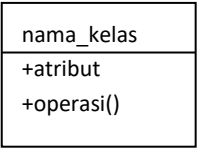
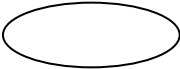
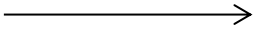
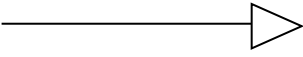
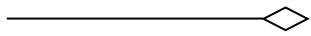
“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak



sinkron.”

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

Tabel 2.3 Simbol-simbol *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 <i>multiplicity</i> .
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus).
6.	kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).



Sumber: Rosa dan Shalahuddin (2018:146)

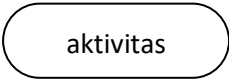
2.2.2.3 Activity Diagram

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

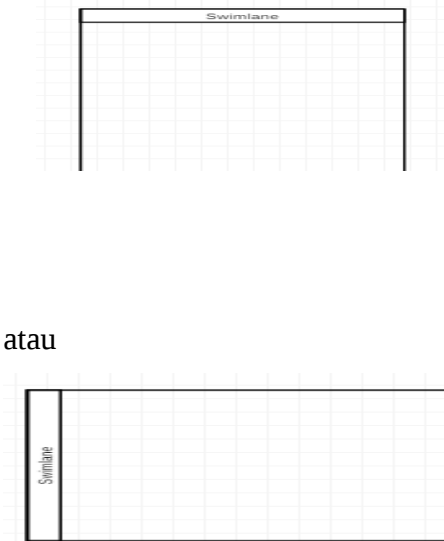
“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

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

Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
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> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.



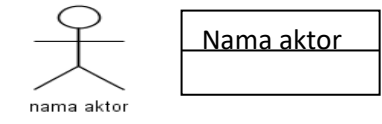
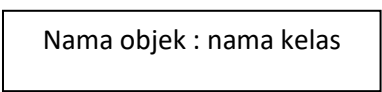
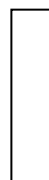
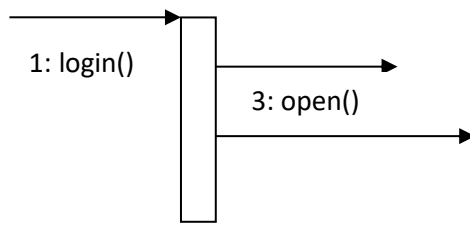
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Rosa dan Shalahuddin (2018:162)

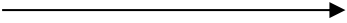
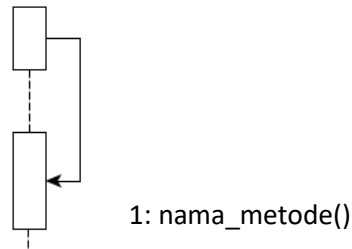
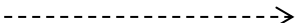
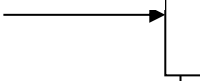
2.2.2.4 Sequence Diagram

(Rosa dan Shalahuddin, 2018:165) “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya. Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.5 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	Actor  atau tanpa waktu aktif	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.	Garis hidup / <i>lifeline</i> 	menyatakan kehidupan suatu objek.
3.	Objek 	menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, 2: cekStatusLogin() ing 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 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, 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.	Pesan tipe send 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	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.
9.	Pesan tipe destroy <<destroy>> 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada



	X	destroy.
--	---	----------

2.3 Teori Judul

2.3.1 Pengertian Website

Menurut Rohi Abdullah (2018:1), “Website dapat diartikan sebagai kumpulan 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 mengakses dan melihat oleh semua orang diseluruh dunia”.

2.3.2 Pengertian Manajemen

Menurut Hasibuan (2016:2) dalam bukunya yang berjudul Manajemen Sumber Daya manusia mendefinisikan bahwa :

“Manajemen adalah ilmu dan seni mengatur proses pemanfaatan sumber daya manusia dan sumber lainnya secara efektif dan efisien untuk mencapai suatu tujuan tertentu”.

2.4 Teori Program

2.4.1 Pengertian PHP

Menurut Abdulloh (2015:3), “PHP singkatan dari *HypertextPreprocessor* yang merupakan server-side programming, yaitu bahasa pemrograman yang diproses disisi server”. Kemudian, menurut Rosa dan Shalahudin (2017:102), “PHP (*HypertextPreprocessor*) adalah bahasa pemrograman untuk aplikasi web yang diolah oleh server web yang kemudian hasilnya dikirimkan ke browser web”. Sedangkan, menurut Winarno, dkk (2014:49),

“PHP (*HypertextProcessor*) adalah bahasa pemrograman web berbasis server-side yang mampu memarsing kode PHP dari kode web dengan ekstensi .php sehingga dapat menampilkan tampilan website yang dinamis di sisi client (browser)”. Jadi, dapat disimpulkan bahwa PHP adalah bahasa



pemrograman web berbasis server-side programming (bahasa pemrograman yang diproses di sisi server) yang memarsing kode PHP dari kode web dengan ekstensi (.php), yang kemudian akan dikirim ke browser web.

2.4.2 CSS

Menurut Abdulloh (2015:2), “CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain website”. Sehingga, dapat ditarik kesimpulan bahwa CSS adalah skrip atau kumpulan perintah yang disusun untuk mengatur desain website

2.4.3 HTML

Menurut Winarno, dkk(2014:1), “*Hypertext Markup Language* (HTML) adalah sebuah bahasa untuk menampilkan konten di web”. Sedangkan, menurut Abdulloh (2015:2), “HTML singkatan dari *HyperText Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur website”. Hal serupa dinyatakan Sujatmiko (2012:128), “HTML (*hypertext markup language*) adalah bahasa komputer yang digunakan untuk membuat sebuah halaman web atau program yang digunakan untuk menulis halaman web di internet”.

Dari beberapa definisi diatas, dapat disimpulkan Website adalah skrip yang berupa tag-tag untuk membuat struktur dan menampilkan konten website di internet.

2.4.4 Pengertian Sublime

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform *operating system* dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerful. Fungsionalitas dari aplikasi ini dapat dikembangkan



dengan menggunakan sublime-packages. Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara *default* dapat lebih dimaksimalkan atau didukung dengan menggunakan *add-ons* yang bisa didownload sesuai kebutuhan *user*'. (Hasan Tarmizi, 2015, <https://hasantarmizi.blogspot.com/2017/04/pengertian-sublime-text.html>, 30 Desember 2019).

2.4.5 Pengertian XAMPP

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat Apache (Web Server), MySQL (Database), PHP (Server Side Scripting), Perl, FTP Server, PHPMyAdmin, dan berbagai pustaka bantu lainnya.

Versi XAMPP terdiri atas aplikasi-aplikasi berikut:

1. Apache
2. Mysql
3. PHP
4. Phpmyadmin
5. Filezillaftp server
6. Tomcat
7. Strawberryperl
8. Xamppcontrol panel

(Wahana Komputer, 2014:7)



2.4.6 Sekilas Tentang MySQL

2.4.6.1 Pengertian MySQL

Menurut Winarno, dkk (2014:102), “MySQL merupakan software database tipe data relasional untuk menyimpan data dalam bentuk tabel yang saling berhubungan”. Sedangkan, menurut Hendry (2015:7), “MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*)”. Disamping itu, menurut Priyo (2014: 58), “MySQL merupakan salah satu database yang banyak digunakan oleh para pengguna komputer diseluruh dunia dengan menggunakan bahasa SQL/Structured Query Language (bahasa yang digunakan/distandarkan untuk berinteraksi dengan database) dalam pengoperasiannya”.

Dapat disimpulkan dari beberapa definisi diatas bahwa MySQL adalah software database tipe data relasional yang merupakan implementasi dari sistem manajemen basis data relasional dibawah lisensi GPL dan banyak digunakan oleh pengguna komputer diseluruh dunia untuk menyimpan data tabel yang saling berhubungan.