



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Perangkat Lunak

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*)”.

Sedangkan Swara dan Febriadi (2016:8), “Perangkat lunak merupakan seluruh perintah yang digunakan untuk memproses informasi”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan cara menggunakan yang ditujukan kepada komputer sesuai kebutuhan pemakai yang digunakan untuk memproses informasi.

2.1.2 Pengertian Komputer

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

Hal senada juga dikemukakan oleh Ikhsan dan Kurniawan (2015:13), “Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah.”

Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari komputer adalah sebuah alat elektronik yang dapat memproses data digital dan informasi yang bermanfaat bagi *user* sehingga dapat menghasilkan keluaran informasi yang dihasilkan setelah diolah.



2.1.3 Pengertian Data

Menurut Kristanto, Andi (2018:8) menyatakan bahwa, “Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi”.

“Data adalah kumpulan berupa pernyataan apa adanya dari fakta yang didapatkan dari hasil pengukuran atau pengamatan suatu variabel baik dalam bentuk angka, kata-kata, atau citra”. (<https://id.m.wikipedia.org/wiki/Data>).

Berdasarkan pendapat di atas, dapat disimpulkan bahwa data adalah kumpulan data baik berupa angka, kata-kata atau citra yang didapatkan melalui pengukuran atau pengamatan yang memerlukan pengolahan selanjutnya untuk menghasilkan informasi yang memberi manfaat bagi penerimanya.

2.1.4. Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Sukamto dan Shalahuddin (2016:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2016:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat.



Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

2.2 Teori Khusus

2.2.1 *Unified Modelling Language (UML)*

Menurut Widodo (dikutip Pranata dkk. 2015:26), “*UML* adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak). *Artifact* dapat berupa model, deskripsi atau perangkat lunak dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya”.

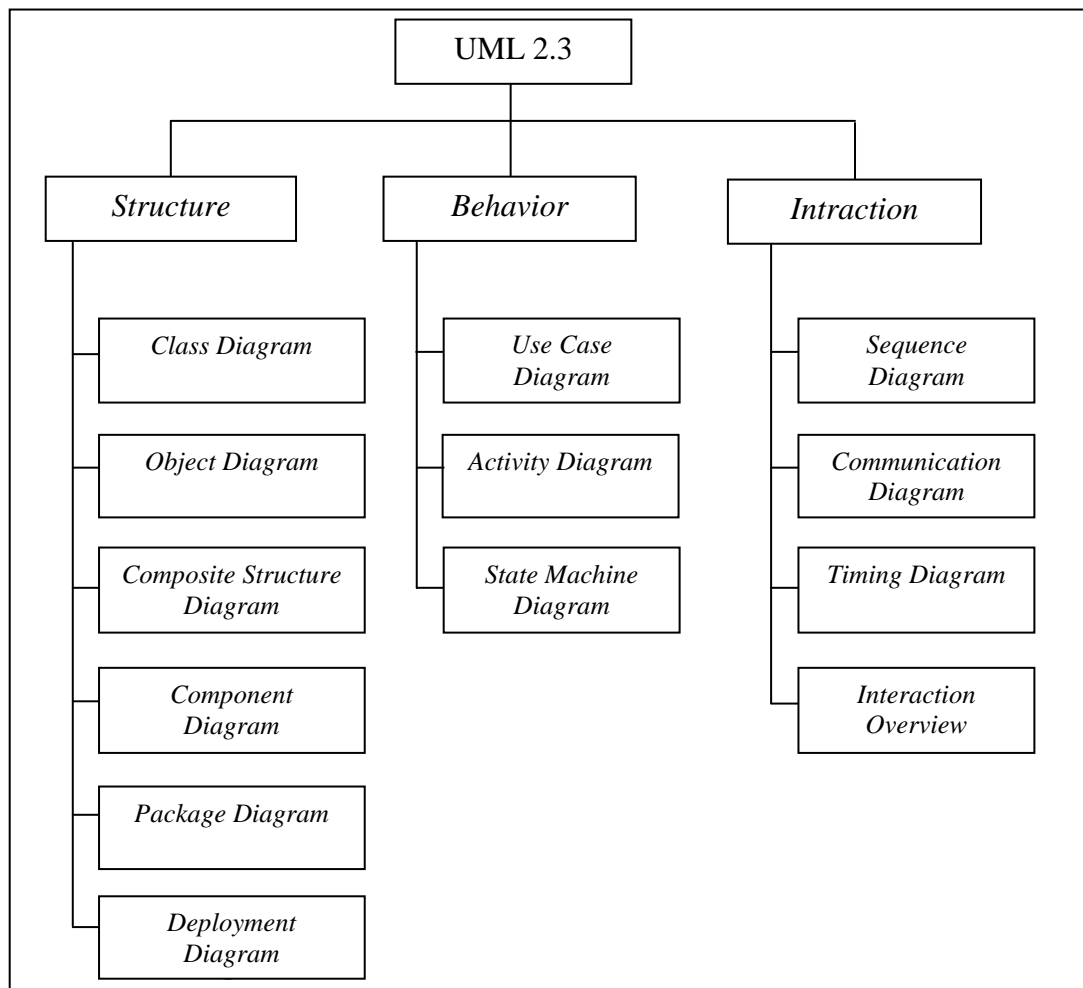
Adapun pendapat lain menurut Sukamto dan Shalahuddin (2016:133), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

Dari pengertian diatas dapat disimpulkan bahwa *UML (Unified Modelling Language)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk

memvisualisasikan hingga mendokumentasikan sebuah sistem pengembangan perangkat lunak berorientasi objek.

2.2.2 Kategori *Unified Modeling Language* (UML)

Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



(Sumber: Sukamto dan Shalahuddin (2018:140))

Gambar 2.1 Kategori dan Macam-macam Diagram UML

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

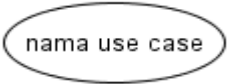
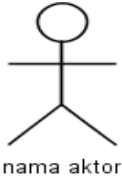
- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.3 Jenis-Jenis Diagram UML


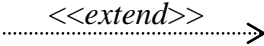
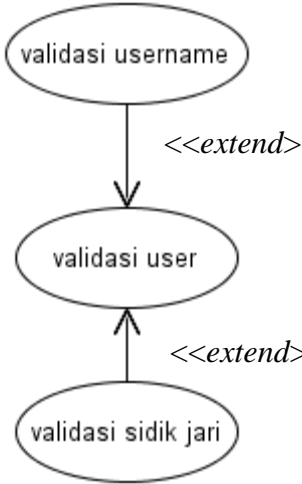
2.2.3.1 Use case Diagram

Sukamto dan Shalahuddin (2016:155), “*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:

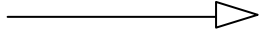
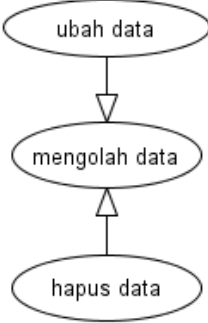
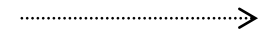

Tabel 2.1 Simbol-simbol *Use case* Diagram

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 actor.

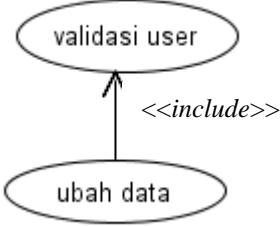
Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

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> 	<p>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</p>  <pre>graph TD; A([validasi username]) -- <<extend>> B([validasi user]); C([validasi sidik jari]) -- <<extend>> B;</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>

Lanjutan Tabel 2.1 Simbol-simbol Use case Diagram

<p>5.</p>	<p>Generalisasi <i>generalization</i></p> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p>6.</p>	<p>menggunakan / include / uses</p>  <p><<include>></p>  <p><<uses>></p>	<p>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</p> <p>ada dua sudut pandang yang cukup besar mengenai include 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, misalnya pada kasus berikut:

Lanjutan Tabel 2.1 Simbol-simbol Use case Diagram

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

Sumber: Sukamto dan Shalahuddin (2016:156)

Ada dua hal utama pada *use case* yaitu:

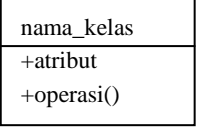


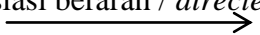
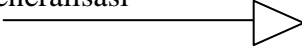
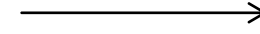
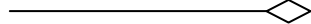
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang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

2.2.3.2 Class Diagram

“*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.” (Sukamto dan Shalahuddin (2016:141)),

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

Tabel 2.2 Simbol-simbol *Class* Diagram

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / 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 antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)



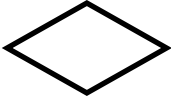


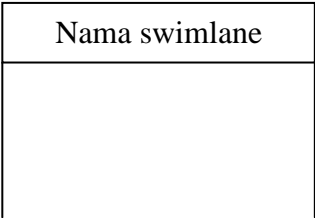
(Sumber: Sukamto dan Shalahuddin (2016:144-147))

2.2.3.3 Activity Diagram

Sukamto dan Shalahuddin (2016:161), “*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 2.3. 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 dimana 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.	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Sukamto dan Shalahuddin (2016:162-163))

2.2.3.4 Sequence Diagram

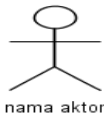
“Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Sukamto dan Shalahuddin, 2016:165).

“Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu” (Safaat, 2015:33-34).


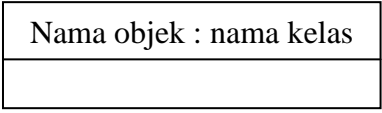
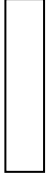
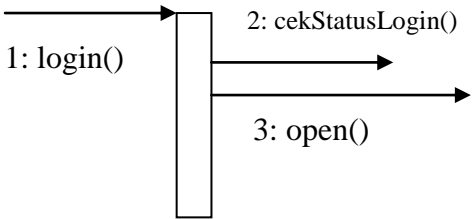
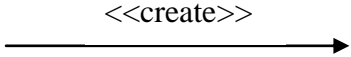
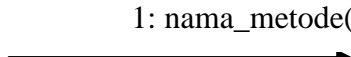
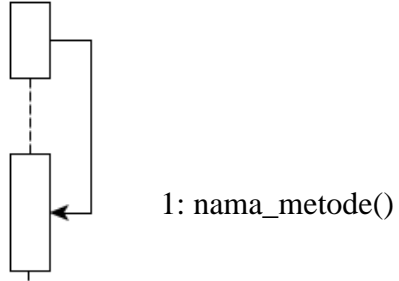
Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada Sequence Diagram :

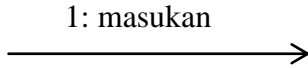
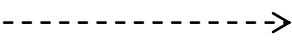
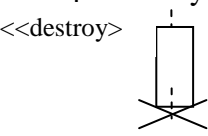
Tabel 2.4 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi		
1.	Actor  nama aktor atau <table border="1" data-bbox="478 1814 678 1915"><tr><td>Nama aktor</td></tr><tr><td> </td></tr></table> tanpa waktu aktif	Nama aktor		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
Nama aktor				

Lanjutan Tabel 2.4 Simbol-simbol pada Sequence Diagram

2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe create</p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe call</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</p>

Lanjutan Tabel 2.4 Simbol-simbol pada Sequence Diagram

		memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe send 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 	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 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

(Sumber: Rosa dan Shalahuddin (2013:165-167))

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Menurut Juansyah (2015:2), dalam kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Sedangkan pendapat Koyuko dkk (dikutip Oki 2016), “Aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya. *suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket



mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut dan juga aplikasi- aplikasi tersebut mempunyai bisa berinteraksi antar aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.”

Dari definisi diatas penulis menyimpulkan bahwa aplikasi adalah perangkat lunak atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu.

2.3.2 Toko Tani Mandiri

Toko Tani Mandiri Muara Enim merupakan sebuah Toko yang bergerak dalam bidang penjualan pupuk dan obat-obat pertanian. Toko Tani ini Berlokasi di Jl. Jendral Sudirman, Muara Enim, Kecamatan Muara Enim, Kabupaten Muara Enim, Sumatera Selatan 31311.

2.3.3 Pengertian Website

Abdulloh (2018:1), “*Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia”. Sedangkan menurut Prasetyo dan Susanti (2016:2) menyatakan bahwa, “*world wide web* (biasa disingkat WWW) atau web merupakan salah satu aplikasi internet yang paling populer.”

Jadi, dapat disimpulkan *website* adalah aplikasi internet yang berisi sekumpulan halaman berbentuk data teks, gambar, video, audio, maupun animasi.



2.3.4 Aplikasi Persediaan Dan Penjualan toko tani mandiri Berbasis Website

Aplikasi persediaan dan penjualan pada Toko Tani Mandiri adalah aplikasi yang dibuat dengan berbagai fitur yang bertujuan untuk memudahkan Toko Tani Mandiri dalam memproses transaksi, menginput serta menyimpan data pelanggan maupun data karyawan.

2.4 Teori Program

2.4.1 Pengertian MySQL

Kadir (dikutip Pranata dkk. 2015:25), “*MySQL* merupakan *software* yang berbasis *structure query language* (SQL) tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *MySQL* adalah sebuah implementasi dari sistem manajemen *database* relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.” Sedangkan menurut Nugroho (2014:31), “*MySQL* atau program aplikasi *database*, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi *text* dan juga angka.”

Dapat disimpulkan bahwa *MySQL* merupakan *software* DBMS yang bersifat *Open Source* dan digunakan untuk menyimpan data atau server *database* yang mendukung Bahasa *database* pencarian SQL.

2.4.2 Pengertian Xampp

Pratama (2014:440), “XAMPP adalah aplikasi web server bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi linux maupun sistem operasi windows”.

Santosa & Radna Nurmalina (2016:86-87) menyatakan bahwa, “Dengan menggunakan XAMPP tidak dibingungkan dengan penginstalan program-program lain, karena semua kebutuhan telah tersedia oleh XAMPP. Yang terdapat



pada XAMPP diantaranya: Apache, MySQL, PHP, FileZilla FTP Server, PhpmyAdmin dll”.

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa XAMPP yaitu aplikasi web server dengan kebutuhan akan program-program telah terinstal.

2.4.3 Pengertian PHP (*Hypertext Preprocessor*)

Menurut Kadir (dikutip Pranata dkk. 2015:26), “PHP merupakan singkatan dari "PHP: *Hypertext Preprocessor*" adalah skrip yang dijalankan di *server*. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Keuntungan PHP, kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi.” Sedangkan menurut Abdulloh (2018:127), *PHP* merupakan kependekan dari *PHP Hypertext preprocessor* yaitu bahasa pemrograman *web* yang dapat disisipkan dalam skrip *HTML* dan bekerja di sisi *server*.” Adapun menurut pendapat Raharjo (2016:38), “PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web”.

2.4.4 Pengertian HTML (*Hypertext Markup Language*)

Madcoms (dikutip Pranata dkk. 2015:26), “HTML (*HyperText Markup Language*) dikenal sebagai bahasa kode berbasis teks untuk membuat sebuah halaman *web*, keberadaannya dikenal dengan adanya ekstensi *.htm atau *.html.”

Adapun pendapat Supriyanto (dikutip Pranata dkk. 2015:26), “HTML merupakan suatu bahasa dari *website* (*www*) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program *browser*.”

Abdulloh (2018:127) menyatakan bahwa, “HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari *website*”.