



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Agung (2017:252), “Komputer adalah sebuah alat elektronik otomatis yang dapat menghitung atau mengolah data secara cermat”.

Menurut Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program. Alat serbaguna ini memegang peranan penting dalam teknologi komunikasi”.

Berdasarkan definisi di atas penulis menyimpulkan bahwa computer adalah sebuah alat elektronik yang dapat mengolah data digital atau mengolah data secara cermat mengikuti perintah atau program.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut Sujatmiko (2012:210), “Perangkat lunak adalah istilah umum untuk data yang di format dan di simpan secara digital, termasuk program komputer, dokumentasi dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.”

Sedangkan menurut Feri dan Asnawati (2015:2), “*Software* adalah perangkat lunak yang berisikan sebuah intruksi yang diperintahkan dan diproses dengan bantuan perangkat keras sehingga tanpa perangkat lunak, perangkat keras tidak bisa dipakai sehingga *software* dan *hardware* tidak bisa dipisahkan.”

Dalam definisi lain, Kadir (2017:2) mengemukakan bahwa 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.

2.1.3 Pengertian Data

Rusmawan (2019:34) mengemukakan bahwa “Data adalah catatan atas kumpulan fakta.”



Sedangkan Sujatmiko (2012:76) mengemukakan bahwa “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.”

Definisi lain dari Sutabri (dalam Rusmawan, 2019:35) mengatakan bahwa “Data adalah kenyataan yang menggambarkan suatu kejadian serta merupakan suatu bentuk yang masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.”

2.1.4 Metode Pengembangan Sistem

Rosa, Shalahudin (2018:26) Metode Pengembangan Sistem atau *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.

Rosa, Shalahudin (2018:28-30) Metode SDLC air terjun (waterfall) sering juga disebut model sekuensial linier (*sequensial linear*) atau alur hidup klasik (*classic life circle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*) :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.



3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2 Teori Khusus

2.2.1 Pengertian *Unified Modeling Language* (UML)

Rosa dan Shalahuddin (2018:133), menjelaskan tentang pengertian *Unified Modeling Language* adalah 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.”

Disini penulis memilih *UML* karena menyediakan serangkaian gambar dan diagram yang sangat baik dan mudah untuk dipahami. Beberapa diagram dapat memfokuskan diri pada ketangguhan dari teori *object-oriented* dan sebagian lagi dapat memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna yang dapat menghasilkan sistem dengan satu kesatuan yang baik.



Gambar 2.1 Tampilan Logo UML

Tabel 2.1 Tipe-Tipe Diagram UML

No	Diagram	Tujuan
1.	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.
2.	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen.
3.	<i>Use case</i>	Diagram ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (dari suatu jenis khusus dari kelas)
4.	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu pada tertentu.
5.	<i>Communication</i>	Sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari obyek yang menerima serta mengirim pesan.
6.	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state),transisi, kejadian serta aktivitas.



Lanjutan Table 2.1 Tipe-Tipe Diagram UML

7.	<i>Activity</i>	Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas lainnya dalam suatu sistem.
8.	<i>Component</i>	Memperlihatkan organisasi serta keberuntungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
9.	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan (<i>run-time</i>)

Sumber: Widodo dan Herlawati (2018:10-12)

2.2.2 Jenis-Jenis Diagram UML

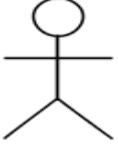
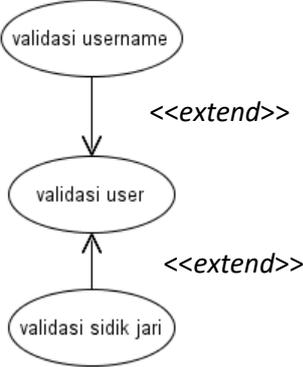
2.2.2.1 Pengertian Use case Diagram

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.” Jadi dapat disimpulkan bahwa *use case* diagram adalah menggambarkan sebuah interaksi yang dilakukan oleh aktor terhadap sistem. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

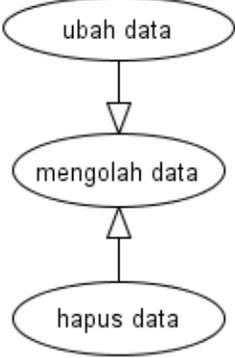
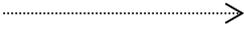
Tabel 2.2 Simbol-simbol Use case Diagram

No	Simbol	Deskripsi
1.		<p>fungsi-fungsionalitas 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></p>

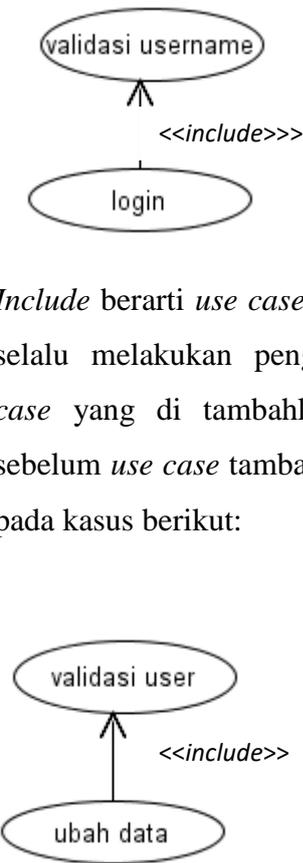
Lanjutan Tabel 2.2 Simbol-simbol *Use Case Diagram*

2.	aktor / <i>actor</i>  nama aktor	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, 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> << <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  biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya

Lanjutan Table 2.2 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
5.	Generalisasi/ <i>generalization</i> 	<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> <div style="text-align: center;">  </div> <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6.	menggunakan/include/uses  <<include>> <<uses> 	<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 2.2 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
6.		 <p>• <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 dijalankan, misal pada kasus berikut:</p> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: 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.

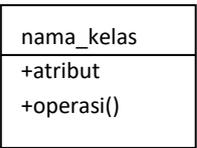
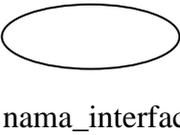


2.2.2.2 Pengertian *Class Diagram*

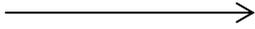
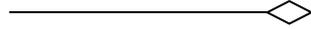
Rosa dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut “*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan 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 	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)

Lanjutan Tabel 2.3 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
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: Rosa dan Shalahuddin (2016:146)

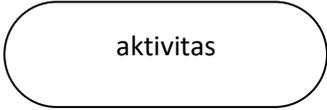
2.2.2.3 Pengertian *Activity Diagram*

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

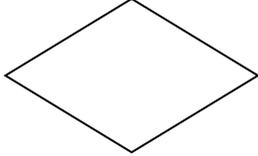
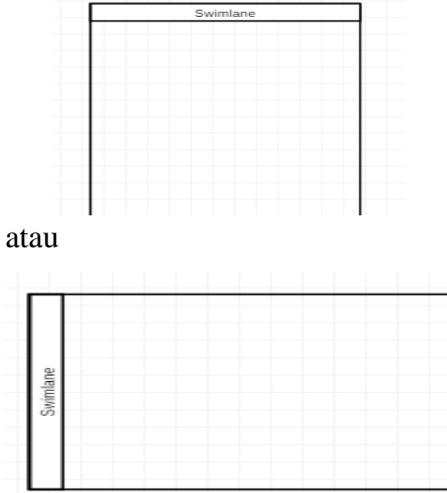
“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat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

Lanjutan Tabel 2.4 Simbol-simbol *Activity Diagram*

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

Sumber: Rosa dan Shalahuddin (2018:162)

2.2.2.4 Pengertian *Sequence Diagram*

“*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 menggambarkan diagram sekuen

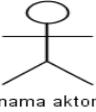
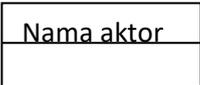
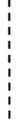
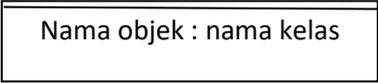


maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Rosa dan Shalahuddin, 2018:165). “Sequence diagram menggambarkan interaksi antar objek di dalam sistem” (Wahyudin dan Munir, 2018:52).

Dapat penulis simpulkan bahwa Sekuen 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.

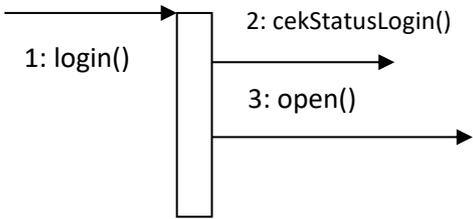
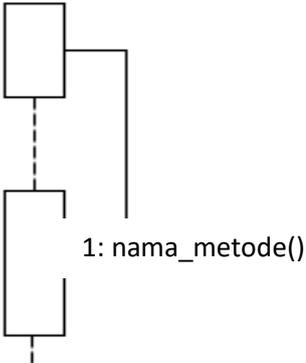
Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.5 Simbol-simbol pada Sequence Diagram

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



Lanjutan Tabel 2.5 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
4.	Waktu aktif 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5.	Pesan tipe create <<create>> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	Pesan tipe call 1: nama metode() 	<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, memanggil</p>



Lanjutan Tabel 2.5 Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
6	Pesan tipe call 1: nama 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 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 <i>create</i> maka ada <i>destroy</i>

Sumber: Rosa dan Shalahuddin (2018:165-167)

2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Menurut Sujatmiko (2012:23), “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Aplikasi berbeda dengan sistem operasi (yang menjalankan komputer), *utiliy* (yang melaksanakan perawatan atau tugas-tugas umum) dan bahasa (yang digunakan untuk ,e, buat program-program computer.” Sedangkan Menurut Agung (2017:33-34) Aplikasi



adalah penerapan, penggunaan, permohonan dan program komputer yang di desain untuk mengerjakan sesuatu.

2.3.2 Pengertian Monitoring

Menurut Prijambodo (2014:9), “Monitoring adalah Kegiatan untuk mengamati perkembangan pelaksanaan program atau proyek”.

Menurut Kemendikbud (dalam Waluyo dan Munawar, 2017:20), “Monitoring adalah upaya pengumpulan informasi berkelanjutan yang di tujukan untuk kepentingan tentang indikasi awal kemajuan dan kekurangan pelaksanaan program dalam rangka perbaikan untuk mencapai tujuan program.”

Berdasarkan definisi di atas penulis menyimpulkan bahwa monitoring adalah suatu kegiatan untuk mengamati perkembangan pengumpulan informasi berkelanjutan untuk kepentingan kemajuan dan kekurangan pelaksanaan program dalam rangka perbaikan untuk mencapai tujuan program.

2.3.3 Pengertian Persediaan

Taufik (2013:76), menyatakan bahwa “Persediaan adalah salah satu komponen yang memiliki peran penting dalam suatu aktivitas bisnis.”

Menurut Wahyudiono (2014:52), “Persediaan adalah produk siap jual, tetapi masih belum terjual.”

Berdasarkan definisi diatas penulis menyimpulkan bahwa persediaan adalah salah satu komponen yang memiliki peran penting dalam suatu aktivitas bisnis dan produk siap jual, tetapi masih belum terjual.

2.3.4 Pengertian Implementasi

Menurut Haryanta dan Sujatmiko (2012:88), “Implementasi merupakan penerapan suatu hal dalam keadaan nyata di lapangan.”

Menurut Suharso dan Retnoningsih (2011:178), “Implementasi adalah pelaksanaan dan penerapan.”

Berdasarkan pengertian di atas, disimpulkan bahwa implementasi adalah penerapan dalam keadaan nyata di lapangan.



2.3.5 Pengertian Portal

Menurut Sujatmiko (2012:219), “Portal adalah Pintu gerbang untuk melakukan akses terhadap situs-situs di internet. Portal berfungsi untuk melakukan pencarian topik atau untuk memperoleh berbagai layanan lainnya.”

Menurut Hakim (2013), “Portal merupakan integrasi dari internet dan sistem informasi eksekutif menjadi solusi yang efektif dan efisien sehingga dapat menjabatangani berbagai sistem berbeda.”

Berdasarkan definisi di atas penulis menyimpulkan bahwa portal adalah pintu gerbang untuk melakukan akses terhadap situs-situs di internet dan sistem informasi eksekutif menjadi solusi yang efektif dan efisien dalam melakukan pencarian topik.

2.3.6 Pengertian Berita

Menurut Fajar (2010:2), “Berita adalah laporan tercepat dari suatu peristiwa dan dari suatu kejadian yang faktual, penting dan menarik bagi sebagian besar pembaca serta menyangkut kepentingan mereka.”

Menurut Wirasutisna (dalam Fajar, 1995:13), “Berita ialah hal-hal yang luar biasa atau istimewa.”

Berdasarkan definisi di atas penulis menyimpulkan bahwa berita adalah laporan suatu peristiwa dari suatu kejadian yang faktual, penting, dan luar biasa.

2.3.7 Pengertian Terpadu

Menurut Kamus Besar Bahasa Indonesia, “Terpadu adalah sudah dipadu (disatukan,dilebur menjadi satu,dan sebagainya).”

Berdasarkan definisi di atas penulis menyimpulkan bahwa terpadu adalah suatu informasi yang telah disatukan.

2.3.8 Pengertian Koperasi

Menurut Feryanto (2018:166) “Koperasi adalah Perserikatan yang bertujuan memenuhi keperluan para anggotanya dengan cara menjual barang keperluan sehari-hari dengan harga murah (tidak bermaksud mencari untung).”



Menurut Agung (2017:257-258) “Koperasi adalah perkumpulan yang bertujuan memenuhi kesejahteraan para anggotanya (biasanya dengan membuat *took* atau semacamnya dengan harga yang murah, tidak ambil untung).”

2.3.9 Pengertian Aplikasi Monitoring Persediaan Barang dan Implementasi Portal Berita Terpadu pada Swalayan Koperasi Wanita Petra Pertamina Plaju

Aplikasi Monitoring Persediaan Barang dan Implementasi Portal Berita Terpadu pada Swalayan Koperasi Wanita Petra Pertamina Plaju adalah suatu aplikasi yang digunakan untuk memantau perkembangan persediaan barang yang dimana persediaan barang tersebut terdiri dari stok barang yang tersisa, supplier, permintaan barang, perhitungan barang masuk dan barang keluar serta menerapkan suatu portal berita agar masyarakat bisa memperoleh berbagai layanan dalam mengakses berita-berita atau informasi terkini.

2.4 Teori Program

2.4.1 Pengertian HTML

Menurut Sujatmiko (2012:128) “HTML (*Hypertext Markup Language*) adalah bahasa computer yang digunakan untuk membuat sebuah halaman web page atau program yang digunakan untuk menulis (membuat) halaman web di internet”.

Menurut Kadir dan Triwahyuni (2013:321) “HTML adalah bahasa markah yang digunakan untuk menyusun halaman web. Kodanya berupa kumpulan tag-tag, dengan setiap ditandai dengan < dan diakhiri dengan >. Tag bisa mewakili elemen di halaman web, misalnya menyatakan elemen gambar serta pasangan <a> dan menyatakan *hyperlink*”.

Menurut Jubilee Enterprise (2015:135) “HTML adalah script pemrograman yang mengatur bagaimana kita menyajikan informasi di dunia internet dan bagaimana informasi itu membawa kita dari satu tempat ke tempat lain”.



2.4.2 Pengertian CSS

Menurut Sujatmiko (2012:72) “CSS atau *Cascading Style Sheet* adalah kumpulan perintah yang dibentuk dari berbagai sumber yang disusun menurut urutan tertentu sehingga mampu mengatasi konflik gaya/style”.

Menurut Rahmad dan Purnama (dalam Pradiatiningtyas dan Suparwanto, 2017:3), “CSS adalah salah satu bahasa yang digunakan untuk mengatur tampilan dalam *web* dan berfungsi untuk memisahkan antara tampilan *desain* dan konten”.

2.4.3 Pengertian PHP

Menurut Sujatmiko (2012:213) “PHP (*Personal Home Page*) adalah bahasa pemrograman yang bisa kita gunakan untuk membuat aplikasi web.”.

Sedangkan Handayani dan Purnama (dalam Pradiatiningtyas dan Suparwanto, 2017:3) berpendapat bahwa PHP adalah bahasa *scripting language* yang dirancang khusus untuk pembuatan web, kaya akan fitur yang membuat perancang web dan pemrograman lebih mudah.

2.4.4 Pengertian XAMPP

Menurut Madcom (dalam Ayu dan Permatasari, 2018:20) “Xampp adalah sebuah paket kumpulan software yang terdiri dari *Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla*, dan lain.”

Sedangkan menurut Riyanto (dalam Pradiatiningtyas dan Suparwanto, 2017:4) “Xampp merupakan paket PHP dan MySQL berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP”.

2.4.5 Pengertian MySQL

Menurut Rusmawan (2019:97) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Databas Management System*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia”.

Sedangkan menurut Kadir dan Triwahyuni (2013:355) “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional.”



2.4.6 Pengertian PHP MyAdmin

Rahman (2018:58) “PhpMyadmin adalah fitur untuk membuat *database*, di mana dalam *website* dinamis seperti CMS Joomla membutuhkan php”. Sedangkan Winoto (dalam Pradiatiningtyas dan Suparwanto, 2017:4) mendefinisikan bahwa PHP MyAdmin adalah suatu program *open source* yang berbasis web yang dibuat menggunakan aplikasi PHP.

Berdasarkan pengertian di atas maka penulis menyimpulkan bahwa PHPMyadmin merupakan *software* yang bertujuan untuk memudahkan dalam mengelola *database*.

2.4.7 Pengertian Basis Data (*Database*)

Menurut Kadir dan Triwahyuni (2013:325) “Basis Data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi”.

Sedangkan menurut Bambang Hariyanto (dalam Rusmawan, 2019:39) “Basis data adalah kumpulan data (*elementer*) yang secara logic berkaitan dalam merepresentasikan fenomena atau fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu”.

2.4.8 Sekilas Tentang Sublime Text

Putra, et.al (dalam Pradiatiningtyas dan Suparwanto, 2017:4) mendefinisikan bahwa sublime text adalah text editor yang elegen, kaya fitur, *cross platform*, mudah dan *simple* yang cukup terkenal dikalangan *developer* (pengembang) dan *desainer*, yang digunakan sebagai editor dari bahasa pemrograman PHP dalam melakukan pengelolaan konten di dalam aplikasi *server*.