



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program.”

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

Dapat disimpulkan bahwa komputer adalah alat bantu dalam pemrosesan data secara elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan manusia.

##### 2.1.2 Pengertian Perangkat Lunak (Software)

Asnawati dan Feri (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.”

Rosa 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 beberapa defisini Perangkat Lunak diatas dapat disimpulkan bahwa perangkat lunak adalah kumpulan beberapa intruksi yang diperintahkan dan diproses dengan bantuan mesin komputer yang terhubung sesuai dengan kebutuhan.

##### 2.1.3 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* 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 berbagai serangkaian dari gambar dan diagram yang sangat baik dan mudah untuk dipahami. Beberapa diagram memfokuskan diri pada ketangguhan dari teori teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi dari *UML* itu sendiri yang semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna agar dapat menghasilkan sesuatu yang mudah untuk dipahami bersama.



**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 (suatu jenis khusus dari kelas)
4.	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan berdasarkan use case dalam suatu waktu tertentu.
5.	<i>Communication</i>	Sebagai pengganti diagram kolaborasi terhadap 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.
7.	<i>Acitivity</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)



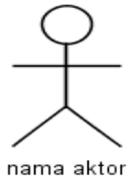
## 2.2.2 Jenis-Jenis Diagram UML

### 2.2.2.1 Pengertian *Use case Diagram*

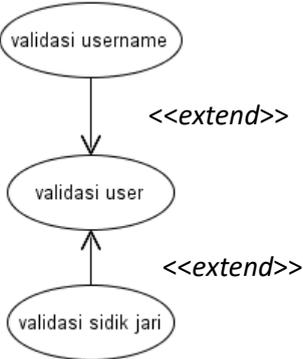
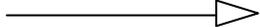
Rosa dan Shalahuddin (2018:155), menjelaskan tentang *use case diagram* sebagai berikut “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.”

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:

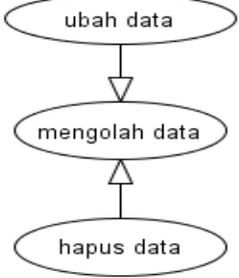
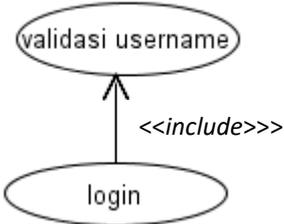
**Tabel 2.2** Simbol-Simbol *Use case Diagram*

No	Simbol	Deskripsi
1.		<p>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>
2.		<p>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 actor.</p>
3.		<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.</p>

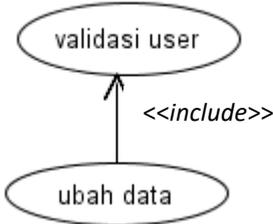
Lanjutan Tabel 2.2 Simbol-Symbol *Use case Diagram*

No	Simbol	Deskripsi
4.	ekstensi / <i>extend</i> <<extend>> 	<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>  <p>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> 	<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>

Lanjutan Tabel 2.2 Simbol-Symbol *Use case Diagram*

No	Simbol	Deskripsi
		 <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
6.	<p>menggunakan/ include / uses</p> <p>.....&gt;</p> <p>&lt;&lt;include&gt;&gt;</p> <p>&lt;&lt;uses&gt;</p> <p>————&gt;</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"> <li>• <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:</li> </ul> 

Lanjutan Tabel 2.2 Simbol-Simbol *Use case Diagram*

No	Simbol	Deskripsi
		<ul style="list-style-type: none"> <li>• <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:</li> </ul>  <pre> graph BT     UC1(ubah data) -- "&lt;&lt;include&gt;&gt;" --&gt; UC2(validasi user)   </pre> <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 :

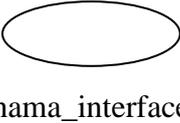
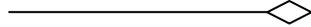
“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar



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)
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 (2018:146)



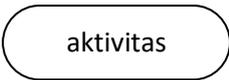
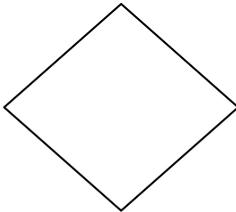
### 2.2.2.3 Pengertian *Activity Diagram*

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

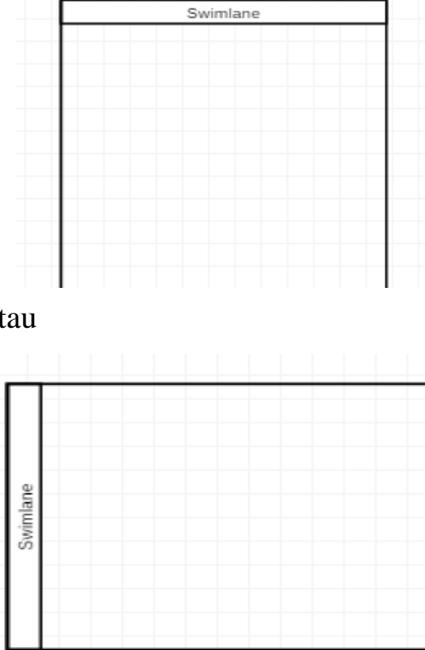
“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat 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 aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

Lanjutan Tabel 2.4 Simbol-Simbol *Activity Diagram*

No	Simbol	Deskripsi
6.	Swimlane  atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa dan Shalahuddin (2018:162)

#### 2.2.2.4 Pengertian *Sequence Diagram*

Rosa dan Shalahuddin (2013:165) Mengemukakan Bahwa: Diagram *sequence* 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.

Sedangkan Menurut (Wahyudin dan Munir, 2018:52) bahwa “Sequence diagram menggambarkan interaksi antar objek di dalam sistem”

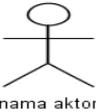
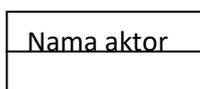
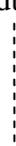
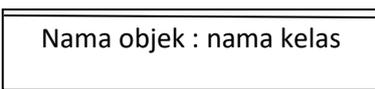
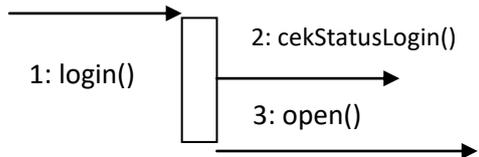
Dapat penulis simpulkan bahwa Sekuen diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian



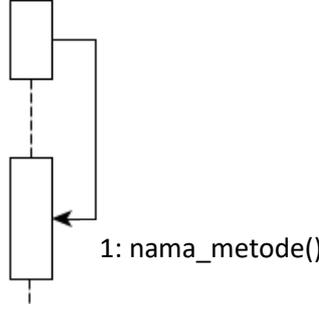
langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada Sequence Diagram :

**Tabel 2.5** Simbol-Simbol *Sequence Diagram*

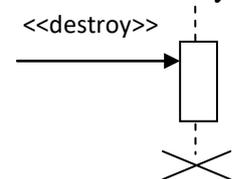
No	Simbol	Deskripsi
1.	<p>Actor</p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan 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>Nama objek : nama kelas</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>1: login()</p> <p>2: cekStatusLogin()</p> <p>3: open()</p> <p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
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 menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian



Lanjutan Tabel 2.5 Simbol-Simbol Sequence Diagram

No	Simbol	Deskripsi
9.	Pesan tipe 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 Teori Judul

### 2.3.1 Pengertian Sistem

Menurut Rusmawan (2019:29), “Sistem adalah suatu jaringan atau komponen-komponen yang membentuk satu kesatuan yang terkumpul untuk menyelesaikan suatu tujuan tertentu”.

Sedangkan menurut Maniah dan Dini (2017:1), “Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan/sasaran tertentu yang sama”.

Riyanto, Ekaputra dan Indelarko (2019:21), “Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling tergantung dan terintegrasi dalam kesatuan variabel atau komponen”.

### 2.3.2 Pengertian Informasi

Menurut Sujatmiko (2012:135), “Informasi adalah serangkaian simbol yang jika disusun menurut aturan tertentu dapat menyampaikan suatu pesan”.

Rusmawan (2019:32), “Informasi adalah sekumpulan data yang diolah sehingga berguna bagi seseorang atau organisasi yang membutuhkan informasi tersebut”.



M.Thoha Miyanto (dalam Rusmawan, 2019:32) berpendapat bahwa Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa informasi adalah serangkaian simbol ataupun sekumpulan data yang diolah sehingga berguna bagi seseorang yang menerimanya.

### **2.3.3 Pengertian Sistem Informasi**

Sutabri (2005:42) mengemukakan bahwa “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manterial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Menurut Sujatmiko (2012:251), “Sistem Informasi adalah sistem pengiriman data melalui fasilitas telekomunikasi dari satu lokasi ke pusat pengolahan data tetapi data yang dikirimkan tidak langsung diproses oleh CPU”.

Menurut O’Brien (dalam Rusmawan, 2019:33), “Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah dan menyebarkan informasi dalam organisasi”.

### **2.3.4 Pengertian Manajemen**

Feryanto (2018:183) mengemukakan bahwa “Manajemen adalah ilmu dan seni dalam perencanaan pengorganisasian pelaksanaan dan pengawasan, penyusunan sumber daya manusia untuk mencapai tujuan yang telah ditetapkan.”

Stoner (dalam Handoko, (2009:8), “Manajemen adalah proses perencanaan, pengorganisasian, pengarahan dan pengawasan usaha-usaha para anggota organisasi dan penggunaan sumber daya-sumber daya organisasi lainnya agar mencapai tujuan organisasi yang telah ditetapkan.”.

Jadi manajemen sendiri dapat penulis simpulkan sebagai ilmu dalam proses perencanaan dan pengawasan untuk mencapai tujuan yang telah ditetapkan.



### **2.3.5 Pengertian Klinik**

Menurut Agung (2017:246), “Klinik adalah bagian rumah sakit atau lembaga kesehatan tempat orang berobat dan memperoleh nasihat medis serta tempat mahasiswa kedokteran mengadakan pengamatan terhadap kasus-kasus penyakit yang diderita pasien.”

Menurut (Permenkes RI No.28,2011), “Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan/atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis.”

Jadi dapat penulis simpulkan bahwa klinik adalah bagian dari rumah sakit yang dapat memberikan fasilitas kesehatan dan melakukan proses pengobatan terhadap masyarakat dan memperoleh nasihat pendapat medis mengenai kondisi yang dialami dan dipimpin oleh seorang tenaga medis.

### **2.3.6 Pengertian Sistem Informasi Manajemen Klinik (SIMKLIN) Pada Klinik Ummat Talang Kelapa Berbasis Website**

Sistem Informasi Manajemen Klinik (SIMKLIN) Pada Klinik Ummat Talang Kelapa Berbasis Website adalah suatu program yang dibangun pada Klinik Ummat untuk mempermudah setiap unit klinik dalam proses pengolahan data yang berhubungan dengan pasien baik itu pendataan pasien maupun data medis pasien serta pengolahan data stok obat maupun biaya pemeriksaan yang dapat dilakukan secara cepat untuk mendukung dan meningkatkan pelayanan yang diberikan oleh klinik.

## **2.4 Teori Program**

### **2.4.1 Pengertian Data**

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

Sedangkan menurut 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.”

#### **2.4.2 Pengertian Basis Data**

Rosa dan Shalahuddin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Sujatmiko (2012:76), “Basis data adalah Representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang tidak perlu untuk memenuhi berbagai kebutuhan.”

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

Dapat disimpulkan bahwa Basis data adalah sistem komputerisasi yang berfungsi sebagai media penyimpanan data dan informasi saat dibutuhkan tanpa adanya pengulangan data yang tersimpan.

#### **2.4.3 Pengertian Microsoft Sql Server**

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

Sedangkan definisi SQL Server menurut Wahana Komputer (2012:2), “*SQL Server* merupakan software RDMS kelas *enterprise* yang cukup banyak digunakan oleh dunia korporat. Dengan ini menggunakan *SQL Server*, *user* dapat menyimpan banyak data dan mengimplementasikannya untuk kepentingan bisnis dan perusahaan.”

Berbagai macam komponen yang terdapat didalam SQL Server menurut Putra (2020:6), “Berikut ini penjelasan dari komponen SQL:



### 1. *Data Definitions Language (DDL)*

Merupakan perintah dasar dari bahasa pemrograman SQL yang diperuntukkan untuk membuat sebuah struktur database dalam pembuatan tabel, beberapa perintah yang termasuk dalam DDL yaitu:

- a. *Create* : untuk membuat atau menciptakan objek basis data.
- b. *Alter* : untuk memodifikasi atau mengubah objek basis data.
- c. *Drop* : untuk menghapus objek basisdata.
- d. *Show* : untuk menampilkan database.
- e. *Rename* : untuk merubah nama dari suatu tabel atau kolom

### 2. *Data Manipulations Language (DML)*

Merupakan perintah dasar dari bahasa pemrograman SQL:

- a. *Select* : digunakan untuk mengambil data dari basis data.
- b. *Delete* : digunakan untuk menghapus data pada basis data.
- c. *Insert* : digunakan untuk menambahkan data kedalam tabel.
- d. *Update* : digunakan untuk memodifikasi data pada basis data.

### 3. *Data Control Language (DCL)*

Adalah perintah dari bahasa pemrograman SQL yang digunakan untuk mengatur hak akses user terhadap database maupun terhadap tabel dan field yang ada.

- a. *Grant* : digunakan untuk memberikan akses kepada *user* tertentu ke basis data.
- b. *Revoke* : digunakan untuk mencabut hak akses dari *user*.

#### 2.4.4 Pengertian HTML

HTML (Hypertext Markup Language) Menurut Surya dan Jannah (2020:1), “HTML (Hypertext Markup Language) merupakan sebuah Bahasa *markup* atau penanda berbasis text atau disebut juga *formatting language* (bahasa untuk memformat).”

Menurut Larry (2012:4), “Pada awalnya HTML dikembangkan sebagai subset SGML (Standard Generalized Markup Language) karena HTML didedikasikan untuk ditransmisikan melalui media internet, maka HTML relative lebih sederhana daripada SGML yang lebih ditekankan pada format dokumen”



#### 2.4.5 Pengertian CSS

CSS (Cascading Style Sheet) Menurut Larry (2012:5) menerangkan bahwa “CSS adalah suatu Bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML”.

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”.

Jadi dapat disimpulkan bahwa CSS adalah suatu Bahasa yang digunakan untuk mengatur tampilan dari suatu halaman web untuk memisahkan tampilan desain dan konten.

#### 2.4.6 Pengertian XAMPP

Menurut Kristanto(2018:29), “XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”.

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.”

#### 2.4.7 Pengertian PHP

Jannah dkk (2019:1), “Hypertext Preprocessor (PHP) merupakan bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. PHP disebut bahasa pemrograman *server-side* karena diproses pada computer server.”

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

Dapat disimpulkan bahwa PHP adalah Bahasa pemrograman yang di desain untuk membuat ataupun mengembangkan sebuah website yang banyak fitur serta pemrograman yang lebih mudah.