



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

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

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

##### 2.1.2 Pengertian Perangkat Lunak

Kadir (2017:2) menegaskan, “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Sukanto dan Shalahuddin (2018:2) menegaskan, “Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).

##### 2.1.3 Pengertian Data

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

Fathansyah (2018:2) menegaskan, “Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya”.

##### 2.1.4 Pengertian Basis Data (*Database*)

Indrajani (2018:2) menegaskan, “Basis Data adalah suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk



memenuhi informasi yang dibutuhkan oleh suatu organisasi”.

Menurut A.S dan Shalahudin (2018:28) “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat”.

### 2.1.5 Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2018:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. “Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup terturut mulai dari analisis, desain, pengkodean, pengujian dan pemeliharaan.

#### a. Analisis kebutuhan perangkat lunak

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

#### b. 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.

#### c. 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.

#### d. 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.

## **2.2 Teori Judul**

### **2.2.1 Pengertian Aplikasi**

Chan (2017:4) menegaskan, “Aplikasi adalah koleksi *window* dan objek-objek yang menyediakan fungsi untuk aktivitas *user*, seperti pemasukan data, proses dan pelaporan”.

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

### **2.2.2 Pengertian Inventaris**

Menurut Elizabeth Lucky Maretha dalam Buku Saku Laporan Keuangan Pokdarwis (2020:44), inventaris adalah daftar barang yang dimiliki suatu perusahaan atau kelompok tertentu. Dalam inventaris dikenal juga istilah inventaris kantor dan inventaris barang.

### **2.2.3 Pengertian Website**

Hidayatullah (2017:1) menegaskan, “*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”.

*Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi visual ataupun berupa teks yang bersifat statis atau dinamis dan membentuk suatu rangkaian halaman yang saling terkait (Hariyanto dikutip Destiningrum dan Adrian, 2017:32).

## **2.3 Teori Khusus**

### **2.3.1 Pengertian UML (*Unified Modeling Language*)**

Sugiarti (2018:100) menjelaskan bahwa “*Unified Modeling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem peranti lunak.”

---



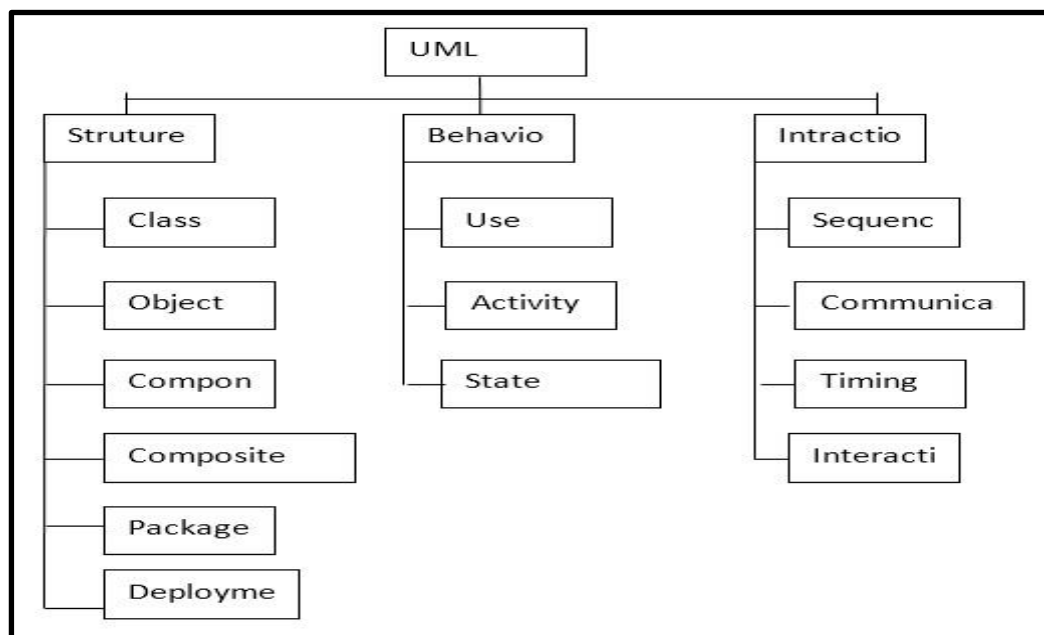
Munawar (2018:49) menjelaskan bahwa “UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek.”

Inillah beberapa tujuan atau fungsi dari penggunaan UML:

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktik-praktik terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemrograman visual yang ekspresif untuk mengembangkan sistem, dan untuk saling tukar model secara mudah.

### 2.3.2 Macam – macam diagram *Unified Modeling Language*

Pada UML terdapat 13 buah diagram yang selanjutnya dikelompokkan menjadi 3 kategori. Pembagian kategori dan macam – macam diagram tersebut dapat dilihat pada tabel dibawah ini :



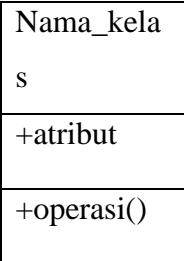



**Tabel 2.1** Macam – Macam Diagram UML





### 2.3.3 Pengertian *Class Diagram*

Menurut Sugiarti (2018:119) menjelaskan bahwa “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian class-class yang akan dibuat untuk membangun sistem.”


**Tabel 2.2** Simbol-simbol dalam *class diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka/ <i>interface</i>  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity

**Tabel 2.3** Lanjutan Tabel Simbol-simbol dalam *class diagram*

Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas



Agregasi/aggregation 	Relasi antarkelass dengan makna semua-bagian (whole-part)
---	---

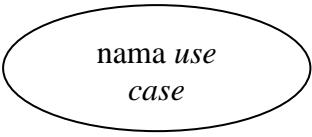
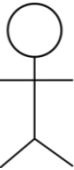
### 2.3.4 Pengertian Use Case Diagram

Munawar (2018:89) menjelaskan bahwa “Use case adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai.”


Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

- a. Aktor merupakan 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.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.


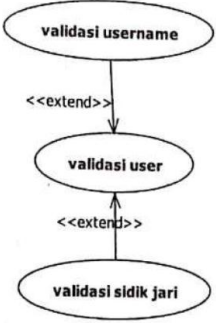
**Tabel 2.4** Simbol-simbol dalam *Use case*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal <i>frase</i> nama <i>use case</i> .
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 <i>frase</i> nama aktor.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.


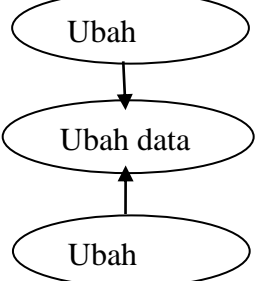
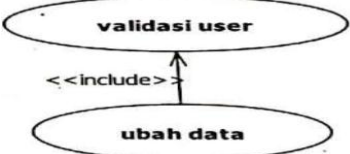
Tabel 2.5 Lanjutan Tabel Simbol-simbol dalam *Use case*

<p>Ekstensi/<i>extend</i></p> <p><code>&lt;&lt;extend&gt;&gt;</code></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun 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, misal;</p>  <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</p>
--	---



	menjadi induknya.
--	-------------------

**Tabel 2.6** Lanjutan Tabel Simbol-simbol dalam *Use case*

<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada use case yang menjadi generalisasinya (umum</p>
	<p>ditambahkan selalu dijalankan sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut:</p>  <p>Kedua interpretasi di atas dapat dianut salah satu keduanya tergantung pada pertimbangan dan interpretasi yang.</p>

### 2.3.5 Pengertian Activity Diagram

Menurut Sugiarti (2018:134) menjelaskan bahwa “Activity Diagram menggambarkan berbagai aliran activity dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi, dan








bagaimana mereka berakhir. Activity Diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.”



Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak

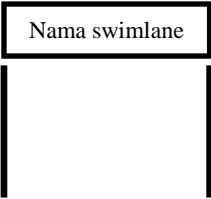
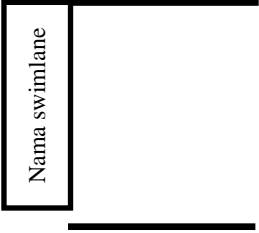
**Tabel 2.7** Simbol-simbol dalam *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

**Tabel 2.8** Lanjutan Tabel Simbol-simbol dalam *Activity Diagram*

Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir



<p>Swimlane</p>  <p>Atau</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
--	--

### 2.3.6 Pengertian Sequence Diagram

Menurut Sugiarti (2018:130) mengatakan bahwa “Diagram sekuens menggambarkan behavior objek pada Usecase dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antarobjek.”

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *Usecase* yang memiliki proses sendiri atau yang penting semua *Usecase* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuens. Oleh karena itu, semakin banyak *Usecase* yang didefinisikan, maka diagram sekuens yang harus dibuat juga semakin banyak.

**Tabel 2.5** Simbol Diagram *Sequence*



No	Simbol	Nama	Keterangan
1.		Garis hidup <i>/ lifeline</i>	Menyatakan kehidupan suatu objek
2.	 Atau <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">Nama aktor</div>	Aktor	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

Tabel 2.9 Lanjutan Tabel Simbol Diagram *Sequence*

3.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
4.	<p>&lt;&lt;create&gt;&gt;</p>	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
5.	<p>1 : nama_metode()</p>	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode
	<p>1 : masukan</p>	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan



6.			data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
7.		Pesan tipe 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

## 2.4. Teori Program

### 2.4.1 Pengertian Sublime Text

Syifani dan Ardiansyah (2018:25) menegaskan, “*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi *Phyton API*”.

Rerung (2018:6), “*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi *API* . Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*. Aplikasi ini sangatlah fleksibel dan powerfull.”

### 2.4.2 Pengertian HTML

Menurut Indrajani (2018:55) merupakan bahasa standar yang digunakan untuk mendesain seluruh halaman web, di mana kita dapat mengontrol tampilan web page dan content-nya, mempublikan dokumen secara online, membuat form online untuk pendaftaran atau transaksi, dan menambahkan objek-objek, seperti image, audio, video, dan java applet ke dalam dokumen *HTML*.

Menurut Hidayatullah dan Jauhari (2020:15), “*HTML* adalah Bahasa standard yang digunakan untuk menampilkan web”.

### 2.4.3 Java Script

Menurut Faisal dan Abadi (2020:140), “*Javascript* adalah bahasa pemrograman yang awalnya dikembangkan oleh *Netscape Communication*. Dan sekarang menjadi standar pada hampir seluruh *web browser*”. Sedangkan,



menurut Abdulloh (2018:193), “*Javascript* adalah bahasa pemrograman *web* yang pemrosesanya dilakukan di sisi *client*”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian *Java Script* adalah bahasa pemrograman web yang pemrosesanya dilakukan di sisi *client* untuk memberikan efek dinamis, interaktif dan bersifat independen, yang dieksekusi di *browser*.

#### **2.4.4 Pengertian CSS**

Abdullah (2018:45) menegaskan, “CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan.”

Sulistiono (2018:2) menegaskan, “CSS (*cascading style sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam”.

#### **2.4.5 Pengertian PHP**

Enterprise (2018:1) menegaskan, “PHP merupakan bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif. Dinamis artinya, website tersebut bisa berubah tampilan dan kontennya sesuai kondisi tertentu”

Menurut Supono & Putratama (2018: 3) mengemukakan bahwa “PHP (PHP: hypertext preprocessor) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat server-side yang ditambahkan ke HTML”.

#### **2.4.6 Pengertian MySQL**

Enterprise (2018:2) menegaskan, “MySQL merupakan server yang melayani database. Untuk membuat dan mengolah database, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL”.

Rosa dan Shalahuddin (2018:46), “SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

#### **2.4.7 Pengertian XAMPP**

---



Idcloud (2019) menegaskan, “XAMPP adalah perangkat lunak gratis yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain Apache HTTP Server, *MySQL* database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.”

Poipessy dan Mirna Umasangadji (2018:40) menegaskan, “XAMPP merupakan paket PHP dan *MySQL* berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP.” Dapat disimpulkan XAMPP merupakan paket konfigurasi Apache, PHP dan *MySQL* yang digunakan untuk membantu pengembangan aplikasi.