



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Menurut Krisbiantoro (2018:1) “Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan”.

Sindu dan Paramartha (2019:2) mengatakan, “Komputer merupakan sebuah mesin penghitung elektronik yang cepat dapat menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan di memorinya (*stored program*) dan menghasilkan output informasi”.

Berdasarkan pengertian di atas dapat disimpulkan bahwa pengertian komputer adalah alat elektronik yang dipakai untuk mengolah dan memproses data sehingga menghasilkan output berupa informasi.

##### 2.1.2 Pengertian Perangkat Lunak

Sukamto dan Shalahuddin (2018:2) mengatakan, “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*)”.

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

Berdasarkan pengertian di atas dapat disimpulkan bahwa pengertian perangkat lunak adalah program komputer yang dapat melaksanakan tugas sesuai kehendak pemakai.

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

“Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.” (Sukamto dan Shalahuddin, 2018:43).



Menurut Hidayatullah dan Kawistara (2017:142), “Basis Data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Berdasarkan pengertian di atas dapat disimpulkan bahwa pengertian basis data suatu media untuk menyimpan data agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

#### **2.1.4 Metode Pengembangan Aplikasi**

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2018:125), “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 (2018:128-131) 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. Aktivitas 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 Sukamto dan Shalahuddin (2018:133), “UML (*Unified Modelling 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”.

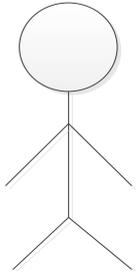
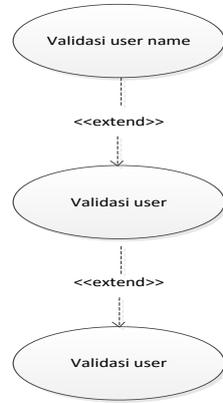
### 2.2.2 *Use Case Diagram*

Sukamto dan Shalahuddin (2018:155) mengatakan, “*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 akan digunakan dalam *use case* adalah sebagai berikut:

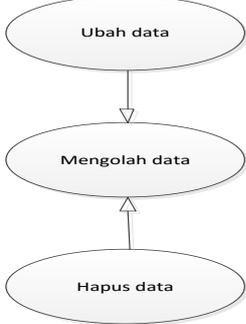
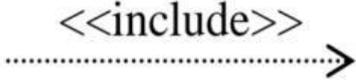
**Tabel 2.1** Simbol-simbol *Use case* Diagram

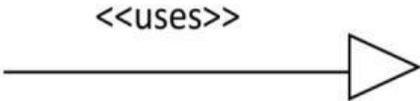
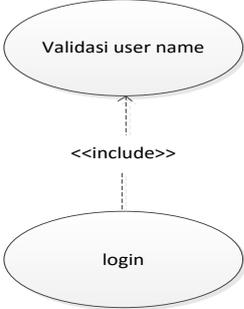
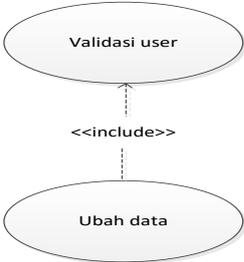
No.	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .



No.	Simbol	Deskripsi
2.	Aktor / <i>actor</i>   nama 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 aktor.
3.	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.
4.	Extensi / <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 objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal    arah panah mengarah pada <i>use case</i> yang



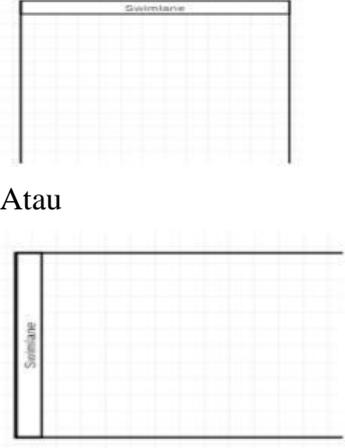
No.	Simbol	Deskripsi
		ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
5.	Generalisasi/ <i>generalization</i>  	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :   misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	Menggunakan / <i>include</i> / <i>uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : - Include berarti <i>use case</i> yang ditambahkan akan selalu di panggil

No.	Simbol	Deskripsi
		<p>saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut :</p>  <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 di jalankan, misal pada kasus berikut:</p>  <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

### 2.2.3 Activity Diagram

Sukanto dan Shalahuddin (2018:161) menyatakan, “*Activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Tabel 2.2 Simbol-simbol *Activity Diagram*

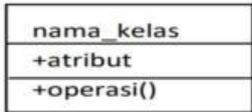
No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas 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 oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  Atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.



### 2.2.4 Class Diagram

Sukamto dan Shalahuddin (2018:141) menyatakan, “*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

**Tabel 2.3** Simbol-simbol *Class Diagram*

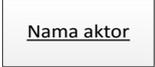
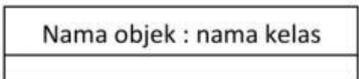
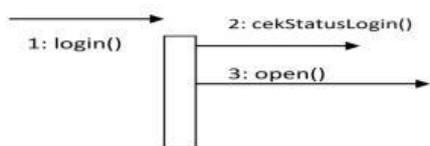
No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) .
6.	Keberuntungan/ <i>depedensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agrgasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ).

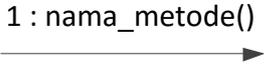
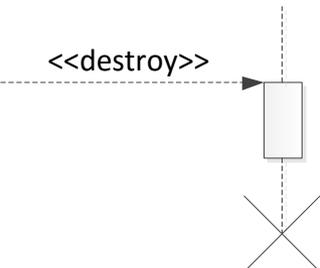


### 2.2.5 Sequence Diagram

Sukamto dan Shalahuddin (2018:165) menyatakan, “*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu daur hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

**Tabel 2.4** Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama actor Atau</p>  <p>Tanpa waktu aktif</p>	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, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cek StatusLogin() dan open()</p>

No.	Simbol	Deskripsi
		dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 	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 <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	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 <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>



## **2.3 Teori Judul**

### **2.3.1 Pengertian Aplikasi**

Menurut Deslianti dan Mutaqqin (2016:2), menyatakan bahwa “Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan”.

Rahman dan Santoso (2015:2), “Aplikasi adalah sekelompok atribut yang terdiri dari beberapa form, report yang disusun sedemikian rupa sehingga dapat mengakses data”.

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa aplikasi adalah penerapan suatu konsep yang terdiri dari beberapa atribut yang disusun sedemikian rupa sehingga dapat mengakses data.

### **2.3.2 Pengertian Pengolahan Data**

Menurut Reksoatmodjo (2018:8), “ Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

Berdasarkan definisi di atas, maka penulis dapat menyimpulkan bahwa pengolahan data adalah waktu yang digunakan untuk menerima data dan memproses menggunakan program tertentu sehingga menjadi informasi yang memiliki kegunaan.

### **2.3.3 Pengertian Keuangan**

Menurut Kamus Besar Bahasa Indonesia, “Keuangan adalah seluk-beluk uang; urusan uang”.

### **2.3.4 Pengertian Web**

Menurut Abdullah (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”.



Prasetyo dan Susanti (2016:2) menyatakan bahwa, “*world wide web* (biasa disingkat WWW) atau *web* merupakan salah satu aplikasi internet yang paling populer”.

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa *website* adalah aplikasi internet yang berisi informasi data digital yang disediakan melalui jalur internet sehingga dapat diakses dan dilihat semua orang di dunia.

### **2.3.5 Pengertian Aplikasi Pengolahan Data Keuangan pada CV. Pati Karya Palembang Berbasis Web**

Dapat penulis simpulkan bahwa aplikasi pengolahan data keuangan pada CV. Pati Karya Palembang berbasis *web* adalah program yang diterapkan pada komputer, yang memiliki fungsi dan kemampuan untuk mengolah data mentah menjadi sebuah informasi yang bermanfaat, terutama dalam melakukan perhitungan biaya produksi suatu barang dan pembuatan laporan laba kotor berdasarkan pesanan.

## **2.4 Teori Program**

### **2.4.1 Pengertian XAMPP**

Menurut Hidayatullah dan Kawistara (2017:125) menyatakan bahwa, XAMPP merupakan aplikasi instalasi PHP, apache, dan MySQL secara praktis agar dapat mengakses web dinamis secara lokal.

Aryanto (2016:4), “Xampp merupakan sebuah aplikasi perangkat lunak pemrograman dan database yang di dalamnya terdapat berbagai macam aplikasi pemrograman seperti : Apache, HTTP, MySQL, database, bahasa pemrograman PHP dan Perl”.

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa XAMPP adalah aplikasi pemrograman dan database yang terdiri dari Apache, HTTP, MySQL, database, bahasa pemrograman PHP dan Perl agar dapat mengakses web dinamis secara lokal.



#### 2.4.2 Pengertian PHP

Subagia (2018:1), “PHP merupakan sebuah bahasa pemrograman yang berjalan dalam web server (server side).”

Madcoms(2016:2), “PHP (*Hypertext preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML”.

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa PHP adalah bahasa pemrograman yang disisipkan ke dalam HTML agar dapat berjalan dalam web server.

#### 2.4.3 Pengertian Sublime Text

Rerung (2018:25), “Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform operating sistem dengan menggunakan teknologi API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah fleksibel dan powerful.”

“Sublime Text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi” (Putratama, 2018:14).

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa sublime text adalah aplikasi editor yang dapat berjalan di beberapa platform untuk membuat atau meng-edit suatu aplikasi.

#### 2.4.4 Pengertian MySQL

Sukanto 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.”

Menurut Hidayatullah dan Kawistara (2017:175) “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi web”.

Berdasarkan kedua definisi di atas, maka penulis dapat menyimpulkan bahwa MySQL adalah pemrograman yang digunakan untuk mengelola data pada RDBMS.