



BAB II TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Atanasoff (dalam Muin, 2020) komputer adalah suatu alat elektronik yang dapat bekerja secara otomatis, terdiri dari komponen-komponen penyusunnya yang dapat saling bekerja sama dengan menggunakan program, sehingga menjadi suatu alat yang dapat digunakan untuk mengolah, mengelola, menyimpan dan membuat berbagai macam data menurut prosedur yang telah dirumuskan.

2.1.2 Pengertian Sistem

Pengertian sistem menurut (Sofyan *et al.*, 2021) adalah jaringan yang saling berhubungan, berkumpul untuk melakukan kegiatan untuk menyelesaikan suatu tujuan tertentu.

2.1.3 Pengertian Perangkat Keras (*Hardware*)

Menurut Mulyani (dalam Fitriadi, 2020), “Perangkat keras komputer (*hardware*) adalah semua bagian fisik komputer, dan dibedakan dengan data yang ada didalamnya atau yang beroperasi didalamnya, dan dibedakan dengan perangkat lunak (*software*) yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya”.

2.1.4 Pengertian Perangkat Lunak (*Software*)

Menurut Mulyani (dalam Fitriadi, 2020), “Perangkat Lunak (*Software*) merupakan sebuah perangkat yang tidak berbentuk secara fisik, namun dapat dioperasikan oleh user atau penggunanya dengan kumpulan program komputer yang digunakan untuk memproses data”.

Berdasarkan beberapa pengertian diatas dapat disimpulkan bahwa perangkat lunak adalah program komputer terasosiasi dengan dokumentasi yang digunakan untuk menjalankan suatu pekerjaan.



2.1.5 Pengertian Internet

Menurut (Arthalita & Prasetyo, 2020) Pengertian internet dapat disimpulkan sebagai seluruh jaringan yang saling terhubung secara fisik dan memiliki kemampuan untuk membaca dan menguraikan protocol komunikasi dengan jumlah pengguna lebih dari 200 negara.

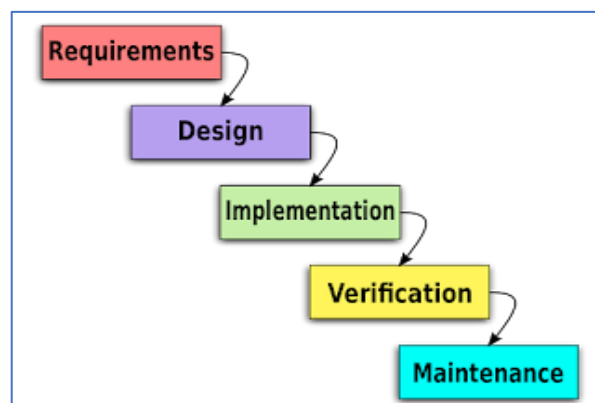
2.1.6 Pengertian Website

Website adalah kumpulan halaman web yang memiliki sebuah domain yang mengandung informasi. Domain website adalah nama unik atau yang tidak duplikat yang dimiliki oleh sebuah institusi sehingga bisa diakses melalui internet. (Sofyan *et al.*, 2021)

2.1.7 Metode Pengembangan Aplikasi

Metode *waterfall* merupakan sebuah metode dalam mengembangkan suatu perangkat lunak yang memungkinkan pembuatan sistem secara sistematis dan terstruktur (algoritma) sesuai dengan siklus dan tahapan yang berlaku. Metode ini disebut *waterfall* karena berfilosofi seperti air terjun, dimana dari tahap satu ke tahap berikutnya adalah berkesinambungan dan berkaitan erat. (Maulana, 2021)

Menurut (Maulana, 2021) secara umum, ada lima tahapan dalam metode *waterfall*, seperti digambar berikut :



Sumber : Maulana (2021)

Gambar 2.1 Tahapan Metode *Waterfall*



1. Requirement (Analisis Kebutuhan)

Requirement adalah proses analisa atau pengumpulan data-data yang berkaitan dengan sistem yang akan dibuat. Pengumpulan data ini bisa dilakukan dengan wawancara, studi literatur, observasi atau penelitian langsung. Dalam fase ini tim analis akan menggali informasi sebanyak-banyaknya dari klien atau user tentang software apa yang mereka inginkan beserta dengan kebutuhan sistem lainnya.

2. Design System (Desain Sistem)

Proses ini akan berfokus pada pembangunan struktur data, arsitektur perangkat lunak, perancangan *interface*, perancangan fungsi internal dan eksternal serta detail dari setiap algoritma prosedural. Tahapan design akan menghasilkan dokumen bernama “*Software Requirement*” yang nantinya menjadi landasan para programmer dalam membuat code-code aplikasi.

3. Implementation (Pengerjaan)

Tahap ini adalah tahapan pembuatan aplikasi oleh para programmer dengan menggunakan kode-kode bahasa pemrograman tertentu. Proses penulisan sinkode (coding) aplikasi mengacu pada dokumen-dokumen yang telah dibuat sebelumnya.

4. Verification (Verifikasi)

Tahapan verifikasi meliputi pengintegrasian sistem dan juga melakukan testing terhadap aplikasi yang telah dibuat. Sistem akan diverifikasi untuk diuji sejauh mana kelayakannya. Dalam tahapan ini semua modul yang dikerjakan oleh programmer berbeda akan digabungkan kemudian diuji apakah telah sesuai dengan spesifikasi yang ditetapkan atau terdapat kesalahan/error dalam sistem sebelum kemudian diperbaiki ulang.

5. Maintenance (Pemeliharaan)

Tahapan ini umumnya meliputi tahapan penginstalasian perangkat lunak dan pengujian aplikasi. *Maintenance* juga adalah bentuk tanggung jawab tim pengembang untuk memastikan aplikasi dapat berjalan lancar setelah diserahkan pada klien dalam periode waktu tertentu.



2.2 Teori Khusus

2.2.1 Pengertian *Unified Modeling Language* (UML)




Unified Modelling Language (UML) merupakan alat perancangan sistem yang berorientasi pada objek. “UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” (Ropianto, 2016)

2.2.2 Jenis-Jenis Diagram *Unified Modeling Language* (UML)



2.2.2.1 Pengertian *Use Case Diagram*

Menurut Sukanto dan Shalahuddin (dalam Ropianto, 2016), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut ini adalah simbol-simbol dari *Use case diagram*:

Tabel 2.1 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
1	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
2	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.
3	Asosiasi/Association 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i>





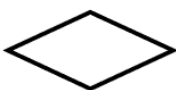

		memiliki interaksi dengan actor.
4	Ekstensi/Extend 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5	Include 	Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

Sumber : Haviluddin (2011)


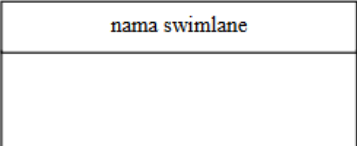
2.2.2.2 Pengertian Activity Diagram

Pengertian Activity Diagram menurut Sukamto dan Shalahuddin (dalam Ropianto, 2016) adalah diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (*software*). Berikut merupakan simbol-simbol dari Activity Diagram :

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/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.




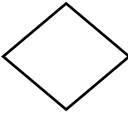
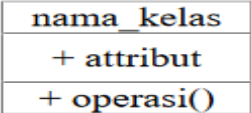
5	Status Akhir 	Status akhir dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Haviluddin (2011)

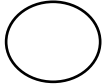

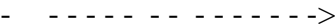

2.2.2.3 Pengertian Class Diagram

Class diagram adalah salah satu diagram yang paling populer bagi software engineering. *Class diagram* mewakili entitas kunci dalam bisnis serta domain teknis. Menurut Munawar (2018) *Class diagram* adalah salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. *Class diagram* menggambarkan atribut, *operation* dan juga *constraint* yang terjadi pada sistem.

Tabel 2.3 Simbol-simbol Class Diagram

No	Simbol	Deskripsi
1	Generalization 	Relasi antarkelas dengan makna generalisasi spesialisasi (umum khusus).
2	Nary Association 	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3	Class 	Kelas pada struktur item.





4	Interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
5	Association 	Relasi antarclass dengan arti umum, asosiasi biasanya juga disertai dengan Multiplicity.
6	Dependency 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7	Aggregation 	Relasi antarkelas dengan makna semua-bagian (whole-part) .

Sumber : Havaluddin (2011)

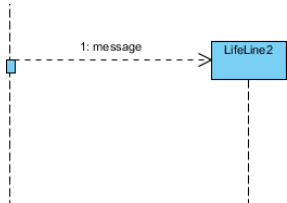
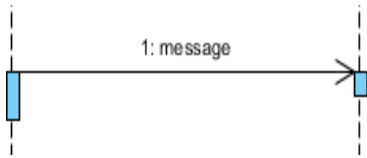
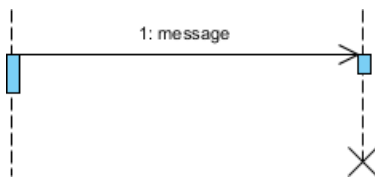
2.2.2.4 Pengertian *Sequence Diagram*

Menurut (Munawar, 2018) *Sequence diagram* adalah salah satu jenis diagram pada UML yang menjelaskan interaksi obyek yang berdasarkan urutan waktu. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case. Berikut ini merupakan simbol-simbol dari *sequence diagram* :

Tabel 2.4 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor/nama aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun aktor dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2	Lifeline 	Menyatakan kehidupan suatu objek.



3	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> nama objek : nama kelas </div>	Menyatakan objek yang berinteraksi pesan.
4	<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5	<p>Pesan tipe create</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	<p>Pesan tipe call</p> <div style="text-align: center; margin: 10px auto;"> <p>l : nama_metode()</p> <p>—————→</p> </div>	Menyatakan suatu objek memanggil atau operasi atau metode yang ada pada objek lain atau dirinya sendiri.
7	<p>Pesan tipe send</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	<p>Pesan tipe destroy</p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri.
9	<p>Pesan tipe return</p>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi menghasilkan



	<p>suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
--	---

Sumber : Haviluddin (2011)

2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Pengertian aplikasi adalah suatu perangkat lunak yang dibuat khusus untuk memenuhi kebutuhan berbagai aktivitas dan pekerjaan, misalnya; pelayanan masyarakat, aktivitas niaga, periklanan, game, dan berbagai aktivitas lainnya. (Pramana, 2020)

2.3.2 Pengertian Codeigniter

Menurut (Ramadhan & Zaidiah, 2022) menjelaskan *codeigniter* adalah sebuah aplikasi yang tersedia dengan bebas berbentuk kerangka kerja atau *framework* untuk mengembangkan sebuah *website* dengan menggunakan bahasa pemrograman *Hypertext Preprocessor (PHP)*. *Codeigniter* salah satu jenis *framework* aplikasi web yang mudah digunakan. *Framework* adalah kerangka kerja yang digunakan untuk membangun sebuah aplikasi yang dapat digunakan pada desktop atau dapat digunakan pada *website*.

2.3.3 Pengertian Pencatatan

Secara bahasa, pencatatan adalah kegiatan atau proses pendokumentasian suatu aktivitas dalam bentuk tulisan, grafik, gambar, atau suara. Menurut Kamus Besar Bahasa Indonesia (KBBI), pencatatan adalah proses, cara, atau perbuatan mencatat. Pencatatan dalam merujuk pada proses pengumpulan data secara teratur.

2.3.4 Pengertian Barang

Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata barang adalah benda umum (segala sesuatu yang berwujud atau berjasad). Barang adalah segala



jenis benda atau material yang memiliki nilai dan dapat dipindahkan dari satu tempat ke tempat lain. Barang bisa berupa produk fisik yang dihasilkan dari proses produksi, seperti pakaian, kendaraan, atau peralatan elektronik. Namun, barang juga bisa berupa produk *digital* seperti *file* musik atau video, serta layanan seperti langganan *streaming* atau jasa pengiriman barang.

2.3.5 Pengertian Gudang

Gudang adalah sebuah tempat yang digunakan untuk menyimpan barang atau inventaris dalam jumlah besar dan untuk jangka waktu yang lama. Gudang sering digunakan oleh perusahaan atau organisasi sebagai distribusi barang, tempat penyimpanan persediaan, atau sebagai tempat pengiriman dan penerimaan barang.

2.3.6 Pengertian Penerapan *Codeigniter* pada Aplikasi Pencatatan Barang Masuk dan Keluar di Gudang Fungsi ICT PT Pertamina Patra Niaga Regional Sumbagsel

Pengertian Penerapan *Codeigniter* pada Aplikasi Pencatatan Barang Masuk dan Keluar di Gudang Fungsi ICT PT Pertamina Patra Niaga Regional Sumbagsel adalah proses pembuatan aplikasi pencatatan barang yang masuk dan barang yang dikirim keluar menggunakan *framework codeigniter* sebagai pengembang *website*.

2.4 Teori Program

2.4.1 Pengertian *Hypertext Markup Language* (HTML)

Menurut (Rerung, 2018) *Hypertext Markup Language* (HTML) merupakan bahasa dasar pembuatan web. HTML menggunakan tanda (mark), untuk menandai bagian-bagian dari text. HTML disebut sebagai bahasa dasar, karena dalam membuat web, jika hanya menggunakan HTML maka tampilan web terasa hambar.



2.4.2 Penulisan HTML

Penulisan HTML sebagai berikut :

```
<HTML>
<head>
<title>Selamat Datang</title>
</head>
<body>
</body>
</HTML>
```

2.4.3 Pengertian *Cascading Style Sheets (CSS)*

Pengertian CSS adalah singkatan dari *Cascading Style Sheets*. Biasanya sering dikaitkan dengan HTML, karena keduanya saling melengkapi dimana HTML ditunjukan untuk membuat struktur, atau konten dari halaman web. Sedangkan CSS digunakan untuk tampilan dari halaman web tersebut. (Rerung, 2018)

2.4.4 Penulisan CSS

2.4.4.1 Penulisan dengan Inline CSS

```
<h1 style="color:red;">
H1 dengan CSS
</h1>
```

2.4.4.2 Penulisan dengan Embedded CSS

```
<head>
<style
type="text/CSS">
h1{color:red;}
</style>
</head>
```



2.4.4.3 Penulisan dengan External CSS

Buat sebuah file CSS, misal style.CSS, lalu tulis kode berikut :

```
p{font-family:arial; font-size:small;}h1{color:red;}
```

2.4.5 Pengertian *Hypertext Preprocessor* (PHP)

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web serve-side* (dieksekusi pada komputer yang memiliki web server) yang bersifat open source (gratis dan bisa disebarluaskan atau digunakan secara gratis). Rasmus Lerdorf yang telah membuat PHP ini pada tahun 1994, yang dikembangkan dari C dan Perl, pada awalnya bahasa pemrograman ini hanya digunakan untuk menangani pengolahan data form atau formulir di internet saja (Siswandi, 2021).

2.4.6 Penulisan Program PHP

Cara penulisan skrip PHP ada dua macam, yaitu *Embedded Script* dan *Non Embedded Script*, contoh :

a. *Embedded Script*

```
<HTML>
<head>
</head>
<body>
<?PHP
Echo "Hallo Dunia";
?>
</body>
</HTML>
```

b. *Non Embedded Script*

```
<?PHP
Echo "<HTML>";
Echo "<head>";
Echo "<title> 'Mengenal PHP'</title>";
Echo "</head>";
Echo "<body>";
Echo "<p> PHP cukup menyenangkan</p>";
Echo "</body>";
Echo "</HTML>";
?>
```



2.4.7 Pengertian JavaScript

JavaScript secara umum adalah sebuah bahasa pemrograman yang menempel dan bekerja pada halaman HTML. Dan disebut sebagai bahasa script karena program *JavaScript* dapat bekerja langsung tanpa diperlukan *compile*. (Siswandi, 2021) “*Javascript* adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti internet *explorer*, *mozilla firefox*, *netscape* dan *opera*. Kode *javascript* dapat disisipkan dalam halaman web menggunakan tag *scripting*.” *JavaScript* dipakai untuk membangun interaktifitas pada halaman-halaman HTML statis.

2.4.8 Penulisan JavaScript

Untuk penulisan kode *Javascript* adalah sebagai berikut :

```
<html>
<head>
<title>Mencoba JavaScript</title>
<script>
console.log(“Halo, ini Jagoan Hosting”);
</script>
</head>
<body>
<script>
document.write(“Ayo belajar JavaScript!”);
</script>
</body>
</html>
```

2.4.9 Pengertian Sublime Text

Sublime Text adalah sebuah aplikasi teks editor yang digunakan untuk menulis kode pemrograman, script, atau dokumen teks lainnya. *Sublime Text* dikembangkan oleh *Sublime HQ* dan tersedia untuk sistem operasi *Windows*, *macOS*, dan *Linux*. *Sublime Text* memiliki tampilan antarmuka yang bersih dan



minimalis, serta dilengkapi dengan fitur-fitur yang memudahkan pengguna dalam menulis kode, seperti penyorotan sintaks, *auto-completion*, dan kemampuan untuk membuka dan mengedit banyak file sekaligus dalam tab terpisah. Selain itu, *Sublime Text* juga dapat diatur dan dikustomisasi sesuai dengan preferensi pengguna dengan bantuan berbagai *plugin* dan tema yang tersedia.

2.4.10 Pengertian XAMPP

Menurut (Wicaksono, 2017) adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP sangat populer di kalangan pengembang web karena mudah digunakan, berfungsi sebagai *server* pengembangan lokal yang stabil, dan mendukung sebagian besar fitur yang dibutuhkan untuk pengembangan web.