



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Fofid dan Suryadi (2019:2), “Komputer merupakan suatu alat yang gunanya sangat bervariasi, secara umum, kegunaan komputer adalah mengolah data menjadi informasi dapat disimpan atau dicek hasilnya. Dengan demikian, karena peralatan yang ada pada suatu unit komputer tidak terlalu sama, karena sangat tergantung dari kebutuhan pemakainya.”

Menurut Manurung (2022:54), “Komputer adalah perangkat elektronik yang diprogram untuk menerima data mentah sebagai input dan memprosesnya dengan sekumpulan *instruksi* (program) untuk menghasilkan *output*.”

2.1.2 Pengertian Perangkat Lunak

Menurut Romindo, dkk (2023:1), “Perangkat lunak (*Software*) merupakan semua perintah yang digunakan untuk mengolah data dan informasi.”

Menurut Purba, dkk (2022:33), “Perangkat lunak merupakan program komputer yang berfungsi menghubungkan antara pengguna dan komputer yang digunakan. Dapat dibilang perangkat lunak merupakan sebagai media penerjemah perintah yang diberikan oleh pengguna kepada komputer untuk selanjutnya diproses melalui perangkat keras komputer tersebut.”

2.1.3 Pengertian Data

Menurut Pane, Zamzam, dan Fadhillah (2020:55), “Data merupakan sekumpulan keterangan atau fakta mentah berupa simbol, angka, kata-kata atau citra yang didapatkan melalui proses pengamatan atau pencarian ke sumber-sumber tertentu. Definisi data adalah kumpulan-kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh dari hasil pengamatan



(*observasi*) dan dapat diolah menjadi bentuk yang lebih kompleks, seperti informasi *database*, atau solusi untuk masalah tertentu.”

Menurut GR (2022:5), “Data adalah sebuah informasi yang berupa tulisan atau angka-angka atau yang biasa kita tahu dalam IT yaitu data yang berisi *filefile/dokumen*, data, mp3, mp4, bahkan tidak menutup kemungkinan berisi data yang sangat penting atau rahasia. Maka dibentuklah sebuah software atau layanan yang dibuat untuk mengatur sebuah data tersebut.”

2.1.4 Pengertian Basis Data (*Database*)

Menurut Pane, Pratama dan Nasution (2020:135), “*Database* adalah sekumpulan sebuah informasi yang disimpan di komputer sehingga dapat di *check* dengan suatu aplikasi komputer untuk mendapat petunjuk dari *database* itu.”

Menurut Rosa dan Shalahuddin (dalam Priyandanu dkk, 2020), “Basis data adalah media untuk menyimpan data agar dapat diakses dengan cepat dan mudah.”

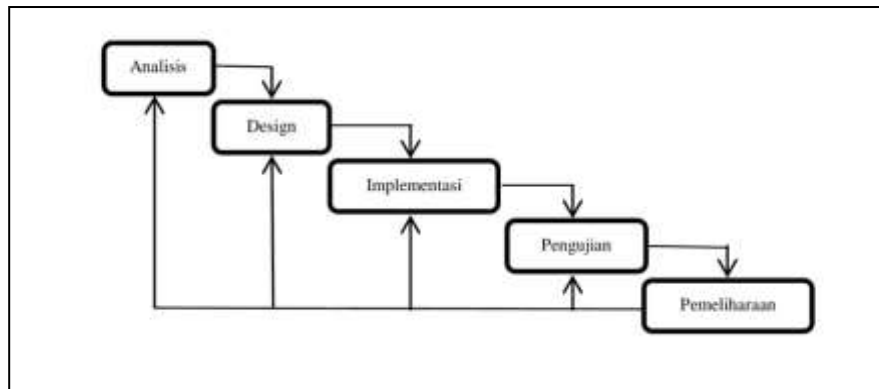
2.1.5 Metode Pengembangan Sistem Waterfall

Menurut Mujiyono, Adhawiyah dan Rohman (2023), metode air terjun atau sering disebut dengan metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linier Sequential Model*” dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*) serta penyerahan sistem ke para pengguna (*deployment*) yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan.

Metode Waterfall merupakan salah satu model SDLC (*System Development Life Cycle*) yang sering digunakan dalam pengembangan system atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengolahan



(Maintenance) dan dilakukan secara bertahap. Pengembang perlu mengetahui lebih lanjut tentang bagaimana proses pengembangan sistem jika menggunakan model *waterfall* dan juga karakteristik dari model *waterfall* tersebut.



Gambar 2.1 Metode Waterfall

1. Analisis

Dalam Analisa kebutuhan ini bertujuan untuk menganalisis kebutuhan yang dibutuhkan dalam perancangan baik berupa dokumen maupun sumber lain yang dapat membantu dalam menentukan solusi permasalahan yang ada baik dari sisi user maupun admin.

2. Design

Pada tahap ini penulis membentuk rancangan desain dan model aplikasi yang dikembangkan ataupun diciptakan berdasarkan hasil Analisa dari proses sebelumnya.

3. Implementasi

Hasil dari tahapan sebelumnya akan direalisasikan pada tahap ini sebagai unit program. Proses pembuatan aplikasi simulai sesuai dengan desain model yang telah dirancang.

4. Pengujian

Pada tahap pengujian ini program dilakukan dengan menggunakan *black box* testing dan *usability* testing dengan harapan bahwa perancangan yang sudah dibuat dapat berjalan dengan sesuai kehendak.

5. Pemeliharaan

Dalam proses pemeliharaan ini penulis mengupayakan pengembangan *system* yang telah dirancang terkait *software* dan *hardware* dapat dibuat maksimal agar aplikasi dapat berjalan dengan baik.



2.2 Teori Khusus

2.2.1 Unified Modeling Language (UML)

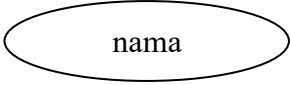
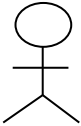

Menurut Habibi, Putra dan Putri (2020:46), “*Unified Modeling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.”

2.2.2 Jenis-Jenis Diagram UML


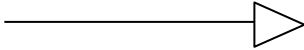
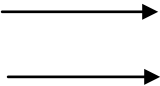
2.2.2.1 Pengertian Use Case Diagram

Menurut Hutabri, dan Putri (2019), “*Use case diagram* atau diagram *use case* adalah diagram untuk memodelkan perilaku suatu sistem yang akan dirancang dengan menggambarkan interaksi antara satu atau lebih aktor yang akan menggunakan system.”

Tabel 2.1 Simbol-simbol *Use case diagram*

No.	Simbol	Deskripsi
1.	<p><i>use case</i></p> 	Merupakan fungsi atau kegunaan sistem untuk unit atau sistem pertukaran pesan antara unit dan aktor
2.	<p>Nama aktor</p>  <p><i>Actor</i> (aktor)</p>	Merupakan aktor atau orang, sistem atau proses yang berperan dan memiliki hubungan dengan sistem informasi yang dibuat.
	<p><i>Association / asosiasi</i></p> 	Merupakan interaksi aktor dengan <i>use case</i> atau komunikasi antara aktor dan <i>use case</i> .

Lanjutan Tabel 2.2 Simbol-simbol *Use case diagram*

4.	<i>Extend</i> (Ekstensi) <<extend>> 	Merupakan relasi tambahan <i>use case</i> dengan <i>use case</i> lain. Tanpa <i>use case</i> yang
		ditambahkan <i>use case</i> dapat berdiri. Arah dari panah ke arah <i>use case</i> yang ditambahkan.
5.	<i>Generalization</i> (Generalisasi) 	Merupakan hubungan umum – khusus (generalisasi dan spesialisasi) untuk dua <i>use case</i> . Satu <i>use case</i> memiliki fungsi yang lebih general (umum), dari <i>use case</i> yang lainnya. Arah panah menunjukkan arah ke <i>use case</i> yang umum
6.	<i>Uses</i> atau <i>Include</i>  <<include>> <<uses>>	Merupakan relasi dua <i>use case</i> , <i>use case</i> yang ditambahkan dengan <i>use case</i> tambahan. <i>use case</i> yang ditambahkan memerlukan <i>use case</i> tambahan untuk menjalankan kegunaan atau fungsi sebagai syarat menjalankan <i>use case</i> .

Sumber : Hutabri dan Putri (2019)



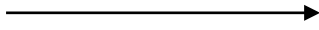
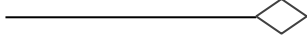
2.2.2.2. Pengertian *Class Diagram*

Menurut Hutabri dan Putri (2019), “*Class Diagram* atau Diagram kelas digunakan untuk membuat sistem dengan mendeskripsikan struktur sistem dari sisi pendefinisian kelas yang dibuat. Kelas memiliki variabel-variabel yang dimiliki oleh kelas (atribut) dan fungsi-fungsi yang dipunyai oleh kelas (metode atau operasi).”

Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	<p><i>Class</i> (kelas)</p> 	Merupakan kelas yang ada pada struktur sistem. Memiliki atribut dan operasi dalam kelas.
2.	<p><i>Interface</i> (antarmuka)</p>  <p>nama_interface</p>	Merupakan kemiripan dengan kelas tetapi memiliki metode yang di deklarasikan tanpa isi dan tanpa atribut kelas.
3.	<p><i>association / asosiasi</i></p> 	Merupakan relasi antarkelas (umum), biasanya dilengkapi dengan <i>multiplicity</i> .
4.	<p><i>Directed association</i> (Asosiasi berarah)</p> 	Merupakan relasi antara kelas bermakna satu kelas digunakan oleh kelas yang lain, biasanya dilengkapi dengan <i>multiplicity</i>
5.	<p>Generalisasi</p> 	Merupakan relasi antar kelas bermakna generalisasi spesialisasi (umum ke khusus)





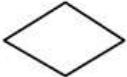

6.	<i>Dependency</i> (Kebergantungan) 	Merupakan relasi kebergantungan (<i>dependency</i>) antara kelas
7.	<i>Aggregation</i> (Agregasi) 	Merupakan relasi antara kelas bermakna semua bagian (<i>wholepart</i>).

Sumber : Hutabri dan Putri (2019)


2.3.2.3 Pengertian *Activity Diagram*

Menurut Hutabri dan Putri (2019), “Diagram aktivitas (*activity diagram*) mendeskripsikan aliran kerja (*workflow*) atau aktivitas sistem atau proses bisnis atau menu yang terdapat di dalam sistem atau perangkat lunak.”

Tabel 2.4 Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal/ Akhir 	Merupakan status awal atau akhir keadaan dari sistem, setiap diagram aktivitas memiliki satu status awal.
2.	Aktivitas 	Merupakan kegiatan yang dilakukan sistem, sering dimulai dengan kata kerja.
3.	<i>Decision</i> (Percabangan) 	Merupakan hubungan percabangan untuk keputusan aktivitas yang memiliki lebih dari satu pilihan.
4.	<i>Join</i> (Penggabungan) 	Merupakan hubungan penggabungan jika satu atau lebih aktivitas menjadi satu.

Lanjutan Tabel 2.5 Simbol-simbol *Activity Diagram*






5.	<i>Swimlane</i> 	Merupakan yang memisahkan organisasi bisnis. Memiliki tanggung jawab untuk aktivitas yang terjadi.
----	--	--

Sumber : Hutabri dan Putri (2019)



2.3.2.4 Pengertian *Sequence Diagram*

Menurut Hutabri dan Putri (2019), “Diagram Sekuen (*Sequence Diagram*) mendeskripsikan perilaku objek pada *use case* dengan menjelaskan alur waktu hidup dari objek dan pesan atau *message* yang diterima dan dikirim antar objek.”

Tabel 2.6 Simbol-simbol *Sequence Diagram*

No	Nama	Keterangan
1.	<i>Entity Class</i> 	Kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
2.	 <i>Boundray Class</i>	Kumpulan kelas menjadi interaksi antar aktor dengan sistem.
3.	 <i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
4.	 <i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
5.	 <i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

Lanjutan Tabel 2.7 Simbol-simbol *Sequence Diagram*

6.	<p style="text-align: center;"><i>Activation</i></p> 	<p><i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi</p>
7.	<p style="text-align: center;"><i>Lifeline</i></p> 	<p>Garis terputus dengan objek sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

Sumber: Putra dkk. (2020:36)

2.3 Pengertian Judul

2.3.1 Pengertian Aplikasi

Menurut Rahayu, Fajri dan Hambali (2019:21), “Aplikasi adalah penggunaan dalam suatu komputer, intruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.”

Menurut Pane, Zamzam dan Fadillah (2020:53), “Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu.”

2.3.2 Pengertian Pengolahan Data

Menurut Julkarnain dkk (2020), “Pengolahan data merupakan proses input dan output data menjadi bentuk yang lain yang sangat dibutuhkan yaitu berupa informasi khususnya pengolahan data ternak.”

2.3.3 Pengertian Pelanggan

Menurut Risudah (2019), “Pelanggan merupakan konsumen yang melakukan pembelian atau pemakaian secara berkala terhadap suatu produk atau jasa, yang dimana konsumen merupakan orang yang membeli atau menggunakan suatu produk atau jasa”.



2.3.4 Pengertian Restitusi

Menurut Nadita dkk (2023), “Restitusi ialah pembayaran ganti kerugian yang dibebankan kepada pelaku berdasarkan putusan pengadilan yang berkekuatan hukum tetap atas kerugian”

2.3.5 Pengertian Permohonan

Menurut Kamus Besar Bahasa Indonesia *online* (2022), “Permohonan adalah permintaan kepada orang yang lebih tinggi kedudukannya dan sebagainya”.

2.3.8 Pengertian Penyambungan Baru

Menurut Bariroh dkk (2017), “Penyambungan Baru adalah penyambungan daya listrik untuk pelanggan yang akan menggunakan listrik pada suatu lokasi dan telah memenuhi persyaratan permintaan penyambungan baru”.

2.4. Teori Program

2.4.1 Pengertian *Hypertext Markup Language* (HTML)

Menurut (Mulyani, 2022:76), “ *Hypertext Markup Language* (HTML) adalah bahasa markup yang digunakan untuk membuat halaman website. HTML menjadi bahasa paling dasar untuk mengembangkan website. HTML dapat melakukan banyak hal saat membangun website, misalnya menyisipkan link, mengatur style dan struktur teks hingga membuat obyek disuatu halaman website”.

2.4.2 Pengertian *Cascading Style Sheets* (CSS)

Menurut Mulyani (2022:77) CSS adalah singkatan dari *Cascading Style Sheets*, yang secara harfiah berarti lembar penataan menurun. CSS adalah kumpulan perintah yang digunakan untuk menjelaskan tampilan sebuah halaman situs web dalam *markup language*



2.4.3 Pengertian *Hypertext Preprocessor* (PHP)

Sari dan Utami (2021:126) mengatakan “PHP atau *Hypertext Preprocessor* merupakan bahasa *scripting* yang dirancang untuk membuat halaman web dinamis”. Adapun pengertian PHP menurut Mulyani (2022:75) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersama dengan HTML.

2.4.4 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*.

Menurut Mulyani (2022:80) *Javascript* adalah salah satu tools dasar untuk *front end developer* yang dapat menentukan fungsi-fungsi yang ada di dalam website, seperti menambahkan audio, video, game dan animasi.

2.4.5 Pengertian *Visual Studio Code*

Menurut (Mulyani, 2022:107), “Visual Studio Code atau VS Code adalah teks editor keluaran dari perusahaan *Microsoft*. Teks editor ini sangat populer dikalangan programmer karena *visual code* memiliki design yang sangat baik dan memudahkan para programmer. Selain itu *text editor* ini cukup ringan dan dapat mendukung hampir semua bahasa pemrograman, seperti vb.net, C#, C, C++, Java, HTML, dan lainnya”.

2.4.6 Pengertian MySQL

Menurut (Anton Subagia, 2018), “MySQL merupakan software database open source yang paling populer di dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang software dan aplikasi baik di platform web maupun desktop”

2.4.7 Pengertian *Framework Laravel*

Menurut (Mulyani, 2022:82), “Laravel adalah sebuah framework website berbasis PHP yang sangat ekspresif. Ia juga memiliki sintaks yang elegan, yang



bertujuan untuk mengurangi kesulitan setiap developer dalam menggunakannya, seperti otentikasi, routing, dan cache. Laravel diciptakan untuk membantu developer membangun sebuah aplikasi yang kuat dan besar, dengan wadah control inversi, migrasi sistem, dan didukung oleh unit integrasi”.

2.4.8 Pengertian XAMPP

Menurut Habibi, Putra dan Putri (2020:5), “Xampp merupakan perangkat lunak bebas, yang mendukung banyak system operasi merupakan kompilasi dari beberapa program.”

Menurut Irsal, dkk (2020:33), “Xampp merupakan singkatan dari X (empat sistem operasi apapun) Apache, MySQL, PHP, dan Perl. Xampp adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket.”