



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Sistem Informasi

Menurut R. Kelly Rainer dan Brad Prince (2021:2) sistem informasi adalah sebagai berikut,” sistem informasi (SI) adalah Sebuah sistem yang mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan tertentu tujuan”. Menurut prehanto (2020: 21) sistem informasi merupakan proses pengumpulan, penyimpanan, analisis sebuah informasi dengan tujuan tertentu. Sistem informasi yang terdiri dari data input dan akan sebuah menghasilkan laporan *output* sehingga diterima oleh sistem lainnya serta kegiatan strategi dalam suatu organisasi dalam melakukan tindakan atau keputusan.

2.1.2 Pengertian Internet

Menurut Wibawanto, (2020) Internet adalah sebuah alat yang berbentuk jaringan dengan sistem komunikasi global yang menghubungkan seluruh jaringan komputer di dunia sehingga segala informasi mudah untuk di dapatkan.

2.1.3 Pengertian Basis Data (*Database*)

Menurut Khairil (2021:19), “Basis data adalah sekumpulan data yang saling berhubungan secara logis dan terorganisir dengan baik. Kumpulan data tersebut yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik”.



Menurut Kristy (2021:290), *Database* merupakan kumpulan data yang saling berhubungan, sehingga kita dapat memperoleh informasi data mengenai dan dengan cepat. *Database* dapat berasal dari satu atau kumpulan beberapa tabel yang terhubung menjadi satu kesatuan. Tabel merupakan tempat untuk menampung data

BAB II Tinjauan Pustaka

dengan susunan tertentu. Setiap kolom dalam tabel merupakan *field* yang harus berisi data dengan jenis yang sama, sedangkan setiap baris dalam tabel merupakan *record* berisi kumpulan data yang terdiri atas beberapa *field*.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya yaitu objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan di mana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Jadi secara konsep basis data atau *database* adalah kumpulan dari data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi. Atau basis data (*database*) merupakan kumpulan dari data yang saling berhubungan (relasi) antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu. Pada komputer, basis data disimpan dalam perangkat *hardware* penyimpanan, dan dengan *software* tertentu dimanipulasi untuk kepentingan atau



kegunaan tertentu. Hubungan atau relasi data biasanya ditunjukkan dengan kunci (*key*) dari tiap *file* yang ada. Data merupakan fakta atau nilai (*value*) yang tercatat atau merepresentasikan deskripsi dari suatu objek. Data yang merupakan fakta yang tercatat dan selanjutnya dilakukan pengolahan (proses) menjadi bentuk yang berguna atau bermanfaat bagi pemakainya akan membentuk apa yang disebut informasi. Data dalam basis data merupakan item terkecil dan terpenting untuk membangun basis data yang baik dan *valid*. Data dalam basis data bersifat *integrated* dan *shared* :

1. Terpadu (*integrated*), berkas-berkas data yang ada pada basis data saling terkait (terjadi dependensi data);
2. Berbagi data (*shared*), data yang sama dapat dipakai oleh sejumlah pengguna dalam waktu yang bersamaan. Sering dinamakan sebagai sistem *multiuser*.

Dalam implementasinya, untuk memudahkan dalam mengakses data, data disusun dalam suatu struktur logis yang menjelaskan bahwa :

- a. Kumpulan tabel menyusun basis data.

- b. Tabel tersusun atas sejumlah *record*.
- c. Sebuah *record* mengandung sejumlah *field*.
- d. Sebuah *field* disimpan dalam bentuk kumpulan bit.

Istilah secara umum tabel yang baik memiliki beberapa karakteristik sebagai berikut :

- a. Memiliki kolom (*field*) dan baris data (*record*).
- b. Nama kolom bersifat unik (tidak ada yang sama).
- c. Sebuah kolom hanya memiliki satu tipe data.
- d. Antar kolom tidak terpisah kolom kosong atau tanpa nama kolom.
- e. *Header* (judul kolom) hanya terdiri dari satu baris dan ditempatkan pada baris pertama.



f. Antar baris data tidak terpisah baris kosong ataupun baris subtotal.

Suatu bangunan basis data memiliki jenjang sebagai berikut :

3. Basis data/*Database*, merupakan kumpulan dari bermacam-macam tipe record yang memiliki hubungan antar *record*.
4. Berkas/*File*, merupakan sekumpulan data rekaman yang berkaitan dengan suatu objek.
5. *Record*, merupakan sekumpulan *field*/atribut/data *item* yang saling berhubungan dengan obyek tertentu dengan panjang *record* yang tetap, semua *field* dalam *record* memiliki ukuran yang tetap. Variabel panjang *record*, *field*- *field* dalam *record* dapat memiliki ukuran yang berbeda (metode penandaan yang digunakan adalah penanda akhir *record*, indikator panjang, dan tabel posisi *record*).
6. *Field*, menyatakan data terkecil yang memiliki makna. *Field* merupakan implementasi dari suatu atribut data. *Field* merupakan unit terkecil dari data yang berarti (*meaningful data*) yang disimpan dalam suatu *file* atau basis data lain untuk *field* yaitu elemen data, kolom *item*, dan atribut. Contoh *field* yaitu nama, alamat, telepon, dan jenis kelamin.
7. *Byte*, adalah bagian terkecil yang di alamatkan dalam memori. *Byte* merupakan kumpulan bit yang secara konvensi terdiri atas kombinasi delapan bit yang menyatakan sebuah karakter dalam memori (1 *byte*= 1 karakter).
8. *Bit*, adalah sistem *binner* r yang terdiri atas dua macam nilai, yaitu 0 dan 1 sistem *binner* merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan rangkaian komponen elektronik dan hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut.



9. Dalam pengembangan suatu sistem informasi, basis data memiliki peran yang penting dan sangat diperlukan.
10. Salah satu komponen penting dalam suatu sistem informasi adalah basis data, karena basis data merupakan dasar dalam menyediakan informasi.
11. Informasi dikatakan lebih bernilai jika memiliki manfaat yang efektif dibandingkan dengan biaya dalam mendapatkannya. Dalam hal ini, basis data akan menentukan kualitas informasi.
12. Data akan dapat saling berelasi dengan mengimplementasikan basis data.
13. Basis data mengurangi duplikasi data (*data redundancy*).

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Dhanta dikutip dari Sanjaya (2020), "Aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Exce. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan". Sedangkan menurut Rachmad hakim S dalam (Tri, 2020) aplikasi adalah perangkat lunak yang digunakan dalam tujuan tertentu, seperti mengelola dokumen, mengatur *Windows* dan permainan (*game*).

2.2.2 Pengertian Penggajian

Menurut Rahmawati, M., (2021) Penggajian merupakan bentuk timbal balik atas suatu ikatan pekerjaan dalam bentuk jasa yang telah dilakukan, yang diberikan oleh perusahaan kepada karyawannya. Dibayarkan dalam bentuk uang yang diatur dan ditetapkan berdasarkan asas perundang-undangan.



2.2.3 Pengertian Karyawan

Menurut Wahyono, T., (2022) Karyawan adalah sumber daya manusia yang menjadi ujung tombak keberhasilan atau tidaknya sistem yang sedang di gerakkan oleh sebuah perusahaan untuk menghasilkan sesuatu yang berguna bagi perusahaan dan masyarakat. Sedangkan menurut Septiana (2020) Karyawan merupakan komponen utama suatu perusahaan yang menjadi perencana dan pelaku aktif dalam setiap aktifitas perusahaan yang mana mereka mempunyai pikiran, perasaan, keinginan, status dan latar belakang orang itu sendiri yang membawa kedalam suatu perusahaan sehingga tidak seperti mesin, uang dan material yang sifatnya pasif dan dapat di kuasai dan di atur sepenuhnya dalam mendukung tercapainya tujuan perusahaan.

2.2.4 Pengertian Aplikasi Penggajian Karyawan Berbasis Website Menggunakan Framework Condeigniter Pada CV Naira Brother Expres

Aplikasi penggajian karyawan berbasis website menggunakan framework CodeIgniter adalah sistem online yang dirancang untuk mengelola dan mengotomatisasi proses penggajian karyawan di CV Naira Brother Expres. Dengan menggunakan CodeIgniter, aplikasi ini memungkinkan pengelolaan data karyawan, perhitungan gaji otomatis, pembuatan slip gaji, dan penyusunan laporan penggajian. Aplikasi ini juga menawarkan keamanan data yang tinggi, aksesibilitas dari mana saja, dan efisiensi dalam pengelolaan penggajian, sehingga memudahkan manajemen dan karyawan dalam mengakses informasi penggajian dengan cepat dan aman.

2.3 Teori Khusus

2.3.1 Pengertian *Unified Modeling Language* (UML)

Menurut Riri Fitri Sari dan Ardiati (2021:107) : “*Unified Modeling Language* (UML) adalah bahasa pemodelan yang digunakan untuk menspesifikasikan, memvisualisasikan, membuat, dan mendokumentasi artefak sistem perangkat lunak baik yang sedang dirancang ataupun dikembangkan”.

Menurut Munawar (2021:49) : “UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek, karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Berdasarkan definisi di atas, maka dapat disimpulkan bahwa UML (*Unified Modelling Language*) merupakan suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Jenis – jenis atau komponen dalam membuat suatu diagram UML, yaitu: *a. Use Case Diagram*

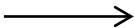
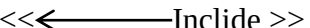
Use case diagram dipergunakan untuk menggambarkan suatu interaksi yang terdapat pada sistem informasi yang menggambarkan satu atau lebih aktor. Menurut (Suhimarita J, dan Susianto D., 2019) *use case diagram* digunakan untuk mengetahui siapa saja yang dapat menggunakan fitur-fitur yang ada pada sistem dan fitur-fitur yang ada pada sistem tersebut. Ada beberapa komponen kunci yang terdapat pada analisis *use case* adalah sebagai berikut ini:

1. *Actor*, merupakan suatu entitas yang akan menggunakan atau digunakan oleh sistem , pada umumnya pengguna tetapi bisa juga sistem eksternal.
2. *Connections*, merupakan suatu penghubung yang menghubungkan antara *actor* ke *use case*.

3. *Relationships*, merupakan suatu hubungan antara aktor itu sendiri dengan *use case*.

Simbol-simbol *Use Case Diagram* akan terlihat seperti tabel 2.1 berikut ini.

Tabel 2.1 simbol-simbol *Use Case Diagram*

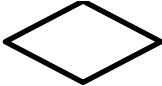
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor.
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika kondisi terpenuhi.

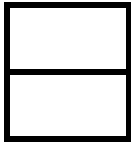
b. *Activity Diagram*

Diagram aktivitas digunakan untuk menggambarkan alur kerja perangkat lunak dari suatu proses bisnis. Yang menjadi perhatian menurut (Suhimarita J , dan Susianto D,. 2019) Diagram aktivitas menggambarkan aktivitas pada sistem, bukan apa yang dilakukan aktor, kapan aktivitas dapat dilakukan pada sistem.

Simbol- simbol *Activity Diagram* pada tabel 2.2 berikut ini.

Tabel 2.2 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas mem-liki sebuah status awal .
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/ <i>decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/ <i>join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas

		memiliki sebuah status akhir.
Simbol	Nama	Keterangan
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

c. *Class Diagram*

Diagram kelas digunakan untuk menggambarkan struktur sistem dan mendefinisikan setiap kelas yang digunakan untuk membangun sistem, yang disebut atribut dan metode. Menurut (Suhimarita J , dan Susianto D,. 2019) *Class* sendiri memiliki apa yang dikatakan dengan *atribute* dan metode.

1. *Attribute*, merupakan suatu *variable* yang dimiliki oleh *class*
2. *Attribute*, akan mendeskripsikan *property* dengan teks yang terdapat didalam kotak *class* tersebut.
3. *Operations/Metode*, merupakan fungsi-fungsi yang dimiliki oleh suatu *class*.

Class diagram sendiri akan mendeskripsikan jenis-jenis pada objek yang terdapat pada suatu sistem serta hubungan secara statis yang terdapat diantara keduanya. Pada *class diagram* akan menunjukan berbagai atribut serta oprasi pada sebuah class dan batasan pada hubungan yang terdapat pada objek tersebut.

Simbolsimbol *class diagram* pada tabel 2.3 berikut ini.

Tabel 2.3 Simbol-Simbol *Clas Diagram*

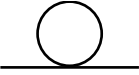
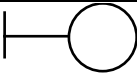
Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan satu hasil yang terukur bagi suatu aktor.
	<i>realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

	<i>dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

d. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada suatu objek, seperti *login*, *input*, *creat*, *edit*, *delete*, *logout* dan lain-lain. Simbol-simbol *Sequence Diagram* pada tabel 2.4 berikut ini.

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dario form
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i> .
	<i>A message</i>	Menggambarkan pengiriman pesan.

2.3.2 Pengertian *Flowchart*

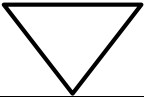
Menurut Indrajani (dalam Budiman *et al.*, 2021: 2186) mengatakan bahwa “*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempengaruhi penyelesaian masalah yang khususnya perludipelajari dan dievaluasi lebih lanjut. *Flowchart* dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan ataupun keduanya”.

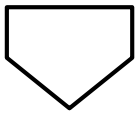
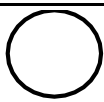
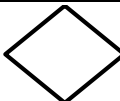
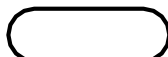
Indrajani (dalam Budiman *et al.*, 2021: 2186) menjelaskan simbol-simbol yang terdapat dalam *Flowchart*:

Tabel 2.5 Simbol-Simbol pada *Flowchart*

No	Simbol	Nama	Keterangan
1.		Dokumen	Dokumen atau laporan, dapat berupa hasil tulisan tangan atau cetakan komputer.
2.		<i>Input/Output;</i>	Proses <i>input</i> atau <i>output</i> data, epresentasi jurnal atau buku besar.

3.		<i>Predefined Process</i>	menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
4.		Proses	Pelaksanaan pemrosesan yang dilakukan oleh komputer.

5.		Proses Manual	Pelaksanaan pemrosesan yang dilakukan secara manual.
6.		<i>Disk Storage</i>	Data disimpan secara online melalui media yang dapat diakses secara langsung.
No	Simbol	Nama	Keterangan
7.		<i>Off-lineStorage</i>	<i>File</i> dokumen yang disimpan secara manual.
8.		Arus dokumen / arus pemrosesan	Arus pemrosesan atau arus dokumen, arus yang normal berada dibawah dan mengarah
9.		<i>Communication link</i>	Pengiriman data dari satu lokasi ke lokasi lainnya melalui jalur komunikasi.

10.		<i>Off-page Connector</i>	Menggambarkan keluar atau masuk proses dalam lembar atau halaman yang lain.
11.		<i>On-page connector</i>	Menghubungkan arus pemrosesan di halaman yang sama.
12.		<i>Decision</i>	Menyatakan kondisi yang menghasilkan beberapa kemungkinan atau aksi.
13.		Terminal	Simbol yang menyatakan permulaan atau akhir suatu proses atau program.

2.3.3 Pengertian Kamus Data

Menurut Prasetya dan Syafitri (2022: 23) mengatakan bahwa “Kamus data adalah dokumen yang berisi gambaran terperinci tentang data yang digunakan dalam basis data dan menggambarkan karakteristik semua item dalam basis data, menjelaskan bagian dari metadata dan nilai dari setiap atribut, mendeskripsikan aliran data, mencakup informasi cara catatan diatur, dan diperlukan untuk membantu programmer dalam membuat kode untuk mengakses data”.

Menurut Sukanto dan Shalahuddin (dalam Sirait dan Seabtian, 2019: 3-4) mengatakan bahwa “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran

(*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Sukamto dan Shalahuddin (dalam Sirait dan Seabtian, 2019:4) menjelaskan simbol-simbol yang terdapat dalam Kamus Data:

Tabel 2.6 Simbol-Simbol pada Kamus Data

No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang/ bernilai banyak
5.	()	data opsional
6.	*...*	batas komentar

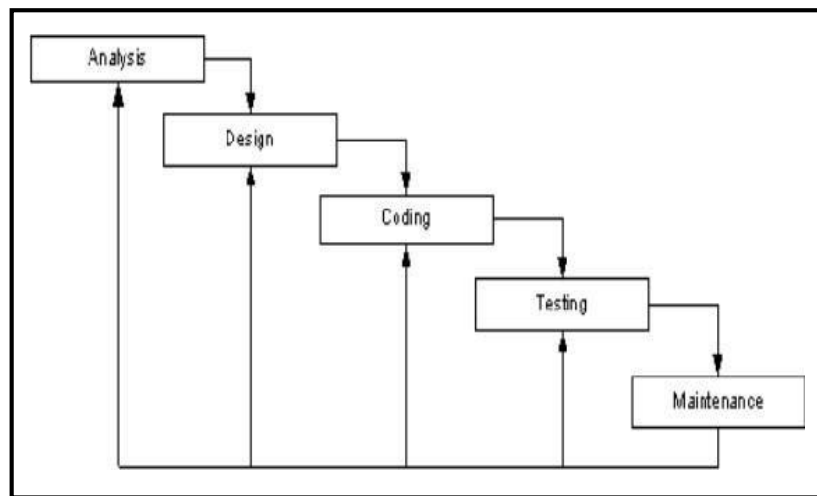
2.3.4 Metode Pengembangan Sistem

Dalam penelitian Laporan Akhir ini, penulis menggunakan metode pengembangan sistem Waterfall. Menurut (Nasution di kutip Rusmawan, 2019:89) “dalam membangun dan mengembangkan aplikasi ini perlu menggunakan metode *Waterfall*. *Waterfall* merupakan salah satu metode pengembangan sistem informasi yang bersifat sistematis dan sekuensial, artinya tahapan dalam metode ini dilakukan secara berurutan dan berkelanjutan”.

Sedangkan menurut Rosa dan Shalahuddin yang dikutip Suryadi dan Zulaikhah (2019:16) “metode waterfall adalah metode yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisa, desain, pengkodean, pengujian dan pendukung (*support*)”.

Berdasarkan kutipan diatas, dapat disimpulkan bahwa metode Waterfall merupakan salah satu metode pengembangan sistem informasi yang sekuensi linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*).

Pedekatan alur hidup perangkat lunak secara sekuensial atau terurut pada model air terjun (*Waterfall*) dapat dilihat pada gambar berikut ini:



Gambar 2.1 Metode *Waterfall*

Menurut Suryadi dan Zulaikhah (2019:16) menjelaskan adapun tahapan dalam metode *waterfall* yaitu:

1. Analisa Analisa merupakan suatu kegiatan yang dimulai dari proses awal di dalam mempelajari sesuatu serta mengevaluasi suatu bentuk permasalahan yang ada.
2. Desain Desain merupakan kegiatan dalam penggambaran, perencanaan, dan perancangan atau pengaturan dari beberapa elemen yang terpisah di dalam sistem menjadi kesatuan dan berfungsi dengan baik.
3. Pengkodean Pengkodean merupakan upaya dalam pengimplementasian desain menjadi perangkat lunak.

4. Pengujian Pengujian merupakan upaya dalam menelusuri lebih lanjut terhadap perangkat lunak yang telah dibuat untuk mendapatkan informasi mengenai kualitas perangkat lunak yang sedang diuji.
5. Pendukung Pendukung merupakan kegiatan yang dilakukan dalam perawatan dan pengubahan atau pengembangan dari perangkat lunak yang telah dibuat dan tidak terdeteksi saat tahapan pengujian.

2.4 Teori Program

2.4.1 Pengertian Visual Studio Code

Menurut Mulyani (2020:82) “Visual Studio Code adalah *text editor* keluaran dari perusahaan microsoft. Meskipun terbilang baru *text editor* yang satu ini sedang sangat populer di kalangan programmer. Karena visual studio code memiliki *design/tampilan* yang sangat baik dan memudahkan bagi para programmer”.

Sedangkan menurut Salamah (2021) “Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*”. Berdasarkan kedua kutipan diatas, penulis menyimpulkan bahwa Visual Studio Code merupakan teks editor keluaran dari perusahaan Microsoft untuk sistem operasi *multiplatform* yang memiliki tampilan yang sangat baik dan memudahkan bagi para programmer.



Gambar 2.2 Logo Visual Studio Code

2.4.2 Pengertian HTML (*Hyper Text Markup Language*)

Menurut Abdulloh (dalam Hidayat et al., 2019: 42) mengatakan bahwa “HTML singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa *tagtag* untuk membuat dan mengatur struktur *website*”. Sedangkan menurut Mulyani

(2020:76) “HTML merupakan singkatan dari *Hyper Text Markup Language*. Bahasa pemrograman HTML yang satu ini bisa anda gunakan untuk membangun *website* dengan mudah dan cepat”

Bedasarkan kedua kutipan diatas, penulis menyimpulkan bahwa HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang menggunakan tag-tag atau skrip untuk membuat dan mengatur struktur *website*. HTML memungkinkan pengguna untuk membangun dan mengembangkan *website* dengan mudah dan cepat.



Gambar 2.3 Logo HTML (*Hyper Text Markup Language*)

2.4.3 Pengertian CSS (*Cascading Style Sheet*)

Menurut Mulyani (2020:77) “*Cascading Style Sheets* adalah kumpulan perintah yang digunakan untuk menjelaskan tampilan sebuah halaman situs web dalam *mark-up language*. Menurut Sengkey (dalam Fatimah dan Samsudin, 2019: 36) mengatakan bahwa “*Cascading Style Sheet* (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih

terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML”.

Dari pernyataan diatas, dapat diambil kesimpulan bahwa *Cascading Style Sheets* (CSS) adalah kumpulan aturan atau perintah yang digunakan untuk mengatur tampilan dan memformat komponen-komponen dalam halaman situs web yang ditulis dengan bahasa markup seperti HTML dan XHTML agar lebih terstruktur, seragam, dan menarik.



Gambar 2.4 Logo CSS (*Cascading Style Sheets*)

2.4.4 Pengertian *Bootstrap*

Menurut Hidayat, et al (2019: 46) mengatakan bahwa “*Bootstrap* adalah sebuah *framework* untuk CSS dan berupa produk *open source* yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya *Bootstrap* ini dibuat untuk membuat standarisasi *front end* untuk semua programmer di perusahaannya. *Bootstrap* telah berubah dari yang sebelumnya adalah *CSSDriven* proyek ke sebuah *host* dari *Javascript plugins* dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol”.

Menurut Abdulloh (dalam Hidayat et al., 2019: 46) mengatakan bahwa “*Bootstrap* merupakan *framework* CSS yang sangat populer di kalangan pecinta pemrograman *Website*. Dengan menggunakan *Bootstrap*, proses desain *Website* tidak dibuat dari nol, sehingga proses desain *Website* lebih cepat dan mudah.

Bahkan tanpa kita membuat skrip CSS sedikit pun, kita sudah dapat membuat desain *Website* yang bagus”.

Dari pernyataan diatas, dapat diambil kesimpulan bahwa *Bootstrap* adalah sebuah kerangka kerja atau *framework* CSS yang digunakan sebagai alat yang sangat berguna bagi pengembang web dalam membangun *website* yang modern dan responsif dengan lebih efisien.



Gambar 2.5 Logo *Bootstrap*

2.4.5 Pengertian PHP (*Hypertext Preprocessor*)

Menurut *Enterprise* (dalam Hidayat et al., 2019: 43) mengatakan bahwa “PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *Website*. PHP merupakan Bahasa pemrograman berjenis *server-side*. Dengan demikian, PHP akan diproses oleh server yang hasil olahannya akan dikirim kembali ke *browser*. Oleh karena itu, salah-satu *tool* yang harus tersedia sebelum memulai pemrograman PHP adalah server”. Sedangkan menurut Mulyani (2020:75) “PHP atau bisa disebut *Hypertext Preprocessor* adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersama dengan HTML”.

Dari pernyataan diatas, dapat diambil kesimpulan bahwa PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang digunakan untuk membuat aplikasi berbasis *website* serta diintegrasikan dengan HTML dalam pengembangan situs web.



Gambar 2.6 Logo PHP (*Hypertext Preprocessor*)

2.4.6 Pengertian *Framework*

Menurut Hidayat, et al (2019: 46) mengatakan bahwa “*Framework* atau dalam bahasa Indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur - prosedur dan *class-class* untuk tujuan tertentu yang sudah siap lebih digunakan sehingga bisa mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal”.

2.4.7 Pengertian *Codeigniter*

Menurut Kurniawan et.al (2023), “*CodeIgniter* adalah sebuah *framework* pengembangan aplikasi web berbasis PHP yang dirancang untuk memudahkan dan mempercepat proses pengembangan. Dengan menggunakan *Framework Codeigniter*, pengembang *programmer* dapat dengan mudah. membangun aplikasi web yang efisien, *skala-able*, dan aman. *Framework* ini mengikuti pola desain *Model-View-Controller* (MVC) yang memisahkan logika bisnis, presentasi, dan interaksi dengan basis data. *CodeIgniter* menyediakan sejumlah fitur dan alat yang kuat. seperti sistem *routing* yang fleksibel, lapisan abstraksi basis data. keamanan tingkat lanjut, validasi *form*, pengujian, dan dokumentasi yang lengkap”.

Sedangkan menurut Isnardi et.al (2021), “*CodeIgniter* adalah sebuah web *application network* yang bersifat *open source* yang digunakan untuk membangun

aplikasi php dinamis. *CodeIgniter* menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya.”

Berdasarkan pendapat di atas, dapat disimpulkan bahwa *CodeIgniter* merupakan *framework* PHP berbasis web yang menggunakan pola desain MVC untuk mempercepat pengembangan aplikasi web. Ia memiliki fitur kuat seperti *routing* fleksibel, lapisan abstraksi basis data, keamanan tingkat lanjut, validasi *form*, pengujian, dan dokumentasi lengkap. *CodeIgniter* digunakan untuk membangun aplikasi web efisien, *skala-able*, dan aman.



Gambar 2.7 Logo Codeigniter

2.4.8 Pengertian MySQL

Menurut Rusmawan (2019:97) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi diseluruh dunia”.

Sedangkan menurut Adi (2020:2) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang multialur, dan multipengguna”. Berdasarkan definisi diatas, penulis menyimpulkan bahwa MySQL merupakan perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang multialur, dan *multi-user*.



Gambar 2.8 Logo MySQL

2.4.9 Pengertian XAMPP

Menurut Fadhlullah (2023:253), *XAMPP* merupakan salah satu *software* web server yang berfungsi untuk mendesain dan membuat sebuah situs website lebih berkembang, terutama pada server *local*. *XAMPP* atau yang disebut dengan *localhost XAMPP* ini berfungsi untuk membuat sebuah server lokal di sebuah komputer. Penggunaan *XAMPP* terdapat data yang harus dibuat biasanya disebut dengan *database*.

Menurut Wardhani (2022:95), *XAMPP* adalah sebuah *software* yang menjalankan peran sebagai *local* web server. *Local* web server artinya, *localhost* komputer yang berperan menjalankan web server dan juga sistem *database*. *Software* ini biasa digunakan untuk pengujian aplikasi web melalui *localhost*. Aplikasi web yang dikembangkan secara native, menggunakan *framework*, maupun CMS bisa diuji menggunakan *XAMPP*.

XAMPP dikembangkan oleh *Apache Friends*. *Apache Friends* sendiri merupakan proyek nirlaba yang bertujuan untuk mempromosikan server web *Apache*. Kepanjangan *XAMPP* sendiri mendefinisikan setiap program yang terintegrasi di dalamnya, sebagai berikut : Kode X (*Cross Platform*) Artinya, *software* ini kompatibel pada berbagai OS komputer seperti *Windows*, *Linux*, maupun *Mac OS*. Kode A (*Apache*) merupakan suatu aplikasi web server *default* yang digunakan oleh *XAMPP*. Aplikasi ini bersifat *open source* jadi bisa digunakan secara gratis dan bisa dikembangkan secara bebas. Kode M (MySQL / MariaDB) adalah aplikasi *database* server *default* yang ada di *XAMPP*.

Program ini digunakan untuk mengelola *database*. Kode P (PHP) merupakan bahasa pemrograman *back end* untuk mengembangkan aplikasi berbasis web. Kode P (Perl) merupakan bahasa pemrograman yang mencakup banyak tugas, seperti pengembangan web, pengembangan GUI, pemrograman jaringan, administrasi sistem program-program yang terintegrasi pada *XAMPP* secara tidak langsung bisa menggambarkan fungsi dan fitur dari *software* ini.



Gambar 2.9 Logo *XAMPP*

2.4.9.1 Mengenal Fungsi *XAMPP*

Menurut Wardhani (2022:95), secara umum, *XAMPP* berfungsi sebagai web server yang bisa diakses melalui *localhost* komputer. Aplikasi web yang belum sepenuhnya jadi, bisa dikembangkan dalam server lokal ini. *Developer* bisa menguji seluruh fungsi aplikasi web, menguji coba desain web, menyusun struktur konten, dan hal-hal lain sebelum resmi diluncurkan ke internet. Dengan *XAMPP*, aplikasi web sudah bisa dipresentasikan secara *real*. Antarmuka, struktur konten, bahkan fungsi yang berjalan di *back-end* cukup identik dengan aplikasi web yang asli.

Setelah pengujian di *XAMPP* selesai dilakukan, aplikasi web bisa di *publish* secara *online* dengan mengunggahnya di web *hosting*. Fitur-fitur yang ada dalam *XAMPP* sangat memudahkan proses pengujian aplikasi web. Selain *developer profesional*, *XAMPP* juga cukup lazim digunakan oleh pelajar atau mahasiswa untuk belajar mengembangkan aplikasi web.

2.4.9.2 Bagian penting dalam *XAMPP* dan Fitur Penunjangnya



a. *Htdocs*

Hasil instalasi *XAMPP* di komputer memiliki sejumlah *folder* di dalamnya. *Htdocs* merupakan salah satu *folder* tersebut. *Folder Htdocs* sendiri berfungsi untuk menyimpan seluruh *file* dan dokumen yang akan menyusun aplikasi web. Karena *XAMPP* merupakan server lokal, kapasitas penyimpanan dalam *htdocs* akan menyesuaikan dengan kapasitas hardisk pada PC. Untuk membuka *Htdocs*, hanya perlu membuka folder instalasi *XAMPP* dan foldernya akan terdapat di situ.

b. *PhpMyAdmin*

PhpMyAdmin merupakan program untuk mengelola sistem *database* pada aplikasi web. *PhpMyAdmin* tidak hanya digunakan pada *XAMPP*, web *hosting* profesional biasanya juga menyediakan program ini. Melalui *PhpMyAdmin*, kita bisa melakukan segala konfigurasi *database*. Kita bahkan bisa menuliskan *script query* menggunakan program ini. Cara membuka *PhpMyAdmin* di *XAMPP* cukup sederhana, hanya perlu menuliskan URL <http://localhost/phpMyAdmin> di web browser.

c. *Control Panel*

Secara umum, *Control Panel* pada *XAMPP* berfungsi untuk mengatur, menjalankan, dan mengelola fitur secara umum. Di *Control Panel* terdapat sejumlah fitur yang bisa diatur, diantaranya:

1. *Apache* -> Web server yang digunakan dalam *XAMPP*.
2. *MySQL* -> *Database* yang digunakan dalam *XAMPP*.
3. *FileZilla* -> *FTP Server* yang memungkinkan Anda untuk memindahkan *file* antar komputer pada jaringan lokal.
4. *Mercury* -> *Mail Server* berfungsi untuk menerima dan mengirim email pada suatu jaringan lokal.



5. *Tomcat* -> Fitur ini berfungsi untuk mempercepat akses data pada halaman aplikasi website dengan mendapatkan *Java Server Pages* (JSP) Setiap fitur tersebut terbagi dalam suatu *module*.

Kita bisa mengatur tiap fitur melalui perintah yang ada dalam modul. Kita bisa mencentang *box* pada fitur yang ingin Anda jalankan. Tombol action yang ada pada tiap modul diantaranya *Start/Stop*, *admin*, *config*, *logs*.

Di Control Panel juga terdapat sejumlah tombol yang menjalankan fungsi penunjang lain:

1. *Config* : Tombol ini menuju halaman konfigurasi dasar pada *XAMPP*.
2. *Netstat* : Untuk mengecek aplikasi yang berjalan di *port default* *XAMPP*.
3. *Shell* : Membuka *CMD*. Anda bisa menggunakannya untuk melakukan konfigurasi pada web server tentu saja dengan *script* *CMD*.
4. *Explorer* : Sebuah *shortcut* bagi Anda yang ingin langsung masuk ke folder instalasi *XAMPP* di *windows explorer*.
5. *Services* : Untuk menampilkan semua *service* yang pada saat itu sedang berjalan di *background*.
6. *Help* : Menu bantuan yang mengarahkan Anda pada *forum* komunitas *XAMPP*.
7. *Quit* : Tombol untuk menutup *software XAMPP*.