



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Hamacher dalam (Daliuwa dan Didipu, 2020:137), “definisi komputer adalah sebuah mesin penghitung elektronik yang bisa dengan cepat dan tepat dapat menerima informasi input digital, kemudian memprosesnya dan menghasilkan *output* berupa informasi.”.

Sementara itu, Menurut Blissmer dalam (Daliuwa dan Didipu, 2020:137), “Pengertian komputer adalah suatu alat elektronik yang mampu untuk melakukan beberapa tugas, yaitu dapat menginput data, memproses data sesuai dengan instruksi yang diinginkan, penyimpanan perintah serta hasil pengolahannya dan juga menyediakan *output* dalam bentuk informasi, maupun *hardcopy*”.

2.1.2 Pengertian GPS (*Global Positioning system*)

Menurut Eddy Prahasta (2020) "*Global Positioning System* (GPS) adalah sistem navigasi satelit yang digunakan untuk menentukan posisi di mana pun di bumi, dengan menggunakan pengirim dan penerima sinyal. GPS terdiri dari 24 satelit yang mengorbit bumi, masing-masing akan mengirimkan sinyal digital yang dikenal dengan kode *pseudo-random*. Penggunaan GPS telah menjadi sangat umum di berbagai bidang, seperti navigasi kendaraan, pelacakan kendaraan dan aset, kartografi, geodesi (pengukuran bentuk bumi), dan lain sebagainya".

GPS (*Global Positioning System*) adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS menggunakan sistem konstelasi satelit yang memancarkan koordinat



keberadaan posisi satelit. *Signal* ditangkap oleh perangkat navigasi, dan diolah menjadi posisi koordinat di bumi. *Signal* GPS dapat dimanfaatkan sebagai:

- a. Penunjuk posisi dimana perangkat berada.
- b. Membuat rute jalan.
- c. Sebagai navigasi atau penunjuk arah jalan
- d. Mencatat dan menghitung perkiraan jarak dan waktu. Seperti jarak tempuh, waktu tempuh, rute jalan alternatif dan lainnya.

2.1.3 Pengertian Perangkat Lunak (*Software*)

Menurut Pressman dalam (Silaen, dkk 2022:253), “Pengertian *Software* (Perangkat lunak) merupakan sebuah perintah program dalam sebuah komputer yang apabila dieksekusi oleh user-nya akan memberikan fungsi dan unjuk kerja seperti yang diharapkan oleh user-nya”.

Menurut Utami & Asnawati dalam (Silaen, dkk 2022:254), “Perangkat lunak (*software*) adalah program yang berisi kumpulan intruksi untuk melakukan proses pengolahan data. *Software* sebagai penghubung antara manusia sebagai pengguna dengan perangkat keras komputer, berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga perangkat keras komputer memahami keinginan pengguna dan menjalankan instruksi yang diberikan serta selanjutnya memberikan hasil yang diinginkan oleh manusia tersebut”.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Harip Santoso dalam (Tri, 2020) “bahwa aplikasi adalah suatu kelompok *file* (*form, class, rePort*) yang memiliki tujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi *payroll*, aplikasi *fixed asset*, dan sebagainya. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Dapat disimpulkan dari kedua pendapat ahli tersebut, bahwa aplikasi adalah sebuah perangkat lunak yang memiliki fungsi untuk mengelola dokumen, contohnya aplikasi *payroll* dan *fixed asset*.”



2.2.2 Pengertian Absensi

Absensi adalah ketidak hadirannya karyawan saat yang bersangkutan dijadwalkan bekerja. Jumlah absensi kerja dalam perusahaan menggambarkan pertukaran *benefit* antara karyawan dan perusahaan yang menggaji. Tentu saja, perusahaan tidak ingin rugi akibat tingginya absen karyawan.

Menurut (Juhartini, 2020) “Absensi merupakan daftar hadir sekumpulan orang dari suatu kelompok orang banyak yang tergabung dalam sebuah instansi secara resmi yang mempunyai peraturan-peraturan, ketentuan-ketentuan, serta batasan-batasan, dan orang-orang yang terlibat di dalamnya terikat oleh peraturan tersebut. Jika sekelompok orang tersebut melanggarnya maka akan dikenakan sanksi sebagai hukuman dari pelanggaran yang dilakukan oleh orang tersebut sesuai dengan peraturan dan ketentuan yang diterapkan oleh instansi tersebut”.

2.2.3 Pengertian Geofencing

Geofencing merupakan fitur yang umumnya hadir pada aplikasi mobile yang memanfaatkan kinerja *global positioning system* (GPS) atau *radio frequency identification* (RFID) untuk memberikan batasan geografis *virtual*. Secara umum, sistem ini melakukan pendeteksian lokasi terhadap objek bergerak seperti manusia, kendaraan, kontainer menggunakan GPS, memberikan notifikasi apabila terdapat objek yang memasuki atau keluar dari wilayah yang telah ditentukan sistem. Budiyanto, G, (2020).

Geofencing adalah fitur dari sebuah software atau program yang memanfaatkan komponen “*global positioning system* (GPS)” atau “*radio frequency identification* (RFID)” guna menentukan batasan geografi secara *virtual* (Prima Dwiyan Nugraha et al., 2020).

Berdasarkan kedua definisi tersebut, dapat disimpulkan bahwa: *Geofencing* adalah sebuah teknologi atau fitur yang umumnya digunakan pada aplikasi *mobile*. teknologi yang menggabungkan kemampuan *positioning* dengan pembatasan wilayah *virtual* untuk memungkinkan deteksi lokasi dan aktivasi tindakan otomatis secara *real-time* berdasarkan pergerakan objek di suatu area yang telah ditetapkan.



Geofencing menggambarkan sebuah area (*geofence*) yang memiliki batas-batas geografis dari suatu peta. *Geofencing* pada umumnya dapat dimanfaatkan untuk membantu melacak pengiriman barang yang dibawa oleh suatu kendaraan, memantau posisi seseorang, menjalankan bisnis komersial tertentu, dan absensi otomatis di suatu perusahaan atau universitas. Ukuran wilayah dari *geofencing* yaitu berkisar dari beberapa meter sampai beberapa kilometer. Bentuk area sebuah *geofence* yaitu berbentuk sebuah lingkaran (*circle*) sedangkan mekanisme menentukan area ditentukan oleh *latitude*, *longitude*, dan radius dari titik yang ditentukan.

2.2.4 Pengertian Teknologi

Menurut Hamalik dalam Makki (2020), teknologi merupakan produk budaya sekaligus bagian dari budaya. Dapat juga dikatakan bahwa inovasi adalah bagian material dari budaya. Masyarakat mengandalkan teknologi sebagai sarana untuk memenuhi kebutuhannya. Sejak ratusan tahun yang lalu, manusia telah menciptakan dan menemukan berbagai teknologi baru melalui penemuan dan penemuan, yang kemudian mereka gunakan untuk menjalani kehidupannya.

Definisi teknologi dapat dipandang sebagai kegiatan yang membentuk atau mengubah kebudayaan. Selain itu, teknologi adalah terapan matematika, sains, dan berbagai seni untuk faedah kehidupan seperti yang dikenal saat ini. Tetapi ada juga definisi yang sama menonjolnya, yakni definisi teknologi sebagai sains terapan, khususnya para ilmuwan dan insinyur. Dalam penggunaan ini, pengertian teknologi mengacu pada alat dan mesin yang dapat digunakan untuk memecahkan masalah di dunia nyata. Kata "teknologi" juga dapat digunakan untuk merujuk kepada kumpulan teknik. Dalam konteks ini, itu adalah keadaan saat ini dari pengetahuan manusia tentang bagaimana menggabungkan sumber daya untuk memproduksi produk yang diinginkan, untuk memecahkan masalah, memenuhi kebutuhan, atau memuaskan keinginan. Teknologi seringkali merupakan konsekuensi dari ilmu dan rekayasa, meskipun kegiatan manusia justru mendahului kedua ranah tersebut.



2.2.5 Pengertian Aplikasi Absensi Kayrawan Besbasis Web dengan Menggunakan Metode Geofencing pada CV Niara Brother Express

Aplikasi Absensi Karyawan Berbasis Web dengan Menggunakan Metode Geofencing pada CV Niara Brother Express adalah sebuah aplikasi yang dapat memaksimalkan Absensi bagi para karyawan CV Niara Brother Express sebagai solusi digital yang memungkinkan karyawan untuk melakukan absensi secara *online* melalui *website*, dengan memanfaatkan teknologi *geofencing* untuk memverifikasi lokasi karyawan saat melakukan absensi, sehingga meningkatkan keakuratan dan integritas data absensi serta meminimalisir tingkat kecurangan dalam absensi.

2.3 Teori Khusus

2.3.1 Pengertian *Unified Modeling Language (UML)*

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:

2.3.1.1 *Use Case Diagram*

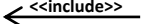
Use case diagram dipergunakan untuk menggambarkan suatu interaksi yang terdapat pada sistem informasi yang menggambarkan satu atau lebih aktor. Menurut (Suhimarita J, dan Susianto D,. 2019) *use case diagram* digunakan untuk mengetahui siapa saja yang dapat menggunakan fitur-fitur yang ada pada sistem 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.

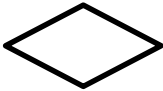
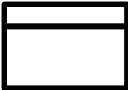
Tabel 2.1 simbol-simbol *Use Case Diagram*

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.

2.3.1.2 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 memiliki 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.
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

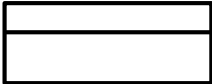
2.3.1.3 Class Diagram

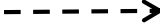
Diagram kelas digunakan untuk menggambarkan struktur sistem dan mendefinisikan setiap kelas yang digunakan untuk membangun sistem, yang disebut atribut dan metode. Menurut (Suhimarita J, dan Susianto D., 2019) *Class* sendiri memiliki apa yang dikatakan dengan *atribute* dan metode.

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 tedapat 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. Simbol-simbol *class diagram* pada tabel 2.3 berikut ini.

Tabel 2.3 Simbol-Simbol Clas Diagram

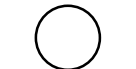
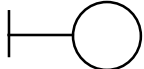
Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada diatasnya ibjek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
Simbol	Nama	Keterangan

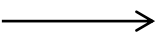
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.3.1.4 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 dari form
Simbol	Nama	Keterangan

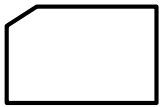
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>message</i> .
	<i>A message</i>	Menggambarkan pengiriman pesan.

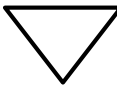
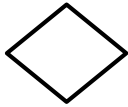
2.3.2 Pengertian *Flowchart*

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 perlu dipelajari 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*

Simbol	Nama	Keterangan
	Dokumen	Dokumen atau laporan, dapat berupa hasil tulisan tangan atau cetakan komputer.
	<i>input/output</i> ;	Proses <i>input</i> atau <i>output</i> data, epresentasi jurnal atau bukubesar.
	<i>redefinedProcess</i>	Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
Simbol	Nama	Keterangan

	Proses	Pelaksanaan pemrosesan yang dilakukan oleh komputer.
	Proses Manual	Pelaksanaan pemrosesan yang dilakukan secara manual.
	Disk Storage	Data disimpan secara online melalui media yang dapat diakses secara langsung.
	Off-lineStorage	File dokumen yang disimpan secara manual.
	Arus dokumen / arus pemrosesan	Arus pemrosesan atau arus dokumen, arus yang normal berada dibawah dan mengarah.
	Communication link	Pengiriman data dari satu lokasi ke lokasi lainnya melalui jalur komunikasi.
	Off-page Connector	Menggambarkan keluar atau masuk proses dalam lembar atau halaman yang lain.
	On-pageconnector	Menghubungkan arus pemrosesan di halaman yang sama.
	Decision	Menyatakan kondisi yang menghasilkan beberapa kemungkinan atau aksi.
	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 Sukamto 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

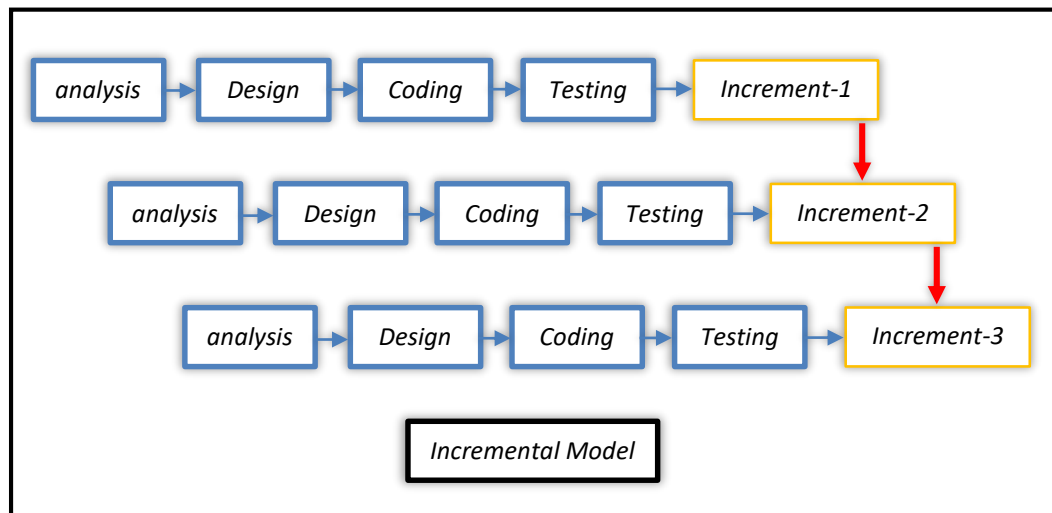
No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang/ bernilai banyak
5.	()	data opsional
6.	*...*	batas komentar

2.3.4 Metode Pengembangan Sistem

Dalam penelitian Laporan Akhir ini, penulis menggunakan metode pengembangan sistem Model Incremental. Model Incremental adalah metode yang menetapkan prosedur di mana perhatian prioritas diberikan pada persyaratan sistem yang diikuti dengan implementasinya dalam tim pengembangan, proses ini digambarkan sebagai inkremental (Rokoyah et al., 2022).

Secara umum, seiring berjalannya waktu, peningkatan semakin kecil, dan setiap kebutuhan yang abadi akhirnya terpenuhi. Setiap versi baru dari sistem mencakup penambahan fungsionalitas yang telah ditetapkan, yang berlanjut hingga semua fitur yang direncanakan telah dimasukkan. Dalam skenario ini, dimungkinkan untuk memangkas biaya, memantau dampak perubahan persyaratan, dan mempercepat pembangunan sistem

yang berfungsi dengan memanfaatkan metodologi tata letak blok standar. Berikut ini adalah daftar langkah-langkah yang membentuk model Incremental, yang dapat ditemukan di bawah ini pada **Gambar 2.1**.



Gambar 2.1 Metode Model Incremental

Manfaat menggunakan model inkremental adalah sebagai berikut:

1. Tidak perlu menunggu sampai seluruh sistem terkirim untuk memanfaatkan sistem karena peningkatan pertama sudah memenuhi persyaratan paling kritis, sehingga perangkat lunak dapat langsung digunakan.
2. Karena kenaikan pertama sudah memenuhi persyaratan paling kritis, tidak perlu menunggu sampai seluruh sistem dikirim.
3. Dapat memanfaatkan peningkatan sistem awal sebagai prototipe dan mengumpulkan pengalaman, yang dapat digunakan untuk menentukan persyaratan untuk peningkatan sistem selanjutnya.
4. Menghasilkan perangkat lunak yang berfungsi dengan cepat dan lebih awal dalam siklus hidup perangkat lunak. Jadikan lebih fleksibel dan lebih murah untuk menyesuaikan ruang lingkup dan persyaratan. Ini akan menghasilkan pengurangan biaya pengiriman awal.



5. Pengujian dan pemecahan masalah model ini jauh lebih sederhana bila dilakukan dalam lebih sedikit iterasi. Setelah itu, pelanggan akan memiliki kesempatan untuk mengomentari setiap build.
6. Risiko dapat dengan mudah dikelola karena komponen individu dari risiko ditemukan dan ditangani selama proses berulang.
7. Mengurangi potensi kegagalan proyek secara keseluruhan
8. Layanan dengan kepentingan tertinggi adalah yang diberikan terlebih dahulu, dan peningkatan berikutnya terintegrasi dengannya. Sangatlah penting bahwa layanan sistem yang paling penting harus melalui pengujian yang paling menyeluruh.

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 *tag-tag* 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”

Berdasarkan 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 *CSS-Driven* 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 *programer*, 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-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, yaitu:

Tabel 2.7 pengertian *XAMPP*

X	<i>Cross Platform</i> Artinya, <i>software</i> ini kompatibel pada berbagai OS komputer seperti <i>Windows</i> , <i>Linux</i> , maupun <i>Mac OS</i> .
A	<i>Apache</i> merupakan suatu aplikasi web server <i>default</i> yang digunakan oleh <i>XAMPP</i> . Aplikasi ini bersifat <i>open source</i> jadi bisa digunakan secara gratis dan bisa dikembangkan secara bebas.
M	MySQL / MariaDB MySQL & MariaDB adalah aplikasi <i>database</i> server <i>default</i> yang ada di <i>XAMPP</i> . Program ini digunakan untuk mengelola <i>database</i> .
P	PHP merupakan bahasa pemrograman <i>back end</i> untuk mengembangkan aplikasi berbasis web.
P	Perl merupakan bahasa pemrograman yang mencakup banyak tugas, seperti: pengembangan web, pengembangan GUI, pemrograman jaringan, administrasi sisteml Program-program yang terintegrasi pada <i>XAMPP</i> secara tidak langsung bisa menggambarkan fungsi dan fitur dari <i>software</i> ini.



Gambar 2.9 Logo XAMPP

2.4.9.1 Mengetahui 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*.