



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

(Rianto, 2021), Komputer merupakan suatu perangkat elektronika yang memiliki kemampuan untuk menerima dan mengolah data menjadi informasi, menjalankan program yang tersimpan dalam memori, serta dapat bekerja secara otomatis berdasarkan perangkat aturan tertentu.

(Krisbiantoro, 2018), Komputer merupakan alat yang dipakai untuk mengolah dan memproses data menurut perintah yang telah dirumuskan. Komputer berasal dari kata *to compute* yang artinya menghitung. Jadi komputer adalah suatu peralatan elektronik yang digunakan untuk perhitungan.

(Iskandar, 2018), Komputer dapat didefinisikan sebagai satu peralatan elektronik yang terdiri dan beberapa komponen yang dapat bekerja sama antara komponen satu dengan yang lain untuk menghasilkan suatu informasi berdasarkan program dan data yang ada.

Berdasarkan pernyataan diatas dapat disimpulkan bahwa pengertian komputer adalah alat elektornik yang saling bekerja sama antara komponen satu dengan yang lain untuk menerima data, memproses data, mengelola data menjadi infromasi, dan menjalankan program secara otomatis sesuai dengan instruksi yang telah dirumuskan.

2.1.2 Pengertian Perangkat Lunak (*Software*)

(Rianto, 2021), “Perangkat Lunak (*Software*) merupakan suatu data yang telah di program sedemikian rupa dan disimpan dalam bentuk digital yang fisik data nya tidak terlihat namun tersimpan dalam media penyimpanan komputer”.

(Arora, 2020), Istilah dari Perangkat lunak adalah kumpulan instruksi atau perintah yang mengendalikan serta mengatur operasi komputer. Instruksi yang dilakukan pada komputer untuk melakukan tugas tertentu disebut program.



(Moschovitis, 2018), Definisi perangkat lunak adalah sekumpulan instruksi yang mengarahkan perangkat keras dalam melakukan suatu tugas atau aktivitas tertentu. Tidak ada yang terjadi tanpa perangkat lunak. Seperti halnya mobil dan sopir, mobil sebagai perangkat keras dan sopir sebagai perangkat lunak.

Dari pernyataan diatas, didapat kesimpulan bahwa perangkat lunak (*Software*) adalah sekumpulan instruksi untuk mengendalikan dan melaksanakan aktivitas dan mengatur operasi suatu komputer.

2.1.3 Pengertian Basis Data (*Database*)

(Fatkhurrochman, Muin, & Priyoatmoko, 2021) menyatakan bahwa, “Basis data terdiri dua kata yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang yaitu tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, barang, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Menurut Harrington dalam (Fatkhurrochman, Muin, & Priyoatmoko, 2021) basis data adalah kumpulan data dan informasi yang saling berkaitan.

(Quahab & Adel, 2017), Basis Data atau dalam bahasa inggris *Database* adalah sekumpulan informasi yang saling terkait yang disimpan pada media penyimpanan yang dapat diakses oleh satu aplikasi atau lebih, tanpa duplikasi dan terstruktur secara independen untuk memenuhi kebutuhan pengguna.

Berdasarkan pernyataan yang telah ditemukan oleh pengarang di atas maka dapat ditarik kesimpulan bahwa basis data adalah sekumpulan fakta dari data atau informasi yang di simpan bahkan bisa di akses di aplikasi manapun dan kapanpun.

2.1.4 Pengertian Framework

(Yudhanto & Prasetyo, 2018) menyatakan bahwa, “*Framework* adalah kerangka kerja atau bisa diartikan sebagai kumpulan *script* yang terdiri dari *class* dan *function* yang dapat membantu *programmer* dalam menangani berbagai masalah pemrograman



seperti koneksi *database* sehingga programmer bisa lebih fokus dan cepat dalam merancang aplikasi. Contoh *Framework* PHP adalah Laravel, cakePHP, *Codeigniter*, dan lainnya.

(Supardi & Hermawan, 2018), *Framework* adalah sekumpulan fungsi/prosedur dan kelas yang memiliki tujuan tertentu yang akan digunakan sehingga bisa lebih mudah dan cepat dalam membangun aplikasi, tanpa harus membuat dari awal.

Berdasarkan pernyataan di atas *Framework* adalah kumpulan teks yang berisikan kelas dan fungsi untuk membangun suatu aplikasi.

2.2 Teori Judul

2.2.1 Pengertian Sistem Pakar

(Arisandi & Sari, 2021), Sistem pakar merupakan kecerdasan buatan di bidang ilmu komputer dan teknologi yang membuat komputer lebih efektif dan efisien sehingga bisa melakukan aktivitas kecerdasan seperti manusia pada umumnya. Kecerdasan buatan adalah salah satu perkembangan komputer dalam *software*.

(Swetapadma & Sarraf, 2018), Sistem pakar adalah sistem yang dirancang untuk memecahkan persoalan yang kompleks dengan penalaran melalui bidang pengetahuan, yang direpresentasikan sebagai *IF-THEN rules*.

(Hayadi, 2018), Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* yaitu suatu sistem aplikasi komputer yang digunakan dalam membantu pengambilan keputusan atau pemecahan permasalahan di bidang yang lebih spesifik. Sistem ini berjalan menggunakan pengetahuan dan metode analisis yang telah diartikan terlebih dahulu oleh ahli pakar yang sesuai dengan bidang keahliannya.

Dari pernyataan di atas maka penulis menarik kesimpulan bahwa sistem pakar merupakan kecerdasan buatan yang digunakan untuk menyelesaikan suatu masalah dalam segala bidang.



2.2.2 Pengertian *Computer Vision Syndrome* (CVS)

Menurut (Dotulong et.al, 2021) dalam penelitian berjudul “*Computer Vision Syndrome*” menyatakan bahwa Sindrom Penglihatan Komputer atau dalam bahasa Inggris *Computer Vision Syndrome* merupakan sekumpulan dari keluhan pada mata yang diakibatkan oleh penggunaan layar digital terkhusus komputer dalam jangka waktu yang lama yang menyebabkan mata lelah, nyeri kepala dan gejala bervariasi lainnya dalam penelitian.

Menurut Forster dalam (Baskaran, dkk, 2020), *Computer Vision Syndrome* (CVS) adalah sindrom atau penyakit dari salah satu dampak dari menghabiskan lebih banyak waktu dengan layar digital yang mengakibatkan peningkatan ketegangan mata dan masalah penglihatan.

(Sulianta, 2018), *Computer Vision Syndrome* (CVS) adalah permasalahan pada mata akibat kelelahan dan ketegangan yang disebabkan karena penggunaan komputer dalam jangka waktu lama sehingga mata terus dipaksa menatap layar monitor. Gejala-gejala yang diderita sindrom ini antara lain mata terasa kering dan gatal, mata menjadi merah dan berair, kehilangan fokus, sakit kepala, nyeri pundak, bahkan otot mata menjadi kejang.

Dari pernyataan di atas maka penulis menarik kesimpulan bahwa *Computer Vision Syndrome* merupakan sindrom akibat kelelahan dan ketegangan pada bagian mata yang disebabkan karena penggunaan komputer atau layar digital lainnya dalam jangka waktu lama.

2.2.3 Pengertian *Forward Chaining*

(Ramadhan & Pane, 2018) “*Forward chaining* adalah teknik pencarian yang diawali dengan fakta - fakta yang telah diketahui kemudian dicocokkan dengan bagian *IF* dari rule *IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam *database*.”



(Hayadi, 2018), *Inferensi Forward Chaining* adalah mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar dimana di mulai dari sekumpulan data yang bersifat fakta menuju kesimpulan. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik dan memulai pelacakannya dengan mencocokkan kaidah.

Menurut pendapat (Egasari, dkk , 2017) berdasarkan penelitian yang berjudul “Sistem Pakar Identifikasi Kesesuaian Lahan untuk Tanaman Perkebunan di Provinsi Bengkulu dengan Metode *Bayes* dan *Inferensi Forward Chaining*” pengertian dari *Inferensi Forward Chaining* adalah suatu pengambilan keputusan yang paling sering digunakan dalam sistem pakar dengan menggunakan proses pencarian pernyataan kesimpulan akhir dari data gejala yang disediakan.

Dari pernyataan di atas maka penulis menarik kesimpulan bahwa *Inferensi Forward Chaining* merupakan metode sistem pakar yang mencari solusi melalui masalah yang mempertimbangkan melalui fakta yang ada dan menarik kesimpulan.

2.2.4 Pengertian *Naive Bayes*

(Fadila, Rahayu, & Saputra, 2020), “*Naive Bayes* adalah pengklasifikasi jaringan Bayesian paling sederhana. Dalam *Naive Bayes*, setiap node fitur memiliki simpul kelas sebagai induknya, tetapi tidak memiliki orangtua dari node fitur lainnya. Namun pengelompokan *naive bayes* dapat bersaing dengan pengklasifikasi canggih seperti C4.5 dan masih merupakan salah satu dari 10 algoritma penambangan data teratas di dunia karena masih banyak menggunakannya”.

(Sidiq, Fatimah, & Riza, 2020), *Naive Bayes Classifier* merupakan sebuah pengklasifikasian probabilistik yang sederhana untuk memperkirakan sekumpulan kemungkinan dengan menampilkan jumlah frekuensi dan kombinasi dari *dataset* yang telah ada. Definisi lain mengatakan bahwa apabila diberikan nilai *output*, probabilitas mengamati secara bersama adalah produk dari probabilitas individu.

Berdasarkan penelitian yang dilakukan oleh (Mudjahid, et.la , 2020) berjudul “Sistem Pakar Berbasis Web untuk Mendiagnosis Penyakit Mata Manusia



menggunakan Metode *Naive Bayes*” bahwa Metode *Naive Bayes* merupakan salah satu metode yang paling sederhana untuk melakukan klasifikasian kemungkinan berdasarkan *Teorema Bayes* yang artinya atribut yang independen.

Dari pernyataan di atas maka penulis menarik kesimpulan bahwa *Naive Bayes* merupakan metode pengklasifikasian yang mengandalkan probabilitas peluang dari data yang telah ada.

2.2.5 Pengertian WEB / Website

(Sudarto, Kristiadi, & Hidayat, 2021), *Web* merupakan kumpulan laman yang berisi informasi yang dapat diambil melalui media internet berupa tulisan, gambar suara, video yang memiliki alamat pada browser.

Menurut Abdulloh dalam (Sa'ad, 2020), *website* atau *web* adalah kumpulan beberapa halaman yang terdiri dari informasi berbentuk data digital, baik berupa teks, gambar, *video*, suara, dan animasi lainnya yang disediakan melalui jalur koneksi internet. Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang dapat diakses oleh *browser* dan mampu memberikan informasi yang berguna bagi para pengaksesnya.

Berdasarkan penjelasan di atas bahwa *website* adalah laman berisi informasi berupa teks, gambar, animasi, suara, dan *video* yang diakses melalui koneksi internet.

2.2.6 Pengertian Perbandingan metode *Inferensi Forward Chaining* dan metode *Naive Bayes* pada sistem pakar untuk mendiagnosa penyakit *computer vision syndrome* (CVS)

Perbandingan metode *Inferensi Forward Chaining* dan metode *Naive Bayes* pada sistem pakar untuk mendiagnosa penyakit *computer vision syndrome* (CVS) adalah sistem pakar yang dibuat dengan membandingkan dua metode untuk mengetahui antara metode *Inferensi Forward Chaining*, *Naive Bayes*, atau penerapan kedua metode yang lebih efektif dan akurat dalam melakukan tugasnya yaitu mendiagnosa dan menganalisis penyakit dari gejala yang dialami oleh pengguna layar digital.



2.3 Teori Khusus

2.3.1 Kamus Data

(Santi, 2020), menyatakan bahwa, Kamus data adalah kumpulan daftar elemen yang berisi data atau katalog fakta tentang data dari suatu sistem informasi. Kamus data biasanya dibuat berdasarkan arus data yang mengalir pada DAD (fisik dan logika) dimana di dalamnya terdapat struktur dari arus data secara lengkap.

Tabel 2.1 Simbol-simbol pada Kamus Data

No	Simbol	Keterangan
1.	=	Terbentuk dari atau terdiri dari
2.	+	Dan
3.	[]	Memilih salah satu elemen data
4.	N{M}	N kali/ bernilai banyak berulang
5.	()	Data opsional
6.	*...*	Batas komentar
7.		Pemisah sejumlah alternatif
8.	@	Identifikasi atribut kunci

Sumber : (Maniah & Hamidin, 2017)

2.3.2 Pengertian UML (Unified Modeling Language)



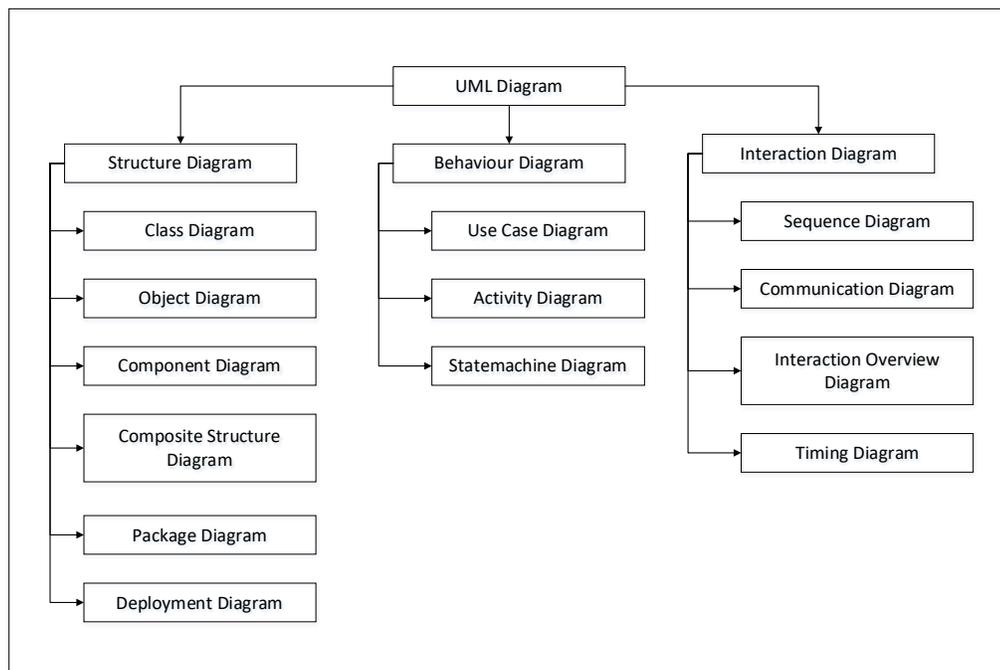
Sumber : https://id.wikipedia.org/wiki/Unified_Modeling_Language

Gambar 2.1 Logo UML



(Habibi, Putra, & Putri, 2020), “*Unified Modeling Language (UML)* adalah sebuah model yang telah menjadi standarisasi untuk memvisualisasi, perancangan dan pendokumentasian sistem dari perangkat lunak. UML dapat dipergunakan untuk merancang atau modeling aplikasi prosedural dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

(Negara, et al., 2021), *Unified Modeling Language (UML)* merupakan bahasa pemrograman visual untuk pemodelan mengenai sebuah sistem perangkat lunak dengan menggunakan diagram dan teks pendukung lainnya. Berikut pembagian kategori dan diagram pada UML :



Sumber: Menurut Hendini dalam (Negara, et al., 2021)

Gambar 2.2 Diagram UML



Berikut ini beberapa penjelasan dari diagram yang sering digunakan dalam pengembangan suatu sistem yaitu :

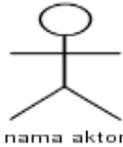
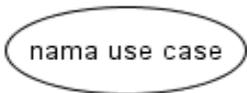
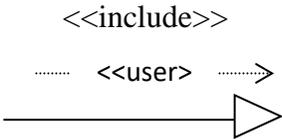
2.3.2.1. Pengertian Use Case Diagram

(Mulyani, 2016), mengatakan bahwa *Use Case Diagram* merupakan sekumpulan diagram atau teks yang saling berhubungan untuk mendokumentasikan cara pengguna (*actor*) berinteraksi dengan sistem.

Menurut (Rusmawan, 2019), *Use Case Diagram* adalah teknik diagram yang digunakan untuk merekam persyaratan fungsional seluruh sistem yang ada. *Use Case Diagram* juga dapat menjelaskan interaksi antara para pengguna sistem dengan sebuah narasi sistem itu sendiri.

Jadi, dapat disimpulkan bahwa pengertian dari *Use Case Diagram* adalah kumpulan dari teks dan diagram yang menjelaskan bagaimana sistem yang digunakan.

Tabel 2.2 Simbol-simbol pada *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang atau unit yang berhubungan dengan sistem yang dibangun.
2.		<i>Use Case</i>	Mendeskripsikan cara orang atau unit menggunakan sebuah sistem.
3.		Relasi Asosiasi	Komunikasi yang digunakan untuk menunjukkan hubungan antara aktor (<i>actor</i>) dan <i>Use Case</i> .
4.		Relasi <i>Include</i>	Kemungkinan satu <i>use case</i> menunjukkan fungsionalitas yang disiapkan oleh <i>use case</i> lain.



5.	ekstensi / <i>extend</i> 	Relasi <i>Extend</i>	Menunjukkan suatu <i>use case</i> secara <i>optional</i> menggunakan fungsional yang disiapkan oleh <i>use case</i> lain.
----	---	----------------------	---

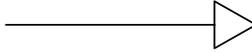
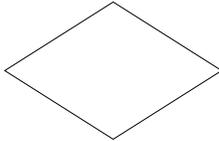
Sumber : (Rusmawan, 2019)

2.3.2.2. Pengertian *Class Diagram*

(Haqi & Setiawan, 2019), berpendapat bahwa *Class Diagram* adalah salah satu model diagram pada UML yang dipergunakan untuk menunjukkan kelas-kelas maupun paket-paket yang terdapat pada suatu sistem yang nantinya akan digunakan.

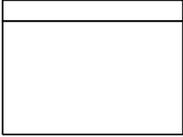
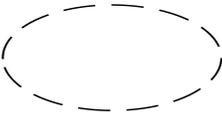
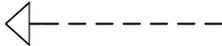
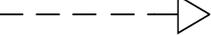
(Alda, 2021), berpendapat bahwa *Class Diagram* merupakan suatu model diagram yang selalu ada di sistem berorientasi objek, apabila spesifikasi yang ada diinstansiasikan maka akan menghasilkan sebuah objek ini dari pengembangan dan desain dari objek tersebut. Jadi *Class Diagram* menunjukkan hubungan antar *class* dalam sebuah sistem yang sedang dibangun dan saling bekerja sama untuk mencapai suatu tujuan tertentu.

Tabel 2.3 Simbol-simbol pada *Class Diagram*

No	Nama	Simbol	Keterangan
1.	<i>Generalization</i>		Hubungan objek anak (<i>descendent</i>) antara perilaku dan struktur data dari objek yang disediakan objek induk (<i>ancestor</i>).
2.	<i>Nary Association</i>		Cara untuk menghindari asosiasi terhadap atribut dan sistem yang terduplikasi.



Lanjutan Tabel 2.3 Simbol-simbol pada *Class Diagram*

No	No	Simbol	Keterangan
3.	<i>Class</i>		Kumpulan dari objek-objek yang terbagi antara atribut dan sistem yang sama.
4.	<i>Collaboration</i>		Menjelaskan urutan dari <i>action</i> yang ditunjukkan sistem menghasilkan <i>result</i> yang terstruktur bagi suatu <i>actor</i>
5.	<i>Realization</i>		Operasi yang dilakukan oleh suatu objek.
6.	<i>Dependency</i>		Relasi dimana perubahan yang terjadi pada elemen mandiri (<i>independent</i>) yang mempengaruhi elemen lain yang tidak mandiri.
7.	<i>Association</i>		Menghubungkan suatu objek.

Sumber: (Alda, 2021)

2.3.2.3. Pengertian Activity Diagram

(Alda, 2021), mengemukakan bahwa *activity diagram* adalah pemodelan fungsi suatu sistem pada aliran kendali antar objek yang bersifat dinamis dan juga tipe dari diagram yang memperlihatkan aliran dari antar aktivitas dalam suatu sistem.

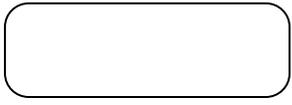
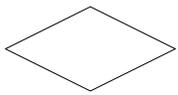


(Suryansah & Habibi, 2020) mengatakan bahwa pengertian dari *Activity Diagram* adalah suatu teknik untuk mendeskripsikan logika yang bersifat terstruktur, atau *workflow* pada proses bisnis dengan kasus yang beanekaragam permasalahan.

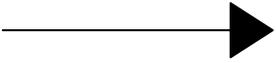
Menurut (Simatupang & Sianturi, 2019) dalam penelitian berjudul “Perancangan Sistem Informasi Pemesanan Tiket Bus pada PO Handoyo Berbasis *Online*” mengatakan bahwa *activity diagram* adalah diagram yang menggambarkan keseluruhan aktivitas suatu sistem di dalam aplikasi yang dilakukan oleh sistem bukan aktor.

Jadi, dapat ditarik kesimpulan bahwa pengertian dari *Activity Diagram* adalah gambaran aluran aktifitas yang akan dilakukan pada satu proses sistem. Adapun simbol-simbol yang sering digunakan dalam *activity diagram* dapat dilihat sebagai berikut ini:

Tabel 2.4 Simbol-simbol pada *activity diagram*

No	Nama	Simbol	Keterangan
1.	<i>Start State</i>		Titik Awal Proses
2.	<i>End State</i>		Titik Akhir Proses
3.	<i>Activity</i>		Aktivitas yang dilakukan aktor
4.	<i>Decision</i>		Pemilihan keputusan

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

No	No	Simbol	Keterangan
5.	<i>Interaction</i>		Alur

Sumber: (Rusmawan, 2019)

2.3.2.4. Pengertian *Sequence Diagram*

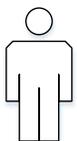
(Alda, 2021), *Sequence Diagram* atau yang bisa disebut Diagram Interaksi adalah gambaran suatu diagram yang menekankan pada pengiriman *message* dalam jangka waktu tertentu, maka dari itulah diagram interaksi ini bersifat dinamis. Tujuan utama dari *Sequence Diagram* adalah mendefinisikan urutan kejadian sehingga menghasilkan *output* yang diharapkan *user*.

(Haqi & Setiawan, 2019), mengatakan bahwa pengertian dari *Sequence Diagram* adalah salah satu tipe diagram pada UML yang menjelaskan interaksi proses sistem berdasarkan urutan waktu.

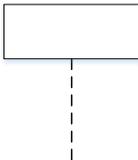
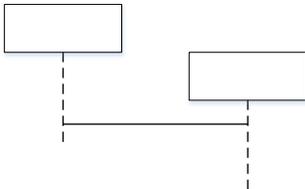
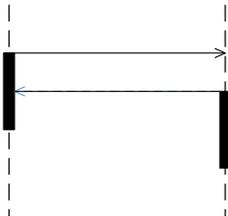
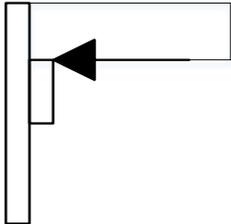
Jadi, dapat disimpulkan bahwa pengertian dari *Sequence Diagram* adalah gambaran diagram yang menjelaskan proses sistem pada waktu pengiriman.

Adapun simbol-simbol yang sering digunakan dalam *Sequence Diagram* sebagai berikut :

Tabel 2.5 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang atau unit yang melakukan kelola sistem

Lanjutan Tabel 2.5 Simbol-simbol pada *Sequence Diagram*

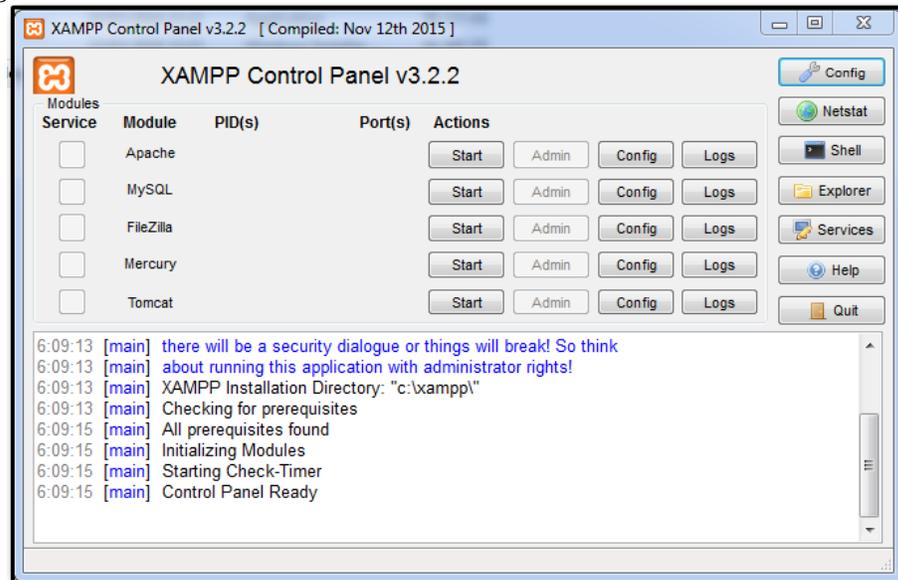
No	Simbol	Nama	Keterangan
2.		<i>Lifeline</i>	Mendeskripsikan suatu objek dalam sistem atau komponen
3.		<i>Create Message</i>	Digunakan untuk pembuatan sebuah pesan sederhana antar elemen dan melakukan komunikasi antar objek
4.		<i>Synchronous Message</i>	Digunakan untuk mengaktifkan pesan sebuah proses sampai selesai, lalu bisa mengirimkan pesan baru.
5.		<i>Message to Self</i>	Hasil yang dikembalikan oleh sebuah sistem dan menuju objek itu sendiri.

Sumber: (Rusmawan, 2019)



2.4 Teori Program

2.4.1 Pengertian XAMPP



Sumber : (Ayu & Permatasari, 2018)

Gambar 2.3 Halaman Utama Aplikasi XAMPP

(Sidik, 2020), mengemukakan pendapat bahwa XAMPP merupakan seperangkat paket *software* yang terdiri dari beberapa *server web* (*Apache*), *database* (*MySQL – MariaDB*), dan pengembangan program aplikasi (*PHP* dan *Perl*).

(Enterprise, 2019), XAMPP merupakan sebuah server yang menjadi terfavorit untuk *programmer PHP* pemula, dari kelengkapan fitur dan mudah digunakan karena yang perlu anda lakukan hanyalah mengunduh, menginstal, dan menjalankan salah satu *module* bernama *Apache* yang dapat melakukan proses *PHP*.

(Yudhanto & Prasetyo, 2018) mengatakan bahwa XAMPP adalah kumpulan atau himpunan program aplikasi gratis yang banyak digunakan oleh kalangan *developer/programmer* dalam hal pengembangan *website* berbasis *PHP* dan *MySQL*.

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa XAMPP adalah sebagai program aplikasi untuk mengkoneksikan *website* pada pemrograman *PHP* dan *MySQL*.



2.4.2 Pengertian PHP



Sumber : (Jannah, dkk, 2019)

Gambar 2.4 Logo PHP

Menurut (Enterprise, 2019), berpendapat bahwa PHP adalah bahasa pemrograman bersifat dinamis dan interaktif yang dipergunakan untuk membuat aplikasi berbasis *web* akan dijalankan menggunakan *web browser*.

Menurut (Yudhanto & Prasetyo, 2018), PHP atau biasa disebut *Hypertext Preprocessor* merupakan Bahasa pemrograman *script* dalam dokumen HTML yang dirancang lebih cenderung untuk mengembangkan *web* agar dapat membuat suatu laman yang dinamis.

Maka dapat disimpulkan bahwa PHP merupakan bahasa pemrograman yang terpasang pada HTML yang dirancang bertujuan untuk membuat *web* yang dinamis.

2.4.3 Pengertian MySQL



Sumber : (Ramadhan & Mukhaiyar, 2020)

Gambar 2.5 MySQL



(Fitri, 2020), MySQL adalah *Database Management System* bersifat *open source* dengan dua bentuk lisensi *Free Software* dan *Shareware* sehingga bisa digunakan untuk keperluan pribadi maupun komersial tanpa harus membayar lisensi yang ada.

(Rusmawan, 2019), MySQL adalah sebuah perangkat lunak (*software*) sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang bisa melakukan lebih dari satu proses dengan sekitar 6 juta pengguna di seluruh dunia.

Dari uraian diatas, dapat disimpulkan bahwa MySQL merupakan salah satu *software* pengelola manajemen data yang bersifat *open source* yang berguna untuk mengolah *database* serta membangun aplikasi *web* dengan basis data sebagai sumber pengelolaan datanya.

2.4.4 Pengertian *phpMyAdmin*



Sumber : (Ramadhan & Mukhaiyar, 2020)

Gambar 2.6 Logo phpMyAdmin

Menurut (Junus, 2020) bahwa “phpMyAdmin adalah perangkat aplikasi berbasis *web Graphical User Interface* (GUI) yang paling banyak digunakan dengan DBMS MySQL.

(Yudhanto & Prasetyo, 2018) berpendapat bahwa phpMyAdmin adalah aplikasi *web* bahasa pemrograman PHP yang digunakan untuk mengelola *database* MySQL dan MariaDB dengan sederhana dan mudah melalui antarmuka grafis. Aplikasi ini bisa dijalankan dan diakses dengan *browser*.



Jadi dapat disimpulkan bahwa phpMyAdmin adalah aplikasi *web* yang digunakan untuk mengelola *database*.

2.4.5 Pengertian Codeigniter



Sumber : (Andarsyah & A, 2019)

Gambar 2.7 Codeigniter

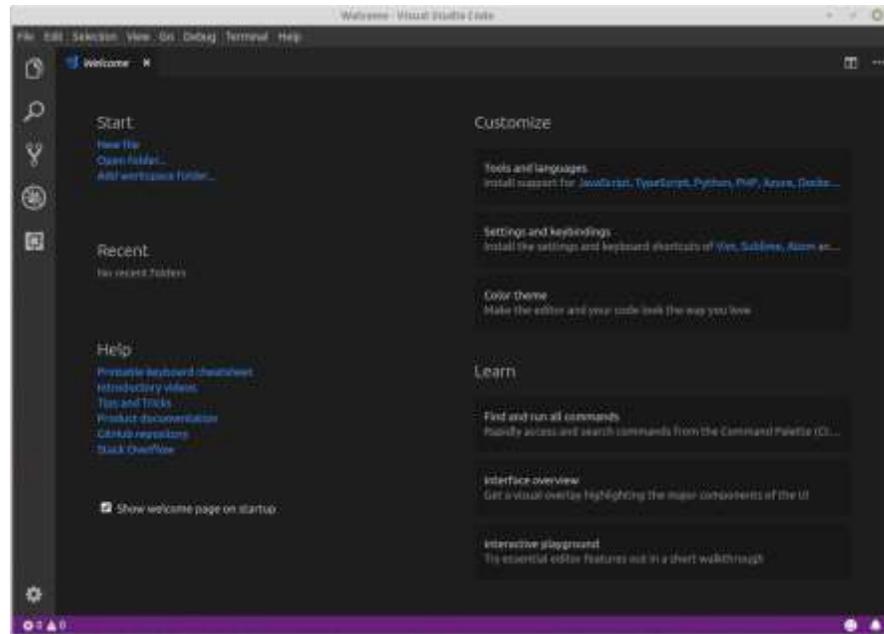
(Setyawati, dkk, 2020), berpendapat bahwa Codeigniter adalah suatu *framework* yang menggunakan bahasa pemrograman PHP dengan kerangka kerja dari prosedur untuk memudahkan programmer dalam membangun aplikasi. Codeigniter memiliki cara kerja pada laman berupa model MVC yaitu *Model View Controller* secara dinamis, sehingga mempermudah dalam pengembangan,

(Supardi & Hermawan, 2018), Codeigniter adalah aplikasi *open source* berupa framework dengan model MVC yang diliris pertama kali pada 28 Februari 2006 oleh ElisLab. Konsep MVC (*Model, View, dan Controller*) merupakan suatu konsep yang digunakan codeigniter dalam membangun aplikasi *web*.

Jadi dapat disimpulkan bahwa codeigniter adalah kerangka PHP yang digunakan dalam pembuatan aplikasi website dimana memiliki konsep MVC (*Model, View, dan Controller*).



2.4.6 Pengertian *Visual Studio Code*



Sumber : (Andarsyah & A, 2019)

Gambar 2.8 Tampilan Visual Studio Code

(Salamah, 2021), berpendapat bahwa Visual Studio Code (VS Code) ini adalah sebuah program *editor* yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya terdapat juga berbagai versi seperti Linux, Mac, dan Windows. VS Code ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js dengan bantuan yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst). Teks *editor* VS Code juga bersifat *open source*, kode sumber dari VS Code ini pun dapat dilihat di *link* Github

Beberapa Fitur-fitur yang ada dalam Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks *editor*. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks *editor* yang lain.



2.4.7 Metode Pengembangan Sistem

2.4.7.1 Metode FAST (Framework for the Application of System Thinking)

Menurut (Warjiyono, dkk, 2020) dalam penelitian berjudul “Metode FAST dan Framework PIECES : Analisis dan Desain Sistem Informasi Penjualan Berbasis Website”, mengatakan bahwa definisi metode FAST (*Framework for the Applications of System Thinking*) adalah suatu metodologi pengembangan sistem aplikasi yang cepat dan berurutan berdasarkan tahap-tahap yang ada dimana tahapan itu sendiri dimulai dari tahap definisi lingkup, analisis masalah, analisis kebutuhan, desain logis, analisis keputusan, desain fisik & integrasi, kontruksi & pengujian, instalasi & pengiriman.

metode FAST (*Framework for the Applications of System Thinking*) sangat cocok digunakan untuk pengembangan suatu proyek yang membutuhkan proses lebih efektif dimana terjadinya kemungkinan perubahan selama pengembangan aplikasi.

2.4.7.2 Tujuan dari Metode FAST (Framework for the Application of System Thinking)

(Noorhansyah & Pratomo, 2016) Tujuan dari pengembangan menggunakan Metode FAST (*Framework for the Application of System Thinking*) untuk melakukan proses pengembangan sistem yang sangat formal dan akurat yang bisa mendeskripsikan himpunan aktivitas, metode, pengimplementasian terbaik, informasi, dan alat yang digunakan dalam pengembangan sistem dan pemeliharaan sistem dan aplikasi.

2.4.7.3 Kerangka FAST (Framework for the Application of System Thinking)

Menurut (Warjiyono, dkk, 2020) dalam penelitian (Warjiyono, dkk, 2020) yang berjudul “Metode FAST dan Framework PIECES : Analisis dan Desain Sistem Informasi Penjualan Berbasis Website”, mengatakan bahwa Metode pengembangan sistem menggunakan *Framework for the Application of System Thinking* (FAST) terdiri dari 3 tahapan yang dibagi kelagi menjadi 8 fase sebagai berikut:



- a. Tahapan Analisa & Perancangan adalah Tahap awal dalam pengembangan sistem, sebagai berikut :
 1. **Definisi Lingkup** yaitu mencari dan mengumpulkan informasi atau data yang berkaitan pada penelitian, permasalahan dan kebutuhan baru manajemen, serta ruang lingkup proyek dengan menggunakan kerangka PIECES (Performance, Information, Economics, Control, Efficiency, Service).
 2. **Analisis Permasalahan** yaitu pada tahap ini menjelaskan permasalahan sistem yang telah ada sebelumnya.
 3. **Analisis Kebutuhan**, tahapan ini melakukan pengurutan prioritas dari informasi dan masalah yang ada, diantaranya mengidentifikasi data yang diinginkan *user*.
 4. **Desain Logis** yaitu tahap transformasi data untuk mendeskripsikan kebutuhan pengguna misalnya struktur data, bisnis proses, alur data, dan *interface* menggunakan diagram UML.
- b. Tahapan Peralihan merupakan tahapan pengembangan sistem selanjutnya, yaitu:
 5. **Analisis Keputusan** yaitu tahap pengambilan keputusan untuk memilih software dan hardware dalam mengimplementasikan sistem.
- c. Tahapan Implementasi yaitu tahapan terakhir dalam pengembangan sistem menggunakan metode FAST, yaitu :
 6. **Desain Fisik** yaitu tahap mentransformasikan ke bentuk fisik yang akan dijadikan sebagai pengukuran dalam membuat pengembangan sistem.
 7. **Konstruksi & Pengujian** yaitu tahap untuk pengujian terhadap sistem. Tahap ini banyak melibatkan sumber daya seperti pengguna, waktu, dan biaya.
 8. **Instalasi & Pengiriman**, yaitu tahapan operasi sistem yang baru.