



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

Dalam sub-bab ini akan diterapkan mengenai landasan teori yang bertujuan untuk membahas permasalahan yang diambil. Sesuai judul dalam pembuatan Laporan Akhir, yang akan dibuat penulis adalah sebagai berikut :

##### **2.1.1 Pengertian Komputer**

Menurut Murhada dan Giap (2011:21), “Komputer adalah alat elektronik yang dapat menerima *Input* data dan mengolahnya menjadi suatu informasi, dengan menggunakan suatu program yang tersimpan di memorinya, serta dapat menyimpan program dan hasil pengolahannya, dimana bekerja secara otomatis”.

Menurut Sanders dalam Wahyudin dan Munir (2018:1), “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data *Input*, memprosesnya dan menghasilkan *Output* berdasarkan instruksi-instruksi yang telah tersimpan di dalam memori”.

##### **2.1.2 Pengertian Perangkat Lunak (*Software*)**

Murhada dan Giap (2011:95), “Perangkat lunak adalah program, dan program adalah sekumpulan instruksi yang disusun secara terstruktur oleh programmer”.

Sedangkan menurut Swara dan Yunes (2016:28), “Perangkat lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program maupun prosedur yang didalamnya merupakan kumpulan perintah yang dimengerti oleh komputer”.



### 2.1.3 Pengertian Perangkat Keras (*Hardware*)

Mushlihudin dan Oktafianto (2016:12), “Perangkat Keras yaitu komponen untuk melengkapi kegiatan memasukkan data, memproses data, dan keluaran data”.

Rianto (2021:4) menjelaskan, “*Hardware* (Perangkat Keras), merupakan kumpulan segala piranti atau komponen dari sebuah komputer yang sifatnya bisa dilihat secara kasat mata dan bisa diraba secara langsung”.

### 2.1.4 Metode Pengembangan Sistem

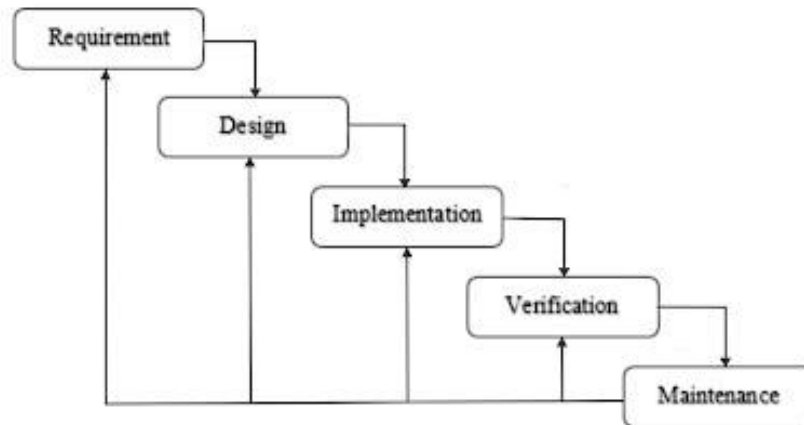
Dalam penelitian ini, penulis menggunakan metode pengembangan sistem waterfall. Rosa dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu waterfall. Menurut Pressman (dikutip Sanubari, et al., 2020:41), menyatakan bahwa "Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup klasik (classic life cycle), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (planning), permodelan (modeling), konstruksi (construction), serta penyerahan sistem ke para pelanggan/pengguna (deployment), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan."

Menurut Teknika: Jurnal Sains dan Teknologi (2020:269), menyatakan bahwa "Metode Waterfall adalah suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan teknologi dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian”.

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa Metode Waterfall adalah suatu proses atau metode pengembangan sistem secara berurutan yang mengalir ke bawah (seperti air terjun) melalui tahapan-tahapan Requirement Analysis, Design System, Implementation, Verification & Testing, dan Maintenance.



Tahapan metode waterfall dapat dilihat pada ilustrasi di bawah ini:



**Gambar 2.1** Ilustrasi Model Waterfall

Berdasarkan gambar di atas, pengembangan sistem dengan menggunakan metode waterfall memiliki beberapa tahapan yang berurut yaitu: Requirement (analisis kebutuhan), Design System (desain sistem), Coding (pengkodean) & Testing (pengujian), Penerapan Program, Pemeliharaan.

Tahapan-tahapan dari metode waterfall adalah sebagai berikut:

a. Requirement Analysis

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

b. System Design

Spesifikasi kebutuhan tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem dipersiapkan. Desain Sistem membantu dalam menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.



c. Implementation

Tahap ini dimana sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

d. Integration & Testing

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

e. Operation & Maintenance

Operation & Maintenance merupakan tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaiki implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

### 2.1.5 Metode Pengujian Aplikasi

Penulis melakukan pengujian aplikasi ini dengan menggunakan metode *Black Box Testing*. Menurut Sukamto (2016:275), “Black Box Testing yaitu menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau model secara offline dan online melalui publik, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan”.

Jaya (2018:45), “*Black-Box Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian Aplikasi dengan Metode ini memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program”.



Keuntungan penggunaan metode *Blackbox Testing* adalah:

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. Programmer dan tester keduanya saling bergantung satu sama lain.

### 2.1.6 Pengertian Basis Data (Database)

Abdulloh (2018:103), "Basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi".

Menurut Marlinda yang dikutip oleh Syaifudin, et al., dalam jurnal (Asmara,2016:84) bahwa : Sistem basis data adalah suatu sistem menyusun dan mengelola *record – record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan.

Setiap *database* mempunyai *Application Programming Interface* (API) tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalam sehingga dapat dimanfaatkan oleh aplikasi lainnya. Secara prinsip, dalam suatu database tercakup dua komponen penting, yaitu data dan informasi.

Menurut Sukanto dan Shalahuddin (2018:44), "DBMS atau sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, menampilkan data".

Sistem aplikasi disebut DBMS jika memenuhi persyaratan minimum :

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan.
4. Mampu menangani backup data.



### **2.1.7 Pengertian Internet**

Sudarsono (2021:69) menjelaskan, “*Interconnected network* atau yang lebih populer dengan sebutan internet adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia”.

Suwarya (2021:7) menjelaskan, ”Internet singkatan dari *Interconnected Network* merupakan sistem jaringan komputer yang saling terhubung secara global untuk menghubungkan perangkat di seluruh dunia”.

## **2.2 Teori Judul**

Dalam sub-bab ini akan diterapkan mengenai teori judul yang membahas judul yang diambil. Sesuai judul dalam pembuatan Laporan Akhir, yang akan dibuat penulis adalah sebagai berikut.

### **2.2.1 Pengertian Aplikasi**

(Kadir, 2017), “Perangkat lunak adalah kumpulan intruksi yang ditujukan kepada komputer. Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa intruksi-intruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi”.

Sedangkan menurut (Rahayu, 2019), Aplikasi merupakan perangkat lunak yang sangat mendukung manusia untuk mengerjakan tugas-tugasnya.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian aplikasi adalah suatu perangkat lunak/program yang berisi perintah-perintah untuk mengendalikan komputer. Aplikasi dibuat untuk memudahkan proses pengerjaan suatu pekerjaan tertentu.

### **2.2.2 Pengertian Data**

Menurut Indrajani (2018:2), “Data adalah fakta atau observasi mentah yang biasanya mengenai fenomena fisik atau transaksi data”.



Data menurut Jaluhu dan Trianovie (2020:2) adalah sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau pada lingkungan fisik sebelum diolah ke dalam format yang bisa dimengerti dan digunakan orang.

### **2.2.3 Pengertian Pemeriksaan**

(Santoso dan Andriyani, 2017), “Pemeriksaan dalam dunia kesehatan adalah tindakan untuk mengetahui sedini mungkin masalah kesehatan yang ada di dalam tubuh”. Sedangkan menurut (Arafah, et al., 2021), Pemeriksaan adalah sebuah proses dari seorang ahli medis memeriksa tubuh pasien untuk menemukantanda klinis penyakit. pemeriksaan akan dicatat dalam rekam medis. Rekam medis dan pemeriksaan akan membantu dalam penegakan diagnosis dan perencanaan perawatan pasien.

Dapat disimpulkan bahwa pengertian pemeriksaan adalah tes yang dilakukan berguna untuk mendeteksi, merencanakan metode penanganan dan pengobatan yang tepat sejak dini sebelum penyakit berkembang.

### **2.2.4 Pengertian Laboratorium**

(Sari dan Resmiaty, 2017), “Kata laboratorium berasal dari bahasa Latin yang berarti “tempat bekerja”. Dalam perkembangannya, kata laboratorium mempertahankan arti aslinya, yaitu tempat bekerja khusus untuk keperluan penelitian ilmiah (ada fasilitas air, listrik, gas dan sebagainya)”. Sedangkan menurut Seyoum dalam (Mardiana dan Rahayu, 2017), laboratorium adalah tempat yang dilengkapi dengan berbagai instrumen, peralatan dan bahan kimia (reagen), untuk melakukan karya eksperimental, kegiatan penelitian dan prosedur pemeriksaan.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian Laboratorium adalah suatu ruangan atau kamar tempat melakukan kegiatan praktek atau penelitian yang ditunjang oleh adanya seperangkat alat-alat serta adanya infrastruktur laboratorium yang lengkap.



### **2.2.5 Pengertian Hasil Laboratorium**

Menurut (Resmiaty dan Sari, 2017) Pemeriksaan Hasil Laboratorium adalah suatu tindakan dan prosedur pemeriksaan khusus dengan mengambil bahan atau sampel dari pasien dalam bentuk darah, sputum (dahak), urine (air kencing/air seni), kerokan kulit, dan cairan tubuh lainnya dengan tujuan untuk menentukan diagnosis atau membantu menegakkan diagnosis penyakit.

(Mardiana dan Rahayu, 2017) mengatakan bahwa hasil pemeriksaan laboratorium merupakan informasi yang berharga untuk membedakan diagnosis, mengkonfirmasi diagnosis, menilai status klinik pasien, mengevaluasi efektivitas terapi dan munculnya reaksi obat yang tidak diinginkan.

Dari pernyataan di atas maka penulis menarik kesimpulan bahwa Pemeriksaan Hasil Laboratorium merupakan pemeriksaan dengan mengambil sampel dari pasien yang berisi informasi diagnosa penyakit berupa pemeriksaan sampel dalam bentuk darah, *urine*, dan bagian dalam tubuh lainnya.

### **2.2.6 Pengertian Pasien**

Prabowo (dalam Wilhamda, 2011) Pasien adalah orang yang memiliki kelemahan fisik atau mentalnya menyerahkan pengawasan dan perawatannya, menerima dan mengikuti pengobatan yang ditetapkan oleh tenaga kesehatan.

Menurut (Soejadi, 1996) pasien adalah individu terpenting dirumah sakit. Sedangkan (Aditama, 2002) berpendapat bahwa pasien adalah mereka yang diobati dirumah sakit.

Berdasarkan pendapat dari para ahli diatas dapat disimpulkan bahwa pasien adalah orang yang memiliki kelemahan fisik atau mentalnya menyerahkan pengawasan dan perawatannya, menerima dan mengikuti pengobatan yang ditetapkan oleh tenaga kesehatan atau para medis yang di obati dirumah sakit.





### **2.2.7 Pengertian Aplikasi Data Hasil Pemeriksaan Pasien pada Laboratorium Rumah Sakit Umum Daerah Kayuagung OKI Menggunakan *Framework Codeigniter***

Aplikasi Data Hasil Pemeriksaan Pasien pada Laboratorium Rumah Sakit Umum Daerah Kayuagung OKI Menggunakan *Framework Codeigniter* adalah sebuah sistem yang digunakan untuk menginput data hasil laboratorium pasien pada Rumah Sakit Umum Daerah Kayuagung agar mempermudah tenaga medis dalam merekapitulasi data dan pasien bisa menerima informasi hasil laboratorium secara cepat dan akurat.

## **2.3 Teori Khusus**

Dalam sub-bab ini akan diterapkan mengenai teori yang berkaitan dengan sejumlah fakta-fakta yang bersifat partikular. Berkaitan dengan judul dalam pembuatan Laporan Akhir, yang akan dibuat penulis adalah sebagai berikut.

### **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

<b>Simbol</b>	<b>Keterangan</b>
=	Disusun atau terdiri dari
+	Dan
[   ]	Baik...atau...
{ }n	n kali diulang/bernilai banyak
( )	Data opsional
* ... *	Batas komentar



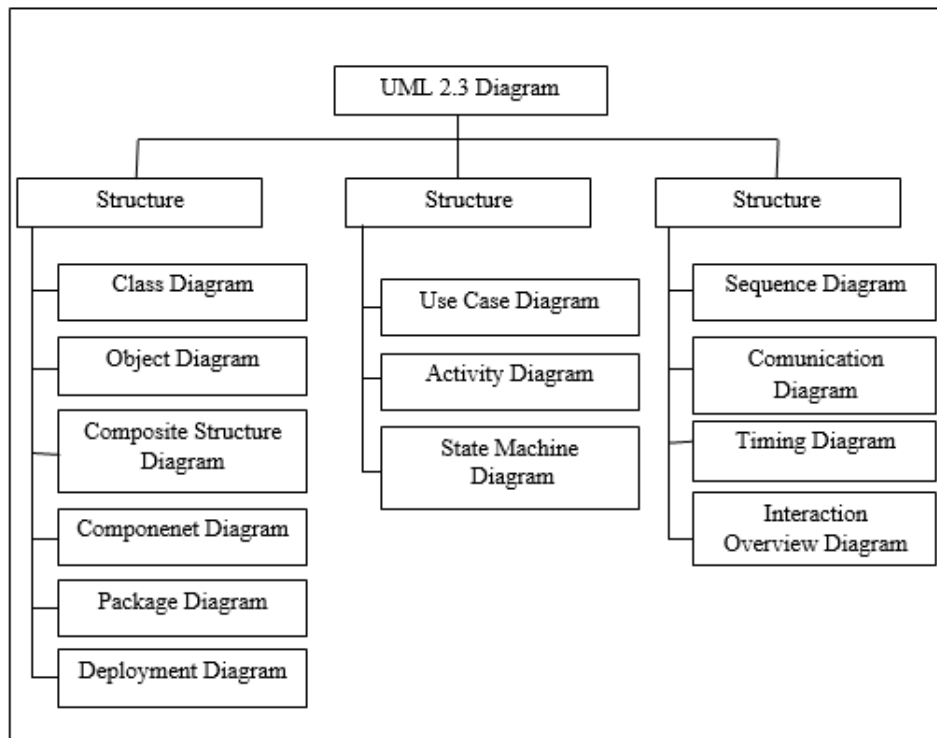
### 2.3.2 Pengertian Unified Modelling Language (UML)

Sukanto dan Shalahuddin (2018:137), menjelaskan Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

Rajendra (2022), “UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh Object Management Group dengan versi awal 1.0 pada bulan Januari 1997. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software”.

Adapun tujuan dan fungsi UML yaitu sebagai berikut:

1. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
2. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
3. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
4. Tidak hanya menggambarkan model sistem software saja, namun dapat memodelkan sistem berorientasi objek.
5. Mempermudah pengguna untuk membaca suatu sistem.
6. Berguna sebagai blueprint, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa coding suatu program.




**Gambar 2.2** Diagram UML

### 2.3.3 Pengertian Use Case Diagram




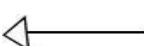



Kusuma dan Yosrita (2016:60), “Use case diagram digunakan untuk menspesifikasikan apa yang dapat dilakukan oleh sistem atau untuk menspesifikasikan kebutuhan fungsional utama dari aplikasi yang dibangun”.

Manalu dalam Haryanto (2018:67), “Use case diagram adalah suatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh actor dalam menyelesaikan sebuah pekerjaan”.



**Tabel 2.2** Simbol-Simbol UML Use Case

No.	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.



2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<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> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor





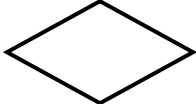
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### 2.3.4 Pengertian Activity Diagram




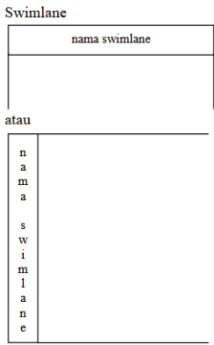
Rachman (2018:90), “Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak”.

Hendini (2016:109), “ Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis”.

**Tabel 2.3** Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas/activity 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu



Penggabungan/join 	Asosiasi penggabungan dimaa lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakuan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Interaksi/interaction 	Alur dari sebuah activity
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 2.3.5 Pengertian Class Diagram

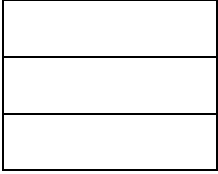
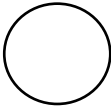



Rachman (2018:88), “Class diagram merupakan diagram class yang digunakan dari sistem yang dibuat”.

Kusuma dan Yosrita (2016:62), “Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur yang dibuat”.

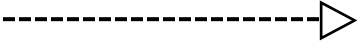
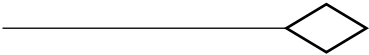


Hendini (2016:111), Class Diagram secara khas meliputi : Kelas (Class), Relasi Assosiations, Generalitation dan Aggregation, atribut (Attributes), operasi (operation/method) dan visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan Multiplicity atau Cardinality”.

**Tabel 2.4** Simbol Class Diagram

<b>Simbol</b>	<b>Deskripsi</b>
Kelas 	Kelas pada struktur sistem
Antarmuka/Interface  Nama_interface	Sama dengan konsep insterface dalam pemtograman berorientasi objek
Asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/ directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multicity
Generalisasi 	Relasi antar kelas dengan makna kebergantungan antarmuka



Kebergantungan/ dependency 	Relasi antar kelas dengan makna kebergantungan antarmuka
Agregasi / Aggregation 	Relasi antar kelas dengan makna semua bagian (whole-part)

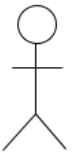
### 2.3.6 Pengertian Sequence Diagram

Rachman (2018:89), “Sequence diagram merupakan diagram yang menjelaskan alur proses dari setiap use case yang sudah dibuat”.

Kusuma dan Yosrita (2016:62), “Sequence diagram menjelaskan interaksi object yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan use case diagram, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case”.

Hendini (2016:110), “Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”.

**Tabel 2.5** Simbol Sequence Diagram

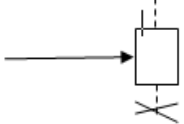
Simbol	Keterangan
<p>Aktor</p>  <p>Nama_aktor</p> <p>Tanpa waktu aktif</p>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal frase nama actor





<p>Garis hidup/lifeline</p> <p>⋮</p>	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>Nama objek : Nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu Aktif</p> <div style="border: 1px solid black; width: 20px; height: 30px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan tipe create</p> <p>—————→</p> <p>&lt;&lt;create&gt;&gt;</p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <p>1: nama_metode()</p> <p>—————→</p>	<p>Menyatakan sebuah objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>
<p>Pesan tipe send</p> <p>1: Masukan</p> <p>—————→</p>	<p>Menyatakan bahwa suatu objek mengirimkan data / masukkan informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return</p> <p>1 : keluaran</p> <p>-----→</p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>



<p>Pesan tipe destroy</p> <p><code>&lt;&lt;destroy&gt;</code></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maa ada destroy.</p>
---	--

## 2.4 Teori Program

Dalam sub-bab ini akan diterapkan mengenai teori yang berkaitan dengan bahasa pemrograman yang diaplikasikan penulis dalam pembuatan Laporan Akhir ini. Teori yang akan dibuat penulis adalah sebagai berikut.

### 2.4.1 Pengertian Visual Studio Code



**Gambar 2.3** Logo *Visual Studio Code*

(Sumber: [www.deviantart.com](http://www.deviantart.com))

Sole (2019:2), “Visual Studio Code telah menjadi alat pengembangan *cross-platform* pertama dalam Microsoft Visual Studio *family* yang berjalan di Windows, Linux, dan macOS secara gratis dan *open source* di (<https://github.com/Microsoft/vscode>), aplikasi ini jelas merupakan kode editor yang memudahkan untuk mengedit file kode dan sistem proyek berbasis folder serta menulis web *cross-platform* dan aplikasi mobile dengan dukungan terintegrasi untuk sejumlah besar bahasa dan fitur pengeditan yang kaya seperti *IntelliSense*, *finding symbol references*, *quickly reaching a type definition*, dan masih banyak lagi”.



Menurut Faisal (2017:13), “visual studio code adalah Integrated Development Environment (IDE) yang dikembangkan oleh Microsoft untuk mempermudah software developer mengembangkan aplikasi pada platform milik Microsoft”.

#### **2.4.2 Pengertian *Hypertext Markup Language* (HTML)**

Menurut Sutarman (2007:27) ” *Hypertext Markup Language* (HTML) adalah suatu bahasa yang digunakan untuk menulis halaman *web*”.

Menurut Larry (2012:3), ”*Hypertext Markup Language* merupakan suatu metode untuk mengimplementasikan konsep hypertext dalam suatu naskah atau dokumen”.

Abdulloh (2018:7), “HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh W3C (*WorldWide Web Consortium* ) berupa tag-tag yang menyusun setiap elemen dari *Website*”.

Menurut Arief (2011:23), “HTML atau HyperText Markup Language merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman *web*”.

Sedangkan Menurut Nugroho (2013:5), “HTML adalah kependekan dari (*HyperText Markup Language*), merupakan sebuah bahasa *Scripting* yang berguna untuk menuliskan halaman *Web*”.



**Gambar 2.4** Logo HTML

Menurut Tabrani dan Pudjiarti (2017:31-32), “Untuk dapat membuat *Website* dengan baik maka langkah awal yang harus dilakukan yaitu



mengenal kode-kode dasar HTML yang sering digunakan oleh programmer *web professional*. Kode HTML memiliki aturan dan struktur penulisan tersendiri yang disebut tag HTML”.

Tag adalah kode yang digunakan untuk memoles (*mark-up*) teks menjadi file HTML. Setiap tag diapit dengan tanda kurung runcing. Ada tag pembuka dan tagpenutup, yang membedakan tag penutup ditandai dengan tanda garis miring (*slash*) di depan awal tulisannya. Dokumen HTML merupakan dokumen yang disajikan dalam *web browser* dan biasanya hanya untuk menampilkan informasi maupun *interface*.

### 2.4.3 Pengertian *Cascading Style Sheets* (CSS)

Menurut Kadir dan Triwahyuni (2013:323), “CSS adalah kode yang dimaksudkan untuk mengatur tampilan halaman *web*”.

Menurut Arief (2011:11), “*Client side scripting* adalah salah satu jenis bahasa pemrograman *web* yang proses pengolahannya dilakukan disisi *client*”.



**Gambar 2.5** Logo CSS

Sedangkan Menurut Sibero (2013:112) menyatakan bahwa, “*Casading Style Sheet* memiliki arti Gaya Menata Halaman Bertingkat, yang artinya setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya”. *Casading Style Sheet* (CSS) digunakan untuk membuat *web* menjadi lebih rapi dan terstruktur.



Adapun contoh skrip dari CSS adalah sebagai berikut.

```
<div style= "font-weight:bold">
  <p> Contoh Paragraf dalam DIV dengan style font weight : bold
</p>

  <span> Contoh SPAN dalam DIV dengan style font-weight : bold
</span>

</div>
<div>
  <p> Contoh Paragraf dalam DIV tanpa style </p>
```

#### 2.4.4 Pengertian *Hypertext Preprocessor* (PHP)

Bahasa pemrograman berupa script yang bersifat *open source* yang digunakan untuk membuat halaman *Website* untuk menghasilkan isi *web* yang sesuai dengan permintaan *client* .



**Gambar 2.6** Logo PHP

Menurut Nugroho (2013:201), “PHP atau *Hypertext Preprocessor* adalah sebuah bahasa pemrograman yang berbentuk *Scripting*, sistem kerja dari program ini adalah sebagai *interpreter* bukan sebagai *compiler*”.

Menurut Arief (2011:43), “PHP (*Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis”.

Menurut Meissa (2009:15) , “PHP adalah suatu bahasa pemrograman yang berbasiskan *web (server side)*”.



Menurut Kadir (2018:358), “PHP merupakan bahasa pemrograman skrip yang diletakkan dalam *server* yang biasa digunakan untuk membuat aplikasi *web* yang bersifat dinamis”.

Sedangkan Menurut Abdullah (2018:127), “PHP merupakan kependekan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman *web* yang dapat disisipkan dalam skrip HTML dan bekerja di sisi *server*. Tujuan dari bahasa ini adalah membantu para pengembangan *web* untuk membuat *web* dinamis dengan cepat”.

#### 2.4.4.1 Sintaks Dasar *Hypertext Preprocessor* (PHP)

Abdulloh (2018:128), menjelaskan bahwa skrip PHP dituliskan di antara tanda `<?php` dan `?>` yang memisahkan skrip PHP dengan skrip lainnya. Setiap baris skrip PHP harus di akhiri dengan tanda *semicolon* (`;`). Jika tidak, maka akan menampilkan pesan eror.

Berikut contoh penulisan skrip PHP di dalam skrip HTML.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> Belajar PHP </title>
  </head>
  <body>
    <?php
      Echo “Belajar PHP untuk Pemula”;
    ?>
  </body>
</html>
```



#### 2.4.4.2 Tipe Data *Hypertext Preprocessor* (PHP)

Tipe data merupakan jenis dari suatu data yang akan di proses oleh bahasa pemrograman. Murya (2014:26), menjelaskan beberapa tipe data dalam PHP, yaitu:

1. **Integer** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. *Range* bilangan integer adalah antara -2.147.4833.647 sampai dengan 2.147.483.647.
2. **Double Floating** adalah tipe data yang berguna untuk menyimpan bilangan desimal. *Range* bilangan *floating point* antara 1e308 sampai dengan 1e308.
3. **Boolean** adalah tipe data yang paling sederhana, berupa *TRUE* dan *FALSE*.
4. **String** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan string harus diapit dengan tanda petik, baik berupa petik tunggal ( ' ... ' ) maupun petik ganda ( " ... " ).
5. **Objek** adalah tipe data dibuat dengan tujuan agar para programmer terbiasa dengan OOP. Tipe data ini bisa berupa bilangan.
6. **Array** merupakan **Tipe Compound Primitif**, terdapat pada bahasa pemrograman lain.
7. **Null** adalah tipe data yang tidak memuat apapun. Setiap variable yang diset menjadi tipe data Null, ini akan menjadikan variabel tersebut kosong.
8. **Resources** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.

#### 2.4.5 Pengertian *My Structured Query Language* (MySQL)

Menurut Hidayatullah dan Kawistara (2017:175), “MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh pemrograman aplikasi *web*”.



Sementara Santoso (2017:86), “*MySQL* merupakan *software* yang tergolong sebagai *Database Management System (DBMS)* yang bersifat *open source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (*code* yang dipakai untuk membuat *MySQL*)”.



**Gambar 2.7** Logo MySQL

Henry Februariyanti (2012: 128), “*MySQL* adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar *Structured Query Language (SQL)* *MySQL* juga telah mendukung bahasa pemrograman berfitur API seperti java sehingga memudahkan para programmer java untuk berkoneksi dengan menggunakan *MySQL*”.

Berikut beberapa keistimewaan yang dimiliki *MySQL* antara lain:

- a. Portabilitas yaitu *MySQL* dapat berjalan secara stabil pada berbagai macam sistem operasi
- b. *Open Source* yaitu *MySQL* di distribusikan secara gratis
- c. *Multi User* yaitu *MySQL* dapat digunakan dengan banyak user tanpa memiliki masalah
- d. *Performance Turning* yaitu *MySQL* memiliki tipe kolom yang sangat kompleks
- e. Perintah dan fungsi yaitu *MySQL* operator dan fungsi yang secara penuh perintah *Select* dan *Where* dalam *query*
- f. Keamanan yaitu *MySQL* memiliki beberapa lapisan keamanan dengan sistem perizinan yang detail serta sandi yang terenkripsi
- g. Skalabilitas dan pembatasan yaitu *MySQL* mampu menangani database berskala besar dengan jumlah record lebih dari 50 juta, table lebih dari 60 ribu dan baris lebih dari 5 milyar





- h. Konektivitas yaitu *MySQL* dapat melakukan koneksi dengan klien dengan menggunakan TCP/IP, UNIX dan Named Pipes
- i. Lokalisasi yaitu *MySQL* mendeteksi kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa meskipun bahasa Indonesia belum masuk didalamnya
- j. Antarmuka yaitu *MySQL* memiliki antarmuka yang mendukung berbagai bahasapemrograman yang menggunakan fungsi API
- k. Klien dan peralatan yaitu *MySQL* dilengkapi dengan berbagai tool yang dapat digunakan secara online
- l. Struktur table yaitu *MySQL* memiliki struktur table yang fleksibel dalam menangani alter table jika dibandingkan dengan bahasa pemrograman lainnya.

#### 2.4.6 Pengertian XAMPP

Menurut Solution (2010:1), “XAMPP merupakan suatu paket instalasi yang berisi Apache, PHP dan *MySQL*”.

Sedangkan menurut Widijanuarto (2010:3), “XAMPP merupakan sebuahaplikasi yang berisi PHP, *MySQL*, dan Apache yang diperlukan untuk melakukan instalasi CMS ”.



**Gambar 2.8** Logo XAMPP

Hidayatullah dan Kawistara (2017:125) dalam bukunya mengatakan bahwa, “XAMPP *support* untuk banyak sistem operasi seperti *Windows*, *Linux*, *Mac OS* dan *Solaris* sehingga tidak terdapat masalah ketika melakukan perpindahan sistem operasi”.



## 2.4.7 Pengertian *Composer*



**Gambar 2.9** Logo *Composer*

(Sumber: [www.deviantart.com](http://www.deviantart.com))

Abdulloh (2018:5), “Composer digunakan untuk menginstal dan menambahkan dependency PHP pada Laravel. Banyak sekali Plugin yang dapat digunakan untuk meningkatkan fungsi laravel yang disediakan oleh pihak ketiga. Plugin-plugin tersebut dapat diinstal dengan mudah menggunakan Composer”.

## 2.4.8 Pengertian *Framework Codeigniter*

### 2.4.8.1 *Framework*

Naista (2017), *Framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau Kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan data. Salah satu keuntungan menggunakan *framework* adalah memberi struktur yang baik dalam program yang dibuat karena *framework* memiliki *library* atau fungsi yang bisa langsung digunakan. Selain itu, *framework* memudahkan dalam pekerjaan program secara tim.

Menurut Wardana (2010:3), “*Framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain, sehingga dalam pembuatan aplikasi website harus mengikuti aturan dari *framework* tersebut. Dengan *framework* (dalam hal ini



framework PHP), kita tidak perlu memikirkan kode perintah/fungsi dasar karena hal-hal penunjang lainnya seperti koneksi database, validasi form, GUI, dan keamanan telah disediakan oleh framework”.

#### **2.4.8.2 Codeigniter**



**Gambar 2.10** Logo Codeigniter

Menurut (Arrhioui et al., 2017), Codeigniter adalah kerangka kerja pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur. Codeigniter memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti helpers and libraries untuk mengimplementasi tugas yang biasa dilakukan. Dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat. Dan pengembang tidak perlu menulis lagi dari awal.