



---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

Di dalam teori ini, terdapat penjelasan dari pengertian Komputer, Perangkat Lunak (*Software*), Basis Data (*Database*) dan Internet.

##### 2.1.1 Pengertian Komputer

Menurut Robert H. Blissmer (dalam Harmayani et al., 2021:3), pengertian computer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima input, memproses input sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahannya, serta menyediakan output dalam bentuk informasi.

Menurut Husda et al., (dalam Hanafri et al., 2019:88), komputer merupakan alat untuk mengolah data sesuai perintah yang sudah dirumuskan secara cepat dan tepat, serta diorganisasikan supaya secara otomatis menerima dan menyimpan data berdasarkan intruksi intruksi yang telah tersimpan di dalam memori.

##### 2.1.2 Pengertian Perangkat Lunak

Indra (dalam Sudarso et al., 2022:5), mendefinisikan *software* adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer. Data elektronik yang disimpan dapat berupa program atau instruksi yang akan menjalankan perintah. *Software* dapat diartikan juga sebagai segala jenis program yang digunakan untuk pengoperasian komputer dan peralatannya.

Menurut Davis (dalam Gede et al., 2022:347), *software* merupakan suatu program yang berisi kumpulan intruksi atau perintah untuk melakukan proses pengolahan data. *Software* digunakan sebagai penghubung antara pengguna atau user dalam hal ini manusia dengan perangkat keras.



### **2.1.3 Pengertian Data**

Menurut Sutabri (dalam Ayu & Permatasari, 2018:15), data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai.

Slamet Riyadi (dalam Jollyta et al., 2020:2), data sebagai kumpulan informasi yang diperoleh dari pengamatan di mana data bisa berupa angka-angka dan lambang-lambang.

### **2.1.4 Pengertian Basis Data (*Database*)**

Menurut Ali (dalam Sudarso et al., 2022:3), Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan disimpanan luar komputer (*external memory*) dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di *SI*, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basis data (*database system*) ini adalah suatu *SI* yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam dalam suatu organisasi.

Indrajadi (dalam Sudarso et al., 2022:3) , mengatakan Basis data adalah kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi. Artinya basis data, tempat penyimpanan data yang besar dimana dapat digunakan oleh banyak pengguna. Seluruh item basis data tidak lagi dimiliki oleh satu departemen, melainkan menjadi sumber daya perusahaan yang dapat digunakan bersama-sama.



### 2.1.5 Metode Pengembangan Sistem Waterfall

Menurut Novitasari (Kurniawan et al., 2020:161) Metode waterfall adalah hal yang menggambarkan pendekatan secara sistematis dan juga berurutan (step by step) pada sebuah pengembangan perangkat lunak. Tahapan dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan yaitu planning, permodelan, konstruksi, sebuah system dan penyerahan sistem kepada pengguna, dukungan pada perangkat lunak lengkap yang dihasilkan.

Sedangkan menurut (Darisman & Widiyanto, 2019:3990) adalah sebagai berikut.

Metode *Waterfall* ini biasa disebut dengan “siklus hidup klasik” atau yang sekarang disebut model air terjun. Metode ini mengambil pendekatan yang terstruktur dan tersistematis mulai dari tingkat kebutuhan sistem hingga ke tahap pemeliharaan. Disebut air terjun karena seperti air terjun yang terjatuh satu per satu, model ini menggunakan penyelesaian tahap sebelumnya baru kemudian dapat dilanjutkan ke tahap berikutnya. Tahapan-tahapan dalam model ini menurut Widiyanto adalah sebagai berikut:

- a. **Rekayasa Sistem**  
Tahap ini dimulai dengan mencari kebutuhan seluruh sistem untuk diterapkan dalam perangkat lunak.
- b. **Analisis Kebutuhan**  
Pada tahap ini pengembang sistem membutuhkan komunikasi yang bertujuan memahami perangkat lunak yang diharapkan oleh pengguna dan keterbatasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung.
- c. **Desain**  
Spesifikasi persyaratan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan.
- d. **Pengodean**  
Pada fase ini, sistem terintegrasi dengan sintaksis sehingga sistem informasi dapat digunakan sesuai kebutuhan, yang akan diintegrasikan pada tahap selanjutnya.
- e. **Pengujian**  
Semua unit yang telah dikembangkan dan pengkodean yang benar diuji langsung untuk penggunaannya, seperti menggunakan pengujian blackbox. Metode pengujian blackbox ini adalah pengujian yang hanya memperhatikan hasil masukan dan keluaran sistem.



f. **Pemeliharaan**

Fase terakhir dalam model air terjun. Perangkat lunak atau sistem informasi yang sudah jadi, dijalankan kemudian dipelihara, karena tidak selamanya sistem akan berjalan dengan baik. Maka, itulah fungsi pemeliharaan, guna menghindari sistem dari bug-bug yang timbul.

## **2.2 Teori Judul**

Di dalam teori ini, terdapat penjelasan dari pengertian Implementasi, Dispensasi, Pemakaian, Jalan Kabupaten, dan *Online*.

### **2.2.1 Pengertian Implementasi**

Menurut Harsono (dalam Rosad, 2019:176), implementasi adalah suatu proses untuk melaksanakan kebijakan menjadi Tindakan kebijakan dari politik ke dalam administrasi. Pengembangan kebijakan dalam rangka penyempurnaan suatu program.

Menurut Mulyasa (dalam Magdalena et al., 2021:120) implementasi merupakan suatu proses penerapan ide, konsep, kebijakan atau inovasi dalam suatu tindakan praktis sehingga memberikan dampak baik berupa perubahan pengetahuan, keterampilan maupun nilai dan sikap.

### **2.2.2 Pengertian Dispensasi**

Menurut Sudarsono (dalam Judiasih, 2020:209), dispensasi adalah merupakan pengecualian dari suatu aturan secara umum untuk sesuatu keadaan yang bersifat khusus, pembebasan dari suatu larangan atau kewajiban.

Sama halnya pula sebagaimana disampaikan oleh C.S.T. Kansil (dalam Judiasih, 2020:209), makna dispensasi merupakan penetapan yang menyatakan bahwa suatu ketentuan peraturan memang tidak berlaku bagi kasus yang diajukan oleh seorang pemohon.



### 2.2.3 Pengertian Pemakaian (Penggunaan)

Kata pemakaian biasa juga kita sebut dengan penggunaan ataupun pemanfaatan. Kata tersebut memiliki arti yang sama. Dalam Kamus Besar Bahasa Indonesia, “Pemakaian diartikan sebagai proses, cara, perbuatan memakai. Arti lainnya adalah penggunaan”.

Pemanfaatan memiliki makna proses, atau perbuatan memanfaatkan. (Kamus Besar Bahasa Indonesia Kontemporer). Pemanfaatan adalah suatu kegiatan, proses, cara atau menjadikan suatu yang ada menjadi bermanfaat.

### 2.3.4 Pengertian Jalan Kabupaten

Menurut Rozy (2020:5) pengertian jalan kabupaten yaitu sebagai berikut:

Jalan kabupaten, merupakan jalan lokal dalam sistem jaringan jalan primer yang tidak termasuk jalan yang menghubungkan ibukota kabupaten dengan ibukota kecamatan, antar ibukota kecamatan, ibukota kabupaten dengan pusat kegiatan lokal, antar pusat kegiatan lokal, serta jalan umum dalam sistem jaringan jalan sekunder dalam wilayah kabupaten, dan jalan strategis kabupaten.

Berdasarkan Undang-undang nomor 38 tahun 2004 tentang jalan, jalan kabupaten merupakan jalan lokal dalam sistem jaringan jalan primer yang tidak termasuk jalan yang menghubungkan ibukota kabupaten dengan ibukota kecamatan, antar ibukota kecamatan, ibukota kabupaten dengan pusat kegiatan lokal, antarpusat kegiatan lokal, serta jalan umum dalam sistem jaringan jalan sekunder dalam wilayah kabupaten, dan jalan strategis kabupaten.

### 2.3.5 Pengertian Online

Kata *online* dalam bahasa Indonesia sering diterjemahkan sebagai “dalam jaringan” atau sering disingkat daring. “Kata “daring” secara bahasa merupakan kata baku dalam bahasa Indonesia yang merupakan singkatan dari “dalam jaringan”, yakni terhubung melalui jejaring komputer, internet, dan sebagainya” (Jundi, 2020:106).

Sastypratiwi & Dwiyani (dalam Anggraini et al., 2021:3012) mengatakan bahwa, Dalam Kamus Besar Bahasa Indonesia, daring merupakan sebagai suatu



keadaan yang sedang menggunakan suatu jaringan, terhubung dengan jaringan, perangkat satu dengan perangkat yang lainnya yang saling terhubung sehingga bisa saling berkomunikasi meskipun dengan jarak jauh.

### **2.3 Teori Khusus**

Dalam teori ini, terdapat pengertian tentang Unified Modeling Language (UML) dan jenis-jenisnya yang meliputi *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

#### **2.3.1 Unified Modeling Language (UML)**

Wira et al., (2019:33) menyatakan bahwa “UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Simatupang & Sianturi (2019:15) menyatakan “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

#### **2.3.2 Jenis-Jenis Diagram UML**

##### **2.3.2.1 Pengertian Use Case Diagram**

Menurut (Simatupang & Sianturi, 2019:16), pengertian use case diagram adalah sebagai berikut.

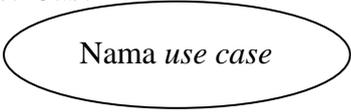
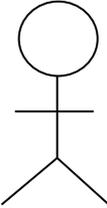
Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan system informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah system informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Marudut & Siregar (2018:252) “Use case menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem, sedangkan aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem. Use case diagram menggambarkan bagaimana proses-proses yang dilakukan oleh aktor terhadap

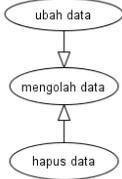
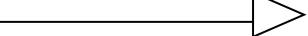
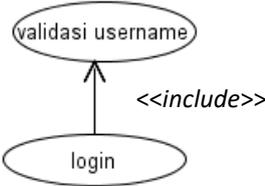


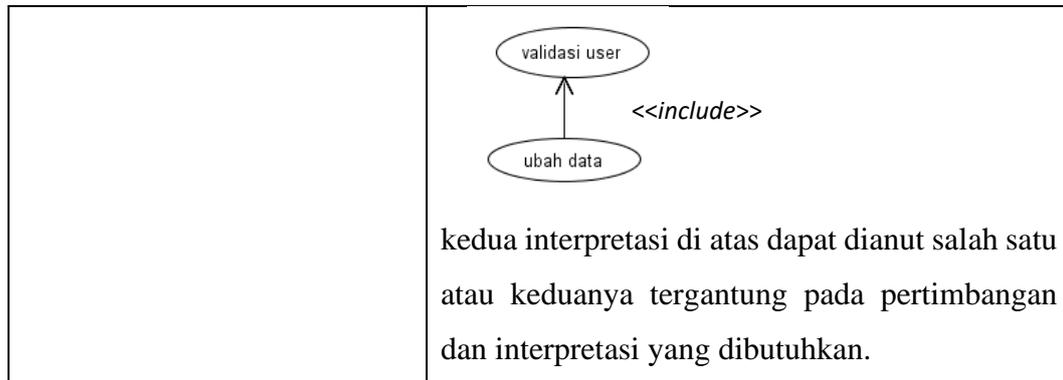
sebuah sistem”.

**Tabel 2.1** Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal diawal frase nama <i>use case</i>.</p>
<p>Aktor/<i>Actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi / <i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>



<p>Generalisasi/generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya: arah panah mengarah pada use case yang menjadi generalisasinya (umum)</p> 
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><code>&lt;&lt;include&gt;&gt;</code></p>  <p><code>&lt;&lt;uses&gt;&gt;</code></p> 	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <p>a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</p>  <p>b. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>



Sumber : Rosa dan Shalahuddin (dalam Simatupang & Sianturi, 2019:16-18)

### 2.3.2.2 Pengertian *Class Diagram*

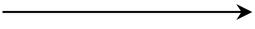
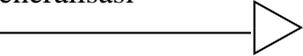
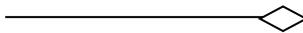
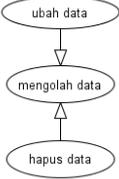
Menurut Wira et al., (2019:33) , “Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Class diagram terdiri dari atribut dan operasi dengan tujuan pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai”.

Marudut & Siregar (2018:252), ” *Class diagram* memberikan gambaran hubungan antara tabel-tabel yang ada dalam database. Masing-masing class memiliki attribute dan metoda atau fungsi sesuai dengan proses yang terjadi”.

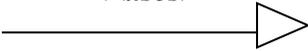
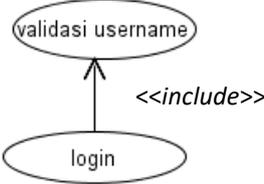
**Tabel 2.2** Simbol-simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / <i>interface</i>  <b>nama_interface</b>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .



Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).
Ekstensi / <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)  



<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p style="text-align: center;"> <code>&lt;&lt;include&gt;&gt;</code>      <code>&lt;&lt;uses&gt;</code>   </p>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <p>c. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</p> <div style="text-align: center;">  </div> <p>d. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <div style="text-align: center;">  </div> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	---

Sumber : Rosa dan Shalahuddin (dalam Simatupang & Sianturi, 2019:20)

### 2.3.2.3 Pengertian Activity Diagram

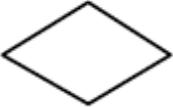
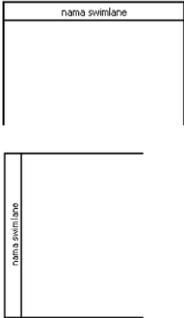
Wira et al., (2019:33) “Activity diagram merupakan diagram yang menggambarkan workflow atau aktivitas dari sebuah sistem yang ada pada perangkat lunak”.



Simatupang & Sianturi (2019:18) mengatakan bahwa :

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor.

**Tabel 2.3** Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Bagaimana objek dibentuk atau diawali.
Aktivitas 	Memperlihatkan bagaimana kelas antarmuka saling berinteraksi satu sama lain.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Bagaimana objek dibentuk dan dihancurkan.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Rosa dan Shalahuddin (dalam Simatupang & Sianturi, 2019:18)



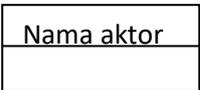
### 2.3.2.4 Pengertian *Sequence Diagram*

Menurut Rosa dan Shalahuddin (dalam Simatupang & Sianturi, 2019:19), Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Wira et al., (2019:33) mendefinisikan *sequence diagram* sebagai berikut.

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambaran *sequence diagram* dibuat minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga semakin banyak.

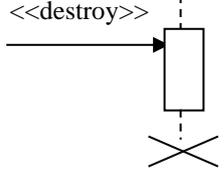
**Tabel 2.4** Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p><i>Actor</i></p>  <p>nama aktor</p> <p>Atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>



3.	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">         Nama objek : nama kelas       </div>	<p>menyatakan objek yang berinteraksi pesan.</p>
4.	<p>Waktu aktif</p> <div style="border: 1px solid black; width: 30px; height: 40px; margin: 10px auto;"></div>	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> <pre> sequenceDiagram     participant Actor     Actor-&gt;&gt;Object: 1: login()     activate Object     Object-&gt;&gt;Object: 2: cekStatusLogin()     Object-&gt;&gt;Object: 3: open()     deactivate Object   </pre> <p>maka cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe create</p> <p style="text-align: center;">&lt;&lt;create&gt;&gt;</p>	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe call</p> <p style="text-align: center;">1: nama_metode()</p>	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> <pre> sequenceDiagram     participant Actor     Actor-&gt;&gt;Object: 1: nama_metode()     activate Object     Object--&gt;&gt;Object: 1: nama_metode()     deactivate Object   </pre> <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus</p>



		ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 1: keluaran 	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.
9.	Pesan tipe destroy <<destroy>> 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber : Rosa dan Shalahuddin (dalam Simatupang & Sianturi, 2019:18)

## 2.4 Teori Program

Di dalam teori ini, terdapat penjelasan dari pengertian Visual Studio Code, HTML, CSS, Bootstrap, PHP, Laravel, Javascript, MySQL, dan XAMPP.

### 2.4.1 Pengertian Visual Studio Code

Menurut Joni Kurniawan (2019:155), pengertian Visual Studio Code adalah sebagai berikut.

Visual Studio Code adalah kode editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan GitHub, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan *refactoring* kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah



tema, pintasan keyboard, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan.

Sedangkan menurut Permana & Romadlon (2019:155), pengertiannya adalah sebagai berikut.

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi multiplatform, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace* Visual Studio Code (seperti C++, C#, Python, Go, Java,dst).

#### 2.4.2 Pengertian HTML

Andri (dalam Muhammad & Suci Ananda, 2020:43) mengatakan bahwa HTML (*Hypertext Markup Language*) adalah semacam bahasa pengkodean bukan sebagai bahasa pemrograman dan juga sekumpulan perintah yang terformat yang digunakan untuk membuat halaman dokumen web *Hypertext*.

Menurut Hariadi dkk (Fatimah & Samsudin, 2019), HTML (*Hypertext Markup Language*) adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman web.

#### 2.4.3 Pengertian CSS

Menurut Sengkey (dalam Fatimah & Samsudin, 2019:36), 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.

Menurut Aditama (dalam Hendri & Mochammad Arief Sutisna, 2021:16), Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web yang bertujuan untuk membuat website agar lebih menarik dan terstruktur.



#### 2.4.4 Pengertian Bootstrap

Menurut Martin J dkk (dalam K. Wijaya et al., 2020:8-9), Bootstrap merupakan salah satu jenis *framework* gabungan dari CSS dan *Java script* yang ditawarkan sebagai alternatif diantaranya *framework* lainnya yang dimana awal *framework* ini dikembangkan oleh Mark Otto dan Jacob Thornton dikantor Twitter dengan maksud untuk menghadirkan konsistensi ketahapan *interface development* dalam membangun sebuah *website*.

Pengertian bootstrap menurut Nugroho & Setyawati (dalam Najib et al., 2022:115) adalah *framework* css untuk membuat tampilan web yang menyediakan class dan komponen yang sudah siap dipakai dan digunakan.

#### 2.4.5 Pengertian PHP

Menurut Anhar (dalam K. Wijaya et al., 2020:9), PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu dibuat oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru.

Menurut Hariadi dkk (dalam Fatimah & Samsudin, 2019:36), PHP adalah bahasa pemrograman web atau *scripting language* yang dijalankan diserver, PHP adalah sekumpulan *script* yang digunakan untuk mengolah data form dari web.

#### 2.4.6 Pengertian Laravel

Menurut WK & Pangestu, (2022:3), “Laravel adalah sebuah *framework* yang digunakan untuk membuat web dengan bahasa pemrograman laravel yang sederhana dan sudah disediakan sumber *code* pada *Github* yang dapat memudahkan pemakai untuk melakukan *coding*”.

Hadi Pratama et al., (dalam Ramadani & Hanafi, 2022:212), Laravel adalah *Framework* berbasis PHP yang sifatnya open source, dan menggunakan konsep model –view –controller. Laravel berada di bawah lisensi MIT License dengan menggunakan *Github* sebagai tempat berbagi *code* menjalankannya.



#### 2.4.7 Pengertian Javascript

Rudjiono & Saputro (2020:61) mengatakan, “*Javascript* adalah bahasa pemrograman atau kode *script* yang diletakan bersama kode HTML ataupun terpisah yang digunakan untuk membuat tampilan *website* lebih dinamis”.

Zainal Arifin & Smitdev (dalam Hendri & Mochammad Arief Sutisna, 2021:16) mengatakan bahwa, *JavaScript* adalah *script* program berbasis *client* yang dieksekusi oleh browser, sehingga membuat halaman web melakukan tugas-tugas tambahan yang tidak bisa dilakukan oleh *script* HTML biasa.

#### 2.4.8 Pengertian MySQL

Menurut Nugroho (dalam WK & Pangestu, 2022:3), *MySql* adalah Sebuah program pembuat dan pengelola *database* atau yang sering disebut *Database Management System* (DBMS). Sifat dari DBMS ini adalah *open source*. *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multi User* (banyak pengguna).

Budi Raharjo (dalam Nofyat et al., 2018:13) mengatakan bahwa, *MySQL* merupakan *software* RDBMS (atau *software database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-treaded*).

#### 2.4.9 Pengertian XAMPP

Menurut Andi (dalam Saed Novendri et al., 2019:48), *Xampp* merupakan singkatan dari X (empat system operasi apapun), *XAMPP* merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Seperti *Apache*, *MYSQL*, *PHPP*, dan *Perl*. *Xampp* adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket *Xampp* sudah terdapat *Apache* (*Web Server*), *Mysql* (*Database*), *PHP* (*server side scripting*), *Perl*, *FTP server*, *PhpMyAdmin*, dan berbagai Pustaka bantu lainnya.

Menurut Mawaddah dan Fauzi (dalam Putra & Nita, 2019:82), *XAMPP* adalah *software* web *server apache* yang di dalamnya tertanam *server MySQL* yang

---



didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri dapat berjalan pada berbagai macam *platform* seperti *Windows*, *Linux*, *Mac OS X* dan *Solaris*.