



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

Dalam teori ini, meliputi penjelasan tentang Pengertian Komputer, Perangkat Lunak (*Software*), Basis Data, Metode Pengembangan Sistem dan Metode Pengujian.

2.1.1 Pengertian Komputer

Komputer adalah suatu perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya (Prawiro, 2019 dalam Harmayani *et al.*, 2021: 1).

Komputer 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 (Blissmer dalam Lonando & Fadillah, 2023: 54)

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa komputer adalah sebuah alat komunikasi yang dapat mengelola data dan informasi yang sangat dibutuhkan secara otomatis.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut (Harmayani *et al.*, 2021: 43) Perangkat lunak atau *Software* komputer merupakan sekumpulan data elektronik, yang tersimpan dan kemudian dikendalikan oleh perangkat komputer.

Menurut (Zalukhu *et al.*, 2023: 62) Perangkat lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual).

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa perangkat lunak atau *Software* adalah sekumpulan data elektronik yang tersimpan dan



dikendalikan oleh perangkat komputer, serta program komputer yang tersosiasi dengan dokumentasi perangkat.

2.1.3 Pengertian Basis Data

Menurut (Pane *et al.*, 2022: 93) Basis Data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari database.

Menurut (Priyandanu *et al.*, 2020: 92) Basis data adalah media untuk menyimpan data yang mana merupakan table-table yang berisi data dan merupakan kumpulan dari field dan kolom.

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa basis data atau *database* adalah kumpulan informasi atau data yang disimpan secara sistematis di dalam komputer. *Database* ini dapat diakses dan diperiksa menggunakan program komputer tertentu untuk memperoleh informasi yang diinginkan.

2.1.4 Pengertian Sistem

Menurut Rasefta dan Esebella (2020: 52) Sistem adalah kumpulan komponen yang saling terkait satu dengan yang lainnya kegiatan pokok untuk mencapai suatu tujuan tertentu.

Menurut Lonando dan Fadillah (2023: 54) Sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang berhubungan yang berfungsi dan dengan tujuan yang sama. Sistem terdiri dari unsur-unsur seperti masukan (input), pengolahan (processing), serta keluaran (output).

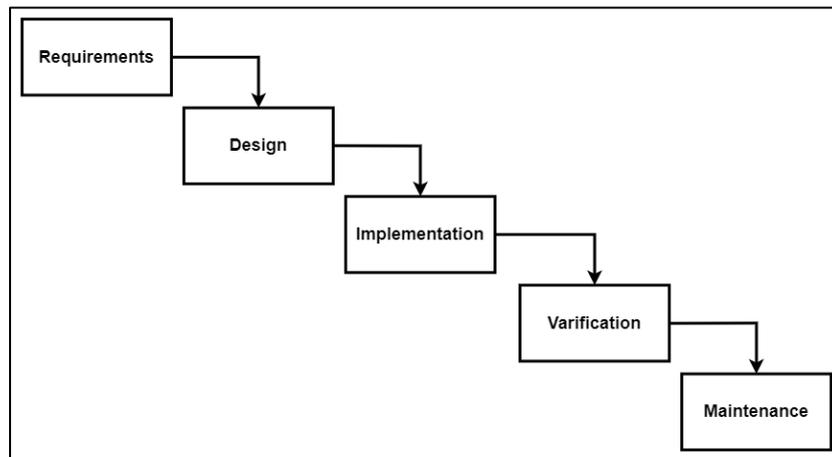
Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa sistem adalah suatu kumpulan komponen atau unsur yang saling terkait dan saling mempengaruhi satu sama lain dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

2.1.5 Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan adalah metode *waterfall*. Menurut Lonando dan Fadillah (2023: 54) Metode *waterfall* adalah rangkaian pada saat proses pengembangan *Software* secara berurutan, dimana kemajuan akan tetap terus mengalir kebawah melalui proses perencanaan, pemodelan, implementasi dan pengujian.

Pengertian dari metode *waterfall* juga rangkaian pada saat proses pengembangan *Software* secara berurutan, dimana kemajuan akan tetap terus mengalir kebawah melalui proses perencanaan, pemodelan, implementasi dan pengujian (Kinaswara *et al.*, 2019 dalam Lonando & Fadillah, 2023: 54).

Pengembangan sistem dengan menggunakan metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari Analisis, Desain, Pengodean, Pengujian dan Tahap Pendukung (*support*). Berikut ini adalah gambar ilustrasi model *waterfall*.



Gambar 2.1 Metode *Waterfall*

Dalam pengembangannya model *waterfall* memiliki tahapan yang berurutan. Menurut Lonando dan Fadillah (2023: 54) Berikut adalah tahapan dari metode *waterfall* :

1. Analisa Kebutuhan dalam pengumpulan data dapat menggunakan cara wawancara dan studi literature. Para analisis akan mencari informasi sebanyak mungkin sehingga sistem yang dibuat akan sesuai yang dibutuhkan.



2. Desain sistem pada langkah ini akan menterjemahkan kebutuhan perangkat lunak sebelum pembuatan coding. Dalam proses ini membuat struktur data, arsitektur perangkat lunak dan tampilan program.
3. Penulisan Kode Program, Kode program adalah penerjemahan desain kedalam bahasa yang dapat dikenali oleh komputer. Pada tahap ini adalah tahapan nyata untuk mengerjakan sistem. Dalam tahap ini biasa menggunakan bahasa PHP, HTML dan *Laravel 10*
4. Pengujian Program pada tahap ini adalah tahapan akhir dalam perancangan sistem. Dari beberapa proses yang dilakukan, maka aplikasi sudah dapat digunakan oleh *user*.
5. Pemeliharaan Perangkat lunak yang dibuat terkadang memiliki berbagai kendala dan terkadang susah dipahami oleh pengguna, sehingga butuh penyesuaian ulang. Perubahan tersebut dikarenakan berbagai faktor seperti penyesuaian lingkungan, sistem operasi baru dan perkembangan fungsional.

2.1.6 Metode Pengujian

Pengujian sistem merupakan proses pengeksekusian sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian bug, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak. Pengujian yang dilakukan pada penelitian ini adalah metode pengujian langsung yaitu dengan menggunakan pengujian *White-Box* dan *Black-Box*. Menurut Wahyudi dan Jakara (2023: 25) *Black-Box* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.



2.2. Teori Khusus

Dalam teori ini, meliputi penjelasan tentang Pengertian *Unified Modeling Language (UML)*, *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan Kamus Data (*Data Dictionary*).

2.2.1 Pengertian Unified Modeling Language (UML)

Menurut Sukamto (2022:179), “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan dalam industri untuk mendefinisikan *requirement* membuat analisis dan desain, serta mendefinisikan arsitektur dalam pemrograman berorientasi objek”.

Munawar (2021:49) “UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek”.

Disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa standar industri yang banyak digunakan untuk mendefinisikan persyaratan, analisis, desain dan arsitektur dalam pengembangan sistem berorientasi objek, dan merupakan alat yang andal untuk mengembangkan sistem ini.

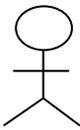
2.2.2 Pengertian Use Case Diagram

Menurut Sukamto (2022:196), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

Munawar (2021:93), “*Use Case* adalah deskripsi fungsi dari sebuah *system* dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tiap interaksi antara *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

Disimpulkan bahwa *Use case* diagram adalah pemodelan yang menggambarkan perilaku suatu sistem informasi dan fungsi suatu sistem dari sudut pandang pengguna, interaksi antar pengguna dan sistem melalui cerita tentang penggunaan sistem.

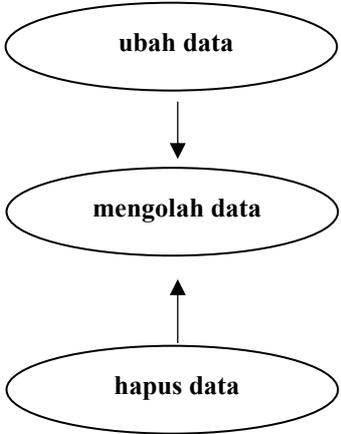
Tabel 2.1 Simbol *Use Case Diagram*

No.	Simbol	Nama	Fungsi
1.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.	 Nama aktor	Aktor <i>/actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
3.		Asosiasi <i>/association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor.



No.	Simbol	Nama	Fungsi
4.	<p style="text-align: center;"> <<extend>> → </p>	Ekstensi <i>/extend</i>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan 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</p> <div style="text-align: center;"> <pre> graph TD A([Validasi username]) -- "<<extend>>" --> B([Validasi user]) B -- "<<extend>>" --> C([Validasi sidik jari]) </pre> </div> <p>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>



No.	Simbol	Nama	Fungsi
5.		Generalisasi <i>/generalization</i>	<p>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</p> 

2.2.3 Pengertian *Class Diagram*

Menurut Sukamto (2022:181), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

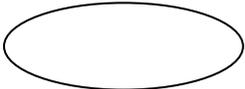
Sedangkan menurut Munawar (2021:105), “Class diagram adalah diagram statis. Ini mewakili pandangan statis dari aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak”.



Disimpulkan bahwa *Class* diagram adalah mendefinisikan kelas yang digunakan untuk membangun sistem, dan digunakan hanya untuk memvisualisasikan aspek sistem, tetapi juga membuat kode eksekusi aplikasi perangkat lunak.

Adapun symbol-simbol yang sering digunakan dalam *Class* Diagram dapat dilihat pada tabel berikut ini.

Tabel 2.2 Simbol *Class* Diagram

No.	Simbol	Nama	Fungsi
1.	nama_kelas + atribut + operasi ()	Kelas	Kelas pada struktur sistem
2.	 nama_interface	Antarmuka/ <i>interface</i>	Arah Aliran Program
3.		Asosiasi/ <i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
5.		Agregasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

6.		Kebergantungan/ <i>dependency</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
7.		Agregasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>)

2.2.4 Pengertian *Activity Diagram*

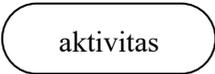
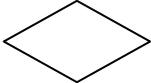
Menurut Sukanto (2022:203), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Sedangkan menurut Munawar (2021:135), “*Activity Diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem”.

Disimpulkan bahwa *Activity diagram* adalah representasi grafis yang menggambarkan alur kerja atau aktivitas suatu sistem atau proses bisnis, serta aspek dinamis dari sistem UML.

Tabel 2. 3 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

No.	Simbol	Keterangan
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

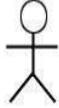
2.2.1 Pengertian *Sequence Diagram*

Menurut Sukanto (2022:206), “Diagram sekuens/urutan menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.

Sedangkan menurut Munawar (2021:221), “*Sequence diagram* adalah grafik dua dimensi dimana objek ditunjukkan dalam dimensi *horizontal* sedangkan *lifeline* ditunjukkan dalam dimensi *vertikal*”.

Disimpulkan bahwa *Sequence Diagram* adalah representasi grafis dua dimensi yang menggambarkan perilaku objek dalam *use case* yang menampilkan objek dalam dimensi *horizontal* dan kehidupan dalam dimensi *vertikal*, dan menyertakan kehidupan objek dan pesan yang dikirim antar objek.

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p><i>Aktor</i></p>  <p>atau</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;"><u>Nama</u> <u>aktor</u></p> </div>	<p>Merupakan orang atau sistem lain atau proses diluar sistem yang dibuat, yang berhubungan dengan sistem yang dibuat. Aktor belum tentu orang, walaupun memiliki simbol bergambar orang.</p>
2.	<p><i>Lifeline</i> (Garis Hidup)</p> 	<p>Merupakan garis hidup objek yang menerangkan kehidupan objek.</p>
3.	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"> <p>Nama objek : nama kelas</p> </div>	<p>Merupakan objek yang melakukan interaksi pesan</p>
4.	<p>Waktu aktif</p> 	<p>Merupakan simbol yang menyatakan bahwa objek dalam keadaan berinteraksi dan keadaan aktif, dan semua yang berhubungan dengan waktu aktif merupakan tahapan yang di lakukan selama dalam keadaan aktif.</p>

No.	Simbol	Deskripsi
5.	Pesan tipe <i>create</i> $\xrightarrow{\llbracket create \rrbracket}$	Merupakan pernyataan satu objek membuat objek lain.
6.	Pesan tipe <i>create</i> $\xrightarrow{\llbracket create \rrbracket}$	Merupakan pernyataan satu objek membuat objek lain.
7.	Pesan tipe <i>call</i> $\xrightarrow{1: \text{nama_metode()}}$	Merupakan pernyataan satu objek memanggil metode atau operasi pada objek lain atau diri sendiri.
8.	Pesan tipe <i>send</i> $\xrightarrow{1 : \text{masukan}}$	Merupakan pernyataan bahwa objek mengirimkan informasi atau masukan atau data ke objek lain.

2.2.5 Pengertian Kamus Data (*Data Dictionary*)

Menurut Sukamto (2022: 94) mengatakan bahwa “Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar cara penulisan)”.

Menurut Yunita *et al.*, (2023: 173), Kamus data merupakan suatu penjelasan tertulis secara lengkap dan jelas tentang suatu data dalam *database* sesuai dengan sistem yang dibuat. Kamus data dibuat untuk mengurangi duplikasi data (redudensi)”.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa kamus data merupakan suatu dokumen yang penting dalam pengelolaan data dan

pengembangan sistem informasi karena memberikan panduan yang jelas tentang struktur data serta aliran-aliran data yang terjadi dalam sistem.

Adapun symbol-simbol yang sering digunakan dalam kamus data dapat dilihat pada tabel berikut ini.

Tabel 2.5 Kamus Data

No.	Simbol	Keterangan
1.	=	Disusun atau terdiri atas
2.	+	Dan
3.	()	Opsional
4.	[]	Memilih salah satu alternatif
5.	**	Komentar
6.	@	Identifikasi atribut kunci
7.		Pemisah <i>alternative</i> simbol []

2.3. Pengertian Judul

Dalam pengertian ini, meliputi penjelasan tentang Pengertian *Aplikasi*, serta Aplikasi Inventaris Untuk Pegawai Berbasis *Web* Pada PT. Sugih Rahayu Bahagia Menggunakan *Framework Laravel*.

2.3.1 Pengertian Aplikasi

Menurut Ovan, dkk (2020:5) aplikasi berbasis web merupakan sebuah aplikasi yang dapat diakses menggunakan web browser atau penjelajah web melalui jaringan internet, yang memiliki keunggulan bahwa aplikasi berbasis web yang dapat dengan mudah diakses oleh pengguna tanpa harus melakukan instalasi.

Menurut Purnomo dalam Safaat (2020:87) Aplikasi merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur windows dan permainan (game) dan sebagainya. Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi.

Berdasarkan beberapa pendapat di atas, dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang dapat diakses melalui web browser tanpa perlu



instalasi, memudahkan pengguna dalam penggunaannya. Selain itu, aplikasi juga merujuk pada perangkat lunak yang dibuat untuk tujuan tertentu, seperti pengolahan dokumen, manajemen sistem, dan hiburan. Dalam konteks Android, sistem operasi ini memungkinkan aplikasi berjalan di perangkat mobile dengan berbagai fungsi yang didukung oleh sistem operasi berbasis Linux.

2.3.2 Pengertian Aplikasi Inventaris untuk Pegawai Berbasis *Web* pada PT. Sugih Rahayu Bahagia menggunakan *Framework Laravel*

Aplikasi Inventaris Untuk Pegawai Berbasis *Web* Pada PT. Sugih Rahayu Bahagia Menggunakan *Framework Laravel* yang akan di gunakan oleh Admin *Finance* dan *Manager* untuk mempermudah melakukan pencatatan barang dan memonitoring pada setiap stok barang yang di simpan.

2.4. Teori Program

Dalam teori ini, meliputi penjelasan tentang Pengertian *Vscode*, *Laragon*, *Mysql*, *Bootstrap*, *Html*, *Css*, *php* dan *laravel*.

2.4.1 Pengertian Visual Studio Code

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. 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*, *PHP*, dst. (Ummy Gusti Salamah, 2021 dalam Ningsih *et al.*, 2022: 95).

2.4.2 Pengertian Laragon

Laragon adalah sebuah aplikasi mirip seperti XAMPP, namun didesain untuk kebutuhan *developer PHP* yang menggunakan *framework Laravel*. *Service* yang *include* dalam Laragon seperti: *Apache*, *MySQL*, *PHP Server*, *Memchaced*, *Redis*, *Composer*, *Xdebug*, *PhpMyAdmin*, *Cmdr*, dan lainnya. Aplikasi ini sangat cocok digunakan oleh seorang *developer PHP* yang menggunakan *framework Laravel*, karena akan mempermudah dalam melakukan pengembangan aplikasi.



2.4.3 Pengertian MySQL

Menurut Wahyudi dan Jakara (2023: 24) MySQL (My Structure Query Language) adalah sebuah program berbasis DOS yang bersifat open Source. *MySQL* adalah produk yang berjalan pada *Platform* baik windows maupun Linux. Selain itu, *MySQL* merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk multi-user (banyak pengguna).

Menurut (Pane *et al.*, 2022: 94) *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi *MySQL* adalah database server yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

2.4.4 Pengertian Bootstrap

Menurut Firmansyah dan Herman (2023: 364) Bootstrap adalah salah satu kerangka kerja dari CSS yang digunakan untuk membantu *developer* dalam mempermudah, mempercepat dan memperindah tampilan website tersebut.

Menurut (Suprayogi dan Rahmanesa, 2019 dalam Suli dan Nirsal, 2023: 27), *Bootstrap* adalah *front-end framework* yang bagus dan luar biasa yang mengedepankan tampilan untuk *mobile device* (*Handphone, smartphone* dll) guna mempercepat dan mempermudah pengembangan *website*. *Bootstrap* menyediakan HTML, CSS dan *Javascript* siap pakai dan mudah untuk dikembangkan.

2.4.5 Pengertian HTML

Menurut Firmansyah dan Herman (2023: 364) HTML (atau singkatan dari HyperText Markup Language) merupakan salah satu bahasa markup standar yang digunakan untuk membuat sebuah halaman *website*.

Menurut Kaban dan Sembiring (2021: 10) HTML (*HyperText Markup Language*) adalah bahasa markup yang digunakan web browser untuk menafsirkan



dan menulis teks, gambar dan konten lainnya ke dalam halaman *web* secara visual maupun suara.

2.4.6 Pengertian CSS

Menurut Firmansyah dan Herman (2023: 364) CSS atau singkatan dari *Cascading Stylesheet* merupakan salah satu bahasa *stylesheet* yang digunakan untuk memperindah atau mempercantik sebuah tampilan *website*. Biasanya CSS membantu HTML dalam proses memperindah sebuah *website*.

2.4.7 Pengertian PHP

Menurut (Ningsih *et al.*, 2022: 95) PHP merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, maksudnya dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.

Menurut Firmansyah dan Herman (2023: 364) PHP atau singkatan dari *Hypertext Preprocessor* merupakan salah satu bahasa pemrograman *open source* yang digunakan untuk komunikasi ke sisi server yang kemudian transfer hasilnya ke *client* yang melakukan permintaan.

2.4.8 Pengertian Javascript

Menurut Hardianto *et al.* (2023: 116), (*JavaScript* adalah bahasa pemrograman yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Aplikasi *client* yang dimaksud merujuk kepada browser seperti Google Chrome, Mozilla Firefox, Opera Mini dan sebagainya.

Menurut Arisantoso *et al.*, (2023: 59), *Javascript* adalah bahasa pemrograman tingkat tinggi yang pertama kali dibuat untuk membuat *website* lebih hidup. *Javascript*, bersama dengan HTML dan CSS, adalah bahasa pemrograman yang paling umum digunakan untuk membuat aplikasi berbasis *website*. Bahasa ini dapat membuat *website* lebih interaktif.



2.4.9 Pengertian *Laravel*

Menurut Hermanto *et al.*, (2019: 19) CSS atau singkatan dari *Cascading Stylesheet* merupakan salah satu bahasa *stylesheet* yang digunakan untuk memperindah atau mempercantik sebuah tampilan *website*. Biasanya CSS membantu HTML dalam proses memperindah sebuah *website*.