



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Komputer adalah sebuah perangkat elektronik yang dapat menerima, menyimpan, dan memanipulasi data untuk menghasilkan *output* sesuai dengan instruksi yang diberikan oleh pengguna atau oleh program komputer (Tanenbaum, 2020:3).

Komputer adalah alat elektronik yang menerima *input* (data), memprosesnya sesuai dengan serangkaian instruksi yang ditentukan, menghasilkan *output*, dan menyimpan hasilnya untuk digunakan di masa mendatang (Stallings, 2021:2).

Berdasarkan pendapat para ahli di atas, maka dapat disimpulkan bahwa komputer adalah perangkat elektronik yang dapat menerima, memproses, dan menyimpan data berdasarkan instruksi, serta menghasilkan output, sesuai dengan instruksi pengguna.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Perangkat lunak (*software*) merupakan kumpulan instruksi yang ditulis dalam bahasa pemrograman tertentu untuk mengontrol operasi perangkat keras (*hardware*) dan menjalankan fungsi-fungsi tertentu (Tanenbaum, 2020:8).

Perangkat lunak (*software*) adalah program-program yang dirancang untuk menjalankan tugas-tugas khusus di komputer, termasuk sistem operasi, aplikasi produktivitas, permainan, dan lain-lain (Stallings, 2021:12).

Dari pandangan para ahli di atas, dapat disimpulkan bahwa perangkat lunak (*software*) adalah kumpulan instruksi yang ditulis dalam bahasa pemrograman tertentu untuk mengontrol operasi perangkat keras (*hardware*), menjalankan fungsi-fungsi tertentu, dan menjalankan tugas-tugas khusus di komputer, seperti sistem operasi, aplikasi produktivitas, permainan, dan lain-lain.



2.1.3 Pengertian Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) merupakan komponen fisik dari sebuah sistem komputer yang mencakup semua bagian yang dapat dilihat dan disentuh, seperti prosesor, memori, perangkat *input/output* (I/O), dan sebagainya (Tanenbaum, 2020:4).

Perangkat keras (*hardware*) terdiri dari komponen fisik sebuah sistem komputer, termasuk prosesor, memori, perangkat *input/output* (I/O), dan perangkat penyimpanan (Stallings, 2021:3).

Dengan mengacu pada dua definisi di atas oleh para ahli, dapat disimpulkan bahwa perangkat keras (*hardware*) merupakan komponen fisik dari sebuah sistem komputer yang mencakup semua bagian yang dapat dilihat dan disentuh, seperti prosesor, memori, perangkat *input/output* (I/O), dan perangkat penyimpanan.

2.1.4 Pengertian Basis Data (*Database*)

Database adalah kumpulan data yang terorganisir, yang biasanya disimpan dan diakses secara elektronik dari sistem komputer (Elmasri dan Navathe, 2023:5).

Database adalah kumpulan data yang terstruktur yang disimpan di komputer dan dapat diakses oleh program komputer untuk melakukan berbagai tugas (Elmasri dan Navathe, 2023:5).

Berdasarkan dua definisi yang telah disampaikan oleh para ahli di atas, dapat disimpulkan bahwa *database* adalah kumpulan data yang terorganisir dan terstruktur yang disimpan dan diakses secara elektronik dari sistem komputer, dan dapat diakses oleh program komputer untuk melakukan berbagai tugas.

2.1.5 Pengertian Internet

Internet adalah jaringan global yang terdiri dari jaringan komputer yang saling terhubung menggunakan protokol standar untuk berkomunikasi dan bertukar data (Tanenbaum dan Wetherall, 2021:12).



Internet adalah jaringan global yang terdiri dari jaringan komputer yang menggunakan protokol standar untuk saling terhubung dan berkomunikasi (Peterson dan Davie, 2024:15).

Dari dua pengertian yang telah disampaikan oleh para ahli tersebut, dapat diambil kesimpulan bahwa internet adalah jaringan global yang terdiri dari jaringan komputer yang saling terhubung menggunakan protokol standar untuk berkomunikasi, bertukar data, dan saling terhubung.

2.1.6 Pengertian Sistem

Sistem adalah suatu kesatuan elemen yang saling berinteraksi untuk mencapai tujuan tertentu (Stair, dkk., 2021:10).

Sistem adalah entitas yang terdiri dari komponen-komponen yang berinteraksi secara teratur untuk mencapai tujuan bersama (Donovan dan Slaughter, 2024:6).

Dari dua pandangan ahli di atas, dapat ditarik kesimpulan bahwa sistem adalah suatu kesatuan entitas yang terdiri dari komponen-komponen yang saling berinteraksi secara teratur untuk mencapai tujuan tertentu atau bersama.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Aplikasi adalah perangkat lunak yang dirancang untuk melakukan tugas-tugas tertentu, seperti mengelola data, menjalankan analisis, atau menyediakan layanan kepada pengguna akhir (Elmasri dan Navathe, 2023:15).

Aplikasi adalah program-program yang dirancang untuk melakukan tugas-tugas tertentu, seperti pengolahan data, analisis, atau pengolahan transaksi (Sudarshan, dkk., 2022:18).

Berdasarkan dua pendapat ahli di atas, dapat disimpulkan bahwa aplikasi adalah perangkat lunak atau program-program yang dirancang untuk melakukan tugas-tugas tertentu, seperti mengelola data, menjalankan analisis, atau menyediakan layanan kepada pengguna akhir, termasuk pengolahan data, analisis, atau pengolahan transaksi.



2.2.2 Pengertian Penjualan

Penjualan adalah proses pertukaran dimana pihak penjual menawarkan produk atau jasa kepada pihak pembeli dengan harapan mendapatkan keuntungan atau manfaat (Futrell, 2023:8).

Penjualan adalah proses interaksi antara penjual dan pembeli dengan tujuan untuk memenuhi kebutuhan dan keinginan pembeli dengan produk atau jasa yang ditawarkan (Ingram, dkk., 2022:12).

Dari pandangan para ahli di atas, dapat disimpulkan bahwa penjualan adalah proses pertukaran dimana pihak penjual menawarkan produk atau jasa kepada pihak pembeli dengan harapan mendapatkan keuntungan atau manfaat, serta merupakan interaksi antara penjual dan pembeli dengan tujuan untuk memenuhi kebutuhan dan keinginan pembeli dengan produk atau jasa yang ditawarkan.

2.2.3 Pengertian Manajemen Data

Manajemen data adalah praktik dan proses untuk mengelola sejumlah besar data, termasuk penyimpanan, pemeliharaan, pembaruan, dan akses data tersebut (Elmasari dan Navathe, 2023:5).

Manajemen data adalah disiplin yang berkaitan dengan penyimpanan, pengaturan, dan manipulasi data dalam sebuah sistem komputer (Sudarshan, dkk., 2022:8).

Berdasarkan dua pengertian diatas, dapat disimpulkan bahwa manajemen data adalah praktik dan proses untuk mengelola sejumlah besar data, termasuk penyimpanan, pemeliharaan, pembaruan, dan akses data tersebut, yang merupakan disiplin yang berkaitan dengan penyimpanan, pengaturan, dan manipulasi data dalam sebuah sistem komputer.

2.2.4 Pengertian Website

Website adalah koleksi halaman *web* yang saling terkait yang dapat diakses melalui internet (Marakas, 2021:20).

Website adalah kumpulan halaman *web* yang berisi informasi, grafik, video, atau konten lainnya yang dapat diakses melalui internet (Stair, 2024:18).

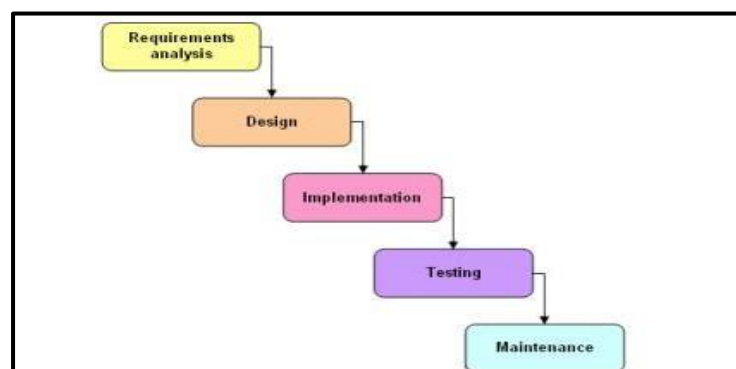
Berdasarkan pendapat para ahli yang telah disampaikan, kesimpulannya adalah bahwa *website* adalah koleksi halaman *web* yang saling terkait yang berisi informasi, grafik, video, atau konten lainnya, dan dapat diakses melalui internet.

2.3 Teori Khusus

2.3.1 Metode Pengembangan Sistem *Waterfall*

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*” 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 (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Wahid, 2020:2).

Menurut Adnyana & Asmarajaya (2022:2) “Model *waterfall* merupakan model klasik yang bersifat sistematis, berurutan dalam membangun perangkat lunak. Nama model ini sebenarnya adalah *Linear Sequinteal Model*. Model ini sering disebut juga dengan siklus hidup klasik atau metode air terjun”. Berikut adalah gambar ilustrasi model air terjun:



Gambar 2.1 Metode *Waterfall*

Sumber : Adnyana & Asmarajaya (2022:3)



2.3.2 Pengertian *Unified Modeling Language (UML)*

UML adalah bahasa *modeling* standar yang digunakan untuk menunjukkan struktur dan perilaku sistem, terutama dalam pengembangan perangkat lunak berbasis objek (Dennis, dkk., 2023:5).

UML adalah bahasa *modeling* grafis standar yang digunakan untuk merancang sistem berbasis objek, menunjukkan struktur, perilaku, dan interaksi antara komponen sistem (Larman, 2022:10).

Berdasarkan pengertian para ahli tersebut, dapat ditarik kesimpulan bahwa *UML* adalah bahasa *modeling* standar grafis yang digunakan untuk menunjukkan struktur, perilaku, dan interaksi antara komponen sistem, terutama dalam pengembangan perangkat lunak berbasis objek.

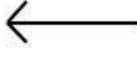
2.3.2.1 Pengertian *Use Case Diagram*

Diagram *Use ase* adalah representasi grafis dari interaksi antara aktor dan sistem, menunjukkan bagaimana sistem berperilaku dalam situasi tertentu (Dennis, dkk., 2023:85).

Diagram *Use Case* adalah diagram yang menunjukkan interaksi antara aktor (pengguna) dengan sistem, menunjukkan fungsionalitas sistem dari perspektif pengguna (Larman, 2022:120).

Dengan mengacu pada dua definisi di atas oleh para ahli, dapat disimpulkan bahwa Diagram *Use Case* adalah representasi grafis dari interaksi antara aktor (pengguna) dan sistem, menunjukkan bagaimana sistem berperilaku dalam situasi tertentu dan fungsionalitas sistem dari perspektif pengguna.

Tabel 2.1 Simbol-Simbol *Use case Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<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>).
3		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil

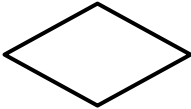
2.3.2.2 Pengertian *Activity Diagram*

Activity Diagram adalah diagram *UML* yang menggambarkan alur kerja (*workflow*) dari suatu proses atau aktivitas, menunjukkan urutan langkah-langkah dan pengambilan keputusan (Dennis, dkk., 2023:120).

Activity Diagram adalah diagram yang digunakan untuk memodelkan alur kerja (*workflow*) dari proses bisnis atau fungsi sistem, menunjukkan langkah-langkah, keputusan, dan aktivitas paralel (Larman, 2022:150).

Berdasarkan dua pengertian yang disebutkan oleh para ahli, dapat disimpulkan bahwa *Activity Diagram* adalah diagram *UML* yang digunakan untuk menggambarkan alur kerja (*workflow*) dari suatu proses atau aktivitas, menunjukkan urutan langkah-langkah, pengambilan keputusan, dan aktivitas paralel dari proses bisnis atau fungsi sistem.

Tabel 2.2 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 dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Interaksi/interaction 	Alur dari sebuah activity.

2.3.2.3 Pengertian *Class Diagram*

Diagram Kelas adalah diagram *UML* yang menggambarkan struktur statis dari sistem, menunjukkan kelas-kelas, atribut-atribut, dan hubungan antara kelas-kelas tersebut (Dennis, dkk., 2023:95).

Diagram Kelas adalah representasi grafis dari struktur statis dari sistem, menunjukkan kelas-kelas dalam sistem, atribut-atributnya, dan hubungan antara kelas-kelas tersebut (Larman, 2022:120).

Berdasarkan dua definisi yang telah disampaikan oleh para ahli di atas, dapat disimpulkan bahwa Diagram Kelas adalah diagram *UML* yang menggambarkan struktur statis dari sistem, menunjukkan kelas-kelas, atribut-atribut, dan hubungan antara kelas-kelas tersebut dalam representasi grafis.

Tabel 2.3 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Asosiasi berarah/<i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multicity</i> .

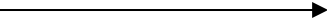
2.3.2.4 Pengertian *Sequence Diagram*

Sequence diagram adalah diagram *UML* yang menunjukkan interaksi antara objek dalam sistem dalam urutan waktu tertentu, menunjukkan pesan-pesan yang dikirim antara objek (Dennis, dkk., 2023:110).

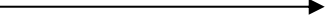
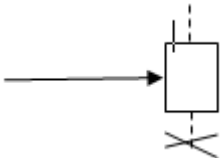
Sequence diagram adalah diagram yang menunjukkan bagaimana objek-objek dalam sistem berinteraksi satu sama lain dalam urutan waktu tertentu, menunjukkan pesan-pesan yang dikirim antara objek (Larman, 2022:135).

Dari dua pandangan ahli tersebut, dapat ditarik kesimpulan bahwa *Sequence diagram* adalah diagram *UML* yang menggambarkan interaksi antara objek dalam sistem dalam urutan waktu tertentu, menunjukkan bagaimana objek-objek berinteraksi dan pesan-pesan yang dikirim di antara mereka.

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Keterangan
Aktor  Tanpa waktu aktif Nama_aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun <i>symbol</i> dari <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.
Garis hidup/<i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek Nama objek : Nama kelas	Menyatakan objek yang berinteraksi pesan.
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan beriteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe <i>create</i>  <<<i>create</i>>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Lanjutan Tabel 2.4 Simbol *Sequence Diagram*

Pesan tipe <i>call</i> 1: nama_metode() 	Menyatakan sebuah objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i> 1: Masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe <i>return</i> 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.
Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i>.

2.3.3 Pengertian *Entity Relationship Diagram (ERD)*

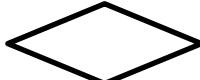
Entity Relationship Diagram (ERD) adalah representasi grafis dari entitas, atribut, dan hubungan yang menentukan struktur basis data. *ERD* digunakan untuk mengkonseptualisasikan dan merancang basis data pada tingkat tinggi, memberikan cetak biru untuk struktur basis data yang mendasarinya (Ramakrishnan dan Gehrke, 2020: 67)

Entity Relationship Diagram (ERD) adalah alat visual yang digunakan untuk memodelkan data yang diperlukan dalam basis data. *ERD* terdiri dari entitas,

yang mewakili objek yang menarik, dan hubungan, yang mewakili asosiasi antara entitas-entitas tersebut. *ERD* sangat penting dalam proses desain basis data untuk memahami dan mendokumentasikan kebutuhan data (Elmasri dan Navathe, 2019:81).

Dari dua pengertian di atas dapat disimpulkan bahwa *Diagram Entity-Relationship (ERD)* adalah alat visual yang digunakan untuk memodelkan dan merancang basis data. *ERD* terdiri dari entitas, atribut, dan hubungan yang menentukan struktur basis data. Alat ini sangat penting dalam proses desain basis data karena membantu dalam memahami, mengkonseptualisasikan, dan mendokumentasikan kebutuhan data pada tingkat tinggi, serta memberikan cetak biru untuk struktur basis data yang mendasarinya.

Tabel 2.5 Simbol *Entity Relationship Diagram*

Simbol	Keterangan
	Objek yang dapat diidentifikasi dalam lingkungan pemakai
	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.
	Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah).

2.3.4 Metode *Black Box Testing*

Black Box Testing adalah teknik pengujian perangkat lunak di mana sistem diuji tanpa memperhatikan struktur internal atau kode sumbernya. Pengujian ini berfokus pada fungsionalitas eksternal sistem (Chopra, 2022:87).



Black Box Testing adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memperhatikan struktur internal atau logika dari program. Pengujian ini berfokus pada input dan output yang diharapkan (Pressman dan Maxim, 2023:275).

Berdasarkan dua pandangan ahli di atas, dapat diambil kesimpulan bahwa *Black Box Testing* adalah teknik atau metode pengujian perangkat lunak di mana sistem atau program diuji tanpa memperhatikan struktur internal atau kode sumbernya, dengan fokus pada fungsionalitas eksternal, *input*, dan *output* yang diharapkan.

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.

2.3.5 Pengertian Kamus Data

Kamus Data adalah kumpulan definisi data yang digunakan dalam suatu organisasi, termasuk nama entitas, atribut, tipe data, dan hubungan antar entitas (Elmasri dan Navathe, 2023:75).

Kamus Data adalah kumpulan definisi data yang mencakup entitas, atribut, tipe data, dan batasan-batasan lainnya yang digunakan dalam suatu basis data (Sudarshan, dkk., 2022:85).

Berdasarkan dua definisi yang telah disampaikan oleh para ahli di atas, dapat disimpulkan bahwa Kamus Data adalah kumpulan definisi data yang mencakup nama entitas, atribut, tipe data, batasan-batasan, dan hubungan antar entitas yang digunakan dalam suatu organisasi atau basis data.



Kamus data biasanya berisi hal-hal sebagai berikut:

1. Nama-nama dari data
2. Digunakan pada proses-proses yang terkait data
3. Deskripsi data
4. Informasi tambahan seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

2.4 Teori Program

2.4.1 Pengertian *Visual Studio Code*

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*, 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 (Sole, 2019:2).

Visual Studio Code adalah *Integrated Development Environment (IDE)* yang dikembangkan oleh *Microsoft* untuk mempermudah *software developer* mengembangkan aplikasi pada platform milik *Microsoft* (Faisal, 2017:13).

Menurut pandangan para ahli, dapat disimpulkan bahwa *Visual Studio Code* merupakan *Integrated Development Environment (IDE)* pertama dalam keluarga *Microsoft Visual Studio* yang berjalan di *Windows*, *Linux*, dan *macOS*, adalah alat pengembangan *cross-platform* yang gratis dan *open source*.

2.4.2 Pengertian *Hypertext Markup Language (HTML)*

Menurut Wahyudi (2022:1), “*HTML (Hyper Text Markup Language)* adalah sebuah bahasa *formatting* yang digunakan untuk membuat sebuah halaman *website*. Dalam dunia pemrograman berbasis *website*, *HTML* menjadi pondasi dasar pada halaman *website*.”



HTML (Hypertext Markup Language) adalah bahasa markup standar untuk membuat halaman *web*. Ini mendefinisikan struktur konten sebuah halaman *web* dengan menggunakan elemen-elemen markup yang disebut *tags* (Robbins, 2022:5).

Dengan mengacu pada dua definisi di atas oleh para ahli, dapat disimpulkan bahwa *HTML (Hypertext Markup Language)* adalah bahasa *formatting* yang digunakan untuk membuat halaman *website*, yang mendefinisikan struktur konten sebuah halaman *web* dengan menggunakan elemen-elemen *markup* yang disebut *tags*, dan menjadi pondasi dasar dalam dunia pemrograman berbasis *website*.

2.4.3 Pengertian Cascading Style Sheets (CSS)

Menurut Wahyudi (2022:33), “CSS merupakan singkatan dari “*Cascading Style Sheet*”. Sesuai dengan namanya CSS memiliki sifat “*style shee language*” yang berarti bahasa pemrograman yang digunakan untuk *web design*. Dalam mendesain halaman *website*, CSS menggunakan penanda yang kita kenal dengan *id* dan *class*.”

CSS (*Cascading Style Sheets*) adalah bahasa yang digunakan untuk mengendalikan tampilan dan tata letak elemen-elemen *HTML* pada sebuah halaman *web* (Robbins, 2019:226).

Menurut pandangan para ahli, dapat disimpulkan bahwa CSS (*Cascading Style Sheets*) merupakan singkatan dari “*Cascading Style Sheet*”, yaitu bahasa pemrograman yang digunakan untuk *web design* dan mengontrol tampilan serta tata letak elemen-elemen *HTML* pada sebuah halaman *web* dengan menggunakan penanda yang dikenal sebagai *id* dan *class*.

2.4.4 Pengertian Hypertext Preprocessor (PHP)

PHP adalah bahasa skrip sisi server yang digunakan untuk membuat halaman *web* dinamis. Ia bekerja dengan menghasilkan *HTML* yang akan dikirim ke peramban pengguna, namun juga mampu melakukan berbagai tugas pemrosesan *server-side* (Zandstra, 2023:5).



PHP adalah bahasa skrip sisi server yang paling umum digunakan untuk membangun situs *web* dinamis. Ia memungkinkan pengguna untuk membuat halaman *web* yang dapat berubah secara dinamis berdasarkan masukan pengguna, data dari *database*, atau variabel lainnya (Thomson dan Welling, 2022:3).

Dari pandangan para ahli di atas, dapat disimpulkan bahwa *PHP* adalah bahasa skrip sisi server yang digunakan untuk membuat halaman *web* dinamis, yang bekerja dengan menghasilkan *HTML* yang dikirim ke peramban pengguna dan mampu melakukan berbagai tugas pemrosesan *server-side*. *PHP* adalah bahasa skrip sisi server yang paling umum digunakan untuk membangun situs *web* dinamis yang dapat berubah secara dinamis berdasarkan masukan pengguna, data dari *database*, atau variabel lainnya.

2.4.5 Pengertian My Structured Query Language (MySQL)

MySQL adalah salah satu sistem manajemen basis data relasional paling populer yang digunakan dalam pengembangan *web* modern (Zandstra, 2021:96).

MySQL adalah sistem manajemen basis data relasional *open-source* yang paling populer di dunia. Ini digunakan untuk menyimpan data dalam bentuk tabel yang dapat diakses oleh aplikasi *web* (Powers, 2021:3).

Berdasarkan pendapat para ahli yang telah disampaikan, kesimpulannya adalah bahwa *MySQL* adalah salah satu sistem manajemen basis data relasional paling populer dan *open-source* yang digunakan dalam pengembangan *web* modern, menyimpan data dalam bentuk tabel yang dapat diakses oleh aplikasi *web*.

2.4.6 Pengertian CodeIgniter 4

CodeIgniter 4 adalah *framework PHP* yang kuat yang memungkinkan pengembang untuk membangun aplikasi *web* dengan cepat dan efisien. Ini menawarkan berbagai fitur termasuk arsitektur yang *modular*, sistem *routing* yang kuat, dan berbagai perpustakaan bawaan yang membantu dalam pengembangan aplikasi *web* yang aman dan *scalable* (Islam, 2021:1).



CodeIgniter 4 adalah *framework PHP* yang dirancang untuk mempermudah pengembangan aplikasi *web*. Dengan struktur kode yang bersih dan *modular*, serta banyaknya fitur bawaan yang kuat, *CodeIgniter 4* memungkinkan pengembang untuk membangun aplikasi *web* dengan cepat dan efisien (Risky, 2022:3).

Berdasarkan pendapat para ahli di atas, maka dapat disimpulkan bahwa *CodeIgniter 4* adalah *framework PHP* yang kuat yang memungkinkan pengembang untuk membangun aplikasi *web* dengan cepat dan efisien. Dengan arsitektur *modular*, sistem *routing* yang kuat, dan berbagai perpustakaan bawaan.

2.4.7 Pengertian Laragon

Laragon adalah sebuah lingkungan pengembangan web lokal yang ringan, cepat, dan modern. Alat ini memungkinkan pengguna untuk mengubah komputer mereka menjadi server yang kuat dalam hitungan menit. Laragon mendukung berbagai aplikasi seperti WordPress, Joomla, Magento, Drupal, dan Prestashop dengan performa yang sangat tinggi (Upendran, 2023).

Laragon adalah aplikasi untuk membuat program atau aplikasi di server lokal (komputer). Salah satu kelebihan Laragon adalah fitur yang lebih lengkap dibanding aplikasi lain dan kemudahan penggunaannya, karena menggunakan tampilan GUI (Puguh, 2023).

Berdasarkan dua pengertian diatas dapat di simpulkan bahwa *laragon* adalah salah satu aplikasi pengembangan web lokal yang efisien dan modern, dirancang untuk mengubah komputer menjadi server yang kuat dengan cepat. Alat ini mendukung aplikasi seperti serta memungkinkan pengembangan aplikasi *web* dengan performa tinggi. Laragon juga dikenal karena fitur-fitur yang lengkap dan kemudahan penggunaan berkat antarmuka *GUI*-nya, yang memudahkan pengembang dalam membuat program atau aplikasi di server lokal.

2.4.8 Pengertian Bootstrap

Bootstrap adalah kerangka kerja *front-end* yang populer untuk pengembangan situs *web responsif*. Ini menyediakan serangkaian komponen,



gaya, dan skrip *JavaScript* yang siap pakai untuk mempercepat proses pengembangan. *Bootstrap* memungkinkan pengembang untuk membuat situs web yang terlihat menarik dan *responsif* dengan mudah (Risky, 2022:2).

Bootstrap adalah kerangka kerja *front-end* yang paling populer untuk membangun situs web *responsif* dan *mobile-first*. Ini menyediakan serangkaian komponen *UI*, gaya, dan skrip *JavaScript* yang dapat digunakan untuk mempercepat pengembangan web. *Bootstrap* memungkinkan pengembang untuk membuat situs web yang terlihat modern dan menarik dengan mudah (Traversy, 2021:1).

Berdasarkan dua pengertian diatas, dapat disimpulkan bahwa *Bootstrap* adalah kerangka kerja *front-end* yang paling populer untuk pengembangan situs web *responsif* dan *mobile-first*, menyediakan serangkaian komponen *UI*, gaya, dan skrip *JavaScript* yang siap pakai untuk mempercepat proses pengembangan. Dengan *Bootstrap*, pengembang dapat dengan mudah membuat situs web yang terlihat menarik, modern, dan *responsif*.