



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Harmayani (2021) Komputer adalah suatu perangkat elektronik yang dapat digunakan untuk mengolah data sesuai dengan prosedur yang telah dirumuskan sebelumnya sehingga menghasilkan informasi bermanfaat bagi penggunanya.

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 (Hanafri et al., 2019).

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

2.1.2 Pengertian Perangkat Lunak

Menurut Alifah et al., (2023) "Perangkat lunak adalah perintah program komputer yang bila di eksekusi memberikan fungsi dan unjuk kerja seperti yang di inginkan". Perangkat lunak adalah obyek tertentu yang dapat dijalankan seperti kode sumber, kode objek, atau sebuah program yang lengkap (Pratama & Iskandar, 2020).

Berdasarkan beberapa definisi di atas, dapat disimpulkan bahwa perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi dan sebagai penghubung antara pengguna dengan komputer.



2.1.3 Pengertian Perangkat Keras

Menurut Shiddiq (2022) *Hardware* atau perangkat keras merupakan salah satu komponen utama dalam sebuah sistem komputer. *Hardware* terdiri dari berbagai macam perangkat seperti *input device*, *output device*, *storage device*, *processing device*, dan *peripheral device*. *Hardware* merupakan elemen yang penting dalam komputer, karena ia berfungsi sebagai pendukung sistem operasi dan aplikasi yang dijalankan di dalamnya. Tanpa *hardware*, komputer tidak akan bisa berjalan dan melayani kebutuhan pengguna. Oleh karena itu, pemahaman mengenai hardware sangat penting bagi para pengguna komputer, mulai dari pengguna pribadi hingga profesional IT.

2.1.4 Pengertian Basis Data (*Database*)

Menurut Jamaludin et al. (2022) bahwa basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Menurut Prayoga et al. (2023) Basis data adalah kumpulan data terorganisir yang disimpan dalam sistem komputer, dan DBMS bertanggung jawab untuk menyediakan metode penyimpanan, pengambilan, pembaruan, dan penghapusan data dalam basis data.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa basis data atau *database* adalah suatu perangkat lunak atau sistem yang untuk menyimpan sekumpulan data secara sistematis di perangkat penyimpanan komputer agar dapat diolah atau diproses oleh perintah-perintah tertentu.



Berikut merupakan contoh struktur tabel dalam basis data:

Tabel 2.1 Struktur Tabel dalam Basis Data

NO	Field	Type	Size	Description
1.	id_transaksi	<i>int</i>	8	<i>Primary key</i>
2.	id_pemesanan	<i>int</i>	8	<i>Foreign key</i>
3.	bukti_tf	<i>varchar</i>	20	-
4.	status_tf	<i>enum</i>	-	-

Penjelasan Mengenai struktur tabel dalam basis data diatas adalah sebagai berikut.

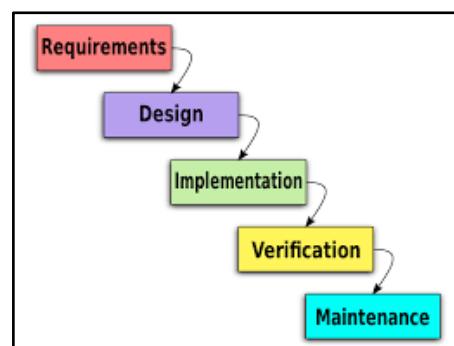
1. Field adalah kolom dalam tabel yang dapat menyimpan satu jenis informasi. Contoh field yaitu id_transaksi, id_pemesanan, bukti_tf, status_tf.
2. Type atau tipe data adalah karakteristik yang menentukan jenis data yang dapat disimpan dalam sebuah field. Contoh id_transaksi memiliki tipe data integer (bilangan bulat).
3. Size mengacu pada panjang atau kapasitas maksimum dari sebuah field. Contoh tipe data integer (int) memiliki ukuran maksimum 8 karakter.
4. Primary key adalah field yang memiliki nilai unik untuk setiap baris data dalam tabel. Primary key memastikan bahwa tidak ada duplikat data dalam tabel.
5. Foreign key adalah field yang menunjukkan hubungan atau relasi antara satu tabel dengan tabel lain. Foreign key digunakan untuk membuat hubungan antara dua tabel, dengan mengaitkan nilai field di satu tabel dengan nilai field yang sesuai di tabel lain.



2.1.5 Metode Pengembangan Sistem

Metode waterfall merupakan metode pengembangan sistem, metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing/verification, dan maintenance. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu disebut waterfall atau air tejun (Danny, 2022)

Menurut Danny (2022) secara umum, ada lima tahapan dalam metode waterfall, seperti digambar berikut:



Gambar 2.1 Tahapan Metode *Waterfall*

Sumber: Danny (2022)

a. Requirement (Analisis Kebutuhan)

Requirement adalah proses analisa atau pengumpulan data-data yang berkaitan dengan sistem yang akan dibuat. Pengumpulan data ini bisa dilakukan dengan wawancara, studi literatur, observasi atau penelitian langsung. Dalam fase ini tim analis akan menggali informasi sebanyak-banyaknya dari klien atau user tentang software apa yang mereka inginkan beserta dengan kebutuhan sistem lainnya.

b. Design System (Desain Sistem)

Proses ini akan berfokus pada pembangunan struktur data, arsitekur perangkat lunak, perancangan interface, perancangan fungsi internal dan eksternal serta detail



dari setiap algoritma prosedural. Tahapan design akan menghasilkan dokumen bernama “Software Requirement” yang nantinya menjadi landasan para programmer dalam membuat code-code aplikasi.

c. Implementation (Pelaksanaan)

Tahap ini adalah tahapan pembuatan aplikasi oleh para programmer dengan menggunakan kode-kode bahasa pemrograman tertentu. Proses penulisan sinkode (coding) aplikasi mengacu pada dokumen-dokumen yang telah dibuat sebelumnya.

d. Verification (Verifikasi)

Tahapan verifikasi meliputi pengintegrasian sistem dan juga melakukan testing terhadap aplikasi yang telah dibuat. Sistem akan diverifikasi untuk diuji sejauh mana kelayakannya. Dalam tahapan ini semua modul yang dikerjakan oleh programmer berbeda akan digabungkan kemudian diuji apakah telah sesuai dengan spesifikasi yang ditetapkan atau terdapat kesalahan/error dalam sistem sebelum kemudian diperbaiki ulang.

e. Maintenance (Pemeliharaan)

Tahapan ini umumnya meliputi tahapan penginstalasian perangkat lunak dan pengujian aplikasi. Maintenance juga adalah bentuk tanggung jawab tim pengembang untuk memastikan aplikasi dapat berjalan lancar setelah diserahkan kepada klien dalam periode waktu tertentu.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Hendriyati & Yusta (2021) “Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung guna melakukan suatu tugas yang diharapkan pengguna”. Di samping itu, menurut Dicapriyo et al. (2021) “Aplikasi ialah suatu hasil dalam bentuk perangkat lunak yang difungsikan untuk pemanfaatan komputer dari user atau pengguna, siap pakai, yang digunakan untuk mendapatkan pencapaian yang sesuai akurat dan dengan tujuan atau fungsi aplikasi itu dibuat”. Berdasarkan kutipan diatas, dapat disimpulkan bahwa Aplikasi adalah program yang digunakan untuk menentukan



aktivitas pemrosesan informasi yang berupa instruksi-instruksi dan mengendalikan komputer.

2.2.2 Pengertian *Monitoring*

Menurut Jakaria & Solahudin (2019) *Monitoring* adalah pemantauan yang dapat dijelaskan sebagai kesadaran tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

2.2.3 Pengertian *Website*

Menurut Joseph (2021) “*Website* adalah halaman web lengkap yang terdapat dalam domain yang berisi data. Sebuah situs online biasanya dibangun di atas banyak halaman web yang saling berhubungan. Hubungan antara satu halaman web dengan halaman web lainnya disebut *hyperlink*, sedangkan konten yang digunakan sebagai media penghubung disebut *hyper*”.

Sedangkan menurut Haryanto et al. (2021) menyatakan “*Website* adalah kumpulan-kumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan satu sama lain serta dapat diakses secara luas melalui halaman depan (*home page*) menggunakan sebuah browser dan juga menggunakan URL *website*”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa *Website* adalah kumpulan halaman suatu domain di internet yang berisi informasi data digital yang terhubung antar halamannya.



2.2.3 Pengertian Pemeliharaan

Menurut Hidayah & Ahmadi (2019) "Pemeliharaan merupakan suatu kegiatan yang diarahkan pada tujuan untuk menjamin kelangsungan fungsional suatu sistem yang berjalan sehingga dari sistem itu dapat diharapkan menghasilkan output sesuai dengan yang dikehendaki".

Pemeliharaan dilakukan untuk memelihara atau menjaga fasilitas atau peralatan pabrik dan mengadakan perbaikan atau penyesuaian atau penggantian yang diperlukan agar terdapat suatu keadaan operasi kegiatan yang memuaskan sesuai dengan apa yang direncanakan.

2.2.4 Pengertian Peralatan

Peralatan adalah benda yang dipakai sebagai sarana untuk mencapai tujuan. Peralatan yang dimaksud adalah alat yang digunakan sebagai sarana dan prasarana PDKB GI dalam mengerjakan pekerjaannya. Alat yang digunakan PDKB GI di PT PLN (Persero UPT Palembang) terdiri dari banyak jenis dan ukuran yang jumlah mencapai ratusan.

2.2.5 Pengertian Aplikasi Monitoring dan Pemeliharaan Peralatan Gardu Induk Pekerjaan Dalam Keadaan Bertegangan (PDKB) di PT PLN (Persero) UPT Palembang Berbasis Website

Pengertian Aplikasi Monitoring dan Pemeliharaan Peralatan Gardu Induk Pekerjaan Dalam Keadaan Bertegangan (PDKB) di PT PLN (Persero) UPT Palembang Berbasis Website adalah Aplikasi yang akan dibangun untuk memudahkan PDKB GI dalam hal memonitoring dan memelihara peralatan kerja dengan lebih mudah, efektif dan akurat.

2.3 Teori Khusus.

2.3.1 Pengertian Kamus Data

Menurut Kristanto (dalam Gunawan et al., 2021) "Kamus data adalah kumpulan-kumpulan elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasi setiap *field* atau file di dalam sistem".



Di samping itu, menurut (Widyaningsih et al., 2023) “Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa Kamus Data merupakan kumpulan simbol-simbol atau katalog fakta data yang digunakan untuk pengidentifikasi *field* dalam suatu sistem informasi. Kristanto (2019:72) menyatakan simbol-simbol yang ada dalam kamus data adalah sebagai berikut.

Tabel 2.2 Tabel Simbol-simbol Kamus Data

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

Sumber: Kristanto (2018:72)

2.3.2 Pengertian *Unified Modelling Language* (UML)

Menurut Nugroho (dalam Rusmawan, 2019), menyatakan bahwa, *Unified Modelling Language* (UML) adalah metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*), dan beberapa metoda lainnya. UML juga merupakan metodologi yang paling sering digunakan saat ini untuk analisis dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek (OOP).

Sedangkan menurut Booch (dalam (Rusmawan, 2019) menyatakan “UML adalah bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artefak dari software intensive system”.



Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa *Unified Modelling Language* (UML) adalah bahasa standar dan metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*), dan beberapa metoda lainnya yang digunakan untuk analisis dan perancangan sistem.

Menurut (Rusmawan, 2019) menyatakan tujuan dan fungsi dari UML antara lain sebagai berikut.

- 1) Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
- 2) Dapat menyatukan praktik-praktik terbaik yang ada dalam pemodelan
- 3) Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
- 4) Dapat berguna sebagai blue print sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
- 5) Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (software) saja.
- 6) Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.3.3 Pengertian Diagram *Use Case*

Menurut Kurniawan (2020) bahwa Diagram *Use Case* merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case diagram terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, keras, ataupun perangkat sistem lain, yang berinteraksi. Jadi *Use Case* adalah diagram untuk mendeskripsikan



sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Berikut merupakan simbol-simbol Diagram *Use Case*:

Tabel 2.3 Tabel Simbol-simbol Diagram *Use Case*

No	Simbol	Nama	Keterangan
1.		Actor	Seseorang apa saja yang berhubungan dengan sistem yang sedang dibangun.
2.		Use Case	Menggambarkan bagaimana menggunakan sistem.
3.		Relasi Asosiasi	Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan use case.
4.		Relasi Include	Memungkinkan satu use case menggunakan fungsionalitas yang disediakan oleh use case lainnya.
5.		Relasi Extend	Memungkinkan suatu use case secara optional menggunakan fungsionalitas yang disediakan oleh use case lainnya.

Sumber: Habiluddin (2011)

2.3.4 Pengertian Diagram *Activity*

Menurut Kurniawan (2020) bahwa diagram *Activity* adalah representasi visual dari berbagai aliran aktivitas dalam suatu sistem yang sedang dirancang. Diagram ini menunjukkan bagaimana setiap aliran dimulai, keputusan yang mungkin terjadi, dan bagaimana aliran tersebut berakhir. Selain itu, activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi selama beberapa eksekusi.

Sedangkan menurut Nugroho (dalam Rusmawan, 2021:79) “Activity Diagram digunakan untuk menampilkan rangkaian kegiatan, menunjukkan alur



kerja dari suatu titik awal ke titik akhir keputusan, merinci banyak jalur yang ada dalam perkembangan peristiwa yang terkandung dalam kegiatan”.

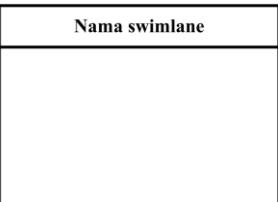
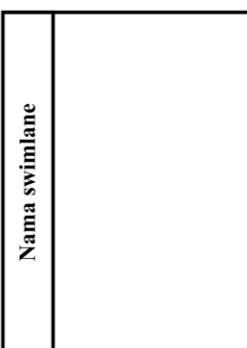
Berdasarkan kedua kutipan diatas, *Activity Diagram* adalah diagram aktivitas yang menggambarkan *workflow* (aliran kerja) dari suatu titik awal ke titik akhir keputusan dari sebuah sistem yang ada pada perangkat lunak. Berikut merupakan simbol-simbol Diagram *Activity*:

Tabel 2.4 Tabel Simbol-simbol Diagram *Activity*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah awal
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



Lanjutan **Tabel 2.4** Tabel Simbol-simbol Diagram Activity

	Swinmlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
6.	<p style="text-align: center;">Nama swimlane</p>  <p style="text-align: center;">Atau</p> 	

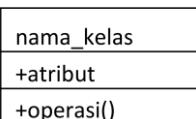
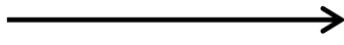
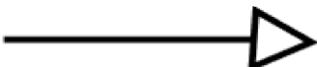
Sumber: Habiluddin (2011)

2.3.5 Pengertian Diagram Class

Menurut Kurniawan (2020) bahwa Diagram *Class* digunakan untuk melakukan visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak digunakan. *Class* diagram juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain atau *logical view* dari suatu sistem. Selama proses desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Berdasarkan definisi diatas, dapat disimpulkan bahwa Class Diagram adalah model statis dari struktur sistem yang menunjukkan kelas-kelas yang akan dibuat dalam membangun sistem. Berikut merupakan simbol-simbol diagram *Class*:

**Tabel 2.5** Tabel Simbol-simbol Diagram *Class*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	antarmuka / <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4.	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> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber: Habiluddin (2011)

2.3.6 Pengertian Diagram *Sequence*

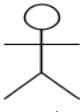
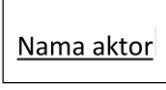
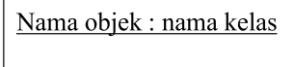
Menurut Kurniawan (2020) bahwa diagram *Sequence* digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang berupa pesan-pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait”). Sedangkan menurut Booch (dalam Rusmawan, 2019:84) “*Sequence Diagram*



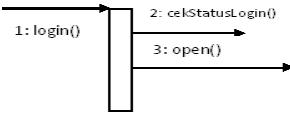
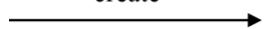
merupakan *Interaction Diagram* yang digunakan untuk menjelaskan eksekusi sebuah skenario semantik. *Sequence Diagram* juga digunakan untuk menjelaskan interaksi antar objek dalam urutan waktu”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa *Sequence Diagram* adalah *Interaction Diagram* yang menggambarkan kelakuan objek pada use case atau menjelaskan interaksi antar objek. Berikut merupakan simbol-simbol Diagram *Sequence*:

Tabel 2.6 Tabel Simbol-simbol Diagram *Sequence*

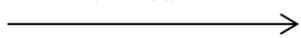
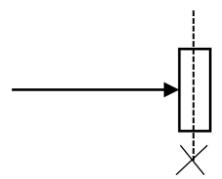
No	Simbol	Deskripsi
1.	Actor  Atau  tanpa waktu aktif	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal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

Lanjutan **Tabel 2.6** Tabel Simbol-simbol Diagram Sequence

4.	Waktu aktif 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status <i>Login()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> aktor tidak memiliki waktu aktif.</p>
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>1: <i>nama_metode()</i> arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai</p>



Lanjutan **Tabel 2.6** Tabel Simbol-simbol Diagram Sequence

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

Sumber: Habiluddin (2011)



2.4 Teori Program

2.4.1 Pengertian PHP (*Hypertext Processor*)



Gambar 2. 2 Logo PHP

Sumber: php net (2022)

Menurut Tukino (2023) Bahasa PHP merupakan bahasa pemrograman server-side yang digunakan untuk pengembangan web dinamis, kode PHP yang dieksekusi di sisi server untuk menghasilkan konten dinamis dalam dokumen HTML.

Menurut Ningsih et al. (2022) 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. PHP/FI merupakan nama awal dari PHP. PHP adalah Personal Home Page, FI adalah Form Interface. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web.

Dengan demikian dapat disimpulkan bahwa PHP merupakan skrip yang dapat membuat halaman web dikendalikan oleh data.



2.4.2 Pengertian Xampp



Gambar 2.3 Logo Xampp

Sumber : Stickpng.com (2022)

XAMPP adalah sebuah software web server apache yang didalamnya sudah tersedia database server MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa modul lainnya (Dinata dalam Ningsih et al., 2022)

2.4.3 Pengertian Laravel



Gambar 2.4 Logo Laravel

Sumber: WikemediaCommons (2022)

Laravel adalah sebuah framework aplikasi web berbasis PHP yang dirancang untuk memudahkan pengembangan aplikasi web dengan menyediakan berbagai fitur dan alat yang kuat. Framework ini dikembangkan dengan tujuan utama untuk



menyederhanakan tugas-tugas umum dalam pengembangan web seperti routing, caching, dan autentikasi, sehingga para pengembang sdapat fokus pada logika bisnis inti aplikasi mereka .(Siswanto, 2023).

2.4.4 Pengertian MySQL



Gambar 2.5 Logo MySQL

Sumber: Logo.wine (2022)

Menurut Tukino (2023) Mysql merupakan Sistem manajemen database relasional (RDBMS) yang *open-source* dan populer. sistem manajemen database relasional ini yang sering digunakan bersama dengan bahasa PHP.

MySQL adalah suatu perangkat lunak database relasi atau Relational Database management sistem (RDBMS) yang didistribusikan gratis di bawah lisensi GPL (*General Public License*). Dimana setiap orang bebas menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang dijadikan closed source atau komersial (Dinata dalam Ningsih et al., 2022) .



2.4.5 Pengertian HTML (HyperText Markup Language)



Gambar 2.6 Logo HTML

Sumber: Wikipedia.org (2022)

Menurut (Tukino, 2023) HTML (*Hypertext Markup Language*) atau Bahasa markup yang digunakan untuk membuat struktur dan konten dasar sebuah halaman web.

HTML ini sendiri digunakan untuk membantu merancang struktur dasar halaman website atau bila dianalogikan HTML merupakan pondasi awal untuk menyusun berdirinya kerangka halaman website secara lebih terstruktur sebelum masuk ke tahap desain dan sisi fungsionalitas. HTML nantinya akan dikolaborasikan dengan Bahasa Pemrograman CSS (Sari et al., 2022).

Berdasarkan beberapa defnisi diatas dapat disimpulkan bahwa HTML adalah bahasa markup yang digunakan untuk membuat struktur dan konten halaman web. HTML terdiri dari tag-tag yang digunakan untuk menandai elemen-elemen halaman web, seperti teks, gambar, tabel, dan link. Kolaborasi HTML dan CSS dapat digunakan untuk membuat halaman web yang menarik dan informatif, serta dapat diakses oleh siapa saja melalui browser internet.



2.4.6 Pengertian CSS (*Cascading Style Sheets*)



Gambar 2.7 Logo CSS

Sumber: WikimediaCommons (2022)

Menurut Tukino (2023) CSS (*Cascading Style Sheets*): Bahasa yang digunakan untuk mengendalikan tampilan dan gaya elemen dalam halaman web. Bahasa yang digunakan untuk mengatur tampilan dan gaya halaman web, seperti warna, font, dan tata letak.

CSS (*Cascading Style Sheet*) adalah bahasa pemrograman untuk memberikan tampilan desain yang akan digunakan pada web seperti warna, font, outline, background, menyesuaikan tampilan website dengan ukuran layar, dsb. CSS digunakan pada pembuatan website ini adalah untuk berkolaborasi dengan HTML agar dapat menghasilkan tampilan website yang menarik (Sari et al., 2022).

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa pengertian CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan elemen-elemen HTML. CSS terdiri dari aturan-aturan yang digunakan untuk menentukan format elemen-elemen HTML, seperti ukuran, warna, font, dan tata letak. CSS dapat digunakan untuk membuat halaman web yang lebih menarik dan informatif.



2.4.7 Pengertian Visual Studio Code



Gambar 2.8 Logo Visual Studio Code

Sumber: aemics.nl (2023)

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 dalam Ningsih et al., 2022)