



## **B AB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

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

Menurut Aditama et al (2022), “Komputer adalah kumpulan objek yang dapat ditemukan di berbagai tempat, seperti di kantor atau rumah, sebagai alat untuk melakukan berbagai tugas”.

Menurut Hanafri et al (2019), “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 didalam memori”.

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

Menurut Syaputra (2019), “Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (customer), pelanggan (customer) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (software) dari pengembangan perangkat lunak”.

Menurut Ferdiansyah et al (2023), “Perangkat lunak adalah sebuah jembatan penghubung antara pengguna dengan komputer yang nantinya akan dijalankan melalui sebuah program”.

##### **2.1.3 Pengertian Data**

Menurut Wahono & Ali (2021), “Data adalah bahan mentah yang perlu dilakukan pengolahan sehingga menghasilkan informasi atau keterangan, baik kualitatif maupun kuantitatif yang menunjukkan fakta sehingga dapat memberi manfaat bagi peneliti atau memberi gambaran kepada peneliti tentang kondisi atau suatu keadaan. Sedangkan informasi adalah sekumpulan data yang sudah diolah sehingga menghasilkan suatu analisa untuk digunakan oleh pihak yang membutuhkan”.



Menuru Dewi (2020), “Data adalah deskripsi dari suatu benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar dan belum diolah menjadi sebuah informasi”.

### **2.1.5 Pengertian Internet**

Menurut Hidayanto & Ilmi (2015), “Internet adalah sistem jaringan komputer global yang saling terhubung menggunakan protokol internet (TCP/IP) untuk menghubungkan perangkat di komputer di seluruh dunia. Internet adalah jaringan dari banyak jaringan yang terdiri dari jaringan pribadi, umum, akademik, bisnis, dan pemerintah lokal”.

Menurut Wiwi & Syahlanisyiam (2022), “internet adalah sebuah jaringan komunikasi global yang terbuka, Internet adalah salah satu bukti dari kecanggihan teknologi yang memberikan dampak positif ataupun negatif tergantung pada penggunaanya”.

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

Menurut Ansori (2019), “Perangkat keras komputer (*hardware*) adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, seperti: *Motherboard, Power Supply, Prosesor, RAM, Hardisk, CD Drive, Battery CMOS, VGA Card, Sound Card, dll*”.

Menurut Karisman (2019), “perangkat keras proses yang berfungsi untuk mengolah data atau mengeksekusi suatu perintah terhadap data yang dimasukan”.

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

Menurut Sudarso (2022), “Basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (perusahaan/organisasi) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”.



Menurut Chandra (2015), “Basis data merupakan data yang dapat didesain dan berintegrasi sehingga dapat memenuhi kebutuhan user dalam perusahaan atau organisasi”.

## **2.2 Teori Khusus**

### **2.2.1 Metode Pengembangan Sistem**

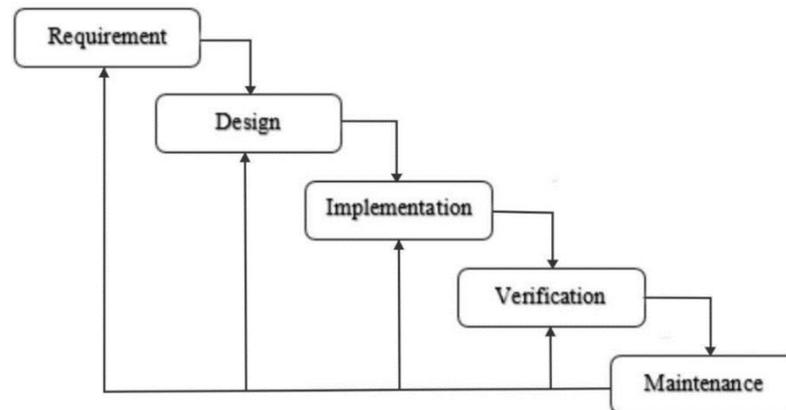
Stefanus Mario (2020), “Model Waterfall merupakan model yang mengembangkan sistem informasi secara sistematis dan sekuensial terdiri dari beberapa tahapan”.

Pricillia Titania (2021), “Model waterfall adalah model pengembangan perangkat lunak yang paling sering digunakan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa Metode Waterfall adalah suatu proses atau metode pengembangan sistem secara berurutan yang mengalir ke bawah (seperti air terjun) melalui tahapan-tahapan Requirement Analysis, Design System, Implementation, Verification & Testing, dan Maintenance.



Tahapan metode waterfall dapat dilihat pada gambar di bawah ini:



**Gambar 2.1** Tahapan - tahapan Metode Waterfall

Sumber: Sanubari, dkk (2020:41)

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

1. Requirement Analysis

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu di dokumentasikan.

2. Design

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan program pada tahap selanjutnya. Desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.

3. Implementation



Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

#### 4. Testing

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

#### 5. Operation & Maintenance

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

### 2.2.2 Pengujian Aplikasi

Blackbox testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat.

Ningrum Cahya (2019) menjelaskan, bahwa Metode Blackbox Testing adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail software. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai.

Taufik Hidayat (2019) menjelaskan, bahwa Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.



Berdasarkan beberapa definisi di atas, maka dapat disimpulkan bahwa *Black Box Testing* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara virtual ke setiap tingkat pengujian perangkat lunak, unit, integrasi, sistem, dan penerimaan.

### 2.2.3 Kamus Data

Menurut Premana (2019), “Kamus data adalah penjabaran dari aliran-aliran data yang ada di dalam sebuah data flow diagram. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem”.

Menurut Handoyo (2019), “Kamus data atau data dictionary adalah katalog fakta tentang data dan kebutuhan - kebutuhan informasi dari suatu sistem informasi. Dengan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap”.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.1** Simbol - simbol Pada Kamus Data

No.	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	dan
3	[   ]	baik ... atau ...
4	{ }"	n kali diulang/ bernilai banyak
5	( )	data opsional

(Sumber: (Sukamto dan Shalahuddin)

Kamus data ini sangat membantu analisis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur.



#### 2.2.4 Pengertian Unified Modeling Language (UML)



**Gambar 2.2** Logo UML

*Sumber: [https://id.wikipedia.org/wiki/Berkas:UML\\_logo.svg](https://id.wikipedia.org/wiki/Berkas:UML_logo.svg)*

Menurut Nugraha et al (2018), “UML (Unified Modeling Language) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

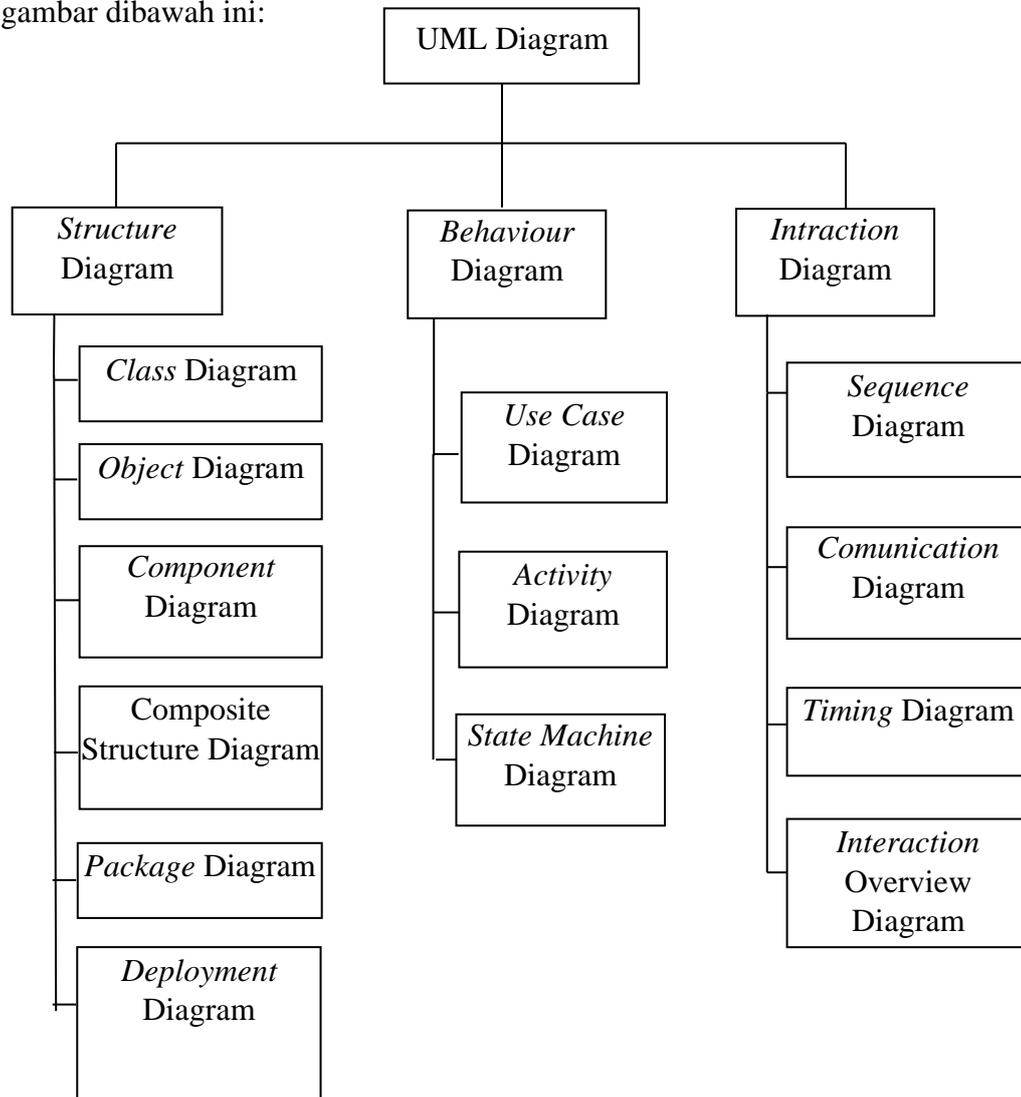
Menurut Yusuf et al (2021), “UML merupakan notasi untuk pemodelan desain berorientasi objek. Diagram-diagram UML di antaranya adalah: use case diagram, class diagram, activity diagram, object diagram, package diagram, dan sequence diagram”.

#### 2.2.5 Klasifikasi Diagram UML (*Unified Modelling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung.



Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



**Gambar 2.3** Klasifikasi Diagram UML (Undefined Modelling Language)

(Sumber: Sukamto dan Shalahuddin)

## 2.2.6 Jenis-Jenis Diagram *Unified Modeling Language* (UML)

### 2.2.6.1 Pengertian *Use Case Diagram*

Menurut Munandar et al (2020), “Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada



didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu”.

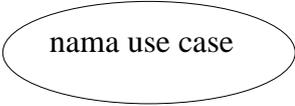
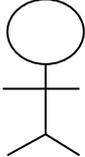
Menurut Kurniawan et al (2021), “Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor”.

Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *use case*.

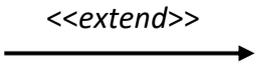
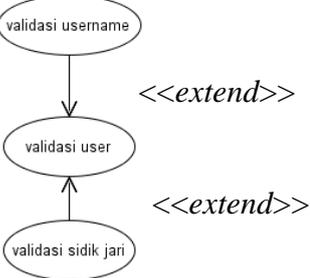
- a. **Aktor** merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

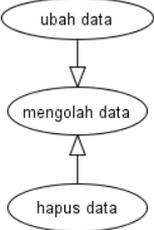
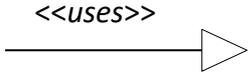
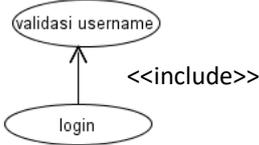
**Tabel 2.2** Simbol - simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.	<p><i>Use Case</i></p>  <p>nama use case</p>	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.	<p>Aktor / actor</p>  <p>nama aktor nama_interface</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu -

Lanjutan Tabel 2.2 Simbol-Symbol *Use Case* Diagram

No	Simbol	Deskripsi
		merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan actor.
4.	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, misalnya</p>  <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>

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
5.	Generalisasi/ <i>generalization</i>  	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).
6.	Menggunakan / <i>include</i> / <i>uses</i>    	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 ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : <i>a. Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:   <i>b. Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan -

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
		<p>apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <pre> graph TD     UC1((ubah data)) -- "&lt;&lt;include&gt;&gt;" --&gt; UC2((validasi user))   </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

(Sumber: Sukamto dan Shalahuddin)

### 2.2.6.2 Pengertian *Class Diagram*

Menurut Julianto & Setiawan (2019), “*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

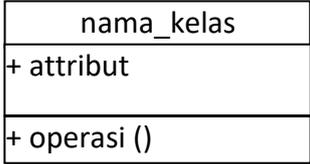
Menurut Sujono et al (2019), “*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (atribut/properti) suatu system”.

Menurut Susandri et al (2016), “*Class* adalah deskripsi sekelompok objek dari property (atribut), sifat (operasi), relasi antar objek dan semantik yang umum. *Class* merupakan template untuk membentuk objek. Setiap objek merupakan contoh dari beberapa *class* dan objek tidak dapat menjadi contoh lebih dari satu *class*”.



Simbol-simbol pada *Class Diagram* adalah sebagai berikut:

**Tabel 2.3** Simbol - simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <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 antarkelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua- bagian ( <i>whole-part</i> ).

(Sumber: Sukamto dan Shalahuddin)



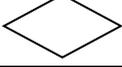
### 2.2.6.3 Pengertian Activity Diagram

Menurut Aditya et al (2021), “Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh system”.

Menurut Sholihin & Ariyani (2023), “*Activity* diagram adalah salah satu model diagram yang menggambarkan *workflow* proses bisnis dan urutan aktifitas dalam sebuah proses dan diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas lainnya atau dari aktifitas ke status”.

Simbol-simbol pada *Activity* Diagram adalah sebagai berikut:

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

No	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status 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 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
6.	<p><i>Swimlane</i></p> <p>atau</p>	<p><i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

(Sumber: Sukamto dan Shalahuddin)

#### 2.2.6.4 Pengertian *Sequence Diagram*

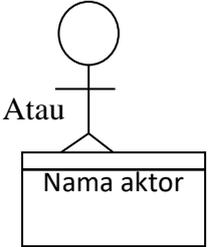
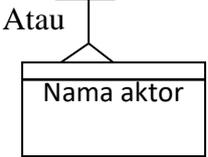
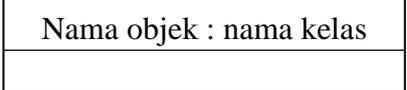
Menurut Julianto & Setiawan (2019), “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek – objek yang terlibat dalam sebuah use case beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek it. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case”.

Menurut Rianto et al (2023), “Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu Sequence Diagram adalah salah satu dari diagram-diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object”.

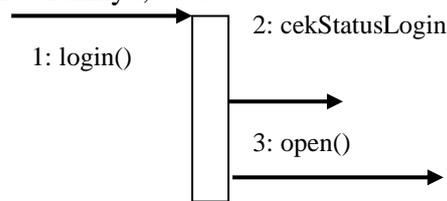
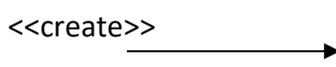
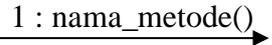
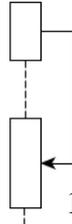


Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

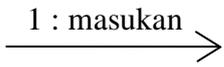
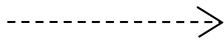
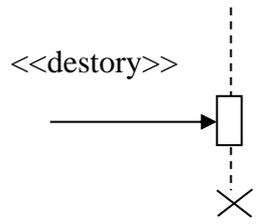
**Tabel 2.5** Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>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 di luar 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>  <p>Nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan.</p>

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnva  maka cek Status Login () dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif.
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> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus -

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
		ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	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>destrory</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber: Sukamto dan Shalahuddin)

## 2.3 Teori Judul

### 2.3.1 Pengertian Aplikasi

Menurut Rasetio & Puspitasari (2019), “Mengungkapkan bahwa media yang digunakan yaitu media elektronik berupa aplikasi. Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya”.



Menurut Huda & Priyatna (2019), “Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia”.

### **2.3.2 Pengertian Persetujuan**

Menurut Kumalasari & Ningsih (2018), “Perjanjian adalah suatu persetujuan dengan dua orang atau lebih saling mengikatkan diri untuk melaksanakan sesuatu hal”.

Menurut Busro (2018), “Sepakat ini merupakan persetujuan yang dilakukan oleh kedua belah pihak”.

### **2.3.3 Pengertian Permintaan**

Menurut Habib & Risnawati (2017), “Permintaan adalah kebutuhan kelompok/ individu terhadap suatu jenis barang tergantung kepada faktor-faktor tertentu”.

Menurut Suhairi et al (2022) mengatakan bahwa permintaan adalah keinginan akan sesuatu yang didukung dengan kemampuan serta kesediaan produk.

### **2.3.4 Pengertian Kebutuhan**

Menurut Wahyu (2022), “Kebutuhan adalah konstruk yang menunjukkan sebuah dorongan dalam wilayah otak, yang mengatur berbagai proses seperti persepsi, pikiran, dan tindakan dengan maksud untuk mengubah kondisi yang ada dan tidak memuaskan”.

Menurut Radissa et al (2020), “Kebutuhan adalah ketika kelompok atau individu mampu untuk melakukan aktivitas - aktivitas yang menunjang kebutuhannya agar terpenuhi”.



### 2.3.5 Pengertian Event

Menurut Syaiful (2019), “event adalah sebuah program yang akan dilakukan secara terencana untuk suatu tujuan. Ragam acara sangat bergantung dari ide yang melandasi terlaksananya hal itu”.

Menurut Syanurdi & Heldi (2019), “Event adalah peristiwa terencana dan bukan kejadian tiba-tiba”.

### 2.3.3 Pengertian Website

Menurut Ulum & Muchtar (2018), “*Website* merupakan informasi deskriptif pada salinan tercetak dan bentuk-bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program”.

Menurut Ronaldo & Pasha (2021), “*Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

### 2.3.4 Aplikasi Persetujuan Permintaan Kebutuhan Event Pada Departemen SSC ICT PT Kilang Pertamina Internasional Refinery Unit III Plaju Berbasis Website

Aplikasi Persetujuan Permintaan Kebutuhan Event Pada Departemen SSC ICT PT Kilang Pertamina Internasional Refinery Unit III Plaju Berbasis Website adalah sebuah Aplikasi tersebut yang dapat melakukan proses approval dari tempat manapun secara *online*, pada aplikasi ini dapat memberikan fasilitas pengelolaan data kebutuhan *event*.



## 2.4 Teori Program

### 2.4.1 Pengertian HTML



**Gambar 2.4** Logo HTML

Sumber: <https://www.freepnglogos.com/pics/html5-logo>

Menurut Ibrahim & Aswin (2023), “HTML (Hypertext Markup Language) yaitu sebuah intruksi pemrograman yang dituliskan dalam bentuk baris – baris kode, yang dikenal dengan istilah Tag. Script HTML dikenal hanya mampu menghasilkan sebuah website dengan output yang sederhana, baik dari segi tampilan maupun kapabilitas yang dimilikinya. Maintenance terhadap halaman web (seperti menambah dan meremajakan isiwebsite), hanya dapat dilakukan dengan mengubah script HTML yang ada dalam program”.

Menurut Sari et al (2022), “HTML (HyperText Markup Language) adalah salah satu bahasa pemrograman yang digunakan untuk membuat website absensi pegawai kantor ini. HTML dominan dengan menggunakan tanda tag < > untuk menyatakan kode –kode yang akan ditafsirkan oleh browser agar halaman dapat ditampilkan dan muncul sesuai dengan posisi yang telah diatur”.

### 2.4.2 Pengertian JavaScript



**Gambar 2.5** Logo JavaScript

Sumber: <https://www.pngwing.com/en/search?q=javascript+Logo>



Menurut Sahi (2020), “*Javascript* adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk *web*”.

Menurut Bekasi (2016), “*JavaScript* adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada *web browser*. Pada awalnya *JavaScript* dikembangkan pada *web browser Netscape* oleh Brenden Eich dengan nama Mocha, kemudian berubah menjadi *Live-Script* dan yang akhirnya menjadi *JavaScript*”.

### 2.4.3 Pengertian CSS



**Gambar 2.6** Logo CSS

Sumber: <https://www.pngwing.com/id/search?q=css+Logo>

Menurut Putri (2020), “CSS atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur dan seragam”.

Menurut Ali & Ambarita (2016), “CSS atau *Cascading Style Sheets* adalah sebuah dokumen yang berisi aturan yang digunakan untuk memisahkan isi dengan layout dalam halaman-halaman web yang dibuat”.



#### 2.4.4 Pengertian PHP



**Gambar 2.7** Logo PHP

Sumber: <https://www.freepnglogos.com/pics/php-logo>

Menurut Afifah & Setyantoro (2021), “*PHP* merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen *HTML* secara *on the fly* yang dieksekusi di *server web*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML*”.

Menurut Mare & Yana (2022), “*PHP* adalah bahasa pemrograman script *serverside* yang sengaja dirancang lebih cenderung untuk membuat dan mengembangkan web”.

#### 2.4.5 Pengertian *Bootstrap*



**Gambar 2.8** Logo Bootstrap

Sumber: <https://getbootstrap.com/docs/5.0/about/brand/>

Menurut Purwanto & Mubarok (2020), “*Bootstrap* merupakan sebuah kerangka *framework* atau wadah untuk pengembangan aplikasi *web* yang lebih cepat dan mudah untuk suatu pengembangan sebuah sistem yang akan dibangun”.

Menurut Ismai (2015), “*Bootstrap* juga merupakan salah satu *framework HTML, CSS dan javascript* yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah *website yang responsive*”.



#### 2.4.6 Pengertian XAMPP



**Gambar 2.9** Logo XAMPP

*Sumber: [https://id.wikipedia.org/wiki/Berkas:Xampp\\_logo.svg](https://id.wikipedia.org/wiki/Berkas:Xampp_logo.svg)*

Menurut Oley et al (2016), “XAMPP adalah perangkat lunak gratis, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”.

Menurut Nirsal et al (2020), “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, 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 (serverside scripting), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya”.

#### 2.4.7 Pengertian MySQL



**Gambar 2.10** Logo MySQL

*Sumber: <https://www.freepnglogos.com/pics/logo-mysql>*

Menurut Novendri (2019), “MYSQL merupakan sistem manajemen database yang bersifat relational. Artinya, data yang dikelola dalam database yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat”.

Menurut Renaldy & Rustam (2020), “MYSQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, pengubahan, dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis”.



#### 2.4.8 Pengertian *Sublime Text*



**Gambar 2.11** Logo Sublime Text

Sumber: <https://seeklogo.com/vector-logo/373309/sublime-text>

Menurut Sukamto (2019), “Sublime text adalah teks editor berbaris Python, sebuah teks editor yang elegan, kaya fitur, cross platform, mudah dan simpel yang cukup terkenal dikalangan developer (pengembang), penulis dan desainer”.

Menurut Farell et al (2018), “Sublime text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating sistem dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerful. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages”.