



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1 Pengertian Komputer

Menurut Nuraini (2023:21) mengatakan bahwa “Komputer merupakan mesin yang dapat dimanfaatkan untuk memproses data berdasarkan perintah dalam program.”

Menurut Sindhu et al (2018:5) “ dalam pengertian paling mendasar, komputer adalah perangkat yang membantu manusia dalam melakukan berbagai macam perhitungan aritmatika. Namun,sekarang ini komputer di artikan kepada sebuah perangkat mesin tersebut.”

Berdasarkan pendapat tersebut pengertian dari komputer adalah sebuah perangkat atau mesin yang dapat menjalankan sebuah perintah berdasarkan informasi digital dan dapat menghasilkan sebuah hasil berupa data informasi.

##### 2.1.2 Pengertian Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2022:2) “Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).”

Menurut Alda (2021:1) “perangkat lunak (*software*) adalah perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan dimana memiliki struktur data yang memungkinkan program memanipulasi informasi secara proporsional dan mempunyai dokumen yang menggambarkan operasi dan kegunaan program.”

Jadi, dapat disimpulkan bahwa perangkat lunak (*software*) merupakan sekumpulan serangkaian perintah yang dapat dimengerti oleh mesin komputer sehingga komputer dapat menjalankan apa yang kita inginkan.



### 2.1.3 Pengertian Perangkat Keras

Menurut Rosallia et al (2019:687) “perangkat keras adalah perangkat fisik yang merupakan bagian dari sistem komputer yang dapat mendukung analisis geografis dan pemetaan.”

Menurut Zaki (2019:3) “Hardware komputer sangat beragam terdiri dari komponen fisik komputer contohnya seperti monitor, CPU, Motherboard, Memory, Peranti drive, printer, hard disk dan lain sebagainya.”

Sehingga dapat disimpulkan bahwa hardware atau perangkat keras adalah kumpulan perangkat fisik yang menjadi satu bagian dari sebuah sistem komputer itu sendiri.

### 2.1.4 Pengertian Internet

Menurut Wulandari (Dalam Sibero 2019:24) “Internet yaitu *Interconnected Network* yang merupakan sebuah jaringan komputer yang saling menghubungkan antar komputer secara global.”

Menurut Wulandari (Dalam Sarwono 2019:24) Internet adalah sekumpulan jaringan yang memiliki skala global. Maka dapat disimpulkan dari kedua pengertian menurut para ahli di atas bahwa internet adalah jaringan global yang menghubungkan seluruh jaringan di dunia melalui jalur satelit.

Maka dapat disimpulkan dari kedua pengertian menurut para ahli di atas bahwa internet adalah jaringan global yang menghubungkan seluruh jaringan di dunia melalui jalur satelit.

### 2.1.5 Pengertian Basis Data

Menurut Jayanti dan Sumiari (2018:1) “Basis data secara umum diartikan sebagai kelompok data yang saling terhubung dalam suatu media penyimpanan tertentu yang diatur oleh pengaturan untuk membuatnya lebih mudah untuk menangani informasi tersebut”. Sedangkan Menurut A.S dan Shalahudin (2018:28) “Sistem basis data adalah sistem yang terkomputerisasi serta memiliki tujuan utama untuk memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media



untuk menyimpan data agar dapat diakses dengan mudah dan cepat.”

Dilihat dari pemaparan pengertian diatas dapat disimpulkan bahwa Basis Data merupakan kumpulan banyak data keterkaitan dengan penyimpanan pada suatu perangkat yang diatur secara sistematis untuk mempermudah pengelolannya.

### 2.1.6 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2022:168), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2022:168-171) adalah sebagai berikut:

1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.



#### 4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

## 2.2 Teori Khusus

### 2.2.1 Unified Modeling Language (UML)

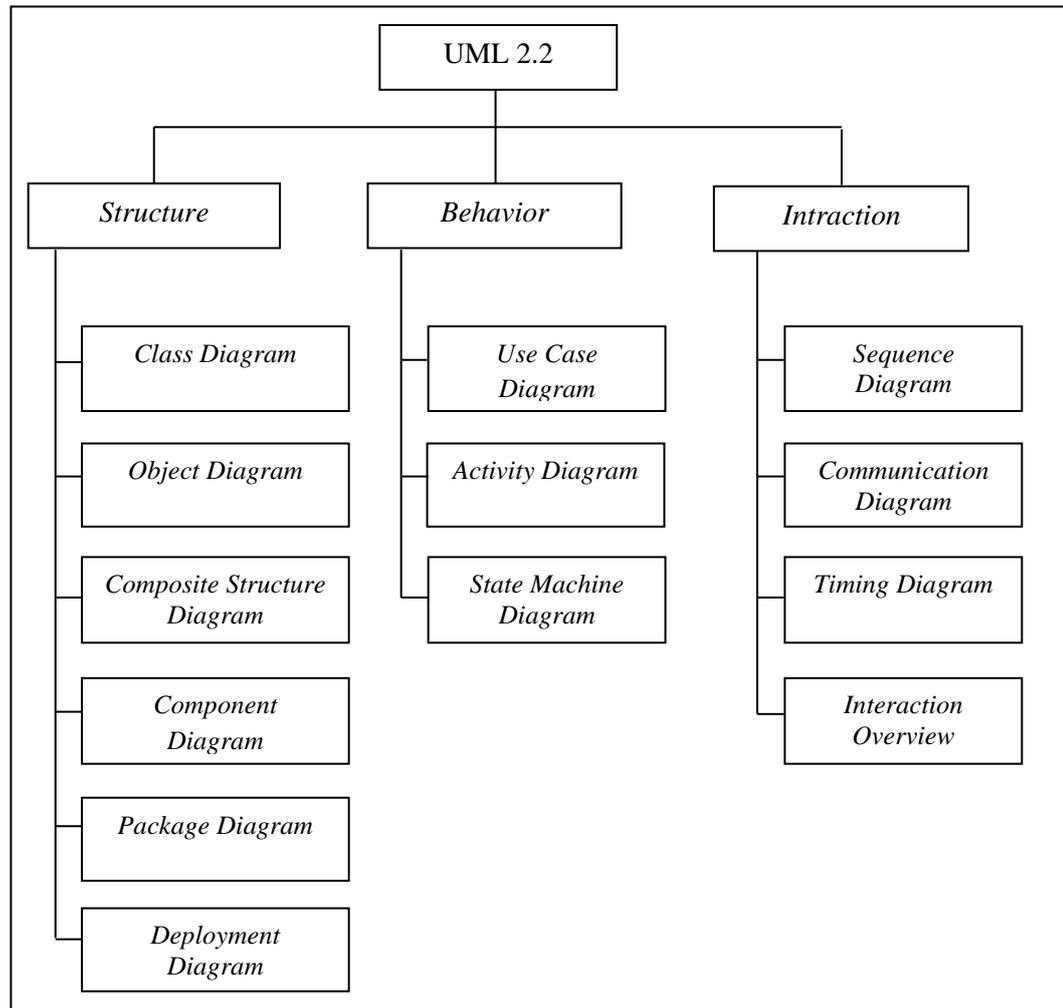
Sukamto dan Shalahuddin (2022:173), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”



**Gambar 2.1.** Tampilan Logo UML

Menurut Sukamto dan Shalahuddin (2022:180), “Pada UML 2.2 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



**Gambar 2.2** Kategori dan Macam-macam Diagram UML

Jadi, bisa disimpulkan bahwa *UML* menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketanggahan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.

## 2.2.2. Jenis-jenis Diagram UML

### 2.2.2.1. Diagram Use Case

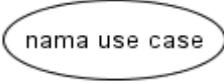
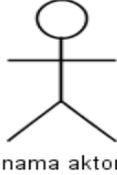
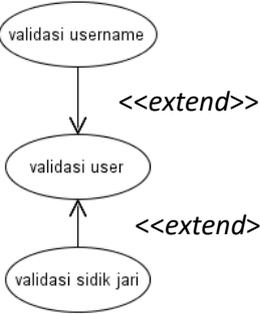
Sukanto dan Shalahuddin (2022:197-199), menjelaskan tentang *use case* diagram sebagai berikut :

“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem

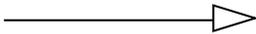
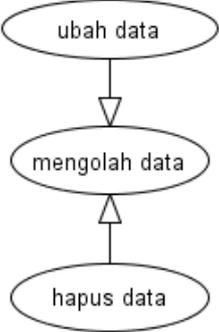
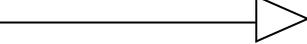
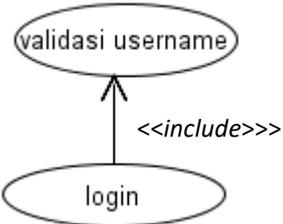


informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

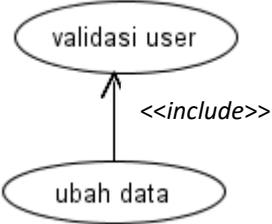
**Tabel 2.1.** Simbol-simbol Diagram *Use Case*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.	<p>aktor / actor</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 actor.</p>
3.	<p>asosiasi / association</p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.</p>
4.	<p>ekstensi / extend</p> 	<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 2.1. Simbol-simbol Diagram *Use Case*

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 / include / uses   <<include>>   <<uses>>	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> : <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</li> </ul> 

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

No	Simbol	Deskripsi
		<ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul>  <pre> graph BT     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 (2022:197-199)

Jadi, dapat disimpulkan bahwa Ada dua hal utama pada *use case* yaitu:

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

#### 2.2.2.2. Diagram *Class*

Sukamto dan Shalahuddin (2022:187), menjelaskan tentang *class diagram* sebagai berikut :

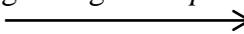
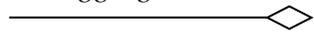
“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat



agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.”

Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

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

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	asosiasi / association 	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: Sukamto dan Shalahuddin (2022:187)

Jadi, dapat disimpulkan bahwa pengertian dari *Class Diagram* adalah sebuah diagram yang digunakan untuk menampilkan kelas-kelas pada aplikasi yang akan dibuat.



### 2.2.2.3. Diagram Activity

Sukanto dan Shalahuddin (2022:203), menjelaskan tentang *activity diagram* sebagai berikut :

“*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 sistem.”

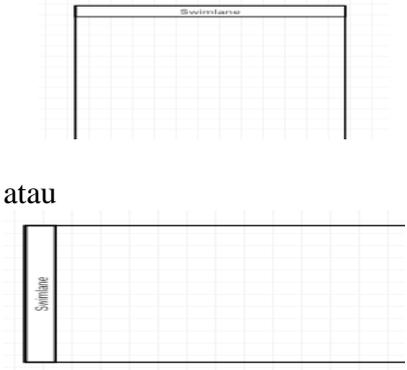
Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

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

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 di mana 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 2.3. Simbol-simbol Diagram Activity

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

Sumber: Sukamto dan Shalahuddin (2022:203)

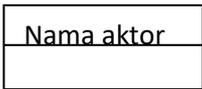
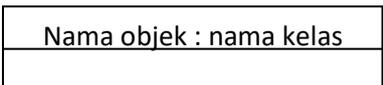
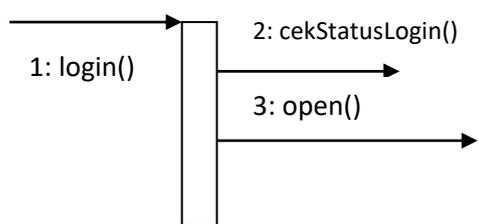
Jadi, bisa disimpulkan bahwa pengertian dari *Activity Diagram* adalah gambaran alur aktifitas yang akan dilakukan dalam aplikasi pada satu proses.

#### 2.2.2.4. Diagram Sequence

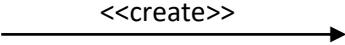
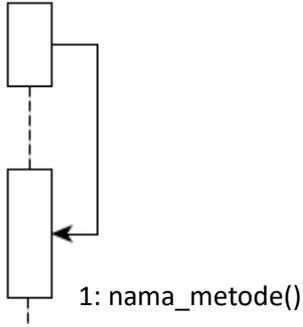
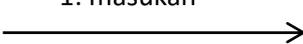
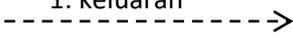
“Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Sukamto dan Shalahuddin, 2022:207-209).

Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.4 Simbol-simbol pada Diagram *Sequence*

No	Simbol	Deskripsi
1.	<p>Actor</p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</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>
4.	<p>Waktu aktif</p> 	<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>1: login()</p> <p>2: cekStatusLogin()</p> <p>3: open()</p> <p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>



5.	Pesan tipe create 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe call 	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 ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe send 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 	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

Lanjutan Tabel 2.4. Simbol-simbol Diagram *Sequence*

No	Simbol	Deskripsi
9.	Pesan tipe destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

Sumber: Sukamto dan Shalahuddin (2022:207-209)

Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

### 2.2.3 Kamus Data

Sukamto dan Shalahudin (2022:94), 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).

Simbol-simbol yang digunakan dalam kamus data, yaitu:

Tabel 2.5 Simbol-Simbol Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau....
{ } <sup>n</sup>	n kali diulang/ bernilai banyak
()	Data opsional
*...*	Batas komentar

(Sumber: Sukamto dan Shalahuddin (2022:94))



## **2.3 Teori Judul**

### **2.3.1 Pengertian Rancang Bangun**

Menurut Pressman (2009:32) rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan sedangkan pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan.

Jadi dapat disimpulkan bahwa Rancang Bangun merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

### **2.3.2 Pengertian Aplikasi**

Menurut Habibi dan Karnovi (2020:14) Aplikasi adalah sebuah program yang dibuat untuk siap dipakai yang bisa menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri.

Pengertian Aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari sebuah rancang sistem yang untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemograman tertentu.

Berdasarkan dari penjelasan diatas dapat disimpulkan pengertian aplikasi adalah sebuah program yang telah di tentukan bahasa nya untuk mengolah serta menjalankan sebuah perintah dari rancangan sistem tertentu.

### **2.3.3 Pengertian Whatsapp**

Whatsapp adalah aplikasi pesan seluler lintas platform yang digunakan untuk bertukar pesan tanpa harus membayar SMS, ini dimungkinkan karena Whatsapp memakai paket data internet sehingga mengirim pesan dan tetap berhubungan dengan teman-teman tidak akan dikenakan biaya (Elianur, 2017).



Jadi dapat disimpulkan bahwa whatsapp merupakan aplikasi kirim pesan canggih untuk smartphone.

### **2.3.5 Chatbot**

“*Chatbot* adalah sebuah program komputer yang dirancang untuk menstimulasikan sebuah percakapan atau komunikasi yang interaktif kepada pengguna (manusia) baik melalui text, suara ataupun visual”. Harahap (2020:6)

Menurut Amalia dan Wibowo (2019:137) *Chatbot* merupakan program aplikasi atau layanan yang berinteraksi dengan pengguna melalui percakapan teks. *Chatbot* bekerja untuk menggantikan peranan manusia dalam melayani pembicaraan melalui aplikasi pesan.

Dari penjelasan diatas dapat disimpulkan bahwa *Chatbot* sendiri merupakan program komputer yang dapat berkomunikasi untuk tujuan memberikan informasi kepada pengguna melalui pelayanan aplikasi pesan.

### **2.3.6 Pengertian Layanan Informasi**

Prayitno (2004:260) menyatakan bahwa layanan informasi adalah salah satu layanan dari sembilan layanan yang ada dalam bimbingan dan konseling. Layanan informasi dimaksudkan sebagai pemberian informasi tentang hal-hal yang dibutuhkan oleh peserta layanan dalam pemenuhan kebutuhannya tentang data dan keterangan yang aktual dalam kehidupan sehari-hari dan perencanaan masa depannya. Winkel (2007:316) menambahkan layanan informasi berusaha memenuhi kekurangan informasi yang diperlukan. Selanjutnya Sukardi (2007:61) mendefinisikan layanan informasi sebagai layanan bimbingan yang memungkinkan peserta didik dan pihak-pihak lain yang dapat memberikan pengaruh besar kepada peserta didik (terutama orang tua) dalam menerima dan memahami informasi yang dapat digunakan sebagai bahan pertimbangan dan mengambil keputusan. Dari beberapa pernyataan ahli tersebut dapat disimpulkan bahwa layanan informasi adalah layanan yang memberikan informasi yang membantu individu dengan berbagai pemahaman tentang berbagai hal yang berguna.



### 2.3.7 Pengertian Judul Secara Keseluruhan

Rancang Bangun Aplikasi Whatsapp Chatbot Untuk Layanan Informasi di Politeknik Negeri Sriwijaya adalah sebuah perancangan dan pembangunan aplikasi whatsapp chatbot yang digunakan sebagai penjawab pesan otomatis yang berfungsi untuk meningkatkan pelayanan di Politeknik Negeri Sriwijaya.

## 2.4 Teori Program

### 2.4.1 Pengertian *Node.js*

*Node.js* merupakan platform karya Ryan Dahl pada tahun 2009 yang berfungsi untuk mengembangkan aplikasi web dengan bahasa pemrograman *JavaScript* dan Platform ini memungkinkan pengembang untuk menjalankan *JavaScript* dari sisi server (Alexandromeo, 2023). Kelebihan menggunakan *node.js* yaitu kita tidak butuh lagi menggunakan webserver untuk menjalankan kode javascript karena sudah disediakan oleh *library Node.js*.

### 2.4.2 Pengertian *SQL*

Menurut Sukamto dan Shalahudin (2018:28) “*SQL* (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. *SQL* awalnya dikembangkan berdasarkan teori aljabar rasional dan kalkulus”.

Menurut Sudiarjo dalam Joko (2019:120) “*SQL* (*Structur Query Language*) merupakan bahasa program yang banyak digunakan dalam berbagai produk database.”

Dari penjelasan diatas dapat diambil kesimpulan bahwa *SQL* merupakan bahasa yang digunakan untuk mengelola data dalam berbagai produk database.

### 2.4.3 Pengertian *MySQL*

Menurut Fitri (2021:2) “*MySQL* adalah DBMS (Database Management System) yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya terbatas)”. Jadi *MySQL* adalah sebuah software yang digunakan untuk mengelola sebuah database menggunakan bahasa pemrograman *SQL*.



#### 2.4.4 Pengertian *Visual Studio Code*

Visual Studio Code adalah editor pengkodean yang bersifat open source yang dikembangkan oleh Microsoft untuk Windows, Linux, dan OS X. Terdapat dukungan untuk *debugging*, *embedded git control*, *snippets*, dan *code refactor*. Visual Studio code juga dapat disesuaikan, sehingga pengguna (*user*) dapat mengubah tema editor (*the theme's editor*), pintasan papan ketik (*keyboard shortcuts*), dan preferensi (*preference*) (Lardinois, 2015).

#### 2.4.5 Pengertian API

API atau *Application Programming Interface* adalah konsep fungsi antarmuka pemrograman aplikasi, yang menjadi salah satu cara agar suatu aplikasi dapat diakses dan dimanfaatkan oleh pihak lain tanpa mengubah struktur kode utama maupun database sistem, serta memudahkan komunikasi antar sistem meskipun berbeda platform (Wijonarko & Mulya, 2018). API gateway adalah antar muka yang menghubungkan aplikasi dari beberapa aplikasi. API gateway biasa disebut gerbang dari API yang menghubungkan beberapa data API dan dapat memajemen data API yang ada pada sebuah instansi untuk meminimalisir permasalahan yang ada pada sebuah instansi tersebut (Saputra, et al., 2018).

#### 2.4.6 *Dialogflow*

*Dialogflow* adalah penyedia layanan yang dibuat oleh *Google Cloud Platform* yang digunakan untuk mendesign dan memproses salah satu struktur didalam pembuatan chatbot yaitu *Natural Language Understanding* atau mengenali bahasa alami manusia (Dialogflow, 2023). Penulis menggunakan *Dialogflow* untuk memproses sebuah kalimat menggunakan *Machine Learning*.

#### 2.4.7 Wablas

WABLAS adalah layanan gateway API Whatsapp dan SMS untuk mengirim dan menerima pesan, notifikasi, penjadwal, pengingat, dan pelacakan dengan integrasi sederhana untuk bisnis (WABLAS, 2023). Jadi penulis



membutuhkan Wablas untuk digunakan sebagai pengirim pesan dan mendapatkan pesan Whatsapp melalui API.