

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Dasar**

##### **2.1.1 Definisi Sistem Informasi**

###### **1. Sistem**

Menurut Abdul Kadir (2014) pada buku berjudul “Pengenalan Sistem Informasi”, “Sistem adalah kumpulan elemen yang saling berhubungan untuk mencapai suatu tujuan.” disimpulkan bahwa sistem adalah suatu kesatuan terorganisasi yang terdiri dari elemen-elemen yang berinteraksi dan saling mempengaruhi satu sama lain untuk mencapai tujuan tertentu. Pengertian ini menggambarkan kompleksitas hubungan antara elemen-elemen dalam suatu kesatuan, yang bekerja bersama untuk mencapai suatu hasil atau fungsi yang diinginkan. Keberadaan sistem juga sangat penting dalam memahami hubungan antara bagian-bagian yang membentuk suatu kesatuan yang utuh dan berfungsi.

###### **2. Informasi**

Pengetahuan yang sangat penting diperlukan oleh suatu kegiatan dalam membuat keputusan agar terhindar dari kesalahan adalah informasi. Informasi dapat dijelaskan sebagai data yang telah diproses menjadi bentuk yang lebih bermanfaat dan signifikan bagi penerima informasi. Menurut Anggraeni dan Irviani (2017:13), “mendefinisikan informasi sebagai kumpulan data atau fakta yang diatur atau diproses dengan cara tertentu sehingga memiliki makna bagi penerimanya”. Menurut Sutabri dalam Trimahardhika dan Sutinah (2017:250) menyatakan bahwa “informasi merupakan data yang telah diolah, diklasifikasikan, diinterpretasikan, dan digunakan untuk proses pengambilan keputusan”. Menurut Munawar dan Putri (2020) “Informasi adalah data yang telah diubah menjadi informasi melalui analisis dan interpretasi, memungkinkan pengambilan keputusan dan penciptaan pengetahuan dalam organisasi”.

Informasi adalah hasil dari pengolahan data yang mempunyai makna dan kegunaan untuk banyak orang. Dengan demikian, informasi bukan hanya sekadar menyajikan fakta, tetapi juga memberikan pemahaman yang lebih dalam dan

berguna bagi penerimanya. Informasi memiliki peran penting dalam mendukung proses pengambilan keputusan dengan menyediakan landasan yang relevan dan bermakna bagi pemakai informasi tersebut.

### 3. Sistem Informasi

Menurut Arief Yanto Rukmana et al. (2022) pada buku berjudul “Pengantar Sistem Informasi” mendefinisikan “Sistem Informasi sebagai kombinasi orang, proses, data, dan teknologi yang bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi dalam suatu organisasi”. Pada intinya, sistem informasi berperan sebagai struktur terorganisir yang memfasilitasi manajemen dan optimalisasi informasi dengan tujuan mendukung proses pengambilan keputusan, meningkatkan efisiensi operasional, serta mendorong upaya strategis.

### 4. Komponen Sistem Informasi

Di dalam sistem informasi ada beberapa komponen-komponen utama, termasuk perangkat keras (hardware), perangkat lunak (software), data, prosedur, dan manusia. Setiap komponen berperan penting dalam fungsi dan kinerja sistem informasi. Perangkat keras (hardware), perangkat lunak (software), data, prosedur, dan individu dapat menjadi komponen sistem informasi.4. Komponen Sistem Informasi

Dalam sistem informasi terdapat komponen-komponen utama, termasuk perangkat keras (hardware), perangkat lunak (software), data, prosedur, dan manusia. Setiap komponen berperan penting dalam fungsi dan kinerja sistem informasi. Sistem informasi dapat mencakup perangkat keras (hardware), perangkat lunak (software), data, prosedur, dan orang-orang yang terlibat.



Gambar 2.1 Komponen Sistem Informasi

Beberapa komponen utama dalam sistem informasi meliputi:

- a. Perangkat Keras (*Hardware*): Komponen fisik sistem informasi, seperti komputer, server, perangkat penyimpanan data, jaringan komunikasi, dan perangkat input/output.
- b. Perangkat Lunak (*Software*): Program-program atau aplikasi yang dijalankan pada perangkat keras untuk melakukan berbagai tugas, seperti sistem operasi, basis data, aplikasi bisnis, dan lainnya.
- c. Data: Informasi yang dikumpulkan, disimpan, dan diolah oleh sistem informasi. Data dapat berupa fakta, angka, teks, gambar, atau bentuk informasi lainnya.
- d. Prosedur: Langkah-langkah atau aturan yang mengatur cara sistem informasi bekerja. Proses-proses ini dapat mencakup aturan pengolahan data, alur kerja, kebijakan keamanan, dan prosedur pengelolaan.
- e. Manusia (*People*): Orang-orang yang menggunakan, mengelola, dan berinteraksi dengan sistem informasi. Ini mencakup pengguna akhir, administrator sistem, analis bisnis, dan pihak-pihak terkait lainnya.

##### 5. Karakteristik Sistem

Menurut Fatta (2017), untuk memahami atau mengembangkan suatu sistem, perlu dilakukan pemisahan unsur-unsur yang membentuknya. Ada beberapa karakteristik yang membedakan suatu sistem dari sistem lainnya, yaitu:

a. Batasan (boundary): Gambaran unsur-unsur dalam sistem dan yang berada di luar sistem.

b. Lingkungan (environment): Semua elemen di luar sistem yang memberikan asumsi, kendala, dan input ke sistem.

c. Masukan (input): Sumber daya dari lingkungan seperti data, bahan baku, peralatan, dan energi yang dikonsumsi dan dimanipulasi oleh sistem.

d. Keluaran (output): Sumber daya atau produk seperti informasi, laporan, dokumen, tampilan layar komputer, dan barang jadi yang dihasilkan oleh kegiatan dalam sistem untuk lingkungan.

e. Komponen (component): Kegiatan atau proses dalam sistem yang mengubah input menjadi bentuk setengah jadi (output), mungkin berupa subsistem.

f. Penghubung (interface): Titik di mana komponen atau sistem berinteraksi dengan lingkungannya.

g. Penyimpanan (storage): Area yang menyimpan informasi, energi, bahan baku, dan lain-lain., berfungsi sebagai media penyangga untuk komponen yang berbeda bekerja dengan data yang sama.

## 6. CBIS

Sistem informasi berbasis komputer (CBIS) merujuk pada suatu sistem informasi yang menjalankan tugas-tugas tertentu menggunakan komputer dan teknologi komunikasi, seperti mendukung pengambilan keputusan. Dalam konteks ini, peran utama komputer sangat signifikan dalam menghasilkan informasi.



Gambar 2.2 CBIS (Computer Based Information System)

### 2.1.2. Rancang Bangun

Menurut R. Pressman dalam (Girsang, 2018), “Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan”.

Menurut Maulani, G., Septiani, D., & Sahara, P. N. dalam (Girsang, 2018) berpendapat bahwa, “Rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut”.

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

Proses rancang bangun meliputi beberapa langkah-langkah sebagai berikut:

1. Analisis Kebutuhan: Identifikasi kebutuhan dan tujuan dari apa yang akan dirancang dan dibangun.
2. Perencanaan: Tentukan strategi dan rencana umum untuk merancang dan membangun solusi.
3. Desain: Buat rancangan detil sistem atau produk, termasuk spesifikasi teknis dan kebutuhan fungsional.
4. Implementasi: Terapkan rancangan menjadi bentuk nyata, baik itu melalui penulisan kode, pembangunan perangkat keras, atau implementasi langkah-langkah lainnya sesuai desain.
5. Pengujian: Lakukan pengujian untuk memastikan bahwa produk atau sistem memenuhi spesifikasi dan bekerja sesuai yang diharapkan.
6. Optimasi: Jika diperlukan, lakukan penyesuaian dan perbaikan untuk meningkatkan kinerja atau mengatasi masalah yang mungkin muncul selama pengujian.
7. Penerapan: Terapkan solusi ke dalam lingkungan yang sesungguhnya atau pasar.
8. Pemeliharaan: Lakukan pemeliharaan dan pembaruan sesuai dengan kebutuhan atau perubahan yang mungkin terjadi.

Setiap langkah dalam proses rancang bangun dapat memiliki tahapan-tahapan yang lebih spesifik tergantung pada bidang dan kompleksitas proyek. Juga, kolaborasi dan komunikasi yang baik antara tim adalah kunci untuk kesuksesan dalam rancang bangun.

### 2.1.3. SDLC

Menurut Marcello Singadji et al. (2022, p. 125) dalam bukunya yang berjudul “Pengantar Sistem Informasi” “System Development Life Cycle (SDLC) adalah proses memahami bagaimana sistem informasi digunakan dan mendukung kebutuhan bisnis dengan merancang sistem, membangun, dan menyampaikannya kepada pengguna”.

SDLC dirancang untuk memenuhi kebutuhan pengguna dan merupakan kerangka kegiatan yang dilakukan selama proses pengembangan sistem. Rangkaian SDLC memiliki empat fase, antara lain:

- a) perencanaan
- b) analisis
- c) desain
- d) implementasi



Gambar 2.2 Gambar SDLC Lifecycle

Adapun model pengembangan SDLC pada perancangan ini menggunakan model *waterfall*, yaitu sebagai berikut :

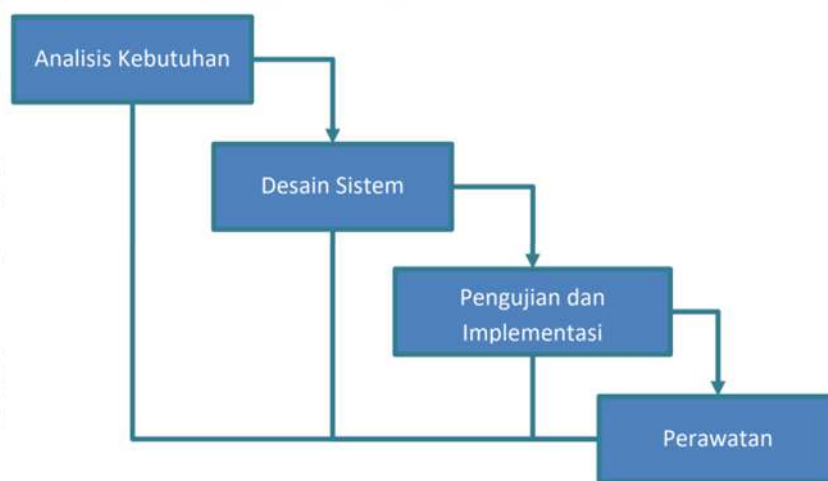
Pada perancangan ini, model pengembangan SDLC menggunakan model *waterfall*, seperti yang ditunjukkan di bawah ini:

### 2.1.8 Metode *Waterfall*

Menurut (Wahid, 2020) “Metode *waterfall* adalah metode yang paling banyak digunakan untuk tahap pengembangan. Metode *Waterfall* merupakan

pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak”. Model air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*Classic cycle*). Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial terurut mulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*).

Model air terjun, yang juga dikenal sebagai model sekuensial linier atau alur hidup klasik, memberikan pendekatan yang berurutan dalam pengembangan perangkat lunak, dimulai dari analisis, desain, pengkodean, pengujian, hingga tahap pendukung. Penelitian ini menerapkan metode pengembangan waterfall, yang merupakan model sistematis dan sekuensial dalam pengembangan sistem informasi. Metode Waterfall melibatkan tahapan-tahapan berikut:



Gambar 2.2 Metode Waterfall

#### 1. Analisis Kebutuhan

Langkah penting dalam pengembangan proyek atau perencanaan strategis. Ini melibatkan identifikasi, pengumpulan, dan evaluasi informasi untuk memahami apa yang diperlukan untuk mencapai tujuan tertentu. Dengan melakukan analisis kebutuhan, kita dapat menentukan sumber daya, waktu, dan langkah-langkah yang diperlukan untuk memenuhi kebutuhan tersebut.



## 2. *Desain Sistem*

Pada tahap ini, spesifikasi sistem yang dibutuhkan dihasilkan berdasarkan hasil analisis. Ini mencakup desain arsitektur sistem, desain antarmuka pengguna, desain basis data, dan spesifikasi perangkat lunak.

## 3. Pengujian dan Implementasi

Dalam langkah ini, desain perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Verifikasi dilakukan pada setiap unit untuk memastikan kesesuaian dengan spesifikasinya. Kemudian, modul-program digabung dan diuji sebagai keseluruhan sistem untuk memverifikasi kesesuaian dengan kebutuhan perangkat lunak sebelum dapat disampaikan kepada pengguna..

## 4. Pemeliharaan

Dalam tahap ini, model pengembangan waterfall mencapai puncaknya dengan implementasi sistem yang telah selesai dikembangkan. Fase ini ditandai oleh periode pemeliharaan, di mana kegiatan pemeliharaan melibatkan perbaikan terhadap kesalahan yang mungkin tidak terdeteksi selama tahapan sebelumnya.

### **2.1.9 Object Oriented Analysis Design (OOAD)**

Menurut Mathiassen (2000), Object Oriented Analysis and Design (OOAD) adalah metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi object. OOAD (Object-Oriented Analysis and Design) adalah sebuah metode pengembangan perangkat lunak yang menggunakan konsep-konsep OOP untuk menganalisis dan merancang sistem. OOAD melibatkan proses analisis kebutuhan pengguna, pemodelan sistem, dan desain sistem berbasis objek. OOAD bertujuan untuk menghasilkan sistem yang lebih modular, fleksibel, dan mudah dipelihara.

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu sebagai berikut:



- 1) Object Oriented Analysis (OOA) atau bisa disebut analisis berorientasi objek adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek. OOAD menggunakan konsep reuse, yaitu memanfaatkan kembali kode yang sudah ada untuk mempercepat pengembangan sistem. OOAD juga menggunakan UML (Unified Modeling Language) sebagai bahasa standar untuk visualisasi, dokumentasi, dan komunikasi antar tim pengembang. Kelebihan OOAD antara lain sistem yang dihasilkan lebih modular, fleksibel, dan mudah dipelihara, serta memungkinkan pengembangan sistem yang lebih cepat. Namun, kekurangan OOAD adalah membutuhkan waktu yang lebih lama untuk berpindah dari metode pengembangan sistem yang sudah ada, serta memerlukan prosedur yang eksplisit terhadap reuse agar dapat diterapkan pada skala besar.
- 2) Desain Berorientasi Objek (OOD) adalah bagian dari metode pengembangan perangkat lunak yang menggunakan konsep-konsep OOP untuk merancang sistem. OOD melibatkan proses pemodelan sistem dan desain sistem berbasis objek. Tujuannya adalah untuk menghasilkan desain sistem yang lebih modular, fleksibel, dan mudah dipelihara. Dalam konteks OOAD, OOD merupakan langkah penting setelah analisis kebutuhan pengguna (OOA) dan sebelum implementasi sistem. OOD memungkinkan pengembang untuk merancang struktur sistem secara lebih terorganisir berdasarkan konsep-konsep objek, seperti pewarisan, enkapsulasi, dan polimorfisme. Hal ini membantu dalam menciptakan desain yang lebih efisien dan mudah dimengerti. OOD juga memungkinkan untuk menerapkan prinsip-prinsip desain yang baik, seperti pemisahan kepentingan (separation of concerns) dan penggunaan pola-pola desain (design patterns) untuk memecahkan masalah umum dalam pengembangan perangkat lunak.

Analisis dan Desain Berbasis Objek (OOAD) merupakan pendekatan dalam pengembangan perangkat lunak yang menggunakan prinsip-prinsip pemrograman berbasis objek (OOP) untuk memodelkan suatu sistem. Berikut ini adalah karakteristik-karakteristik utama dari OOAD:

1. Abstraksi (*Abstraction*)

Abstraksi adalah konsep dasar dalam OOAD. Ini melibatkan penyederhanaan sistem yang kompleks dengan memodelkan kelas-kelas berdasarkan karakteristik-karakteristik esensialnya, tanpa memperhatikan detail yang tidak perlu.

2. Enkapsulasi (*Encapsulation*)

Enkapsulasi melibatkan pengelompokan data (atribut) dan metode-metode (fungsi) yang beroperasi pada data ke dalam satu unit yang disebut kelas. Ini membatasi akses ke detail internal suatu objek dan hanya mengekspos apa yang diperlukan.

3. Pewarisan (*Inheritance*)

Pewarisan atau bias dikenal dengan *Inheritance* adalah mekanisme yang memungkinkan suatu kelas baru (subkelas atau kelas turunan) mewarisi properti dan perilaku dari suatu kelas yang sudah ada (superkelas atau kelas dasar). Ini mendorong penggunaan kembali kode (*reuse*) dan membentuk hubungan hierarki antar kelas.

4. Polimorfisme (*Polymorphisme*)

Polimorfisme memungkinkan objek-objek dari kelas-kelas yang berbeda diperlakukan sebagai objek dari suatu kelas dasar yang umum. Ini memungkinkan penggunaan antarmuka tunggal untuk mewakili berbagai jenis objek dan mendukung penempatan metode dan pengikatan dinamis.

### 2.1.10 Unified Modeling Language (UML)

Menurut Ivar Jacobson (2018): “UML adalah alat visual yang kuat untuk merancang dan mendokumentasikan arsitektur sistem perangkat lunak. Dengan notasi yang jelas, UML memfasilitasi pemahaman yang mendalam tentang struktur dan interaksi komponen-komponen dalam suatu sistem.”

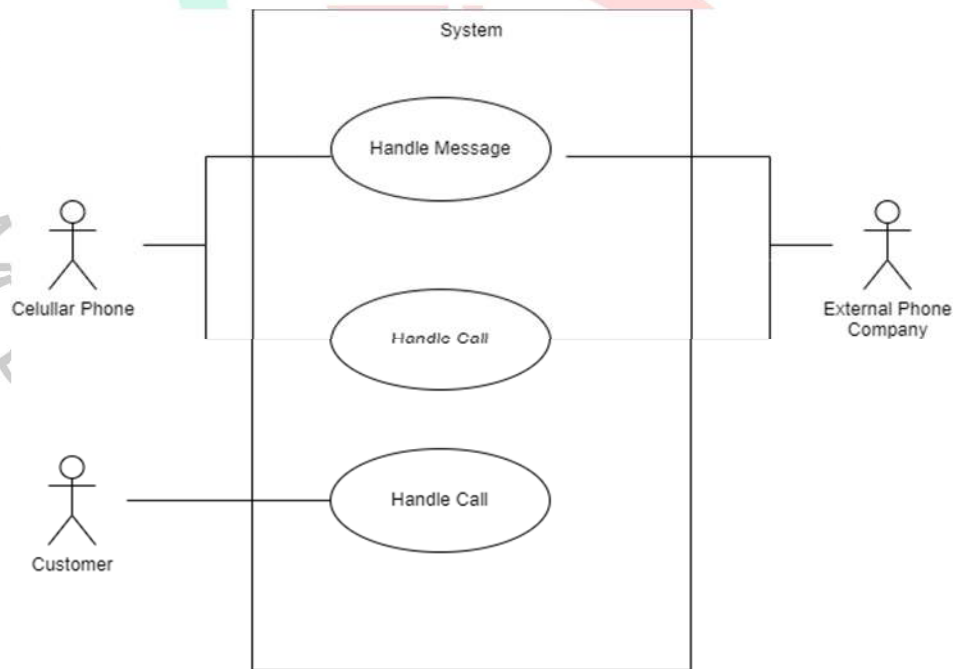
#### 1. Use Case Diagram

Menurut Setiawan dan Khairuzzaman (2017), “Diagram Use Case menggambarkan bagaimana interaksi antara use case dan actor terjadi, dimana actor bisa berupa individu, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dikembangkan”. Use case berfungsi untuk memvisualisasikan

fungsionalitas sistem atau persyaratan yang harus dipenuhi oleh sistem dari sudut pandang tertentu.


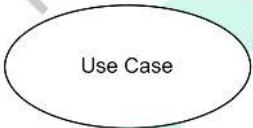


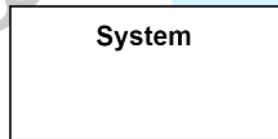
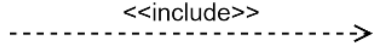
Lisnawanty (2014) menyatakan bahwa “Use Case adalah representasi pemodelan perilaku aplikasi perangkat lunak yang akan dibuat”. Use Case menjelaskan interaksi antara satu atau lebih actor dengan aplikasi yang sedang dikembangkan.

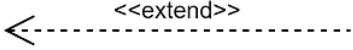
Dari beberapa definisi di atas, dapat disimpulkan bahwa Use Case menggambarkan fungsionalitas sistem atau persyaratan yang harus dipenuhi oleh sistem. Diagram ini juga mengilustrasikan interaksi antara satu atau lebih actor dengan aplikasi yang akan dikembangkan. Perilaku sistem yang dimaksud adalah apa yang dilakukan sistem dan bagaimana itu digunakan, bukan menjelaskan metode apa yang dilakukan agar sistem itu bekerja. Secara sederhana, diagram ini membantu pengguna memahami bagaimana sistem memenuhi kebutuhan mereka.



**Gambar 2.1.13 Contoh Use Case Diagram**

Tabel 2.1.13 Notasi Use Case Diagram

Simbol	Deskripsi
 <p>Actor</p>	<p>Aktor berinteraksi dengan fungsi sistem, memiliki tanggung jawab terhadap sistem (input), dan memiliki harapan terhadap sistem (output)</p>
 <p>Use Case</p>	<p>Abstraksi dan interaksi antara sistem dan aktor</p>
	<p>Association: Abstraksi dari penghubung antara aktor dengan use case</p>
	<p>Generalisasi: Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case</p>
 <p>System</p>	<p>Sistem mendefinisikan ruang lingkup use case dan digambarkan dalam bentuk persegi panjang. Elemen ini bersifat opsional, tapi biasa digunakan untuk memvisualisasikan sistem besar.</p>
 <p>&lt;&lt;include&gt;&gt;</p>	<p>Include: Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.</p>


	<p>Extend: Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi</p>
---	--



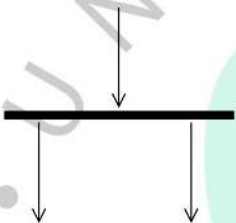
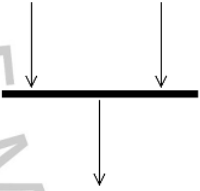
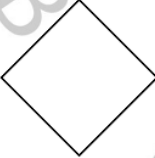
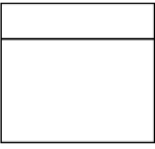
## 2. Activity Diagram

Menurut penelitian yang dilakukan oleh Lisnawanty (2014), “*Activity Diagram* merupakan suatu teknik yang digunakan untuk menguraikan secara rinci logika prosedural, proses bisnis, dan aliran kerja dalam berbagai situasi. Menurut Irmayani dan Susyatih (2017), “*Activity Diagram* juga berfungsi sebagai representasi visual dari aktivitas utama yang dilakukan oleh pengguna pada suatu sistem informasi”.

Dengan merangkum kedua pendapat tersebut, dapat disimpulkan bahwa *Activity Diagram* merupakan suatu metode yang digunakan untuk menggambarkan secara terinci logika prosedural, proses bisnis, dan aliran kerja dalam berbagai konteks. Selain itu, diagram ini juga mencerminkan aktivitas utama yang dilakukan oleh pengguna pada sistem informasi yang dibuat. Oleh karena itu, dapat diartikan bahwa *Activity Diagram* memiliki peran sebagai representasi visual aktivitas utama pengguna dalam konteks sistem informasi.

Tabel 2.1.14 Notasi Activity Diagram

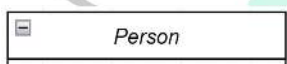



Simbol	Nama	Deskripsi
	<p><i>Start Point</i></p>	<p><i>Start Point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktivitas</p>

	<i>End Point</i>	<i>End Point</i> , akhir aktivitas
	<i>Activities</i>	Aktivitas, menggambarkan suatu kegiatan bisnis atau proses bisnis
	<i>Fork</i> atau Percabangan	<i>Fork</i> atau percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> atau Penggabungan	<i>Join</i> (penggabungan), digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i>	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, true atau false
	<i>Swimlane</i>	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

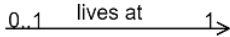


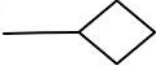
### 3. Class Diagram

Menurut (Wira et al., 2019) “*Class Diagram* adalah menjelaskan secara garis besar mengenai kelas-kelas perancangan sistem dari sudut pandang struktur sistem yang dapat memperjelas fungsi-fungsinya”. Atribut dan operasi merupakan bagian dari *Class Diagram* yang dapat memberi gambaran hubungan antara perancangan dan perangkat lunaknya sehingga sesuai dengan pembuatan programnya.

Tabel 2.1.15 Notasi Class Diagram

Simbol	Nama	Deskripsi
	Kelas	yang merepresentasikan obyek atau sekumpulan obyek yang memiliki persamaan struktur
	Atribut	merupakan variabel data, yang dapat memberikan informasi keadaan dimana tiap objek dari suatu kelas mempunyai nilai tersendiri.
	Asosiasi / <i>association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai multiplicity Generalisasi Relasi antark
	<i>Operation/method</i>	prosedur atau fungsi yang menjadi perilaku kelas-&-objek dan menjadi tanggung





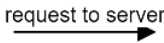
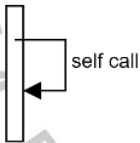

		jawab objek tersebut.
	Asosiasi berarah / <i>directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
	generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Kebergantungan/ <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas
	Agregasi / <i>aggregation</i>	Relasi antarkelas dengan makna wholepart


#### 4. *Sequence Diagram*

Menurut Rosa dan Shalahuddin (2015), “*sequence diagram* menggambarkan cara objek berinteraksi dalam situasi penggunaan dengan menunjukkan periode waktu di mana objek ada dan pesan yang dikirim dan diterima di antara objek tersebut”.

Tabel 2.1.16 Notasi Sequence Diagram

Simbol	Nama	Deskripsi
--------	------	-----------

	Aktor	Aktor, merujuk pada jenis peran yang dimainkan oleh suatu entitas yang berinteraksi dengan subjeknya
	Control Class,	Control Class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i>	<i>Message</i> , simbol mengirim pesan antar kelas
	Recursive	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	Activation	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi

	<i>Lifeline</i>	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek sepanjang linelife terdapat activation
---	-----------------	--

#### 2.1.4 Sistem Basis Data

##### 1. Basis Data

Menurut Jogiyanto (2005) dalam bukunya 'Sistem Informasi', "basis data adalah data di dalam basis data perlu diorganisasikan sedemikian rupa supaya dapat diakses dan dimanipulasi dengan mudah". Menurut Nurhadi dkk (2019), "Sistem Basis Data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan data yang diperlukan pemakai untuk proses tertentu".

Menurut Marlinda (2004), "sistem basis data ialah suatu sistem yang mengatur dan mengelola catatan-catatan dengan menggunakan komputer untuk menyimpan, merekam, serta merawat dan menjalankan seluruh operasional suatu organisasi atau perusahaan". Hal ini bertujuan agar dapat memberikan informasi yang optimal bagi pengguna guna mendukung proses pengambilan keputusan.

##### 2. Sistem Manajemen Basis Data (DBMS)

"Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang mempermudah pembuatan, pengaturan, dan pengelolaan basis data. Perangkat lunak ini memiliki kemampuan seperti menentukan struktur data, mengelola data, menjalankan kueri, dan mengatur transaksi. Peran utama DBMS adalah memastikan keberlanjutan, konsistensi, dan keamanan data" (Manik, 2023; A. Y. Rukmana, 2023b).

#### Komponen Basis Data

Sistem basis data terdiri dari komponen-komponen utama yang saling bekerjasama untuk mengakomodasi struktur penyimpanan data dan akses data. Berikut adalah komponen-komponen utama dari sebuah sistem basis data:

1. Tabel (Table): Tabel adalah entitas utama dalam basis data. Data dalam basis data diorganisir dalam format tabel yang terdiri dari baris dan kolom. Setiap baris dalam tabel mewakili satu rekaman atau data, sedangkan setiap kolom mewakili atribut atau field dari data tersebut.
2. Entitas (Entity): mewakili objek yang dapat disimpan dalam basis data, seperti orang, tempat, barang, atau kejadian. Atribut sebagai properti dari suatu entitas, contohnya seperti nama, nim, alamat, dan lainnya.
3. Atribut (Attribute): Atribut adalah karakteristik atau properti dari suatu entitas. Misalnya, jika entitasnya adalah "mahasiswa", atributnya mungkin termasuk nama, nim, alamat, dan lainnya.
4. Relasi atau Hubungan (Relationship): Menunjukkan bagaimana entitas-entitas terhubung satu sama lain. Relasi ini mencerminkan keterkaitan antar entitas dalam basis data.
5. Kueri (Query) digunakan untuk mengambil data dari basis data dengan cara yang fleksibel, memungkinkan pengguna untuk menyortir, memfilter, dan mengakses data sesuai kebutuhan mereka. Bahasa Basis Data (Database Language): Bahasa yang digunakan untuk berinteraksi dengan basis data. Contoh bahasa basis data termasuk SQL (Structured Query Language) yang umum digunakan.
6. Manajemen Transaksi (Transaction Management): Komponen ini bertanggung jawab untuk memastikan integritas data dan konsistensi basis data selama proses transaksi.
7. Keamanan Basis Data (Database Security): Melibatkan kontrol akses terhadap data untuk memastikan bahwa hanya pengguna yang diizinkan yang dapat mengakses dan memodifikasi informasi tertentu.
8. Indeks (Indexes): Digunakan untuk meningkatkan kecepatan pencarian data dalam basis data dengan membuat struktur data tambahan yang mempercepat proses pencarian.

9. Pemulihan (Recovery): Merupakan mekanisme untuk memulihkan basis data ke keadaan yang konsisten setelah terjadi kegagalan sistem atau kerusakan data.

Semua komponen ini bekerja secara bersama-sama, membentuk sistem basis data yang efisien, terorganisir, dan mudah diakses, memenuhi kebutuhan pengguna dengan baik."

### **2.1.3 Web Aplikasi**

Menurut Webster dan Brown (2017), "web adalah suatu ruang informasi elektronik yang terdiri dari halaman-halaman yang saling terhubung", sementara Menendez et al. (2018) mendefinisikan "web aplikasi sebagai program komputer yang dijalankan di peramban web untuk memberikan fungsionalitas tertentu kepada pengguna".

Web aplikasi merujuk pada suatu sistem informasi yang diakses melalui internet dan dirancang untuk memberikan pengalaman interaktif kepada pengguna.

### **2.1.6 Metode Penelitian Deskriptif Kualitatif**

Menurut Miles dan Huberman (2020): metode deskriptif kualitatif adalah pendekatan penelitian yang fokus pada pengumpulan, analisis, dan interpretasi data deskriptif untuk memahami konteks dan detail suatu fenomena.

Menurut Creswell dan Poth (2020): pendekatan deskriptif kualitatif melibatkan pengumpulan dan analisis data naratif, kutipan, dan temuan kualitatif lainnya untuk memberikan gambaran rinci tentang suatu fenomena.

Menurut Patton (2020): deskriptif kualitatif adalah metode penelitian yang bertujuan untuk menyediakan deskripsi yang rinci dan lengkap mengenai suatu fenomena, kejadian, atau konteks tertentu.

Metode deskriptif kualitatif adalah suatu pendekatan penelitian yang berfokus pada pengumpulan, analisis, dan interpretasi data deskriptif. Pendekatan ini digunakan untuk memahami konteks dan detail suatu fenomena dengan mendalam. Pendekatan tersebut melibatkan pengumpulan data naratif, kutipan, dan temuan kualitatif lainnya guna memberikan gambaran yang rinci tentang suatu

fenomena. Metode ini bertujuan menyediakan deskripsi yang lengkap dan mendalam terkait dengan suatu kejadian atau konteks tertentu.

Dengan demikian, metode deskriptif kualitatif menjadi alat penting dalam penelitian kualitatif untuk menjelajahi dan menggambarkan fenomena dengan cermat, memberikan pemahaman yang mendalam terhadap konteks, dan menghasilkan deskripsi yang kaya tentang suatu subjek penelitian.

### **2.1.5 Laundry**

Jasa laundry adalah kegiatan pembersihan dan perawatan pakaian atau tekstil lainnya oleh pihak ketiga yang secara profesional menangani proses pencucian dan perawatan. Definisi ini sejalan dengan pandangan Simons (2016), yang menggambarkan jasa laundry sebagai layanan yang menyediakan proses pencucian, pengeringan, dan pelipatan pakaian dengan standar kebersihan dan kualitas tertentu.

Menurut Bagyono (2006:59), laundry adalah bagian dari *housekeeping* yang bertanggung jawab atas pencucian, baik pencucian pakaian tamu (*laundry, dry cleaning dan pressing*) seragam karyawan maupun linen linen hotel (*house laundry*) dan pencucian pakaian atau linen dari luar hotel (*outside laundry*). Sedangkan menurut Rumekso SE,(2001:23), Laundry adalah seksi yang bertanggung jawab atas semua cucian yang dikirim kepadanya. Jadi secara umum laundry adalah bagian yang bertanggung jawab atas semua cucian yang diterima kepadanya, baik dari house laundry maupun dari tamu atau guest laundry.ww

### **2.1.15 Visual Studio Code**

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman Javascript, Typescript, dan Node. Js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code seperti : C++, C#, Python, Go, Java, PHP, dst. (Ummy Gusti Salamah, 2021).

## 2.2 Tinjauan Studi

Berikut beberapa referensi yang mendukung penelitian ini, antara lain:

1. Jurnal penelitian yang dilakukan oleh Juni Ary Pangga dengan judul Penelitian ini membahas mengenai **“Analisis dan Perancangan Sistem Informasi Berbasis Web Pada Jasa Laundry Rumah Cuci DOYAZ CLEANS Pulang Pisau”**. Penelitian ini membahas mengenai rancangan sistem pelayanan laundry di website dengan menggunakan database mysql dan xampp, serta php sebagai front end, dengan HTML menjadi bahasa markup yang digunakan untuk menampilkan data.
2. Jurnal penelitian yang dilakukan oleh Anwar Azmi Abdullah dengan judul **“Sistem Informasi Pelayanan Jasa Laundry pada Kedai Laundry Bandung Berbasis Desktop”**. Penelitian ini merancang sebuah sistem informasi pelayanan jasa laundry berbasis desktop untuk mempermudah transaksi pelayanan jasa bagi customer dan juga membantu perusahaan dalam pengolahan data, perbedaan disini adalah jurnal ini menghasilkan desktop apps.
3. Jurnal penelitian yang dilakukan oleh Erna Susanti, Alusyanti Primawati dan Adhi Susano yang berjudul **“PERANCANGAN SISTEM INFORMASI PELAYANAN JASA ZIFA LAUNDRY BERBASIS JAVA”**. Penelitian ini membahas system pada pelayanan Jasa Zifa Laundry, menggunakan aplikasi berbasis java dengan bahasa pemrograman JAVA.
4. Jurnal penelitian yang dilakukan oleh Bayu Isman yang berjudul **“SISTEM INFORMASI JASA LAUNDRY SEPATU DI SHOES CARE MENGGUNAKAN WEB RESPONSIF”**. Penulis bertujuan untuk meneliti di laundry sepatu ini agar membuat proses bisnis yang ada di dalam perusahaan menjadi lebih baik dan efisien, web responsif berguna untuk akses web yang tetap responsif jika diakses via *mobile*.
5. Jurnal penelitian yang dilakukan oleh Ani Nuraini yang berjudul **“SISTEM INFORMASI JASA LAUNDRY BERBASIS WEB PADA LAUNDRY KEISYA KARAWANG”**. Penelitian ini membahas mengenai rancangan



sistem pelayanan laundry berbasis web di website Jasa Laundry Keisya Karawang.

