

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Perancangan

Perancangan merupakan sebuah kegiatan menyimpulkan hasil studi yang telah diteliti ke dalam sebuah sistem dan kemudian dilakukan praktik yang lebih mendalam lagi. Tujuan dari sebuah perancangan pada sistem adalah agar para pakar yang terlibat dalam pembangunan suatu sistem, bisa mendapatkan representasi sistem yang akan dibangun secara nyata. Perancangan dibuat sesederhana mungkin dengan manfaat yang banyak agar mudah pada saat pemakaiannya (Ziliwu et al., 2021). Sutabri (Ziliwu et al. 2021) menjelaskan bahwa rancang merupakan sebuah aktivitas dengan target membuat desain baru pada sebuah sistem, yang dapat membasmi persoalan-persoalan yang ada pada perusahaan dan didapat dari hasil penyeleksian opsi sistem terbaik.

2.1.2 Sistem Informasi

Sistem merupakan sebuah kumpulan dari elemen-elemen yang berupa data, jaringan kerja dari serangkaian metode yang saling berinteraksi, sumber daya manusia, teknologi yang berupa *software* dan *hardware*. Semua komponen tersebut saling terkoneksi satu sama lain dalam rangka mencapai tujuan yang sama. Informasi merupakan data yang telah diolah untuk kemudian dapat digunakan atau dimanfaatkan oleh penerimanya. Sistem informasi merupakan sebuah sistem yang terdapat pada suatu organisasi yang mengaitkan kebutuhan pelaksanaan akan transaksi rutin, bersifat

administratif dengan strategi tertentu dari suatu organisasi yang dapat dicetak secara tersusun pada sebuah laporan tertentu (Julianti Ramaddan M. et al., 2019). Biasanya sistem informasi melibatkan beberapa orang dalam satu atau lebih divisi. Tujuannya adalah membuat pembagian tugas lebih terstruktur dan masing-masing divisi akan mengerjakan bagian mereka sendiri melalui sistem informasi tersebut.

2.1.3 Basis Data

Basis merupakan tempat penampungan atau dapat juga dikatakan sebagai gudang. Sementara data adalah sebuah kumpulan informasi yang didapat melalui sumber atau metode tertentu. Basis data merupakan gabungan kelompok data yang saling terkait dan disusun sesuai keperluan agar bisa diakses dan digunakan secara mudah (Dhika et al., n.d.).

2.1.4 Website

Website merupakan susunan-susunan halaman dengan tiap halaman memiliki informasi yang berbeda. Informasi yang ditampilkan bisa berupa teks, gambar, audio, ataupun gabungan ketiganya yakni video. Pengguna dapat mengakses informasi–informasi pada *website* dengan syarat harus terkoneksi internet (Abdulloh, 2016).

2.1.5 Metode *Prototype*

Metode *prototype* merupakan sebuah tahapan pengembangan *software* yang memungkinkan para pengembang sistem berkorelasi dengan pengguna sistem. Tujuannya adalah menemukan keselarasan antara pengembang dan pengguna sistem sehingga tercipta keadilan untuk kedua belah pihak. Pengguna dapat menggambarkan model sistem yang mereka butuhkan ke pengembang sistem. Hal tersebut

memungkinkan terpakainya seluruh fitur yang akan dibuat pengembang pada sistem (Zuhri et al., n.d.). Dikutip dari penelitian yang dibuat oleh Khanza Azeez Obayes dan Ali Hamzah yang berjudul “*Using of prototyping in develop an employee information management*” pada tahun 2022, penggunaan metode *prototype* mengurangi biaya dan waktu desain dengan performa sistem lebih cepat dan efisien.

2.2 Teori Khusus

2.2.1 Wihara Amitabha Batam



Gambar 2.1 Gedung Wihara Amitabha Batam

(Sumber : <https://www.buddhayana.or.id/vihara/102/wihara-amitabha-batam>)

Wihara Amitabha Batam merupakan salah satu tempat ibadah untuk umat beragama Buddha yang beralamat di Komplek Sumber Agung Blok H No. 7-9, Sungai Jodoh, Batu Ampar, kota Batam dan sudah berdiri di kota Batam sejak pertengahan tahun 2000-an. Wihara ini didirikan oleh Yayasan Amitabha Batam di dalam kompleks perumahan Sumber Agung yang awal mulanya hanya menggunakan satu gedung dan telah diekspansi menjadi tiga gedung yang digabung menjadi satu. Wihara Amitabha

Batam memiliki kegiatan rutin mingguan berupa Puja Bhakti melafal nama Buddha Amitabha setiap hari kamis, Puja Bhakti Palivacana atau kebaktian umum setiap hari sabtu, Namaskara 108 Buddha dan Sekolah Minggu Buddhis untuk tingkatan TK – Umum setiap hari minggu. Selain itu, wihara Amitabha Batam juga rutin menjalankan kegiatan rutin tahunan yang cukup besar seperti Imlek, Ceng Beng, Ulambana, 4 hari raya agama Buddha (Magha Puja, Waisak, Asadha Kathina), dsb. Kegiatan-kegiatan di wihara Amitabha Batam dijalankan oleh sebuah organisasi yang beranggotakan remaja-remaja beragama Buddha yang disebut dengan Persatuan Muda-mudi Buddhis Wihara Amitabha Batam (PMBWA) dan dibantu oleh bapak-bapak serta ibu-ibu beragama Buddha yang telah berkeluarga dan merupakan anggota dari Yayasan Amitabha Batam serta Wanita Buddhis Indonesia Wihara Amitabha Batam.

2.2.2 HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>HTML</title>
7 </head>
8 <body>
9   <h1 class="judul">Bahasa HTML</h1>
10 </body>
11 </html>
```

Gambar 2.2 Contoh Sintaks *HTML*
(Sumber : Data peneliti, 2023)

HTML merupakan singkatan dari *Hyper Text Mark Up Language* dan sebuah bahasa yang digunakan sebagai kerangka dasar dari sebuah halaman web (Tampubolon & Simanjuntak, 2023). Dengan menggunakan *HTML*, para pembuat web dapat

menampilkan informasi secara umum yang terkoneksi dengan internet. (Henderson dalam (Pahlevi, Mulyani, and Khoir 2018).

2.2.3 CSS

```
1  .class {  
2      display: grid;  
3      border: 2px solid rebeccapurple;  
4      font-family: Arial, Helvetica, sans-serif;  
5      font-size: 2vh;  
6      color: lightblue;  
7      position: relative;  
8  }
```

Gambar 2.3 Contoh Sintaks CSS
(Sumber : Data peneliti, 2023)

CSS merupakan singkatan dari *Cascading Style Sheet* dan merupakan sebuah bahasa yang dapat membuat tampilan sebuah web menjadi lebih menarik dan rapi. Jika sebuah halaman web hanya berbasis *HTML*, maka para pengunjung situs akan kesulitan melihat informasi yang disajikan. CSS sendiri dapat menghias ukuran *font*, warna *font*, warna latar belakang web, gambar-gambar yang diperlukan dalam web, dsb. CSS dibuat terpisah dengan *HTML* agar mempermudah para pengembang web sewaktu-waktu ingin merubah struktur dari *HTML* maupun CSS tersebut. (Henderson dalam (Pahlevi et al. 2018).

2.2.4 JavaScript

```
1  const mahasiswa = {  
2      namaDepan: "Willy",  
3      namaBelakang: "Kho",  
4      npm: 190210018,  
5      programStudi: "Teknik Informatika"  
6  }  
7  
8  console.info( data[0]: mahasiswa);
```

Gambar 2.4 Contoh Sintaks Pemrograman JavaScript

(Sumber : Data peneliti, 2023)

JavaScript merupakan sebuah bahasa pemrograman yang banyak digunakan pada halaman web. *JavaScript* bersifat *Client Side Programming Language* yang berarti pemrosesan bahasa pemrogramannya dilakukan oleh *client*. *Client* disini berupa sebuah *web browser* yang biasa digunakan seperti *Google Chrome*, *Safari*, dan sebagainya. Bahasa pemrograman ini muncul di era 1990-an. Walaupun terdapat kata *Java* pada *JavaScript*, bahasa pemrograman ini sama sekali berbeda dengan bahasa pemrograman *Java* yang sudah ada terlebih dahulu. Hampir serupa dengan *CSS*, *JavaScript* juga dibuat terpisah dengan *HTML*, terutama jika sebuah halaman web menggunakan *JavaScript* dengan banyak. *JavaScript* sendiri berperan agar sebuah halaman web lebih interaktif dan terasa lebih futuristik (Henderson dalam (Pahlevi et al. 2018).

2.2.5 PHP



Gambar 2.5 PHP
(Sumber : <https://php.net>)

PHP (Hypertext Preprocessor) merupakan sebuah bahasa berbentuk skrip yang yang diletakkan dan diproses pada *server*, dimana bahasa tersebut diterjemahkan

terlebih dahulu, dan hasil terjemahan akan dikirim ke browser *client* (T. Abdillah et al., 2023)

2.2.6 Framework Laravel



Gambar 2.6 Framework Laravel
(Sumber : <https://laravel.com/>)

Laravel adalah sebuah *framework* web atau kerangka kerja dari bahasa pemrograman *PHP* yang bersifat *open-source*. *Framework* ini diciptakan oleh Taylor Otwell dengan tujuan pengembangan aplikasi web yang memakai metode *MVC*. Struktur metode *MVC* pada *Laravel* terdapat *routing* yang menjembatani antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima request tersebut (T. Abdillah et al., 2023).

2.2.7 MySQL



Gambar 2.7 MySQL
(Sumber : <https://data-flair.training/blogs/javascript-frameworks/>)

MySQL (MyStructureQueryLanguage) merupakan *database* yang termasuk *DBMS* bersifat *open source* dan berfungsi untuk mengelola database atau manajemen data. *MySQL* dikenal sebagai sistem yang memiliki kinerja yang efisien dan reliable, serta proses *query* yang cepat dan mudah, sehingga cocok digunakan untuk aplikasi berbasis web (Hakam et al., 2022).

2.2.8 Web Hosting



Gambar 2.8 *Web Hosting*

(Sumber : <https://badoystudio.com/pengertian-hosting-dan-fungsinya/>)

Web hosting bisa dikatakan sebagai tempat bernaungnya sebuah *website*. *Web hosting* diibaratkan sebagai rumah sementara, *domain* merupakan alamat dari rumah tersebut. *Web hosting* sendiri memiliki peran sebagai penyedia layanan bagi para perancang web agar *website* mereka dapat diakses oleh publik melalui *World Wide Web*. Para penyedia layanan *web hosting* memberikan ruang kosong pada sebuah *server* agar dokumen- dokumen halaman web bisa tersimpan. Selain itu, disediakan juga sambungan ke internet agar pengguna dapat mengakses *website* yang telah dikembangkan para perancang. *Web hosting* juga melayani jasa titip *server* dari pihak lain dengan sambungan internet yang terakses pada pusat data (Hartati, 2020).

2.2.9 Adobe Illustrator



Gambar 2.9 Adobe Illustrator

(Sumber : <https://www.adobe.com/products/illustrator.html>)

Adobe Illustrator merupakan *software* khusus untuk desain grafis dan dapat merancang berbagai macam desain berbasis vector. Hasil yang diberikan antara *adobe illustrator* dan *photoshop* akan berbeda dikarenakan perbedaan pada kegunaan toutput keduanya, namun tetap memiliki keunggulan dan kekurangannya masing-masing (Vieri et al., 2023).

2.2.10 Visual Studio Code



Gambar 2.10 Logo Visual Studio Code

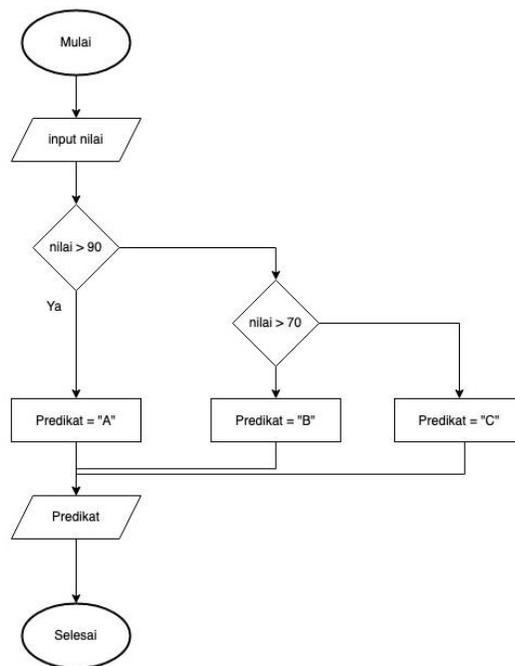
(Sumber : <https://code.visualstudio.com/>)

Visual Studio Code merupakan sebuah perangkat lunak untuk membuat kode pada perangkat lunak ataupun web yang akan dibangun. *Visual Studio Code* merupakan perangkat lunak yang cukup ringan namun, sangatlah kuat. *Visual Studio Code*

mendukung banyak bahasa pemrograman seperti *JavaScript*, *C++*, *C#*, *Python*, *PHP*, dan sebagainya (Hartati, 2020).

2.2.11 Flowchart

Menurut Bambang Hartono (dalam (Dewi and Malfiany 2017), *flowchart* adalah sebuah kerangka dasar dari penggambaran sebuah tahapan kerja sebuah sistem dan berkesinambungan satu sama lain. Pada saat mendesain alur kerja sebuah sistem dengan *flowchart*, digunakan beberapa simbol-simbol khusus dengan fungsi yang berbeda-beda. Selain itu, simbol-simbol tersebut akan merepresentasikan bagaimana hubungan sebuah fungsi dengan fungsi lainnya pada sistem secara garis besar.



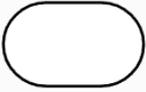
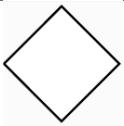
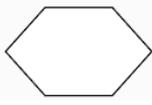
Gambar 2.11 Contoh *Flowchart* Sederhana

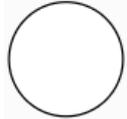
(Sumber : <https://www.heuyheuy.com/flowchart-dan-contoh-kasusnya/>)

Gambar di atas merupakan salah satu contoh kasus kondisi percabangan yang digambarkan secara sederhana dengan menggunakan *flowchart*. Jika kita menginput

nilai di atas 90, maka predikat yang tampil adalah “A”. Begitu pula saat kita menginput nilai diantara 90 dan 71, maka predikat yang tampil adalah “B”. Jika kita mnginput nilai di bawah 70, maka predikat yang tampil adalah “C”. Agar dapat lebih jelas memahami komponen pada *flowchart*, akan dijabarkan beberapa komponen *flowchart* yang umum digunakan dalam bentuk tabel sebagai berikut.

Tabel 2.1 Komponen pada *Flowchart*

Simbol	Nama	Fungsi
	<i>Connecting Line</i>	Digunakan untuk menggabungkan satu simbol dengan simbol lainnya.
	<i>Terminator</i>	Digunakan untuk menyatakan titik awal maupun titik akhir dari sebuah <i>flowchart</i> .
	<i>Input / Output</i>	Digunakan untuk menyatakan proses inputan atau untuk menampilkan output dari suatu nilai.
	<i>Decision</i>	Digunakan ketika ada suatu kondisi yang memungkinkan terdapat 2 jawaban antara iya atau tidak.
	<i>Preparation</i>	Digunakan untuk memberikan nilai awal pada saat deklarasi. Bisa juga digunakan sebagai proses perulangan sesuai kondisi.

	<i>Document</i>	Digunakan untuk menyatakan bahwa inputan berasal dari dokumen atau output yang perlu dicetak.
	<i>On-Page Reference</i>	Digunakan untuk menyatakan penyambungan sebuah <i>flowchart</i> dalam halaman yang sama.
	<i>Off-Page Reference</i>	Digunakan untuk menyatakan penyambungan sebuah <i>flowchart</i> dalam halaman yang berbeda.
	<i>Process</i>	Digunakan untuk menyatakan sebuah proses yang sedang berlangsung pada <i>flowchart</i> .
	<i>Manual Operation</i>	Digunakan untuk menyatakan proses yang tidak dilakukan oleh komputer.
	<i>Predefined Process</i>	Digunakan untuk pelaksanaan sebuah bagian atau prosedur.
	<i>Display</i>	Digunakan untuk peralatan output yang digunakan.

2.2.12 UML (*Unified modeling language*)



Gambar 2.12 Logo *UML*

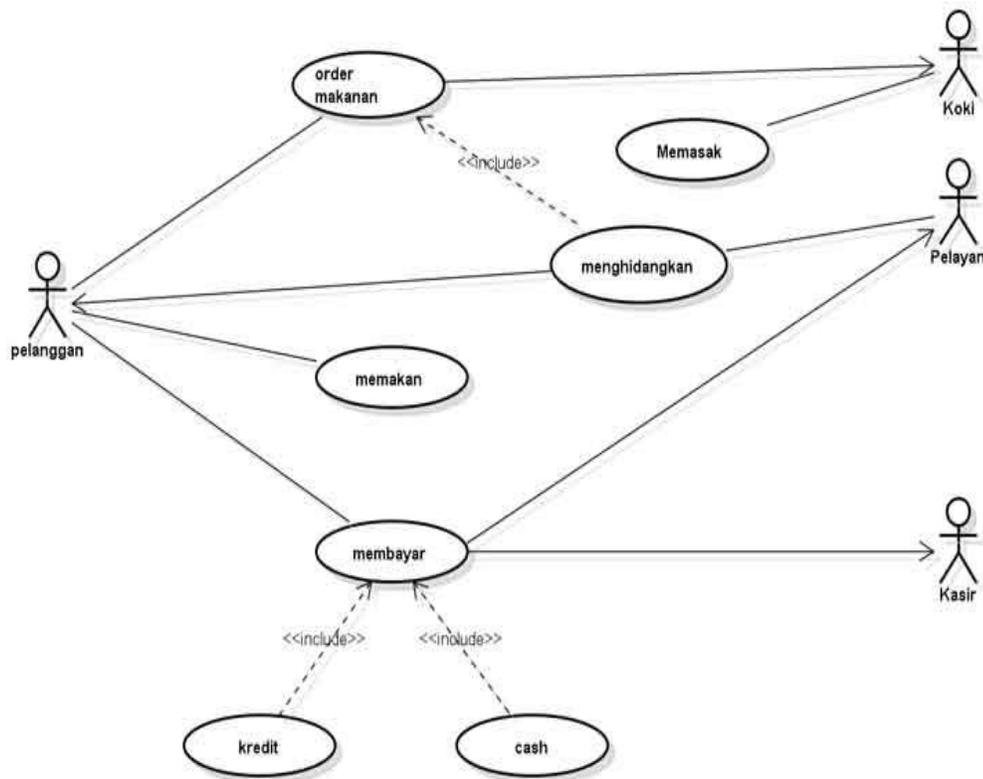
(Sumber : <https://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>)

Unified modeling language (UML) merupakan sebuah bahasa untuk memodelkan gambaran sederhana dari perangkat lunak yang akan dibangun. *UML* telah menjadi standar sebagai blueprints atau sebuah media penulisan bercetak biru untuk perangkat-perangkat lunak yang akan dibangun. *UML* dapat digunakan untuk merepresentasi, merancang, mengkhususkan, dan mengarsipkan bagian-bagian pada sistem yang akan dirancang pada perangkat lunak. Hal ini serupa dengan seorang arsitek yang biasanya akan menggambar struktur atau kerangka dasar dari sebuah bangunan, lalu dicetak pada sebuah dokumen bercetak biru yang nantinya dokumen tersebut bisa diarsipkan. *UML* sangat membantu para perancang ataupun programmer pada saat akan membuat sebuah perangkat lunak. Terdapat beberapa *UML* yang umum digunakan pada saat merancang sebuah perangkat lunak dan akan dijabarkan beserta fungsi-fungsi pada komponennya (R. Abdillah, 2021).

2.2.12.1 *Use Case Diagram*

Use Case Diagram adalah sebuah diagram *UML* yang merepresentasikan sebuah tingkah laku pada sistem yang akan dibangun. Diagram *use case* akan

menjelaskan hubungan antara satu atau lebih aktor dengan sistem yang akan dibangun. Secara garis besar, diagram *use case* dirancang agar memahami peran-peran pada sebuah sistem dan siapa saja yang bisa menjangkau sistem tersebut. Diagram *use case* merepresentasikan manfaat sesuai kebutuhan dari sistem. Representasi pada diagram *use case* lebih berfokus kepada apa yang akan dilakukan sebuah sistem, bukan bagaimana cara sistem melakukannya. Perancangan diagram *use case* membutuhkan satu aktor atau lebih sebagai penghubung dengan sistem. Aktor pada diagram *use case* bisa berupa manusia ataupun mesin yang menjalin hubungan dengan sistem dan akan menyelesaikan tugas-tugas sesuai kebutuhan (Sari & Istikoma, 2018).

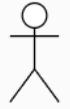
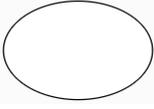
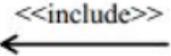


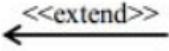
Gambar 2.13 Contoh *Use Case Diagram*

(Sumber : <https://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>)

Gambar di atas merupakan sebuah contoh sederhana dari *use case* diagram. Jika dilihat dari komponen-komponen yang ada, *use case* tersebut sedang merepresentasikan bagaimana sistem sederhana dari sebuah restoran. Komponen-komponen pada *use case* diagram akan dijabarkan dalam bentuk tabel sebagai berikut.

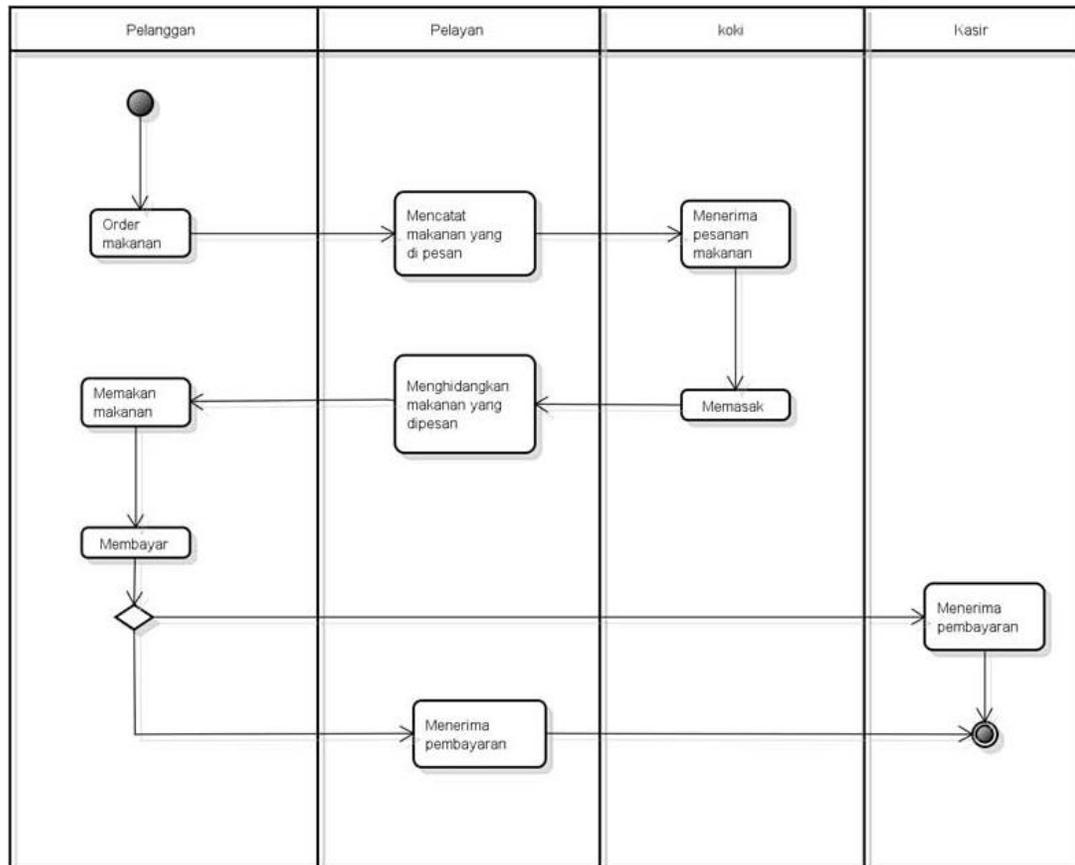
Tabel 2.2 Komponen pada *Use Case Diagram*

Simbol	Nama	Fungsi
	<i>Actor</i>	Mewakili peran orang, pengguna, sistem yang lain, atau alat ketika berkomunikasi dengan <i>Use Case</i> .
	<i>Use Case</i>	Sebuah fungsi dari sebuah sistem yang nantinya dapat berkorelasi dengan aktor.
	<i>Association</i>	Sebagai penghubung antara actor dengan <i>Use Case</i> maupun sebaliknya.
	<i>Generalization</i>	Sebagai fungsi khusus yang dimiliki oleh actor dan dapat berkorelasi dengan <i>Use Case</i> tertentu.
	<i>Include</i>	Sebagai penghubung dan penanda bahwa sebuah <i>Use Case</i> merupakan bagian dari sebuah <i>Use Case</i> tertentu.

	<i>Extend</i>	Sebagai penghubung dan penanda bahwa sebuah <i>Use Case</i> dapat memiliki sebuah <i>Use Case</i> lainnya dengan kondisi tertentu.
---	---------------	--

2.2.12.2 Activity Diagram

Activity Diagram merepresentasikan sebuah aliran pada aktivitas antara sistem dan aktor yang mencakup sistem bisnis, alur kerja sebuah program, maupun aktivitas-aktivitas lainnya. Diagram *activity* menekankan apa atau siapa yang melakukan dan bertanggung jawab terhadap kapasitas yang sedang dilakukan. Jika kita melihat fungsi pada diagram *activity*, sekilas mirip dengan diagram alur atau *flowchart*. Tetapi terdapat perbedaan yang cukup mencolok, yakni diagram *activity* dapat merepresentasikan kelakuan antara aktor dengan sistem secara paralel. Biasanya diagram *activity* dibuat berpatokan pada diagram *use case* yang telah dibuat (Sari & Istikoma, 2018).



Gambar 2.14 Contoh Activity Diagram

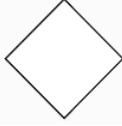
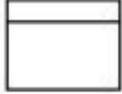
(Sumber : <https://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>)

Gambar di atas merupakan contoh dari *activity* diagram sebuah restoran.

Berikut merupakan tabel penjelasan komponen *activity* diagram.

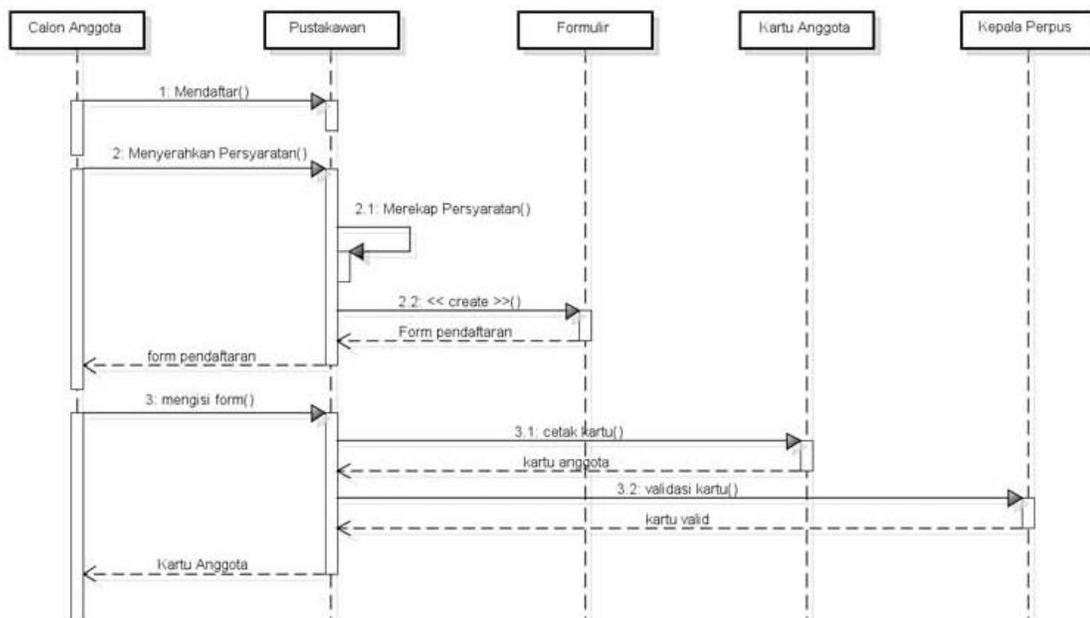
Tabel 2.3 Komponen Pada Activity Diagram

Simbol	Nama	Fungsi
	<i>Start</i>	Merupakan status awal yang memulai rentetan kerja yang akan dilakukan oleh komponen-komponen yang dipisahkan oleh swimlane.

	<i>Activity</i>	Sebuah aktivitas yang akan dijalankan oleh sistem sesuai dengan urutan yang ada.
	<i>Condition</i>	Akan dilakukan percabangan, dimana akan ada 2 pilihan dan 2 hasil yang berbeda. Biasanya pilihan tersebut berupa iya atau tidak dan benar atau salah.
	<i>Fork/Join</i>	Beberapa aktivitas yang berjalan dan kemudian akan menjadi satu jalur, akan digabungkan dengan simbol ini.
	<i>End</i>	Merupakan status akhir yang menandakan bahwa rentetan kerja yang telah dirancang sedemikian rupa telah selesai dijalankan dengan benar.
	<i>Swimlane</i>	Sebagai pemisah agar sistem bisa dibagi dan diproses oleh bagian-bagian yang berkompeten.

2.2.12.3 Sequence Diagram

Sequence Diagram merupakan sebuah diagram pada *UML* yang merepresentasikan kerjasama yang sedang aktif di antara objek-objek. Diagram *sequence* memiliki fungsi untuk menampilkan rentetan pesan dan interaksi antar objek. Diagram *sequence* akan memaparkan eksekusi yang terjadi pada sistem dengan sebuah sketsa semantik. Diagram ini juga dapat menerangkan interaksi yang terjadi pada objek berdasarkan urutan waktu. Sebuah objek akan mengirimkan pesan kepada objek lainnya yang bisa berupa sistem, kemudian pesan yang dikirim akan dikembalikan dalam bentuk tertentu sesuai sistem yang akan dirancang (Sari & Istikoma, 2018).



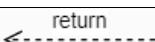
Gambar 2.15 Contoh *Sequence Diagram*

(Sumber : <https://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>)

Gambar di atas merupakan contoh dari *sequence* diagram. Pada contoh tersebut, digambarkan sebuah sistem dari perpustakaan. Berikut merupakan penjelasan dari komponen-komponen dari *sequence diagram*.

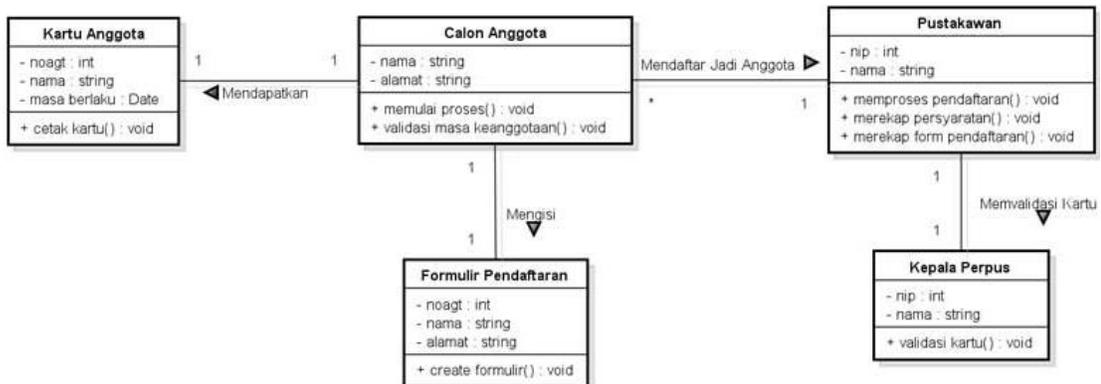
Tabel 2.4 Komponen pada *Sequence Diagram*

Simbol	Nama	Fungsi
	<i>Actor</i>	Merupakan pengguna atau <i>user</i> yang akan berinteraksi dengan sistem.
	<i>Activation Box</i>	Disaat sebuah <i>object</i> sedang melakukan tugasnya, maka akan dibuat <i>Activation Box</i> sesuai dengan berapa lamanya <i>object</i> tersebut melakukan tugasnya hingga selesai.
	<i>Lifeline</i>	Digambarkan dengan garis putus-putus. Memiliki fungsi untuk menggambarkan aktivitas dari <i>object</i> tersebut.
	<i>Object</i>	Menggambarkan bagaimana sebuah entitas dalam sistem akan berperilaku. Perilaku tersebut akan digabungkan dengan <i>Activation Box</i> sebagai penanda waktunya dan <i>Lifeline</i> sebagai penanda bahwa terdapat aktivitas pada <i>object</i> .

	<i>Message</i>	Sebagai komunikasi antara <i>object</i> . Biasanya panah ini akan disambungkan ke bagian <i>Activation Box</i> pada <i>object</i> .
	<i>Swimlane</i>	Merupakan keluaran atau <i>output</i> yang biasanya akan dikirimkan ke <i>actor</i> .

2.2.12.4 Class Diagram

Class diagram merupakan sebuah diagram yang merepresentasikan objek-objek yang ada pada sistem, yang menjadi poin utama untuk pengembangan serta penggambaran sebuah sistem berorientasi objek. Diagram *class* diperuntukkan menampilkan eksistensi terhadap kelas-kelas yang ada beserta korelasinya dalam ruang lingkup yang logis pada sebuah sistem. Diagram *class* terdiri dari nama *class* yang diikuti oleh bagian atribut dan bagian metode atau fungsi (Sari & Istikoma, 2018).



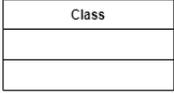
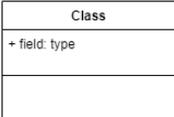
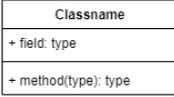
Gambar 2.16 Contoh *Class Diagram*

(Sumber : <https://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>)

Gambar di atas merupakan contoh dari *class diagram*. Bisa dilihat bahwa gambar tersebut menerjemahkan sistem perpustakaan menjadi bentuk *class diagram*.

Agar mudah memahami *class* diagram tersebut, dijabarkanlah komponen-komponen pada *class* diagram dalam bentuk tabel sebagai berikut.

Tabel 2.5 Komponen pada *Class* Diagram

Simbol	Nama	Fungsi
	<i>Class</i>	Merepresentasikan nama sebuah <i>class</i> . Nama kelas pasti berbeda-beda sesuai kebutuhan sistem yang akan dirancang.
	<i>Attribute</i>	Berisikan sebuah data yang memiliki nilai yang dapat mendeskripsikan <i>class</i> yang dibuat.
	<i>Method / Function</i>	Metode atau fungsi memungkinkan suatu <i>class</i> dapat berperilaku khusus saat kita memberikan metode atau fungsi sesuai dengan <i>class</i> yang dibuat.
	<i>Association</i>	Menghubungkan dua <i>class</i> berbeda yang memiliki kaitan satu sama lain
	<i>Aggregation</i>	Sebuah penghubung antar <i>class</i> yang mana salah satu <i>class</i> merupakan bagian dari <i>class</i> lainnya, namun <i>class</i> tersebut dapat berdiri sendiri seandainya <i>class</i> tersebut dihapus.

	<p><i>Composition</i></p>	<p>Sebuah penghubung antar <i>class</i> yang merupakan bagian dari <i>class</i> lain. <i>Class</i> yang terhubung tidak dapat berdiri sendiri. Seandainya sebuah <i>class</i> terhapus, maka <i>class</i> yang terhubung dengan <i>composition</i> akan ikut terhapus,</p>
	<p><i>Inheritance</i></p>	<p>Sebuah penghubung yang dapat mewarisi atribut dan fungsi pada suatu <i>class</i>. Sehingga, kita tidak perlu menulis lagi atribut dan fungsi yang sama.</p>

2.3 Penelitian Terdahulu

Untuk memperluas pengetahuan dan sudut pandang dalam pembuatan penelitian, maka dikumpulkanlah beberapa penelitian terdahulu dalam bentuk tabel yang sekaligus menjadi acuan dalam melakukan penelitian ini.

Tabel 2.6 Tabel Penelitian Terdahulu

No	Penulis	Judul	Kesimpulan
1	Khanza Azeez Obayes, Ali Hamzah (Obayes & Hamzah, 2022)	<i>Using of prototyping in develop an employee information management Journal ScienceDirect</i>	Penggunaan metode <i>prototype</i> menawarkan banyak keuntungan seperti berkurangnya biaya dan waktu desain, perancangan lebih cepat, mendapatkan desain yang

		<i>Measurement: Sensor 24 (2022) 100557</i>	lebih ideal, performa sistem lebih cepat dan efisien.
2	M'hamed Rahmouni, Mouad Bouzaidi, Samir Mbarki (Rahmouni et al., 2023)	<i>Approach by modeling to generate an e-commerce web code from Laravel model Indonesian Journal of Electrical Engineering and Computer Science Vol.30, No.1, April2023, pp. 257~266 ISSN: 2502-4752, DOI: 10.11591/</i>	Penggunaan <i>framework</i> dalam pembuatan <i>website</i> di masa kini dinilai sangat efisien. <i>Website</i> yang kompleks dapat dirancang lebih mudah menggunakan <i>framework</i> dan <i>Laravel</i> merupakan salah satu contohnya.
3	Mohammed Saleh Mohammed Ba Karman, Cahyo Darujati, Anni Wulandari	<i>A Fully Functional Enterprise Resource Planning (ERP) System for Honey Home Company in Indonesia</i>	Pengimplementasian sistem ERP pada perusahaan Honey Home menghemat waktu hingga 80%, menyajikan informasi yang berkualitas hingga 88,6%, bantuan serta layanan yang baik dan juga kepuasan pengguna

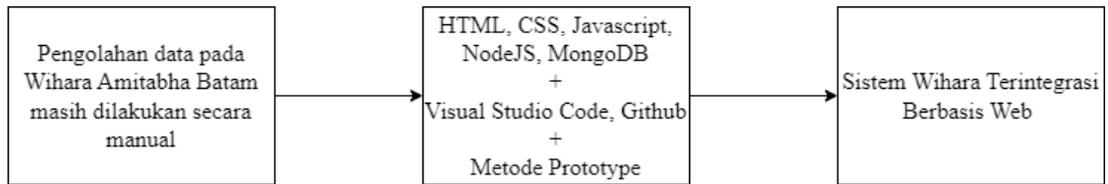
	(Saleh et al., 2022)	<i>International Journal of Electrical Engineering and Information Technology</i> Vol. 05, No. 02, September 2022 E-ISSN : 2615-2096	meningkat 84,2%, dan meningkatkan kinerja karyawan hingga 77,2%.
4	Tajuddin Abdillah, Roviana H. Dai, Sry Yunarti, Rachmat Rifaldi Hadju (T. Abdillah et al., 2023)	Perancangan Sistem Informasi Penguatan Kapasitas Mahasiswa, Alumni dan Organisasi Kemahasiswaan Menggunakan <i>Framework Laravel</i> Digital Transformation Technology Vol. 3, No. 2, September 2023 E-ISSN : 2807-9000	Perancangan sistem dengan framework Laravel dan metode prototype dapat berjalan dan diimplementasikan di berbagai perguruan tinggi yang menerapkan pengelolaan keuangan Badan Layanan Umum.
5	Sukrisna Andrianto, Hadion Wijoyo	Rancang Bangun Sistem Informasi Siswa Berbasis Web di Sekolah Minggu	Dengan adanya sistem informasi siswa berbasis web di SMB Vihara Dharmaloka Pekanbaru,

	(Andrianto & Wijoyo, 2020)	Buddha Vihara Dharmaloka Pekanbaru Terapan Informatika Nusantara Vol. 1, No.2, Juli 2020 E-ISSN : 2722-7987	membuat pengolahan data guru, data siswa, data nilai siswa, dan data lainnya serta laporan semua data menjadi lebih mudah dan cepat serta meminimalisir kesalahan karena sudah terprogram.
6	Syafaatun Khasanah, Affandy (Khasanah & Affandy, 2019)	Sistem Informasi Kepakaran Pegawai Berbasis <i>Odoo</i> dengan Metode <i>Prototype</i> pada PT Z OISAA Journal Vol. 2, No. 1, 2019 ISSN : 2615-675X	Sistem informasi yang sudah dirancang dengan metode <i>prototype</i> memudahkan proses pengajuan tender dalam megalokasikan sumber daya yang sesuai dengan sertifikat keahlian yang dimiliki oleh pegawai PT Z. Sistem tersebut juga memudahkan dalam mengetahui masa berlaku dan bidang sertifikasi dari sertifikat milik pegawai PT Z. Sistem yang baru dirancang juga lebih cepat pengolahan datanya jika

			dibandingkan dengan sistem lama.
7	Fajri Abimayu, Pastima Simanjuntak (Abimayu & Simanjuntak, 2023)	ANALISIS DAN PERANCANGAN APLIKASI E-LIBRARY SEKOLAH BERBASIS WEB MENGGUNAKAN METODE SCRUM Jurnal Comasie Vol. 9, No. 2, 2023 E-ISSN : 2715-6265	Pengembangan aplikasi <i>e-library</i> berbasis web menggunakan metode Scrum dan memanfaatkan bahasa pemrograman <i>PHP Framework</i> Laravel serta sistem basis data <i>MySQL</i> yang memperhatikan kebutuhan siswa dan petugas perpustakaan di sekolah. Aplikasi <i>e-library</i> berbasis <i>web</i> mempermudah siswa dalam memeriksa ketersediaan buku dan petugas perpustakaan dalam mengelola data di perpustakaan.

2.4 Kerangka Pemikiran

Berdasarkan dari permasalahan dan peluang yang ada, maka dibuatlah sebuah kerangka pemikiran dengan tujuan memanfaatkan peluang yang ada, untuk kemudian dicari salah satu solusi untuk permasalahan, agar permasalahan tersebut bisa teratasi. Berikut gambaran kerangka pemikiran penulis secara sederhana.



Gambar 2.17 Kerangka Pemikiran
(Sumber : Data olahan penulis, 2023)