

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian terdahulu /tinjauan pustaka dapat dilihat pada tabel dibawah ini :

Tabel 2.1 Penelitian Terdahulu

No.	Nama	Judul Penelitian	Metode	Hasil
1.	(Hanif et al., 2024)	Implementasi Economic Order Quantity Pada Sistem Informasi Inventory Berbasis Website Untuk Agung Rezeki Elektro	<i>Economic Order Quantity (EOQ)</i>	Sistem inventory yang dibuat dapat mengelola stok barang secara otomatis berdasarkan input pengguna, meminimalisir kesalahan, dan terintegrasi dengan database untuk menjaga keamanan dan kerahasiaan data. Penambahan fitur Perhitungan Economic Order Quantity (<i>EOQ</i>) membantu dalam pengendalian stok barang, menghemat biaya pembelian, dan mengoptimalkan pengelolaan persediaan.
2.	(Fahruliansyah et al., 2023)	Implementasi Metode Economic Order Quantity (<i>EOQ</i>) Dalam Sistem Pengendalian Inventory Di PT Sinergi Kreasi Utama	<i>Economic Order Quantity (EOQ)</i>	Penerapan metode Economic Order Quantity (<i>EOQ</i>) dalam aplikasi inventory membantu meminimalisir biaya pemesanan barang secara efektif dan efisien. Aplikasi ini memungkinkan perusahaan mengontrol persediaan barang, melakukan stock opname, dan menghasilkan informasi serta laporan persediaan yang tepat waktu, akurat, dan sesuai kebutuhan.

No.	Nama	Judul Penelitian	Metode	Hasil
				Dibangun menggunakan PHP dan database <i>MySQL</i> , aplikasi ini memberikan alert atau peringatan untuk stok minimum produk, membantu menghindari penumpukan barang di gudang.
3	(Darusalam & Sugiyono, 2021)	Penerapan Metode <i>EOQ</i> Dalam Pengendalian Persediaan Barang Pada PT.Cakraindo Mitra Internasional	<i>Economic Order Quantity (EOQ)</i>	Penelitian ini mengidentifikasi masalah pengarsipan manual di divisi gudang PT. Cakraindo Mitra Internasional dan mengusulkan solusi menggunakan metode <i>EOQ (Economy Order Quantity)</i> . Melalui pengumpulan data dan penerapan metode <i>EOQ</i> , penelitian ini menghasilkan sistem berbasis web yang membantu pengguna melihat interval transaksi pelanggan dalam setahun terakhir. Ini memungkinkan PT. Cakraindo Mitra untuk memahami pola transaksi pelanggan dan jumlah barang yang dipesan selama interval tersebut.
4	(Gung Prabawa et al., 2018)	Pengembangan Sistem Pendukung Keputusan Pengendalian Persediaan Barang Menggunakan Metode <i>Economic Order Quantity (EOQ)</i> dan MIN-MAX Berbasis Web (Studi Kasus : Apotek Sahabat Kita)	<i>Economic Order Quantity (EOQ)</i>	Hasil penelitian menunjukkan bahwa pengembangan sistem pendukung keputusan pengendalian persediaan barang menggunakan metode <i>EOQ (Economic Order Quantity)</i> dan Minmax dengan model The Waterfall mendapat respon positif dari pengguna, dengan tingkat pencapaian

No.	Nama	Judul Penelitian	Metode	Hasil
				mencapai 96.6% dari admin dan 97.3% dari pegawai dalam hal kebermanfaatan sistem di Apotek Sahabat Qita. Saran untuk pengembangan lebih lanjut termasuk pengembangan sistem berbasis web online untuk mengirim laporan order langsung ke supplier, memberikan item discount pada pemesanan, meningkatkan user interface, dan mengembangkan sistem pendukung keputusan dengan memperhitungkan faktor-faktor seperti hari raya dan kadaluarsa barang.
5	(Rafliana & Suteja, 2018)	Penerapan Metode <i>EOQ</i> dan <i>ROP</i> Untuk Pengembangan Sistem Informasi Inventory Bengkel MJM Berbasis Web	<i>Economic Order Quantity (EOQ) dan ROP</i>	Sistem Informasi Pengelolaan Inventory Bengkel MJM berbasis web dengan metode <i>EOQ</i> dan <i>ROP</i> memiliki kemampuan untuk melakukan transaksi pembelian dan penjualan barang serta jasa dengan tampilan yang mudah digunakan. Penggunaan metode <i>EOQ</i> memungkinkan perhitungan optimal untuk pembelian barang dengan meminimalkan biaya pesan dan biaya simpan, sementara metode <i>ROP</i> membantu menentukan waktu yang tepat untuk melakukan pemesanan kembali, mengoptimalkan pembelian barang.

Berdasarkan peneltiian terdahulu yang telah dilakukan berikut ini perbedaan atau kelebihan peneliti yaitu :

1. Tempat study kasus yang diambil.
2. Sistem dapat dikelolah oleh admin dan pimpinan.
3. Sistem dapat mencetak laporan secara periode.
4. Sistem dapat menampilkan detail hasil perhitungan.
5. Metode pengembangan system yang *digunakan Waterfall SDLC (Software Development Life Cycle)*.
6. Metode pengujian sistem yang digunakan *Black Box Testing* dan *Sistem Usability Scale (SUS)*.

2.2 Sistem Informasi

Menurut (Tukino, 2020) sistem merupakan suatu kesatuan yang terdiri dari berbagai unsur yang saling melengkapi untuk mencapai tujuan dan sasaran yang telah ditetapkan. Unsur-unsur dalam sistem disebut sebagai subsistem yang harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan agar sistem dapat bekerja secara efektif dan efisien. Sistem juga dapat didefinisikan sebagai kumpulan objek yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu. Dalam konteks teknologi informasi, sistem informasi merupakan kombinasi teratur dari orang-orang, hardware, software, jaringan komunikasi, dan sumber daya data yang digunakan untuk mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi juga berfungsi untuk memproses transaksi rutin, memberi sinyal kepada manajemen terhadap

kejadian-kejadian penting, dan menyediakan dasar informasi untuk pengambilan keputusan.

2.3 Rancang Bangun

Rancang bangun adalah proses yang melibatkan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi. Dengan kata lain, ini adalah kegiatan yang mengubah hasil analisis menjadi paket perangkat lunak dan mengimplementasikan atau memperbaiki sistem yang telah ada.(Rauf & Prastowo, 2021)

Menurut (Tuwllah et al., 2020) Perancangan sistem dapat didefinisikan sebagai penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini menyangkut konfigurasi dari komponen-komponen perangkat lunak dan keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancangan bangun yang telah ditetapkan pada akhir tahap analisa system.

2.4 Sistem Inventory

Sistem inventory adalah bahwa itu adalah kumpulan kebijakan dan pengendalian yang mengatur dan memantau tingkat persediaan barang. Sistem ini bertujuan untuk menentukan kapan persediaan harus diisi ulang, seberapa banyak yang harus dipesan, dan bagaimana cara melakukannya. Tujuan akhirnya adalah untuk mengoptimalkan tingkat persediaan agar perusahaan dapat menjaga

ketersediaan barang yang memadai tanpa membiarkan stok terlalu banyak atau terlalu sedikit.

Sistem inventory juga melibatkan struktur organisasi dan kebijakan operasi produksi yang berkaitan dengan pengelolaan persediaan barang. Ini mencakup proses pemesanan, penerimaan, dan pengawasan barang yang dipesan. Manajer bertanggung jawab untuk memastikan bahwa pesanan ditempatkan tepat waktu, barang diterima dengan benar, dan bahwa hubungan dengan vendor dipertahankan. (Asiz & Hadi Sirad, 2019)

2.5 Metode *Economic Order Quantity* (EOQ)

EOQ, singkatan dari Economic Order Quantity, adalah model umum dalam manajemen persediaan yang sederhana. Model ini juga menjadi dasar bagi pengembangan model persediaan yang lebih kompleks. Terdapat berbagai pengembangan model persediaan untuk barang-barang dengan batasan yang berbeda, seperti barang dengan masa kadaluarsa atau dengan nilai depresiasi seiring waktu. Misalnya, ada model yang mempertimbangkan pengendalian sistem produksi untuk barang dengan nilai depresiasi. Selain itu, model EOQ juga telah dikembangkan untuk barang-barang kadaluarsa dengan biaya penyimpanan yang berubah seiring waktu.

Penentuan jumlah pemesanan dalam model EOQ dapat dilakukan dengan dua cara: pemesanan individu untuk setiap barang (single-item) atau pemesanan bersama-sama untuk beberapa barang (multi-items). Dalam pemesanan bersama-sama, perusahaan perlu menentukan periode pemesanan (T) secara bersama-sama untuk meminimalkan biaya pesan (O_p) dan mengoptimalkan total biaya persediaan (O_T). (Agung Awaludin Sayuti, 2022)

Metode perhitungan menggunakan Model *EOQ* bertujuan untuk menentukan jumlah pemesanan agar dapat memenuhi persediaan barang, dengan tujuan meminimalkan biaya penyimpanan dan biaya pemesanan yang timbul. Model *EOQ* mengasumsikan beberapa kondisi serupa dengan yang terdaftar di bawah ini .(Hanif et al., 2024).

1. Permintaan stabil dan independen.
2. Lead time (waktu tunggu) yang tetap.
3. Tersedianya produk setiap saat.
4. Tidak ada diskon harga.
5. Biaya penyimpanan dan biaya pemesanan dapat berubah sepanjang waktu.
6. Pesanan dilakukan tepat waktu untuk mencegah kehabisan stok.

Rumus untuk menghitung *EOQ* adalah sebagai berikut:

$$EOQ = \frac{\sqrt{2DS}}{H}$$

Keterangan:

- D adalah jumlah yang dibutuhkan dalam satu periode.
- S adalah biaya setiap pesanan.
- H adalah biaya penyimpanan per unit per tahun.

Pada metode ini juga dirancang dengan mempertimbangkan Stok Pengamanan (Safety Stock) dan titik kembali pemesanan (Reorder Point).

Rumusnya sebagai berikut :

$$SS = (MaxP - RataP) * L$$

Dimana :

SS = Safety Stock

MaxP = Penjualan maksimal dalam waktu tertentu

RataP = rata-rata penjualan dalam waktu tertentu

L = lead time (waktu tunggu antara pemesanan dengan barang datang)

Langkah selanjutnya setelah menghitung safety stok adalah mencari titik pemesanan kembali atau reorder point (ROP). Kegunaan ROP adalah menentukan kapan waktu yang tepat untuk melakukan pemesanan kembali produk. Adapun rumusnya sebagai berikut :

$$ROP = (d * L) + SS$$

Dimana :

ROP = jumlah stok yang tepat untuk pemesanan kembali

d = jumlah kebutuhan

L = lead time (waktu tunggu antara pemesanan dengan barang datang)

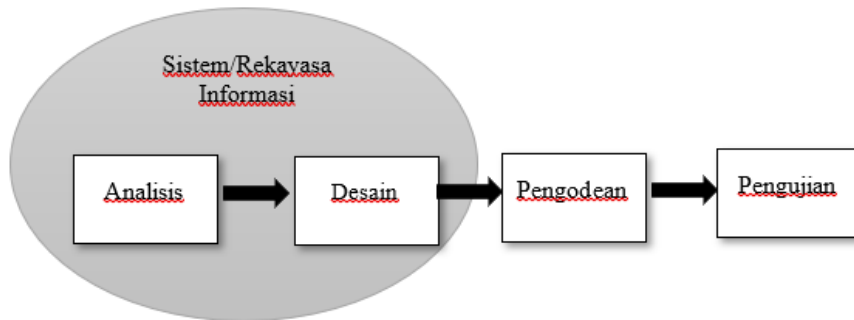
SS = Safety stock

Penulis melakukan pengambilan data barang dari TB Sumber Tani . Pengumpulan data dilakukan dengan observasi dan wawancara secara langsung dengan pimpinan TB Sumber Tani untuk mendapatkan data yang dibutuhkan.

2.6 Metode Waterfall SDLC (Software Development Life Cycle)

Pendekatan yang digunakan oleh penulis untuk mengembangkan perangkat lunak ini adalah model siklus hidup pengembangan perangkat lunak waterfall (SDLC). SDLC merupakan proses untuk mengembangkan atau mengubah sebuah sistem perangkat lunak dengan menggunakan model dan metodologi yang telah digunakan sebelumnya oleh para pengembang untuk mengembangkan sistem perangkat lunak. Proses *SDLC* dengan pendekatan *Waterfall* dapat dilihat pada gambar 2.1.

Gambar 2.1 Proses SDLC dengan pendekatan *Waterfall*



Sumber : (Rosa A. S & M. Shalahuddin, 2019)

Menurut (Rosa A. S & M. Shalahuddin, 2018) *waterfall model* memiliki siklus hidup perangkat lunak berurutan. Langkah Selanjutnya dalam proses SDLC dengan pendekatan Waterfall yaitu:

1. Analisis

Melakukan analisis kebutuhan perangkat lunaka, fungsi dan proses dari web yang dibuat, pengidentifikasian kendala dalam pembuatan web, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

2. Rancangan/Desain

Desain perangkat lunak adalah proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

3. Pembuatan Kode Program

Setelah tahap desain yang dilakukan, desain yang telah dibuat maka desain tersebut harus diubah kedalam sebuah program perangkat lunak, yang hasilnya menjadi sebuah aplikasi sistem pengelolaan inventory sesuai dengan desain yang telah dibuat pada tahap desain. Dalam tahapan ini perancangan sistem yang digunakan yaitu *Visual Studio Code* sebagai text editor, framework yang digunakan yaitu *Bootstrap*, *Codeigniter 3* dan *Jquery*. Sedangkan untuk bahasa pemrograman yang digunakan yaitu *Javascript*, *Cascading Style Sheet (CSS)*, *Hypertext Markup Language (HTML)*, *Hypertext Preprocessor (PHP)*. Penulis juga menggunakan software pendukung yaitu *XAMPP* sebagai penghubung server database.

4. Pengujian

Setelahnya, program harus menjalani pengujian yang difokuskan pada tiga aktivitas utama. Pertama, pengujian harus memeriksa logika internal dari perangkat lunak. Kedua, pengujian harus memastikan bahwa semua perintah yang ada telah diuji coba. Dan ketiga, pengujian harus menguji fungsi eksternal untuk memastikan bahwa dengan input tertentu, fungsi tersebut akan menghasilkan keluaran yang sesuai dengan yang diharapkan. Peneliti melakukan pengujian dengan menjalankan program yang telah dibuat, memeriksa dan memastikan bahwa fungsi dan logika program, serta keluaran yang dihasilkan sudah sesuai dengan kebutuhan pengguna dan system.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

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

2.7 Website

Pengertian Website adalah suatu pengenalan ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi global yang disebut Uniform Resource Identifier (URI) atau lebih dikenal dengan istilah yang lebih populer yaitu Uniform Resource Locator (URL)". Perkembangan World Wide Web (WWW) yang sangat pesat ditandai dengan munculnya berbagai macam website dengan halaman web yang interaktif. 16 Berdasarkan isinya website terdiri dari dua jenis yaitu:

1. Website Statis (Static Website) adalah web yang biasanya user tidak bisa mengubah content dari web tersebut secara langsung menggunakan browser. Interaksi yang terjadi hanya seputar pemrosesan link yang ada.
2. Website Dinamis (Dynamic Website) adalah web yang biasanya user dapat mengubah content dari halaman tertentu dengan menggunakan browser.

2.8 PHP

PHP adalah bahasa pemrograman yang dikembangkan pertama kali oleh Rasmus Lerdorf untuk membuat website pribadinya. Awalnya, *PHP* terdiri dari

skrip sederhana untuk menghasilkan konten dinamis pada halaman web, tetapi kemudian berkembang menjadi bahasa pemrograman yang kuat dengan dukungan untuk pemrograman berorientasi objek.

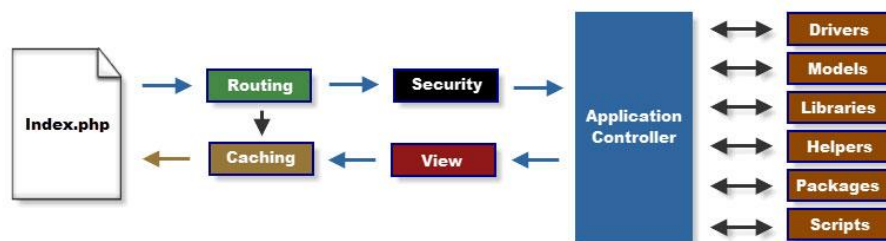
Salah satu keunggulan utama PHP adalah kemampuannya untuk dijalankan di server web, sehingga kode PHP dieksekusi di sisi server sebelum halaman web dikirim ke pengguna. Ini membuat pengguna tidak perlu tahu atau memiliki akses ke kode sumber PHP, karena yang mereka lihat adalah output HTML.

Selain itu, PHP dapat dengan mudah digabungkan dengan HTML, memungkinkan pengembang untuk menciptakan halaman web yang dinamis dan interaktif. Dengan fitur-fitur ini, PHP telah menjadi salah satu bahasa pemrograman paling populer dan umum digunakan dalam pengembangan website.(Kinaswara et al., 2019).

2.9 Freamwork CodeIgneter

CodeIgniter adalah sebuah framework aplikasi web sumber terbuka yang digunakan untuk membangun aplikasi PHP dinamis. Framework ini menyediakan perpustakaan lengkap untuk berbagai operasi umum dalam aplikasi web, seperti mengakses basis data dan memvalidasi formulir. *CodeIgniter* dikenal memiliki dokumentasi yang komprehensif dan mudah dipahami, serta kode sumber yang dilengkapi dengan komentar untuk menjelaskan fungsinya. Website yang dibangun dengan *CodeIgniter* cenderung bersih dan ramah mesin pencari. Dalam penggunaannya, *CodeIgniter* menerapkan pola MVC (*Model, View, Controller*) untuk memudahkan pengembangan aplikasi web tanpa harus membuatnya dari

awal. Konsep *MVC* memisahkan aplikasi menjadi tiga komponen utama: model, view, dan controller, di mana masing-masing memiliki tanggung jawab tersendiri. Secara spesifik, proses kerja aplikasi *PHP CodeIgniter* melibatkan penggunaan elemen-elemen *MVC*, sesuai dengan yang dijelaskan dalam gambar 2.3 . (Mery & Bernanda, 2022).



Gambar 2.2. Alur Kerja *Framework CodeIgniter* (Mashadov, 2020)

2.10 Bootstrap

Bootstrap adalah sebuah framework yang awalnya dikembangkan oleh pengembang Twitter pada pertengahan tahun 2010. Sebelum menjadi sebuah kerangka kerja open-source, *Bootstrap* dikenal sebagai Twitter Blueprint. Framework ini terus berkembang hingga saat ini dan telah menjadi salah satu framework front-end yang paling terkenal dan merupakan proyek open source yang populer di seluruh dunia. *Bootstrap* awalnya dikembangkan pada ajang Hackweek, sebuah acara pengembangan yang diadakan oleh Twitter. *Bootstrap* dijelaskan sebagai CSS yang sederhana namun dibangun dengan pre-processor yang menyediakan lebih banyak daya dan fleksibilitas daripada CSS standar. Framework ini pertama kali dirilis pada Jumat, 19 Agustus, 2011, dan telah mengalami lebih dari 20 rilis produk termasuk versi 2 dan versi 3 yang merupakan

yang terbesar. Fungsionalitas responsif terhadap seluruh framework dapat ditemukan sebagai *stylesheet* opsional dalam *Bootstrap*.(Anggraini, 2021)

2.11 MySQL

MySQL adalah sebuah perangkat lunak yang termasuk dalam kategori server basis data dan memiliki sifat open source. Open source mengindikasikan bahwa perangkat lunak ini dilengkapi dengan kode sumber (source code) yang dapat digunakan untuk membuat *MySQL*, selain dari bentuk executable-nya atau kode yang dapat dijalankan langsung dalam sistem operasi, dan dapat diunduh secara gratis melalui internet. Sebuah aspek menarik lainnya adalah *MySQL* juga memiliki sifat multiplatform. *MySQL* dapat dijalankan pada berbagai sistem operasi.(Ruhul Amin, 2017).

Seperti yang sudah disebutkan sebelumnya, *MySQL* masuk ke dalam jenis RDBMS (Relational database Management Sistem). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada *MySQL*. Contohnya di dalam *MySQL* sebuah database terdapat satu atau beberapa tabel. *MySQL* merupakan database engine atau server database yang mendukung bahasa database *SQL* sebagai bahasa interaktif dalam mengelola data. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS yang multithread, multi-user.(Rianto Sitanggang, 2022)

2.12 Unified Modeling Language (UML)

Dalam perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang menggunakan teknik pemrograman berorientasi objek, yaitu Unified Modeling

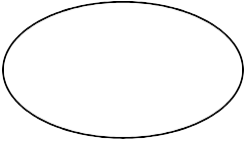
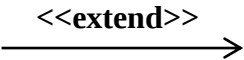
Language (UML). UML muncul karena adanya kebutuhan akan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan suatu perangkat lunak. UML merupakan bahasa visual yang digunakan untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks pendukung.(Anggraini, 2021)

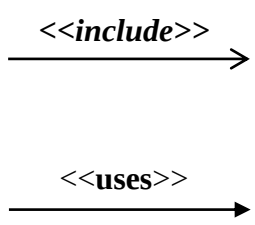
Berdasarkan (Rosa A. S & M. Shalahuddin, 2019) UML (Unified Modeling Language) adalah suatu bahasa visual yang digunakan untuk memodelkan dan berkomunikasi tentang suatu sistem melalui diagram dan teks pendukung. Dengan kata lain, UML dapat dipahami sebagai bahasa visual yang digunakan untuk menggambarkan definisi-definisi kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek dengan menggunakan teks pendukung.

2.12.1 Use Case Diagram

Menurut (Rosa A. S & M. Shalahuddin, 2019), use case atau diagram adalah representasi dari perilaku (behavior) sistem informasi yang akan dibuat. Diagram ini menggambarkan bagaimana aktor atau pengguna berinteraksi dengan sistem tersebut. Mereka juga menekankan bahwa dalam penggunaan use case, penting untuk menggunakan nama yang sederhana dan mudah dipahami sesuai dengan kebutuhan sistem yang dijelaskan. Berikut adalah simbol-simbol *Use Case Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Use Case Diagram*

Nama Simbol	Simbol	Keterangan
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
Aktor/ <i>actor</i>		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, biasanya dinyatakan menggunakan kata benda diawali frase nama aktor.
Asosiasi/ <i>association</i>		Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi. pada <i>use case</i> memiliki interaksi dengan aktor.
<i>Ekstensi/extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip inheritance pada

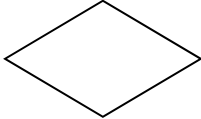
		pemrograman berorientasi objek, biasanya use case memiliki nama depan yang sama dengan use case yang ditambahkan, misal arah panah mengarah pada use case yang ditambahkan, biasanya use case yang menjadi extend nya merupakan jenis yang sama dengan use case yang menjadi induknya..
<i>Include/uses</i>		Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

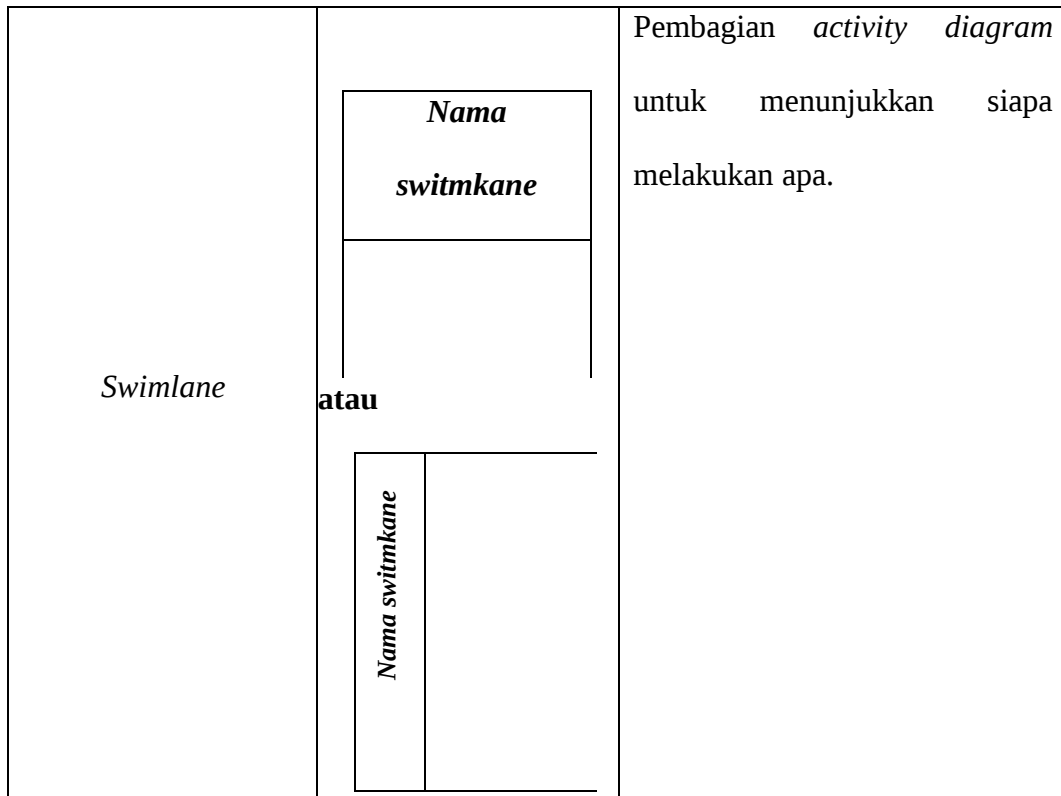
Sumber: (Rosa A. S & M. Shalahuddin, 2019)

2.12.2 Activity Diagram

(Rosa A. S & M. Shalahuddin, 2019) menjelaskan bahwa diagram aktivitas atau activity diagram digunakan untuk menggambarkan alur kerja atau aktivitas dari suatu sistem, proses bisnis, atau menu yang terdapat dalam perangkat lunak. Mereka juga menyebutkan bahwa simbol-simbol yang digunakan dalam activity diagram dapat ditemukan dalam Tabel 2.3 :

Tabel 2.3 Simbol *Activity Diagram*

Nama Simbol	Simbol	Keterangan
<i>Activities</i>		<i>Activites</i> , menggambarkan suatu proses atau kegiatan bisnis.
<i>Start Point</i>		Status awal aktivitas system sebuah diagram aktivitas memiliki sebuah status awal
Percabangan/ <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
<i>End Point</i>		<i>End point</i> , status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir.



Sumber: (Rosa A. S & M. Shalahuddin, 2019).

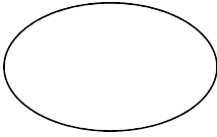
2.12.3 Class Diagram

Diagram kelas atau class diagram adalah representasi dari struktur sistem dalam hal definisi kelas-kelas yang diperlukan untuk membangun sistem. Kelas dalam diagram ini memiliki atribut dan metode, yang disebut juga sebagai operasi:

- Atribut adalah variabel yang dimiliki oleh kelas tersebut.
- Metode atau operasi adalah fungsi yang dimiliki oleh kelas tersebut.

(Rosa A. S & M. Shalahuddin, 2019) juga menyebutkan bahwa simbol-simbol yang digunakan dalam class diagram dapat ditemukan dalam Tabel 2.4.

Tabel 2.4 Simbol *Class Diagram*

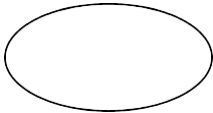
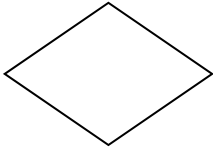
Nama Simbol	Simbol	Keterangan
Class	Nama_kelas +atribut +operasi()	Kelas pada struktur sistem.
Antarmuka/ <i>Interface</i>		Sama dengan konsep <i>interface</i> Dalam pemrograman <i>berorientasi objek</i> .
Asosiasi/ <i>association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multilicity</i> .
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> .
Generalisasi		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan / <i>depedency</i>		Kebergantungan antar kelas.
Agregasi/ <i>agregation</i>		Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: (Rosa A. S & M. Shalahuddin, 2019)

2.12.4 Entity Relationship Diagram (ERD)

(Sukamto dan Shalahuddin., 2018) menjelaskan bahwa ERD (Entity Relationship Diagram) digunakan sebagai alat untuk memodelkan basis data relasional. ERD digunakan khususnya untuk memodelkan struktur database relasional dengan tujuan menggambarkan hubungan antar data dalam database tersebut. Dengan demikian, Entity Relationship Diagram (ERD) dapat dianggap sebagai suatu model yang digunakan dalam perancangan database untuk mengilustrasikan hubungan antar entitas dalam basis data dapat ditemukan dalam Tabel 2.5.

Tabel 2.5 Simbol *Entity Relation Diagram*

Nama Simbol	Simbol	Keterangan
Entitas/entity		Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/ berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data yang modelkan.
Atribut		Proses atau fungsi atau prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemogramaan terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur didalam kode program.
Relasi		Hubungan yang terjadi antara salah satu entitas. Jenis hubungannya yaitu <i>One to one</i> , <i>One to many</i> , dan <i>Many to</i>

		<i>many</i>
Asosiasi/ <i>association</i>		Garis penghubung suatu entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya
Atribut multinilai <i>/multivalued</i>		<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih

Sumber:(Rosa A. S & M. Shalahuddin, 2019)

2.13 Testing (Pengujian)

Tahapan Pengujian adalah proses dimana perangkat lunak diperiksa untuk memastikan bahwa telah memenuhi persyaratan dan spesifikasi yang sesuai dengan tujuan yang diinginkan. Tahapan ini sering dikenal sebagai verifikasi dan validasi. Dalam tahap ini, sistem diuji menggunakan metode pengujian Black-Box Testing dan SUS (System Usability Scale) untuk menguji kegunaan sistem.

2.13.1 *Black-Box Testing*

Black-Box Testing adalah metode pengujian kualitas perangkat lunak yang terutama difokuskan pada fitur-fitur perangkat lunak (Setiyani, 2019). Tujuannya adalah untuk mengidentifikasi apakah terdapat fungsi-fungsi yang tidak sesuai atau kesalahan yang tidak diinginkan.

Black-Box Testing terdiri atas beberapa cara; *Equivalence Partitioning*, *Boundary Value Analysis*, *Comparison Testing*, *Sample Testing*, *Robustness Testing*, dan lain-lain. *Black-Box Testing* ditujukan untuk menemukan kesalahan dalam kategori seperti, fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan struktur data atau akses database eksternal, kesalahan kinerja, dan

inisialisasi dan kesalahan terminasi. Dalam pengujiannya, *Black-Box Testing* menggunakan 3 metode, yakni pengujian graph-based, partisi ekuivalensi, dan analisis nilai batas (Efendi, 2021).

Metode *Black-Box Testing* juga tergolong metode yang mudah karena hanya memerlukan batas bawah dan atas dari data yang diperlukan. Perhitungan data yang diuji harus melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi, dan kasus batas atas dan bawah yang memenuhi. Peneliti menggunakan metode ini untuk mengetahui apabila fungsionalitas masih perlu menerima masukan data yang tidak diharapkan akan menyebabkan penyimpanan data kurang valid (Nur Cholifah & Melati Sagita, 2018)

2.13.2 Sistem Usability Scale (SUS)

Sistem Usability Scale (SUS) adalah sebuah metode yang digunakan untuk menilai tingkat kegunaan atau usability dari suatu sistem informasi, khususnya dalam konteks perangkat lunak dan antarmuka pengguna. Pengujian SUS melibatkan pengisian kuesioner yang terdiri dari 10 pertanyaan oleh pengguna yang telah berinteraksi dengan sistem informasi yang sedang diuji (Harisnur & Gunawan, 2022). Tujuan utama dari SUS adalah untuk mengevaluasi seberapa mudah pengguna dapat menggunakan sistem informasi tersebut dan seberapa puas mereka dengan pengalaman penggunaannya. Skala penilaian berkisar dari 1 hingga 5, dengan nilai 1 menunjukkan tingkat ketidaksetujuan yang tinggi dan nilai 5 menunjukkan tingkat persetujuan yang tinggi dapat dilihat pada Tabel 2.6.

Tabel 2.6 Tabel Skala *Likert*

Tingkat Kepuasan	Skala
Sangat Setuju (SS)	5
Setuju (S)	4
Netral (N)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Kuesioner ini menggunakan skala Likert (skala lima poin) dengan kriteria "sangat tidak setuju", "tidak setuju", "netral", "setuju", dan "sangat setuju". Sepuluh pernyataan dibagi menjadi kategori positif dan negatif. Pernyataan positif memiliki nomor ganjil (nomor 1,3,5,7, dan 9) dan yang negatif memiliki nomor genap (nomor 2,4,6,8, dan 10). Perhitungan skor untuk setiap pernyataan positif dilakukan dengan mengurangi 1 skor dari setiap skor yang diberikan oleh responden. Sementara perhitungan skor untuk setiap pernyataan negatif dilakukan dengan mengurangi skor yang diberikan oleh responden dari 5 (Setemen et al., 2019). Oleh karena itu, setiap item pernyataan memiliki skor minimal 0 dan skor maksimal 2,5, sehingga skor akhir setiap responden berkisar dari 0 hingga 100. Aturan penilaian menggunakan System Usability Scale (SUS) memiliki beberapa peraturan saat menjumlahkan skor. Berikut adalah peraturan-peraturan yang digunakan dalam perhitungan kuesioner:

1. Skor setiap pertanyaan bernomor ganjil akan dikurangi 1 dari skor yang diberikan oleh pengguna.
2. Skor setiap pertanyaan bernomor genap akan dikurangi dari nilai 5 sesuai dengan skor yang diberikan oleh pengguna.

3. Skor SUS akan diperoleh dengan menjumlahkan skor dari setiap pertanyaan, lalu hasilnya akan dikalikan dengan 2,5.
4. Aturan penilaian ini berlaku untuk satu responden. Untuk perhitungan lebih lanjut, skor SUS dari setiap responden akan dijumlahkan, kemudian hasilnya akan dibagi dengan jumlah responden untuk mendapatkan skor rata-ratanya. Rumus untuk menghitung skor SUS adalah:

Skor *system usability scale* = ((Q1-1) + (5-Q2) + (Q3-1) + (5-Q4) + (Q5-1) + (5-Q6) + (Q7-1) + (5-Q8) + (Q9-1) + (5-Q10)) * 2,5).

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

$\bar{x}$ = Skor rata-rata, $\sum x$ = Jumlah Skor SUS, n = Jumlah Responden

Adapun syarat-syarat yang perlu di perhatikan sebelum menggunakan.

rumus *system usability scale* sebagai berikut :

1. Pertanyaan bernomor ganjil, setiap nilai yang telah didapatkan dari skor pengguna selanjutnya akan dikurangi 1.
2. Pertanyaan bernomor genap, setiap skor akhir yang didapat dari nilai 5 akan dikurangi nilai yang didapatkan pengguna.
3. Nilai akhir yang didapatkan dari penjumlahan setiap pernyataan kemudiandikali 2,5.