

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Berikut ini uraian penjelasan mengenai teori dasar yang diterapkan dalam penelitian ini. Teori dasar ini bertujuan untuk memperkuat isi teori yang telah ada, sehingga hasil penelitian dapat lebih optimal. Adapun isi dari teori dasar dapat diuraikan sebagai berikut:

2.1.1 Rancang Bangun

Rancang bangun merupakan proses menggambarkan, merencanakan, serta Menyusun sketsa atau pengaturan dari berbagai elemen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi. Dengan demikian, rancang bangun dapat diartikan sebagai kegiatan yang mengubah hasil analisis menjadi bentuk perangkat lunak, kemudian membangun sistem baru atau mengembangkan sistem yang sudah ada (Wulandari & Fadly, 2021). Berdasarkan pendapat (Adi Setiawan et al., 2024), rancang bangun merupakan suatu aktivitas untuk menjelaskan hasil analisis ke dalam perangkat lunak, lalu menghasilkan sebuah sistem baru atau memperbarui sistem yang sudah ada. Menurut pendapat (Ratu et al., 2022), rancang bangun merupakan prosedur yang dibuat menjadi beberapa rangkaian yang kemudian hasil analisis tersebut diterjemahkan ke dalam bahasa pemrograman dan menghasilkan sistem baru atau sistem yang sudah ada diperbaiki.

2.1.2 Sistem Informasi

Sistem informasi adalah gabungan dari beberapa komponen sistem yang saling terhubung dan bekerja sama untuk memproses pengumpulan, penyimpanan, dan analisis data menjadi informasi yang dibutuhkan bagi pengguna. Sistem informasi menyediakan informasi yang akurat, relevan, *real-time* bagi pengguna maupun organisasi, sehingga mencapai tujuan bisnis yang ditetapkan dan kemudahan dalam pengambilan keputusan yang baik, pengelolaan operasional yang efisien (Tucunan & Heryana, 2023). Pengertian lainnya sistem informasi adalah serangkaian yang saling terhubung antara individu, data, proses, maupun antarmuka untuk mencapai suatu bisnis agar kegiatan operasional meningkat (Nasution et al., 2020).

Dapat disimpulkan bahwa sistem informasi merupakan gabungan antara manusia, teknologi, protokol, dan informasi yang saling mendukung untuk mencapai tujuan informasi atau bisnis.

2.1.3 Penjualan

Penjualan adalah kegiatan transaksi jual-beli berupa barang atau jasa antara dua belah pihak atau lebih, dengan tujuan untuk mendapatkan keuntungan. Menurut (Mustopa et al., 2021) penjualan merupakan pembelian suatu barang atau jasa dari suatu pihak dengan pihak lain dengan mendapatkan ganti berupa uang dari pihak tersebut dan penjualan merupakan sumber pendapatan bagi perusahaan, semakin besar penjualan, maka semakin besar pendapatan perusahaan yang diperoleh.

2.1.4 Sistem Informasi Penjualan

(Nurjamil & Sembiring, 2021) menyatakan bahwa sistem untuk mengelola informasi berupa penjualan produk atau layanan di sebuah organisasi atau perusahaan. Dan sistem ini merupakan bagian dari sub sistem informasi bisnis dan sub sistem pemasaran, sumber daya manusia, keuangan akuntansi maupun manufaktur produksi. Sistem informasi penjualan adalah untuk menawarkan produk atau jasa kepada calon pelanggan guna mencapai tujuan tertentu, serta memungkinkan promosi yang lebih cepat dan mudah diterima oleh pelanggan.

Sistem informasi penjualan merupakan kumpulan prosedur yang dirancang secara khusus untuk mengelola, mencatat, menghitung, menghasilkan dokumen, Serta menyajikan informasi yang berkaitan dengan aktivitas penjualan (Fatawa Imam Al Muftin & Fendi Hidayat, 2024).

Fungsi dari sistem informasi penjualan mencakup pengelolaan seluruh proses penjualan, mulai dari pemesanan hingga penyelesaian transaksi, dengan tujuan mendukung berbagai kegiatan penjualan serta menyediakan data yang dibutuhkan oleh berbagai divisi dalam perusahaan (Fatawa Imam Al Muftin & Fendi Hidayat, 2024).

Implementasi sistem informasi penjualan berkontribusi pada peningkatan efisiensi operasional, mempercepat transaksi penjualan, meningkatkan kepuasan pelanggan, serta memberikan wawasan yang lebih mendalam bagi perusahaan dalam mengambil keputusan strategis.

2.1.5 Database

Database adalah kumpulan kumpulan data yang telah disimpan secara cermat menggunakan suatu sistem data sebelum digunakan. Basis data ini menggunakan file, film, gambar, dan frasa (Pulungan et al., 2023). Dalam penggunaan sehari-hari database merupakan kumpulan data atau informasi sistematis yang disimpan sedemikian rupa, sehingga mudah dipelihara, diakses, dan diperbarui (Noviyana & Nasution, 2024). *Structure Query Language (SQL)* merupakan bahasa standar digunakan untuk memproses, mengakses, dan mengelola informasi dalam *database* relasional. *Relational Database Management System (RDBMS)* menjadi pilihan bagi *programmer* aplikasi *web* karena efisien dalam pengelolaan *database*, mengontrol integrasi data dan mampu menganalisis data yang lebih kompleks (Sinlae et al., 2024).

2.1.6 Website

Website berasal dari kata *World Wide Web* yang merupakan layanan yang dapat diakses menggunakan perangkat komputer yang terkoneksi internet. Aplikasi *Website* berjalan di atas *platform* atau sistem peramban (*browser*) atau dapat diartikan sebagai sekumpulan halaman informasi yang tersedia secara online dan dapat diakses secara lokal maupun jarak jauh (Surentu et al., 2020). *Website* menyediakan berbagai informasi seperti teks, gambar, tabel, grafik, dan video yang bisa diakses secara online yang terhubung dengan internet melalui komputer, *smartphone*, dan laptop. Menurut pendapat (Mohammad Ahmadar & Perwito, 2021) *Website* merupakan halaman-halaman yang saling terhubung satu sama lain

yang memuat berupa informasi atau teks, gambar bergerak atau diam, suara, animasi, atau kombinasi dari semuanya yang bersifat dinamis atau statis.

2.1.7 MySQL

MySQL merupakan *Relational Database Management System (RDMS)* yang menggunakan bahasa *Structure Query Language (SQL)* yang bersifat *Open-Source* yang memiliki dua lisensi, yaitu *Free Software* (perangkat lunak gratis) dan *Shareware* (perangkat lunak berpelik dengan penggunaan terbatas). *MySQL* dapat digunakan untuk pribadi maupun komersil tanpa membeli lisensi karna sudah memiliki lisensi *GNU General Public License (GPL)* (Fauzan & Nurhidayah, 2020).

MySQL merupakan salah satu konsep *database* sejak lama dari turunan *SQL* (*Structure Query Language*). Konsep manipulasi *SQL* merujuk pada basis data, secara khusus merujuk pada proses pemilihan, memasukkan, mengubah, dan menghapus data yang mudah dilakukan dan otomatis (Wahyuni, 2022).

2.1.8 JavaScript

JavaScript adalah bahasa *scripting* yang digunakan di sisi *client*. *JavaScript* sering digunakan untuk membuat animasi dan tampilan *web* menjadi interaktif. *Library JavaScript* banyak dipakai oleh *programmer* untuk membuat tampilan web menjadi lebih interaktif. Menjalankan kode *JavaScript* dibutuhkan *browser* yang mendukung (Sinlae et al., 2024). *JavaScript* dapat dijalankan hampir semua *web browser* seperti *Google*, *Mozilla FireFox*, dan lain-lain. Karena *JavaScript* bahasa pemrograman yang dinamis, dan sangat populer.

Menurut pendapat (Sulistyorini et al., 2022), *JavaScript* adalah bahasa pemrograman berbasis skrip yang berjalan pada dokumen *HTML*. Bahasa ini digunakan untuk memberikan kemampuan tambahan pada *HTML* dengan memungkinkan eksekusi perintah-perintah di *client-side*. Dalam *JavaScript*, penamaan variabel atau fungsi bersifat *case-sensitive*, artinya variabel dengan nama `TEST` dianggap berbeda dengan variabel `test`. Setiap intruksi dalam *JavaScript* umumnya diakhiri dengan karakter titik koma (;), meskipun dalam beberapa kasus, *JavaScript* tetap dapat berfungsi tanpa titik koma karena adanya fitur *Automatic Semicolon Insertion (ASI)*. Berbeda dengan bahasa pemrograman yang berjalan di sisi server, *JavaScript* membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil. Contohnya, variabel `Nama` dan `nama` akan dianggap sebagai dua variabel yang berbeda.

2.1.9 *ReactJS*

ReactJS merupakan sebuah perpustakaan *JavaScript* yang *Open Source* digunakan untuk membangun *User Interface (UI)* dan *Single Page Application (SPA)* atau aplikasi berbasis satu halaman pada *web*. Pada awalnya, *ReactJS* adalah perpustakaan *JavaScript* yang dikembangkan oleh *Jordan Walke* di tahun 2013 yang digunakan untuk membuat tampilan *front-end* pada aplikasi *web*. *ReactJS* bisa membuat komponen menjadi *reusable* atau komponen *front-end* yang bisa digunakan secara berulang-ulang tanpa harus membuatnya dari awal. *ReactJS* dibuat untuk mendesain tampilan *web* dari yang sederhana hingga kompleks. Banyak perusahaan besar menggunakan *ReactJS* seperti, *Facebook*, *Netflix*,

Instagram, Airbnb, Dropbox, Ebay, dan bahkan ratusan *programmer* menggunakan *ReactJS* untuk pembuatan aplikasi *web* (Sulistyorini et al., 2022).

ReactJS merupakan sebuah *library* yang di kembangkan oleh *Facebook* yang digunakan untuk membangun sebuah UI pada *front-end* agar menjadi interaktif dan *reusable*. *ReactJS* sendiri mempunyai beberapa keunggulan seperti *scalability*, kecepatan, dan *simplicity* (Panjaitan & Pakpahan, 2021) .

2.1.10 PHP

PHP adalah bahasa pemrograman yang digunakan untuk pengembangan *web* yang bekerja di sisi server yang disisipkan di *HTML* (Naofal et al., 2022). *PHP* merupakan bahasa skrip yang disisipkan di skrip *HTML* untuk membuat aplikasi menjadi dinamis dari segi kemampuan pemrosesan maupun manipulasi data. Semua sintaks yang dituliskan akan dieksekusi sepenuhnya di sisi server, sedangkan yang diteruskan ke browser hanyalah hasil pemrosesannya. Bahasa ini berbentuk skrip yang diletakkan di *server* dan dijalankan di sana, kemudian hasil akhirnya dikirimkan ke client atau pengguna melalui *browser* (Hermiati et al., 2021). *PHP* singkatan dari *Hypertext Preprocessor* merupakan bahasa pemrograman *server-side* yang digunakan untuk membuat situs *web* dapat berinteraksi dengan *database* dan menyajikan konten yang bersifat dinamis. *PHP* merupakan bahasa skrip yang terintegrasi dengan *HTML* dan dijalankan di *server-side*, sehingga seluruh sintaks yang dituliskan akan diproses sepenuhnya oleh *server* (Sinlae et al., 2024).

2.1.11 *Laravel*

Laravel merupakan *web development framework* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan menekan biaya pengembangan dan pemeliharaan. Selain itu, *Laravel* juga membantu meningkatkan produktivitas kerja melalui sintaks yang rapi dan efisien, sehingga dapat mempercepat proses implementasi (Naofal et al., 2022). *Laravel* merupakan *framework PHP* yang bersifat *open source* (terbuka) dan berlisensi *MIT* yang disimpan di repository *github* tempat berbagi kode (Gibran et al., 2024)

2.1.12 *Payment Gateway*

Payment Gateway adalah fasilitas yang menerima pembayaran untuk transaksi *digital* yang berjalan secara otomatis dengan sepenuhnya dan mempermudah proses transaksi, dan semua transaksi keuangan diproses dengan aman dan *real-time*. Secara umum sistem pembayaran berbasis elektronik sudah dikenal oleh masyarakat sebagai sistem pembayaran dalam beberapa tahun terakhir (Maysaroh & Diansyah, 2022). *Payment Gateway* merupakan sistem yang menghubungkan rekening bank dengan proses pembayaran yang sesuai, dengan cara mengirimkan data transaksi secara *digital* melalui layanan pembayaran berbasis web dan API, atau via terminal pembayaran (Gibran et al., 2024)

2.1.13 *Midtrans*

Midtrans merupakan sistem pembayaran dipakai antara pembeli dan penjual saat transaksi. Fitur yang dimiliki *Midtrans* sudah terintegrasi dengan pembayaran secara online yang menggunakan kartu kredit, kartu debit, serta penarikan dan pengiriman uang tunai (Wardana et al., 2021). *Midtrans* sebagai *Payment*

Gatewaynya untuk menunjang proses jual-beli dan keamanan saat transaksi. Aplikasi penjualan berbasis *web* diperlukan agar pembeli dapat langsung melakukan akses web tanpa perlu mengunduh aplikasi (Maharani et al., 2022).

Penggunaan layanan *Midtrans* wajib registrasi terlebih dahulu melalui *website* resmi *Midtrans* dengan melengkapi data yang dibutuhkan. setelah melakukan pendaftaran, pengguna akan menerima *ID Merchant*, *Server Key*, dan *Client Key* supaya dapat menggunakan layanan *Midtrans* termasuk dokumentasi proses integrasi pembayaran pada penggunaan *Midtrans* (Putri et al., 2022).

2.1.14 Visual Studio Code

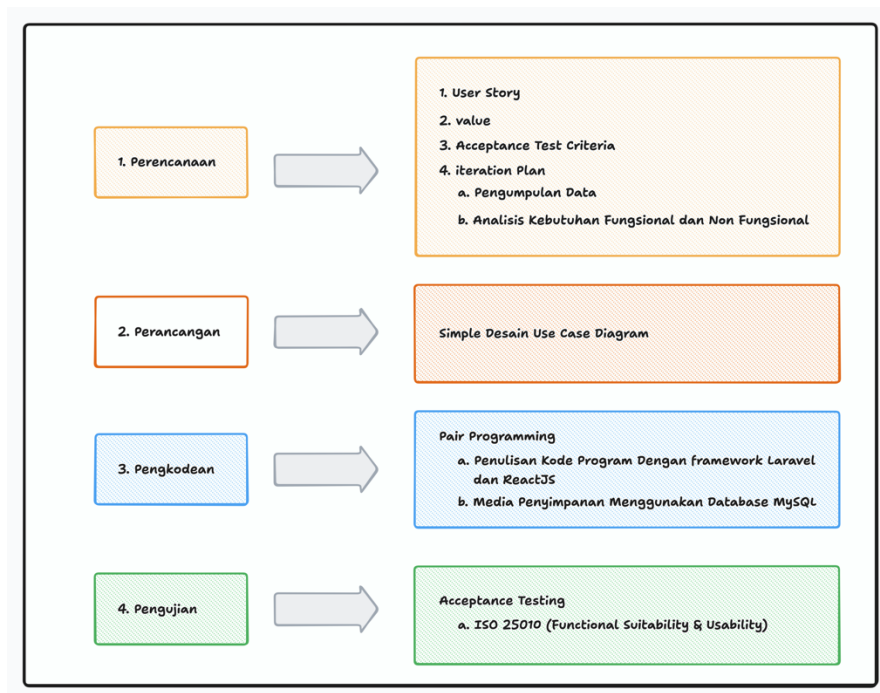
Visual Studio Code merupakan *text editor* yang bersifat *open source*, ringan, dan handal yang dibuat oleh *Microsoft* yang bisa dijalankan banyak sistem operasi atau multiplatform, artinya tersedia untuk versi *Linux*, *Mac*, dan *windows*. Tersedia dengan dukungan bawaan untuk *JavaScript*. *TypeScript*. Dan *NodeJS* dan memiliki banyak ekstensi seperti: *C++*, *C#*, *Java*, *Python*, *Go*, dan *PHP* dan runtime seperti: *.NET*, dan *Unity*, yang memudahkan bagi *Programmer* (Vaati, 2020).

2.1.15 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) adalah suatu metodologi yang digunakan untuk proses pembuatan dan pengembangan sistem. Sistem tersebut adalah sistem komputer atau sistem informasi. Berbagai jenis model pengembangan dari metode *Software Development Life Cycle (SDLC)* seperti *prototype*, *Rapid Application Development (RAD)*, *agile*, *fountain*, *v-model*, *waterfall*, *scrum*, *iterative*, *spiral*, *big bang*, dan *extreme programming* (Ridwan et al., 2021).

Penelitian ini menggunakan model *Extreme Programming (XP)*. *Extreme Programming (XP)* merupakan salah satu bagian dari beberapa proses *Agile* yang familiar. Model ini terbukti sukses diterapkan oleh banyak perusahaan dari berbagai skala dan industri di seluruh dunia. Penerapan model *Extreme Programming (XP)* lebih menekankan pada kepuasan pengguna. Pada proses pengembangannya cenderung fleksibel membuat *Extreme Programming (XP)* menjadi unik. Pada saat terjadinya perubahan proses bisnis yang dinamis yang dibutuhkan pengguna mengikuti perkembangan, pengguna tidak harus menunggu di waktu tertentu. Pada *Extreme Programming (XP)* diberdayakannya pengembang perangkat lunak untuk menghadapi perubahan permintaan pengguna secara lebih optimal, bahkan pada siklus pengembangan akhir (Wiratama & Santoso, 2023).

Extreme Programming dengan skala tim kecil sudah cukup, rancangan yang mudah dipahami, tetapi lebih menekankan sisi *user story* dan pengkodean yang dilakukan dengan bertahap, sehingga proses pengembangan sistem dapat lebih singkat (Ayunandita & Riskiono, 2021). Pengembangan sistem dengan menggunakan metode ini termasuk cepat, efisien, beresiko rendah, fleksibel dan terprediksi, akan tetapi tidak mempunyai dokumentasi formal karena proses pengumpulan kebutuhan sistem yang akan dibuat lebih difokuskan. Ketika melakukan observasi pada tahapan perencanaan (Pasha et al., 2023)



Gambar 2.1 Metode *Extreme Programming*

Perancangan sistem informasi penjualan berbasis web, *Extreme Programming (XP)* menjadi metode yang dipilih oleh peneliti. Adapun beberapa tahapan sebagai berikut:

1. Perencanaan (*Planning*)

Tahapan ini dimulai dengan mengidentifikasi permasalahan serta mengumpulkan berbagai kebutuhan sistem yang akan dikembangkan. Langkah ini bertujuan untuk proses bisnis dalam sistem lebih mudah dipahami serta memberikan gambaran yang jelas mengenai fitur utama, fungsionalitas, dan output yang diharapkan. Dalam pengembangan sistem informasi penjualan di Toko Budi Luhur, tahap perencanaan diawali dengan mengidentifikasi potensi serta permasalahan yang ada, kemudian menganalisis kebutuhan sistem yang akan dibangun.

2. Perancangan (*design*)

Tahapan ini dilakukan perancangan dengan memodelkan perangkat lunak yang dimulai dari pemodelan sistem, pemodelan arsitektur, alur data teknologi yang digunakan, dan pemodelan basis data.

3. Pengkodean (*Code*)

Tahap ini merupakan tahap implementasi yang akan dibahas lebih lanjut bab berikutnya. Dalam proses pengembangan sistem informasi penjualan Toko Budi Luhur, seluruh rancangan proses bisnis dan tampilan yang telah disusun akan diubah menjadi kode program. Bahasa pemrograman yang digunakan dalam implementasi ini adalah *JavaScript*, dengan *framework ReactJS, Laravel*, dan *MySQL* sebagai sistem manajemen basis data.

4. Pengujian (*Testing*)

Tahapan ini bertujuan untuk menguji sistem yang telah dikembangkan setelah proses pengkodean selesai, pengujian ini dilakukan untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan kebutuhan pengguna. Terdapat dua metode pengujian yang dapat digunakan, yaitu *White Box Testing*, *Black Box Testing*, serta pendekatan *Technology Acceptance Model (TAM)*.

2.1.16 *Unified Modeling Language (UML)*

Unified Model Language (UML) merupakan metode pemodelan yang digunakan untuk mendukung proses perancangan sistem untuk mengurangi risiko

kegagalan dalam pengembangan aplikasi. Penerapan *Unified Modeling Language (UML)* berfungsi untuk menggambarkan actor yang menggunakan aplikasi, aktivitas dari masing-masing actor, serta proses dan mekanisme sistem dalam sistem dalam penjadwalan antrian cuci mobil. Dengan menerapkan *Unified Model Language (UML)*, proses perancangan menjadi lebih mudah dan mendukung tahap pengkodean dalam pengembangan aplikasi berbasis web untuk sistem antrian cuci mobil (Saifudin & Zakaria, 2023).

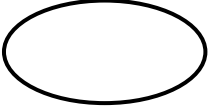
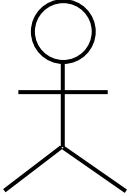
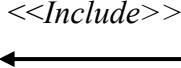
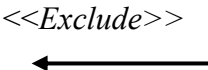
Unified Model Language (UML) adalah sebuah standar visual yang digunakan untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak. *Unified Model Language (UML)* menyediakan berbagai jenis diagram untuk menggambarkan struktur, perilaku, serta interaksi dalam suatu sistem, sehingga mempermudah pemahaman dan komunikasi antara pengembangan, analisis, dan pemangku kepentingan lainnya.

Unified Model Language (UML) digunakan dalam berbagai tahap pengembangan perangkat lunak, mulai dari analisis kebutuhan hingga implementasi. Beberapa diagram *Unified Model Language (UML)* yang umum digunakan meliputi:

1. *Usecase Diagram*

Pemodelan *Usecase Diagram* digunakan untuk menggambarkan interaksi antara actor (pengguna sistem) dengan sistem itu sendiri. *Usecase Diagram* menggambarkan hubungan antara sistem, pengguna, serta beberapa modul yang terdapat dalam aplikasi sistem antrian online.

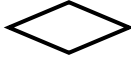
Tabel 2. 1 Use Case Diagram

Simbol	Nama	Keterangan
	<i>Use Case</i>	Mewakili aksi atau scenario tertentu yang dilakukan oleh aktor dalam berinteraksi dengan sistem. Ini mencakup serangkaian langkah-langkah yang menggambarkan alur kerja dan interaksi yang diharapkan dari pengguna sistem.
	<i>Actor</i>	Merupakan entitas, baik individu, proses, maupun sistem lain yang berinteraksi dengan sistem informasi yang akan dikembangkan, tetapi berada di luar sistem itu sendiri. Meskipun secara <i>symbol</i> dari aktor dipresentasikan dengan gambar manusia, penamaannya umumnya diawali dengan kata benda untuk mendeskripsikan perannya dalam sistem.
	<i>Assosiation</i>	Menghubungkan antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
	<i>Include</i>	Menunjukkan suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	<i>Exclude</i>	Sebagai penambahan komunikasi antara <i>use case</i> dengan <i>use case</i> yang lain. <i>Use case</i> yang berdiri sendiri tanpa <i>use case</i> tambahan.
	<i>Generalization</i>	Adalah koneksi umum untuk dua scenario penggunaan tertentu. Sebuah <i>use case</i> memiliki fungsi yang lebih umum dibandingkan <i>use case</i> lainnya. Panah menunjuk ke arah kasus penggunaan yang populer.

Dengan diagram ini, pengembang dapat memahami secara visual bagaimana interaksi antara pengguna dan sistem terjadi, sekaligus memperoleh gambaran

mengenai alur proses antrian dari awal hingga akhir (Friadi et al., 2023). *Use Case* Diagram merupakan sebuah cara yang sering digunakan dalam pengembangan *software* atau sistem informasi guna untuk memenuhi kebutuhan pada fungsionalitas sistem (Prasetyo & Sutopo, 2020). *Use Case* Diagram merupakan suatu diagram digunakan untuk menggambarkan interaksi antara satu aktor atau lebih aktor pada sistem yang akan dibuat (Panjaitan & Pakpahan, 2021).

Tabel 2. 2 Activity Diagram

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana aktivitas yang dijalankan proses, biasa berawal dengan kata kerja.
	Status Awal	Menunjukkan sebuah diagram aktivitas dari suatu proses yang mempunyai sebuah status awal.
	<i>Decision</i> / percabangan	percabangan yang dipakai ketika lebih dari satu aktivitas.
	Join / penggabungan	Ketika terdapat aktivitas lebih dari satu maka terjadi penggabungan.
	Status akhir	Mewakili sebuah diagram aktivitas sebagai status akhir dari suatu proses yang dilakukan sistem.
	Swimlane	Elemen visual yang digunakan sebagai pemisah dari organisasi bisnis dan memiliki tanggung jawab dari setiap aktivitas.

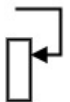
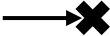
2. Activity Diagram

Activity diagram adalah pemodelan pada perangkat lunak untuk membuat proses-proses yang sedang terjadi, membuat alur proses pada perangkat lunak dengan menggunakan diagram secara vertikal. Dapat disimpulkan bahwa pembuatan *activity* diagram hanya digunakan untuk

menggambarkan aktivitas sistem atau alur kerja, dan diagram aktivitas tidak menjelaskan kelakuan aktor (Fatman et al., 2023).

3. *Sequence Diagram*

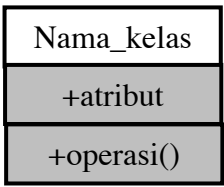
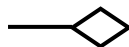
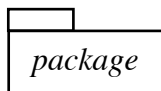
Tabel 2. 3 *Sequence Diagram*

Gambar	Nama	keterangan
	<i>Actor</i>	Mewakili sebuah entitas eksternal berkomunikasi dengan sistem. Aktor tidak hanya manusia, bisa berupa sistem, atau entitas eksternal lainnya.
	<i>Object</i>	Sebuah entitas atau objek dalam sistem melakukan suatu proses atau aktivitas tertentu.
	<i>Lifeline</i>	Berupa garis bujur yang mewakili pelaku atau objek dalam sebuah sistem. <i>Lifeline</i> menjelaskan usia pelaku atau objek dan titik waktu disetiap elemen-elemen berinteraksi dengan objek lainnya.
	<i>activation</i>	sebuah diagram yang mewakili tugas atau aktivitas tertentu diimplementasikan oleh pengguna sistem atau objek lain.
	<i>message</i>	Menunjukkan deklarasi sebuah objek atau pengirim sedang menunggu respon dari penerima sebelum melanjutkan.
	<i>Reply Message</i>	Diagram ini menunjukkan bahwa penerima telah merespon pesan dari objek lain dan menyelesaikan tugasnya, mengembalikan respon ke pengirim.
	<i>Create Message</i>	Sebagai objek baru yang telah dibuat.
	<i>Self-Message</i>	Adalah objek atau pengguna sedang melakukan aktivitas tertentu pada dirinya sendiri atau <i>recursive</i> .
	<i>Delete Message</i>	Menunjukkan sebuah objek yang tersedia sedang dihapus.

Sequence Diagram merupakan diagram yng digunakan untuk menggambarkan interaksi antara objek di dalam maupun sekitar sistem, Diagram ini berfungsi untuk memodelkan perilaku suatu scenario dalam sistem, yang kemudian dapat diterapkan sebagai acuan dalam implementasi menggunakan bahasa pemrograman (Khairul et al., 2023).

4. *Class Diagram*

Tabel 2. 4 *Class Diagram*

Gambar	Nama	Keterangan
	<i>Class</i>	Sebuah kelas dalam suatu struktur organisasi atau sistem yang memiliki atribut serta fungsi yang mendukung operasional atau proses dalam kelas tersebut.
	<i>Interface</i>	Mirip seperti konsep <i>interface</i> pada pemrograman berorientasi objek.
	<i>Association</i>	Hubungan antar kelas yang umum, asosiasi juga dilengkapi dengan <i>multiplicity</i> .
	<i>Direct Association</i>	Menunjukkan relasi antar kelas yang mempunyai arti kelas itu sendiri yang digunakan oleh kelas lain, dan <i>multiplicity</i> biasanya digunakan untuk asosiasi.
	<i>generalisasi</i>	Hubungan antar kelas yang memiliki arti generalisasi-spesialisasi.
	<i>Dependency</i>	Menunjukkan kebergantungan hubungan antar kelas.
	<i>Aggregation</i>	Menunjukkan hubungan antar kelas yang memiliki
	<i>Package</i>	Kelas yang tergabung dalam satu <i>package</i> dapat saling mengakses atribut dan metode.

Tujuan dari *Class Diagram* adalah menggambarkan struktur statis sistem. *Class Diagram* adalah elemen dasar dan inti utama dalam hampir semua metodologi berbasis pemrograman berorientasi objek (Friadi et al., 2023).

2.2 Penelitian Terdahulu

Beberapa penelitian sebelumnya digunakan sebagai referensi dalam penelitian ini. Pencatuman penelitian terdahulu bertujuan untuk memastikan penelitian yang dilakukan menghasilkan temuan yang valid dan optimal. Berikut adalah beberapa penelitian yang dijadikan sebagai.

1. Penelitian yang dilakukan oleh Juventauricula et al., (2024) yang berjudul “Pengembangan Sistem Informasi Point of Sale (POS) berbasis *Web* menggunakan Pendekatan Metode *Waterfall* (Studi Kasus: Restoran Altari)” dalam penelitiannya menggunakan *waterfall*. Sistem informasi POS yang dibangun menjadi solusi dalam meningkatkan efisiensi, efektivitas, dan akurasi pencatatan transaksi dan pengumpulan data bisnis restoran.
2. Haqqi & Vivianti, (2022), penelitian yang berjudul “Pengembangan Sistem Informasi Penjualan dan Stok Barang Toko Penjualan Plafon Berbasis *Web*” menghasilkan sistem informasi penjualan dan stok barang toko penjualan plafon berbasis *web* ini dapat melakukan fungsi seperti perhitungan mesin kasir, pencatatan data dan stok barang, pencatatan pengeluaran, pembelian dan penjualan toko, terdapat fitur laporan pendapatan yang digunakan pemilik toko dalam merekap pendapatan

bulanan. Metode *waterfall* yang dipilih dari penelitian ini.

3. Gultom & Maryam, (2020) penelitian tentang “Sistem Informasi Penjualan Material Bangunan Pada Toko Bangunan Berkah” menghasilkan sebuah sistem informasi penjualan yang membantu transaksi penjualan, mengelola stok barang masuk menjadi lebih efektif dan efisien, dapat melakukan cetak laporan laba di Toko Bangunan Berkah, yang sebelumnya dilakukan dengan cara manual. Metode yang digunakan penelitian ini adalah metode *waterfall*.
4. Ardiansah et al., (2023) penelitian yang berjudul “Penerapan *Extreme Programming* Dalam Sistem Informasi Akademik SDN Kuala Teladas” mengimplementasikan metode *Extreme Programming* untuk rancang bangun sistem informasi akademik di SD Negeri Kuala Teladas terbukti efektif karena tahapan yang sederhana, fleksibel, dan memudahkan komunikasi dengan pengguna. Hasil pengujiannya menggunakan *Technology Acceptance Model (TAM)* dengan tingkat kelayakan 92,72% yang dikategorikan sebagai sangat layak.
5. Effendy et al., (2022) penelitian tentang “*Design And Build A Web-Based Asset Management Information System at Pt Thamrin Telekomunikasi Network*” menghasilkan sistem yang dapat mengelola data asset TI secara terpusat, pengajuan dan perbaikan aset dapat dilakukan secara digital, dan menyediakan fitur pelaporan untuk pengambilan keputusan. Metode *Extreme Programming* yang dipilih pada penelitian ini.
6. Al-Kindi et al., (2021) penelitian yang berjudul “Pengembangan Sistem Informasi Penjualan Terintegrasi berbasis Website menggunakan

Payment Gateway (Studi Kasus: Usaha Kuliner Pentol Buto)” menghasilkan sistem informasi penjualan dengan fitur otomatis pembayaran melalui *payment gateway*, sehingga transaksi tidak perlu mengunggah bukti pembayaran pada admin via *whatsapp*. Penelitian ini menggunakan *waterfall*.

7. Thifalia & Fahmi, (2024) penelitian tentang “Penerapan Metodologi Rapid Application Development Pada Sistem Informasi Penjualan Produk Usaha Mikro Kecil Menengah” memanfaatkan eksistensi teknologi sehingga menghasilkan sistem informasi penjualan dengan berbagai layanan jual beli secara online dengan pembayaran otomatis yang terintegrasi dengan *midtrans* dan layanan pengelolaan informasi promosi, dan data penjualan.
8. Maulana et al., (2024) penelitian yang berjudul “*Development Of An E-Commerce Platform Using Extreme Programming Methodology*” dilakukan di Toko Andalas yang mengimplementasikan teknologi MERN Stack dan metodologi *Extreme Programming* sebagai platform *e-commerce* yang dikembangkan berhasil meningkatkan aksesibilitas bagi pelanggan, memungkinkan mereka untuk menjelajahi dan membeli buku dengan lebih mudah.
9. Penelitian yang dilakukan oleh Sholihin, (2023) yang berjudul “Penerapan *Payment Gateway Application Programming Interface* (API) Pada Sistem *Point Of Sales* (POS)” mengimplementasikan API *Midtrans* pada sistem *Point of Sales* (POS) membuat proses transaksi menjadi lebih cepat, dan efisien, pembayaran lebih terjamin

keamanannya, dan reputasi bisnis serta kepuasan konsumen meningkat. Metodologi *Rapid Application Development (RAD)* adalah metode yang digunakan penelitian.

10. Prasetyo & Sutopo, (2020) penelitian yang berjudul “Implementasi Layanan *Payment Gateway* Pada Sistem Informasi Transaksi Pembayaran” menyimpulkan bahwa sistem dengan pembayaran online mempermudah pengguna dalam transaksi pembayaran secara online dengan aman, admin dan vendor dapat mengetahui informasi status order dari pemesanan online pelanggan.
11. Habibullah et al., (2023) penelitian tentang “Aplikasi Toko Online dengan Penerapan Sistem *Payment Gateway* untuk Transaksi Pembayaran Berbasis Web” implementasi sistem *payment gateway* berbasis *web* dapat mempermudah proses jual-beli online seperti promosi, dan pengelolaan produk bagi pemilik usaha, dan pembeli dapat melihat langsung proses transaksi pembayaran dengan berbagai pilihan metode pembayaran online melalui *website*. Metodologi yang dipakai penelitian adalah *waterfall*.
12. Anugrah et al., (2023) penelitian yang berjudul “Sistem Informasi Penjualan Ikan Segar Di PT Tirta Lestari Indonesia Berbasis Framework Laravel” berhasil mengimplementasikan sistem untuk mempermudah proses transaksi pembayaran dengan *e-wallet*, pengelolaan stok produk ikan, dan pengiriman produk untuk pelanggan dengan metode *waterfall*.

13. Penelitian yang dilakukan oleh Lim & Ridho, (2021) yang berjudul “Rancang Bangun Sistem Informasi Point Of Sale Dengan Framework Codeigniter Pada CV Powershop” menghasilkan aplikasi *Point of Sale* yang memudahkan pembuatan laporan harian tanpa mencocokkan nota kontan penjualan, penginputan data, dan mengurangi penggunaan kertas. Metode yang digunakan pada penelitian adalah metode *waterfall*.
14. Rifandi, (2020) penelitian yang berjudul “*Analysis And Design Of Point Of Sale System In D'astore Shop*” menghasilkan sistem POS yang memberikan manfaat signifikan bagi pemilik toko dalam hal efisiensi operasional dan peningkatan pelanggan. Metode yang dipakai pada penelitian ini dengan wawancara, observasi, dan pustaka.
15. “*Implementation Of Information Systems On E-commerce Websites As Media To Deliver Information*” yang ditulis oleh Alwiyah et al., (2019) dengan adanya sistem informasi pada *website e-commerce* mahasiswa dapat dengan mudah mengetahui produk yang tersedia, sehingga pemilihan produk menjadi lebih efisien. Metode yang digunakan pada perancangan adalah metode kualitatif dan studi literatur.
16. Mulyana & Rusmawan, (2023), “Rancang Bangun Sistem Informasi *Point of Sale* (POS) Berbasis Web (Studi Kasus Toko Andorio)” peneliti berhasil mengimplementasikan sistem aplikasi sistem informasi *Point Of Sale* berbasis *web* untuk mempermudah kegiatan operasional Toko Andorio mencakup proses transaksi pembelian dan penjualan,

memudahkan rekapitulasi laporan pembelian, penjualan, dan stok.

peneliti menggunakan metode *waterfall* sebagai metode penelitian.