

BAB II

LANDASAN TEORI

2.1 Kajian Teori (*State Of The Art*)

Untuk menegaskan kebaruan dan posisi unik penelitian ini. Disajikan dalam beberapa penelitian yang relevan yang memiliki keterkaitan topik sebagai landasan untuk perbandingan yang dijadikan tinjauan penelitian.

Tabel 2.1 Penelitian Terdahulu (*State of the Art*)

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan yang dijadikan alasan tinjauan penelitian
1.	Widanengsih et al. (2024) https://jurnal.itscience.org/index.php/digitech/article/view/4381	Rancang Bangun Sistem Informasi Stok Barang Berbasis laravel	SDLC	Toko	Mengurangi kesalahan pencatatan dan meningkatkan akurasi stok, tapi belum spesifik pada bengkel dan tidak membahas laporan penjualan
2.	Murod et al. (2024) https://journal.eng.unila.ac.id/index.php/jitet/article/view/4706	Rancang Bangun Sistem Informasi Manajemen Persediaan Barang Menggunakan Laravel	SDLC	Perusahaan	Mengatasi pencatatan manual dan meningkatkan akurasi stok
3.	Wijaya & lomban (2022) https://doi.org/10.51401	Sistem <i>inventory</i> barang menggunakan metode <i>waterfall</i>	<i>Waterfall</i>	Pengelolaan stok usaha umum	Sistem pencatatan stok terstruktur, belum terintegrasi dengan transaksi

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan yang dijadikan alasan tinjauan penelitian
	/jinteks.v4i3.1963				
4.	Kurnia dan Aryani (2024) https://jurnal.polgan.ac.id/index.php/jmp/article/view/15306	Rancang Bangun Sistem Informasi Stok dan Penjualan Berbasis Web menggunakan Laravel	<i>Agile</i>	Toko Sparepart	Sistem menghasilkan pencatatan stok real-time, tetapi lebih berfokus pada transaksi penjualan
5.	Abhirama et al. (2023) https://doi.org/10.31539/intecom.v6i2.7662	Rancang Bangun Sistem Informasi berbasis web pada bengkel k41_garage	<i>Waterfall</i>	Bengkel K41_Garage	Sistem Mempermudah transaksi penjualan dan pengecekan sparepart, namun belum menyediakan laporan penjualan dan manajemen stok secara mendalam
6.	Linawati et al. (2023) https://doi.org/10.31294/justian.v4i2.2957	Rancang Bangun Sistem Informasi Penjualan dan Service Motor Berbasis Web dengan Metode <i>Prototype</i> pada Bengkel Maju Jaya Motor	<i>Prototype</i>	Bengkel Maju Jaya Motor	Sistem mampu membantu penjualan secara real-time, namun belum berfokus pada pengelolaan stok sparepart dan laporan penjualan secara terintegrasi

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan yang dijadikan alasan tinjauan penelitian
7.	Saifudin et al. (2023) https://openjournal.unpa.ac.id/index.php/JTSI/article/view/21197	Pengembangan Sistem Informasi Persediaan Barang Menggunakan Metode <i>Waterfall</i>	<i>Waterfall</i>	Sistem Persediaan Barang	Fokus kepada inventory nya, belum mengintegrasikan dengan Riwayat penjualan
8.	Mukti dan Latipah (2024) https://doi.org/10.30656/jsii.v1i1.8112	Rancang Bangun Aplikasi Sistem Informasi <i>Inventory</i> Menggunakan Metode Trend linier berbasis web pada bengkel	Kuantitatif	Bengkel UD Barokah Motor	Sistem dapat meningkatkan manajemen stok dan memprediksi kebutuhan barang dengan lebih baik.
9.	Suryani et al. (2024) https://e-journalbattuta.ac.id/index.php/jutek/article/view/114	Sistem Informasi <i>Inventori</i> Barang pada Bengkel Berbasis Web	<i>Waterfall</i>	Bengkel Alfiy Motor	Mempecepat pencatatan barang masuk dan keluar dan mengurangi kesalahan manual
10.	Lestari (2024) https://doi.org/10.23960/jitet.v13i2.6413	Pengembangan Sistem Manajemen Bengkel Menggunakan Laravel	Kuantitatif	Bengkel	Meningkatkan efisiensi operasional dan mengurangi kesalahan pencatatan
11.	Fathoni & Kustiyono (2025)	Pengembangan Sistem Informasi	<i>Waterfall</i>	Umum	Fokus pada efisiensi stok, belum terintegrasi

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan yang dijadikan alasan tinjauan penelitian
	https://doi.org/10.33364/algoritma/v22-2.2533	persediaan barang			secara menyeluruh
12.	Samuel et al. (2025) https://doi.org/10.31294/jais.v5i02.10443	Implementasi Sistem Informasi <i>Inventory</i> pada Klinik	<i>Waterfall</i>	Klinik	Sistem manajemen inventory pada klinik, objek nya berbeda
13.	Fauzan et al. (2024) https://doi.org/10.33884/jif.v12i02.9232	Merancang Sistem Informasi <i>Inventory</i> di PT Nubos perkasa jaya	<i>Waterfall</i>	PT Nubos perkasa jaya	Sistem informasi inventory, belum diterapkan transaksi
14.	Kurniasih & Hidayat (2025) https://doi.org/10.52436/1.jpti.816	Sistem Informasi manajemen stok berbasis Laravel	<i>Waterfall</i>	Umum	Sistem menggunakan Laravel, tapi bukan pada bengkel
15.	Sriyeni et al. (2024) https://doi.org/10.57152/malcom.v4i1.1159	Rancang Bangun Sistem Informasi Penjualan Motor dan Bengkel Menggunakan Metode <i>Prototype</i>	<i>Prototype</i>	PD. Panca Motor KM 14 Palembang	Sistem mengintegrasikan data penjualan dan bengkel, belum focus pada pengelolaan stok dan laporan penjualan.
16.	Pandu Widiansyah (2026)	Rancang Bangun Sistem Informasi <i>Workshop</i>	<i>Waterfall</i>	Setia Jaya Motor	Sistem Informasi berbasis Website Setia Jaya Motor

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan yang dijadikan alasan tinjauan penelitian
		Setia Jaya Motor untuk Pengelolaan Stok dan Laporan Penjualan Berbasis Website menggunakan <i>framework</i> Laravel			terintegrasi pengelolaan stok dan Laporan penjualan serta sistem POS untuk mengurangi kesalahan pada pencatatan yang masih manual serta terdapat notifikasi stok dan notifikasi aktivitas.

Berdasarkan penelitian terdahulu, sebagian besar penelitian hanya berfokus pada pengelolaan persediaan barang, sistem penjualan, atau laporan penjualan secara terpisah. Penelitian yang mengintegrasikan kategori sparepart, stok sparepart, barang masuk, barang keluar, sistem POS dan laporan penjualan, laporan stok, laporan penjualan, dan manajemen akses pengguna ke bengkel motor masih sangat terbatas. Karena itu, penelitian yang dapat mengintegrasikan semua proses ini ke dalam sistem informasi berbasis web adalah yang diperlukan. Kebaruan (*novelty*) penelitian ini terletak pada pengembangan sistem yang terintegrasi di *Workshop* Setia Jaya Motor berbasis web yang dibangun menggunakan *framework* Laravel dengan mengintegrasikan berbagai fungsi operasional ke dalam satu platform. Sistem tersebut mencakup pengelolaan kategori sparepart, data sparepart, pencatatan barang masuk dan barang keluar, transaksi penjualan, penyusunan laporan stok dan laporan penjualan, serta penyediaan fitur notifikasi. Selain itu, sistem menerapkan manajemen pengguna berbasis hak akses yang membedakan peran Administrator, Kasir, dan Pemilik sehingga setiap pengguna memperoleh akses sesuai dengan tanggung jawabnya.

Tabel 2.2 Tabel Perbandingan Metode

Aspek	<i>Waterfall</i>	<i>Prototype</i>	<i>Agile</i>
Pendekatan	Setiap tahapan dilakukan secara berurutan	Sistem dikembangkan melalui prototipe yang terus disempurnakan	Sistem dikembangkan secara bertahap dalam beberapa iterasi
Kebutuhan	Kebutuhan ditentukan secara jelas sejak awal pengembangan	Kebutuhan dapat berubah berdasarkan evaluasi prototipe	Kebutuhan bersifat dinamis dan dapat berubah selama pengembangan
Dokumentasi	Lengkap pada setiap tahapan pengembangan	Dokumentasi dibuat sesuai kebutuhan pengembangan	Lebih sederhana dengan fokus pada hasil pengembangan
Fleksibilitas	Rendah karena perubahan setelah implementasi cukup sulit dilakukan	Sedang karena perubahan dapat dilakukan melalui penyempurnaan prototipe	Tinggi karena sistem dirancang untuk mempermudah adaptasi terhadap perubahan
Cocok untuk sistem	Sistem dengan kebutuhan yang stabil dan telah diketahui sejak awal	Sistem yang kebutuhan pengguna belum jelas	Sistem yang sering mengalami perubahan kebutuhan dan pengembangan cepat.
Kelebihan	Tahapan pengembangan	Memudahkan pengguna	Fleksibel terhadap perubahan dan

Aspek	<i>Waterfall</i>	<i>Prototype</i>	<i>Agile</i>
	jelas, terstruktur, dan menghasilkan dokumentasi yang lengkap	memahami gambaran sistem sejak awal	mampu menyesuaikan kebutuhan pengguna dengan cepat
Kekurangan	Sulit melakukan perubahan apabila sistem sudah memasuki tahap implementasi	Membutuhkan revisi berulang sehingga waktu pengembangan dapat bertambah	Membutuhkan komunikasi yang intensif serta keterlibatan pengguna secara berkelanjutan

Berdasarkan tabel perbandingan di atas menunjukkan bahwa metode *Waterfall* dipilih karena paling sesuai dengan karakteristik penelitian ini. Sejak awal penelitian, kebutuhan sistem telah diidentifikasi dengan jelas. Hasil dari observasi dan wawancara memungkinkan pengembangan Sistem Informasi Pengelolaan Stok Sparepart dan Laporan Penjualan di *Workshop* Setia Jaya Motor. Kondisi tersebut memungkinkan proses pengembangan sistem dilaksanakan secara bertahap dengan urutan yang jelas, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga tahap pemeliharaan. Setiap fase dalam metode *Waterfall* menghasilkan dokumentasi yang sistematis sehingga mendukung pengendalian proses pengembangan, mempermudah penyusunan laporan tugas akhir, serta menjadi dasar dalam melakukan evaluasi terhadap hasil penelitian. Jika dibandingkan dengan metode *Prototype* maupun *Agile*, pendekatan *Waterfall* dinilai lebih sesuai untuk penelitian yang memiliki kebutuhan sistem dan ruang lingkup yang telah ditetapkan sejak awal sehingga tidak mengalami banyak perubahan selama proses pengembangan.

2.2 Tinjauan Pustaka

Kajian teori pada penelitian ini berfungsi sebagai dasar konseptual yang mendukung proses identifikasi permasalahan sekaligus menjadi acuan dalam menentukan solusi untuk pengembangan Sistem Informasi Pengelolaan Stok dan Laporan Penjualan di *Workshop Setia Jaya Motor*. Melalui landasan teori tersebut, penelitian memperoleh pijakan ilmiah dalam menganalisis kebutuhan sistem serta merancang solusi yang sesuai dengan kondisi dan permasalahan yang dihadapi.

Pembahasan dalam tinjauan pustaka mencakup berbagai konsep, teori, dan teknologi yang berkaitan dengan pengembangan sistem informasi. Materi yang dikaji meliputi konsep sistem informasi, sistem informasi berbasis web, sistem informasi bengkel, pengelolaan stok, sistem *Point of Sale* (POS), laporan penjualan, serta berbagai teknologi pendukung, seperti bahasa pemrograman PHP, framework Laravel, arsitektur *Model-View-Controller* (MVC), basis data MySQL, Tailwind CSS, metode *Waterfall*, dan lingkungan pengembangan aplikasi XAMPP. Seluruh konsep tersebut menjadi referensi dalam proses analisis, perancangan, implementasi, hingga pemeliharaan sistem.

Selain itu, bab ini menguraikan penggunaan *Unified Modeling Language* (UML) sebagai pendekatan untuk memodelkan sistem yang akan dikembangkan. Pemodelan dilakukan menggunakan beberapa diagram, yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Entity Relationship Diagram* (ERD) untuk menggambarkan kebutuhan, proses bisnis, struktur data, serta hubungan antarentitas dalam sistem. Pengujian fungsionalitas sistem dilakukan menggunakan metode *Black Box Testing* guna memastikan setiap fitur berjalan sesuai dengan kebutuhan yang telah ditetapkan. Dengan demikian, seluruh pembahasan pada bab ini menjadi landasan yang mendukung tahapan analisis, perancangan, implementasi, dan evaluasi Sistem Informasi Pengelolaan Stok dan Laporan Penjualan agar mampu meningkatkan efektivitas pengelolaan persediaan dan penyusunan laporan penjualan di *Workshop Setia Jaya Motor*.

2.2.1 Sistem Informasi Bengkel

Sistem informasi bengkel merupakan sistem yang dikembangkan untuk mendukung berbagai aktivitas operasional pada bengkel melalui pemanfaatan teknologi informasi. Sistem ini mencakup pengelolaan layanan, transaksi, serta persediaan barang dalam satu aplikasi yang terintegrasi. Dengan menggantikan proses yang sebelumnya dilakukan secara manual menjadi berbasis komputer, sistem mampu meningkatkan efektivitas pengelolaan, mempercepat proses kerja, serta menghasilkan data yang lebih akurat dan mudah dikelola. Penelitian oleh E. Suryani et al. (2021) Penerapan metode pengembangan sistem dalam pembangunan sistem informasi bengkel menunjukkan bahwa digitalisasi proses operasional mampu meningkatkan efisiensi layanan serta mempercepat pengelolaan dan pencatatan data. Selain itu, sistem yang dikembangkan memberikan kemudahan bagi pemilik bengkel dalam memantau seluruh aktivitas operasional secara lebih terorganisasi dan sistematis.

2.2.2 Website

Website merupakan media digital yang memanfaatkan jaringan internet sebagai sarana penyampaian informasi kepada pengguna. Informasi yang disajikan dapat berupa teks, gambar, audio, video, maupun layanan interaktif yang diakses melalui aplikasi peramban (*web browser*). Seiring dengan perkembangan teknologi, fungsi website tidak lagi terbatas sebagai media penyebaran informasi, tetapi juga dimanfaatkan untuk mendukung berbagai kebutuhan, seperti kegiatan bisnis, pendidikan, pemerintahan, hingga aktivitas pribadi melalui layanan yang terintegrasi. Ini memudahkan proses pengelolaan data, penyampaian informasi, dan akses ke layanan tanpa dibatasi oleh waktu dan lokasi (Rochmah & Suwardi, 2023).

Untuk membuat web dinamis, berbagai teknologi digunakan, termasuk *HyperText Markup Language* (HTML) sebagai struktur halaman, *Cascading Style Sheets* (CSS) sebagai pengaturan tampilan, dan bahasa pemrograman seperti PHP dan *JavaScript*. Selain itu, website modern umumnya terhubung dengan sistem basis data sehingga mampu mengelola informasi secara

terstruktur, mendukung proses autentikasi pengguna, pengolahan data, serta penyajian informasi secara real-time. Penerapan teknologi tersebut menjadikan website sebagai media yang fleksibel, efisien, dan mudah dikembangkan sesuai kebutuhan organisasi maupun perusahaan. Aplikasi web, atau web app, adalah pengganti halaman web statis. Aplikasi web ini interaktif dan responsif dan dapat diakses melalui berbagai perangkat, seperti komputer, tablet, dan telepon pintar. Untuk meningkatkan kinerja, keamanan, dan skalabilitas sistem, website kontemporer telah mengadopsi berbagai arsitektur pengembangan, termasuk desain web responsif dan aplikasi progresif (PWA). Oleh karena itu, situs web menjadi salah satu sumber penting untuk menerapkan sistem informasi berbasis web di banyak bidang, seperti perdagangan, pendidikan, kesehatan, dan pemerintahan (Sebastian et al., 2023).

2.2.3 Pengelolaan Stok

Dalam operasional bisnis, pengelolaan stok barang mencakup perencanaan, pencatatan, dan pengawasan barang yang masuk dan keluar untuk memastikan bahwa stok tetap tersedia sesuai kebutuhan. Penelitian oleh A. Saifudin et al. (2023) menunjukkan bahwa penerapan sistem inventori berbasis web mampu meminimalkan kesalahan pencatatan yang umumnya terjadi pada proses pengelolaan secara manual, sekaligus meningkatkan ketepatan data persediaan yang tersimpan dalam sistem. Ini menunjukkan betapa pentingnya sistem informasi dalam manajemen stok barang.

Pemanfaatan sistem informasi berbasis web maupun perangkat bergerak telah mengubah proses pengelolaan persediaan menjadi lebih terintegrasi. Kondisi ini memungkinkan aktivitas pemantauan stok dilakukan secara otomatis dengan pembaruan data yang berlangsung *real-time*. Informasi persediaan yang lebih akurat tersebut dapat dijadikan acuan dalam menentukan pengadaan barang, melakukan peramalan kebutuhan, serta menyusun laporan secara lebih efektif. Dengan demikian, sistem pengelolaan stok menjadi salah satu faktor penting dalam mendukung peningkatan efisiensi operasional dan daya saing perusahaan (Jimsan & Pradipta, 2023).

2.2.4 Point Of Sale (POS)

Sistem *Point of Sale* (POS) adalah suatu sistem informasi yang membantu mengelola transaksi penjualan di mana pelanggan membeli barang atau jasa. Sistem ini tidak hanya mencatat transaksi, tetapi juga mengintegrasikan berbagai proses bisnis seperti pengelolaan persediaan (*inventory*), pengelolaan data produk, pencatatan pelanggan, pelaporan penjualan, dan analisis kinerja bisnis *secara real-time* (Eko Andri Wibowo et al., 2025).

Penerapan sistem *Point of Sale* mampu meningkatkan efisiensi operasional melalui otomatisasi proses transaksi, mempercepat pelayanan kepada pelanggan, meningkatkan akurasi pengelolaan stok, serta mempermudah penyusunan laporan bisnis. Oleh karena itu, sistem POS menjadi salah satu komponen penting dalam digitalisasi bisnis, khususnya pada sektor perdagangan retail, bengkel, restoran, maupun usaha kecil dan menengah yang memerlukan pengelolaan transaksi dan persediaan secara terintegrasi (Aditya Lutfi et al., 2023).

2.2.5 Bahasa Pemrograman PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang dirancang untuk dijalankan pada sisi server (*server-side scripting*) dan banyak dimanfaatkan dalam pengembangan aplikasi berbasis web. Salah satu keunggulan PHP adalah kemampuannya dalam membangun halaman web yang bersifat dinamis serta mengolah data melalui integrasi dengan berbagai sistem manajemen basis data, seperti MySQL. Selain itu, PHP didukung oleh berbagai *framework*, di antaranya Laravel dan CodeIgniter, yang membantu pengembang menerapkan struktur aplikasi secara lebih terorganisasi sehingga proses pengembangan, pemeliharaan, dan pengembangan lanjutan dapat dilakukan dengan lebih efisien. Kombinasi antara fleksibilitas, kemudahan implementasi, serta dukungan komunitas yang luas menjadikan PHP tetap relevan dan banyak digunakan dalam pengembangan aplikasi web hingga saat ini (Kansha et al., 2023).

2.2.6 Framework Laravel

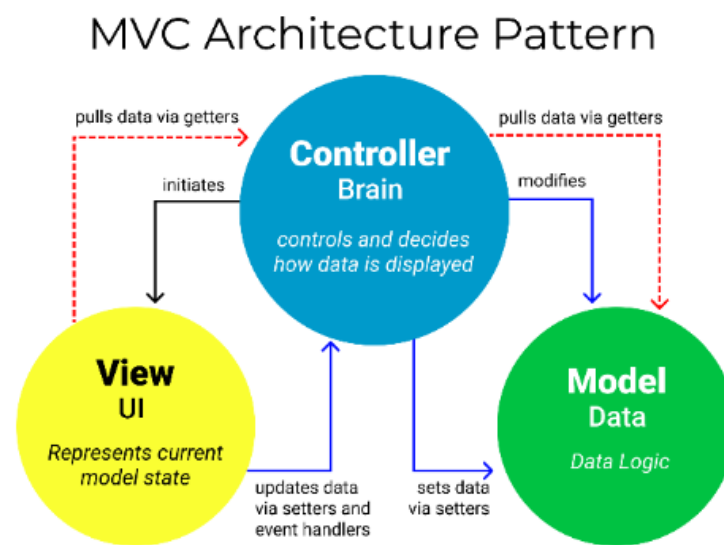
Laravel merupakan salah satu *framework* sumber terbuka (*open source*) yang dikembangkan menggunakan bahasa pemrograman PHP serta menerapkan pola arsitektur *Model-View-Controller* (MVC) sebagai dasar dalam pengembangan aplikasi. Dengan menyediakan berbagai fitur bawaan seperti *routing*, *authentication*, *middleware*, *database migration*, *Eloquent Object Relational Mapping* (ORM), *template engine Blade*, dan mekanisme pengelolaan dependensi melalui *Composer*, *framework* ini dimaksudkan untuk mempercepat proses pengembangan aplikasi web. Ketersediaan fitur-fitur tersebut memungkinkan pengembang membangun aplikasi yang lebih terstruktur, aman, mudah dipelihara, dan memiliki waktu pengembangan yang lebih efisien dibandingkan pengembangan menggunakan PHP murni (Sinlae et al., 2024).

Dalam pengembangan aplikasi web berbasis PHP, Laravel menjadi salah satu *framework* yang banyak digunakan karena menyediakan struktur kerja yang sistematis. Penerapan arsitektur *Model-View-Controller* (MVC) memungkinkan setiap komponen aplikasi, seperti logika program, tampilan, dan pengolahan data, dikelola secara terpisah sesuai fungsinya. Pendekatan tersebut memberikan kemudahan bagi pengembang dalam menyusun, mengembangkan, dan memelihara kode program sehingga aplikasi lebih mudah dikelola. Menurut M. Kurniasih dan W. Widayat (2022) Dalam pengembangan sistem informasi, Laravel dapat meningkatkan efisiensi pengembangan dan meningkatkan keamanan aplikasi. Selain itu, Laravel menawarkan fitur seperti *routing*, ORM (*Eloquent*), dan *middleware* yang membantu pengembangan sistem yang lebih terstruktur.

Berbagai sistem informasi berbasis web dikembangkan dengan memanfaatkan *framework* Laravel karena mampu memenuhi kebutuhan pengembangan aplikasi pada beragam sektor. Implementasinya dapat dijumpai pada lingkungan pendidikan, lembaga pemerintahan, aktivitas perdagangan, maupun kegiatan bisnis yang memerlukan aplikasi web yang terstruktur dan mudah dikelola.

Popularitas Laravel didukung oleh dokumentasi yang lengkap, komunitas pengembang yang aktif, serta ekosistem yang terus berkembang, seperti Laravel Breeze, Jetstream, Livewire, Inertia, dan Sail. Kombinasi antara kemudahan pengembangan, performa, keamanan, dan skalabilitas menjadikan Laravel sebagai framework yang banyak dipilih untuk membangun aplikasi web modern (Kadim et al., 2023).

2.2.7 Arsitektur MVC



Gambar 2.1 Arsitektur MVC

Sumber: (Rahmawati & Sumarsono, 2024)

Arsitektur *Model-View-Controller* (MVC) merupakan pendekatan dalam pengembangan perangkat lunak yang mengorganisasikan aplikasi ke dalam tiga komponen utama, yaitu *Model*, *View*, dan *Controller*. Masing-masing komponen memiliki peran yang berbeda dalam mengelola data, mengatur logika aplikasi, dan menampilkan antarmuka kepada pengguna. Pembagian tersebut bertujuan menciptakan struktur aplikasi yang lebih terorganisasi sehingga proses pengembangan, pemeliharaan, maupun pengembangan lanjutan dapat dilakukan secara lebih efisien. Oleh karena itu, pola MVC banyak diimplementasikan pada pengembangan aplikasi web

modern, termasuk pada framework Laravel yang digunakan dalam penelitian ini (Rahmawati & Sumarsono, 2024).

Pola arsitektur MVC menerapkan pembagian tugas ke dalam tiga komponen utama agar setiap bagian memiliki fungsi yang jelas. Pengelolaan data dan komunikasi dengan basis data dilakukan oleh *Model*, penyajian informasi kepada pengguna ditangani oleh *View*, sedangkan *Controller* berfungsi mengatur alur pemrosesan sekaligus menjadi penghubung antara data yang dikelola *Model* dengan tampilan yang disediakan *View*. Pembagian tanggung jawab tersebut membuat struktur aplikasi lebih terorganisasi dan mempermudah proses pengembangan maupun pemeliharaan sistem. Dengan pemisahan tersebut, pengembangan sistem menjadi lebih terstruktur, mudah dipelihara, dan memungkinkan pengembangan fitur baru tanpa mempengaruhi keseluruhan sistem (Pelango et al., 2024).

2.2.8 Tailwind CSS

Tailwind CSS dikembangkan untuk meningkatkan efisiensi proses pengembangan antarmuka web melalui penggunaan kelas utilitas yang mewakili setiap properti CSS, seperti pengaturan warna, ukuran, tipografi, margin, padding, display, hingga *responsive breakpoint*. Dengan konsep tersebut, pengembang dapat membangun halaman web secara lebih cepat tanpa harus berpindah antara dokumen HTML dan berkas CSS. Selain itu, Tailwind CSS mendukung proses *responsive web design* melalui sistem *breakpoint* yang telah terintegrasi sehingga tampilan aplikasi dapat menyesuaikan berbagai ukuran perangkat, mulai dari telepon pintar hingga komputer desktop (Azhariyah & Mukhlis, 2024).

Tailwind CSS banyak diimplementasikan bersama berbagai *framework* seperti Laravel, React, Vue, maupun Next.js karena mampu meningkatkan produktivitas pengembang serta menghasilkan antarmuka yang ringan, konsisten, dan mudah dikustomisasi. Melalui mekanisme *Just-In-Time (JIT) compiler*, Tailwind CSS hanya menghasilkan kelas CSS yang benar-benar digunakan dalam proyek sehingga ukuran berkas CSS menjadi lebih kecil dan performa aplikasi web dapat ditingkatkan. Pemanfaatan Tailwind CSS dalam

pengembangan aplikasi web terus meningkat karena mampu mendukung perancangan antarmuka yang adaptif terhadap berbagai ukuran perangkat. Selain memberikan fleksibilitas dalam proses desain, *framework* ini juga membantu meningkatkan efisiensi pengembangan sehingga sesuai digunakan pada pembangunan sistem informasi berbasis web (Mardiana et al., 2024).

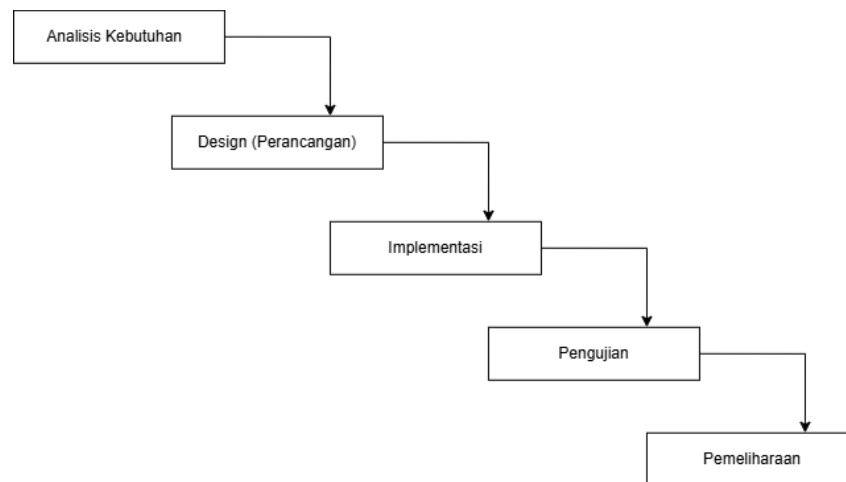
2.2.9 Database MySQL

MySQL merupakan salah satu sistem manajemen basis data relasional (*Relational Database Management System* atau RDBMS) yang banyak dimanfaatkan dalam pengembangan aplikasi web. Sistem ini menggunakan bahasa SQL (*Structured Query Language*) untuk mengelola, menyimpan, dan memproses data sehingga informasi dapat disusun secara terstruktur. Ini terlihat dalam pembuatan *backend* website profil perusahaan, di mana data konten, data pengguna, dan informasi pendukung lainnya disimpan menggunakan MySQL. Karena integrasinya dengan Laravel melalui fitur *migration* dan ORM *Eloquent*, pengelolaan *database* menjadi lebih mudah dan konsistensi *database* tetap konsisten (Sasongko, 2021).

2.2.10 Xampp

Dalam pengembangan aplikasi web, XAMPP banyak dimanfaatkan sebagai lingkungan server lokal karena menggabungkan berbagai perangkat lunak yang dibutuhkan dalam satu paket instalasi. Komponen yang tersedia meliputi Apache untuk menjalankan layanan web, MySQL sebagai pengelola basis data, PHP untuk memproses logika aplikasi, dan phpMyAdmin sebagai antarmuka administrasi basis data. Ketersediaan seluruh komponen tersebut dalam satu lingkungan kerja memberikan kemudahan bagi pengembang dalam membangun, menjalankan, dan menguji aplikasi sebelum diterapkan pada server produksi (Salsabila et al., 2024).

2.2.11 Metode *Waterfall*



Gambar 2.2 Metode Waterfall

Sumber: (Pressman,2020)

Menurut Roger S. Pressman (2020), Metode *Waterfall* adalah Pendekatan pengembangan perangkat lunak yang menerapkan tahapan kerja secara berurutan, mulai dari identifikasi kebutuhan sistem hingga proses pemeliharaan setelah implementasi. Penyelesaian setiap fase menjadi dasar untuk melanjutkan ke tahapan berikutnya sehingga alur pengembangan dapat dikendalikan dengan baik. Selain mendukung penyusunan dokumentasi secara lengkap, metode ini juga memudahkan proses pemantauan dan evaluasi selama pengembangan berlangsung. Dengan karakteristik tersebut, metode ini lebih tepat diterapkan pada proyek yang memiliki spesifikasi dan kebutuhan sistem yang relatif stabil sejak tahap perencanaan.

2.2.12 *Unified Modeling Language (UML)*

Pada pengembangan perangkat lunak berorientasi objek, *Unified Modeling Language (UML)* digunakan sebagai standar pemodelan untuk merepresentasikan kebutuhan dan rancangan sistem secara visual. UML membantu pengembang mendokumentasikan berbagai aspek sistem, mulai dari analisis hingga desain, sehingga rancangan dapat dipahami dengan lebih mudah sebelum proses implementasi dimulai. Selain mendukung penyusunan dokumentasi yang sistematis, penggunaan UML juga mempermudah

penyamaan persepsi antara pengembang dan pengguna terhadap fungsi serta kebutuhan sistem yang akan dibangun (Anardani et al., 2023).

1. Use Case Diagram

Use Case diagram dimanfaatkan sebagai alat permodelan untuk merepresentasikan interaksi antara aktor dan sistem berdasarkan kebutuhan fungsional yang harus dipenuhi. Diagram ini tidak hanya menunjukkan layanan atau fungsi yang tersedia bagi setiap aktor, tetapi juga menggambarkan batasan ruang lingkup sistem yang dibangun. Dengan adanya representasi tersebut, pengembang memperoleh pemahaman yang lebih jelas mengenai kebutuhan pengguna sehingga proses analisis dan perancangan sistem dapat dilaksanakan secara lebih terarah dan terdokumentasi dengan baik (Wayahdi & Ruziq, 2023).

Tabel 2.3 Simbol *Use Case*

Sumber: (Dicoding,2021)

Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili peran individu, sistem, atau alat saat berkomunikasi dengan use case.
	<i>Use Case</i>	Struktur dan interaksi antara aktor dan sistem
	<i>Association</i>	Struktur penghubung antara aktor dan use case

Simbol	Nama	Keterangan
	Generalisasi	Menunjukkan keahlian aktor untuk berpartisipasi dalam use case
	<i>Relationship (include)</i>	Menunjukkan bahwa fungsionalitas suatu aplikasi sepenuhnya identik dengan fungsionalitas suatu aplikasi lainnya
	<i>Relationship (Extend)</i>	Menunjukkan bahwa jika suatu kondisi terpenuhi, suatu use case memiliki fitur tambahan dari use case lainnya.

2. Activity Diagram

Activity Diagram dimanfaatkan sebagai media untuk memvisualisasikan aliran proses yang terjadi pada suatu sistem atau proses bisnis. Diagram ini menunjukkan urutan pelaksanaan setiap aktivitas beserta kemungkinan terjadinya percabangan berdasarkan kondisi atau keputusan tertentu hingga proses mencapai tahap akhir. Penyajian alur kerja secara visual membantu pengembang menganalisis proses bisnis dan memastikan setiap aktivitas dalam sistem telah dirancang secara logis serta terstruktur (Anggoro et al., 2024).

Tabel 2.4 Simbol *Activity* Diagram

Sumber: (Dicoding,2021)

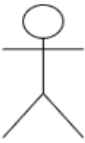
Simbol	Nama	Keterangan
	Status Awal / Titik Mulai	Status awal ditunjukkan pada diagram aktivitas.
	Aktivitas (<i>Activity</i>)	Semua aktivitas yang dilakukan sistem biasanya dimulai dengan kata kerja.
	Percabangan (<i>Decision</i>)	Menunjukkan suatu kondisi di mana ada lebih dari satu cara untuk melakukan sesuatu
	Penggabungan (Join)	Penggabungan adalah ketika lebih dari satu aksi digabungkan menjadi satu
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab atas tindakan yang dilakukan
	Status Akhir (<i>Activity Final Node</i>)	Menunjukkan status akhir sistem

3. Sequence Diagram

Sequence diagram dimanfaatkan untuk menggambarkan urutan komunikasi yang terjadi di antara objek-objek dalam suatu sistem ketika menjalankan sebuah proses. Representasi tersebut memperlihatkan aliran pesan dari awal hingga akhir sesuai kronologi eksekusinya, sehingga hubungan antarobjek dan alur proses dapat dipahami secara lebih jelas sebelum sistem diimplementasikan. (Anardani et al., 2023).

Tabel 2.5 Simbol Sequence Diagram

Sumber: (Dicoding,2021)

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang / pengguna yang berinteraksi dengan sistem tetapi tidak berada di dalamnya
	<i>Activation Box</i>	Menggambarkan Panjang waktu yang dibutuhkan sebuah objek dalam mengerjakannya
	<i>Lifeline</i>	Menggambarkan garis hidup sebuah objek
	<i>Message</i>	Menggambarkan pesan atau interaksi yang terjadi di antara objek

Simbol	Nama	Keterangan
	<i>Message To Self</i>	Menguraikan pesan balik atau reaksi yang dihasilkan oleh objek sebelumnya

4. Class Diagram

Class Diagram dimanfaatkan untuk memodelkan struktur statis sistem dengan menggambarkan kelas, atribut, metode, serta relasi yang terbentuk di antaranya. Diagram ini berfungsi sebagai acuan dalam pengembangan perangkat lunak berorientasi objek karena memberikan gambaran visual mengenai organisasi data dan hubungan antarobjek yang akan diterapkan pada sistem. Dengan adanya model tersebut, proses analisis, perancangan, dan pengembangan dapat dilakukan secara lebih sistematis dan konsisten. (Fatahila et al., 2023).

Tabel 2.6 Simbol *Class Diagram*

Sumber: (Sekawan Media,2023)

Simbol	Nama	Keterangan
	<i>Generalization</i>	hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dengan objek induk (<i>ancestor</i>)
	<i>N-ary Association</i>	Upaya untuk tidak terhubung dengan lebih dari satu objek

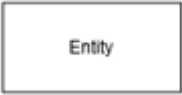
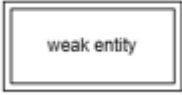
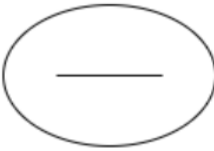
Simbol	Nama	Keterangan
	<i>Class</i>	Grup dari objek yang memiliki karakteristik dan operasi yang sama
	<i>Realization</i>	Suatu objek benar-benar melakukan operasi
	<i>Depedency</i>	hubungan di mana elemen yang tidak mandiri bergantung pada elemen mandiri yang berubah
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

5. Entity Relationship Diagram (ERD)

Diagram Hubungan Entitas (ERD) ialah sebuah model konseptual yang berfungsi untuk menggambarkan keterkaitan antara entitas di dalam database. ERD mendukung perancang sistem dalam mengidentifikasi entitas, atribut, dan relasi yang terdapat dalam database. Hal ini menjadikan struktur database yang dihasilkan menjadi lebih rapi, seragam, dan selaras dengan kebutuhan sistem.(Palinggi et al., 2024).

Tabel 2.7 Simbol ERD

Sumber: (Domainesia,2023)

Simbol	Nama	Keterangan
	<i>Entity</i>	Menunjukkan entitas utama yaitu objek nyata bisa berupa suatu element lingkungan atau sumber daya
	<i>Weak Entity</i>	Entitas lemah yang tidak bisa berdiri sendiri dan bergantung pada entitas lain seperti contohnya detail penjualan bergantung pada penjualan
	Attribute	Menunjukkan nama-nama atribut entity yang ada
	<i>Primary Key Attribute</i>	Atribut yang digaris bawah berfungsi sebagai kunci di antara nama-nama atribut yang ada pada suatu entitas

Simbol	Nama	Keterangan
	<i>Relationship</i>	Digunakan untuk menunjukkan hubungan yang ada antara entitas yang satu dengan entitas yang lainnya, ekspresi relasi ini
	<i>Link</i>	Menggambarkan sebagai hubungan antara himpunan hubungan dan himpunan entitas dengan atributnya

2.2.13 Black Box Testing

Pengujian menggunakan *Black Box Testing* dilakukan dengan menilai perilaku sistem berdasarkan hasil yang diperoleh dari setiap masukan yang diberikan, tanpa memperhatikan implementasi kode maupun struktur internal perangkat lunak. Fokus utama metode ini adalah memverifikasi bahwa seluruh fungsi dapat dijalankan sesuai dengan kebutuhan pengguna serta mendeteksi kesalahan yang berkaitan dengan fungsionalitas, antarmuka, validasi data, dan proses pengolahan data. Dalam penelitian ini, metode tersebut diterapkan pada seluruh fitur Sistem Informasi Pengelolaan Stok Sparepart dan Laporan Penjualan, termasuk autentikasi pengguna, pengelolaan kategori, data sparepart, barang masuk, barang keluar, transaksi penjualan, serta laporan penjualan. Hasil pengujian digunakan untuk memastikan bahwa setiap fitur telah berfungsi secara benar sesuai spesifikasi sistem yang dirancang pengguna (Praniffa et al., 2023.).

2.2.14 Profil Perusahaan



Gambar 2.3 Logo SJM

Setia Jaya Motor merupakan usaha yang bergerak di bidang jasa perawatan dan perbaikan kendaraan bermotor roda dua serta penjualan berbagai jenis sparepart. Bengkel ini didirikan pada awal tahun 2026 sebagai bentuk upaya untuk memenuhi kebutuhan masyarakat akan layanan perbaikan kendaraan yang cepat, berkualitas, dan didukung dengan ketersediaan suku cadang. Dalam menjalankan kegiatan usahanya, *Workshop* Setia Jaya Motor melayani berbagai jenis perbaikan, mulai dari servis berkala, penggantian oli, perbaikan sistem pengereman, perbaikan mesin, sistem kelistrikan, suspensi, hingga penggantian berbagai komponen kendaraan sesuai dengan kebutuhan pelanggan. Selain menyediakan layanan perbaikan, bengkel ini juga menyediakan berbagai sparepart dan aksesoris kendaraan bermotor untuk mendukung kebutuhan perawatan maupun penggantian komponen kendaraan.