

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem Informasi

Sistem informasi merupakan sebuah kesatuan yang terdiri dari komponen-komponen seperti perangkat keras, perangkat lunak, prosedur, basis data, dan sumber daya manusia yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi. Tujuan utama dari sistem informasi adalah menyediakan data yang akurat, relevan, dan tepat waktu sehingga dapat mendukung proses pengambilan keputusan serta meningkatkan efektivitas dan efisiensi suatu organisasi. Melalui pemanfaatan teknologi yang terstruktur, sistem informasi mampu mengotomatisasi berbagai aktivitas operasional sehingga mengurangi kesalahan manual, mempercepat proses kerja, dan memperbaiki kualitas layanan. Sistem informasi juga menjadi fondasi penting dalam mendukung proses bisnis modern yang membutuhkan kecepatan dan ketepatan dalam mengelola data.

Penerapan konsep dasar sistem informasi menjadi landasan penting dalam pengembangan sistem pemesanan *Service online* di bengkel karena seluruh proses pemesanan, pencatatan data pelanggan, pengaturan jadwal, hingga penyimpanan riwayat layanan membutuhkan aliran informasi yang terstruktur dan dapat diandalkan. Sistem yang dibangun harus mampu memproses data secara otomatis, menyediakan informasi jadwal yang akurat bagi pelanggan, serta membantu pihak bengkel dalam mengelola data *Service* tanpa risiko kehilangan atau kesalahan pencatatan. Dengan memanfaatkan prinsip-prinsip sistem informasi, layanan yang sebelumnya dilakukan secara manual dapat diubah menjadi proses digital yang lebih efisien, terorganisir, dan mudah diakses kapan saja oleh pelanggan maupun pengelola bengkel.

2.1.1. Pengertian Sistem

Sistem merupakan sekumpulan komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu melalui alur kerja yang terstruktur [1]. Komponen - komponen tersebut dapat berupa manusia, prosedur, metode, perangkat keras, maupun perangkat lunak yang saling terhubung sehingga membentuk sebuah kesatuan yang memiliki fungsi jelas. Sebuah sistem biasanya memiliki *Input*, proses, dan *Output* yang saling terkait serta dijalankan berdasarkan aturan atau mekanisme yang telah ditetapkan. Dengan adanya sistem, suatu organisasi atau aktivitas dapat berjalan lebih teratur karena setiap bagian memiliki peran yang mendukung tercapainya efisiensi, ketepatan, dan konsistensi dalam pengolahan informasi maupun pelaksanaan tugas.

Konsep sistem tersebut menjadi dasar penting dalam pengembangan layanan pemesanan *Service* berbasis web, karena proses reservasi, pengaturan jadwal, penyimpanan data pelanggan, serta pengelolaan informasi layanan harus berjalan secara terstruktur dan saling terhubung [2]. Setiap komponen dalam aplikasi mulai dari antarmuka pengguna, proses pemesanan, pencatatan otomatis, hingga pengolahan data oleh pihak bengkel harus dirancang sebagai satu kesatuan yang mampu mendukung alur kerja pemesanan *Service* secara menyeluruh. Pemahaman terhadap sistem memungkinkan pengembang merancang alur yang efisien, meminimalkan kesalahan, serta memastikan bahwa setiap informasi yang masuk dapat diproses menjadi keluaran yang berguna bagi pelanggan maupun pengelola bengkel. Dengan demikian, implementasi teknologi informasi dapat memberikan nilai tambah berupa peningkatan kualitas layanan dan optimalisasi operasional secara keseluruhan.

2.1.2. Pengertian Informasi

Informasi merupakan hasil dari pengolahan data mentah sehingga menghasilkan makna yang dapat dipahami dan digunakan oleh penerimanya [3]. Data yang belum diolah tidak memiliki arti, tetapi setelah melalui proses pemrosesan seperti pengorganisasian, klasifikasi, atau analisis data

tersebut berubah menjadi informasi yang memiliki nilai guna. Informasi berfungsi sebagai dasar pengambilan keputusan, karena mampu menggambarkan kondisi, peristiwa, atau kebutuhan secara lebih jelas dan terstruktur. Kualitas informasi sangat ditentukan oleh ketepatan, ketelitian, relevansi, serta kejelasan penyajiannya, sehingga informasi yang baik dapat memberikan manfaat maksimal bagi siapa pun yang menggunakannya.

Pemanfaatan informasi menjadi elemen penting dalam pengembangan sistem pemesanan *Service online* di bengkel otomotif, karena setiap proses pelayanan sangat bergantung pada aliran informasi yang tepat dan akurat [4]. Pengguna membutuhkan informasi mengenai jadwal ketersediaan *Service*, jenis layanan yang ditawarkan, serta status pemesanan yang mereka lakukan. Pihak bengkel juga membutuhkan informasi yang rapi dan terkelola untuk menyusun jadwal pekerjaan, meminimalkan kesalahan pencatatan, serta meningkatkan kualitas layanan.

Kehadiran sistem yang mampu mengelola informasi dengan baik melalui teknologi berbasis web memungkinkan seluruh proses pemesanan, pencatatan data, dan pengaturan jadwal menjadi lebih mudah, cepat, dan dapat dipantau secara real time sehingga meningkatkan efisiensi operasional dan kepuasan pelanggan.

2.1.3. Pengertian Sistem Informasi

Sistem informasi merupakan suatu kesatuan komponen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan data menjadi informasi yang bermanfaat bagi penggunanya [5]. Komponen - komponen tersebut mencakup perangkat keras, perangkat lunak, basis data, prosedur, dan sumber daya manusia yang bekerja secara terintegrasi untuk menghasilkan informasi yang akurat, relevan, dan tepat waktu. Sistem informasi tidak hanya berfungsi sebagai alat bantu pemrosesan data, tetapi juga sebagai sarana pendukung pengambilan keputusan, koordinasi, pengendalian, serta analisis dalam berbagai aktivitas organisasi. Dengan adanya sistem informasi yang

terstruktur, berbagai proses operasional dapat berjalan lebih efektif dan efisien sesuai dengan kebutuhan organisasi.

Penerapan konsep sistem informasi memberikan peluang besar untuk meningkatkan kualitas layanan pada proses pemesanan *Service* kendaraan di sebuah bengkel. Integrasi data pelanggan, jadwal *Service*, jenis layanan, dan proses *Administrasi* dalam satu platform digital memungkinkan pengelolaan layanan dilakukan lebih cepat, tepat, dan minim kesalahan. *TransFormasi* dari pencatatan manual menjadi sistem informasi berbasis web membuat alur pelayanan menjadi lebih transparan dan mudah dipantau, baik oleh pihak bengkel maupun pelanggan.

Selain itu, kehadiran sistem informasi mampu mengurangi ketergantungan pada proses *Administratif* konvensional serta meningkatkan kepuasan pelanggan melalui penyediaan layanan yang lebih teratur, responsif, dan modern.

2.2. Alat Bantu dalam Pengembangan Sistem

Alat bantu dalam pengembangan sistem berperan penting untuk mempermudah proses perancangan, pemodelan, hingga implementasi perangkat lunak. Beberapa alat bantu yang umum digunakan meliputi diagram alir data (DFD), diagram konteks, flowchart, *Unified Modeling Language (UML)*, serta perangkat pendukung seperti text editor, framework, dan aplikasi manajemen basis data. Penggunaan alat bantu tersebut membantu pengembang memahami alur kerja sistem, menggambarkan struktur proses, serta memastikan bahwa rancangan yang dibuat sesuai dengan kebutuhan pengguna. Selain itu, alat bantu juga mendukung dokumentasi yang baik sehingga memudahkan proses evaluasi dan pemeliharaan sistem di kemudian hari.

Penggunaan alat bantu tersebut sangat mendukung proses perancangan sistem pemesanan *Service online* di Bengkel Otomotif Risky Painting karena membantu menguraikan kebutuhan pengguna, memetakan alur pemesanan, serta menggambarkan hubungan antara pengguna dan

sistem secara lebih terstruktur. Diagram seperti *use case* dan *Activity Diagram* memudahkan pengembang dalam memahami interaksi pelanggan ketika melakukan pemesanan layanan, sementara *Class Diagram* dan perancangan basis data membantu menyusun struktur informasi agar mudah dikelola. Pemanfaatan alat bantu ini menghasilkan rancangan yang lebih jelas, mengurangi potensi kesalahan selama pengembangan, serta mempercepat proses implementasi sistem berbasis PHP yang dirancang menggunakan metode *waterfall* sehingga setiap tahap dapat dijalankan secara terarah dan sistematis.

2.2.1. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk menggambarkan, merancang, dan memvisualisasikan sistem perangkat lunak secara terstruktur [6]. UML menyediakan berbagai jenis diagram, seperti *Use case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Activity Diagram*, yang masing-masing memiliki fungsi spesifik untuk menunjukkan alur proses, hubungan antarkomponen, hingga perilaku sistem secara keseluruhan [7]. Pemodelan dengan UML membantu pengembang memahami kebutuhan sistem, mengurangi ambiguitas, serta memberikan gambaran yang jelas sebelum proses pengembangan dilakukan. Dengan pendekatan visual ini, setiap elemen sistem dapat dijelaskan secara lebih rinci sehingga mempermudah koordinasi antaranggota tim maupun pihak yang berkepentingan.

UML digunakan untuk menggambarkan alur pemesanan *Service*, interaksi pengguna dengan sistem, struktur basis data yang dibutuhkan, serta logika proses yang berjalan pada sistem berbasis web yang akan dibangun [8]. Pemanfaatan UML memudahkan penyusunan kebutuhan fungsional dan nonfungsional karena setiap diagram memberikan representasi visual yang jelas, mulai dari bagaimana pelanggan melakukan reservasi, bagaimana *Admin* mengelola jadwal, hingga bagaimana sistem memproses data pemesanan. Selain itu, penggunaan UML membantu memastikan bahwa desain sistem telah terstruktur dengan baik sebelum memasuki tahap

pengembangan sesuai metode *waterfall*, sehingga setiap tahapan dapat berjalan lebih terarah, sistematis, dan minim kesalahan [9]. Dengan adanya pemodelan yang rinci menggunakan UML, implementasi sistem menjadi lebih mudah dilakukan karena setiap komponen sudah memiliki gambaran desain yang matang dan terdokumentasi dengan baik.

1. Use case Diagram

Use case Diagram merupakan alat pemodelan yang digunakan untuk menggambarkan interaksi antara pengguna dengan sistem melalui serangkaian aktivitas yang disebut *use case* [10]. Diagram ini menampilkan aktor yang terlibat, fungsi yang dapat dijalankan, serta hubungan di antara keduanya, sehingga memberikan pemahaman yang jelas mengenai kebutuhan fungsional sistem sebelum proses pengembangan dilakukan. Dengan memvisualisasikan alur dan peran pengguna, *Use case Diagram* membantu pengembang dalam memastikan bahwa setiap fungsi yang dirancang benar-benar sesuai dengan kebutuhan operasional dan tujuan system [11].

Tabel 2. 1. Simbol Use case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Penerapan *Use case Diagram* memberikan gambaran menyeluruh mengenai alur interaksi antara pelanggan, *Admin*, dan mekanik dengan sistem pemesanan *Service online* yang akan dibangun [12]. Diagram ini membantu memetakan proses pemesanan layanan, pengelolaan jadwal,

pengolahan data *Service*, hingga proses konfirmasi pemesanan secara lebih terstruktur. Melalui pemodelan tersebut, kebutuhan fungsional seperti pendaftaran pengguna, pembuatan reservasi, pengecekan status, dan manajemen data dapat diidentifikasi secara jelas, sehingga sistem yang dikembangkan dapat berjalan sesuai prosedur kerja bengkel [13]. Representasi visual ini juga memungkinkan tim pengembang mengevaluasi kelengkapan fitur dan mengurangi potensi kesalahan dalam perancangan, sehingga menghasilkan aplikasi yang lebih efektif, terarah, dan mampu meningkatkan kualitas pelayanan di Bengkel Otomotif Risky Painting.

2. Activity Diagram

Activity Diagram merupakan salah satu jenis diagram pemodelan yang digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi dalam suatu sistem secara terstruktur [14]. Diagram ini menampilkan urutan langkahlangkah kerja, keputusan, dan aliran informasi dari satu aktivitas ke aktivitas lainnya. Dengan visualisasi yang jelas, *Activity Diagram* memudahkan pengembang untuk memahami bagaimana suatu proses berjalan, siapa yang terlibat, serta bagaimana alur operasional dipetakan dari awal hingga akhir. Selain itu, diagram ini sangat efektif untuk mengidentifikasi potensi hambatan, redundansi, maupun kebutuhan sistem sebelum proses pengembangan dimulai, sehingga dapat membantu perancangan solusi yang lebih akurat dan efisien.

Tabel 2.2. Simbol Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Aktivitas pemesanan *Service*, pengelolaan jadwal, dan konfirmasi layanan dapat digambarkan lebih jelas melalui *Activity Diagram* sehingga alur kerja dapat dipahami oleh pengembang maupun pihak bengkel secara lebih mendetail. Penggambaran langkah-langkah mulai dari pelanggan melakukan *Login*, memilih jenis layanan, menentukan jadwal, hingga *Admin* memverifikasi pemesanan menjadi lebih sistematis dan terstruktur. Alur tersebut membantu memastikan bahwa setiap kebutuhan operasional bengkel dapat diterjemahkan ke dalam sistem secara tepat dan tidak ada proses penting yang terlewat. Dengan visualisasi ini, proses analisis kebutuhan hingga tahap implementasi dapat dilakukan dengan lebih terarah, serta membantu menghasilkan sistem yang benar-benar sesuai dengan alur kerja yang dijalankan di bengkel.

3. Class Diagram

Class Diagram merupakan salah satu bagian penting dalam perancangan sistem berorientasi objek yang berfungsi untuk memodelkan struktur statis dari suatu perangkat lunak. Diagram ini menggambarkan kelas-kelas yang membentuk sistem, atribut yang dimiliki setiap kelas, metode yang dapat dijalankan, serta hubungan antarkelas seperti asosiasi, agregasi, komposisi, maupun pewarisan. Dengan adanya *Class Diagram*, pengembang dapat memahami bagaimana data disimpan, bagaimana objek saling berinteraksi, dan bagaimana alur logika sistem dibangun.

Tabel 2. 3. Simbol - Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Class Diagram berperan sebagai landasan penting dalam membangun sistem pemesanan *Service online* di Bengkel Otomotif Risky Painting, karena mampu menggambarkan kebutuhan data dan interaksi antarentitas seperti pelanggan, layanan *Service*, jadwal pemesanan, dan *Admin* pengelola. Diagram ini memastikan setiap fitur yang dirancang, seperti pembuatan akun, pemesanan *Service*, pengaturan jadwal, serta pengelolaan

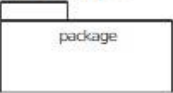
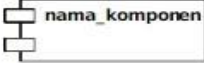
data layanan, memiliki struktur yang logis dan saling terhubung. Dengan model yang jelas, proses implementasi menggunakan PHP menjadi lebih terarah, meminimalkan kesalahan desain, serta memudahkan proses pengembangan pada setiap tahap metode *waterfall*, mulai dari analisis hingga pemeliharaan. *Class Diagram* juga mempermudah proses evaluasi dan pengembangan lanjutan jika sistem perlu ditingkatkan di masa mendatang.

4. Component Diagram

Component Diagram merupakan salah satu diagram pada *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan struktur sistem dari sudut pandang komponen - komponen penyusunnya. Diagram ini menunjukkan bagaimana bagian - bagian utama perangkat lunak diorganisasi, saling berinteraksi, dan bagaimana dependensi antar komponen terbentuk untuk menjalankan fungsi tertentu. Komponen yang digambarkan biasanya berupa modul, paket, kelas besar, antarmuka, atau layanan yang berperan sebagai blok bangunan sistem.

Component Diagram membantu menggambarkan arsitektur perangkat lunak pada level tinggi sehingga memudahkan pengembang dalam memahami pemisahan tanggung jawab setiap bagian, integrasi antar komponen, serta aliran kerja sistem secara keseluruhan. Dengan struktur yang jelas, diagram ini membantu memastikan bahwa setiap komponen memiliki peran yang spesifik dan tidak saling tumpang tindih, serta memudahkan proses perawatan maupun pengembangan sistem pada tahap berikutnya.

Tabel 2.4. Simbol - Simbol *Component Diagram*

Simbol	Diskripsi
Package 	package merupakan sebuah bungkus dari satu atau lebih komponen
Komponen 	Komponen sistem
Kebergantungan / dependency	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai

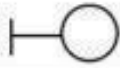
Penerapan *Component Diagram* pada sistem pemesanan *Service online* di Bengkel Otomotif Risky Painting bertujuan untuk memperjelas hubungan dan pembagian tugas antar komponen utama seperti komponen antarmuka pengguna, komponen pengelolaan pemesanan, komponen pengolahan data *Service*, serta komponen basis data. Diagram ini membantu menunjukkan bagaimana alur kerja dimulai dari pelanggan yang melakukan reservasi melalui browser, kemudian diteruskan ke komponen pemrosesan pemesanan yang mengelola jadwal, mencatat detail layanan, hingga menyimpan data ke dalam basis data yang terstruktur. Selain itu, komponen *Admin* yang digunakan pihak bengkel juga diperlihatkan keterhubungannya dengan komponen lain sehingga seluruh proses, mulai dari *Input* data, verifikasi jadwal, hingga pengelolaan riwayat *Service* dapat berjalan terintegrasi. Dengan pemetaan ini, pengembangan sistem menjadi lebih terarah karena setiap bagian sistem memiliki tanggung jawab yang jelas serta koneksi yang logis, sehingga sistem dapat dibangun dengan arsitektur yang kuat, mudah dipelihara, dan mampu mendukung kebutuhan layanan *Service* secara efisien.

5. *Sequence Diagram*

Sequence Diagram merupakan salah satu jenis diagram pemodelan pada *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan alur interaksi antar objek dalam suatu sistem secara

berurutan dari waktu ke waktu. Diagram ini memperlihatkan bagaimana pesan dikirimkan antar objek, bagaimana setiap objek merespons pesan tersebut, serta urutan proses yang terjadi selama suatu fungsi atau fitur dijalankan. Dengan menampilkan lifeline dan pesan yang saling berkaitan, *Sequence Diagram* membantu pengembang memahami detail proses bisnis, alur logika sistem, serta hubungan antar komponen secara lebih jelas.

Tabel 2. 5. Simbol - Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Sequence Diagram digunakan untuk menggambarkan alur interaksi antara pengguna, sistem, dan komponen pendukung ketika layanan pemesanan *Service* dijalankan. Diagram ini menjelaskan langkah-langkah mulai dari pengguna melakukan *Login*, memilih layanan, membuat jadwal *Service*, hingga sistem menyimpan data pemesanan dan memberikan konfirmasi. Setiap pesan yang ditampilkan menunjukkan proses yang terjadi secara runtut, sehingga seluruh mekanisme pemesanan dapat terlihat jelas dan terstruktur. Penggunaan *Sequence Diagram* membantu memastikan setiap langkah kerja sistem terdefinisi dengan baik, meminimalkan kesalahan logika, serta memberikan gambaran yang lengkap mengenai bagaimana komponen saling berkomunikasi. Diagram ini berperan penting dalam memastikan sistem yang dibangun berjalan sesuai kebutuhan, mudah

diimplementasikan, dan mampu memenuhi fungsionalitas yang diharapkan oleh pengguna maupun pihak pengelola bengkel.

2.3. Database

Database merupakan kumpulan data yang terorganisir dan disimpan secara sistematis sehingga dapat diakses, dikelola, dan diperbarui dengan mudah [15]. Struktur *Database* dirancang untuk memungkinkan penyimpanan informasi dalam bentuk tabel yang saling terhubung, sehingga mempermudah proses pencarian, pengelompokan, dan pengolahan data [16]. Dengan menggunakan sistem manajemen *Database*, data dapat disimpan secara aman, konsisten, dan terhindar dari duplikasi. Selain itu, *Database* berperan penting dalam menjaga integritas data, memastikan bahwa setiap informasi yang tersimpan tetap akurat dan dapat digunakan secara efektif untuk mendukung berbagai kebutuhan operasional maupun analisis.

Penerapan *Database* sangat diperlukan untuk mendukung pengelolaan data pemesanan *Service* kendaraan, data pelanggan, serta informasi jadwal layanan secara terstruktur dan terintegrasi [17]. Seluruh proses *Input*, pembaruan, maupun penelusuran riwayat pemesanan akan menjadi lebih cepat dan akurat ketika ditangani menggunakan sistem *Database*. Penggunaan *Database* juga memungkinkan pihak bengkel menghindari kesalahan pencatatan manual, meminimalkan kehilangan data, serta memudahkan proses pengelolaan informasi secara berkelanjutan [18]. Keberadaan *Database* memberikan fondasi yang kuat bagi sistem pemesanan *Service online*, karena setiap fitur dalam aplikasi membutuhkan data yang terorganisir untuk dapat berjalan dengan baik dan menghasilkan layanan yang lebih efektif serta tepat waktu.

2.3.1. MySQL

MySQL merupakan salah satu sistem manajemen basis data relasional (RDBMS) yang paling banyak digunakan untuk mengelola dan menyimpan data secara terstruktur [19]. *MySQL* mendukung bahasa SQL sebagai dasar

pengelolaan data, seperti melakukan query, insert, update, dan delete. Keunggulannya terletak pada kinerja yang cepat, stabilitas yang tinggi, serta kemampuan menangani jumlah data besar dengan efisiensi yang baik [20]. *MySQL* juga bersifat open-source sehingga memberikan fleksibilitas bagi pengembang untuk menyesuaikan konfigurasi sesuai kebutuhan. Selain itu, *MySQL* kompatibel dengan berbagai bahasa pemrograman dan platform server, sehingga memudahkan integrasi dalam berbagai jenis aplikasi, terutama aplikasi berbasis web.

Penggunaan *MySQL* memberikan dukungan penting dalam pembangunan sistem pemesanan *Service online* untuk pengelolaan data pelanggan, data pemesanan, jadwal *Service*, serta riwayat layanan yang tersimpan secara terstruktur dan mudah diakses. Sistem membutuhkan basis data yang mampu memproses transaksi secara cepat dan menjaga konsistensi informasi agar proses pemesanan berjalan lancar dan akurat. *MySQL* menawarkan kemampuan untuk menangani banyak permintaan secara bersamaan sehingga cocok digunakan untuk aplikasi layanan yang diakses oleh pelanggan dan *Admin* secara real time. Dengan struktur tabel yang dapat dirancang sesuai kebutuhan dan kemudahan dalam melakukan pengelolaan data, *MySQL* menjadi fondasi yang kuat untuk membangun sistem yang stabil, responsif, dan mendukung kelancaran operasional bengkel secara keseluruhan.

2.3.2. Konsep Dasar Database MySQL

MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang banyak digunakan untuk membangun aplikasi berbasis web karena bersifat opensource, cepat, serta mampu menangani volume data yang besar. *MySQL* menggunakan bahasa SQL (Structured Query Language) untuk mengelola data melalui proses seperti penyimpanan, pengubahan, penghapusan, dan pengambilan data. Struktur basis datanya memungkinkan pengorganisasian informasi dalam bentuk tabel yang saling terhubung menggunakan relasi, sehingga memudahkan pengembang dalam mengelola data secara sistematis dan terstruktur. Selain itu, *MySQL* memiliki dukungan

keamanan, skalabilitas, serta kompatibilitas tinggi dengan berbagai bahasa pemrograman, sehingga menjadi pilihan utama bagi banyak pengembang aplikasi modern.

Penggunaan *MySQL* memberikan keuntungan besar dalam membangun sistem pemesanan *Service* berbasis web karena mampu menyimpan berbagai data yang diperlukan secara terorganisir, seperti data pelanggan, jadwal *Service*, jenis layanan, dan riwayat pemesanan. Kemampuan *MySQL* dalam mengeksekusi query dengan cepat mendukung proses pemesanan agar dapat ditampilkan secara realtime, sehingga pelanggan dan pihak bengkel memperoleh informasi terbaru mengenai ketersediaan jadwal. Struktur relasionalnya juga mempermudah pengelolaan data yang saling terhubung, seperti menghubungkan pelanggan dengan layanan yang dipilih atau mengatur pemetaan waktu pengerjaan *Service*. Dengan dukungan fitur keamanan, *MySQL* membantu menjaga kerahasiaan data pelanggan, memastikan integritas informasi, dan meminimalkan risiko kesalahan penyimpanan data dalam sistem berbasis web yang dibangun.

2.4. Sekilas Tentang PHP

PHP merupakan salah satu bahasa pemrograman server-side yang banyak digunakan untuk membangun aplikasi berbasis web. Bahasa ini dikenal karena sintaksnya yang sederhana, fleksibel, serta didukung oleh komunitas pengembang yang sangat besar. PHP bekerja dengan cara memproses skrip pada server untuk kemudian menghasilkan *Output* berupa halaman web yang dinamis. Selain itu, PHP memiliki kompatibilitas yang tinggi dengan berbagai sistem manajemen basis data seperti *MySQL*, *MariaDB*, dan *PostgreSQL*, sehingga memudahkan pengembang dalam membangun sistem informasi yang membutuhkan pengolahan data secara intensif. Keunggulan lainnya adalah ketersediaan berbagai framework serta library yang membantu mempercepat proses pengembangan, menjadikan PHP tetap relevan dan banyak digunakan meskipun telah hadir berbagai teknologi baru.

Penggunaan PHP sangat cocok untuk membangun sistem pemesanan layanan karena kemampuan bahasa ini dalam menangani proses *Input* pengguna, mengelola data secara terstruktur, serta menghasilkan antarmuka web yang interaktif. PHP memungkinkan pengembangan fitur seperti pendaftaran pengguna, pengelolaan jadwal, serta pencatatan riwayat layanan secara efisien dan mudah diintegrasikan dengan *Database*. Selain itu, stabilitasnya dalam menjalankan aplikasi berskala kecil hingga menengah membuatnya ideal untuk diterapkan pada sistem pemesanan *Service* di bengkel, yang membutuhkan kinerja cepat, keandalan tinggi, serta kemudahan pemeliharaan. Fleksibilitas PHP juga memberikan peluang bagi pengembangan sistem agar dapat ditingkatkan di kemudian hari sesuai kebutuhan operasional dan pertumbuhan layanan bengkel.

2.5. Sistem Informasi Pemesanan *Service Online*

2.5.1. Pengertian Sistem Informasi Pemesanan *Service Online* Fasilitas

Sistem informasi pemesanan *Service online* merupakan sebuah platform berbasis web yang dirancang untuk mempermudah proses reservasi layanan secara digital. Sistem ini berfungsi sebagai penghubung antara pelanggan dan penyedia jasa melalui fitur - fitur yang memungkinkan pengguna melihat ketersediaan jadwal, memilih jenis layanan, hingga melakukan pemesanan tanpa harus hadir secara langsung. Dengan adanya otomatisasi proses pencatatan, pengaturan waktu, dan penyimpanan data, sistem ini mampu mengurangi risiko kesalahan manual serta meningkatkan efisiensi pelayanan. Selain itu, informasi terkait status pemesanan, estimasi waktu pengerjaan, dan riwayat *Service* dapat disajikan secara terstruktur sehingga mendukung transparansi dan kenyamanan pelanggan.

Penerapan sistem informasi pemesanan *Service online* di Bengkel Otomotif Risky Painting menjadi relevan karena mampu menjawab berbagai kendala yang muncul dari proses reservasi konvensional. Melalui sistem yang dirancang menggunakan bahasa pemrograman PHP dan dikembangkan dengan metode *waterfall*, alur pemesanan dapat diatur secara

lebih terstruktur sehingga memudahkan pelanggan dalam memilih layanan serta memudahkan pihak bengkel dalam mengelola jadwal *Service*. Aksesibilitas berbasis web memungkinkan pelanggan melakukan pemesanan kapan saja, sementara pihak bengkel memperoleh manfaat berupa pencatatan data yang lebih rapi, terintegrasi, dan minim kesalahan. Kehadiran sistem ini tidak hanya meningkatkan kualitas pelayanan, tetapi juga mendukung proses bisnis bengkel agar lebih modern, responsif, dan kompetitif.

2.5.2. Konsep Dasar Sistem Informasi Pemesanan *Service Online*

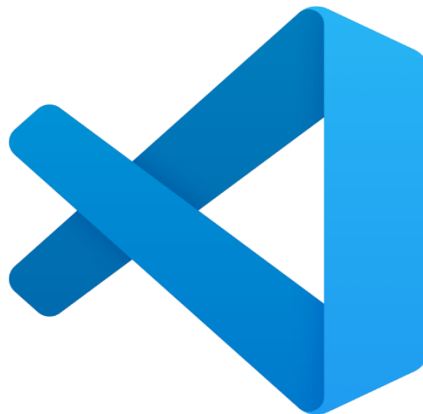
Sistem informasi pemesanan *Service online* merupakan sebuah platform berbasis web yang dirancang untuk memfasilitasi proses reservasi layanan secara digital. Sistem ini memungkinkan pelanggan untuk memilih jenis *Service*, menentukan jadwal, dan melakukan pendaftaran tanpa harus datang langsung ke lokasi. Konsep ini mengandalkan pemrosesan data secara otomatis, penyimpanan informasi dalam *Database*, dan penyajian antarmuka yang mudah digunakan sehingga seluruh aktivitas pemesanan dapat berlangsung lebih cepat, akurat, dan terstruktur. Selain itu, sistem ini bertujuan mengurangi ketergantungan pada pencatatan manual dan komunikasi tatap muka yang sering kali memakan waktu dan rentan terjadi kesalahan. Dengan adanya otomasi, proses pengelolaan jadwal, pemantauan status *Service*, serta penyampaian informasi kepada pelanggan dapat dilakukan secara lebih efisien dan efektif.

Pemanfaatan sistem informasi pemesanan *Service online* sangat relevan untuk diterapkan pada bengkel otomotif yang membutuhkan manajemen jadwal dan pelayanan pelanggan yang lebih optimal. Sistem berbasis web yang dikembangkan menggunakan PHP dapat menjadi solusi untuk meningkatkan akurasi pencatatan data, meminimalkan antrean, serta memberikan fleksibilitas bagi pelanggan dalam melakukan reservasi kapan saja. Selain itu, proses pengembangan menggunakan metode *waterfall* memungkinkan desain sistem dilakukan secara bertahap dan terstruktur sehingga fitur - fitur yang dibutuhkan bengkel dapat diterapkan dengan

baik. Melalui penerapan sistem ini, proses kerja di bengkel dapat berjalan lebih terorganisir, staf lebih mudah mengelola permintaan *Service*, dan pelanggan memperoleh pengalaman pelayanan yang lebih modern serta efisien.

2.6. Aplikasi Visual Studio Code

Visual Studio Code merupakan sebuah *source - code editor* yang dikembangkan oleh *Microsoft* dan dikenal sebagai salah satu editor paling fleksibel dan ringan untuk pengembangan perangkat lunak. Aplikasi ini mendukung berbagai bahasa pemrograman seperti *PHP*, *Java Script*, *Python*, dan banyak lainnya melalui sistem ekstensi yang mudah ditambahkan. Fitur - fitur unggulannya meliputi *Intellisense*, *debugging* terpadu, *version control* berbasis *Git*, terminal internal, serta kemampuan kostumisasi yang tinggi melalui tema dan konfigurasi. Kemudahan navigasi kode, per *Forma* yang stabil, serta dukungan komunitas yang kuat menjadikan *Visual Studio Code* pilihan populer bagi pengembang yang membutuhkan lingkungan pengembangan modern dan efisien.



Gambar 2. 1. Aplikasi Visual Studio Code

Visual Studio Code memberikan dukungan optimal dalam proses pengembangan sistem pemesanan *Service online* berbasis *PHP* yang dirancang untuk meningkatkan kualitas layanan bengkel. Fitur *Intellisense* membantu mempercepat penulisan kode dengan menampilkan saran fungsi dan variabel secara otomatis, sehingga meminimalkan kesalahan sintaks.

Terminal bawaan memungkinkan pengembang menjalankan server lokal, melakukan migrasi *Database*, atau menguji fungsi aplikasi tanpa harus beralih ke aplikasi lain. Integrasi Git memudahkan pengelolaan versi kode, terutama ketika melakukan revisi dan penyempurnaan fitur. Kombinasi kemampuan tersebut menjadikan *Visual Studio Code* sebagai alat yang sangat mendukung proses pengembangan sistem agar dapat dibangun secara terstruktur, rapi, dan sesuai kebutuhan fungsional yang telah direncanakan.

2.7. Aplikasi XAMPP

XAMPP merupakan sebuah paket aplikasi yang berisi server web Apache, *Database MySQL/MariaDB*, serta interpreter untuk bahasa pemrograman PHP dan Perl yang dirancang agar mudah diinstal dan digunakan oleh pengembang. Aplikasi ini bersifat open-source serta mendukung berbagai sistem operasi seperti Windows, Linux, dan macOS. Dengan menyediakan lingkungan server lokal yang lengkap, *XAMPP* memungkinkan pengguna menjalankan dan menguji aplikasi berbasis web tanpa harus melakukan konfigurasi server secara manual. Kombinasi fitur yang sederhana, fleksibel, dan praktis menjadikan *XAMPP* salah satu pilihan utama bagi pengembang dalam membangun, menguji, serta memelihara sistem informasi berbasis web sebelum dipublikasikan ke server sebenarnya.



Gambar 2.2. Aplikasi XAMPP

XAMPP digunakan sebagai alat pendukung utama karena menyediakan lingkungan server lokal yang memungkinkan proses pengembangan berjalan lebih cepat, terstruktur, dan bebas risiko terhadap sistem produksi yang digunakan bengkel. Ketersediaan Apache sebagai

server, *MySQL/MariaDB* sebagai pengelola basis data, serta *PHP* sebagai bahasa pemrograman memberikan kemudahan dalam membangun dan menguji setiap fitur sistem pemesanan *Service online* secara langsung. Pengembang dapat melakukan proses debugging, pengujian fungsional, serta pemantauan alur kerja aplikasi secara real time, sehingga memastikan setiap modul berfungsi dengan baik sebelum diimplementasikan sepenuhnya. Dengan memanfaatkan *XAMPP*, pembuatan sistem berbasis web menjadi lebih efisien karena semua kebutuhan teknologi telah tersedia dalam satu paket yang stabil dan mudah digunakan.

2.8. Penelitian Terdahulu

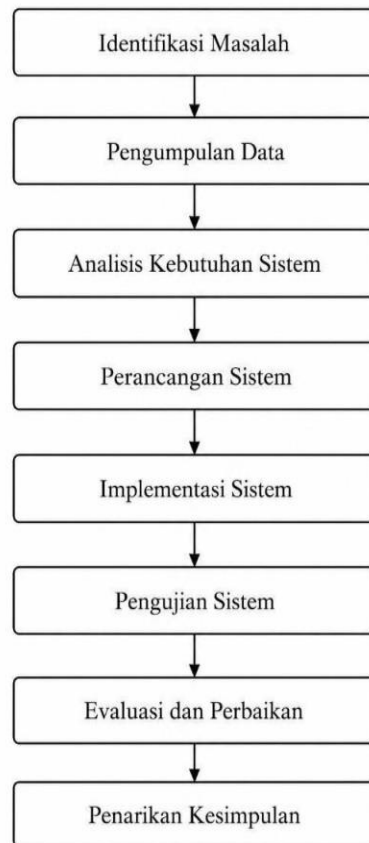
No	Penulis	Judul	Tahun	Metode	Hasil
1	Octavianus Sabi Rura1), Henri Septanto2) [21]	PERANCANGAN SISTEM INFORMASI INVENTORI STOK PRODUK KOSMETIK BERBASIS WEB MENGGUNAKAN MODEL <i>WATERFALL</i> PADA TOKO PAVLIN BEAUTY	2024	Metode <i>Waterfall</i>	Penelitian ini bertujuan membangun Sistem Informasi Inventori Stok Produk Kosmetik berbasis web untuk Toko Pavlin Beauty guna meningkatkan efisiensi, akurasi stok, dan kualitas layanan. Sistem dikembangkan menggunakan metode <i>Waterfall</i> , dimulai dari analisis kebutuhan hingga implementasi dan pemeliharaan, dengan data dikumpulkan melalui observasi dan wawancara. Hasilnya diharapkan memudahkan pengelolaan inventaris, mengurangi

					kesalahan pencatatan, serta mengoptimalkan proses bisnis toko.
2	Firmasyah Budi Utomo ^{1*} , Nungky Awang Chandra ² [22]	Pengembangan Aplikasi Stok Barang Pada Cv. Lentera Abadi Nusantara Menggunakan Metode <i>Waterfall</i>	2023	Metode <i>Waterfall</i>	CV. Lentera Abadi Nusantara saat ini masih menggunakan pencatatan stok barang secara manual, yang menyebabkan kesulitan dalam pencarian data dan risiko kesalahan pencatatan. Untuk mengatasi hal ini, diperlukan sistem informasi inventory yang mempermudah pengelolaan barang masuk dan keluar. Sistem ini dikembangkan

					menggunakan metode <i>Waterfall</i> , sehingga proses pengelolaan stok menjadi lebih efisien, akurat, dan sesuai dengan kebutuhan pengguna.
3	Rafli Fadillah Agustio a,1,*, Ahnaf Irfan Baharianto b,2, Riyan Pratama Mulia c,3, Wasis Haryono d,4 [23]	Pengembangan Sistem Informasi Persediaan Gudang Berbasis Website Dengan Metode <i>Waterfall</i>	2024	Metode <i>Waterfall</i>	Penelitian ini mengembangkan sistem web untuk pengelolaan stok barang menggunakan metode <i>Waterfall</i> , meliputi analisis kebutuhan, desain, implementasi, dan pengujian. Sistem ini memantau stok secara real-time, mengurangi kesalahan pencatatan, dan meningkatkan efisiensi serta akurasi transaksi pembelian.
4	Muhammad Faithullah Akbar [24]	Penerapan Metode <i>Waterfall</i> pada Sistem Informasi Penjualan Dan Persediaan Pada Warung Makan Hejo Karawang	2022	Metode <i>Waterfall</i>	Warung Makan Hejo, sebagai UMKM, masih menggunakan sistem manual yang memakan waktu dalam pelayanan dan pengelolaan persediaan. Untuk meningkatkan efisiensi dan kenyamanan pelanggan, penerapan sistem informasi berbasis

					website dirancang menggunakan metode <i>Waterfall</i> , sehingga memudahkan pengelolaan penjualan dan persediaan secara terkomputerisasi.
5	Supiyandi1, Muhammad Muhammad Nur Kholid Fathoni1*, kustiyono2 [25]	Pengembangan Sistem Informasi Persediaan Barang Menggunakan Model <i>Waterfall</i> untuk Meningkatkan Efisiensi Pengelolaan Stok	205	Metode <i>Waterfall</i>	Penelitian ini merancang sistem persediaan berbasis web untuk Toko Sinar Mulia menggunakan metode <i>Waterfall</i> , bertujuan menggantikan pengelolaan stok manual yang rentan kesalahan dan meningkatkan efisiensi pencatatan serta pengambilan keputusan.

2.9. Kerangka Kerja Penelitian



Identifikasi Masalah merupakan tahap awal penelitian yang dilakukan untuk mengetahui berbagai kendala yang terjadi pada proses pemesanan servis di Bengkel Otomotif Risky Painting. Pada tahap ini ditemukan bahwa proses pemesanan masih dilakukan secara manual sehingga sering menyebabkan antrean tidak teratur, kesalahan pencatatan data, serta ketidakpastian jadwal servis yang berdampak pada kualitas pelayanan pelanggan.

Pengumpulan Data merupakan tahap yang dilakukan untuk memperoleh informasi dan data yang dibutuhkan dalam penelitian. Data dikumpulkan melalui observasi langsung ke bengkel, wawancara dengan pihak terkait, serta studi pustaka dari jurnal dan referensi yang berhubungan dengan sistem pemesanan servis online berbasis web.

Analisis Kebutuhan Sistem merupakan tahap untuk menentukan kebutuhan sistem yang akan dibangun sesuai dengan kebutuhan pengguna.

Pada tahap ini dilakukan analisis terhadap fitur - fitur yang dibutuhkan seperti login, registrasi pelanggan, pemesanan servis, pengelolaan data servis, serta pengolahan data pemesanan agar sistem dapat berjalan sesuai tujuan penelitian.

Perancangan Sistem merupakan tahap pembuatan desain sistem sebelum proses pengkodean dilakukan. Perancangan dilakukan menggunakan diagram UML seperti use case diagram, activity diagram, sequence diagram, class diagram, dan deployment diagram untuk menggambarkan alur serta struktur sistem yang akan dibangun.

Implementasi Sistem merupakan tahap penerapan hasil perancangan ke dalam bentuk aplikasi menggunakan bahasa pemrograman PHP dan database MySQL. Pada tahap ini seluruh desain sistem diterjemahkan menjadi program yang dapat dijalankan sesuai dengan kebutuhan pengguna dan proses pelayanan bengkel.

Pengujian Sistem merupakan tahap untuk memastikan bahwa sistem yang dibangun dapat berjalan dengan baik dan sesuai dengan fungsi yang telah dirancang. Pengujian dilakukan pada setiap fitur sistem seperti login, pemesanan servis, pengelolaan data, dan penyimpanan database agar tidak terjadi kesalahan saat sistem digunakan.

Evaluasi dan Perbaikan merupakan tahap yang dilakukan setelah pengujian sistem selesai. Pada tahap ini dilakukan pemeriksaan terhadap kekurangan atau kesalahan yang masih ditemukan pada sistem, kemudian dilakukan perbaikan agar sistem dapat berjalan lebih optimal dan sesuai kebutuhan operasional bengkel.

Penarikan Kesimpulan merupakan tahap akhir penelitian yang dilakukan untuk menyimpulkan hasil dari pembangunan sistem pemesanan servis online. Kesimpulan diperoleh berdasarkan hasil implementasi dan pengujian sistem yang telah dilakukan sehingga dapat diketahui apakah sistem mampu membantu meningkatkan pelayanan di Bengkel Otomotif Risky Painting.