

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait dilakukan sebagai acuan, perbandingan, dan landasan dalam memosisikan penelitian ini di antara studi sebelumnya. Melalui kajian literatur, peneliti dapat memetakan kesamaan fitur, menganalisis metode, serta menemukan celah pembaruan (*research gap*). Langkah ini penting guna menghindari duplikasi karya ilmiah sekaligus memperkuat relevansi digitalisasi pada sistem administrasi operasional bengkel motor.

Penelitian terkait dilakukan sebagai acuan, perbandingan, dan landasan dalam memosisikan penelitian ini di antara studi sebelumnya. Melalui kajian literatur, peneliti dapat memetakan kesamaan fitur, menganalisis metode, serta menemukan celah pembaruan (*research gap*). Langkah ini penting guna menghindari duplikasi karya ilmiah sekaligus memperkuat relevansi digitalisasi pada sistem administrasi operasional bengkel motor.

Penelitian mengenai digitalisasi sistem pelayanan dan administrasi pada sektor jasa otomotif telah berkembang secara dinamis, memberikan landasan teoritis serta komparasi fitur yang kuat bagi pengembangan aplikasi ini. Salah satu penelitian mendasar dilakukan oleh Firdi (2023) dalam *Jurnal Ilmu Komputer dan Informatika*, yang merancang bangun sistem informasi berbasis web pada Bengkel Try Motor Racing menggunakan metode *Waterfall*. Penelitian tersebut berfokus pada penyelesaian kendala pencatatan manual yang

menyulitkan pihak manajemen dalam memberikan informasi jadwal servis dan ketersediaan suku cadang secara cepat kepada konsumen. Melalui sistem berbasis web ini, mekanisme reservasi daring dapat terstruktur dengan baik, memberikan referensi penting bagi peneliti dalam memetakan diagram *Activity* untuk fungsi *Booking*

Aspek tata kelola operasional yang lebih kompleks dalam manajemen perbengkelan dikaji oleh Bryan Rafferty dan Mulyati (2026) dalam *Klik - Jurnal Ilmu Komputer*. Mereka mengembangkan Sistem Informasi Manajemen Bengkel Sejahtera Mobil berbasis web untuk mengatasi kendala efisiensi akibat pengelolaan data pendaftaran layanan, penugasan teknisi, serta administrasi internal yang masih manual. Penelitian ini mengandalkan metode pengembangan *Waterfall* dengan pemanfaatan bahasa pemrograman *PHP* dan basis data *Mysql*. Kontribusi utama dari studi ini yang relevan dengan penelitian peneliti adalah adanya keterlibatan multi-aktor secara *real-time* (Admin, Pelanggan, dan Teknisi/Mekanik) dalam memantau status pekerjaan kendaraan yang sedang diservis

Fokus penelitian yang menitikberatkan pada kemudahan aksesibilitas konsumen dilakukan dalam studi rancang bangun aplikasi *Booking service* kendaraan. Melalui implementasi arsitektur perangkat lunak yang sistematis, penelitian tersebut menguji bagaimana platform berbasis web dapat membagi hak akses ke dalam beberapa *role* utama pengguna guna menjaga integritas transaksi. Hasil pengujian menunjukkan bahwa otomatisasi data kendaraan serta

riwayat servis yang disimpan ke dalam sistem basis data RDBMS mampu memangkas durasi waktu antrean di bengkel secara berkala. Studi tersebut memberikan gambaran bagi peneliti mengenai pembatasan ruang lingkup data operasional bengkel agar aplikasi tetap responsif ketika diakses via *browser*

Selanjutnya, optimalisasi layanan berbasis daring juga diperkuat oleh penelitian yang menerapkan sistem informasi reservasi berbasis web untuk efisiensi manajemen bengkel. Penelitian tersebut menegaskan bahwa hambatan utama pada bengkel konvensional adalah ketidakpastian jadwal mekanik dan penumpukan kendaraan pada jam-jam sibuk. Dengan menerapkan sistem informasi yang mampu mendokumentasikan setiap antrean transaksi digital, pemilik bengkel dapat mengoptimalkan produktivitas harian mekanik. Pembahasan dalam studi ini menjadi pembanding yang bernilai bagi peneliti, khususnya terkait penerapan *framework* penulisan kode (*coding framework*) yang rapi guna mempermudah pemeliharaan jangka panjang aplikasi

2.2 Landasan Teori

2.2.1 Website

Website Sistem Informasi *Booking* Servis Bengkel Motor Berbasis Web Menggunakan *Laravel* dirancang untuk memudahkan pelanggan dalam melakukan pemesanan servis secara *online* serta membantu pihak bengkel dalam memantau dan mengelola proses pelayanan servis. Melalui website ini, pelanggan dapat mengakses halaman utama untuk melakukan *Booking* servis dengan mengisi data seperti nama, nomor kontak, jenis

motor, jenis servis, serta jadwal servis yang diinginkan. Pengembangan web berbasis tim ini juga bisa dikembangkan dengan menerapkan manajemen tim yang lincah



Gambar 2. 1 Website
(sumber : <https://www.inc.com/>)

2.2.2 PHP

PHP (Hypertext PreProcessor) adalah bahasa pemrograman *server-side* yang dirancang untuk pengembangan aplikasi web dinamis. Menurut Sinlae et al. (2024) dalam *Jurnal Siber Multi Disiplin*, *PHP* banyak digunakan karena fleksibilitasnya dalam menangani berbagai kebutuhan pengembangan web serta kemampuannya dalam berintegrasi dengan berbagai sistem basis data seperti *Mysql* [12].



Gambar 2. 2 *PHP*
(sumber : <https://id.wikipedia.org/wiki/PHP>)

2.2.3 *Laravel*

Laravel merupakan *framework PHP* yang mengadopsi arsitektur *Model-View-Controller* (MVC) dan menyediakan berbagai fitur modern seperti routing, middleware, dan Eloquent ORM. Menurut Sinlae dkk., *Laravel* mempermudah pengembangan aplikasi web karena struktur kodenya yang rapi dan efisien, serta dukungan terhadap manajemen dependensi melalui Composer.



Gambar 2. 3 *Laravel*
(sumber : <https://medium.com/@anteprocess/>)

2.2.4 *Laragon*

Laragon adalah *local development environment* yang ringan dan portabel, digunakan untuk menjalankan *server lokal* dengan cepat dan efisien. Alat ini menyediakan fitur *auto-virtual host*, integrasi multi-versi

PHP, serta kemudahan dalam mengelola proyek web secara lokal tanpa konfigurasi manual yang kompleks. Sistem ini terbukti andal dalam menunjang pembuatan aplikasi berskala diler maupun bengkel mandiri [14].



Gambar 2. 4 *Laragon*
(sumber : <https://www.kreaweb.be/laragon/>)

2.2.5 *Mysql*

Mysql adalah sistem manajemen basis data relasional (RDBMS) open-source yang banyak digunakan dalam pengembangan aplikasi web., *Mysql* memiliki performa tinggi dan stabilitas yang baik, serta mendukung pengelolaan data secara efisien dalam aplikasi berbasis web.



Gambar 2. 5 *Mysql*
(sumber : <https://id.wikipedia.org/wiki/Mysql>)

2.2.6 Vscode

Visual Studio Code (VS Code) merupakan sebuah *source-code* editor yang dikembangkan oleh Microsoft dan banyak digunakan dalam pengembangan aplikasi berbasis web. VS Code mendukung berbagai bahasa pemrograman seperti *JavaScript*, *PHP*, dan *Python*, serta dilengkapi dengan fitur *extensions* yang memudahkan integrasi *framework* maupun *library* yang relevan dengan kebutuhan sistem. proses pengembangan sistem menjadi lebih efisien, terstruktur, dan mudah dalam pemeliharaan



Gambar 2. 6 Vscode
(sumber : <https://apps.microsoft.com/>)

2.2.7 UML (*Unified Modeling Language*)

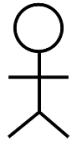
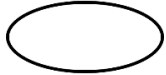
Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk membuat diagram atau model perangkat lunak. UML memungkinkan pengembang atau *developer* untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem perangkat lunak secara jelas dan terstruktur. Analogi yang sering digunakan adalah seperti seorang arsitek bangunan yang membuat denah sebagai acuan bagi

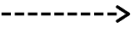
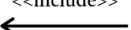
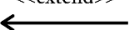
perusahaan konstruksi; demikian pula seorang arsitek perangkat lunak membuat diagram UML agar pengembang dapat membangun sistem sesuai dengan rancangan yang telah ditetapkan. Dengan memahami notasi dan kosakata UML, seseorang dapat lebih mudah memahami alur dan struktur sistem, serta menjelaskan desain perangkat lunak kepada pihak lain yang terlibat dalam pengembangan.

1. *Usecase Diagram*

Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *Use case Diagram* terdapat (6 enam aktor) yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 2. 1 *Usecase Diagram*

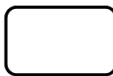
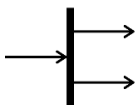
No	Gambar	Simbol	Keterangan
1.		Aktor	Mewakili peran seorang <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>Use case</i> .
2.		<i>Use case</i>	Mewakiili Abstraksi dan interaksi antara sebuah <i>System</i> dan <i>Actor</i> .

No	Gambar	Simbol	Keterangan
3.		<i>Association</i>	Abstraksi penghubung antar <i>Actor</i> dengan <i>Use case</i> .
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi <i>Actor</i> untuk dapat berpartisipasi dengan <i>Use case</i> .
5.		<i>Include</i>	Menunjukkan bahwa suatu <i>Use case</i> seluruhnya merupakan fungsionalitas dari <i>Use case</i> lainnya.
6.		<i>Extend</i>	Menunjukkan bahwa suatu <i>Use case</i> tambahan fungsional dari <i>Use case</i> lainnya.

2. Activity Diagram

Menggambarkan alur kerja dari suatu proses dan urutan aktivitas, yang dimodelkan sebagai alur kerja dari satu aktivitas ke aktivitas berikutnya atau dari satu aktivitas ke aktivitas lainnya. Di dalam *Activity Diagram* terdapat beberapa simbol yang digunakan untuk menggambarkan alur aktivitas dan proses kerja dalam sistem.

Tabel 2. 2 *Activity Diagram*

No	Gambar	Simbol	Keterangan
1.		<i>Initial node</i>	Merupakan awal aktivitas yang dibuat
2.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi
3.		<i>Decicion</i>	Menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
5.		<i>Fork Join node</i>	Digunakan untuk menunjukan kegiatan yang dilakukan secara paralel untuk dari dua kegiatan paralel jadi satu.
6.		<i>Fork node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran.

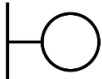
No	Gambar	Simbol	Keterangan
7.		<i>Activity Final node</i>	Merupakan akhir aktivitas yang telah dibuat.

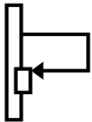
3. *Sequence Diagram*

Menggambarkan interaksi antar objek didalam dan disekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Di dalam *Sequence Diagram* terdapat beberapa simbol yang digunakan untuk alur komunikasi pesan antar objek dalam sistem.

Tabel 2. 3 *Sequence Diagram*

No	Gambar	Simbol	Keterangan
1.		<i>Entity Class</i>	Menggambarkan awal dari sebuah sistem, menjadi dasar penyusunan basisdata.
2.		<i>Boundary Class</i>	Menggambarkan komunikasi antara sistem seperti <i>Form</i> .
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>table</i> .

No	Gambar	Simbol	Keterangan
4.		<i>Recursive</i>	Mengirimkan pesan yang dikirim ke dirinya sendiri
5.		<i>Activation</i>	Menghubungkan proses durasi aktivasi sebuah proses
6.		<i>Life Line</i>	Menghubungkan tiap-tiap objek yang mana terdapat <i>activation</i>
7.		<i>Message</i>	Sebagai pengirim pesan ke tiap-tiap <i>Class</i>

4. *Class Diagram*

Mengambaran struktur dan deskripsi dari *Class*, package dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya. Di dalam *Class Diagram* terdapat beberapa simbol yang digunakan untuk hubungan dan struktur data dalam sistem.

Tabel 2. 4 *Class Diagram*

No	Gambar	Simbol	Keterangan
1.		<i>Class</i>	Menggambarkan kumpulan objek-objek dengan atribut dan operasi yang sama
2.		Nary Association	Sebagai upaya menghindari asosiasi dengan lebih dari 2 objek
3.		<i>Collaboration</i>	Urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan suatu hasil yang tertukar bagi suatu <i>Actor</i>
4.		<i>Generalization</i>	Menggambarkan dimana hubungan objek <i>descendent</i> berperilaku dan struktur data objek yang di atasnya objek induk <i>ancestor</i>

No	Gambar	Simbol	Keterangan
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandirimempengaruhi elemen yang tidak mandiri.