

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Pada penelitian Widiyanto, Deniar Angga, Ginanjar Wiro Sasmito, dan Dyah Apriliani yang berjudul “Website-Based Shoe Washing Service Information System” Journal of Robotics and Embedded System 1.1 (2025), penelitian ini membangun sebuah sistem informasi guna mempermudah proses pemesanan, pencatatan, serta pemantauan status layanan oleh *customer* dan Admin. Metode yang digunakan dalam penelitian ini adalah metode pengembangan sistem dengan membangun sistem informasi pelayanan cuci sepatu berbasis web menggunakan *framework Laravel*. Hasil implementasi menunjukkan bahwa sistem informasi ini mampu meningkatkan efisiensi dan akurasi dalam pengelolaan data layanan cuci sepatu, mempermudah *customer* dalam melakukan pemesanan serta memantau status layanan, dan membantu Admin dalam mengelola data serta transaksi secara lebih terstruktur[10].

Pada penelitian tahun (2025) yang dilakukan oleh Asroni, Dede Chandra dan Cahya Darmajati berjudul “Pengembangan Aplikasi Perawatan Sepatu Berbasis Website”, yang di publikasikan dalam *Proceeding of Informatics Collaborations and Dessimation Meeting. Vol. 1. No. 1. 2025*, Dengan sistem ini, transaksi tercatat otomatis, data *customer* dapat memeriksa status secara mandiri. Metode yang digunakan dalam penelitian

ini adalah metode *Waterfall* pada SDLC dengan analisis dan perancangan sistem menggunakan *UML*, serta pengembangan sistem menggunakan PHP (*Laravel*), *Bootstrap*, dan *MySQL*. Hasil penelitian menunjukkan bahwa sistem informasi yang dibangun mampu memudahkan proses pemesanan layanan, pencatatan transaksi secara digital, serta pemantauan status pengerjaan sepatu oleh *customer* dan pemilik usaha[11].

Pada penelitian oleh Rohaturohman, Iko Risko, Ahmad Faizin dan Tresna Yudha Prawira (2022) dalam Jurnal “Perancangan Dan Implementasi Sistem Informasi Pemesanan Sepatu Di Toko Tasik Salem Berbasis Web” menyebutkan bahwa Metode yang digunakan dalam penelitian ini adalah metode SDLC (*System Development Life Cycle*) dengan pengembangan sistem menggunakan *PHP*, *HTML*, *CSS*, dan *database MySQL*. Hasil penelitian menunjukkan bahwa sistem informasi pemesanan berbasis website yang dibuat mampu mempermudah proses pemesanan serta membantu Admin dan *customer* dalam melakukan transaksi secara lebih mudah dan terkomputerisasi[12].

2.2 Landasan Teori

2.2.1 Sistem Informasi *Booking Shoe Washing*

Sistem Informasi *Shoe Washing* merupakan sebuah sistem teknologi informasi yang dibuat untuk memudahkan Admin dalam melakukan pencatatan data *customer*, sehingga proses *booking* menjadi lebih teratur, terstruktur, dan efisien. Sistem ini berbasis

website sehingga data *customer* dapat terotomatisasi dengan tertata tanpa menggunakan metode tulis tangan lagi. Selain itu, sistem ini membantu pengelola usaha dalam mengatur jadwal pencucian, mencatat data *customer*, serta memantau status pesanan secara sistematis guna meningkatkan kualitas pelayanan dan kepuasan *customer*.

2.2.2 *MySql*

MySql ialah multiuser *database* yang menggunakan Bahasa structured query language (SQL). *MySql* juga software yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source*. *MySql* ialah *database* server yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat anda pakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada. Berikut adalah contoh pada gambar *MySql*:



Gambar 2. 1 Logo *MySql*

2.2.3 *Laravel*

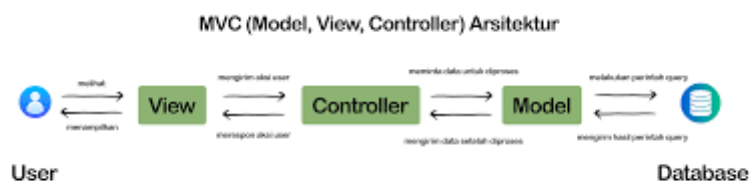
Laravel merupakan salah satu kerangka kerja pengembangan aplikasi web yang paling populer dan kuat dalam dunia PHP. Dirancang untuk memudahkan proses pengembangan aplikasi,

Laravel menawarkan beragam fitur alat yang memungkinkan para pengembang untuk menghasdikan aplikasi web yang efisien, aman, dan mudah dikelola. Dengan sintakasis yang ekspresif, dokumentasi yang kaya, dukungan komunitas yang luas, *Laravel* menjadi pilihan utama bagi banyak pengembang yang ingin mempercepat pengembangan aplikasi web mereka. Berikut adalah contoh script pada *Laravel* beserta struktur *Laravel* 12:

```
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/halo', function () {
    return "Halo Dunia Laravel";
});
```



Gambar 2. 2 Struktur *Laravel* 12

2.2.4 LARAGON

LARAGON merupakan *local development environment* yang ringan dan portabel, digunakan untuk menjalankan server lokal dengan cepat dan efisien. Menurut artikel dari Institut Teknologi dan Bisnis *PalComTech*(2024), *LARAGON* menyediakan fitur *auto virtual host*, integrasi multi-versi *PHP*, serta kemudahan dalam

mengelola proyek web secara lokal tanpa konfigurasi manual yang kompleks. Berikut adalah contoh gambar pada *LARAGON*:



Gambar 2. 3 Logo *LARAGON*

2.2.5 *Visual Studio Code*

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. Berikut adalah contoh gambar pada VS Code:



Gambar 2. 4 Logo VSCode

2.2.6 CSS (*Cascading Style Sheets*)

CSS (Cascading Style Sheets) merupakan adalah bahasa yang digunakan untuk mengatur tampilan dan desain halaman website, seperti warna, ukuran teks, tata letak, dan elemen visual lainnya agar terlihat lebih rapi dan menarik. Berikut adalah contoh gambar pada CSS:

```
body {  
    font-family: Arial;  
    background: #f5f5f5;  
}  
h1 {  
    text-align: center;  
}  
button {  
    background: blue;  
    color: white;  
    padding: 10px;  
    border: none;  
}
```



Gambar 2. 5 Logo CSS

2.2.7 UML (*Unified Modeling Language*)

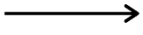
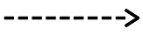
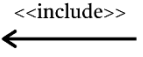
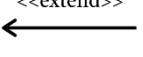
Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk membuat diagram atau model perangkat lunak. UML memungkinkan pengembang atau *developer* untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem perangkat lunak secara jelas dan terstruktur. Perancangan sistem menggunakan metode *Unified Modeling Language* (UML) untuk membantu dalam pembuatan desain dalam bentuk diagram. 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. UML membantu pemaparan (deskripsi) dan perancangan sistem, khususnya pada pemrograman berorientasi objek[13][14].

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. Use Case Diagram

Menggambarkan adalah teknik untuk menunjukkan hubungan antara pengguna sistem dengan sistem itu sendiri. Hasilnya berupa skema sederhana untuk memudahkan user membaca dan memahami informasi yang diberikan.

Tabel 2. 1 Use Case Diagram

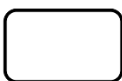
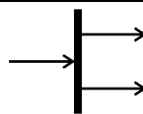
No	Gambar	Simbol	Keterangan
1.		Aktor	Mewakili peran seorang <i>system</i> yang lain, atau alat ketika berkomunikasi dengan <i>Use Case</i> .
2.		<i>Use Case</i>	Mewakili Abstraksi dan interaksi antara sebuah <i>System</i> dan <i>Actor</i> .
3.		<i>Association</i>	Abstraksi penghubung antar <i>Actor</i> dengan <i>Use Case</i> .
4.		<i>Generalisasi</i>	Menunjukkan spesialisasi actor 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.

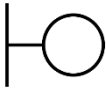
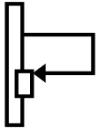
Tabel 2. 2 *Activity Diagram*

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

3. *Sequence diagram*

Di dalam *Sequence diagram* adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan atau memberi tanda komunikasi diantara objek tersebut.

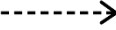
Tabel 2. 3 *Sequence diagram*

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> .
4.		<i>Recursive</i>	Mengirimkan pesan yang akan 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*

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.		Class	Menggambarkan kumpulan objek-objek dengan atribut dan operasi yang sama.
2.		Nary Association	Sebagai upaya menghindari asosiasi dengan lebih dari 2 objek.
3.		Collaboration	Urutan aksi-aksi yang ditampilkan oleh sistem yang menghasilkan suatu hasil yang tertukar bagi suatu objek.
4.		Generalization	Menggambarkan dimana hubungan objek <i>descendent</i> berperilaku dan struktur data objek yang di atasnya objek induk <i>ancestor</i> .
5.		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang tidak mandiri.