

SISTEM PENGELOLAAN LAYANAN SERVIS SMARTPHONE BERBASIS WEB PADA PHONE REPAIR

M.Addien Gilang Saputra^{1*}, Arip Solehudin², Agung Susilo Yuda Irawan³

^{1,2,3} Universitas Singaperbangsa Karawang; Jl. HS. Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; Telp. (0267) 641177

Keywords:

Sistem Pengelolaan Layanan;
Servis *Smartphone*; Laravel;
Waterfall; *Website*

Correspondent Email:

adingilang1@gmail.com



Copyright © 2026 The Author(s),
Published by Jurnal Informatika
dan Teknik Elektro Terapan. This
is an open-access article
distributed under the terms of the
Creative Commons Attribution-
NonCommercial 4.0 International
License (CC BY-NC 4.0).

Abstrak. Penelitian ini bertujuan untuk merancang dan membangun sistem pengelolaan layanan servis *smartphone* berbasis web pada PHONE REPAIR guna membantu proses *administrasi* layanan yang sebelumnya masih dilakukan secara manual menggunakan nota kertas. Permasalahan yang ditemukan meliputi risiko kehilangan data, kesulitan dalam pencarian data servis, rekap laporan, serta kurangnya transparansi informasi status perbaikan kepada pelanggan. Metode pengembangan sistem yang digunakan yaitu *Software Development Life Cycle* (SDLC) dengan model Waterfall yang terdiri dari tahapan *requirement analysis*, *system design*, *coding*, *testing*, dan *maintenance*. Perancangan sistem dilakukan menggunakan *Unified Modeling Language* (UML) berupa *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Sistem dibangun menggunakan *framework* Laravel dengan arsitektur *Model View Controller* (MVC), bahasa pemrograman PHP, *database* MySQL, serta Visual Studio Code sebagai *code editor*. Pengujian sistem dilakukan menggunakan metode *Black box Testing* dan *White box Testing* untuk memastikan fungsi dan logika program berjalan sesuai kebutuhan. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu membantu *admin* dalam pengelolaan data servis, pembaruan status perbaikan, pengelolaan *invoice*, serta pembuatan laporan secara lebih terstruktur dibandingkan proses manual. Selain itu, pelanggan dapat memantau status servis secara *real-time* melalui halaman web yang telah disediakan.

Abstract. This research aims to design and develop a web-based *smartphone* service management system at PHONE REPAIR to assist administrative processes previously conducted manually using paper notes. Problems identified include the risk of data loss, difficulties in searching service data, report recapitulation, and lack of transparency regarding repair status information for customers. The system development method used is the *Software Development Life Cycle* (SDLC) with the Waterfall model consisting of *requirement analysis*, *system design*, *coding*, *testing*, and *maintenance* stages. System design was carried out using *Unified Modeling Language* (UML), including *use case diagrams*, *activity diagrams*, *sequence diagrams*, and *class diagrams*. The system was developed using the Laravel framework with *Model View Controller* (MVC) architecture, PHP programming language, MySQL database, and Visual Studio Code as code editor. System testing was conducted using *Black box Testing* and *White box Testing* methods to ensure functions and program logic run according to requirements. Research results indicate that the developed system assists administrators in managing service data, updating repair statuses, managing invoices, and generating reports in a more structured manner compared to manual processes. In addition, customers can monitor service statuses in *real-time* through the provided web page.

1. PENDAHULUAN

Dalam perkembangan teknologi yang begitu pesat, terdapat peningkatan signifikan dalam penggunaan perangkat elektronik, khususnya *smartphone*, sebagai bagian penting dalam aktivitas sehari-hari masyarakat. Berdasarkan laporan dari Grand View Research, nilai pasar global untuk jasa perbaikan dan pemeliharaan elektronik konsumen diproyeksikan mencapai USD 25,15 miliar pada tahun 2030, dengan tingkat pertumbuhan tahunan gabungan (CAGR) sebesar 4,9% sepanjang periode 2024 hingga 2030.

Service smartphone PHONE REPAIR merupakan salah satu usaha jasa yang bergerak di bidang perbaikan dan pemeliharaan *smartphone*. Berdasarkan observasi awal, proses administrasi seperti pencatatan layanan, transaksi, dan status perbaikan masih dilakukan secara manual menggunakan nota kertas. Cara kerja konvensional ini menimbulkan beberapa kendala, seperti risiko kehilangan data, sulit dalam melakukan rekapitulasi laporan, serta kurangnya transparansi informasi kepada pelanggan [1].

Penerapan sistem pengelolaan berbasis web telah terbukti memberikan dampak positif bagi peningkatan kinerja suatu organisasi. Hasil penelitian yang dilakukan Zulhayyir (2024) menunjukkan bahwa implementasi sistem pengelolaan manajemen berbasis web telah meningkatkan efisiensi operasional melalui pengurangan kesalahan pencatatan, efisiensi waktu, dan peningkatan kecepatan pengambilan keputusan [2]. Kondisi serupa juga ditemukan dalam penelitian sebelumnya yang menunjukkan bahwa ketiadaan sistem pengelolaan layanan menyebabkan kesulitan dalam pencatatan dan pelacakan data pelanggan [3].

Penelitian ini berfokus pada pengembangan sistem pengelolaan berbasis web yang digunakan khusus oleh *admin* untuk mengelola data layanan, mencatat layanan servis yang masuk, memperbarui status servis, dan menghasilkan laporan. Selain itu, sistem menyediakan halaman bagi pelanggan untuk memantau status perbaikan perangkat secara *real-time* berdasarkan data yang diinput oleh *admin*. Rumusan masalah penelitian ini adalah: (1) Bagaimana merancang dan membangun sistem pengelolaan berbasis web yang dapat membantu *admin* dalam mengelola data layanan servis *smartphone* pada PHONE REPAIR secara terstruktur dan terintegrasi? (2) Bagaimana sistem dapat menyediakan fitur pelacakan status perbaikan secara *real-time* bagi pelanggan?

2. TINJAUAN PUSTAKA

2.1 Sistem Pengelolaan Layanan

Sistem pengelolaan layanan merupakan suatu sistem yang dirancang untuk mengatur, mencatat, dan memantau seluruh proses layanan yang diberikan kepada pelanggan secara terstruktur dan terintegrasi. Sistem ini berfungsi mengelola data layanan mulai dari penerimaan layanan, proses pengerjaan, hingga penyelesaian layanan, sehingga informasi tersimpan dengan rapi dan mudah diakses [4]. Penerapan sistem pengelolaan layanan berbasis web mampu meningkatkan efisiensi operasional melalui pengurangan kesalahan pencatatan, percepatan proses layanan, serta kemudahan dalam penyusunan laporan [2].

2.2 Software Development Life Cycle (SDLC) dan Model Waterfall

SDLC merupakan kerangka kerja yang terstruktur dan sistematis untuk diterapkan dalam pengembangan atau modifikasi perangkat lunak. SDLC berfungsi sebagai panduan lengkap dari analisis hingga pemeliharaan guna memastikan proyek pengembangan perangkat lunak bisa dijalankan secara terencana [5]. Metode Waterfall adalah salah satu model dari pengembangan SDLC yang memakai pendekatan berurutan dan terstruktur. Tahapan-tahapannya harus diselesaikan sebelum lanjut ke tahap berikutnya, meliputi *requirement analysis*, *system design*, *implementation*, *testing*, *deployment*, dan *maintenance* [5].

2.3 Unified Modeling Language (UML)

UML merupakan bahasa pemodelan standar yang digunakan untuk mendeskripsikan, merancang, dan mendokumentasikan sistem perangkat lunak secara visual. UML mendukung berbagai diagram, di antaranya *use case diagram* untuk menggambarkan kebutuhan fungsional sistem, *activity diagram* untuk memodelkan alur aktivitas, *sequence diagram* untuk menjelaskan interaksi objek berdasarkan urutan waktu, dan *class diagram* untuk menjelaskan struktur kelas beserta atribut dan hubungan antar kelas [6].

2.4 Framework Laravel dan Arsitektur MVC

Laravel adalah salah satu *framework* PHP modern yang dirancang untuk mempermudah proses pengembangan aplikasi web. *Framework* ini menyediakan sintaksis elegan dan pendekatan modular yang memungkinkan pengembang bekerja secara efisien. Arsitektur MVC (*Model View Controller*) membagi aplikasi menjadi tiga komponen: Model yang bertanggung jawab menangani logika bisnis dan interaksi basis data, View yang mempresentasikan antarmuka pengguna, dan *Controller* sebagai perantara yang mengatur alur logika aplikasi [7].

2.5 Metode Pengujian

Black box Testing adalah metode pengujian yang berfokus pada pemeriksaan fungsi atau hasil keluaran sistem tanpa mengetahui struktur atau logika program di dalamnya [8]. *White box Testing* adalah metode pengujian yang memfokuskan pada pemeriksaan bagian dalam program, khususnya pada struktur dan alur logika [9]. *User Acceptance Testing* (UAT) merupakan metode pengujian yang dilakukan oleh pengguna akhir untuk memastikan sistem telah sesuai dengan kebutuhan pengguna dan dapat digunakan dengan baik sebelum diterapkan secara penuh [10].

2.6 Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan antara lain: Febriansyah (2025) yang mengembangkan sistem pengelolaan jasa servis elektronik berbasis web menggunakan metode Waterfall, menghasilkan sistem yang meningkatkan efisiensi operasional [11]; Maulana et al. (2024) yang merancang *website* layanan jasa reparasi elektronik menggunakan *framework* Laravel dengan hasil sistem yang mampu mengurangi kesalahan pencatatan [12]; serta Lukito et al. (2022) yang mengembangkan *web-based service app* dengan hasil uji *usability* skor tinggi yang membuktikan efektivitas dalam meningkatkan kepuasan pelanggan [13].

3. METODE PENELITIAN

Penelitian ini menggunakan metode SDLC dengan model Waterfall. Objek penelitian adalah perancangan dan pembangunan sistem pengelolaan layanan servis pada *Service smartphone PHONE REPAIR* yang berlokasi di Jl. Raya Pd. Ungu Permai BLK II No. 4, Kaliabang Tengah, Kec. Bekasi Utara, Kota Bekasi. Sistem difokuskan pada pengembangan aplikasi berbasis web yang membantu *admin* mengelola data layanan servis, pencatatan transaksi, dan pembaruan status perbaikan.

3.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui: (1) Observasi langsung ke toko servis untuk mempelajari alur kerja dan proses bisnis yang berjalan; (2) Wawancara informal dengan pemilik PHONE REPAIR untuk mendapatkan informasi mendalam mengenai kebutuhan sistem; dan (3) Studi pustaka terhadap jurnal ilmiah, buku, dan penelitian terdahulu yang relevan. Hasil analisis mengidentifikasi kebutuhan fungsional utama, yaitu pencatatan data servis, pengelolaan pelanggan, pembaruan status perbaikan, manajemen *invoice*, dan pembuatan laporan.

3.2 Perancangan Sistem

Perancangan sistem mencakup perancangan antarmuka pengguna (UI Design), perancangan

diagram sistem menggunakan UML (*use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*), serta perancangan basis data menggunakan Entity Relationship Diagram (ERD). Sistem memiliki tiga aktor utama: Pelanggan yang dapat melakukan cek status servis menggunakan kode unik, *Admin* yang bertanggung jawab mengelola seluruh data servis dan operasional, serta *Owner* yang dapat memantau laporan dan statistik usaha.

3.3 Implementasi Sistem

Implementasi sistem menggunakan lingkungan pengembangan berikut: PHP 8.x dan JavaScript sebagai bahasa pemrograman; Laravel 10/11 sebagai *framework*; MySQL sebagai *database*; Laravel sebagai *web server* lokal;

Visual Studio Code sebagai *code editor*; dan Blade *Template engine* Laravel untuk desain UI. Sistem dibangun mengikuti arsitektur MVC dengan tiga model utama (*User*, *Servis*, *Setting*), beberapa *controller* (*ServisController*, *LaporanController*, *StatistikController*, *InvoiceController*, *SettingController*), serta *RoleMiddleware* untuk kontrol akses berbasis peran.

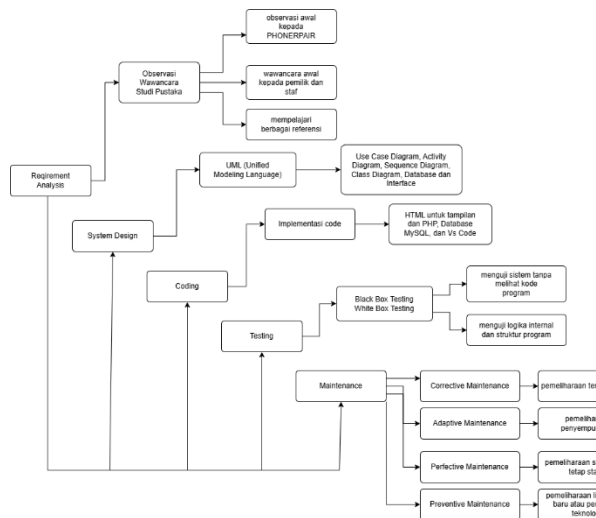
3.4 Pengujian Sistem

Pengujian dilakukan menggunakan tiga metode: (1) *Black box Testing* untuk memverifikasi fungsionalitas sistem berdasarkan input dan output tanpa memeriksa struktur internal kode; (2) *White box Testing* menggunakan teknik Basis Path *Testing* untuk memverifikasi setiap jalur eksekusi, kondisi percabangan, dan pengulangan dalam kode program; serta (3) *User Acceptance Testing* (UAT) yang dilakukan oleh pengguna akhir (*Admin* dan *Owner*) untuk memastikan sistem sesuai dengan kebutuhan pengguna.

3.5 Rancangan Penelitian

Pada gambar 3.1 rancangan penelitian metode pengembangan sistem yang digunakan adalah metode *Waterfall* yang dimulai dari tahap requirement analysis melalui observasi, wawancara, dan studi pustaka, kemudian dilanjutkan dengan tahap system design menggunakan UML seperti *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *database*, dan *interface*.

Setelah itu dilakukan tahap *coding* menggunakan *HTML*, *PHP*, *MySQL*, dan Visual Studio Code, lalu tahap *testing* menggunakan *Black box Testing* dan *White box Testing* untuk menguji fungsi serta logika program, dan tahap terakhir yaitu *maintenance* untuk memperbaiki *bug*, menyesuaikan kebutuhan sistem, serta menjaga sistem tetap berjalan dengan baik.



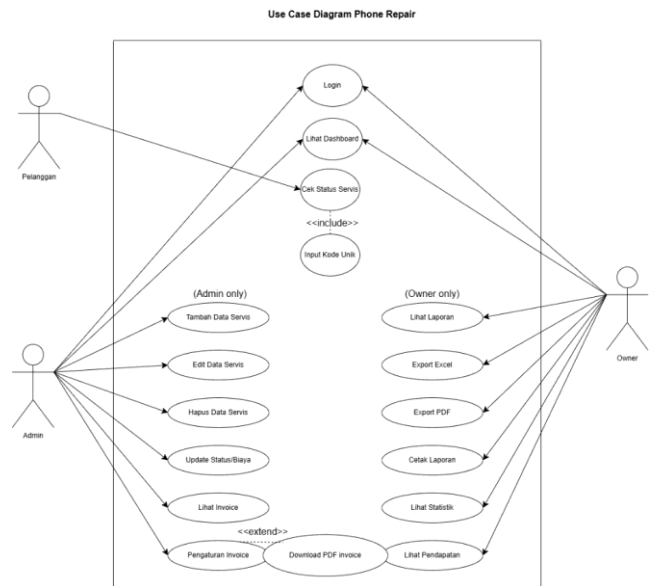
Gambar 3.1 Rancangan Penelitian

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian, sistem informasi pengelolaan layanan servis smartphone berbasis web pada PHONE REPAIR berhasil dibangun dan diimplementasikan menggunakan metode SDLC model waterfall melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian. Sistem ini dirancang untuk menggantikan proses pencatatan manual menggunakan nota kertas menjadi sistem terkomputerisasi yang lebih terstruktur sehingga dapat membantu admin dalam mengelola data pelanggan, data layanan servis, serta pembaruan status perbaikan yang tersimpan pada basis data MySQL.

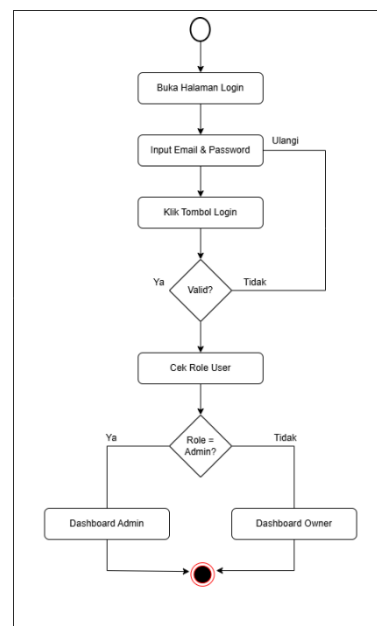
Selain itu, pelanggan juga dapat memantau status servis melalui halaman yang telah disediakan sehingga proses pelayanan menjadi lebih efektif dan terorganisir. Berdasarkan hasil implementasi, sistem yang dibangun telah sesuai dengan kebutuhan dasar operasional usaha dan mampu mengatasi permasalahan utama, seperti kesulitan pencarian data serta pencatatan yang sebelumnya kurang tertata. Meskipun demikian, sistem masih memiliki keterbatasan karena fitur yang tersedia belum mencakup pengelolaan stok sparepart maupun notifikasi otomatis kepada pelanggan sehingga pengembangan lebih lanjut masih diperlukan untuk meningkatkan fungsi dan kualitas sistem.

4.1 Perancangan Sistem (System design)



Gambar 4.1 Use case diagram

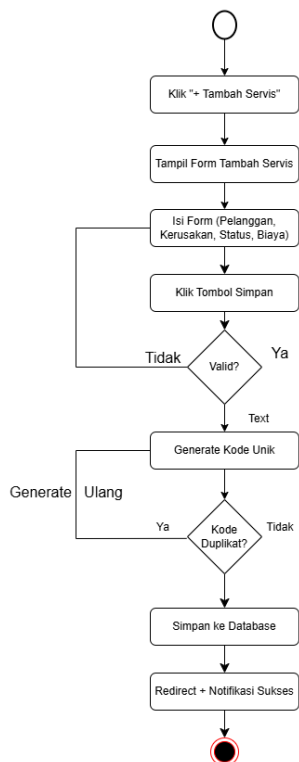
Pada gambar 4.1 merupakan perancangan sistem menggunakan UML menghasilkan beberapa diagram. *Use case diagram* menggambarkan tiga aktor utama (Pelanggan, Admin, Owner) beserta interaksinya dengan sistem. Admin memiliki akses penuh untuk mengelola data servis mulai dari tambah, edit, hapus, update status, pengelolaan invoice, hingga pengaturan sistem. Owner dapat mengakses laporan, statistik, dan pendapatan. Pelanggan hanya dapat melakukan pengecekan status servis menggunakan kode unik tanpa perlu login.



Gambar 4.2 Activity diagram Login

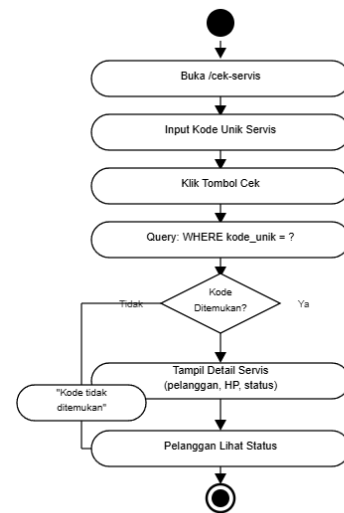
Pada Gambar 4.2 diagram ini menggambarkan alur dari *Phone Repair* dalam sistem.

Pada gambar 4.2, rangkaian aktivitas dimulai dari pengguna membuka halaman *login*, kemudian memasukkan *email* dan *password* serta menekan tombol *login*. Sistem akan melakukan validasi data, jika tidak valid maka pengguna diminta mengulangi *input*, sedangkan jika valid sistem akan mengecek *role* pengguna. Berdasarkan *role* tersebut, pengguna akan diarahkan ke *dashboard admin* atau *dashboard owner*, sehingga proses *login* dinyatakan selesai.



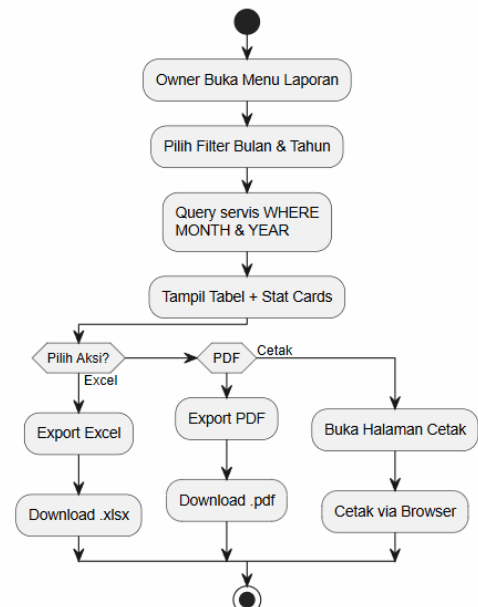
Gambar 4.3 Activity Diagram 2 - Tambah Data Servis

Pada gambar 4.3 Dalam diagram ini menggambarkan aktivitas sistem dalam menambah data servis. Pada gambar 4.3, aktivitas dimulai saat pengguna menekan tombol tambah servis dan sistem akan menampilkan *form input*. *Admin* kemudian mengisi data seperti pelanggan, kerusakan status, dan biaya, lalu menekan tombol simpan. Sistem melakukan validasi data, jika tidak valid maka *admin* akan diminta mengulangi pengisian, jika valid sistem akan menghasilkan kode unik dan memastikan tidak terjadi duplikasi.



Gambar 4.4 Aktivitas Cek Status

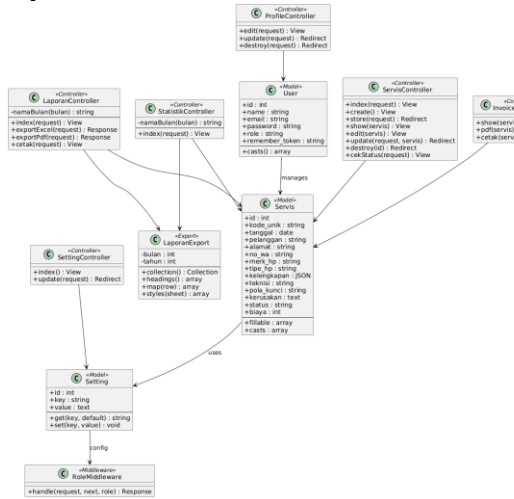
Pada Gambar 4.4, aktivitas dimulai saat pengguna membuka halaman cek servis, setelah itu memasukkan kode unik servis dan menekan tombol cek. Sistem akan melakukan pencarian data berdasarkan kode unik yang diinput oleh pengguna. Jika kode tidak ditemukan, maka sistem akan menampilkan pesan bahwa kode tidak dapat ditemukan, sedangkan jika kode ditemukan, maka sistem akan menampilkan detail servis seperti data pelanggan, perangkat, dan status servis, sehingga pengguna dapat melihat informasi status servis tersebut.



Gambar 4.5 Activity Laporan

Pada Gambar 4.5, aktivitas diawali saat *owner* membuka halaman laporan dan memilih filter data seperti bulan atau tahun, kemudian sistem akan menampilkan data sesuai kriteria yang dipilih. *Owner*

dapat melakukan aksi lanjutan seperti mencetak atau mengekspor laporan ke dalam beberapa format tertentu. Selanjutnya sistem akan memproses permintaan tersebut, dan jika berhasil maka laporan akan ditampilkan atau dapat diunduh sesuai dengan pilihan pengguna, sehingga proses pengelolaan laporan dinyatakan selesai.

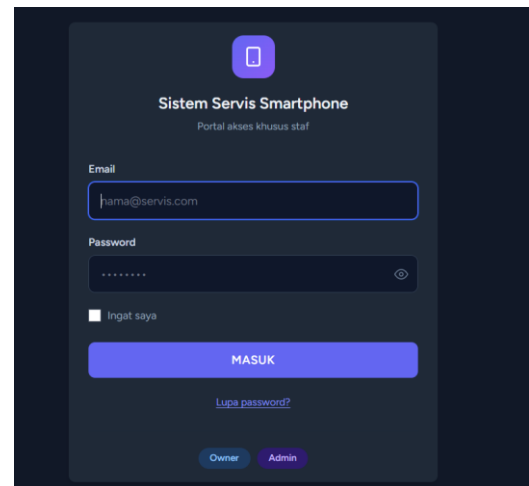


Gambar 4.6 Class diagram Phone Repair

Pada gambar 4.6 merupakan *Class diagram* yang mempresentasikan struktur sistem berbasis arsitektur MVC pada *framework* Laravel dengan tiga entitas utama: *User* (menyimpan data pengguna beserta *role*), *Servis* (mencatat seluruh data transaksi layanan servis), dan *Setting* (menyimpan konfigurasi sistem dalam format *key-value*). Relasi *one-to-many* terbentuk antara *User* dan *Servis*, di mana satu pengguna dapat menangani banyak data servis. ERD menunjukkan bahwa setiap proses servis terhubung dengan satu teknisi yang bertanggung jawab melalui atribut teknisi pada tabel servis.

4.2 Implementasi Fitur Sistem (*Coding*)

Pada gambar 4.7 merupakan hasil dari Implementasi fitur *login* menggunakan sistem autentikasi bawaan Laravel. Setelah berhasil *login*, pengguna diarahkan ke halaman yang sesuai berdasarkan *role* yang divalidasi oleh *RoleMiddleware*. Fitur tambah data servis secara otomatis membangkitkan kode unik dengan format SRV-YYYY-XXXXXX menggunakan mekanisme do-while untuk memastikan tidak terjadi duplikasi kode pada *database*. Fitur cek status memungkinkan pelanggan memantau perkembangan perbaikan perangkat tanpa perlu *login* ke sistem.



Gambar 4.7 Halaman *Login*

Pada gambar 4.8 merupakan hasil dari implementasi Fitur tambah data servis digunakan oleh *Admin* untuk mencatat layanan servis baru. Sistem secara otomatis membangkitkan kode unik dengan format SRV-YYYY-XXXXXX menggunakan mekanisme *do-while* untuk memastikan tidak terjadi duplikasi kode pada *database*.

Tambah Data Servis

Informasi Servis

Tanggal Masuk *

Selensi

Estimasi Pengiriman
* Estimasi pengiriman berdasarkan lokasi kendaraan

Tanggal Estimasi Salasil *

Durasi Estimasi (dalam hari)
 hari

Data Pelanggan

Nama Pelanggan *

Alamat

Jl. Contoh No. 1

Nomor WhatsApp

08123456789

Data Handphone

Mark HP

Type HP

Gambar 4.8 Tambah Data Servis

Pada gambar 4.9 merupakan hasil dari implementasi fitur tambah data servis yang digunakan oleh *admin* untuk melakukan edit data servis pada layanan yang berisi tanggal, bulan, tahun, terdapat Alamat dan merk hp dan pola,kata sandi untuk menyimpan setelah *admin* melakukan pengeditan pada edit data servis

[illegible]

Gambar 4.9 Edit Data Servis

Pada gambar 4.10 merupakan hasil implementasi fitur cek status servis, pada fitur ini memungkinkan pelanggan memantau perkembangan

perbaikan perangkat tanpa perlu *login* ke sistem. Pelanggan cukup memasukkan kode unik yang diberikan oleh *Admin* saat perangkat diterima.

Gambar 4.10 Cek Status Servis

Pada gambar 4.11 merupakan hasil implementasi fitur laporan memungkinkan *Owner* melihat rekapitulasi data servis berdasarkan filter bulan dan tahun, dengan kemampuan ekspor ke format Excel (menggunakan library *Maatwebsite Excel*) dan PDF (menggunakan *Barryvdh DomPDF*). Fitur *invoice* memungkinkan *Admin* mencetak atau mengunduh bukti transaksi servis dalam format PDF berukuran 58mm untuk *thermal printer*. Konten *footer invoice* bersifat dinamis dan dapat dikonfigurasi melalui halaman pengaturan menggunakan Model Setting.

Gambar 4.11 Tampilan Halaman Laporan

Pada gambar 4.12 merupakan hasil implementasi dari Fitur *invoice* memungkinkan *Admin* mencetak atau mengunduh bukti transaksi servis dalam format PDF berukuran 58mm (*thermal printer*). Konten *footer invoice* bersifat dinamis dan dapat dikonfigurasi melalui halaman Pengaturan.

Gambar 4.12 Tampilan Halaman Invoice

Pada gambar 4.13 merupakan hasil dari implementasi fitur *invoice* servis pada sistem *PHONE REPAIR* yang berisi data pelanggan, data handphone, jenis kerusakan, status servis, rincian biaya, dan total pembayaran sebagai bukti transaksi layanan servis *smartphone* berbasis web.

Gambar 4.13 Tampilan Halaman Invoice Struk

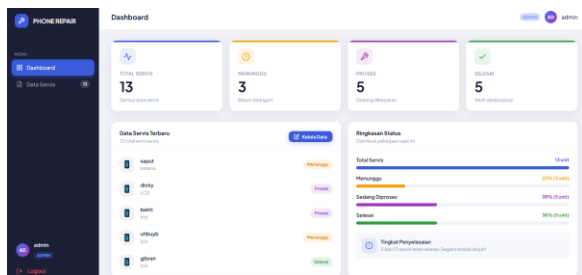
Pada gambar 4.14 merupakan hasil dari implementasi fitur pengaturan *invoice*. Gambar tersebut menampilkan halaman data servis pada sistem *PHONE REPAIR* yang digunakan untuk mengelola informasi layanan servis *smartphone*. Halaman ini berisi daftar data pelanggan, kode servis, nomor WhatsApp, jenis handphone, teknisi, status

pengerjaan, serta aksi untuk melihat detail, mencetak *invoice*, mengedit, dan menghapus data servis.

TANGGAL	KODE SERVIS	PELANGGAN	NO. WA	MERK / TYPE	TEKNIK	STATUS	AKSI
10/04/2024	SRV-2024-NT2Y3E	raput Jember	0819400885	Poco M6 Pro	A servis	Menunggu	Detail Invoice Edit Hapus
10/04/2024	SRV-2024-SK3K5K	diiky Jember	081020072	Iphone 12	A servis	Proses	Detail Invoice Edit Hapus
10/04/2024	SRV-2024-ND9M9J	bakti	081020032	samsung s23	A servis	Proses	Detail Invoice Edit Hapus
04/03/2024	SRV-2024-FAT5Y7	ufhuyp Jember	0819644878	Samsung s23	A servis	Menunggu	Detail Invoice Edit Hapus
04/03/2024	SRV-2024-ZMLC19	gitran	0810764839	Samsung s23	A servis	Selesai	Detail Invoice Edit Hapus

Gambar 4.14 Halaman Pengaturan Invoice

Pada gambar 4.15 merupakan hasil dari implementasi menampilkan halaman *dashboard Admin* dan *Owner* pada sistem *PHONE REPAIR* yang digunakan untuk memantau data layanan servis *smartphone*. *Dashboard* menampilkan jumlah total servis, status pengerjaan servis, data servis terbaru, serta ringkasan progres pekerjaan sehingga memudahkan *admin* dalam mengelola dan memonitor proses servis secara *real-time*.



Gambar 4.15 Halaman Dashboard Admin

4.3 Hasil Pengujian Black Box Testing

Pengujian Black box dilakukan terhadap delapan fitur utama sistem dengan total 49 kasus uji. Seluruh pengujian berhasil sesuai harapan dengan tingkat keberhasilan 100%.

Tabel 1. Ringkasan Hasil Pengujian Black Box Testing

Fitur yang Diuji	Jumlah Kasus	Valid	Tidak Valid	Persentase	Ket
Login	7	7	0	100%	Sangat Baik
Tambah Data Servis	8	8	0	100%	Sangat Baik
Edit & Hapus Servis	7	7	0	100%	Sangat Baik
Cek Status Servis	5	5	0	100%	Sangat Baik
Laporan & Export	7	7	0	100%	Sangat Baik
Invoice	5	5	0	100%	Sangat Baik
Pengaturan Invoice	5	5	0	100%	Sangat Baik
Statistik	5	5	0	100%	Sangat Baik
8 fitur	49	49	0	100%	Sangat Baik

4.4 Hasil Pengujian White Box Testing

Pengujian White box menggunakan teknik Basis Path Testing dilakukan terhadap enam fungsi kritis sistem dengan total 24 jalur eksekusi. Seluruh jalur eksekusi menghasilkan output yang sesuai, menghasilkan tingkat keberhasilan 100%. Fungsi-fungsi yang diuji meliputi *store()* dan *update()* pada *ServisController*, *cekStatus()* pada *ServisController*, *index()* pada *LaporanController*, *handle()* pada *RoleMiddleware*, serta *get()* dan *set()* pada *Model Setting*.

Tabel 2. Ringkasan Hasil Pengujian White Box Testing

Fitur yang Diuji	Jumlah Kasus	Valid	Tidak Valid	Persentase	Ket
<i>store()</i>	<i>ServisController</i>	5	5	100%	Sangat Baik
<i>update()</i>	<i>ServisController</i>	4	4	100%	Sangat Baik
<i>cekStatus()</i>	<i>ServisController</i>	3	3	100%	Sangat Baik
<i>index()</i>	<i>LaporanController</i>	4	4	100%	Sangat Baik
<i>handle()</i>	<i>RoleMiddleware</i>	4	4	100%	Sangat Baik
<i>get() & set()</i>	<i>Model Setting</i>	4	4	100%	Sangat Baik
6 Fungsi	-	24	24	0	Sangat Baik

4.5 Hasil Pengujian User Acceptance Testing (UAT)

Pengujian UAT dilakukan oleh dua jenis pengguna (*Admin* dan *Owner*) dengan total 33 kasus uji yang mencakup 8 modul fungsional. Dari keseluruhan kasus uji, sebanyak 30 kasus dinyatakan lulus dan 3 kasus tidak lulus. Tabel 3 menunjukkan persentase keberhasilan UAT per fitur.

Tabel 3. Perhitungan Persentase Keberhasilan UAT

Fitur	Total Kasus	Lulus	Tidak Lulus	Rumus	Hasil (%)	Ket
Autentikasi	4	4	0	$(4/4) \times 100$	100%	Sangat Baik
Servis	10	9	1	$(9/10) \times 100$	90%	Baik
Invoice	3	2	1	$(2/3) \times 100$	66.67%	Cukup
Laporan	6	6	0	$(6/6) \times 100$	100%	Sangat Baik
Statistik	3	3	0	$(3/3) \times 100$	100%	Sangat Baik
Setting	2	2	0	$(2/2) \times 100$	100%	Sangat Baik
Profil	2	1	1	$(1/2) \times 100$	50%	Kurang
Cek Status	3	3	0	$(3/3) \times 100$	100%	Sangat Baik
Total	33	30	3	$(30/33) \times 100$	90.91%	Sangat Baik

Nilai 90,91% berada pada rentang 85%–100% sehingga termasuk dalam kategori "Sangat Baik" berdasarkan skala interpretasi yang digunakan. Terdapat tiga kasus uji yang tidak lulus, yaitu: (1) UAT-14 terkait pembatasan akses setting untuk *role owner* yang belum berfungsi sepenuhnya; (2) UAT-17 terkait format cetak *invoice* yang belum sesuai; dan (3) UAT-29 terkait fitur pembaruan profil pengguna yang masih mengalami kendala.

4.6 Pemeliharaan Sistem (*Maintenance*)

Tahap *maintenance* meliputi empat jenis pemeliharaan: (1) *Corrective maintenance*: perbaikan bug pada *query* laporan yang semula menggunakan `orderBy('created_at','asc')` sehingga data terbaru tidak muncul di atas, diperbaiki dengan mengubah menjadi `'desc'`; (2) *Adaptive maintenance*: penyesuaian cache pada lingkungan *hosting* dengan menambahkan fungsi `clearCache` dan pembersihan cache menggunakan perintah PHP artisan `config:clear`; (3) *Perfective maintenance*: peningkatan tampilan antarmuka halaman *login* dengan desain lebih modern, penambahan logo, dan penyesuaian tata letak; serta (4) *Preventive maintenance*: penerapan *role-based access control* melalui *RoleMiddleware* untuk membatasi akses pengguna sesuai perannya.

4.7 Pembahasan

Sistem yang dikembangkan berhasil menggantikan proses pencatatan manual yang sebelumnya menggunakan nota kertas menjadi sistem terkomputerisasi yang lebih terstruktur. Hasil ini sejalan dengan penelitian Maulana et al. (2024) yang menunjukkan bahwa sistem pengelolaan layanan berbasis web mampu mengurangi kesalahan pencatatan dan meningkatkan kemudahan akses informasi [12]. Penggunaan kode unik dengan format SRV-YYYY-XXXXXX terbukti efektif sebagai solusi pelacakan data layanan tanpa memerlukan autentikasi dari pelanggan.

Hasil pengujian Black box (100%) dan White box (100%) membuktikan bahwa secara fungsional dan logika internal, sistem telah berjalan sesuai dengan yang dirancang. Tingkat keberhasilan UAT sebesar 90,91% menunjukkan bahwa sistem telah memenuhi sebagian besar kebutuhan pengguna. Ketiga kasus uji yang tidak lulus telah diidentifikasi penyebabnya dan diberikan rekomendasi perbaikan. Sistem masih memiliki beberapa keterbatasan, seperti belum adanya mekanisme *backup* data otomatis, belum mendukung notifikasi otomatis kepada pelanggan, dan belum terintegrasi dengan sistem pembayaran digital.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan sebagai berikut: (1) Sistem pengelolaan layanan servis *smartphone* berbasis web berhasil dirancang dan dibangun menggunakan *framework* Laravel dengan arsitektur MVC serta *database* MySQL, mampu membantu *admin* mengelola data layanan servis secara lebih terstruktur dan terintegrasi. (2) Sistem menyediakan fitur pelacakan status servis secara *real-time* bagi pelanggan melalui kode unik, meningkatkan transparansi informasi layanan tanpa mengharuskan pelanggan datang langsung ke lokasi. (3) Pengujian *Black box Testing* terhadap 49 kasus uji dan *White box Testing* terhadap 24 jalur eksekusi menghasilkan tingkat keberhasilan 100%, sedangkan UAT memperoleh tingkat keberhasilan 90,91% (kategori Sangat Baik). (4) Sistem dinyatakan layak diimplementasikan dengan catatan perbaikan pada tiga poin yang tidak lulus dalam UAT.

Saran untuk pengembangan selanjutnya: penambahan fitur notifikasi otomatis melalui email atau WhatsApp; integrasi pembayaran online; penambahan fitur manajemen inventori stok suku cadang; *deployment* ke *server hosting* untuk akses yang lebih luas; serta pengujian *usability* lebih lanjut untuk mengetahui tingkat kepuasan pengguna.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Arip Solehudin, M.Kom. dan Bapak Agung Susilo Yuda Irawan, M.Kom. selaku dosen pembimbing yang telah memberikan arahan, ilmu, dan bimbingan selama penelitian ini berlangsung. Ucapan terima kasih juga disampaikan kepada Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang serta pihak PHONE REPAIR yang telah bersedia menjadi objek penelitian.

DAFTAR PUSTAKA

- [1] E. Putri Hardi Ningtyas and A. Nur Cholis, "Rancang Bangun Sistem Informasi Penjualan HP Dan Aksesoris Berbasis *Website*," JTMEI, vol. 2, no. 1, 2023.
- [2] Zulhayyir, "Sistem Informasi Pengelolaan Manajemen Berbasis Web," IJAPPI, vol. 2, no. 1, 2024.
- [3] M. Purnomo, Nurmalitasari, and Nurchim, "Digitalisasi Proses Bisnis Berbasis Web," Jurnal Teknologi Informasi, 2024.
- [4] E. Putri Hardi Ningtyas and A. Nur Cholis, "Rancang Bangun Sistem Informasi Penjualan HP Dan Aksesoris Berbasis *Website*," JTMEI, vol. 2, no. 1, 2023.
- [5] W. Erawati, S. Heristian, R. A. Purnama, "Rancang Bangun Sistem Informasi Akademik Berbasis *Website* Dengan Metode SDLC," CO-SCIENCE, vol. 3, no. 2, 2023.

- [6] A. G. Kamal and T. Wahyono, "Rancang Bangun Aplikasi E-Learning Perguruan Tinggi dengan Pemodelan Berbasis UML," IT-EXPLORE, 2024.
- [7] L. Rahmawati and S. Sumarsono, "Desain Pengembangan *Website* dengan Arsitektur *Model View Controller* pada *Framework* Laravel," JTEKSIS, vol. 6, no. 4, pp. 785-790, 2024.
- [8] R. A. A. Pratama et al., "Penerapan Metode Black box dalam Pengujian Aplikasi Informasi Stok Barang Berbasis Web," Jurnal Teknologi Informatika dan Komputer, vol. 9, no. 1, pp. 174-183, 2023.
- [9] A. Setiawan et al., "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web di SMAN 1 Gunung Sindur Dengan *White box Testing*," Scientia Sacra, vol. 2, no. 1, 2022.
- [10] B. D. A. Andi Leo, "*User Acceptance Testing* Sistem Informasi," Poters, 2025.
- [11] R. Febriansyah, "Perancangan Sistem Informasi Layanan Servis Elektronik Rumah Tangga Berbasis Web," Sosial dan Bisnis, vol. 3, no. 1, pp. 21-32, 2025.
- [12] A. Maulana, I. Purnamasari, and I. Maulana, "Rancang Bangun *Website* Layanan Jasa Reparasi Alat Elektronik Rumah Tangga Menggunakan *Framework* Laravel (Studi Kasus: CV. XYZ)," JITET, vol. 12, no. 3, 2024.
- [13] A. Lukito, J. Wiratama, and H. Santoso, "Development of *Web-based ServiceApp* Applications and *Usability Testing* Using the *Usability* Scale System (SUS) Case Study: PT Telemedia Mitra Erajaya," Computatio, vol. 6, no. 2, 2022.
- [14] D. Matthew et al., "Perancangan Sistem Manajemen Servis Kendaraan Berbasis Web pada PT XYZ," INTECOMS, vol. 8, no. 6, 2025.
- [15] F. Agustian and A. Yuliana, "Aplikasi Chatbot Pelayanan Publik Berbasis *Website*," JITET, vol. 12, no. 3S1, 2024.
<https://doi.org/10.23960/jitet.v12i3s1.5202>