PENGEMBANGAN SISTEM INFORMASI MANAJEMEN LAYANAN BENGKEL STUDI KASUS KRGARAGE



Disusun Oleh:

Nama : Daffa Arya Kesuma

NIM : 21523010

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN LAYANAN BENGKEL STUDI KASUS KRGARAGE

TUGAS AKHIR

Disusun Oleh:

Nama : Daffa Arya Kesuma

NIM : 21523010

Yogyakarta, 2 Juli 2026

Pembimbing,



(Dr. Novi Setiani, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN LAYANAN BENGKEL STUDI KASUS KRGARAGE

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Juli 2026

Tim Penguji

Dr. Novi Setiani S.T., M.T. _____

Anggota 1

Dr. Syarif Hidayat, S.Kom., M.I.T. _____

Anggota 2

Sri Mulyati, S.Kom., M.Kom. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Daffa Arya Kesuma

NIM : 21523010

Tugas akhir dengan judul:

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN LAYANAN BENGKEL STUDI KASUS KRGARAGE

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 2 Juli 2026



(Daffa Arya Kesuma)

HALAMAN PERSEMBAHAN

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, karunia, serta ridho-Nya sehingga penyusunan tugas akhir ini dapat diselesaikan. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga, sahabat, dan seluruh umatnya hingga akhir zaman.

Tugas akhir ini penulis persembahkan dengan sepenuh hati:

Untuk Mama, Abang-abang, dan Adikku tercinta

Terima kasih atas doa, kasih sayang, dukungan, dan semangat yang tidak pernah berhenti diberikan. Segala proses yang penulis lalui hingga sampai pada tahap ini tidak terlepas dari kehadiran dan pengorbanan kalian.

Untuk Almarhum Papa tercinta

Terima kasih atas cinta, nasihat, dan teladan hidup yang akan selalu penulis kenang. Semoga pencapaian ini menjadi bentuk bakti dan kebanggaan serta menjadi doa terbaik untuk Papa.

Untuk sahabat-sahabat seperjuangan

Terima kasih atas kebersamaan, bantuan, semangat, dan dukungan yang diberikan selama menjalani masa studi. Setiap proses, tantangan, dan pengalaman yang dilalui bersama menjadi bagian berharga dalam perjalanan ini.

Untuk diri sendiri

Terima kasih telah berusaha, bertahan, dan terus melangkah hingga tugas akhir ini dapat diselesaikan.

Untuk tanah kelahiranku, Pekanbaru, Kota Bertuah

Terima kasih telah menjadi tempat penulis tumbuh dan belajar. Dari tanah ini, penulis membawa kenangan, doa, serta semangat untuk terus melangkah hingga sampai pada titik ini.

Semoga karya ini dapat menjadi bentuk rasa syukur, kebanggaan, dan manfaat bagi semua pihak yang telah menjadi bagian dari proses penyusunan tugas akhir ini.

HALAMAN MOTO

Semua kemenangan diawali rasa takut

Kamu pejuang, menangkan panggungmu

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Segala puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "Pengembangan Sistem Informasi Manajemen Layanan Bengkel Studi Kasus KRGarage". Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penyusunan laporan Tugas Akhir ini, penulis menyadari bahwa terdapat berbagai tantangan dan kendala yang dihadapi. Namun, berkat pertolongan Allah SWT serta dukungan, bimbingan, doa, dan bantuan dari berbagai pihak, laporan Tugas Akhir ini dapat diselesaikan dengan baik. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Almarhum Papa, Mama, kedua abang, dan adik penulis yang senantiasa menjadi sumber doa, dukungan, semangat, kasih sayang, serta motivasi yang sangat berarti bagi penulis dalam menyelesaikan laporan Tugas Akhir ini.
2. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN Eng. selaku Rektor Universitas Islam Indonesia.
3. Ibu Prof. Dr. Sri Kusumadewi, S.Si., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Program Sarjana Universitas Islam Indonesia.
6. Ibu Dr. Novi Setiani, S.T., M.T. selaku Dosen Pembimbing Tugas Akhir yang telah dengan sabar memberikan bimbingan, arahan, masukan, serta dukungan selama proses pengerjaan hingga penyelesaian laporan Tugas Akhir ini.
7. Seluruh Bapak dan Ibu Dosen Program Studi Informatika yang telah memberikan ilmu pengetahuan, pengalaman, dan wawasan yang sangat bermanfaat selama masa perkuliahan.
8. Pihak KRGarage yang telah bersedia menjadi objek penelitian serta memberikan informasi, masukan, dan kesempatan kepada penulis dalam proses analisis kebutuhan, pengembangan, dan pengujian sistem.

9. Sahabat-sahabat Kedom yang telah memberikan dukungan, semangat, kebersamaan, serta menjadi tempat berbagi cerita dan melepas penat selama penulis menjalani masa perkuliahan dan proses penyusunan laporan Tugas Akhir ini.
10. Sahabat-sahabat Kost Putra Kidung yang telah memberikan dukungan, bantuan, semangat, dan kebersamaan yang berarti bagi penulis selama masa studi.
11. Sahabat dan rekan-rekan seperjuangan Zenith Informatika 21 yang telah memberikan dukungan, semangat, bantuan, dan kenangan berharga selama masa studi.
12. Mahliga Coffee and Space yang telah menjadi tempat penulis mengerjakan dan menyelesaikan laporan Tugas Akhir ini.
13. Semua pihak yang tidak dapat penulis sebutkan satu per satu, terima kasih atas segala bantuan, doa, dan dukungan yang telah diberikan.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki keterbatasan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang. Semoga laporan Tugas Akhir ini dapat memberikan manfaat bagi para pembaca, pihak KRGarage, serta menjadi referensi bagi penelitian selanjutnya dalam bidang pengembangan sistem informasi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 2 Juli 2026



(Daffa Arya Kesuma)

SARI

KRGarage merupakan bengkel spesialis perawatan dan restorasi mesin Vespa 2-tak yang dalam kegiatan operasionalnya masih mengandalkan pencatatan manual. Hal ini menimbulkan berbagai kendala, mulai dari lambatnya respons admin akibat pencatatan jadwal di kalender ponsel, tidak akuratnya rekapitulasi ketersediaan suku cadang, hingga masalah keterbatasan akses informasi manajerial bagi pemilik terkait laporan keuangan bengkel. Penelitian ini bertujuan untuk merancang dan membangun Sistem Informasi Manajemen Layanan Bengkel berbasis *website* untuk mendigitalkan proses utama layanan bengkel di KRGarage, mulai dari pengelolaan pemesanan hingga pelaporan pendapatan.

Pengembangan sistem dalam penelitian ini menerapkan metode *Prototyping*, yang memungkinkan kolaborasi secara iteratif antara pengembang dan pengguna untuk memastikan kesesuaian spesifikasi fitur dengan kebutuhan di lapangan. Perangkat lunak ini dibangun menggunakan *framework* Vue.js pada sisi *frontend* dan Laravel pada sisi *backend*, dengan antarmuka *web* yang responsif agar dapat diakses melalui perangkat *desktop* maupun *mobile*. Fitur utama yang diimplementasikan meliputi manajemen pemesanan servis *online*, antarmuka khusus mekanik, kalkulasi total biaya secara otomatis, peringatan batas stok kritis, pengingat servis berkala pada *dashboard* pelanggan, dan *dashboard* pemilik untuk pemantauan pendapatan, pengeluaran, dan keuntungan secara *real-time*.

Hasil pengujian fungsional sistem menggunakan metode *Black Box Testing* menunjukkan bahwa fitur utama sistem telah berjalan sesuai dengan luaran yang diharapkan. Pada pengujian awal melalui API, ditemukan beberapa ketidaksesuaian validasi *input* pada proses registrasi dan pengelolaan data Vespa. Temuan tersebut kemudian diperbaiki melalui penyesuaian validasi *backend*, sehingga hasil pengujian ulang menunjukkan bahwa sistem telah mampu menolak *input* yang tidak valid sesuai ketentuan. Selanjutnya, berdasarkan hasil evaluasi *User Acceptance Testing* (UAT) yang melibatkan pihak pemilik, admin, mekanik, dan pelanggan, diperoleh persentase kelayakan sebesar 93,80% yang berada pada kategori Sangat Layak.

Kata kunci: Sistem Informasi, Manajemen Bengkel, *Prototyping*, Laravel, Vue.js, UAT.

GLOSARIUM

Backend	Bagian sistem yang berjalan di sisi server dan berfungsi untuk mengelola logika bisnis, autentikasi, pengolahan data, serta komunikasi dengan basis data.
Black Box Testing	Metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem berdasarkan *input* dan *output* tanpa melihat struktur kode program.
Dashboard	Halaman ringkasan yang menampilkan informasi penting secara cepat, seperti statistik pemesanan, pendapatan, stok, atau aktivitas terbaru.
Database	Kumpulan data yang disimpan secara terstruktur agar dapat diakses, dikelola, dan digunakan oleh sistem.
Frontend	Bagian sistem yang berinteraksi langsung dengan pengguna melalui tampilan antarmuka pada *browser*.
Framework	Kerangka kerja perangkat lunak yang menyediakan struktur dan fitur dasar untuk mempermudah proses pengembangan aplikasi.
Prototyping	Metode pengembangan perangkat lunak yang menggunakan rancangan awal atau purwarupa sistem untuk memperoleh umpan balik dari pengguna sebelum sistem dikembangkan secara penuh.
UAT	*User Acceptance Testing*, yaitu metode evaluasi penerimaan pengguna untuk mengetahui apakah sistem telah sesuai dengan kebutuhan pengguna.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiv
DAFTAR GAMBAR xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 4
1.6 Metodologi Penelitian 5
1.7 Struktur Penelitian 6
BAB II KAJIAN PUSTAKA 7
2.1 Tinjauan Penelitian Terdahulu 7
2.2 Perbandingan Metodologi Pengembangan 9
2.3 Landasan Teori 10
2.3.1 Sistem Informasi Manajemen Bengkel 10
2.3.2 Aktor 12
2.3.3 Proses Bisnis 13
2.3.4 Metode *Prototyping* 14
2.3.5 *Framework* Laravel 15
2.3.6 *Framework* Vue.js 15
2.3.7 *Database* MySQL 16
2.3.8 *Black Box Testing* 16

2.3.9 *User Acceptance Testing* 17
BAB III METODE PENELITIAN 18
3.1 Analisis Kebutuhan 18
3.2 Perancangan Prototipe 20
3.3 Evaluasi Prototipe 21
3.4 Implementasi Sistem 21
3.5 Pengujian Sistem 22
3.6 Evaluasi Sistem 24
3.7 Penerapan Sistem 24
BAB IV HASIL DAN PEMBAHASAN 26
4.1 Hasil Analisis Kebutuhan 26
4.1.1 Kebutuhan Pihak Bengkel 26
4.1.2 Kebutuhan Pelanggan 27
4.2 Perancangan Prototipe 28
4.2.1 Prototipe Pelanggan 28
4.2.2 Prototipe Admin 32
4.2.3 Prototipe Mekanik 35
4.2.4 Prototipe Pemilik 36
4.3 Evaluasi Prototipe dan Iterasi Desain 38
4.4 Perancangan Sistem 40
4.4.1 *Use Case Diagram* 40
4.4.2 *Activity Diagram* 42
4.4.3 *Entity Relationship Diagram* 51
4.5 Implementasi Sistem 53
4.5.1 Implementasi RESTful API 54
4.5.2 Implementasi Antarmuka Pelanggan 58
4.5.3 Implementasi Antarmuka Admin 66
4.5.4 Implementasi Antarmuka Mekanik 72
4.5.5 Implementasi Antarmuka Pemilik 75
4.6 Pengujian Sistem 79
4.7 Evaluasi Sistem 96
4.7.1 Karakteristik Responden 96
4.7.2 Hasil Pengujian Kuesioner UAT 97
4.7.3 Analisis dan Perhitungan Indeks Persentase 100

4.7.4 Interpretasi Persentase Kelayakan Sistem 101
4.7.5 Kesimpulan Evaluasi Sistem 102
4.8 Penerapan Awal Sistem 102
4.9 Pembahasan 103
BAB V KESIMPULAN DAN SARAN 105
5.1 Kesimpulan 105
5.2 Saran 105
DAFTAR PUSTAKA 107
LAMPIRAN 109

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu 7
Tabel 2.2 Perbandingan Metodologi Pengembangan Sistem 9
Tabel 3.1 Rekapitulasi Hasil Wawancara dan Analisis Kebutuhan 19
Tabel 4.1 Hasil Analisis Kebutuhan Fungsional Pihak Bengkel 26
Tabel 4.2 Hasil Analisis Kebutuhan Fungsional Pelanggan 27
Tabel 4.3 Matriks Evaluasi Prototipe dan Bukti Pemenuhan Implementasi 39
Tabel 4.4 Daftar API Autentikasi dan Akses Publik 54
Tabel 4.5 Daftar API Pengguna Terautentikasi (Umum) 54
Tabel 4.6 Daftar API Fitur Pelanggan 54
Tabel 4.7 Daftar API Fitur Admin 55
Tabel 4.8 Daftar API Fitur Mekanik 56
Tabel 4.9 Daftar API Fitur Pemilik Bengkel 57
Tabel 4.10 Hasil Pengujian Registrasi, *Login*, Profil, dan Hak Akses 80
Tabel 4.11 Hasil Pengujian Fitur Pelanggan 86
Tabel 4.12 Hasil Pengujian Fitur Mekanik 91
Tabel 4.13 Hasil Pengujian Fitur Admin 92
Tabel 4.14 Hasil Pengujian Fitur Pemilik 95
Tabel 4.15 Distribusi Responden Pengujian UAT 96
Tabel 4.16 Rekapitulasi Skor Variabel Fungsionalitas Sistem 97
Tabel 4.17 Rekapitulasi Skor Variabel Kinerja Sistem 98
Tabel 4.18 Rekapitulasi Skor Variabel Tampilan Antarmuka Sistem 99
Tabel 4.19 Rekapitulasi Skor Variabel Efisiensi & Produktivitas 99
Tabel 4.20 Rekapitulasi Hasil UAT 101
Tabel 4.21 Interpretasi Persentase Kelayakan Sistem 101

DAFTAR GAMBAR

Gambar 3.1 Alur Metode *Prototyping* 18
Gambar 4.1 Halaman Pemesanan Servis Pelanggan 29
Gambar 4.2 Halaman Vespa Pelanggan 30
Gambar 4.3 Halaman Riwayat Servis Pelanggan 31
Gambar 4.4 Tampilan Pengingat Servis Pelanggan 32
Gambar 4.5 Halaman Kelola Pemesanan Servis Admin 33
Gambar 4.6 Halaman Kelola Layanan Admin 34
Gambar 4.7 Halaman Kelola Suku Cadang Admin 35
Gambar 4.8 Halaman Daftar Pekerjaan Mekanik 36
Gambar 4.9 Halaman *Dashboard* Pemilik 37
Gambar 4.10 Halaman Laporan Keuangan Pemilik 38
Gambar 4.11 *Use Case Diagram* Sistem KRGarage 41
Gambar 4.12 *Activity Diagram* Pemesanan Servis 43
Gambar 4.13 *Activity Diagram* Kelola Pemesanan 45
Gambar 4.14 *Activity Diagram* Laporan Keuangan 47
Gambar 4.15 *Activity Diagram* Tambah atau Edit Suku Cadang 48
Gambar 4.16 *Activity Diagram Restock* Suku Cadang 49
Gambar 4.17 *Activity Diagram* Hapus Suku Cadang 50
Gambar 4.18 *Entity Relationship Diagram* Sistem KRGarage 51
Gambar 4.19 Antarmuka Halaman Beranda 58
Gambar 4.20 Antarmuka Modal Masuk atau Daftar dengan Google 59
Gambar 4.21 Antarmuka Halaman *Dashboard* Pelanggan 60
Gambar 4.22 Antarmuka Pilih Layanan 61
Gambar 4.23 Antarmuka Pilih Vespa 62
Gambar 4.24 Antarmuka Pilih Jadwal 63
Gambar 4.25 Validasi Ketersediaan Jadwal Servis 64
Gambar 4.26 Antarmuka Halaman Vespa Saya 65
Gambar 4.27 Antarmuka Halaman Riwayat Servis Pelanggan 66
Gambar 4.28 Antarmuka Halaman *Dashboard* Admin 67
Gambar 4.29 Antarmuka Lonceng Notifikasi Pemesanan Admin 68
Gambar 4.30 Antarmuka Halaman Kelola Transaksi 69
Gambar 4.31 Antarmuka Halaman Kelola Layanan 70
Gambar 4.32 Antarmuka Halaman Kelola Suku Cadang 71

Gambar 4.33 Antarmuka Halaman *Restock* Suku Cadang 72
Gambar 4.34 Antarmuka Halaman *Dashboard* Mekanik 73
Gambar 4.35 Antarmuka *Pop-up* Penambahan Suku Cadang 74
Gambar 4.36 Antarmuka *Dashboard* Pemilik 75
Gambar 4.37 Antarmuka Halaman Laporan Keuangan 77
Gambar 4.38 Antarmuka Halaman Analisis Inventaris 78
Gambar 4.39 Antarmuka Halaman Audit Aktivitas 79

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini telah menjadi kebutuhan mendasar bagi berbagai sektor bisnis, termasuk industri jasa otomotif. Pemanfaatan sistem informasi manajemen tidak hanya berfungsi sebagai alat pencatatan, tetapi juga sebagai instrumen strategis untuk meningkatkan efisiensi operasional, transparansi data, dan kualitas pelayanan kepada pelanggan. Hal ini sejalan dengan penelitian Aripin et al. (2025) yang menyatakan bahwa masih banyak bengkel yang menjalankan proses bisnis secara manual, sehingga menyebabkan inefisiensi dan kesalahan pencatatan yang menghambat pertumbuhan usaha. Oleh karena itu, peralihan dari sistem manajemen konvensional menuju digitalisasi menjadi langkah krusial untuk menjaga keberlangsungan bisnis di tengah persaingan yang semakin ketat (Wuladari & Fersellia, 2025).

KRGarage merupakan salah satu bengkel yang berfokus pada perawatan dan perbaikan Vespa 2-tak. Seiring dengan meningkatnya jumlah pelanggan dan intensitas transaksi servis, kompleksitas operasional KRGarage turut bertambah. Berdasarkan observasi awal dan hasil wawancara dengan pihak pengelola, proses bisnis di KRGarage saat ini masih didominasi oleh metode konvensional. Proses pendaftaran servis mayoritas dilakukan melalui aplikasi pesan singkat WhatsApp, pencatatan jadwal menggunakan kalender di telepon genggam, serta arsip riwayat servis yang masih mengandalkan salinan bon fisik.

Penerapan metode manual tersebut memunculkan sejumlah kendala signifikan yang menghambat efisiensi layanan bengkel. Dari sisi manajerial, pemilik menghadapi kendala keterbatasan visibilitas data keuangan akibat pencatatan keuangan yang dilakukan secara manual di buku tulis atau catatan telepon genggam. Pencatatan manual ini menyulitkan pemilik dalam melacak total pendapatan harian, mencatat pengeluaran untuk *restock* suku cadang, dan menghitung keuntungan operasional secara cepat dan akurat. Nurlaila & Mulyono (2023) dalam penelitiannya menekankan bahwa sistem manual seringkali menyulitkan pembuatan laporan keuangan yang akurat dan rapi. Hal ini mengakibatkan pemilik sulit memantau kondisi operasional dan margin keuntungan secara *real-time*, serta rentan terhadap risiko kehilangan data transaksi yang krusial bagi pengambilan keputusan strategis (Santoso et al., 2024).

Permasalahan operasional juga dirasakan oleh bagian administrasi dan teknisi. Admin mengalami inefisiensi waktu yang cukup besar saat harus mencari data riwayat pelanggan lama

karena harus membongkar tumpukan arsip bon fisik untuk melihat rekam jejak servis sebelumnya. Selain itu, proses pembuatan nota transaksi yang mewajibkan admin menghitung total biaya jasa dan harga suku cadang secara manual seringkali menyebabkan selisih perhitungan, sehingga data pendapatan menjadi tidak akurat. Saputri et al. (2024) menyebutkan bahwa pelayanan yang masih manual membuat pelanggan harus mengantri lama dan pengelolaan data transaksi menjadi lambat. Di sisi lain, mekanik menilai alur kerja saat ini kurang efisien karena instruksi kerja hanya disampaikan melalui catatan manual yang berisiko hilang, kotor, atau sulit terbaca, yang berpotensi menyebabkan kesalahan dalam penanganan keluhan pelanggan.

Dari sisi pelayanan pelanggan, ketiadaan sistem yang memadai membuat pelanggan kesulitan mendapatkan kepastian jadwal dan transparansi informasi. Maulana et al. (2024) mengungkapkan bahwa penerapan sistem pemesanan servis sangat membantu masyarakat untuk memperoleh informasi layanan dengan lebih mudah dan efektif. Tanpa sistem tersebut, pelanggan KRGarage seringkali lupa jadwal perawatan berkala dan harus memasukkan data Vespa secara berulang-ulang setiap kali melakukan pemesanan via WhatsApp, yang mengurangi kenyamanan pelanggan.

Melihat kompleksitas permasalahan tersebut, dibutuhkan sebuah solusi digital yang mampu mengintegrasikan seluruh alur layanan bengkel ke dalam satu *platform*. Penelitian ini mengusulkan pembangunan sistem menggunakan teknologi Laravel dan Vue.js. Kombinasi teknologi ini dipilih karena kemampuannya dalam mengelola logika *backend* yang kompleks dan antarmuka *frontend* yang interaktif (Ginting & Pandia, 2025; Liaotando et al., 2024). Sistem ini dirancang untuk mendigitalisasi proses pemesanan servis, pencatatan riwayat, dan pelaporan keuangan guna meningkatkan efisiensi operasional dan kualitas layanan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, pengelolaan operasional dan pelayanan di bengkel KRGarage masih terkendala oleh inefisiensi pencarian data riwayat servis, risiko kesalahan perhitungan biaya transaksi, dan keterbatasan akses informasi bagi pelanggan. Agar solusi yang ditawarkan terarah dan hasil penelitian dapat diukur secara objektif, maka rumusan masalah dalam penelitian ini difokuskan pada dua pertanyaan utama:

a. Bagaimana merancang dan membangun sistem informasi manajemen layanan bengkel berbasis *web* yang dapat mengintegrasikan proses pemesanan servis *online*, pengelolaan

data Vespa dan riwayat servis, serta kalkulasi otomatis untuk mencatat pengeluaran, pendapatan, dan keuntungan di KRGarage?

b. Bagaimana hasil pengujian fungsionalitas dan kelayakan sistem informasi yang dibangun untuk memastikan kesesuaiannya dengan kebutuhan operasional pengguna?

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan yang ditetapkan. Batasan ini bertujuan untuk memperjelas ruang lingkup penelitian, sehingga solusi yang dikembangkan dapat lebih terarah dan sesuai dengan kebutuhan pengguna utama yaitu pemilik, admin, mekanik, dan pelanggan bengkel KRGarage. Adapun batasan yang diterapkan dalam penelitian ini adalah sebagai berikut:

a. Penelitian ini berfokus pada pengembangan sistem informasi untuk manajemen layanan jasa, pemesanan servis secara *online*, pencatatan riwayat servis, dan pengelolaan data suku cadang sederhana. Fitur manajemen suku cadang mencakup pengurangan otomatis saat transaksi servis dan pencatatan *restock* sederhana sebagai dasar perhitungan pengeluaran bengkel, tanpa mencakup metode penilaian persediaan yang kompleks.
b. Sistem yang dikembangkan berbasis *website* menggunakan *framework* Laravel dan Vue.js yang dirancang responsif, sehingga dapat diakses melalui *browser* perangkat seluler maupun komputer tanpa mencakup pengembangan aplikasi *mobile*.
c. Fitur laporan keuangan dibatasi pada pencatatan pendapatan servis, pencatatan pengeluaran *restock* suku cadang, dan kalkulasi keuntungan operasional sederhana. Sistem tidak mencakup pembukuan akuntansi kompleks seperti neraca atau laporan arus kas standar.
d. Evaluasi sistem berfokus pada pengujian fungsionalitas fitur utama serta kemudahan penggunaan oleh admin, mekanik, dan pemilik dalam mengelola operasional, serta pelanggan dalam melakukan pemesanan layanan.
e. Sistem tidak menyediakan fitur persetujuan penambahan suku cadang secara langsung oleh pelanggan selama proses servis berlangsung. Apabila diperlukan penambahan suku cadang di luar pemesanan awal, konfirmasi kepada pelanggan dilakukan secara manual oleh admin melalui kontak WhatsApp yang tertera pada detail pemesanan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, tujuan utama yang ingin dicapai dalam penelitian ini adalah:

a. Merancang dan Membangun Sistem Informasi Bengkel

Merancang dan membangun sistem informasi manajemen layanan bengkel berbasis *web* yang mampu mengintegrasikan proses pemesanan servis *online*, pengelolaan data Vespa dan riwayat servis, serta kalkulasi otomatis untuk mencatat pengeluaran, pendapatan, dan keuntungan operasional secara terpusat di KRGarage.

b. Mengetahui Hasil Pengujian Kelayakan Sistem

Mengukur keberhasilan sistem melalui pengujian fungsional dan pengujian penerimaan pengguna untuk memastikan bahwa fitur-fitur yang dikembangkan telah berjalan sesuai spesifikasi dan dapat diterima sebagai solusi operasional di KRGarage.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi nyata dalam transformasi digital pada sektor UMKM otomotif, khususnya dalam optimalisasi manajemen operasional bengkel. Harapan manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

a. Sistem ini memberikan solusi atas kendala pencatatan manual melalui fitur kalkulasi otomatis yang meminimalisir risiko kesalahan hitung dalam penggabungan biaya jasa dan suku cadang. Selain itu, pemilik bengkel dapat memantau laporan pendapatan, pengeluaran *restock*, margin keuntungan, serta status pembayaran pelanggan secara *real-time* tanpa harus berada di lokasi, sehingga pengambilan keputusan bisnis menjadi lebih cepat dan berbasis data.
b. Memberikan kenyamanan dan kepastian layanan bagi pelanggan melalui fitur pemesanan servis *online* yang transparan dan akses mandiri terhadap riwayat perawatan Vespa. Penerapan fitur pengingat servis pada *dashboard* pelanggan juga bermanfaat agar pelanggan dapat mengetahui Vespa yang telah memasuki atau melewati jadwal perawatan berkala, yang secara tidak langsung menjaga performa Vespa tetap prima.
c. Hasil perancangan sistem menggunakan arsitektur *Client-Server* dengan *framework* Laravel dan Vue.js ini diharapkan dapat menjadi referensi akademis peneliti atau pengembang lain yang ingin membangun sistem manajemen layanan serupa. Penelitian ini juga memperkaya literatur mengenai penerapan metode *Prototyping* dalam menyelesaikan masalah inefisiensi administrasi pada usaha bengkel skala kecil hingga menengah.

1.6 Metodologi Penelitian

Untuk mencapai tujuan penelitian yang telah ditetapkan, penulis menyusun langkah-langkah sistematis yang mencakup seluruh rangkaian kegiatan dari awal perencanaan hingga penyelesaian laporan akhir. Langkah-langkah tersebut meliputi:

a. Studi Literatur

Tahap ini dilakukan untuk membangun fondasi teoritis yang kuat sebelum memulai pengembangan sistem. Penulis mempelajari literatur berupa jurnal ilmiah nasional dan internasional serta buku teks yang diterbitkan dalam kurun waktu 5 tahun terakhir. Fokus materi yang dikaji meliputi konsep Sistem Informasi Manajemen Bengkel untuk memahami proses bisnis standar, teori metode pengembangan perangkat lunak *Prototyping*, serta dokumentasi teknis terkait teknologi *framework* Laravel dan Vue.js. Kajian ini bertujuan untuk menemukan celah penelitian dari penelitian terdahulu dan menentukan solusi teknologi yang paling tepat.

b. Identifikasi Masalah dan Pengumpulan Data

Penulis terjun langsung ke lapangan untuk memahami kondisi operasional yang berjalan. Langkah ini dilakukan melalui observasi terhadap alur kerja manual dan wawancara mendalam dengan pemilik, admin, mekanik, serta pelanggan bengkel. Tujuannya adalah untuk memetakan akar permasalahan seperti inefisiensi pencarian riwayat servis, kendala pelaporan keuangan yang tidak *real-time*, dan kebutuhan akan pengingat jadwal servis, sehingga kebutuhan fungsional sistem dapat didefinisikan secara akurat.

c. Pengembangan Sistem

Setelah kebutuhan sistem diperoleh, penulis melakukan pengembangan sistem menggunakan metode *Prototyping*. Tahapan ini meliputi perancangan prototipe antarmuka, evaluasi prototipe bersama pengguna, implementasi sistem, pengujian fungsional menggunakan metode *Black Box Testing*, serta evaluasi penerimaan pengguna menggunakan *User Acceptance Testing* (UAT). Proses tersebut dilakukan secara bertahap untuk memastikan sistem yang dikembangkan sesuai dengan kebutuhan operasional KRGarage.

d. Penyusunan Laporan

Tahap akhir dari rangkaian penelitian ini adalah mendokumentasikan seluruh proses dan hasil penelitian ke dalam bentuk laporan tugas akhir. Penyusunan laporan dilakukan secara sistematis mulai dari Bab I hingga Bab V sesuai dengan kaidah penulisan akademis yang berlaku, yang kemudian akan dipresentasikan sebagai syarat kelulusan.

1.7 Struktur Penelitian

Untuk memberikan gambaran menyeluruh mengenai alur penyusunan laporan tugas akhir ini, sistematika penulisan disusun ke dalam lima bab utama sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang permasalahan yang mendasari penelitian, identifikasi dan rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta metodologi dan sistematika penulisan yang digunakan.

BAB II KAJIAN PUSTAKA

Bab ini menyajikan telaah terhadap penelitian-penelitian terdahulu yang relevan serta teori-teori pendukung yang menjadi landasan penelitian. Pembahasan mencakup konsep sistem informasi manajemen, operasional bengkel, serta tinjauan teknis terhadap teknologi pengembangan yang digunakan.

BAB III METODE PENELITIAN

Bab ini menguraikan tahapan metodologi penelitian secara rinci menggunakan metode *Prototyping*. Penjelasan meliputi alur kerja mulai dari analisis kebutuhan, perancangan prototipe, evaluasi prototipe, implementasi sistem, pengujian sistem, hingga evaluasi dan penerapan sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari perancangan dan implementasi sistem yang telah dibangun. Pembahasan mencakup tampilan antarmuka sistem, hasil pengujian fungsional, serta analisis evaluasi pengguna terhadap kinerja sistem yang dikembangkan.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan akhir dari seluruh rangkaian penelitian yang telah dilakukan serta saran-saran konstruktif untuk pengembangan sistem lebih lanjut di masa mendatang.

BAB II
KAJIAN PUSTAKA

2.1 Tinjauan Penelitian Terdahulu

Dalam penyusunan penelitian ini, penulis melakukan penelusuran mendalam terhadap berbagai literatur ilmiah guna membangun landasan teoritis yang kokoh. Kajian ini melibatkan analisis komparatif terhadap penelitian-penelitian sebelumnya yang diterbitkan dalam kurun waktu lima tahun terakhir. Tujuannya adalah untuk mengidentifikasi tren pengembangan sistem informasi bengkel saat ini, sekaligus menemukan celah penelitian yang belum terselesaikan secara optimal oleh peneliti terdahulu, baik dari aspek fungsionalitas sistem maupun pemilihan teknologi pengembangan.

Secara garis besar, literatur yang dikaji dapat diklasifikasikan ke dalam dua domain utama. Pertama, penelitian yang berfokus pada fungsionalitas dan fokus bisnis. Banyak penelitian terdahulu, seperti yang dilakukan oleh Wuladari & Fersellia (2025) serta Nurlaila & Mulyono (2023), menitikberatkan solusi sistem informasi pada aspek manajemen inventori gudang dan akuntansi keuangan yang kompleks. Pendekatan ini, meskipun efektif untuk bengkel skala besar, seringkali kurang relevan untuk diterapkan pada bengkel UMKM yang lebih memprioritaskan kecepatan pelayanan jasa. Hal ini membuka peluang untuk mengembangkan sistem yang lebih berorientasi pada manajemen layanan dan kenyamanan pelanggan, bukan sekadar pencatatan stok barang.

Kedua, kajian terhadap pendekatan teknis pengembangan. Mayoritas sistem bengkel sebelumnya masih dikembangkan menggunakan metode *Waterfall* yang bersifat linear, atau jika menggunakan *Prototyping* (Saputri et al., 2024), implementasinya masih terbatas pada *website* standar. Selain itu, beberapa solusi mewajibkan penggunaan aplikasi *mobile* yang harus diinstal (Santoso et al., 2024), yang seringkali menjadi hambatan bagi pelanggan insidental. Oleh karena itu, penelitian ini menawarkan pengembangan sistem berbasis *web* responsif dengan menggabungkan *Prototyping* serta *framework* Laravel dan Vue.js, sehingga sistem dapat digunakan melalui *browser* tanpa perlu instalasi aplikasi *mobile*.

Rangkuman perbandingan antara penelitian ini dengan penelitian-penelitian terdahulu disajikan secara rinci pada Tabel 2.1 berikut:

Tabel 2.1 Perbandingan Penelitian Terdahulu

No	Judul & Topik	Metode & Fokus Utama	Perbedaan & Solusi pada Penelitian Ini

1	Implementasi Sistem Informasi Manajemen Bengkel Berbasis Web untuk Peningkatan Efisiensi Operasional (Studi Kasus: Bengkel AA Motor) (Wuladari & Fersellia, 2025)	*Waterfall*. Fokus pada efisiensi operasional dengan fitur manajemen stok barang dan pelaporan keuangan	Penelitian ini menggunakan metode *Waterfall* yang bersifat linear. Penelitian penulis menggunakan metode *Prototyping* agar lebih adaptif. Selain itu, penulis menyederhanakan fitur stok pada pengurangan otomatis saat transaksi dan pencatatan *restock* sederhana, bukan manajemen gudang yang kompleks
2	Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Service Pada Bengkel Sido Motor Berbasis Website (Saputri et al., 2024)	*Prototyping*. Fokus pada pelayanan jasa servis, penjualan *sparepart*, dan laporan transaksi manual	Menggunakan metode yang sama, namun fitur pemesanan servis pada penelitian tersebut masih sederhana. Penelitian penulis mengembangkan fitur pemesanan servis dengan pilihan jam interaktif untuk kepastian jadwal dan status pengerjaan *real-time*
3	Sistem Informasi Manajemen Bengkel Den Bagus Dinamo (Santoso et al., 2024)	*Web & Mobile App*. Mengembangkan aplikasi *mobile* khusus untuk pelanggan dan *web* untuk admin	Penelitian tersebut mengharuskan pelanggan menginstal aplikasi *mobile*. Penelitian penulis menggunakan teknologi *web* responsif dengan Vue.js, sehingga pelanggan dapat mengakses fitur layaknya aplikasi tanpa perlu instalasi
4	Sistem Informasi Manajemen Bengkel Berbasis Web pada Bengkel Ikhsan Jaya Motor (Nurlaila & Mulyono, 2023)	RAD. Fokus berat pada akuntansi, laporan laba rugi, dan mengatasi selisih stok barang	Sistem tersebut berorientasi pada *Point of Sales* (POS) dan Keuangan. Penelitian penulis membatasi fitur keuangan pada rekapitulasi pendapatan, pengeluaran *restock* dan perhitungan keuntungan operasional sederhana tanpa pembukuan akuntansi kompleks
5	Pembangunan Sistem Informasi CV Sembilan Matahari Sejahtera Berbasis Website (Liaotando et al., 2024)	Laravel dan Vue.js. Fokus pada pencatatan transaksi jual beli produk F&B dan stok gudang.	Menggunakan *tech stack* yang sama (Laravel & Vue.js), namun diterapkan pada konteks jasa. Penulis memanfaatkan reaktivitas Vue.js untuk memantau status pengerjaan servis, bukan sekadar transaksi jual beli produk
6	Perancangan Sistem Informasi Pelayanan Surat Pengantar Dukuh Berbasis Website "Dukuhku Online" (Rahmadi et al., 2020)	*Prototyping*. Fokus pada pelayanan publik (surat pengantar) untuk memudahkan warga	Menjadi referensi penerapan metode *prototyping* yang efektif. Penulis mengadopsi pendekatan ini untuk sistem reservasi bengkel agar alur antarmuka mudah dipahami oleh pelanggan awam.

Berdasarkan tabel di atas, mayoritas penelitian terdahulu masih didominasi oleh metode *Waterfall* dan berfokus pada manajemen stok/keuangan. Penelitian yang menggunakan metode *Prototyping* sudah ada, namun penelitian ini menekankan pengembangan sistem *web* responsif yang mendukung proses pemesanan servis, pemantauan status pengerjaan, dan pelaporan

operasional bengkel secara terintegrasi. Oleh karena itu, penelitian ini mengisi celah tersebut dengan membangun sistem informasi manajemen layanan yang menggunakan metode *Prototyping* dengan teknologi Laravel dan Vue.js untuk menghasilkan aplikasi yang responsif, fokus pada kenyamanan pelanggan, dan sesuai dengan model bisnis bengkel.

2.2 Perbandingan Metodologi Pengembangan

Pemilihan metodologi pengembangan perangkat lunak merupakan faktor krusial yang menentukan keberhasilan sebuah proyek sistem informasi. Kesalahan dalam memilih metode dapat berakibat pada pembengkakan biaya, keterlambatan penyelesaian, hingga ketidaksesuaian fitur aplikasi dengan kebutuhan pengguna akhir. Dalam konteks rekayasa perangkat lunak, terdapat dua pendekatan utama yang sering digunakan, yaitu pendekatan linier dan pendekatan iteratif.

Pendekatan linier yang diwakilkan oleh metode *Waterfall*, menekankan pada perencanaan yang matang di awal dengan tahapan yang mengalir secara berurutan dari analisis hingga pemeliharaan. Metode ini sangat efektif untuk proyek dengan kebutuhan yang sudah terdefinisi secara jelas dan tidak berubah-ubah. Sebaliknya, pendekatan iteratif seperti *Prototyping* menawarkan fleksibilitas yang lebih tinggi dengan melibatkan pengguna secara aktif dalam proses evaluasi model awal sistem. Pendekatan ini lebih cocok diterapkan pada kondisi di mana kebutuhan pengguna masih bersifat dinamis atau sulit didefinisikan secara rinci di awal proyek.

Dalam penelitian ini, penulis melakukan komparasi karakteristik antara kedua metode tersebut untuk menentukan pendekatan yang paling relevan dengan kondisi operasional di bengkel KRGarage. Mengingat kompleksitas alur kerja bengkel yang melibatkan interaksi langsung antara admin, mekanik, dan pelanggan, diperlukan metode yang mampu mengakomodasi umpan balik secara cepat untuk memastikan sistem yang dibangun benar-benar memberikan solusi yang tepat.

Tabel 2.2 Perbandingan Metodologi Pengembangan Sistem

Metodologi	**Karakteristik Utama**	**Kelebihan**	**Kekurangan**
Waterfall	Linear (bertahap) (Wuladari & Fersellia, 2025)	Terstruktur, dokumentasi lengkap, jadwal mudah diprediksi (Aripin et al., 2025)	Kaku, sulit mengakomodasi perubahan kebutuhan di tengah jalan (Wahyudi & Rahmi, 2022)

RAD	Pengembangan cepat berbasis komponen (Nurlaila & Mulyono, 2023)	Waktu pengembangan sangat cepat (Maulana et al., 2024)	Membutuhkan sumber daya manusia yang banyak dan keterlibatan *user* yang sangat intensif (Nurlaila & Mulyono, 2023)
Agile (Scrum)	Iteratif dengan siklus pendek (*Sprint*) (Liaotando et al., 2024)	Sangat fleksibel dan adaptif terhadap perubahan cepat (Liaotando et al., 2024)	Membutuhkan komunikasi harian yang intens dan komitmen waktu tinggi dari klien/*owner*
Prototyping	Iteratif, berbasis model tiruan (Arizal et al., 2022)	Melibatkan pengguna secara aktif dalam evaluasi desain sejak awal (Saputri et al., 2024)	Bisa memakan waktu jika pelanggan terus menerus meminta revisi (Rahmadi et al., 2020)

Berdasarkan hasil komparasi karakteristik metodologi pada Tabel 2.2, penelitian ini menetapkan metode *Prototyping* sebagai pendekatan yang paling relevan untuk diterapkan. Keputusan ini didasarkan pada analisis kebutuhan operasional KRGarage yang dinamis, di mana preferensi antarmuka pengguna memegang peranan vital dalam keberhasilan implementasi sistem. Berbeda dengan metode *Waterfall* yang kaku dan mensyaratkan definisi kebutuhan final di awal, *Prototyping* menawarkan fleksibilitas melalui siklus iteratif. Pendekatan ini memungkinkan pengembang untuk mendemonstrasikan purwarupa fitur krusial seperti alur pemesanan servis *online* dan *dashboard* admin lebih awal untuk mendapatkan umpan balik langsung. Hal ini sejalan dengan pandangan Arizal et al. (2022), yang menyatakan bahwa validasi visual di tahap awal sangat efektif untuk meminimalkan risiko ketidaksesuaian sistem di akhir masa pengembangan.

2.3 Landasan Teori

Bagian ini menguraikan konsep-konsep teoritis dan spesifikasi teknologi yang relevan untuk mendukung tahapan pengembangan sistem, mulai dari metode pendekatan, *framework*, hingga teknik pengujian yang diterapkan.

2.3.1 Sistem Informasi Manajemen Bengkel

Sistem Informasi Manajemen (SIM) Bengkel adalah sistem terpadu yang mengelola data operasional bengkel, mulai dari pendaftaran pelanggan, transaksi servis, hingga pelaporan, guna meningkatkan efisiensi kerja (Santoso et al., 2024). Penerapan SIM pada bengkel bertujuan untuk menggantikan pencatatan manual yang rentan terhadap kesalahan dengan sistem basis data yang terstruktur dan mudah diakses. Selain berfokus pada manajemen

transaksi dasar, beberapa penelitian sejenis juga mengintegrasikan fungsi rekapitulasi laporan laba rugi serta sinkronisasi data transaksi penjualan dan pembelian suku cadang untuk meminimalkan selisih perhitungan keuangan (Nurlaila & Mulyono, 2023).

Kebutuhan akan basis data yang andal menjadi aspek krusial dalam SIM Bengkel. Kegagalan dalam pemilihan *Database Management System* (DBMS) dapat menyebabkan inefisiensi pengerjaan apabila sistem diakses oleh beberapa administrator secara bersamaan, seperti pada penggunaan Microsoft Access yang kurang optimal dalam arsitektur jaringan lokal sehingga memerlukan migrasi ke DBMS yang lebih stabil seperti MySQL (Kurniawan et al., 2022). Oleh karena itu, transformasi digital melalui integrasi basis data yang tepat menjadi langkah strategis untuk memperlancar manajemen operasional harian bengkel secara terpusat (Silaban et al., 2025).

Secara spesifik, penelitian yang dilakukan oleh Silaban et al. (2025) memiliki fokus objek pada pengembangan *website* layanan bengkel otomotif dengan cakupan utama berupa integrasi formulir masukan konsumen dan sistem pemantauan admin. Meskipun memiliki kesamaan dalam penggunaan platform berbasis *web* dengan bahasa pemrograman PHP dan basis data MySQL, terdapat perbedaan cakupan yang signifikan antara penelitian tersebut dengan penelitian yang dikembangkan saat ini. Fungsionalitas pada sistem Silaban et al. (2025) hanya memfasilitasi dua aktor utama dan terbatas pada pengisian data reservasi oleh konsumen dan panel pemantauan status pekerjaan oleh admin tanpa mencakup manajemen inventaris suku cadang maupun rekapitulasi laporan keuangan. Sebaliknya, sistem yang dibangun dalam penelitian ini memiliki kelengkapan fitur yang jauh lebih komprehensif. Sistem KRGarage mengintegrasikan empat hak akses aktor secara spesifik, yaitu pemilik, admin, mekanik, dan pelanggan, dilengkapi dengan modul manajemen inventaris suku cadang yang terpotong otomatis, kalkulasi biaya transaksi secara *real-time*, antarmuka khusus pembaruan status pengerjaan oleh mekanik, serta fitur *dashboard* analisis pelaporan keuangan dan operasional yang terintegrasi langsung untuk pemilik bengkel.

Secara karakteristik, sistem ini bekerja dengan mengintegrasikan alur informasi antara administrasi, mekanik, dan pemilik dalam satu *platform* (Wuladari & Fersellia, 2025). Hal ini memungkinkan data seperti riwayat kendaraan dan stok layanan dapat diakses secara *real-time* tanpa batasan waktu dan tempat. Dalam perkembangannya, fungsionalitas SIM Bengkel juga diperluas untuk mengakomodasi fitur pemesanan servis interaktif untuk mengurai antrean fisik pelanggan di lokasi bengkel (Saputri et al., 2024). Pembangunan *platform* ini umumnya diimplementasikan menggunakan bahasa pemrograman dinamis berbasis *server-side* seperti

PHP, dikombinasikan dengan *framework* responsif untuk menghasilkan panel konsumen mandiri serta *dashboard monitoring* yang informatif bagi pihak pengelola bengkel (Silaban et al., 2025).

Keandalan dan tingkat kelayakan dari fungsionalitas SIM Bengkel yang terintegrasi wajib diuji secara langsung sebelum diterapkan sepenuhnya. Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* untuk memvalidasi bahwa seluruh modul utama sistem seperti pengelolaan data transaksi, stok suku cadang, dan layanan servis dapat merespons input dengan benar serta berfungsi sesuai spesifikasi tanpa ditemukan kesalahan fungsional (Silaban et al., 2025). Selain pengujian fungsi, tingkat penerimaan pengguna akhir terhadap sistem dievaluasi secara komprehensif menggunakan metode *User Acceptance Testing* (UAT) (Sriyeni & Priatama, 2024). Evaluasi berbasis kuesioner UAT ini dilakukan dengan melibatkan pengguna langsung untuk mengukur persentase kelayakan sistem, kegunaan, serta kenyamanan antarmuka saat sistem dioperasikan secara nyata untuk membantu operasional pelayanan bengkel di lapangan (Sriyeni & Priatama, 2024).

2.3.2 Aktor

Dalam perancangan sistem informasi manajemen bengkel, entitas pengguna atau aktor dikelompokkan ke dalam beberapa hak akses dan pembagian peran yang spesifik dengan tujuan untuk mengoptimalkan pembagian tugas operasional serta menjaga tingkat keamanan integritas data di dalam sistem (Saputri et al., 2024).

Berdasarkan tinjauan literatur pada pengembangan sistem informasi bengkel, kelompok aktor pertama yang terlibat secara langsung adalah pelanggan atau pengguna eksternal yang memiliki otorisasi untuk melakukan pendaftaran akun mandiri, memasukkan informasi data diri beserta spesifikasi kendaraan, memilih kategori jenis layanan perbaikan yang dibutuhkan, serta melakukan pemesanan jadwal servis secara *online* untuk meminimalisir waktu tunggu di lokasi bengkel (Santoso et al., 2024).

Selanjutnya, admin bertindak sebagai pengelola operasional harian sistem. Aktor ini bertanggung jawab penuh dalam mengelola data master keseluruhan, memverifikasi setiap pesanan atau reservasi servis yang masuk dari pelanggan, mengatur penjadwalan kunjungan, serta menangani pencatatan administrasi transaksi pembayaran dan pengelolaan persediaan suku cadang (Aripin et al., 2025).

Selain itu, terdapat aktor mekanik yang bertugas sebagai pelaksana teknis utama di lapangan. Aktor ini memiliki hak akses untuk menerima informasi rincian jadwal pengerjaan,

mencatat keluhan serta proses perbaikan kendaraan, menambahkan komponen suku cadang yang terpakai selama servis, dan memperbarui status progres pengerjaan kendaraan secara berkala (Maulana et al., 2024).

Terakhir, pemilik bengkel ditempatkan sebagai aktor pada tingkat manajerial tertinggi yang berwenang memantau keseluruhan aktivitas operasional bengkel secara terpusat melalui tampilan *dashboard* analitik serta meninjau laporan keuangan, tren pendapatan, dan rekam jejak transaksi secara periodik untuk mendukung pengambilan keputusan (Wuladari & Fersellia, 2025).

2.3.3 Proses Bisnis

Proses bisnis pada sistem informasi bengkel dirancang dan dikembangkan untuk mentransformasikan alur kerja operasional yang sebelumnya masih konvensional dan bergantung pada pencatatan manual, menjadi berbasis digital agar pengelolaan data menjadi jauh lebih efektif, akurat, dan efisien (Wahyudi & Rahmi, 2022). Perubahan alur kerja ke bentuk digital ini menjadi langkah krusial dalam mengatasi berbagai kendala administratif, seperti lambatnya pencarian data, potensi kesalahan pencatatan, serta tidak adanya rekam jejak transaksi yang terpusat.

Tahapan alur proses bisnis tersebut dimulai dari tahap pendaftaran dan reservasi layanan. Pada tahap ini, proses dimulai ketika pelanggan mengakses sistem berbasis *web* atau aplikasi untuk melakukan pendaftaran akun mandiri, memasukkan informasi spesifik kendaraan, serta mengajukan pemesanan jadwal servis secara *online*. Dengan mekanisme ini, pelanggan dapat menentukan waktu kunjungan yang diinginkan tanpa harus mengalami kendala antrean panjang secara langsung di lokasi bengkel(Silaban et al., 2025).

Tahap berikutnya adalah proses verifikasi dan penugasan kerja yang dikelola oleh pihak admin. Pada tahap ini, admin menerima data pemesanan servis yang masuk ke dalam sistem, memeriksa ketersediaan slot waktu dan sumber daya, melakukan konfirmasi pesanan kepada pelanggan, serta mendistribusikan penugasan pengerjaan perbaikan kepada mekanik yang bertugas secara terstruktur (Maulana et al., 2024).

Setelah penugasan dikonfirmasi, alur berlanjut ke tahap pengerjaan servis di area kerja bengkel. Pada tahap ini, mekanik menjalankan tugas perbaikan atau perawatan kendaraan sesuai dengan rincian keluhan yang dilaporkan oleh pelanggan, melakukan pemeriksaan ketersediaan inventaris suku cadang, serta memperbarui status progres pengerjaan kendaraan secara *real-time* langsung melalui antarmuka sistem (Santoso et al., 2024).

Siklus operasional ini diakhiri dengan tahap penyelesaian transaksi dan pelaporan. Pada tahap akhir ini, admin memproses pembayaran transaksi akhir, mencatat rincian biaya yang mencakup jasa perbaikan serta komponen suku cadang yang diganti, dan menghasilkan nota atau *invoice* pembayaran. Bersamaan dengan itu, sistem secara otomatis memperbarui data sisa inventaris di dalam basis data dan menyusun rekapitulasi laporan pendapatan harian maupun bulanan yang dapat ditinjau langsung oleh pemilik bengkel untuk kebutuhan evaluasi bisnis (Nurlaila & Mulyono, 2023).

2.3.4 Metode *Prototyping*

Metode *Prototyping* merupakan pendekatan dalam rekayasa perangkat lunak yang berfokus pada penyajian model kerja awal kepada pengguna sebelum tahap konstruksi kode sesungguhnya dilakukan. Menurut Saputri et al. (2024), metode ini dirancang secara khusus untuk menjembatani kesenjangan komunikasi yang sering terjadi antara pengembang sistem dan pengguna akhir. Dalam banyak skenario pengembangan, pengguna sering kali mengalami kesulitan untuk mendefinisikan kebutuhan sistem secara rinci dan spesifik hanya melalui dokumen spesifikasi teks. Akibatnya, sering terjadi perbedaan persepsi yang baru disadari ketika sistem sudah selesai dibangun. Dengan adanya prototipe, pengguna memperoleh gambaran visual sejak tahap awal pengembangan sehingga risiko kesalahpahaman dapat diminimalkan.

Secara prosedural, metode ini memiliki karakteristik iteratif atau berulang yang melibatkan partisipasi aktif pengguna. Mengacu pada penelitian Arizal et al. (2022), siklus pengembangan dimulai dengan tahap pengumpulan kebutuhan untuk memetakan objektif sistem, dilanjutkan dengan perancangan cepat untuk memvisualisasikan antarmuka pengguna. Model tersebut kemudian masuk ke tahap evaluasi, di mana pengguna memberikan umpan balik terhadap desain yang diajukan. Jika desain telah disepakati, pengembang melangkah ke tahap implementasi sistem untuk menerjemahkan desain ke dalam bahasa pemrograman, yang diikuti oleh pengujian sistem untuk memastikan fungsi berjalan tanpa kesalahan. Rangkaian ini ditutup dengan evaluasi akhir dan kesiapan penerapan sistem sesuai hasil pengujian serta evaluasi pengguna.

Keunggulan utama pendekatan ini adalah fleksibilitasnya dalam mengakomodasi perubahan kebutuhan di tengah proses pengembangan. Rahmadi et al. (2020) menekankan bahwa berbeda dengan metode konvensional yang bersifat linier dan kaku, *Prototyping* memungkinkan adanya siklus umpan balik yang cepat. Hal ini memberikan jaminan bahwa

sistem yang dibangun benar-benar sesuai dengan ekspektasi pengguna, terutama dalam kasus pembangunan sistem layanan yang membutuhkan validasi alur kerja yang presisi seperti pada layanan bengkel.

2.3.5 *Framework* Laravel

Laravel adalah kerangka kerja (*framework*) PHP yang populer dengan arsitektur *Model-View-Controller* (MVC). Laravel menyediakan fitur keamanan yang kuat, manajemen basis data yang ekspresif (*Eloquent ORM*), dan kemudahan dalam membangun *RESTful API* yang dibutuhkan untuk komunikasi data dengan *frontend* (Liaotando et al., 2024). Pada sistem ini, Laravel berfungsi sebagai *backend* yang menangani logika bisnis dan keamanan data transaksi.

Karakteristik yang menonjol dari Laravel adalah sintaksnya yang ekspresif dan elegan, serta ketersediaan fitur bawaan yang lengkap seperti sistem autentikasi, *routing*, dan manajemen sesi. Hal ini memungkinkan pengembang untuk fokus pada logika aplikasi tanpa harus membangun fitur dasar dari nol, sehingga mempercepat proses pengembangan (Ginting & Pandia, 2025).

Dari sisi keunggulan, Laravel menawarkan tingkat keamanan yang tinggi. *Framework* ini secara otomatis menangani berbagai celah keamanan umum seperti *SQL injection* dan *Cross-Site Scripting (XSS)*, memberikan jaminan bahwa data pelanggan dan transaksi bengkel tersimpan dengan aman dan terlindungi dari akses yang tidak sah (Ginting & Pandia, 2025).

2.3.6 *Framework* Vue.js

Vue.js adalah kerangka kerja JavaScript progresif yang digunakan untuk membangun antarmuka pengguna yang interaktif dan responsif. Vue.js mendukung pembaruan data pada tampilan secara dinamis, sehingga pengguna dapat memperoleh pengalaman penggunaan yang lebih cepat dan nyaman tanpa harus memuat ulang seluruh halaman secara berulang (Ginting & Pandia, 2025). Teknologi ini diterapkan pada antarmuka pemesanan servis, *dashboard*, dan pengelolaan data agar perubahan informasi dapat ditampilkan secara lebih interaktif.

Secara karakteristik, Vue.js mengadopsi konsep *Virtual DOM* dan arsitektur berbasis komponen. Ini berarti setiap elemen antarmuka, seperti tombol atau formulir, dibangun sebagai komponen terpisah yang dapat digunakan kembali, membuat kode program lebih rapi dan mudah dikelola (Liaotando et al., 2024).

Kelebihan utama penggunaan Vue.js adalah performa aplikasi yang sangat ringan dan responsif. Pengalaman pengguna yang dihasilkan terasa mulus menyerupai aplikasi *mobile* asli,

yang sangat krusial untuk kenyamanan pelanggan saat melakukan pemesanan servis melalui *smartphone* (Ginting & Pandia, 2025).

2.3.7 *Database* MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang digunakan untuk menyimpan data secara terstruktur dalam tabel-tabel yang saling berelasi. MySQL dipilih karena kehandalannya, kemudahan penggunaan, dan kompatibilitasnya yang sangat baik dengan PHP dan Laravel (Aripin et al., 2025).

Karakteristik sistem ini terletak pada penggunaan bahasa standar SQL (*Structured Query Language*) untuk pengelolaan data. MySQL mampu menangani relasi data yang kompleks, seperti hubungan antara pelanggan dengan banyak Vespa, atau satu transaksi dengan banyak layanan, dengan tetap menjaga konsistensi dan integritas data (Wuladari & Fersellia, 2025).

Kelebihannya adalah kecepatan dan stabilitas performa, bahkan saat menangani volume data yang besar. Dukungan komunitas yang luas dan lisensi *open-source* menjadikannya solusi yang efisien dan hemat biaya untuk pengembangan sistem informasi skala menengah seperti pada bengkel KRGarage (Aripin et al., 2025).

2.3.8 *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada spesifikasi fungsional aplikasi tanpa memperhatikan struktur kode internalnya. Pengujian ini bertujuan untuk memastikan bahwa *input* yang diberikan menghasilkan *output* yang diharapkan sesuai dengan kebutuhan fungsional sistem yang telah didefinisikan (Wahyudi & Rahmi, 2022).

Metode ini memiliki karakteristik pengujian yang berbasis pada sudut pandang pengguna akhir. Penguji tidak perlu memahami logika pemrograman di balik layar, melainkan fokus pada apakah tombol berfungsi, apakah data tersimpan, dan apakah alur navigasi berjalan lancar sesuai skenario penggunaan sehari-hari (Wuladari & Fersellia, 2025).

Kelebihan dari pendekatan ini adalah objektivitas dan efisiensi waktu. *Black Box Testing* sangat efektif untuk mendeteksi kesalahan antarmuka, fungsi yang hilang, atau kesalahan inisialisasi data, memastikan bahwa sistem siap digunakan oleh orang awam sekalipun sebelum diluncurkan secara resmi (Wahyudi & Rahmi, 2022).

2.3.9 *User Acceptance Testing*

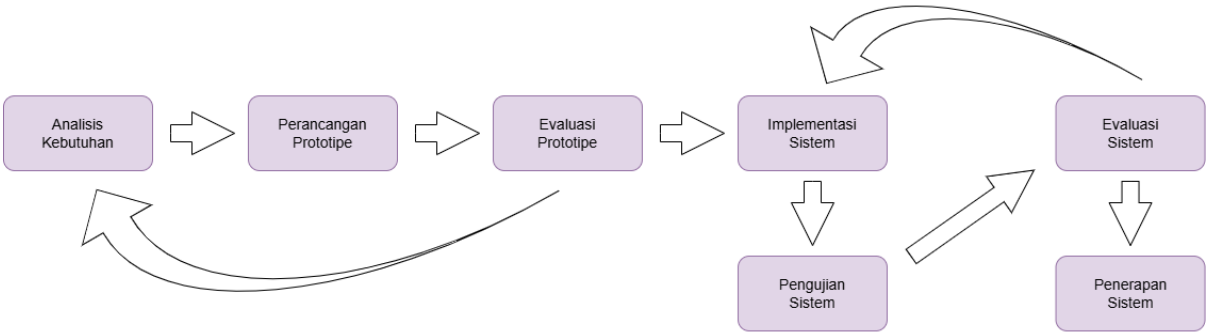
Menurut Aliyah et al. (2025), *User Acceptance Testing* (UAT) merupakan metode pengujian kelayakan sistem yang melibatkan pengguna akhir secara langsung untuk memastikan bahwa perangkat lunak yang dikembangkan telah memenuhi persyaratan fungsional dan kebutuhan pengguna yang sebenarnya. Pengujian ini dilaksanakan setelah pengujian tingkat sistem selesai dilakukan, dengan tujuan untuk memvalidasi kualitas perangkat lunak dari sudut pandang pemangku kepentingan yang akan berinteraksi langsung dengan aplikasi sehari-hari. Dengan demikian, tahapan ini berfungsi untuk mengetahui tingkat penerimaan pengguna terhadap sistem serta memastikan bahwa sistem telah memenuhi kebutuhan pengguna sebelum digunakan secara luas.

Secara karakteristik, metode UAT mengevaluasi tingkat kelayakan sistem dengan mengacu pada beberapa variabel utama yang krusial bagi keberhasilan operasional. Evaluasi ini mencakup aspek fungsionalitas sistem untuk menguji keakuratan laporan dan pengolahan data, kinerja sistem terkait kecepatan respon dan stabilitas, pengalaman dan tampilan antarmuka untuk kenyamanan navigasi, serta efisiensi dan produktivitas untuk melihat kontribusi sistem dalam memangkas waktu kerja. Penilaian terhadap variabel-variabel pengujian tersebut diturunkan ke dalam instrumen kuesioner menggunakan Skala Likert 1-5 untuk memperoleh data persepsi pengguna secara kuantitatif, terstruktur, dan objektif. Hasil penilaian kuesioner kemudian diolah dengan menghitung total skor aktual dan membandingkannya dengan skor maksimal ideal untuk memperoleh nilai persentase tingkat penerimaan pengguna terhadap sistem. Nilai persentase tersebut selanjutnya dikonversi ke dalam kategori interpretasi tertentu untuk menentukan tingkat kelayakan sistem.

Kelebihan utama dari implementasi *User Acceptance Testing* adalah memberikan kepastian ilmiah bahwa sistem informasi yang dibangun benar-benar relevan dengan kebutuhan bisnis di lapangan dan memfasilitasi ekspektasi pengguna. Melalui pengujian ini, pengembang dapat mendeteksi sisa kesalahan logika yang belum teridentifikasi pada tahapan pengujian internal sebelumnya, sekaligus menghasilkan dokumen autentik yang menjadi bukti penerimaan pengguna. Validasi langsung dari pengguna akhir ini secara signifikan meningkatkan kualitas produk, meminimalkan risiko hambatan adaptasi teknologi, serta mendukung kelancaran produktivitas kerja secara keseluruhan.

BAB III
METODE PENELITIAN

Bab ini menguraikan metodologi penelitian yang digunakan dalam pengembangan Sistem Informasi Manajemen Layanan Bengkel KRGarage. Berdasarkan karakteristik kebutuhan sistem yang dinamis dan memerlukan keterlibatan aktif pengguna dalam memvisualisasikan fitur, penelitian ini menerapkan metode *Prototyping*. Metode ini dipilih karena pendekatannya yang iteratif dan partisipatif, di mana pengembang dan pengguna dapat berinteraksi secara intensif selama proses perancangan. Hal ini bertujuan untuk memastikan bahwa sistem yang dibangun benar-benar sesuai dengan kebutuhan operasional bengkel dan mudah digunakan. Alur penelitian yang dilaksanakan terdiri dari tujuh tahapan sistematis seperti yang digambarkan pada Gambar 3.1.



Gambar 3.1 Alur Metode *Prototyping*

3.1 Analisis Kebutuhan

Tahap ini merupakan langkah awal penelitian yang bertujuan untuk mengidentifikasi permasalahan operasional secara mendalam dan mendefinisikan kebutuhan sistem secara komprehensif. Penulis melakukan pengumpulan data melalui pendekatan kualitatif, yaitu wawancara mendalam dan observasi partisipatif dengan para pemangku kepentingan utama di KRGarage, meliputi pemilik, admin, mekanik, dan pelanggan. Aktivitas utama pada tahap ini diawali dengan analisis proses bisnis berjalan untuk memetakan alur kerja mulai dari pendaftaran hingga pembayaran, yang bertujuan menemukan titik inefisiensi dan risiko kesalahan data. Hasil analisis menunjukkan bahwa ketergantungan pada pencatatan manual dan komunikasi asinkron via WhatsApp tidak hanya menyebabkan tingginya latensi respon admin, tetapi juga meningkatkan risiko kesalahan perhitungan dalam proses rekapitulasi transaksi.

Selanjutnya, identifikasi masalah per aktor mengungkapkan kendala yang berbeda-beda. Admin menghadapi kesulitan sinkronisasi jadwal yang sering berujung pada benturan jadwal serta inefisiensi waktu saat harus menghitung total biaya jasa dan harga suku cadang secara manual menggunakan kalkulator, yang seringkali menyebabkan selisih data. Sementara itu, pemilik kesulitan dalam pemantauan kinerja bisnis akibat ketidakmampuan melihat data keuangan harian secara *real-time* dan laporan keuangan yang tidak akurat karena pencatatan manual. Di sisi lain, pelanggan mengeluhkan minimnya transparansi estimasi biaya di awal dan seringnya kehilangan nota fisik riwayat perawatan, yang berdampak pada ketidakpuasan layanan.

Berdasarkan pemetaan masalah tersebut, penulis merumuskan spesifikasi kebutuhan sistem yang mencakup kebutuhan fungsional seperti fitur pemesanan servis *online*, manajemen data Vespa, mekanisme kalkulasi transaksi otomatis (biaya jasa dan penggunaan suku cadang), serta *dashboard* statistik pendapatan. Selain itu, didefinisikan pula kebutuhan non-fungsional yang meliputi aspek keamanan data, kecepatan akses, dan responsivitas tampilan antarmuka pada perangkat *mobile* untuk menjamin kenyamanan pengguna.

Untuk memastikan spesifikasi sistem yang dirancang relevan dengan permasalahan nyata di lapangan, penulis merangkum poin-poin krusial dari hasil wawancara dengan pemilik dan admin KRGarage. Transkrip wawancara tersebut kemudian dipetakan menjadi kebutuhan fitur spesifik sebagaimana disajikan pada Tabel 3.1 berikut:

Tabel 3.1 Rekapitulasi Hasil Wawancara dan Analisis Kebutuhan

Narasumber & Fokus Masalah	**Intisari Jawaban Narasumber**	**Implikasi Kebutuhan Fungsional Sistem**
Pemilik (Laporan Keuangan)	1. Pencatatan tidak rapi karena mencatat di buku atau di hp. 2. Butuh laporan pendapatan bulanan dan pengeluaran.	Dibutuhkan fitur penyusunan laporan pendapatan, pengeluaran, dan keuntungan secara otomatis.
Admin (Efisiensi Data)	1. Membutuhkan waktu yang agak lama untuk mencari rekap pelanggan. 2. Riwayat hanya ada di salinan bon fisik.	Dibutuhkan fitur pengelolaan dan pencarian riwayat servis pelanggan secara terpusat.
Admin (Manajemen Stok)	1. Pernah terjadi kekurangan stok saat dibutuhkan. 2. Penting memiliki sistem yang memberi notifikasi jika stok menipis.	Dibutuhkan fitur pengelolaan stok suku cadang serta notifikasi ketika jumlah stok mencapai batas minimum.
Mekanik (Operasional)	1. Sistem saat ini kurang efisien. 2. Butuh info jenis servis dan keluhan pelanggan di layar.	Dibutuhkan antarmuka yang menampilkan informasi pekerjaan servis, keluhan pelanggan, dan status pengerjaan yang dapat diakses melalui perangkat *mobile*.

	3. Lebih nyaman pakai *handphone*.	
Pelanggan (Retensi & Loyalitas)	1. Penting sekali melihat riwayat servis secara *online* untuk menjaga keawetan mesin. 2. Saya perlu pengingat servis.	Dibutuhkan fitur pengelolaan data Vespa, akses riwayat servis, serta pengingat jadwal servis secara berkala.

3.2 Perancangan Prototipe

Setelah kebutuhan sistem berhasil diidentifikasi, tahap selanjutnya adalah perancangan prototipe sebagai representasi awal sistem yang akan dikembangkan. Pada tahap ini, penulis menerjemahkan kebutuhan pengguna ke dalam bentuk rancangan antarmuka yang dapat divisualisasikan dan dievaluasi sebelum proses implementasi sistem dilakukan. Prototipe berfungsi sebagai media komunikasi antara pengembang dan pengguna untuk memastikan bahwa fitur serta alur kerja yang dirancang telah sesuai dengan kebutuhan operasional bengkel.

Perancangan prototipe dilakukan menggunakan perangkat lunak Figma dengan menghasilkan *wireframe* dan *mockup* dari setiap halaman utama sistem. Rancangan dibuat berdasarkan kebutuhan masing-masing aktor yang telah diperoleh pada tahap analisis kebutuhan, yaitu pelanggan, admin, mekanik, dan pemilik bengkel. Setiap antarmuka dirancang dengan mempertimbangkan kemudahan penggunaan, kejelasan informasi, dan efisiensi proses kerja agar dapat mendukung aktivitas operasional bengkel secara optimal.

Pada sisi pelanggan, prototipe difokuskan pada kemudahan proses pemesanan servis, pengelolaan data Vespa, serta akses terhadap riwayat servis. Pada sisi admin, dirancang halaman yang mendukung pengelolaan pemesanan servis, akses informasi pelanggan melalui detail pemesanan, pengelolaan layanan jasa, serta pengelolaan suku cadang. Untuk mekanik, dirancang antarmuka yang menampilkan daftar pekerjaan, informasi keluhan pelanggan, serta fitur pembaruan status pengerjaan. Sementara itu, pada sisi pemilik bengkel dirancang *dashboard* yang menyajikan informasi pendapatan, pengeluaran, dan keuntungan secara ringkas untuk membantu proses pengambilan keputusan.

Hasil perancangan prototipe ini kemudian digunakan sebagai bahan evaluasi bersama pengguna pada tahap selanjutnya. Melalui proses tersebut, pengguna dapat memberikan masukan terkait tampilan, navigasi, maupun kebutuhan fitur sehingga rancangan yang dihasilkan dapat lebih sesuai dengan kebutuhan operasional KRGarage sebelum dikembangkan menjadi sistem yang sesungguhnya.

3.3 Evaluasi Prototipe

Tahap ini merupakan fase penting dalam metode *Prototyping* yang bertujuan untuk memvalidasi rancangan sistem dan menyamakan persepsi antara pengembang dan pengguna sebelum proses implementasi sistem dimulai. Penulis mendemonstrasikan rancangan antarmuka yang telah dibuat kepada pengguna. Tujuannya adalah untuk memastikan bahwa alur navigasi, tata letak antarmuka, dan fitur yang dirancang telah sesuai dengan kebutuhan operasional bengkel.

Proses evaluasi dilakukan melalui diskusi interaktif, di mana pengguna diminta untuk meninjau dan mencoba simulasi alur penggunaan sistem yang telah dirancang. Skenario evaluasi mencakup proses pemesanan servis, pengelolaan antrean servis, pencatatan transaksi, hingga penyajian informasi laporan. Fokus utama pada tahap ini adalah memastikan bahwa alur bisnis, navigasi sistem, dan kebutuhan fungsional yang dirancang telah sesuai dengan harapan pengguna serta mampu mendukung aktivitas operasional bengkel secara efektif.

Penulis mencatat setiap masukan, kritik, maupun saran yang diberikan pengguna selama proses evaluasi. Hasil evaluasi tersebut menjadi dasar dalam menentukan langkah pengembangan berikutnya.

a. Jika prototipe dinilai telah memenuhi kebutuhan pengguna dan mudah digunakan, maka rancangan dianggap valid dan dapat dilanjutkan ke tahap implementasi sistem.
b. Apabila pengguna menemukan ketidaksesuaian alur, tampilan, atau kebutuhan fitur yang belum terpenuhi, maka penulis akan kembali ke tahap Perancangan Prototipe untuk melakukan perbaikan. Proses ini dilakukan secara berulang hingga diperoleh rancangan yang sesuai dengan kebutuhan pengguna.

3.4 Implementasi Sistem

Tahap implementasi sistem dilaksanakan setelah rancangan prototipe dinyatakan valid melalui proses evaluasi bersama pengguna. Pada tahap ini, seluruh rancangan antarmuka dan kebutuhan fungsional yang telah disepakati diterjemahkan ke dalam bentuk aplikasi yang dapat dijalankan dan digunakan oleh pengguna.

Sebelum proses implementasi dilakukan, penulis menyusun rancangan teknis sistem sebagai acuan pengembangan. Rancangan teknis tersebut meliputi pemodelan kebutuhan menggunakan *Use Case Diagram*, pemodelan alur proses menggunakan *Activity Diagram*, serta perancangan struktur basis data menggunakan *Entity Relationship Diagram* (ERD). Rancangan ini digunakan untuk memastikan bahwa proses implementasi berjalan sesuai

dengan kebutuhan pengguna dan proses bisnis yang telah diidentifikasi pada tahap analisis kebutuhan.

Implementasi sistem dilakukan dengan menerapkan arsitektur *client-server* yang memisahkan antarmuka pengguna dan logika bisnis aplikasi. Antarmuka sistem dikembangkan menggunakan Vue.js untuk menghasilkan tampilan yang responsif serta mudah digunakan pada berbagai ukuran perangkat. Sementara itu, layanan *backend* dikembangkan menggunakan *framework* Laravel yang berfungsi untuk mengelola proses bisnis, autentikasi pengguna, serta pertukaran data antara *frontend* dan basis data melalui REST API.

Pada tahap ini, penulis mengimplementasikan seluruh fitur utama yang telah dirancang pada prototipe, meliputi fitur pemesanan servis *online*, akses informasi pelanggan melalui data pemesanan, pengelolaan data Vespa, manajemen layanan jasa, pengelolaan suku cadang, pencatatan transaksi servis, pengelolaan status pengerjaan, serta *dashboard* informasi untuk admin, mekanik, dan pemilik bengkel. Selain itu, diimplementasikan pula mekanisme kalkulasi otomatis yang menggabungkan biaya jasa dan penggunaan suku cadang untuk menghasilkan total transaksi secara akurat.

Untuk mendukung penyimpanan dan pengelolaan data, sistem menggunakan basis data MySQL yang dirancang berdasarkan kebutuhan operasional bengkel. Basis data tersebut menyimpan berbagai informasi yang dibutuhkan sistem, seperti data pengguna, Vespa, layanan, suku cadang, pemesanan, transaksi, serta data pendukung lainnya yang saling terhubung melalui relasi tabel. Implementasi basis data ini bertujuan untuk menjaga konsistensi, integritas, dan keakuratan data yang digunakan dalam proses operasional maupun penyusunan laporan.

Hasil dari tahap implementasi ini berupa Sistem Informasi Manajemen Layanan Bengkel KRGarage yang telah mengakomodasi kebutuhan pengguna dan siap untuk memasuki tahap pengujian sistem.

3.5 Pengujian Sistem

Tahap pengujian merupakan fase krusial yang dilaksanakan setelah proses implementasi selesai. Tujuan utama tahap ini adalah memverifikasi bahwa seluruh komponen perangkat lunak telah berfungsi sesuai dengan spesifikasi kebutuhan yang didefinisikan pada tahap awal, serta memastikan sistem bebas dari kesalahan logika maupun kegagalan sistem.

Dalam penelitian ini, strategi pengujian yang diterapkan adalah *Black Box Testing*. Sesuai dengan landasan teori yang dipaparkan pada Bab II, metode ini berfokus pada pengujian

fungsionalitas antarmuka dan logika bisnis tanpa perlu memeriksa struktur kode internal program. Penulis memposisikan diri sebagai pengguna akhir yang memberikan berbagai variasi *input* data, baik data valid maupun data *in*valid untuk melihat apakah *output* yang dihasilkan sesuai dengan hasil yang diharapkan.

Rangkaian pengujian ini disusun dalam dokumen skenario uji yang mencakup validasi terhadap empat modul utama sistem KRGarage, yaitu:

a. Pengujian Keamanan dan Akses

Pengujian ini bertujuan memvalidasi mekanisme *multi-role user*. Skenario yang dijalankan meliputi percobaan login dengan kombinasi email dan *password* yang salah, serta pengujian batasan hak akses. Penulis memastikan bahwa akun dengan peran Pelanggan tidak dapat mengakses halaman Admin, dan sebaliknya, untuk menjamin keamanan data sensitif bengkel.

b. Pengujian Transaksional dan Kalkulasi Otomatis

Ini merupakan prioritas pengujian tertinggi untuk menjawab rumusan masalah mengenai kesalahan hitung manual. Skenario pengujian dipusatkan pada validasi akurasi perhitungan *real-time* dan logika perubahan status pengerjaan. Sistem diuji kemampuannya dalam melakukan kalkulasi ulang total tagihan secara otomatis tanpa kesalahan aritmatika setiap kali mekanik melakukan penambahan atau pengurangan item suku cadang pada pesanan yang aktif. Selanjutnya, indikator keberhasilan pengujian ditentukan oleh ketepatan mekanisme pemicu laporan keuangan, di mana sistem dinyatakan valid apabila sistem mampu mencatat pengeluaran saat terjadi *restock*, serta data pendapatan hanya masuk ke dalam kalkulasi keuntungan secara otomatis ketika admin mengubah status pembayaran transaksi menjadi Lunas.

c. Pengujian Logika Inventori

Pengujian dilakukan untuk memvalidasi integrasi antara modul kasir dan modul gudang. Skenario dilakukan dengan menyelesaikan transaksi servis yang menggunakan suku cadang tertentu. Indikator keberhasilan adalah apabila stok di *database* berkurang secara otomatis sesuai jumlah yang digunakan dalam transaksi tersebut, serta munculnya notifikasi sistem apabila stok mencapai batas minimum.

d. Pengujian Alur Pemesanan Servis dan Responsivitas Antarmuka

Pengujian difokuskan pada validasi formulir pemesanan servis di sisi pelanggan, termasuk pengecekan logika tanggal. Selain itu, dilakukan pula pengujian responsivitas dengan

mengakses sistem melalui berbagai resolusi layar gawai (*smartphone*, tablet, dan *desktop*) untuk memastikan elemen antarmuka tetap rapi dan mudah dioperasikan.

Hasil dari setiap skenario uji akan dicatat dalam tabel hasil pengujian. Apabila ditemukan ketidaksesuaian antara hasil aktual dengan hasil yang diharapkan, sistem akan dikembalikan ke tahap implementasi untuk menjalani proses perbaikan. Selanjutnya, pengujian dilakukan kembali secara berulang hingga seluruh fungsi dinyatakan valid dan lulus uji sepenuhnya.

3.6 Evaluasi Sistem

Setelah sistem dinyatakan lulus uji fungsional secara teknis, tahap selanjutnya adalah evaluasi sistem yang melibatkan pengguna akhir secara langsung. Dalam penelitian ini, evaluasi dilakukan menggunakan metode *User Acceptance Testing* (UAT) untuk mengetahui tingkat penerimaan pengguna terhadap sistem yang dikembangkan serta menilai kesesuaiannya dengan kebutuhan operasional bengkel KRGarage.

Pelaksanaan UAT melibatkan perwakilan pengguna yang terdiri atas pemilik, admin, mekanik, dan pelanggan. Setiap responden diminta untuk mencoba fitur sistem sesuai dengan peran masing-masing, kemudian memberikan penilaian melalui instrumen kuesioner yang disusun berdasarkan skala Likert lima tingkat, yaitu Sangat Setuju (5), Setuju (4), Netral (3), Tidak Setuju (2), dan Sangat Tidak Setuju (1).

Instrumen evaluasi disusun untuk mengukur empat variabel utama, yaitu fungsionalitas sistem, kinerja sistem, tampilan antarmuka sistem, serta efisiensi dan produktivitas sistem. Penilaian dari seluruh responden kemudian diolah untuk memperoleh nilai persentase tingkat penerimaan pengguna terhadap sistem yang dikembangkan.

Hasil evaluasi digunakan untuk menentukan apakah sistem telah memenuhi kebutuhan pengguna dan siap diterapkan secara bertahap pada lingkungan operasional bengkel. Apabila hasil evaluasi menunjukkan tingkat penerimaan yang baik, maka sistem dinyatakan layak digunakan. Sebaliknya, apabila masih ditemukan kekurangan berdasarkan masukan pengguna, maka dilakukan perbaikan sebelum sistem diterapkan secara penuh.

3.7 Penerapan Sistem

Tahap penerapan sistem merupakan tahap akhir dalam metode *Prototyping*, yaitu ketika sistem yang telah melalui proses implementasi, pengujian, dan evaluasi dinyatakan siap untuk diterapkan secara bertahap oleh pengguna. Pada penelitian ini, tahap penerapan sistem

difokuskan pada kesiapan awal sistem untuk diterapkan pada lingkungan operasional KRGarage.

Aktivitas yang dilakukan pada tahap ini meliputi penyiapan sistem dan pengisian data awal yang dibutuhkan sebagai dasar penggunaan sistem. Data awal yang dimasukkan mencakup data pengguna, data layanan servis, dan data suku cadang. Data tersebut digunakan sebagai data master awal agar fitur utama sistem dapat dijalankan sesuai alur yang telah dirancang.

Melalui tahap ini, Sistem Informasi Manajemen Layanan Bengkel KRGarage diharapkan telah berada pada kondisi siap untuk diterapkan secara bertahap. Sistem dapat digunakan sebagai dasar digitalisasi pengelolaan layanan bengkel, khususnya dalam mendukung pengelolaan layanan servis, data suku cadang, dan hak akses pengguna sesuai perannya.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui observasi dan wawancara dengan pihak yang terlibat dalam operasional bengkel KRGarage, yaitu pemilik, admin, mekanik, dan pelanggan. Kegiatan ini bertujuan untuk mengidentifikasi permasalahan yang terjadi pada proses bisnis bengkel serta menentukan kebutuhan sistem yang akan dikembangkan. Hasil analisis kebutuhan menjadi dasar dalam perancangan prototipe dan implementasi Sistem Informasi Manajemen Layanan Bengkel KRGarage.

Berdasarkan hasil pengumpulan data, diperoleh beberapa kebutuhan fungsional yang berasal dari pihak internal bengkel dan pelanggan. Kebutuhan tersebut kemudian diterjemahkan menjadi fitur-fitur yang diimplementasikan pada sistem untuk mendukung aktivitas operasional bengkel secara lebih efektif dan efisien.

4.1.1 Kebutuhan Pihak Bengkel

Kebutuhan pihak bengkel diperoleh dari hasil wawancara dengan pemilik, admin, dan mekanik. Hasil analisis menunjukkan bahwa permasalahan utama yang dihadapi meliputi kesulitan dalam mengakses informasi pelanggan dan riwayat servis, pencatatan transaksi, pengelolaan stok suku cadang, serta pemantauan aktivitas operasional bengkel. Ringkasan kebutuhan tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Analisis Kebutuhan Fungsional Pihak Bengkel

Fitur	Deskripsi	Kebutuhan
Dashboard & Laporan	Sistem menampilkan ringkasan pendapatan, pengeluaran, dan statistik layanan secara otomatis	Menyediakan fitur laporan keuangan harian/bulanan yang dapat diakses pemilik secara *real-time*
Kelola Pemesanan	Admin dapat melihat dan mengelola pemesanan servis yang masuk	Fitur pengelolaan pemesanan servis untuk memudahkan proses konfirmasi dan pemantauan jadwal servis
Manajemen Stok	Admin dapat mengelola data suku cadang dan memantau ketersediaan barang	Fitur CRUD data barang, *update* stok otomatis saat transaksi, dan notifikasi stok menipis
Operasional Mekanik	Mekanik dapat melihat antrean pekerjaan, detail keluhan pelanggan, dan memperbarui status pengerjaan servis	Menyediakan tampilan daftar pekerjaan servis beserta fitur pembaruan status pengerjaan
Kelola Transaksi	Admin dapat mencatat transaksi pembayaran akhir	Fitur kasir sederhana untuk menghitung total biaya servis dan suku cadang

Tabel 4.1 menunjukkan bahwa kebutuhan utama pihak bengkel meliputi pengelolaan pemesanan servis, akses informasi pelanggan melalui data pemesanan, pengelolaan suku cadang, pencatatan transaksi servis, pemantauan aktivitas mekanik, serta penyajian laporan operasional dan keuangan. Kebutuhan tersebut kemudian dijadikan dasar dalam perancangan dan implementasi fitur pada Sistem Informasi Manajemen Layanan Bengkel KRGarage.

4.1.2 Kebutuhan Pelanggan

Selain kebutuhan pihak internal bengkel, analisis kebutuhan juga dilakukan terhadap pelanggan sebagai pengguna layanan. Hasil analisis menunjukkan bahwa pelanggan membutuhkan kemudahan dalam melakukan pemesanan servis, mengakses riwayat Vespa, pengelolaan data Vespa, informasi status pengerjaan, serta pengingat jadwal servis berkala. Ringkasan kebutuhan pelanggan disajikan pada Tabel 4.2.

Tabel 4.2 Hasil Analisis Kebutuhan Fungsional Pelanggan

Fitur	Deskripsi	Kebutuhan
Pemesanan Servis	Pelanggan dapat melakukan pemesanan servis melalui sistem tanpa harus menunggu balasan chat manual	Menyediakan formulir pemilihan layanan, Vespa, tanggal, dan jam servis
Riwayat Servis	Pelanggan dapat melihat kembali data servis yang pernah dilakukan pada Vespa mereka	Menampilkan informasi tanggal servis, jenis layanan, status servis, dan suku cadang yang digunakan
Vespa Saya	Pelanggan dapat menyimpan dan mengelola data Vespa mereka ke dalam akun	Menyediakan fitur untuk menambahkan, mengedit, dan menyimpan data Vespa agar dapat digunakan pada proses pemesanan servis berikutnya
Status Pengerjaan	Pelanggan dapat memantau progres servis Vespa mereka melalui sistem	Menampilkan status pengerjaan servis pada *dashboard* maupun riwayat pemesanan pelanggan
Pengingat Servis	Sistem menampilkan pengingat ketika Vespa telah memasuki atau melewati jadwal servis berikutnya	Menyediakan pengingat servis berkala pada *dashboard* pelanggan berdasarkan tanggal servis berikutnya yang tersimpan pada data Vespa
Info Layanan	Pelanggan dapat melihat informasi layanan dan harga sebelum melakukan pemesanan servis	Menyediakan informasi daftar layanan servis yang tersedia secara transparan

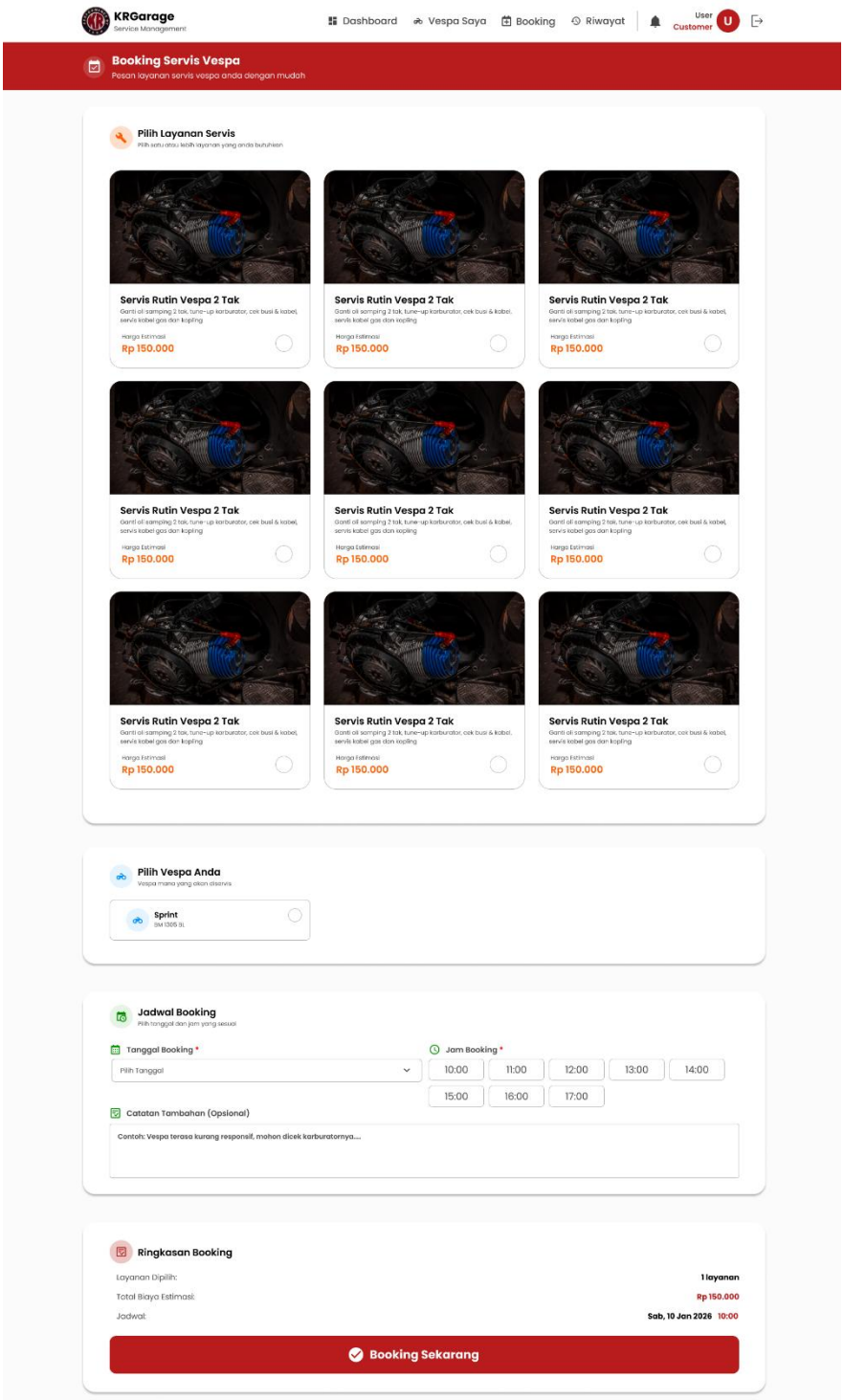
Tabel 4.2 menunjukkan bahwa kebutuhan utama pelanggan berfokus pada kemudahan akses layanan, transparansi informasi, dan ketersediaan riwayat servis secara digital. Kebutuhan tersebut kemudian dijadikan dasar dalam perancangan fitur pelanggan, seperti pemesanan servis, fitur Vespa Saya, riwayat pemesanan, status pengerjaan, dan pengingat servis berkala.

4.2 Perancangan Prototipe

Setelah tahap analisis kebutuhan selesai dilakukan, tahap berikutnya adalah perancangan prototipe sistem. Perancangan dilakukan menggunakan Figma untuk menghasilkan *wireframe* dan *mockup* yang menggambarkan antarmuka serta alur penggunaan sistem. Prototipe dirancang berdasarkan kebutuhan masing-masing pengguna, yaitu pelanggan, admin, mekanik, dan pemilik bengkel. Hasil perancangan ini kemudian digunakan sebagai acuan dalam proses implementasi sistem.

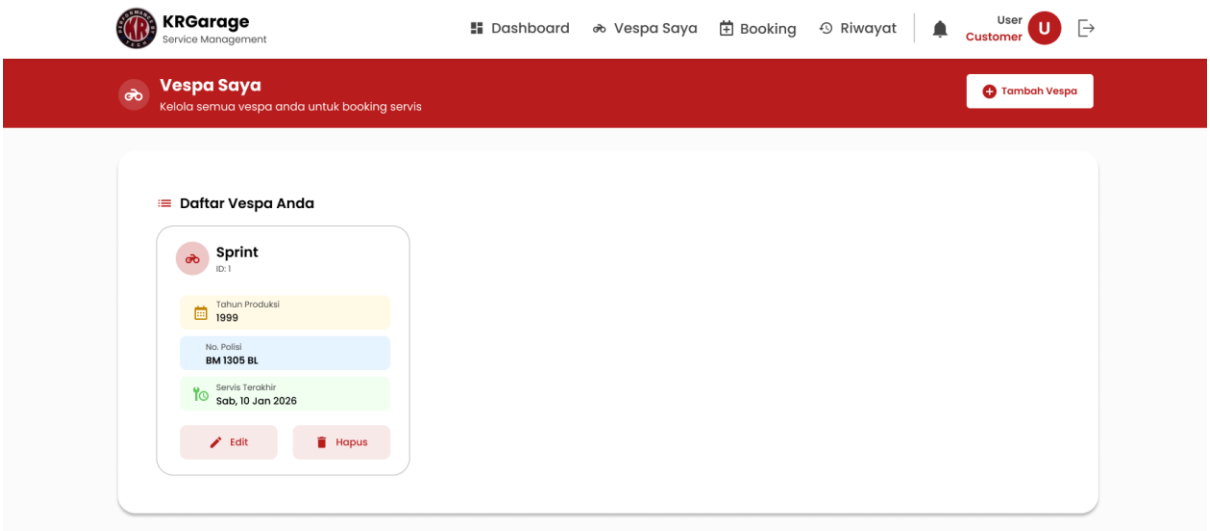
4.2.1 Prototipe Pelanggan

Prototipe pelanggan dirancang untuk memudahkan pelanggan dalam melakukan pemesanan servis, mengelola data Vespa, memantau riwayat servis, serta memperoleh informasi pengingat servis berkala. Perancangan antarmuka dilakukan dengan mengutamakan kemudahan penggunaan dan akses informasi yang dibutuhkan pelanggan selama menggunakan layanan bengkel KRGarage.



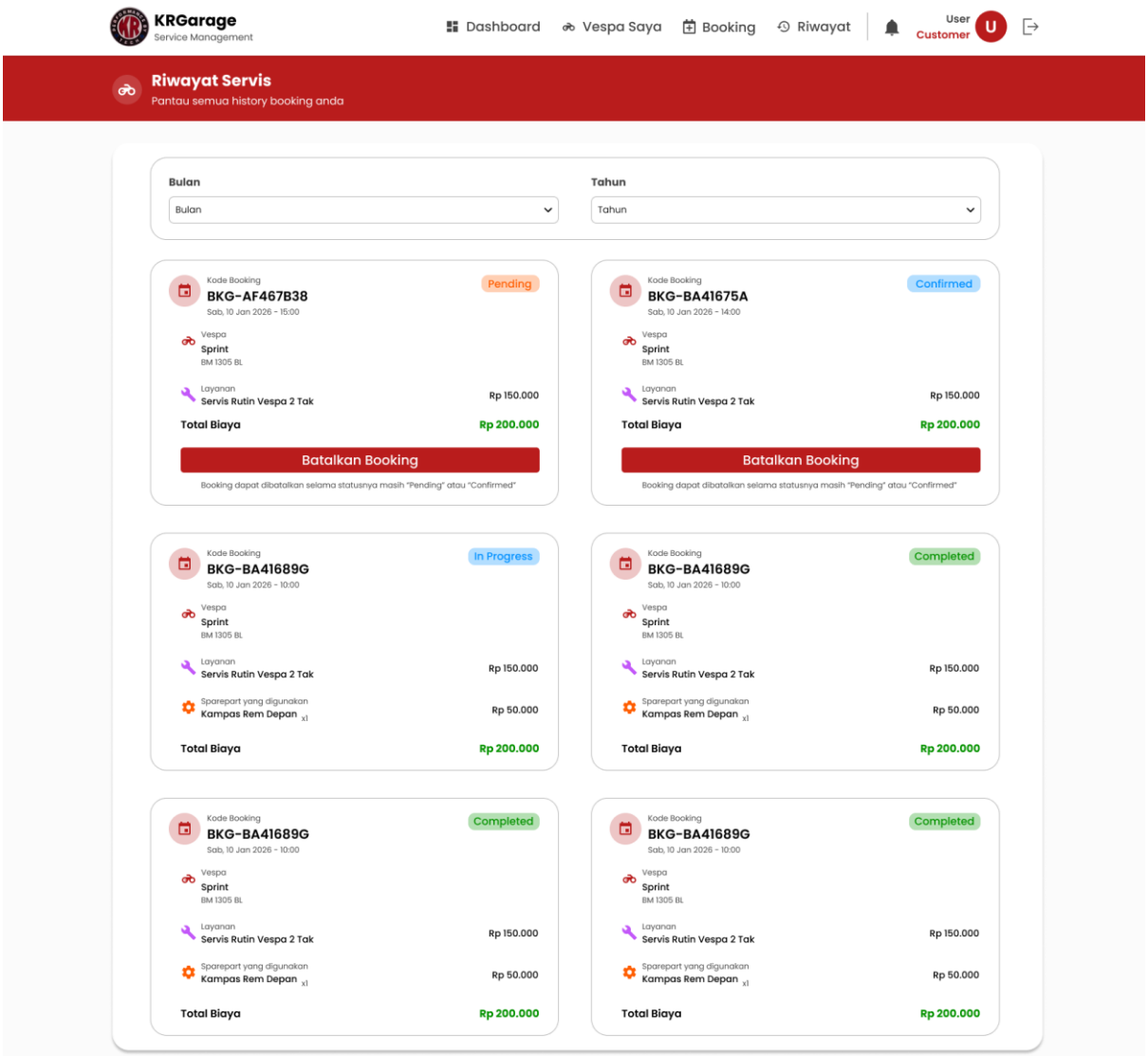
Gambar 4.1 Halaman Pemesanan Servis Pelanggan

Halaman pemesanan servis pada Gambar 4.1 dirancang untuk memfasilitasi pelanggan dalam melakukan pemesanan servis secara *online*. Pelanggan dapat memilih layanan servis yang tersedia, menentukan Vespa yang akan diservis, memilih tanggal dan jam pemesanan, serta menambahkan catatan keluhan atau kebutuhan servis. Sistem juga menampilkan ringkasan pemesanan sebagai konfirmasi sebelum pelanggan mengirimkan permintaan servis.



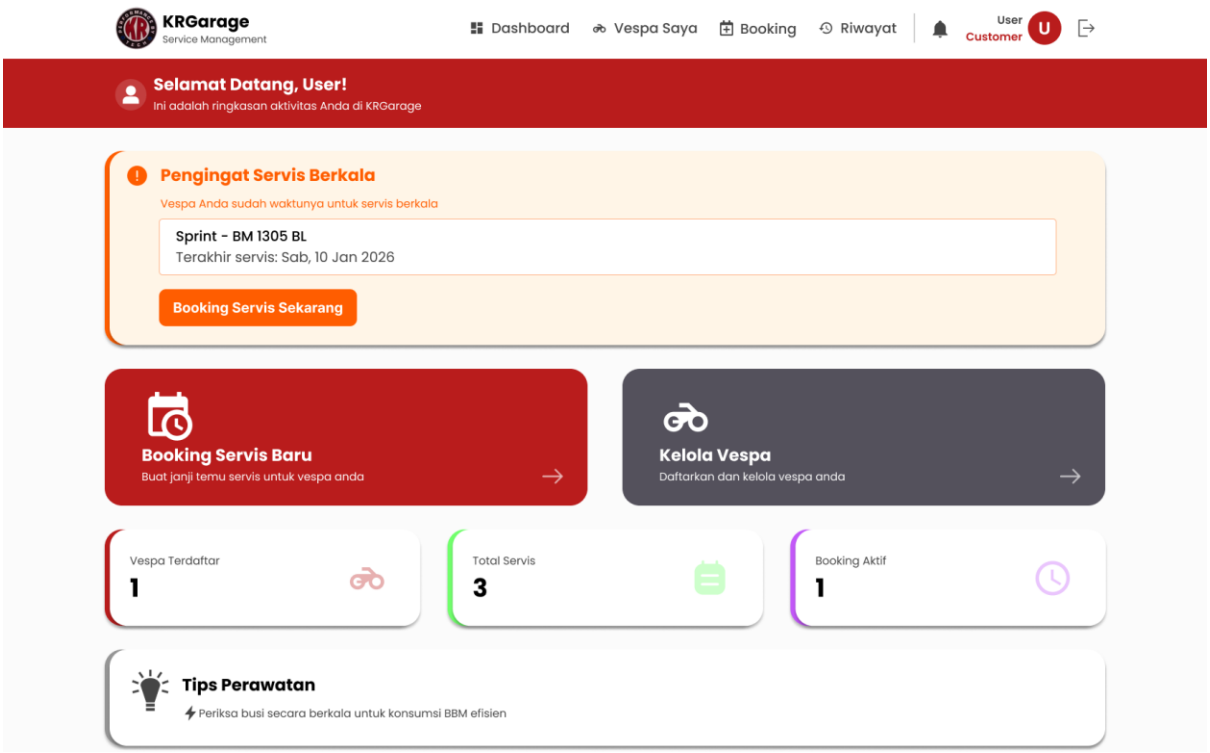
Gambar 4.2 Halaman Vespa Pelanggan

Pengelolaan data Vespa pelanggan ditampilkan melalui halaman Vespa Saya pada Gambar 4.2. Halaman ini menyediakan informasi identitas Vespa, nomor polisi, tahun produksi, serta riwayat servis terakhir. Selain itu, pelanggan dapat menambahkan, mengubah, maupun menghapus data Vespa yang tersimpan pada sistem.



Gambar 4.3 Halaman Riwayat Servis Pelanggan

Halaman riwayat servis pada Gambar 4.3 dirancang untuk memberikan akses terhadap data pemesanan dan servis yang pernah dilakukan pelanggan. Informasi yang ditampilkan meliputi kode pemesanan, data Vespa, layanan yang digunakan, status pengerjaan, suku cadang yang digunakan, serta total biaya servis. Halaman ini membantu pelanggan dalam melihat rekam jejak perawatan Vespa secara digital.

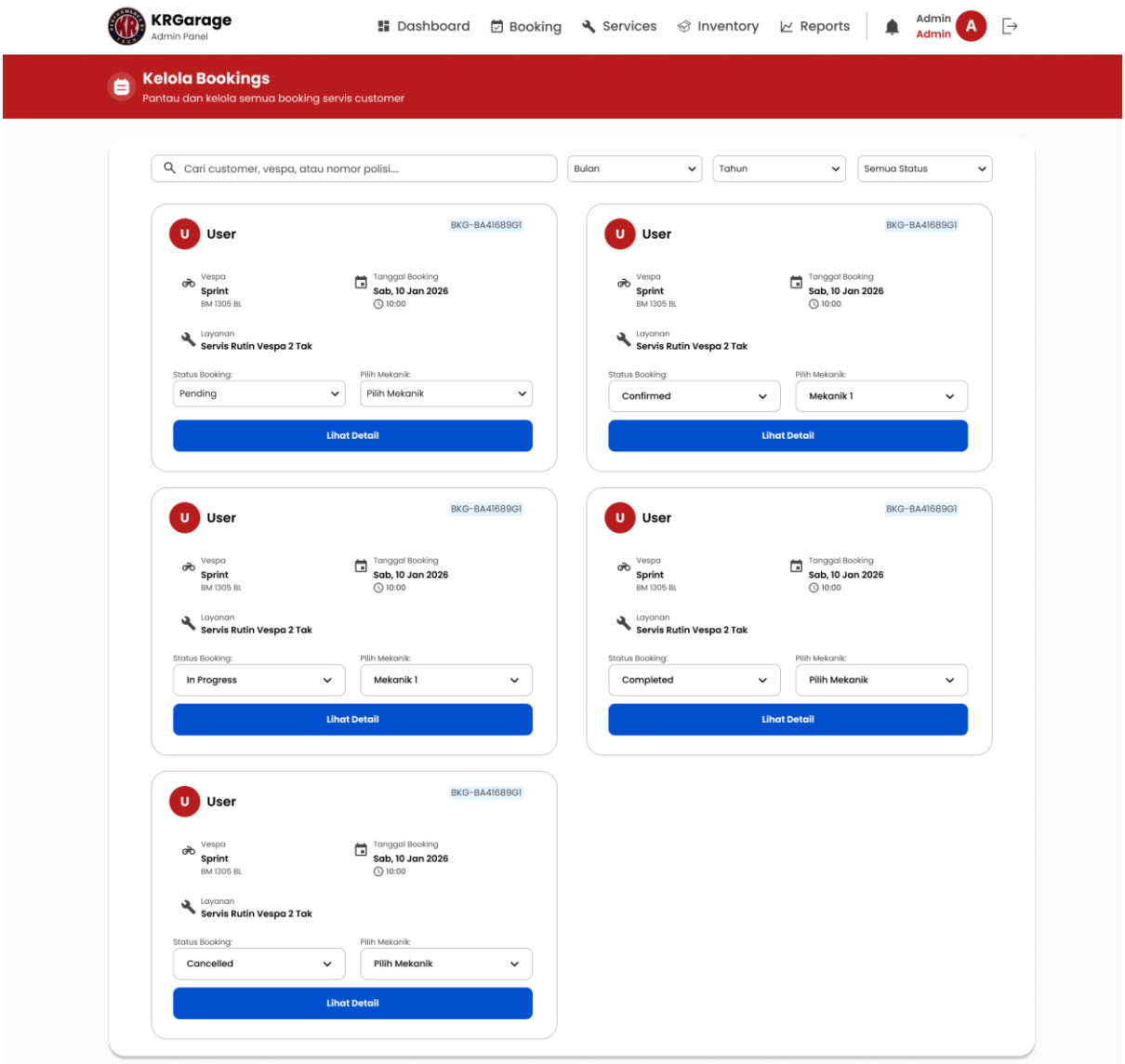


Gambar 4.4 Tampilan Pengingat Servis Pelanggan

Fitur pengingat servis berkala ditampilkan pada *dashboard* pelanggan seperti pada Gambar 4.4. Fitur ini muncul ketika Vespa telah memasuki atau melewati jadwal servis berikutnya. Informasi yang ditampilkan meliputi identitas Vespa dan tanggal servis terakhir. Fitur ini dirancang untuk membantu pelanggan menjaga keteraturan perawatan Vespa serta meningkatkan kemudahan dalam melakukan pemesanan servis kembali melalui sistem.

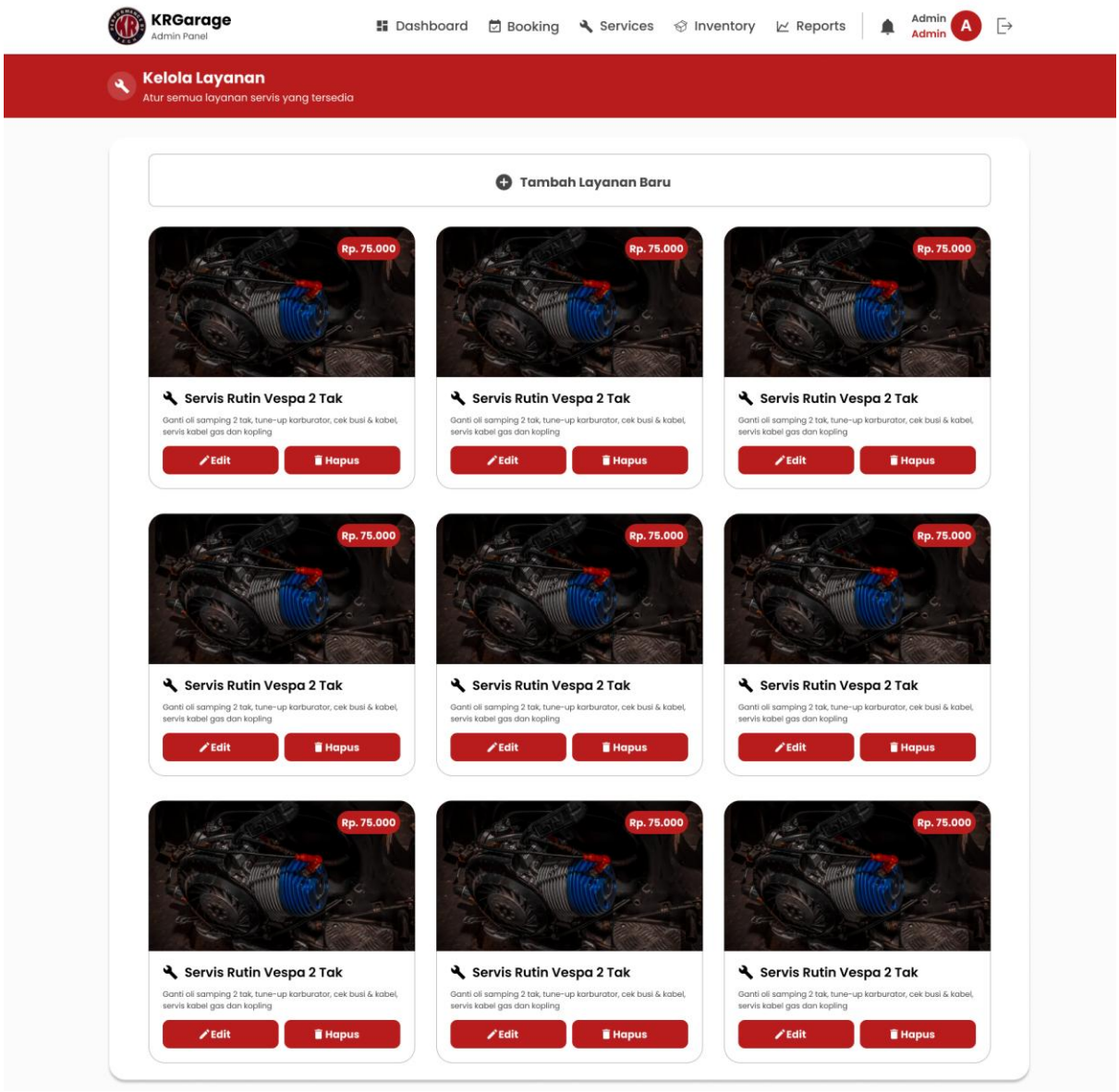
4.2.2 Prototipe Admin

Prototipe admin dirancang untuk mendukung proses pengelolaan operasional bengkel yang meliputi pengelolaan pemesanan servis, layanan servis, dan data suku cadang. Antarmuka admin dibuat agar proses administrasi dapat dilakukan secara terpusat sehingga memudahkan pemantauan dan pengelolaan aktivitas bengkel.



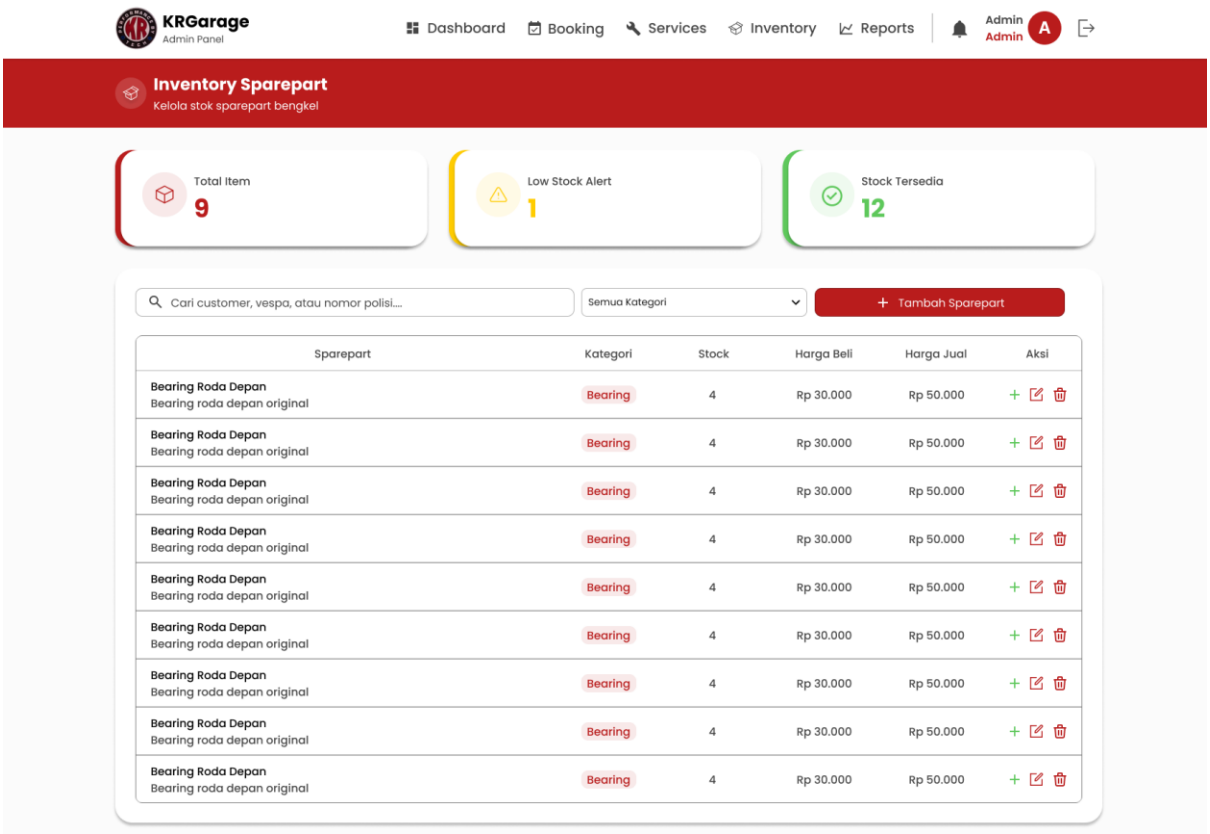
Gambar 4.5 Halaman Kelola Pemesanan Servis Admin

Halaman kelola pemesanan servis pada Gambar 4.5 digunakan untuk menampilkan seluruh data pemesanan yang masuk dari pelanggan. Admin dapat melihat informasi pelanggan, data Vespa, layanan yang dipilih, jadwal pemesanan, mengubah status pemesanan, serta menetapkan mekanik yang akan menangani proses servis.



Gambar 4.6 Halaman Kelola Layanan Admin

Pengelolaan data layanan servis dilakukan melalui halaman kelola layanan pada Gambar 4.6. Pada halaman ini, admin dapat menambahkan layanan baru, mengubah informasi layanan, maupun menghapus layanan yang sudah tidak digunakan sehingga informasi layanan yang ditampilkan kepada pelanggan tetap sesuai dengan kondisi bengkel.

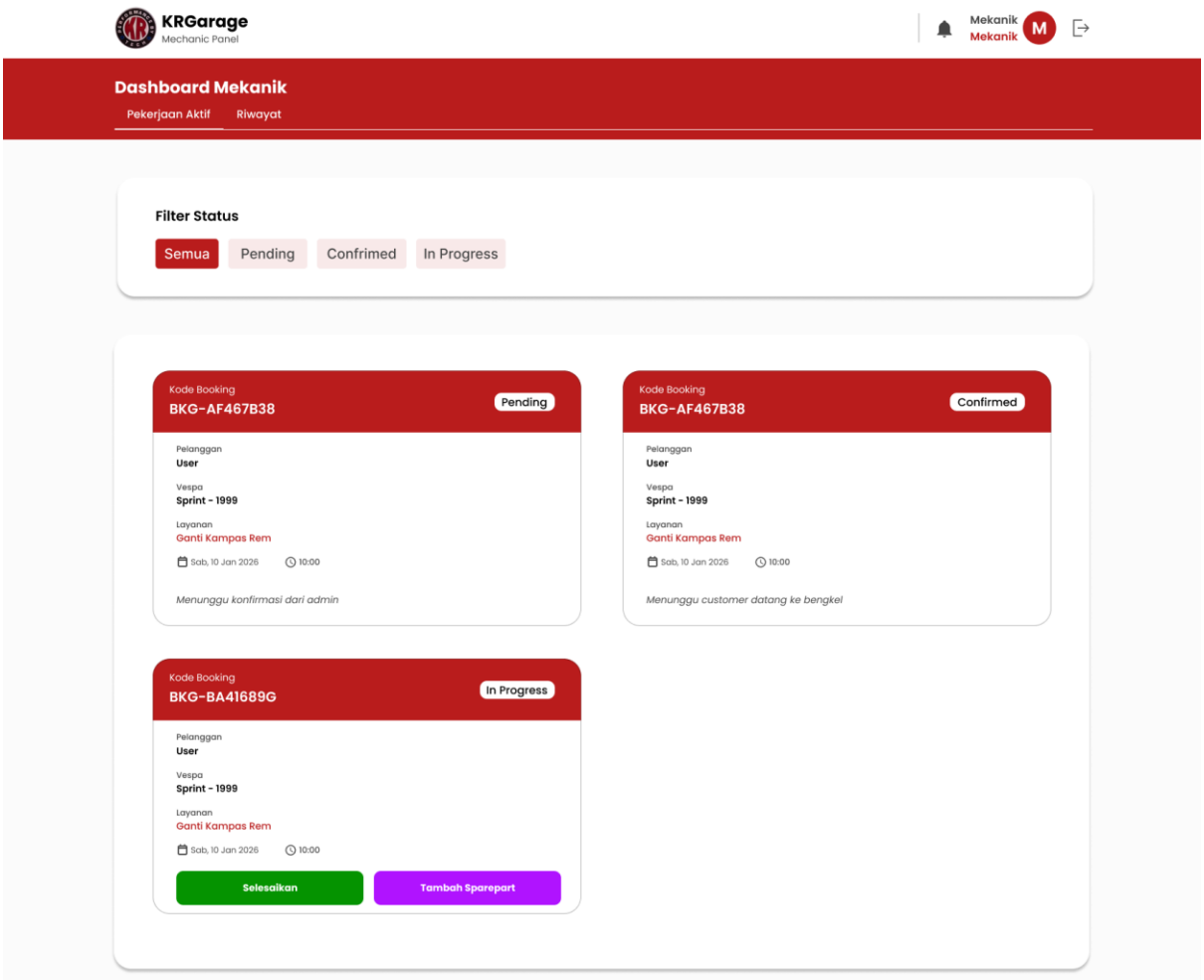


Gambar 4.7 Halaman Kelola Suku Cadang Admin

Halaman kelola suku cadang pada Gambar 4.7 digunakan untuk mengelola data suku cadang yang tersedia di bengkel. Halaman ini menampilkan informasi stok, harga beli, harga jual, dan kategori suku cadang. Selain itu, admin dapat menambahkan data suku cadang baru, melakukan perubahan data, serta melakukan *restock* suku cadang.

4.2.3 Prototipe Mekanik

Prototipe mekanik dirancang untuk membantu mekanik dalam memantau pekerjaan servis yang telah ditugaskan oleh admin. Antarmuka dibuat sederhana agar mekanik dapat dengan mudah melihat detail pekerjaan, memperbarui status servis, serta menambahkan suku cadang yang digunakan selama proses pengerjaan.

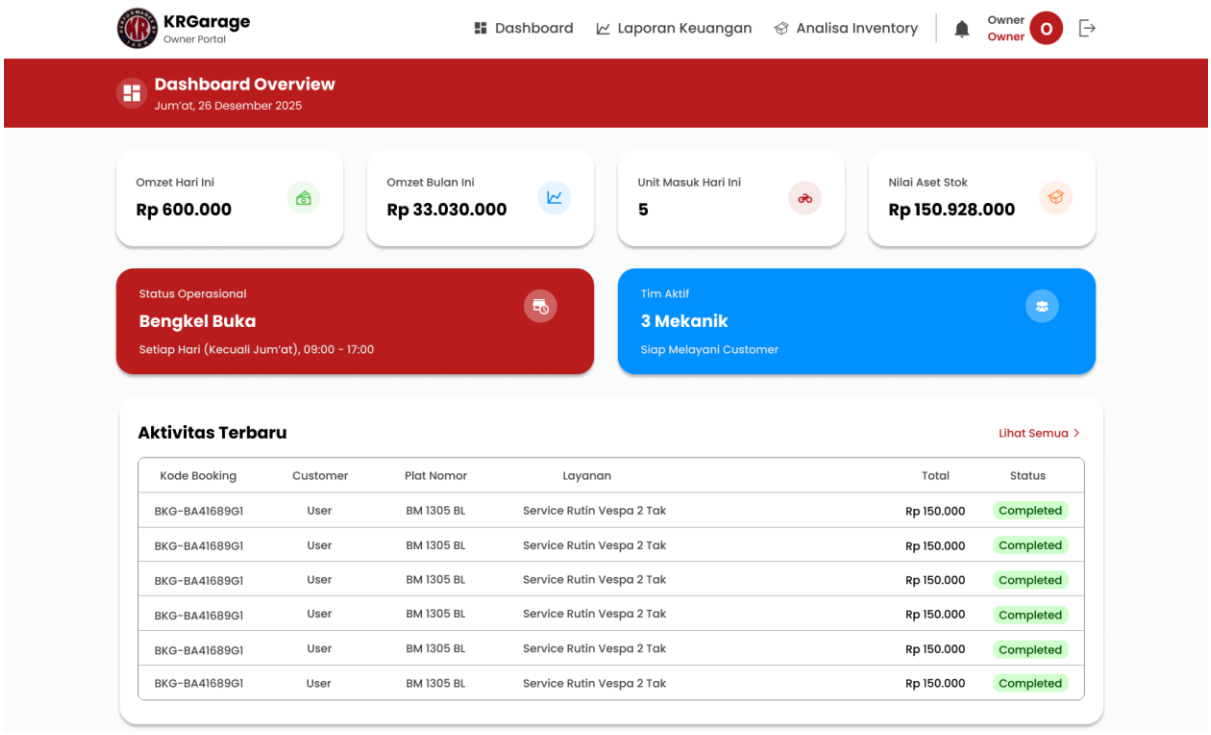


Gambar 4.8 Halaman Daftar Pekerjaan Mekanik

Daftar pekerjaan mekanik ditampilkan melalui halaman pada Gambar 4.8. Halaman ini menampilkan seluruh pekerjaan servis yang menjadi tanggung jawab mekanik. Informasi yang ditampilkan meliputi kode pemesanan, data pelanggan, data Vespa, layanan yang dipilih, serta jadwal servis. Selain itu, tersedia fitur penyaringan berdasarkan status pengerjaan untuk memudahkan mekanik dalam memantau pekerjaan yang sedang berlangsung. Pada pekerjaan yang berstatus In Progress, mekanik dapat menambahkan suku cadang yang digunakan selama proses servis serta mengubah status pekerjaan menjadi selesai setelah pengerjaan servis selesai dilakukan.

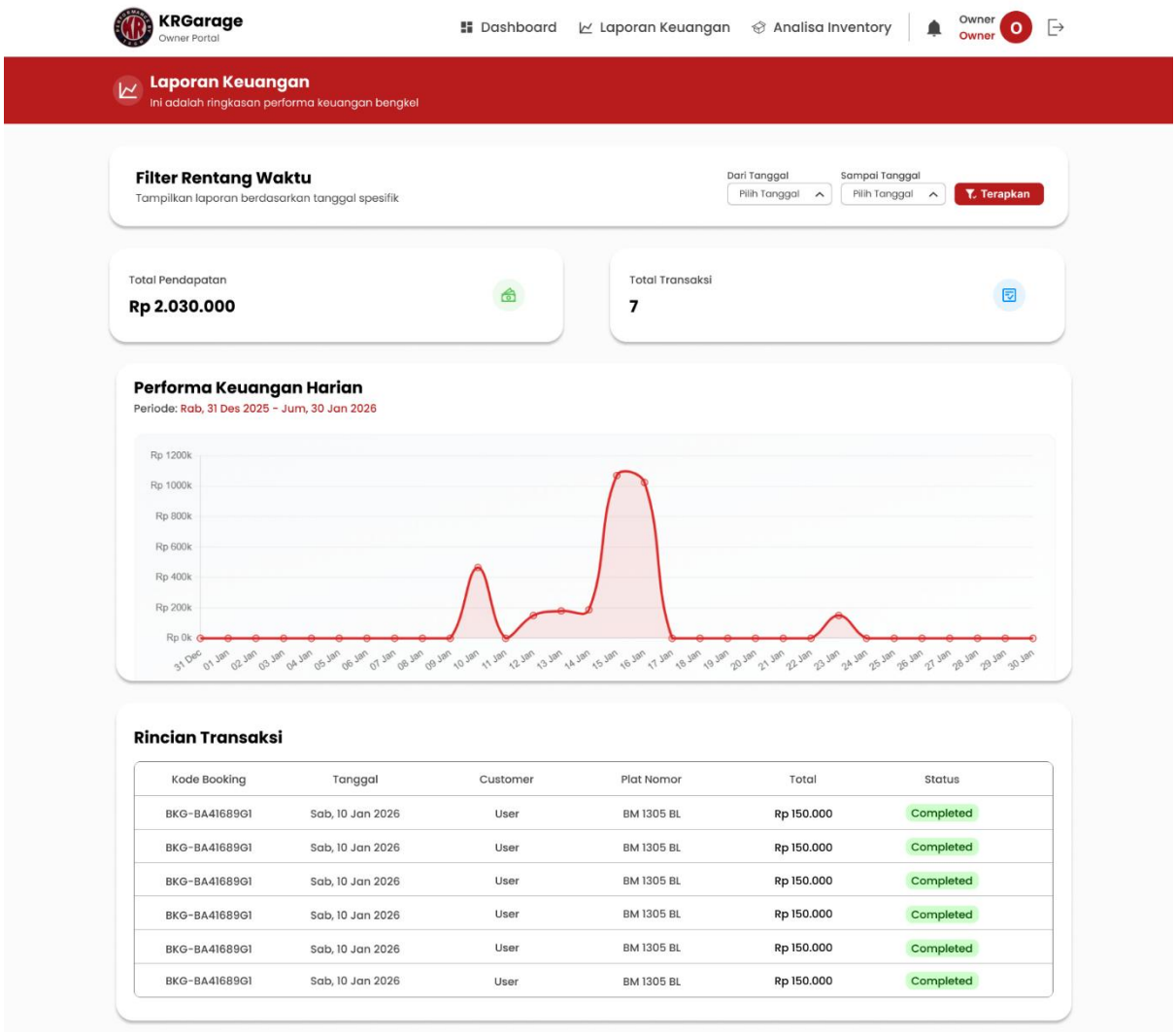
4.2.4 Prototipe Pemilik

Prototipe pemilik dirancang untuk membantu pemilik bengkel memantau kondisi operasional dan kinerja bisnis secara keseluruhan. Antarmuka difokuskan pada penyajian informasi yang ringkas dan mudah dipahami, sehingga pemilik dapat memperoleh gambaran mengenai aktivitas bengkel, pendapatan, serta performa layanan secara cepat.



Gambar 4.9 Halaman *Dashboard* Pemilik

Halaman *dashboard* pemilik pada Gambar 4.9 menampilkan ringkasan informasi operasional bengkel dalam bentuk indikator dan statistik. Informasi yang disajikan meliputi omzet harian, omzet bulanan, jumlah unit servis yang masuk, nilai aset stok, status operasional bengkel, jumlah mekanik aktif, serta daftar aktivitas servis terbaru. Halaman ini dirancang untuk membantu pemilik memantau kondisi bengkel secara *real-time* tanpa harus melihat data transaksi secara langsung.



Gambar 4.10 Halaman Laporan Keuangan Pemilik

Halaman laporan keuangan pada Gambar 4.10 digunakan untuk menampilkan informasi pendapatan bengkel berdasarkan rentang waktu tertentu. Pemilik dapat melakukan penyaringan data menggunakan filter tanggal yang tersedia. Selain menampilkan total pendapatan dan jumlah transaksi, sistem juga menyajikan grafik performa keuangan harian serta rincian transaksi yang telah selesai. Informasi tersebut membantu pemilik dalam melakukan pemantauan dan evaluasi terhadap kinerja keuangan bengkel.

4.3 Evaluasi Prototipe dan Iterasi Desain

Sebagai bagian dari penerapan metode *Prototyping*, prototipe yang telah dirancang kemudian dievaluasi bersama calon pengguna sistem yang terdiri dari pemilik, admin, mekanik, dan pelanggan bengkel KRGarage. Tahap evaluasi ini bertujuan untuk memperoleh

masukan mengenai kesesuaian fitur, kemudahan penggunaan antarmuka, serta dukungan sistem terhadap kebutuhan operasional bengkel.

Masukan dan temuan yang diperoleh selama proses evaluasi dicatat secara terstruktur, kemudian dijadikan dasar dalam melakukan penyempurnaan desain. Untuk memastikan seluruh masukan tersebut telah diakomodasi, dilakukan pemetaan antara hasil evaluasi prototipe dengan bukti pemenuhannya pada tahap implementasi sistem seperti yang disajikan pada Tabel 4.3 berikut.

Tabel 4.3 Matriks Evaluasi Prototipe dan Bukti Pemenuhan Implementasi

Aspek / Aktor	**Temuan dan Masukan**	**Bukti Pemenuhan Implementasi**
Penyesuaian Terminologi (Pemilik)	Beberapa data contoh pada prototipe masih menggunakan istilah umum yang kurang sesuai dengan kebiasaan operasional bengkel Vespa klasik	Telah diimplementasikan pada basis data layanan dan suku cadang dengan menerapkan istilah spesifik lingkungan KRGarage, seperti “Tali-tali”, “Stang Seher”, dan “Platina”
Optimasi Antarmuka (Admin)	Proses pelayanan pelanggan sering kali menghadapi kondisi antrean yang ramai. Admin membutuhkan fitur pembaruan data yang cepat tanpa harus membuka banyak jendela tambahan	Telah diterapkan fitur pembaruan status langsung pada halaman *dashboard* admin berbasis desktop/laptop, sehingga admin dapat memperbarui status servis secara langsung melalui komponen interaktif sekali klik
Aksesibilitas Perangkat Seluler (Mekanik)	Mekanik memiliki mobilitas tinggi di area bengkel dan memiliki tingkat literasi teknologi yang beragam, sehingga memerlukan perangkat yang praktis dan antarmuka yang sederhana	Telah dibangun antarmuka khusus mekanik dengan desain responsif pada perangkat seluler yang mengutamakan tata letak minimalis serta tombol aksi berukuran besar yang mudah dijangkau saat bekerja
Validasi Fitur & Identitas (Pelanggan)	1. Fitur pemesanan servis *online* divalidasi mampu mengurai benturan jadwal 2. Pelanggan meminta dukungan pengelolaan multi-kendaraan 3. Muncul usulan penambahan atribut warna Vespa, namun disepakati nomor polisi sudah cukup mewakili identitas Vespa	1. Telah diimplementasikan relasi basis data *one-to-many* yang memungkinkan pelanggan mendaftarkan lebih dari satu Vespa dalam satu akun 2. Atribut warna Vespa resmi tidak ditambahkan pada *form* registrasi Vespa untuk menjaga efisiensi *input* data pelanggan

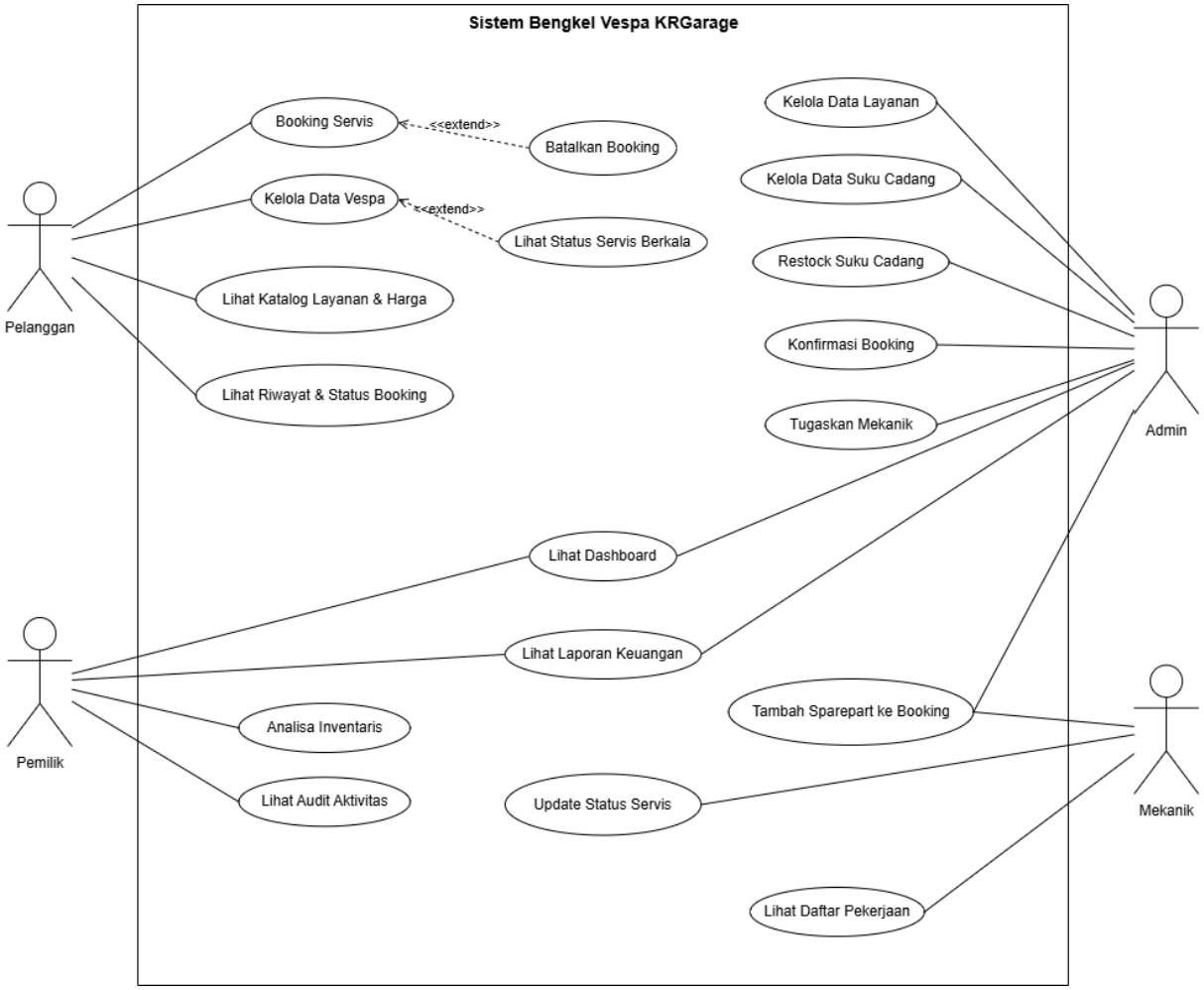
Berdasarkan matriks evaluasi dan bukti pemenuhan pada Tabel 4.3 di atas, dapat disimpulkan bahwa seluruh rangkaian iterasi rancangan prototipe telah selaras dan diwujudkan secara penuh pada tahap implementasi sistem. Masukan yang diperoleh dari pemilik, admin, mekanik, dan pelanggan terbukti menjadi landasan kuat dalam menghasilkan sistem informasi manajemen layanan bengkel yang tidak hanya berjalan sesuai spesifikasi teknis, melainkan juga adaptif terhadap proses bisnis nyata di bengkel KRGarage.

4.4 Perancangan Sistem

4.4.1 *Use Case Diagram*

Perancangan *Use Case Diagram* bertujuan untuk memvisualisasikan fungsionalitas sistem dari sudut pandang pengguna. Diagram ini menggambarkan interaksi antara aktor dengan sistem serta menunjukkan hak akses yang dimiliki oleh masing-masing aktor sesuai perannya dalam operasional bengkel.

Dalam pengembangan Sistem Informasi Manajemen Layanan Bengkel KRGarage, identifikasi aktor menjadi langkah krusial untuk memisahkan peran antara manajemen operasional, teknis, dan pelanggan. Berdasarkan analisis kebutuhan yang telah dilakukan, sistem ini dirancang untuk memfasilitasi empat aktor utama, yaitu admin, mekanik, pelanggan, dan pemilik. Gambar 4.11 berikut menyajikan rancangan *Use Case Diagram* secara keseluruhan.



Gambar 4.11 *Use Case Diagram* Sistem KRGarage

Gambar 4.11 di atas menunjukkan bahwa sistem memiliki empat aktor utama, yaitu pelanggan, admin, mekanik, dan pemilik. Setiap aktor memiliki peran serta hak akses yang berbeda sesuai dengan kebutuhan operasional bengkel KRGarage.

Aktor pelanggan berinteraksi dengan sistem untuk melakukan pemesanan servis secara *online*. *Use case* ini memiliki relasi *extend* dengan batalkan pemesanan yang memungkinkan pelanggan membatalkan pemesanan selama masih memenuhi ketentuan yang berlaku. Selain itu, pelanggan dapat mengelola data Vespa melalui fitur kelola data Vespa yang memiliki relasi *extend* dengan lihat status servis berkala untuk menampilkan informasi kebutuhan servis berdasarkan riwayat perawatan Vespa. Pelanggan juga dapat melihat katalog layanan dan harga sebagai referensi sebelum menentukan pemesanan servis, serta memantau riwayat dan status pemesanan untuk mengetahui perkembangan proses servis yang sedang berlangsung maupun yang telah selesai.

Aktor admin bertanggung jawab dalam pengelolaan operasional dan administrasi bengkel. Admin memiliki hak akses untuk mengelola data layanan servis melalui fitur kelola

data layanan, mengelola data suku cadang melalui fitur kelola data suku cadang, serta melakukan *restock* suku cadang untuk menjaga ketersediaan stok. Selain itu, admin juga berperan dalam proses pengelolaan pemesanan dengan melakukan konfirmasi pemesanan dan menugaskan mekanik yang akan menangani pekerjaan servis sesuai kebutuhan dan ketersediaan sumber daya bengkel.

Aktor mekanik berperan sebagai pelaksana proses servis Vespa pelanggan. Mekanik dapat melihat daftar pekerjaan yang telah ditugaskan oleh admin melalui fitur lihat daftar pekerjaan. Selama proses pengerjaan berlangsung, mekanik dapat memperbarui progres servis melalui fitur *update* status servis. Selain itu, mekanik juga dapat menambahkan suku cadang yang digunakan selama proses perbaikan melalui fitur tambah suku cadang ke pemesanan sehingga rincian pekerjaan dan biaya servis dapat tercatat secara lengkap pada sistem.

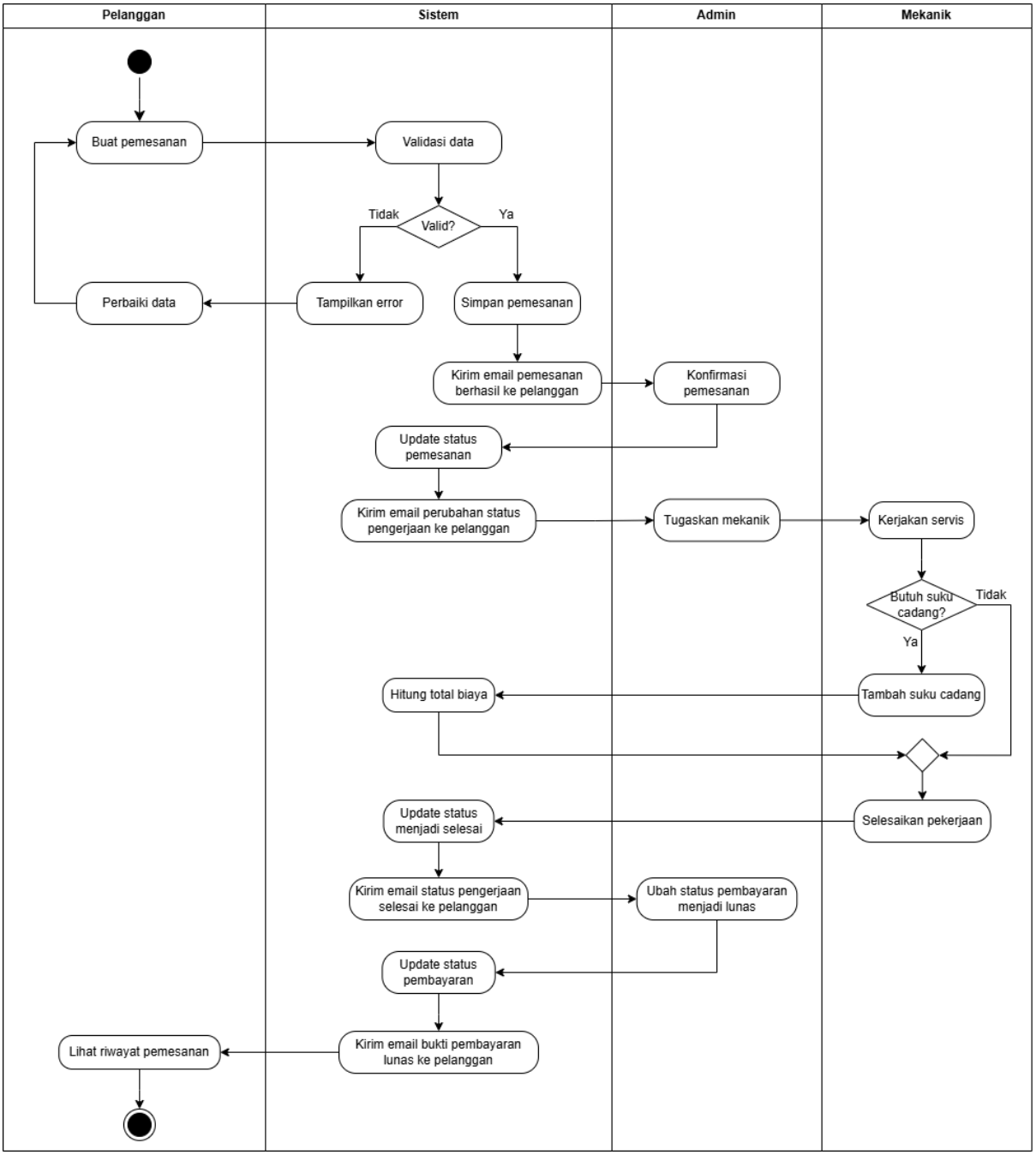
Aktor pemilik berperan sebagai pihak yang melakukan pengawasan terhadap operasional dan kinerja bisnis bengkel. Pemilik dapat memantau kondisi bengkel melalui fitur lihat *dashboard* yang menyajikan ringkasan aktivitas operasional secara *real-time*. Pemilik juga dapat mengakses fitur lihat laporan keuangan untuk memantau pendapatan dan aktivitas transaksi bengkel. Untuk mendukung pengambilan keputusan, sistem juga menyediakan fitur analisis inventaris yang digunakan untuk memantau kondisi stok suku cadang dan informasi inventaris bengkel. Selain itu, pemilik juga dapat melihat audit aktivitas untuk memantau riwayat aktivitas pengguna pada sistem.

4.4.2 *Activity Diagram*

Activity Diagram digunakan untuk memodelkan alur kerja dari sebuah sistem dan urutan aktivitas pada suatu proses. Diagram ini memberikan gambaran detail mengenai langkah-langkah prosedural dari *Use Case* utama yang ada pada Sistem Informasi Manajemen Layanan Bengkel KRGarage. Berikut merupakan uraian *Activity Diagram* untuk proses bisnis inti pada sistem.

a. *Activity Diagram* Pemesanan Servis

Proses pemesanan servis merupakan alur utama dalam interaksi antara pelanggan dan bengkel. Alur ini melibatkan pelanggan, sistem, admin, dan mekanik, mulai dari pembuatan pemesanan, validasi data, penugasan mekanik, pengerjaan servis, hingga pembaruan status pemesanan menjadi selesai.



Gambar 4.12 *Activity Diagram* Pemesanan Servis

Alur pemesanan servis pada Gambar 4.12 diawali ketika pelanggan mengisi data pemesanan melalui sistem. Selanjutnya sistem melakukan validasi terhadap data yang dimasukkan. Apabila data tidak valid, sistem akan menampilkan pesan kesalahan dan pelanggan diminta melakukan perbaikan data. Sebaliknya, apabila data valid maka sistem akan menyimpan data pemesanan dan mengirimkan notifikasi pemesanan berhasil kepada pelanggan.

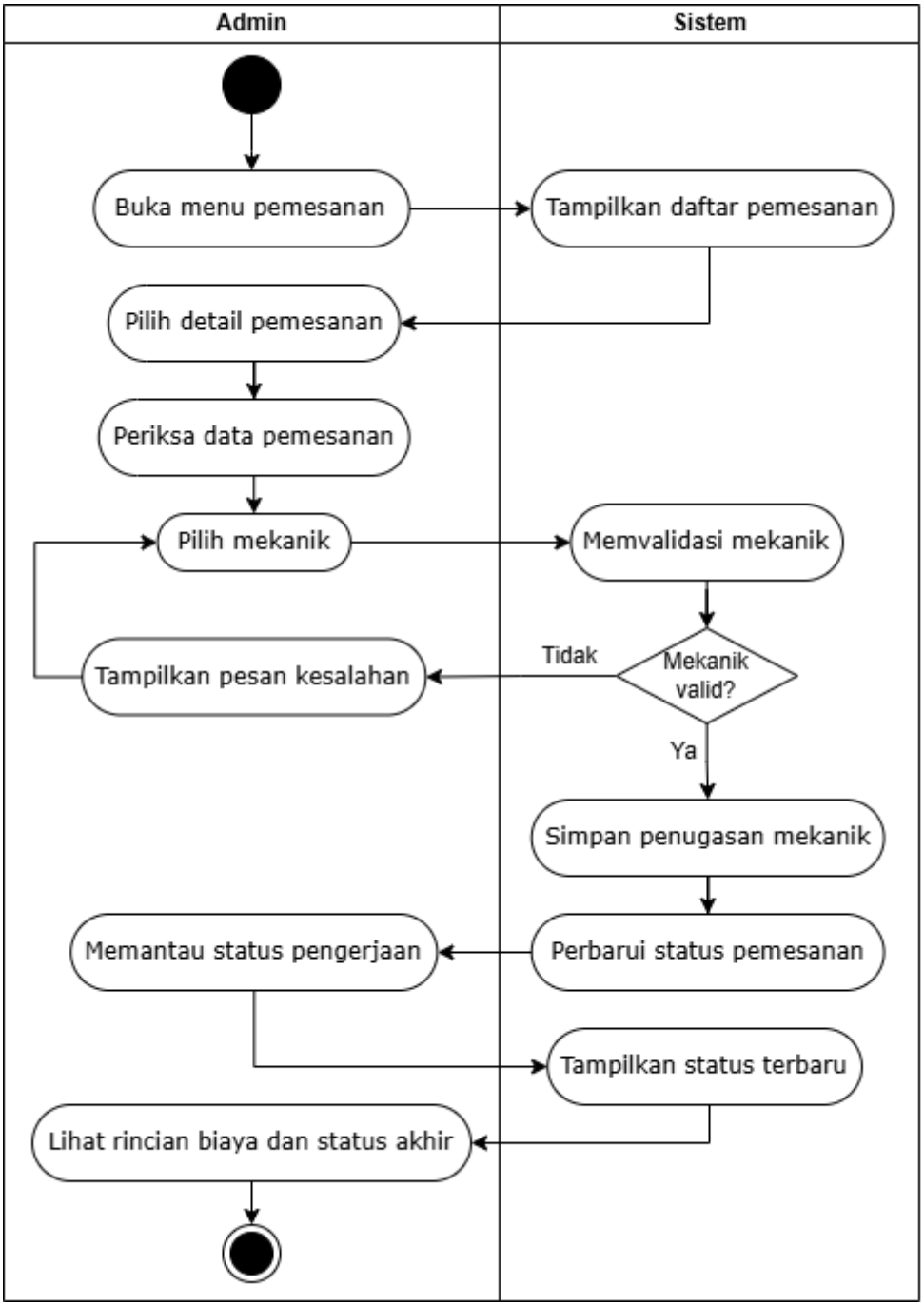
Setelah pemesanan tersimpan, admin melakukan konfirmasi terhadap data pemesanan servis yang masuk. Sistem kemudian memperbarui status pemesanan dan mengirimkan informasi perubahan status kepada pelanggan. Selanjutnya admin menugaskan mekanik untuk menangani pekerjaan servis sesuai dengan jadwal yang telah ditentukan.

Mekanik kemudian melaksanakan proses servis sesuai dengan layanan yang dipilih pelanggan. Selama proses pengerjaan berlangsung, mekanik dapat menambahkan suku cadang apabila ditemukan kebutuhan penggantian komponen. Apabila diperlukan konfirmasi kepada pelanggan terkait penambahan suku cadang, proses konfirmasi dilakukan secara manual oleh admin melalui kontak WhatsApp yang tersedia pada data pemesanan. Jika terdapat penambahan suku cadang, sistem akan melakukan perhitungan ulang total biaya servis berdasarkan layanan dan suku cadang yang digunakan.

Setelah pekerjaan selesai dilakukan, mekanik menyelesaikan pekerjaan dan sistem memperbarui status servis menjadi selesai. Sistem kemudian mengirimkan informasi penyelesaian servis kepada pelanggan. Setelah pelanggan melakukan pembayaran, admin mengubah status pembayaran menjadi lunas. Sistem kemudian memperbarui status pembayaran, mencatat waktu pembayaran, serta mengirimkan bukti pembayaran digital kepada pelanggan melalui email. Seluruh informasi pemesanan yang telah selesai selanjutnya dapat diakses pelanggan melalui fitur riwayat pemesanan.

b. *Activity Diagram* Kelola Pemesanan

Diagram ini dibuat untuk menjelaskan aktivitas admin dalam mengelola pemesanan yang telah dibuat oleh pelanggan. Diagram ini merupakan turunan dari proses pemesanan servis, khususnya pada tahap pemeriksaan data pemesanan, penugasan mekanik, serta pemantauan status pengerjaan servis.



Gambar 4.13 *Activity Diagram* Kelola Pemesanan

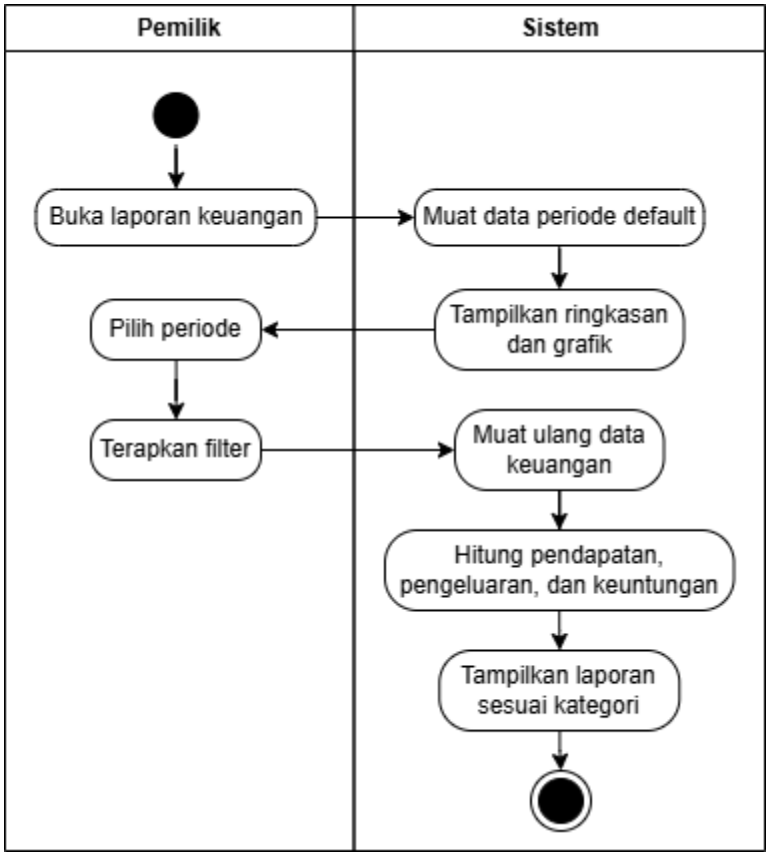
Activity Diagram kelola pemesanan pada Gambar 4.13 terdiri dari dua *swimlane*, yaitu admin dan sistem. Proses dimulai ketika admin membuka menu pemesanan, kemudian sistem menampilkan daftar pemesanan yang tersedia. Admin kemudian memilih salah satu detail pemesanan untuk diperiksa. Pada tahap ini, admin dapat melihat dan memeriksa informasi pemesanan, seperti data pelanggan, data Vespa, layanan yang dipilih, jadwal servis, serta catatan tambahan yang diberikan pelanggan.

Selanjutnya admin memilih mekanik yang akan ditugaskan untuk menangani pemesanan tersebut. Sistem kemudian melakukan validasi terhadap mekanik yang dipilih. Apabila mekanik yang dipilih tidak valid, sistem akan menampilkan pesan kesalahan dan admin diminta untuk memilih mekanik kembali. Sebaliknya, apabila mekanik valid, sistem akan menyimpan data penugasan mekanik dan memperbarui status pemesanan. Admin kemudian dapat memantau perkembangan pengerjaan servis yang sedang berlangsung. Sistem kemudian menampilkan status terbaru sehingga admin dapat melihat perkembangan pekerjaan yang dilakukan oleh mekanik.

Pada tahap akhir, admin dapat melihat rincian biaya serta status akhir pemesanan yang telah diproses. Dengan adanya alur ini, proses pengelolaan pemesanan dapat dilakukan secara terstruktur mulai dari pemeriksaan data pemesanan, penugasan mekanik, pemantauan status pengerjaan, hingga pengecekan hasil akhir pemesanan.

c. *Activity Diagram* Laporan Keuangan

Activity diagram laporan keuangan menggambarkan alur aktivitas pemilik dalam mengakses dan melihat informasi keuangan pada sistem KRGarage. Diagram ini menunjukkan proses mulai dari pemilik membuka halaman laporan keuangan, sistem menampilkan data awal, hingga pemilik memilih periode tertentu untuk memperoleh laporan keuangan yang lebih spesifik. Melalui diagram ini, dapat terlihat bagaimana sistem membantu pemilik dalam memantau kondisi keuangan bengkel secara terstruktur.



Gambar 4.14 *Activity Diagram* Laporan Keuangan

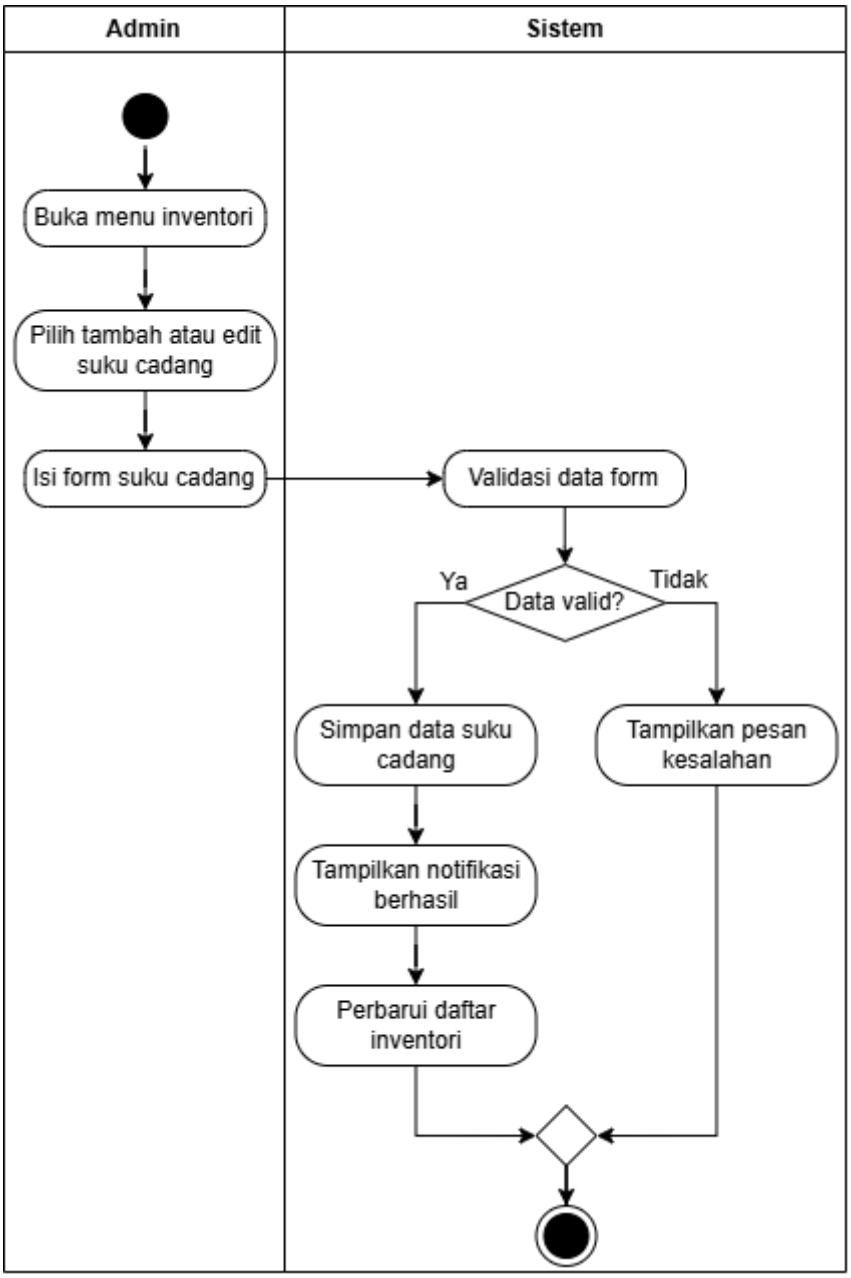
Proses laporan keuangan pada Gambar 4.14 dimulai ketika pemilik membuka halaman laporan keuangan. Setelah halaman dibuka, sistem secara otomatis memuat data berdasarkan periode *default* dan menampilkan ringkasan keuangan beserta grafik. Informasi awal ini memudahkan pemilik untuk melihat gambaran umum kondisi keuangan bengkel tanpa harus melakukan pengaturan periode terlebih dahulu.

Selanjutnya, pemilik dapat memilih periode laporan sesuai kebutuhan, kemudian menerapkan filter yang telah dipilih. Setelah filter diterapkan, sistem akan memuat ulang data keuangan berdasarkan periode tersebut. Sistem kemudian menghitung pendapatan, pengeluaran, dan keuntungan berdasarkan data transaksi pada periode yang dipilih, lalu menampilkan laporan sesuai kategori yang tersedia. Dengan adanya alur ini, pemilik dapat memperoleh informasi keuangan yang lebih akurat dan sesuai dengan rentang waktu yang dibutuhkan.

d. *Activity Diagram* Kelola Data Suku Cadang

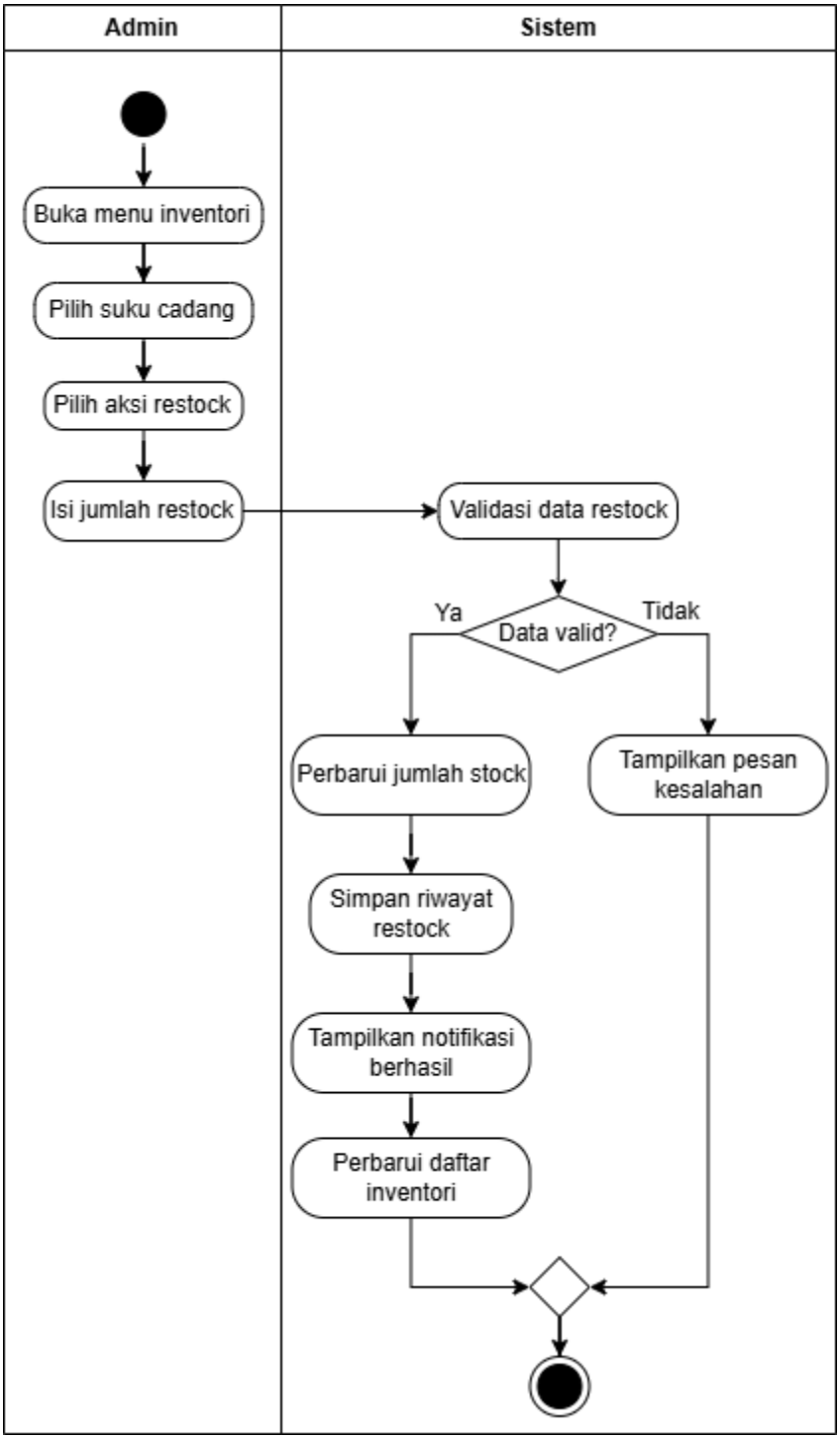
Activity diagram kelola suku cadang digunakan untuk menggambarkan alur aktivitas admin dalam mengelola data inventori suku cadang pada sistem KRGarage. Modul ini memiliki beberapa jenis proses yang berbeda, yaitu tambah atau edit suku cadang, *restock* suku

cadang, dan hapus suku cadang. Oleh karena itu, *activity diagram* pada modul ini dibagi menjadi tiga bagian agar setiap alur dapat dijelaskan secara lebih jelas dan terstruktur.



Gambar 4.15 *Activity Diagram* Tambah atau Edit Suku Cadang

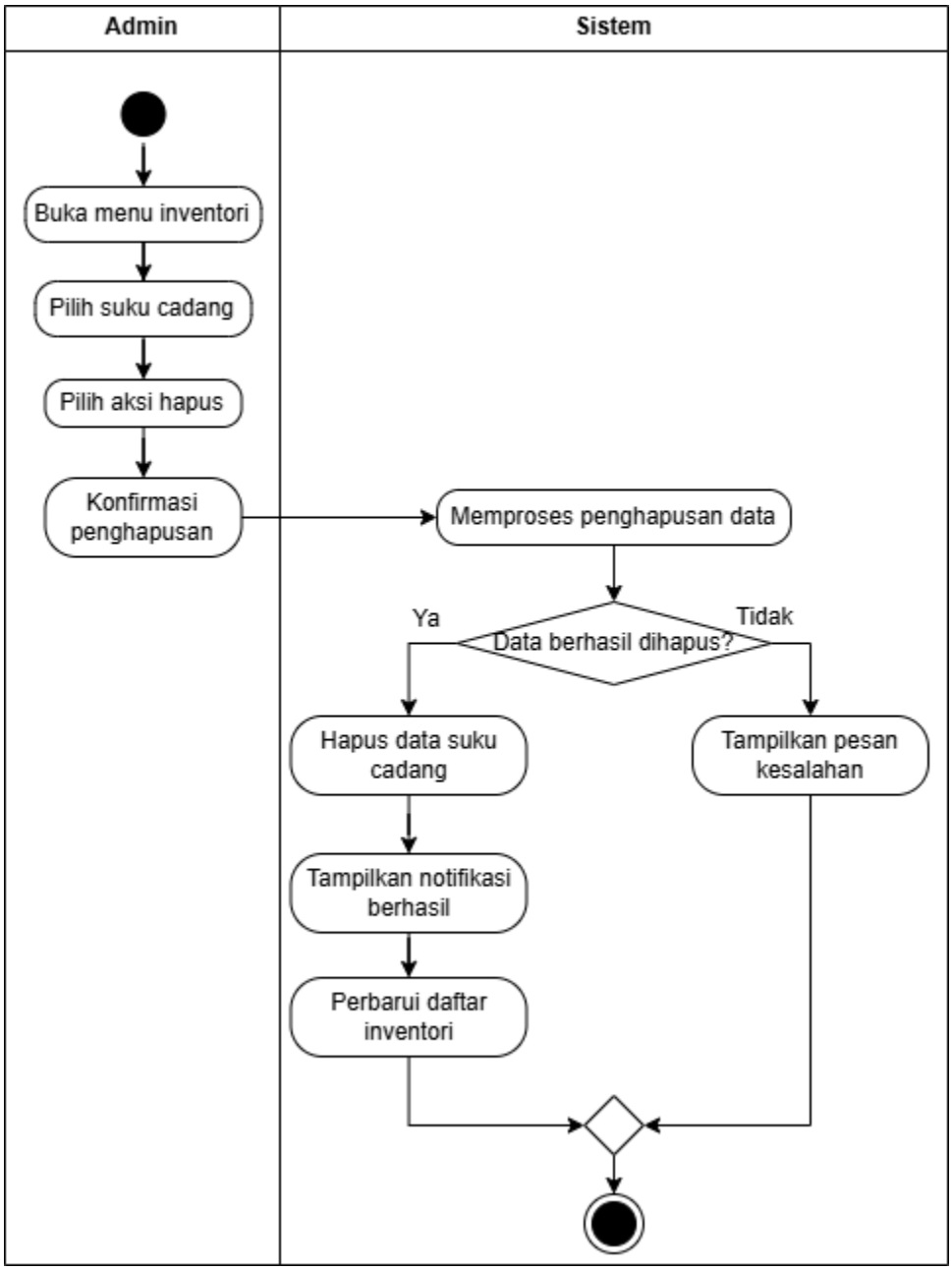
Pada *Activity Diagram* Tambah atau Edit Suku Cadang, proses dimulai ketika admin membuka menu inventori. Setelah itu, admin memilih aksi tambah atau edit suku cadang, kemudian mengisi *form* data suku cadang sesuai informasi yang diperlukan oleh sistem. Setelah data suku cadang diisi, sistem melakukan validasi data yang dimasukkan. Jika data valid, sistem menyimpan data suku cadang, menampilkan notifikasi berhasil, dan memperbarui daftar inventori. Namun, jika data tidak valid, sistem akan menampilkan pesan kesalahan agar admin dapat memperbaiki data yang dimasukkan.



Gambar 4.16 *Activity Diagram Restock* Suku Cadang

Pada *Activity Diagram Restock* Suku Cadang, proses dimulai ketika admin membuka menu inventori dan memilih suku cadang yang akan ditambahkan stoknya. Admin kemudian memilih aksi *restock* dan mengisi jumlah *restock* sesuai kebutuhan. Setelah jumlah *restock* diinputkan, sistem melakukan validasi data *restock* untuk memastikan jumlah yang dimasukkan sesuai ketentuan. Apabila data valid, sistem memperbarui jumlah stok, menyimpan riwayat *restock*, menampilkan notifikasi berhasil, dan memperbarui daftar

inventori. Jika data tidak valid, sistem akan menampilkan pesan kesalahan sehingga admin dapat memperbaiki jumlah *restock* yang dimasukkan.



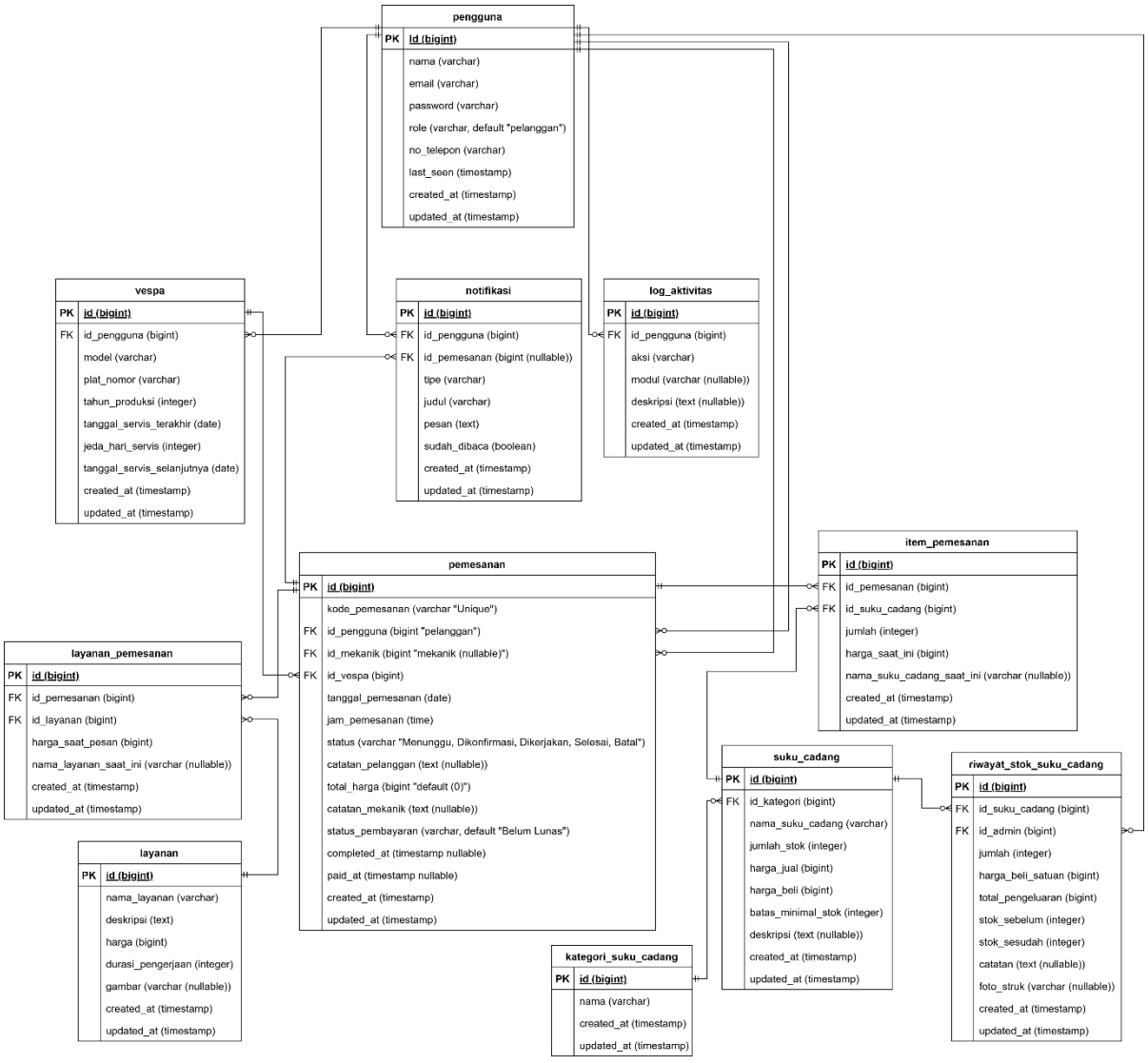
Gambar 4.17 *Activity Diagram* Hapus Suku Cadang

Pada *Activity Diagram* Hapus Suku Cadang, proses dimulai ketika admin membuka menu inventori, memilih suku cadang, lalu memilih aksi hapus. Sebelum data dihapus, admin harus melakukan konfirmasi penghapusan terlebih dahulu. Setelah konfirmasi diberikan, sistem memproses penghapusan data. Jika data berhasil dihapus, sistem menghapus data suku cadang, menampilkan notifikasi berhasil, dan memperbarui daftar inventori. Namun, jika proses penghapusan gagal, sistem akan menampilkan pesan kesalahan. Dengan adanya tiga

alur tersebut, proses pengelolaan suku cadang menjadi lebih terstruktur karena setiap aktivitas admin memiliki tahapan validasi dan *respons* sistem yang jelas.

4.4.3 *Entity Relationship Diagram*

Berikut merupakan *Entity Relationship Diagram* (ERD) pada Sistem Informasi Manajemen Layanan Bengkel KRGarage. ERD ini digunakan untuk menggambarkan struktur basis data serta hubungan antar entitas yang digunakan dalam sistem. Melalui rancangan ini, dapat terlihat bagaimana data pengguna, Vespa, pemesanan, layanan, suku cadang, notifikasi, riwayat stok, dan log aktivitas saling terhubung untuk mendukung proses operasional bengkel.



Gambar 4.18 *Entity Relationship Diagram* Sistem KRGarage

Pada ERD tersebut, tabel pengguna menjadi salah satu entitas utama karena menyimpan data seluruh pengguna sistem, baik pelanggan, admin, mekanik, maupun pemilik. Tabel ini berelasi dengan tabel Vespa, karena satu pengguna dengan peran pelanggan dapat memiliki

satu atau lebih data Vespa. Selain itu, tabel pengguna juga berelasi dengan tabel pemesanan, baik sebagai pelanggan yang membuat pemesanan melalui kolom id_pengguna maupun sebagai mekanik yang ditugaskan untuk menangani servis melalui kolom id_mekanik. Kolom id_mekanik bersifat *nullable* karena mekanik belum tentu langsung ditetapkan ketika pemesanan pertama kali dibuat. Tabel pengguna juga memiliki relasi dengan tabel notifikasi, riwayat_stok_suku_cadang, dan log_aktivitas.

Tabel Vespa digunakan untuk menyimpan data Vespa milik pelanggan, seperti model, plat nomor, tahun produksi, tanggal servis terakhir, jeda hari servis, dan tanggal servis selanjutnya. Tabel ini berelasi dengan tabel pemesanan karena satu Vespa dapat memiliki banyak riwayat pemesanan servis. Relasi ini memungkinkan sistem menampilkan riwayat servis serta menentukan informasi pengingat servis berkala berdasarkan data servis terakhir dan jadwal servis berikutnya.

Tabel pemesanan digunakan untuk menyimpan data pemesanan servis yang dilakukan oleh pelanggan. Tabel ini menyimpan informasi kode pemesanan, pelanggan, mekanik, Vespa, tanggal pemesanan, jam pemesanan, status pengerjaan, catatan pelanggan, catatan mekanik, total harga, status pembayaran, waktu penyelesaian, dan waktu pembayaran. Tabel pemesanan berelasi dengan tabel layanan_pemesanan untuk mencatat layanan yang dipilih pada suatu pemesanan. Karena satu pemesanan dapat memiliki lebih dari satu layanan, maka relasi antara pemesanan dan layanan dihubungkan melalui tabel perantara layanan_pemesanan. Tabel ini juga menyimpan harga layanan dan nama layanan saat pemesanan dilakukan agar data transaksi tetap konsisten meskipun data layanan berubah di kemudian hari.

Selain layanan, tabel pemesanan juga berelasi dengan tabel item_pemesanan untuk mencatat penggunaan suku cadang selama proses servis. Tabel item_pemesanan menghubungkan tabel pemesanan dengan tabel suku_cadang, sehingga sistem dapat mencatat suku cadang yang digunakan, jumlah pemakaian, harga saat digunakan, serta nama suku cadang saat transaksi berlangsung. Dengan adanya tabel ini, rincian biaya servis dapat dihitung berdasarkan gabungan biaya layanan dan penggunaan suku cadang.

Tabel suku_cadang digunakan untuk menyimpan data inventori suku cadang bengkel, seperti nama suku cadang, jumlah stok, harga jual, harga beli, batas minimal stok, dan deskripsi. Tabel ini berelasi dengan tabel kategori_suku_cadang, karena satu kategori dapat memiliki banyak data suku cadang. Relasi ini digunakan untuk mengelompokkan suku cadang berdasarkan kategori tertentu agar pengelolaan inventori menjadi lebih terstruktur.

Untuk mendukung pencatatan perubahan stok, sistem menyediakan tabel riwayat_stok_suku_cadang. Tabel ini digunakan untuk mencatat aktivitas *restock* suku cadang, meliputi suku cadang yang ditambahkan, admin yang melakukan pencatatan, jumlah *restock*, harga beli satuan, total pengeluaran, stok sebelum *restock*, stok setelah *restock*, catatan, dan foto struk. Tabel ini berelasi dengan tabel suku_cadang sebagai barang yang ditambahkan stoknya dan tabel pengguna sebagai admin yang mencatat aktivitas *restock*. Dengan adanya tabel ini, sistem dapat mendukung pencatatan pengeluaran *restock* serta analisis inventaris.

Tabel notifikasi digunakan untuk menyimpan pemberitahuan kepada pengguna terkait aktivitas pemesanan, seperti perubahan status pemesanan atau informasi servis. Tabel ini berelasi dengan tabel pengguna sebagai penerima notifikasi dan dapat berelasi dengan tabel pemesanan apabila notifikasi tersebut berkaitan dengan suatu pemesanan tertentu. Kolom id_pemesanan pada tabel notifikasi bersifat *nullable* karena tidak semua notifikasi harus berkaitan langsung dengan pemesanan.

Selain itu, sistem juga memiliki tabel log_aktivitas yang digunakan untuk mencatat aktivitas pengguna di dalam sistem. Tabel ini berelasi dengan tabel pengguna melalui kolom id_pengguna. Data yang disimpan meliputi aksi, modul, dan deskripsi aktivitas. Keberadaan tabel ini bertujuan untuk membantu proses pelacakan aktivitas pengguna serta mendukung kebutuhan *monitoring* sistem.

Secara keseluruhan, rancangan ERD ini mendukung kebutuhan utama sistem KRGarage, mulai dari pengelolaan pengguna, data Vespa, pemesanan servis, penugasan mekanik, pencatatan layanan, penggunaan suku cadang, pengelolaan stok, notifikasi, log aktivitas, hingga penyajian laporan operasional dan keuangan.

4.5 Implementasi Sistem

Tahap implementasi sistem merupakan proses penerjemahan dari perancangan sistem dan prototipe yang telah disetujui ke dalam bentuk bahasa pemrograman yang sesungguhnya. Sistem Informasi Manajemen Layanan Bengkel KRGarage ini dibangun menggunakan *framework* Vue.js pada sisi *frontend* dan Laravel pada sisi *backend*. Implementasi ini menghasilkan sistem berbasis *web* yang responsif dan dapat diakses melalui berbagai perangkat. Berikut adalah penjabaran dari hasil implementasi antarmuka pada sistem yang telah dibangun.

4.5.1 Implementasi RESTful API

Dalam menerapkan arsitektur *Client-Server*, komunikasi data antara antarmuka pengguna dan logika *server* dijembatani oleh *RESTful Application Programming Interface* (API). *Framework* Laravel bertugas memproses logika bisnis, keamanan, serta manipulasi basis data, dan kemudian mengirimkan *respons* dalam format JSON ke antarmuka klien.

Untuk memastikan struktur data yang rapi dan terorganisir, seluruh *endpoint* API yang diimplementasikan pada sistem KRGarage telah dikelompokkan berdasarkan hak akses pengguna dan kategori fungsionalitasnya. Berikut adalah rincian daftar *endpoint* API yang digunakan di dalam sistem:

Tabel 4.4 Daftar API Autentikasi dan Akses Publik

No	*Method*	*Endpoint* URL	Fungsi
1	POST	/api/daftar	Mendaftarkan akun pelanggan baru ke dalam sistem
2	POST	/api/masuk	Autentikasi *login* pengguna dan mengembalikan token Sanctum
3	POST	/api/auth/google-login	Autentikasi *login*/registrasi otomatis menggunakan Google OAuth2
4	GET	/api/layanan	Mengambil daftar seluruh layanan servis Vespa untuk halaman utama
5	GET	/api/storage/{path}	Menyajikan *file* media publik langsung dari penyimpanan lokal server

Berdasarkan Tabel 4.4, *endpoint* pada kelompok autentikasi dan akses publik ini dirancang untuk dapat diakses secara terbuka tanpa memerlukan token autentikasi. *Endpoint* ini menangani proses awal pengguna seperti pendaftaran akun baru, proses *login* konvensional maupun menggunakan integrasi Google OAuth2, serta pengambilan data publik seperti daftar layanan bengkel di halaman utama dan akses ke *file* media.

Tabel 4.5 Daftar API Pengguna Terautentikasi (Umum)

No	*Method*	*Endpoint* URL	Fungsi
1	POST	/api/keluar	Membatalkan dan menghapus token autentikasi Sanctum saat *logout*
2	GET	/api/profil	Mengambil data profil dan peran dari pengguna yang sedang aktif
3	PUT	/api/profil	Memperbarui informasi profil pengguna
4	PUT	/api/profil/password	Mengubah kata sandi pengguna aktif
5	POST	/api/ping	Memperbarui waktu status keaktifan (*last_seen*) untuk deteksi *online*
6	GET	/api/notifikasi	Mengambil daftar notifikasi sistem milik pengguna
7	POST	/api/notifikasi/{id}/tandai-dibaca	Menandai satu notifikasi spesifik sebagai sudah dibaca
8	POST	/api/notifikasi/tandai-semua-dibaca	Menandai seluruh notifikasi pengguna sebagai sudah dibaca

Tabel 4.5 menunjukkan daftar *endpoint* umum yang hanya dapat diakses oleh pengguna yang telah berhasil *login* ke dalam sistem dan memiliki token Sanctum yang valid.

Fungsionalitas pada kelompok ini bersifat universal untuk seluruh hak akses, yang meliputi proses *logout* untuk menghapus sesi, pengelolaan profil pengguna, pembaruan kata sandi, deteksi keaktifan pengguna, serta pengelolaan notifikasi sistem seperti menandai pesan telah dibaca.

Tabel 4.6 Daftar API Fitur Pelanggan

No	*Method*	*Endpoint* URL	Fungsi
1	GET	/api/vespa-saya	Mengambil daftar seluruh unit Vespa milik pelanggan
2	POST	/api/vespa-saya	Menambahkan unit Vespa baru ke garasi pelanggan
3	PUT	/api/vespa-saya/{id}	Mengedit informasi unit Vespa milik pelanggan
4	DELETE	/api/vespa-saya/{id}	Menghapus unit Vespa dari garasi pelanggan
5	GET	/api/vespa-saya/perlu-servis	Mengambil data pengingat Vespa yang memasuki jadwal servis berkala
6	GET	/api/pemesanan	Mengambil riwayat seluruh pemesanan servis milik pelanggan
7	GET	/api/pemesanan/cek-slot	Memeriksa ketersediaan kuota jam servis pada tanggal tertentu
8	GET	/api/pemesanan/{id}	Mengambil detail lengkap satu transaksi pemesanan servis
9	POST	/api/pemesanan	Membuat pengajuan pemesanan servis Vespa baru
10	POST	/api/pemesanan/{id}/batalkan	Membatalkan pemesanan servis oleh pelanggan

Penjelasan untuk Tabel 4.6 menguraikan *endpoint* yang secara khusus diotorisasi hanya untuk pengguna dengan hak akses pelanggan. Melalui *endpoint* ini, sistem memfasilitasi kebutuhan pelanggan dalam mengelola data Vespa mereka, memeriksa ketersediaan jadwal servis, membuat atau membatalkan pengajuan pemesanan servis, serta melihat rekam jejak riwayat perawatan Vespa beserta pengingat jadwal servis berkala.

Tabel 4.7 Daftar API Fitur Admin

No	*Method*	*Endpoint* URL	Fungsi
1	GET	/api/admin/pemesanan	Mengambil seluruh daftar pemesanan servis masuk dengan filter status
2	GET	/api/admin/pemesanan/{id}	Mengambil detail pemesanan untuk administrasi dan pemrosesan kasir
3	PATCH	/api/admin/pemesanan/{id}/status	Memperbarui status alur pemesanan (Konfirmasi, Diproses, Selesai, Batal)
4	PATCH	/api/admin/pemesanan/{id}/status-pembayaran	Memperbarui status pembayaran transaksi (Pending/Lunas)
5	POST	/api/admin/pemesanan/{id}/tambah-suku-cadang	Menambahkan item suku cadang ke nota pemesanan pelanggan
6	DELETE	/api/admin/pemesanan/{id}/item/{idItem}	Menghapus item suku cadang dari nota pemesanan
7	GET	/api/admin/mekanik	Mengambil daftar mekanik yang tersedia untuk penugasan

8	PATCH	/api/admin/pemesanan/{id}/tugaskan-mekanik	Menugaskan mekanik spesifik untuk mengerjakan unit pemesanan
9	POST	/api/admin/layanan	Menambahkan jenis layanan servis baru
10	PUT	/api/admin/layanan/{id}	Memperbarui informasi dan tarif harga layanan servis
11	DELETE	/api/admin/layanan/{id}	Menghapus jenis layanan servis dari sistem
12	GET	/api/admin/inventori/kategori	Mengambil daftar kategori suku cadang
13	POST	/api/admin/inventori/kategori	Menambahkan kategori suku cadang baru
14	GET	/api/admin/inventori	Mengambil seluruh katalog suku cadang beserta jumlah stoknya
15	POST	/api/admin/inventori	Menambahkan data suku cadang baru ke sistem inventori
16	GET	/api/admin/inventori/stok-menipis	Mengambil data peringatan suku cadang yang stoknya menipis
17	GET	/api/admin/inventori/{id}	Mengambil detail informasi satu suku cadang
18	PUT	/api/admin/inventori/{id}	Memperbarui informasi data suku cadang (harga beli, harga jual, nama)
19	DELETE	/api/admin/inventori/{id}	Menghapus data suku cadang dari sistem
20	POST	/api/admin/inventori/{id}/tambah-stok	Melakukan *restock* suku cadang dan mencatat transaksi pengeluaran
21	GET	/api/admin/dashboard/statistik	Mengambil indikator statistik *dashboard* admin
22	GET	/api/admin/dashboard/pemesanan-terbaru	Mengambil daftar pemesanan servis yang baru masuk
23	GET	/api/admin/laporan/keuangan	Menggenerasi laporan arus kas keuangan bengkel
24	GET	/api/admin/karyawan	Mengambil daftar seluruh staf (Admin dan Mekanik)
25	POST	/api/admin/karyawan	Menambahkan data akun staf/karyawan baru
26	PUT	/api/admin/karyawan/{id}	Memperbarui data informasi staf/karyawan
27	DELETE	/api/admin/karyawan/{id}	Menghapus akun staf/karyawan dari sistem

Berdasarkan Tabel 4.7, *endpoint* fitur admin merupakan kelompok API dengan fungsionalitas paling krusial untuk mengendalikan alur operasional harian bengkel. *Endpoint* ini memberikan kewenangan kepada admin untuk mengelola pemesanan yang masuk, mengubah status pengerjaan dan pembayaran, menetapkan penugasan kepada mekanik, serta mengelola data master seperti katalog layanan, inventaris suku cadang, dan akun karyawan.

Selain itu, API ini juga bertugas memproses aktivitas kasir dan pencatatan restok barang yang akan terintegrasi langsung dengan laporan keuangan.

Tabel 4.8 Daftar API Fitur Mekanik

No	*Method*	*Endpoint* URL	Fungsi
1	GET	/api/mekanik/pemesanan	Mengambil daftar tugas servis yang ditugaskan kepada mekanik terkait
2	PUT	/api/mekanik/pemesanan/{id}/status	Mengubah status pengerjaan servis secara bertahap
3	GET	/api/mekanik/suku-cadang	Mengambil daftar suku cadang yang dapat dipilih saat pengerjaan
4	POST	/api/mekanik/pemesanan/{id}/tambah-suku-cadang	Menambahkan penggunaan suku cadang langsung dari halaman pengerjaan
5	DELETE	/api/mekanik/pemesanan/{id}/item/{idItem}	Membatalkan/menghapus penggunaan suku cadang dari pengerjaan

Tabel 4.8 menjelaskan daftar *endpoint* yang didedikasikan secara khusus untuk hak akses mekanik. *Endpoint* ini dirancang sederhana agar mekanik dapat fokus pada tugas teknis di lapangan, yaitu mengambil daftar antrean Vespa yang telah ditugaskan kepadanya, memperbarui status progres pengerjaan servis secara bertahap, serta menambahkan atau membatalkan penggunaan suku cadang langsung ke dalam nota pemesanan pelanggan.

Tabel 4.9 Daftar API Fitur Pemilik Bengkel

No	*Method*	*Endpoint* URL	Fungsi
1	GET	/api/pemilik/statistik	Mengambil statistik eksekutif (pendapatan harian, pendapatan bulanan, unit servis, nilai aset stok)
2	GET	/api/pemilik/pemesanan-terbaru	Mengambil daftar transaksi pemesanan servis paling baru
3	GET	/api/pemilik/perbandingan-performa	Mengambil data perbandingan harian (Hari Ini vs Kemarin) untuk grafik *dashboard*
4	GET	/api/pemilik/tren-pendapatan	Mengambil data grafik tren pendapatan berdasarkan rentang tanggal
5	GET	/api/pemilik/transaksi	Mengambil riwayat seluruh transaksi pendapatan masuk
6	GET	/api/pemilik/pengeluaran-restok	Mengambil rincian pengeluaran modal untuk *restock* inventori suku cadang
7	GET	/api/pemilik/layanan-terpopuler	Mengambil peringkat layanan servis paling sering dipesan pelanggan
8	GET	/api/pemilik/suku-cadang-terlaris	Mengambil peringkat suku cadang paling terlaris/banyak digunakan
9	GET	/api/pemilik/stok-menipis	Mengambil data peringatan barang suku cadang yang perlu segera dibeli/*restock*
10	GET	/api/pemilik/log-aktivitas	Mengambil *audit trail* (catatan jejak aktivitas seluruh pengguna sistem)
11	GET	/api/pemilik/mekanik-online	Mengambil data jumlah mekanik yang sedang aktif/*online*

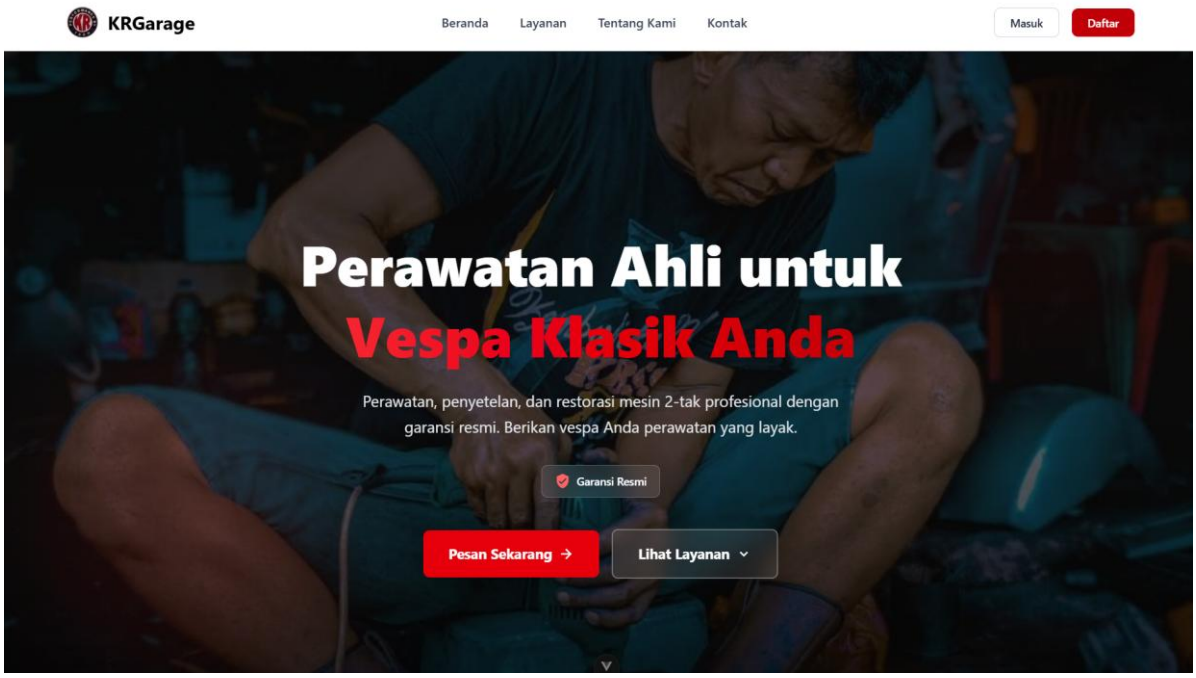
12	GET	/api/pemilik/ringkasan	Mengambil ringkasan data operasional bengkel secara menyeluruh
13	GET	/api/pemilik/metrik-keuangan	Mengambil indikator metrik laba rugi dan margin bisnis

Merujuk pada Tabel 4.9, *endpoint* fitur pemilik bengkel difokuskan pada penyajian data analitik dan pelaporan tanpa memiliki akses untuk mengubah data transaksi. *Endpoint* ini bertugas menarik data dari *database* untuk disajikan pada *dashboard* pemilik, yang meliputi metrik statistik keuangan, tren performa harian, peringkat layanan dan suku cadang terlaris, peringatan stok menipis, hingga catatan jejak aktivitas dari seluruh pengguna sistem.

4.5.2 Implementasi Antarmuka Pelanggan

Antarmuka pelanggan diimplementasikan untuk memenuhi kebutuhan pelanggan yang telah dirumuskan pada Tabel 4.2, yaitu kemudahan melakukan pemesanan servis, pengelolaan data Vespa, akses riwayat servis, pemantauan status pengerjaan, serta pengingat servis berkala. Oleh karena itu, antarmuka pelanggan dirancang responsif dan sederhana agar dapat digunakan dengan mudah melalui perangkat pelanggan.

a. Halaman Beranda



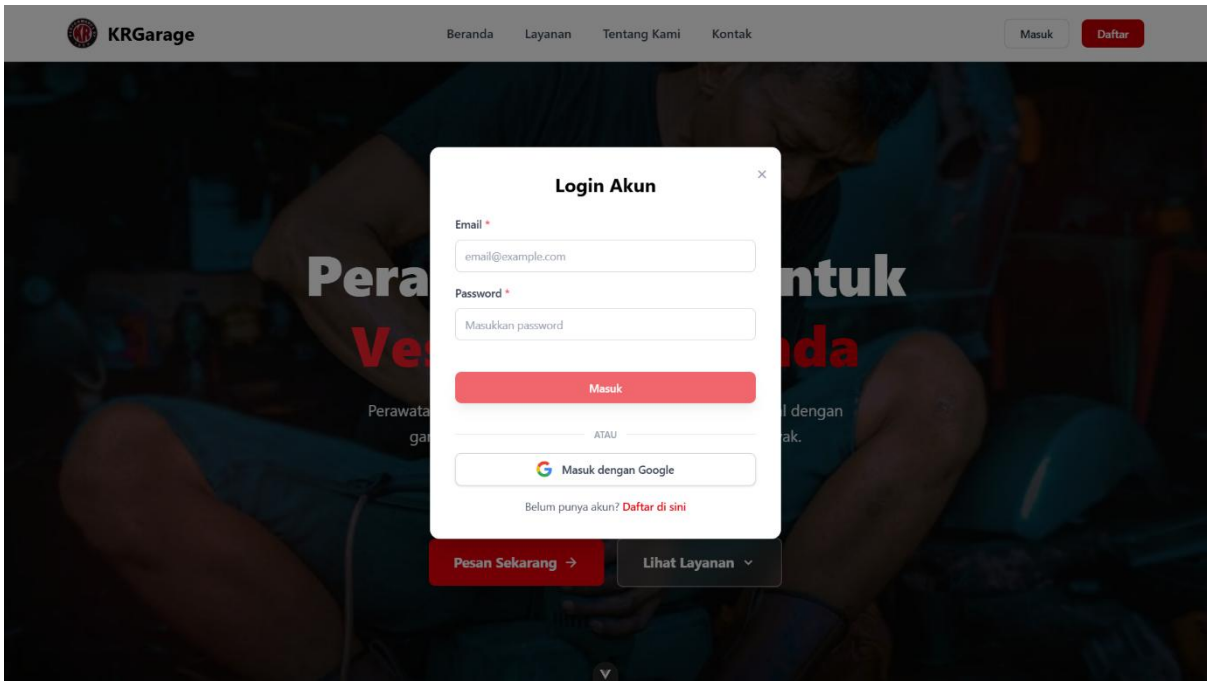
Gambar 4.19 Antarmuka Halaman Beranda

Halaman beranda pada Gambar 4.19 berfungsi sebagai halaman awal yang memperkenalkan layanan KRGarage kepada pengguna. Pada bagian atas halaman tersedia navigasi menuju menu Beranda, Layanan, Tentang Kami, dan Kontak, serta tombol Masuk dan

Daftar. Struktur navigasi ini membantu pengguna memperoleh informasi awal mengenai layanan bengkel sebelum masuk ke sistem.

Bagian utama halaman menampilkan informasi layanan dengan judul “Perawatan Ahli Untuk Vespa Klasik Anda”. Informasi ini menjelaskan bahwa KRGarage menyediakan layanan perawatan, perbaikan, dan restorasi mesin Vespa 2-tak. Halaman beranda juga dilengkapi tombol Pesan Sekarang yang mengarahkan pengguna untuk masuk ke sistem sebelum membuat pemesanan, serta tombol Lihat Layanan untuk melihat daftar layanan yang tersedia. Dengan demikian, halaman beranda berperan sebagai media pengenalan layanan sekaligus pintu awal bagi pelanggan untuk mengakses fitur utama sistem KRGarage.

b. Modal Masuk atau Daftar



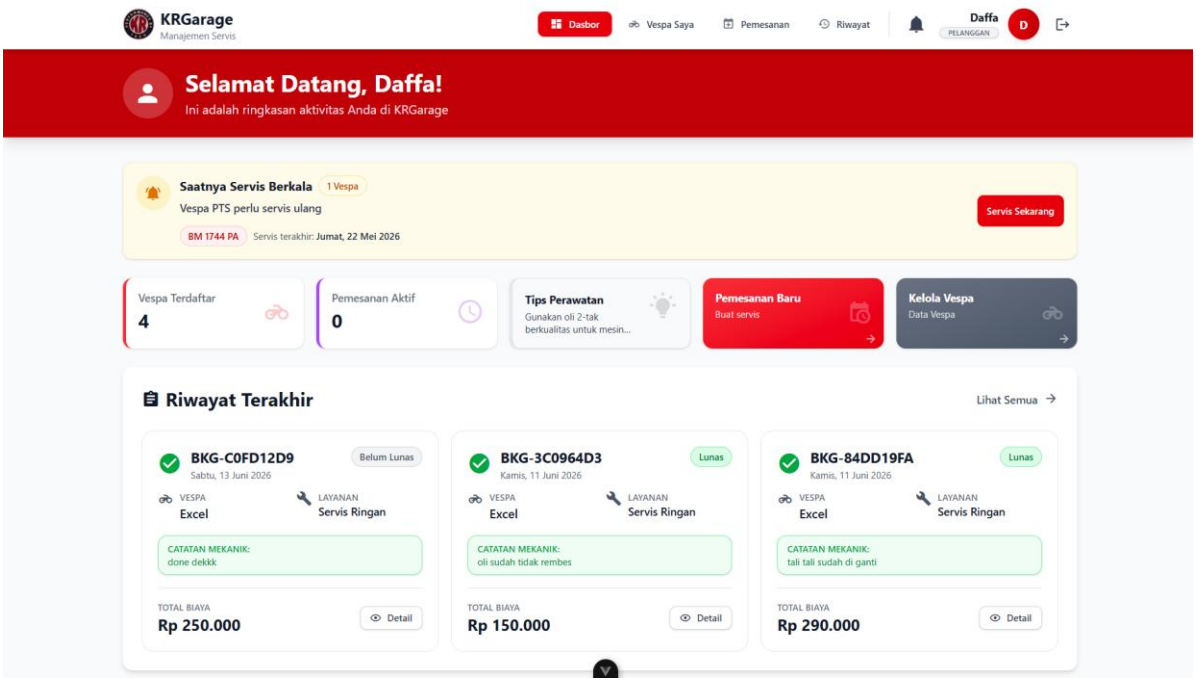
Gambar 4.20 Antarmuka Modal Masuk atau Daftar dengan Google

Antarmuka pada Gambar 4.20 menampilkan modal autentikasi masuk akun (*login*) bagi pengguna sistem KRGarage. Pada antarmuka ini, pengguna diberikan dua pilihan mekanisme masuk ke dalam sistem, yaitu melalui pengisian formulir kredensial email dan kata sandi secara manual, atau menggunakan fitur integrasi *Single Sign-On* (SSO) berbasis Google OAuth2 melalui tombol "Masuk dengan Google".

Ketika pengguna menekan tombol "Masuk dengan Google", sistem secara otomatis akan menampilkan jendela autentikasi resmi dari layanan Google OAuth2. Setelah pengguna memberikan otorisasi pada akun Google miliknya, *server backend* berbasis Laravel akan memvalidasi *identity token* tersebut secara aman. Apabila verifikasi berhasil, sistem akan

mengautentikasi sesi pengguna dan langsung mengarahkannya ke dalam antarmuka utama tanpa harus mengetikkan kata sandi secara manual, sehingga meningkatkan kemudahan dan keamanan akses bagi pengguna.

c. Halaman *Dashboard* Pelanggan



Gambar 4.21 Antarmuka Halaman *Dashboard* Pelanggan

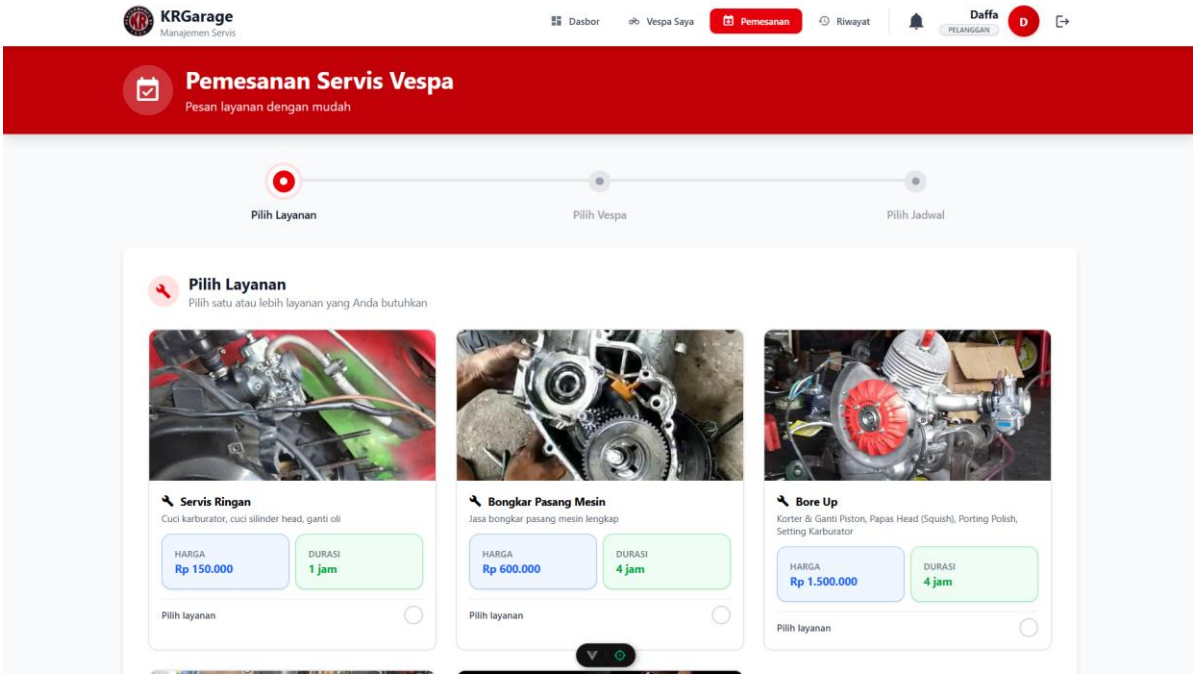
Untuk memudahkan pelanggan memantau aktivitas servisnya, sistem menyediakan halaman *dashboard* pelanggan seperti yang ditampilkan pada Gambar 4.21. Halaman ini mendukung kebutuhan pelanggan pada Tabel 4.2 terkait pemantauan status pengerjaan, akses riwayat servis, dan pengingat servis berkala. Informasi yang ditampilkan meliputi jumlah Vespa terdaftar, jumlah pemesanan aktif, serta akses cepat menuju fitur pemesanan servis dan pengelolaan data Vespa.

Selain menampilkan ringkasan aktivitas, halaman ini juga menyajikan pengingat servis berkala apabila terdapat Vespa yang telah memasuki atau melewati jadwal servis berikutnya. Pada bagian bawah halaman ditampilkan riwayat servis terbaru yang memuat informasi kode pemesanan, status servis, status pembayaran, catatan mekanik, dan total biaya servis. Penyajian informasi tersebut membantu pelanggan memantau aktivitas servis secara lebih cepat tanpa harus membuka setiap menu secara terpisah.

d. Halaman Pemesanan Servis

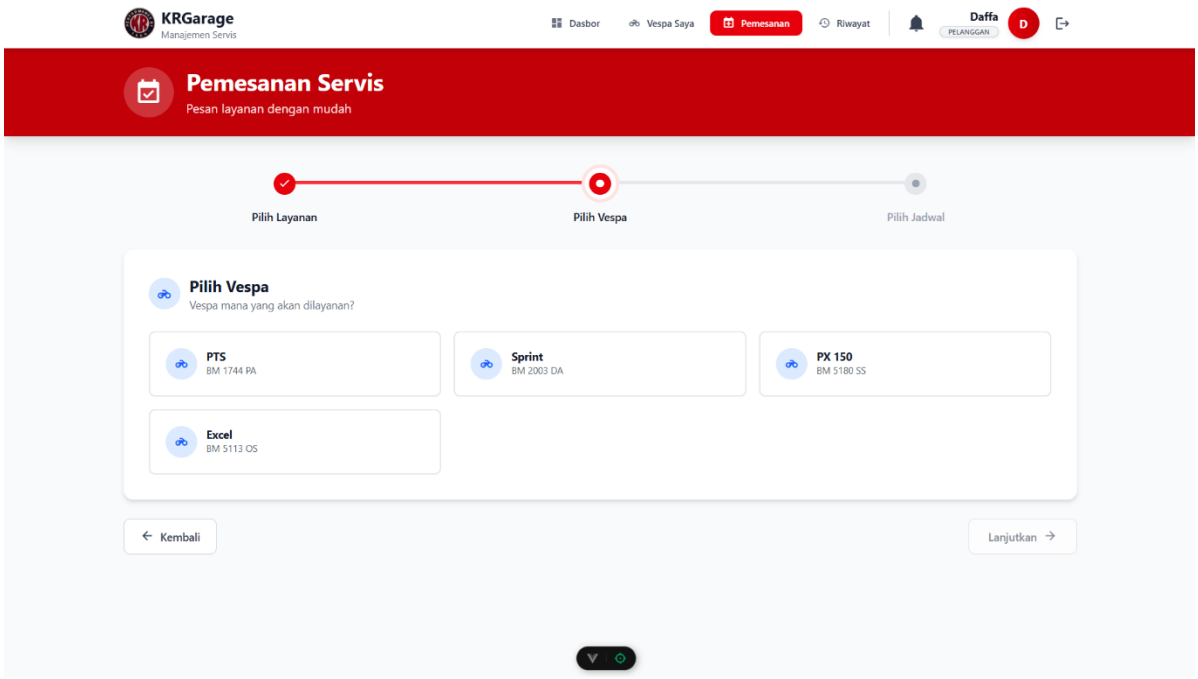
Kebutuhan pelanggan dalam melakukan pemesanan servis secara mandiri diwujudkan melalui halaman pemesanan servis. Fitur ini disusun dalam beberapa tahapan, mulai dari

memilih layanan, memilih Vespa, hingga memilih jadwal servis. Penyusunan tahapan tersebut bertujuan agar pelanggan dapat mengisi data pemesanan secara lengkap dan berurutan.



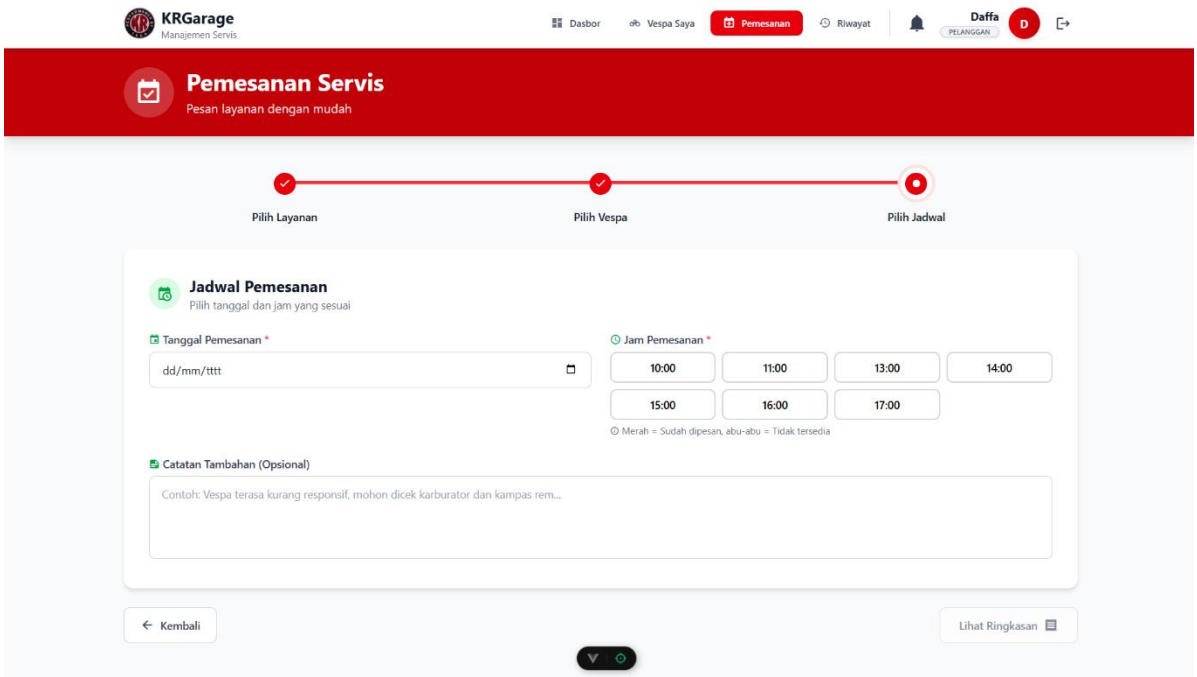
Gambar 4.22 Antarmuka Pilih Layanan

Tahap awal pemesanan pada Gambar 4.22 difokuskan pada pemilihan layanan servis. Sistem menampilkan daftar layanan dalam bentuk kartu yang memuat gambar layanan, deskripsi singkat, harga, dan estimasi durasi pengerjaan. Informasi tersebut mendukung kebutuhan pelanggan pada Tabel 4.2 terkait ketersediaan informasi layanan dan harga sebelum melakukan pemesanan. Setelah layanan dipilih, sistem menyimpan pilihan tersebut dan mengarahkan pelanggan ke tahap pemilihan Vespa.



Gambar 4.23 Antarmuka Pilih Vespa

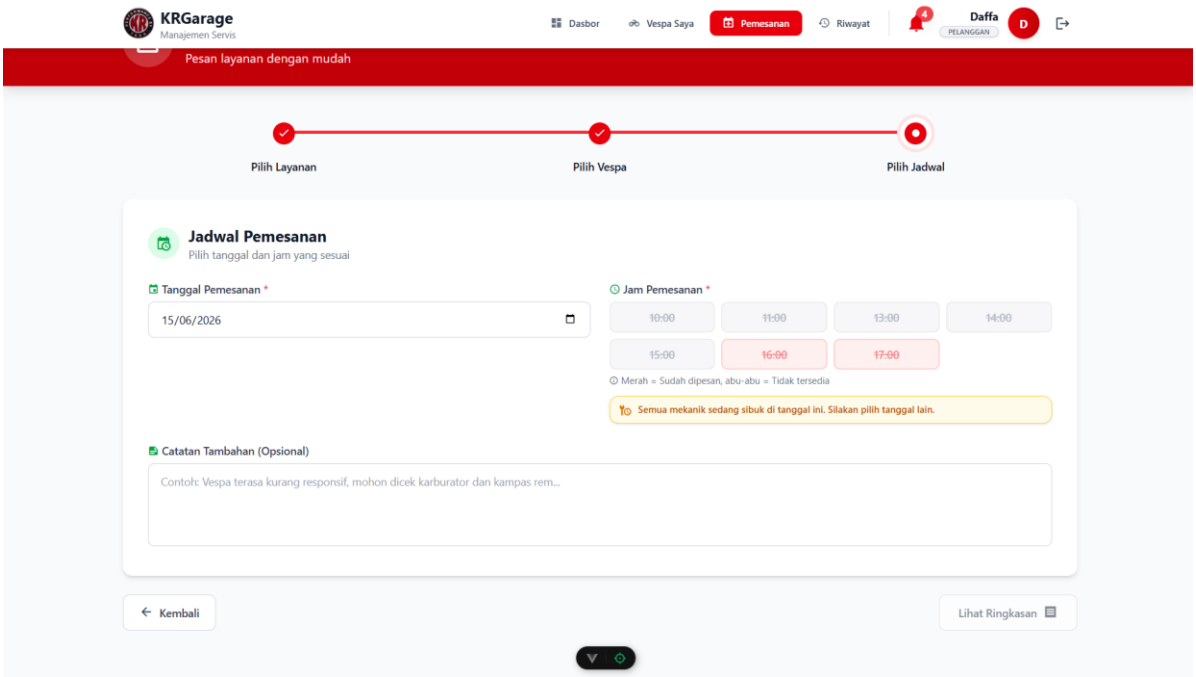
Setelah layanan dipilih, pelanggan diarahkan ke tahap pemilihan Vespa sebagaimana ditampilkan pada Gambar 4.23. Sistem menampilkan seluruh Vespa yang telah terdaftar pada akun pelanggan dalam bentuk kartu sederhana yang memuat nama Vespa dan nomor polisi. Tahapan ini memastikan bahwa setiap pemesanan terhubung dengan data Vespa yang tepat, sehingga riwayat servis dapat tersimpan secara terstruktur berdasarkan kendaraan yang dipilih.



Gambar 4.24 Antarmuka Pilih Jadwal

Tahap pemilihan jadwal pada Gambar 4.24 digunakan untuk mengatur waktu servis sesuai kebutuhan pelanggan dan ketersediaan bengkel. Pada halaman ini, pelanggan dapat memilih tanggal servis, jam kedatangan, serta menambahkan catatan keluhan atau kebutuhan servis sebagai informasi tambahan bagi mekanik. Sistem juga menerapkan validasi jadwal secara otomatis, seperti pembatasan pemilihan jam yang telah berlalu, penolakan pemesanan pada hari libur bengkel, serta penonaktifan slot waktu yang telah penuh. Setelah seluruh data diisi dengan benar, pelanggan dapat melihat ringkasan pemesanan sebelum mengirimkan permintaan servis ke sistem.

e. Validasi Ketersediaan Jadwal Servis

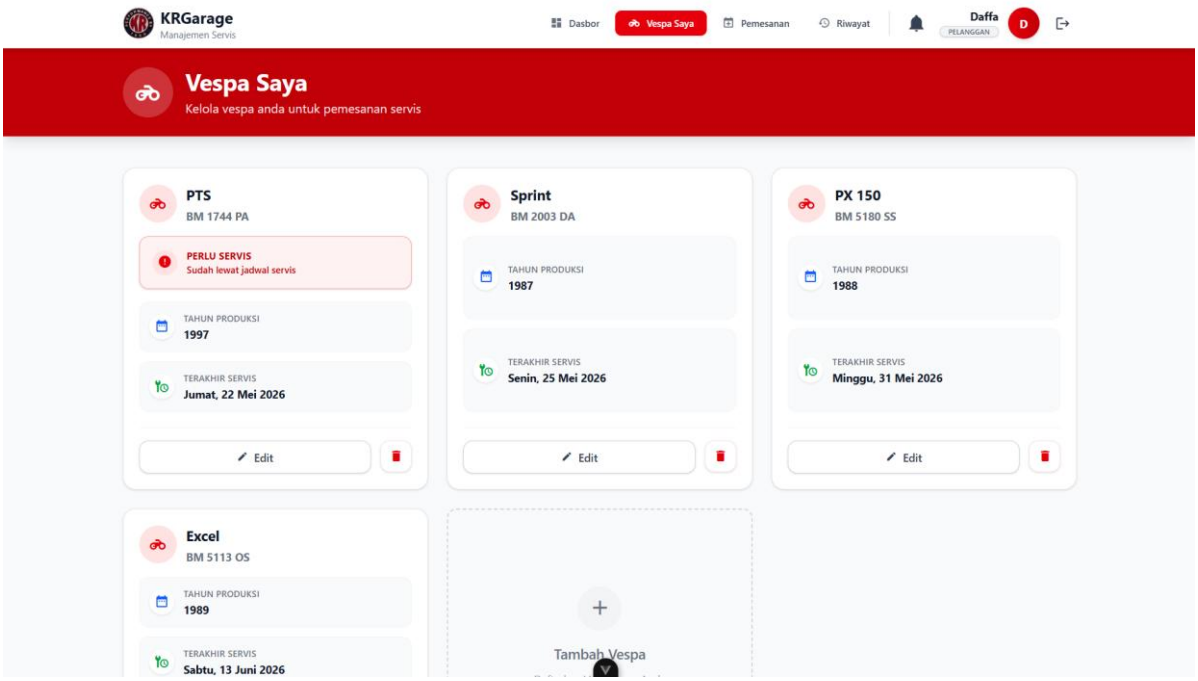


Gambar 4.25 Validasi Ketersediaan Jadwal Servis

Untuk mencegah benturan jadwal, sistem menerapkan validasi ketersediaan slot servis seperti yang ditampilkan pada Gambar 4.25. Validasi ini mempertimbangkan kapasitas mekanik yang tersedia pada waktu tertentu. Apabila seluruh mekanik telah memiliki penugasan, sistem akan menonaktifkan slot pemilihan jam sehingga pelanggan tidak dapat melakukan pemesanan pada jadwal yang sudah penuh.

Mekanisme validasi ini juga memperhatikan hari operasional bengkel dan pembatasan pemilihan jam yang telah berlalu. Dengan adanya validasi tersebut, proses pemesanan dapat berjalan lebih terstruktur, sesuai dengan kapasitas operasional bengkel, dan memberikan kepastian jadwal servis kepada pelanggan.

f. Halaman Vespa Saya

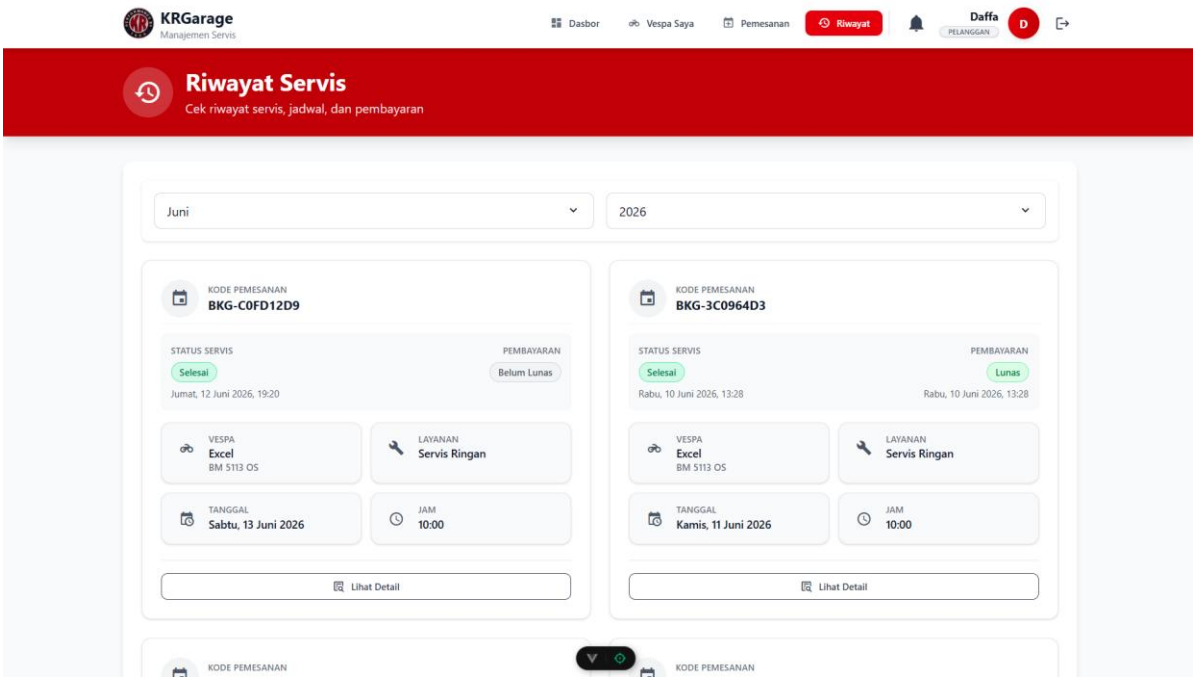


Gambar 4.26 Antarmuka Halaman Vespa Saya

Kebutuhan pelanggan dalam menyimpan dan mengelola data Vespa diwujudkan melalui halaman Vespa Saya. Pada Gambar 4.26, setiap data Vespa ditampilkan dalam bentuk kartu yang memuat informasi identitas Vespa, tahun produksi, plat nomor, serta riwayat servis terakhir. Halaman ini juga menampilkan indikator kebutuhan servis berkala untuk membantu pelanggan mengetahui kondisi perawatan masing-masing Vespa.

Melalui halaman ini, pelanggan dapat menambahkan, mengubah, maupun menghapus data Vespa sesuai kebutuhan. Data yang tersimpan selanjutnya digunakan dalam proses pemesanan servis, sehingga pelanggan tidak perlu memasukkan informasi Vespa secara berulang setiap kali melakukan pemesanan. Implementasi fitur ini mendukung kebutuhan pelanggan pada Tabel 4.2 terkait pengelolaan data Vespa dan riwayat servis yang lebih terstruktur.

g. Halaman Riwayat Servis



Gambar 4.27 Antarmuka Halaman Riwayat Servis Pelanggan

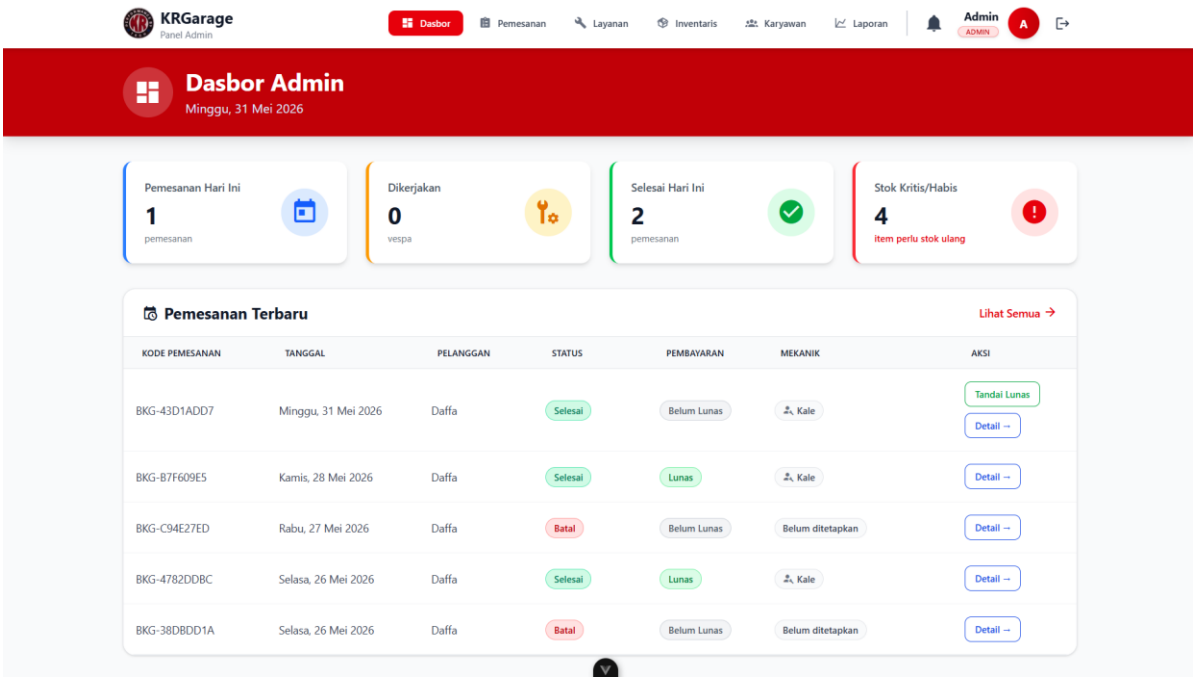
Akses terhadap riwayat servis digital disediakan melalui halaman riwayat servis yang ditampilkan pada Gambar 4.27. Fitur ini mendukung kebutuhan pelanggan pada Tabel 4.2 untuk melihat kembali data pemesanan dan servis yang pernah dilakukan. Sistem juga menyediakan filter bulan dan tahun agar pelanggan dapat mencari riwayat servis berdasarkan periode tertentu.

Setiap riwayat servis menampilkan informasi kode pemesanan, status servis, status pembayaran, data Vespa, layanan yang digunakan, tanggal dan jam servis, serta tombol untuk melihat detail transaksi. Dengan adanya fitur ini, pelanggan dapat mengakses rekam jejak perawatan Vespa secara digital sehingga informasi servis terdahulu dapat diperoleh dengan lebih mudah dibandingkan penyimpanan nota fisik.

4.5.3 Implementasi Antarmuka Admin

Antarmuka admin diimplementasikan untuk memenuhi kebutuhan operasional bengkel yang telah dirumuskan pada Tabel 4.1, khususnya dalam pengelolaan pemesanan servis, layanan, transaksi, suku cadang, dan pencatatan stok. Seluruh fitur admin ditempatkan dalam satu panel agar proses administrasi dapat dilakukan secara terpusat, terdokumentasi, dan lebih efisien.

a. Halaman *Dashboard* Admin

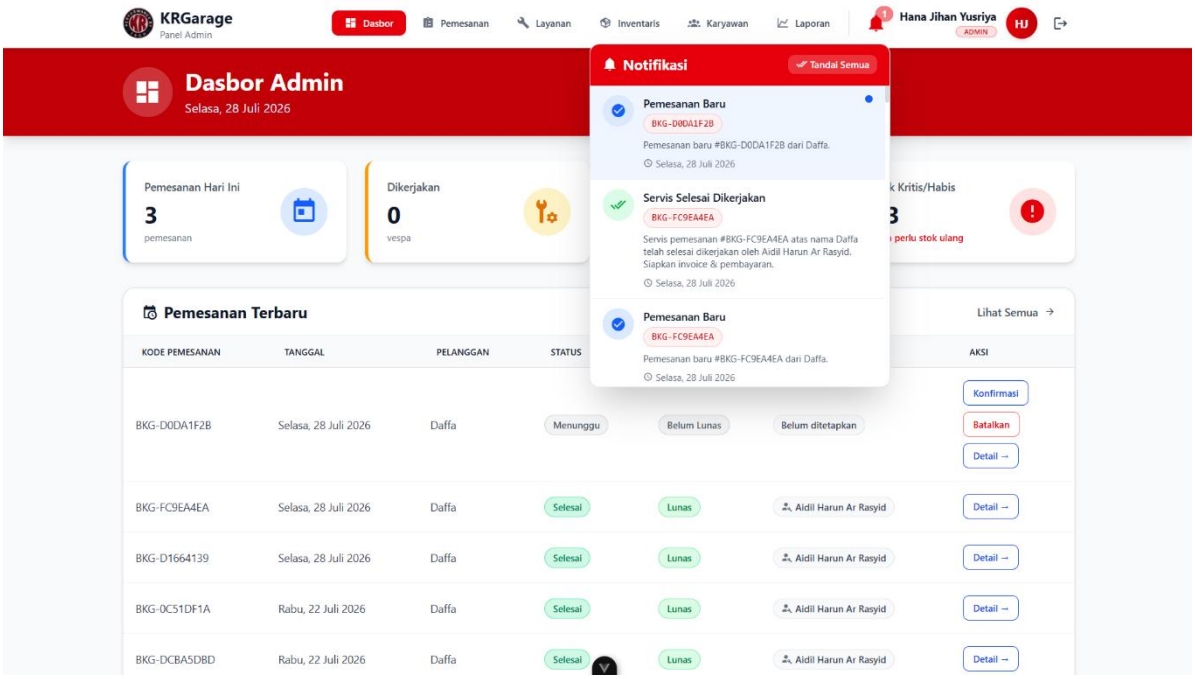


Gambar 4.28 Antarmuka Halaman *Dashboard* Admin

Halaman *dashboard* admin pada Gambar 4.28 menjadi tampilan utama yang digunakan admin setelah berhasil masuk ke dalam sistem. Halaman ini mendukung kebutuhan admin pada Tabel 4.1 dalam memantau kondisi operasional bengkel secara cepat. Pada bagian atas halaman, sistem menampilkan beberapa kartu indikator yang memuat jumlah pemesanan hari ini, jumlah Vespa yang sedang dikerjakan, jumlah pemesanan yang telah selesai hari ini, serta jumlah suku cadang dengan status stok kritis atau habis.

Selain menampilkan ringkasan operasional, halaman *dashboard* juga menyajikan tabel pemesanan terbaru yang berisi daftar transaksi yang sedang berlangsung maupun yang baru diterima. Informasi yang ditampilkan meliputi kode pemesanan, tanggal servis, nama pelanggan, status pengerjaan, status pembayaran, serta mekanik yang ditugaskan. Admin juga dapat menggunakan tombol aksi seperti “Konfirmasi”, “Batalkan”, “Tandai Lunas”, dan “Detail” untuk mempercepat pengelolaan pemesanan servis dan transaksi. Dengan adanya halaman ini, admin dapat memantau aktivitas bengkel secara terpusat tanpa harus membuka setiap menu secara terpisah.

b. Notifikasi Lonceng Admin

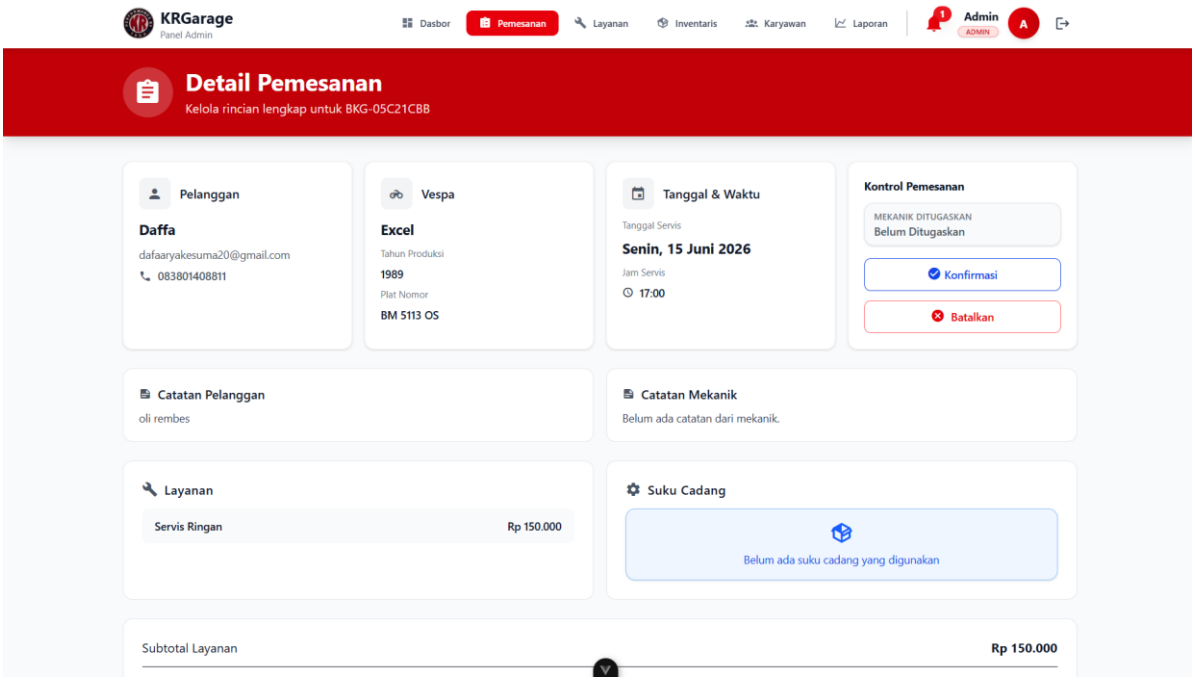


Gambar 4.29 Antarmuka Lonceng Notifikasi Pemesanan Admin

Antarmuka pada Gambar 4.29 menampilkan komponen *pop-up* lonceng notifikasi yang terdapat pada bagian *header* panel admin KRGarage. Antarmuka ini berfungsi sebagai pusat pemberitahuan *real-time* bagi admin untuk memantau setiap transaksi pemesanan servis baru yang diajukan oleh pelanggan maupun status pengerjaan unit Vespa oleh mekanik.

Dalam antarmuka ini, diterapkan aturan logika bisnis khusus admin untuk menjamin kepastian penanganan transaksi. Ketika notifikasi pemesanan baru masuk, angka *badge* merah di atas ikon lonceng tidak akan hilang atau berkurang hanya dengan sekadar diklik atau dibaca oleh admin. Angka *badge* indikator tersebut baru akan berkurang secara otomatis setelah admin melakukan aksi penanganan pemesanan secara nyata di dalam sistem, yaitu melalui penekanan tombol aksi "Konfirmasi" atau "Batalkan" pemesanan. Penerapan logika ini bertujuan untuk memastikan setiap pemesanan masuk mendapatkan kepastian tindakan dari admin dan tidak ada transaksi pelanggan yang terlewat tanpa penanganan.

c. Halaman Kelola Transaksi

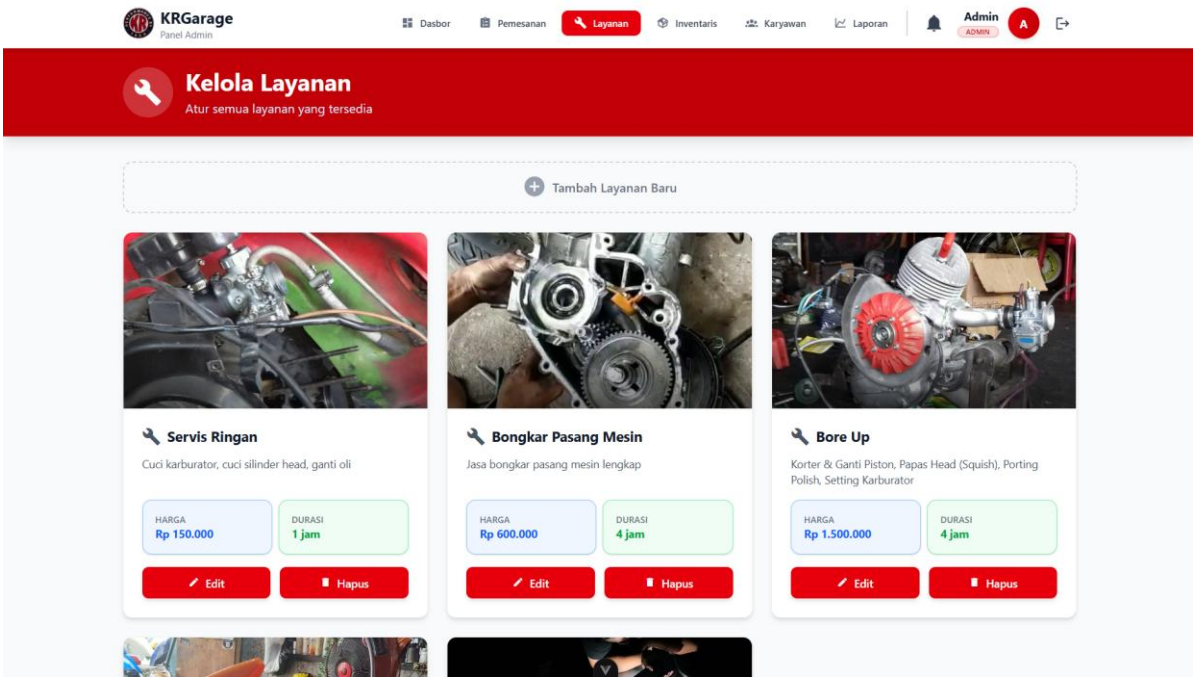


Gambar 4.30 Antarmuka Halaman Kelola Transaksi

Pengelolaan transaksi servis dilakukan melalui halaman kelola transaksi seperti yang ditampilkan pada Gambar 4.30. Halaman ini digunakan admin untuk melihat informasi lengkap dari suatu pemesanan servis, mulai dari data pelanggan, data Vespa, tanggal dan waktu servis, catatan pelanggan, hingga rincian layanan yang dipilih. Informasi tersebut membantu admin melakukan verifikasi data pemesanan sebelum servis dikerjakan oleh mekanik.

Selain digunakan untuk melihat informasi pemesanan, halaman ini juga mendukung proses pengelolaan awal servis. Admin dapat melakukan konfirmasi pemesanan, memilih mekanik yang akan ditugaskan, serta memantau status pengerjaan pemesanan. Pada bagian bawah halaman ditampilkan rincian biaya layanan, biaya suku cadang apabila digunakan, dan total biaya yang harus dibayarkan pelanggan. Dengan demikian, halaman kelola transaksi membantu admin memastikan bahwa data pemesanan telah sesuai sebelum servis diproses lebih lanjut oleh mekanik.

d. Halaman Kelola Layanan

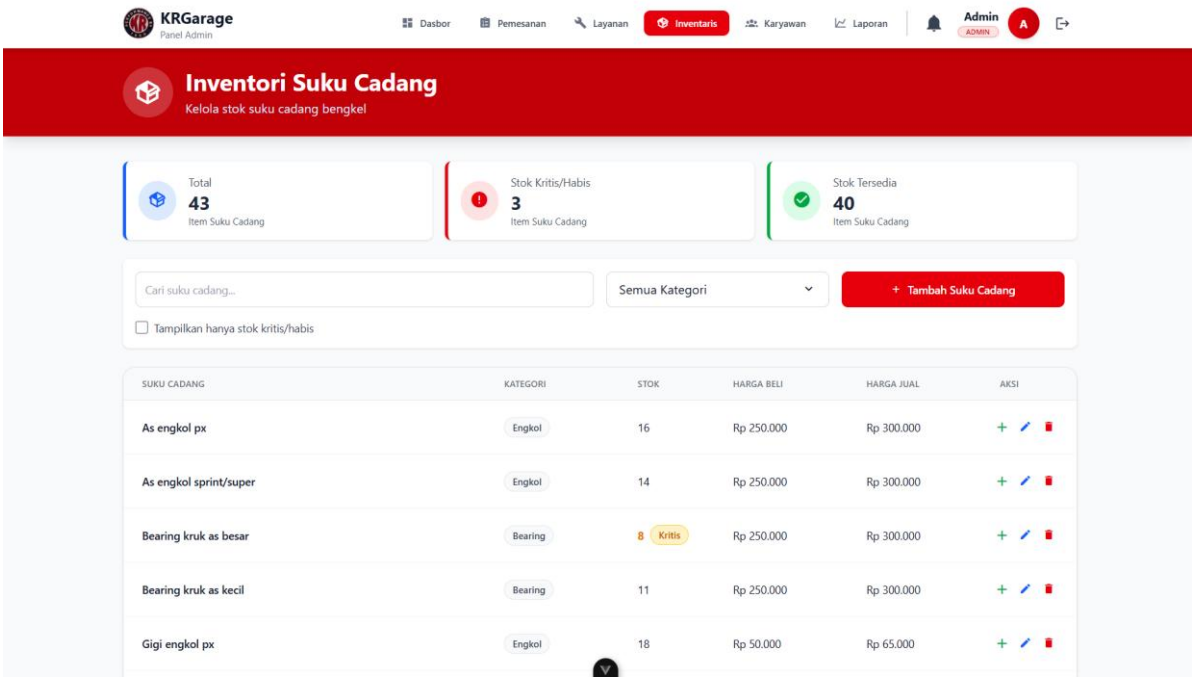


Gambar 4.31 Antarmuka Halaman Kelola Layanan

Kebutuhan admin dalam mengelola data layanan servis diwujudkan melalui halaman kelola layanan pada Gambar 4.31. Setiap layanan ditampilkan dalam bentuk kartu yang memuat gambar layanan, nama layanan, deskripsi singkat, harga jasa, serta estimasi durasi pengerjaan. Penyajian data layanan dalam bentuk kartu memudahkan admin untuk memeriksa informasi layanan yang tersedia sebelum ditampilkan kepada pelanggan.

Pada halaman ini, admin dapat menambahkan, mengubah, dan menghapus data layanan sesuai kebutuhan operasional bengkel. Perubahan data layanan mencakup jenis layanan, deskripsi, harga, dan estimasi waktu pengerjaan. Dengan adanya fitur ini, informasi layanan yang tersedia pada sistem dapat selalu disesuaikan dengan kondisi layanan bengkel.

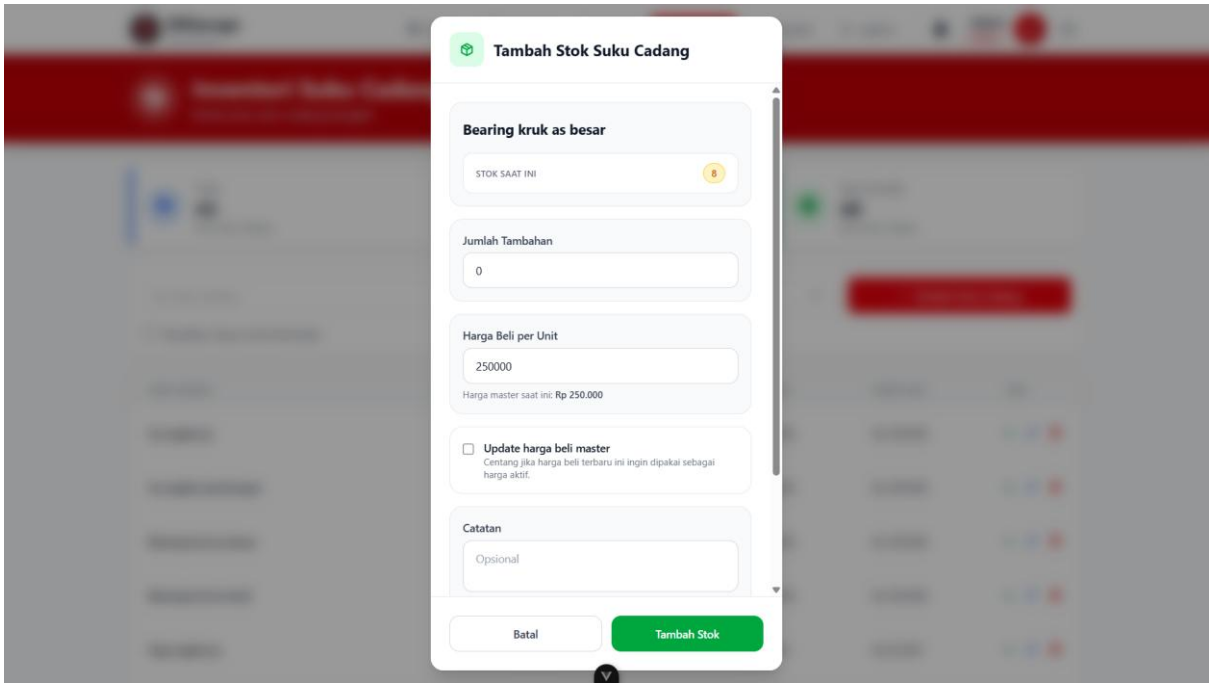
e. Halaman Kelola Suku Cadang



Gambar 4.32 Antarmuka Halaman Kelola Suku Cadang

Pengelolaan inventaris suku cadang dilakukan melalui halaman kelola suku cadang seperti yang ditampilkan pada Gambar 4.32. Halaman ini mendukung kebutuhan admin pada Tabel 4.1 dalam memantau ketersediaan stok dan mengurangi risiko kekurangan suku cadang. Pada bagian atas halaman, sistem menampilkan ringkasan jumlah stok suku cadang, jumlah suku cadang dengan stok kritis atau habis, serta jumlah suku cadang yang masih tersedia.

Selain ringkasan stok, sistem juga menyediakan tabel inventaris yang menampilkan nama suku cadang, kategori, jumlah stok, harga beli, dan harga jual. Admin dapat melakukan pencarian dan penyaringan data berdasarkan kategori tertentu. Pada setiap data suku cadang, tersedia aksi untuk menambah stok, mengubah data, maupun menghapus data suku cadang. Dengan adanya halaman ini, proses pengelolaan inventaris menjadi lebih terorganisir dan memudahkan admin dalam mengendalikan ketersediaan barang.



Gambar 4.33 Antarmuka Halaman *Restock* Suku Cadang

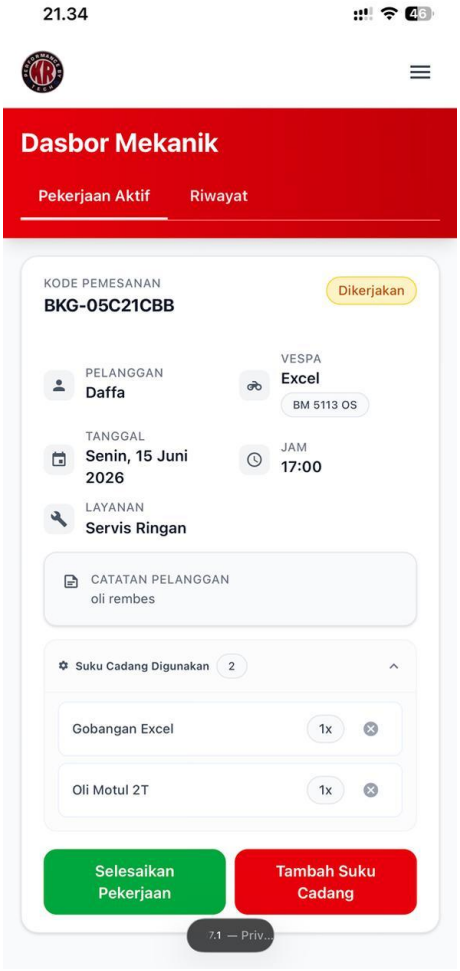
Penambahan stok suku cadang dilakukan melalui halaman *restock* suku cadang sebagaimana ditampilkan pada Gambar 4.33. Halaman ini menampilkan informasi stok saat ini agar admin dapat mengetahui jumlah persediaan sebelum melakukan penambahan stok. Informasi tersebut membantu admin memastikan bahwa data penambahan stok sesuai dengan kondisi inventaris yang ada.

Pada halaman ini, admin dapat memasukkan jumlah stok yang akan ditambahkan, harga beli per unit, serta catatan tambahan apabila diperlukan. Sistem juga menyediakan opsi untuk memperbarui harga beli pada data master suku cadang berdasarkan transaksi pembelian terbaru. Setelah data disimpan, jumlah stok suku cadang akan bertambah secara otomatis dan tercatat pada sistem. Fitur ini membantu menjaga keakuratan data inventaris sekaligus mendukung pencatatan pengeluaran pembelian suku cadang secara lebih terstruktur.

4.5.4 Implementasi Antarmuka Mekanik

Antarmuka mekanik diimplementasikan untuk memenuhi kebutuhan operasional mekanik yang telah dirumuskan pada Tabel 4.1, yaitu melihat daftar pekerjaan servis, memahami keluhan pelanggan, mencatat penggunaan suku cadang, serta memperbarui status pengerjaan. Antarmuka ini dirancang sederhana dan responsif agar mekanik dapat mengakses informasi pekerjaan melalui perangkat seluler saat berada di area bengkel.

a. Halaman *Dashboard* Mekanik



Gambar 4.34 Antarmuka Halaman *Dashboard* Mekanik

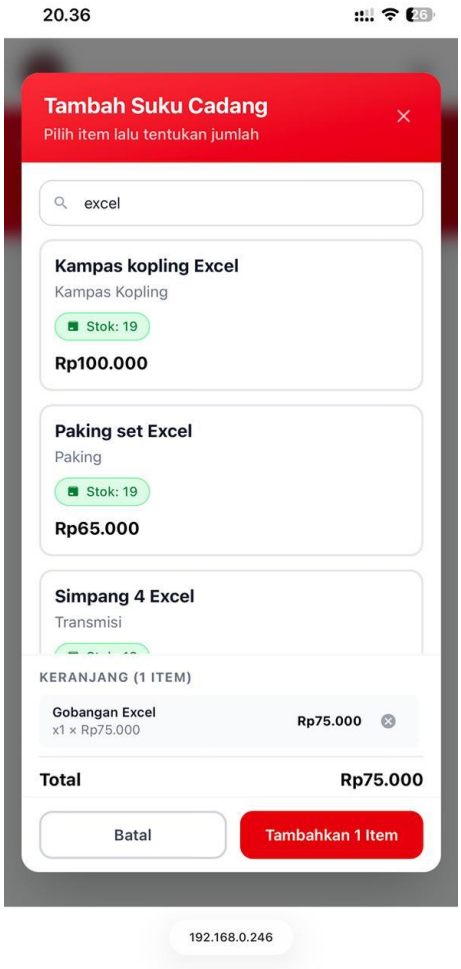
Untuk membantu mekanik memantau pekerjaan yang menjadi tanggung jawabnya, sistem menyediakan halaman *dashboard* mekanik seperti yang ditampilkan pada Gambar 4.34. Pada bagian atas halaman terdapat dua tab utama, yaitu Pekerjaan Aktif dan Riwayat. Tab Pekerjaan Aktif menampilkan pemesanan servis yang masih dalam proses pengerjaan, sedangkan tab Riwayat digunakan untuk melihat daftar pekerjaan yang telah selesai dikerjakan.

Pada tampilan pekerjaan aktif, setiap pemesanan ditampilkan dalam bentuk kartu yang berisi informasi penting, seperti kode pemesanan, nama pelanggan, data Vespa, tanggal dan jam servis, jenis layanan, serta catatan yang diberikan pelanggan. Informasi tersebut membantu mekanik memahami kebutuhan servis sebelum melakukan pengerjaan. Selain itu, sistem juga menampilkan daftar suku cadang yang telah digunakan pada pekerjaan tersebut beserta jumlah penggunaannya.

Halaman ini juga menyediakan tombol aksi Selesaikan Pekerjaan dan Tambah Suku Cadang. Tombol Selesaikan Pekerjaan digunakan untuk memperbarui status servis menjadi selesai dengan menambahkan catatan mekanik sebagai laporan hasil pengerjaan. Sementara

itu, tombol Tambah Suku Cadang digunakan untuk mencatat komponen yang digunakan selama proses servis. Dengan adanya halaman ini, pekerjaan mekanik dapat dikelola secara lebih terstruktur dan seluruh data pengerjaan terdokumentasi di dalam sistem.

b. Antarmuka *Pop-up* Penambahan Suku Cadang



Gambar 4.35 Antarmuka *Pop-up* Penambahan Suku Cadang

Pencatatan penggunaan suku cadang dilakukan melalui antarmuka *pop-up* penambahan suku cadang seperti yang ditampilkan pada Gambar 4.35. Antarmuka ini muncul ketika mekanik menekan tombol Tambah Suku Cadang pada halaman pekerjaan. Sistem menampilkan daftar suku cadang yang tersedia, lengkap dengan informasi nama barang, kategori, jumlah stok, dan harga.

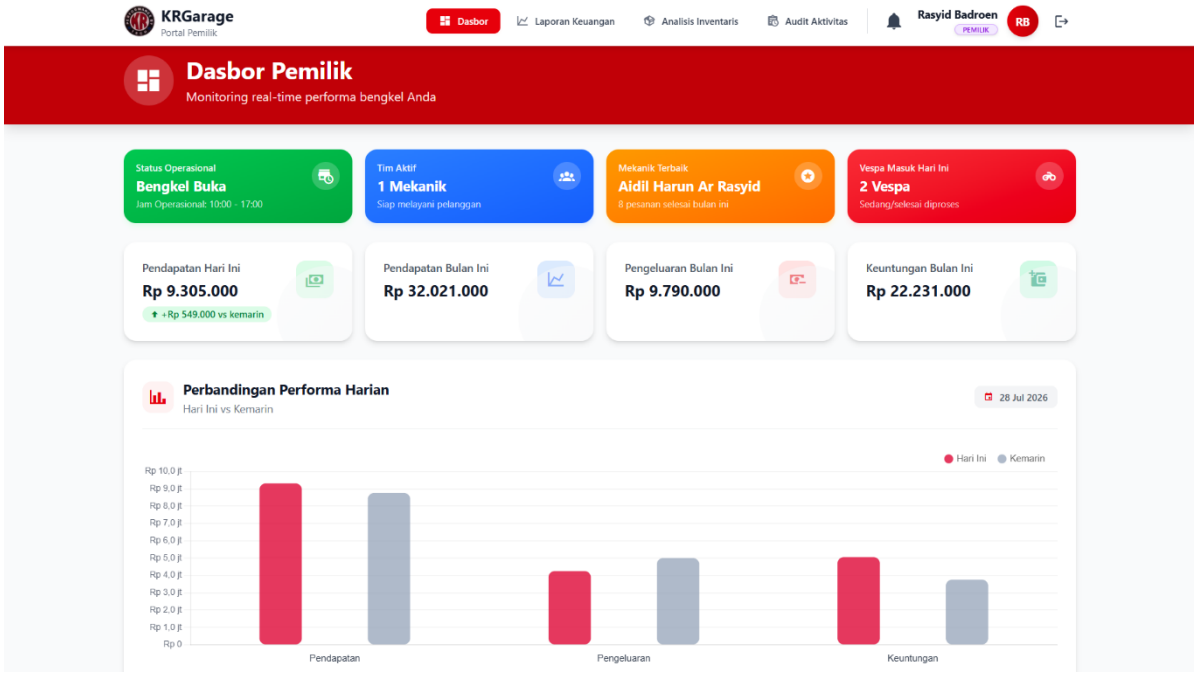
Pada antarmuka ini, mekanik dapat menggunakan fitur pencarian untuk menemukan suku cadang berdasarkan nama atau kategori. Setelah suku cadang dipilih, sistem akan menambahkannya ke bagian keranjang dan menampilkan jumlah item serta total biaya suku cadang yang digunakan. Mekanik dapat membatalkan proses melalui tombol Batal atau menyimpan data melalui tombol Tambahkan Item. Dengan adanya fitur ini, penggunaan suku

cadang dapat tercatat lebih akurat, biaya tambahan pada pemesanan dapat dihitung secara otomatis, dan data inventaris dapat diperbarui sesuai penggunaan selama proses servis.

4.5.5 Implementasi Antarmuka Pemilik

Antarmuka pemilik diimplementasikan untuk memenuhi kebutuhan pemilik bengkel yang telah dirumuskan pada Tabel 4.1, khususnya dalam memantau kondisi operasional, laporan keuangan, performa layanan, kondisi inventaris, dan aktivitas pengguna pada sistem. Antarmuka ini terdiri dari empat menu utama, yaitu *Dashboard* Pemilik, Laporan Keuangan, Analisis Inventaris, dan Audit Aktivitas.

a. Halaman *Dashboard* Pemilik



Gambar 4.36 Antarmuka *Dashboard* Pemilik

Halaman *dashboard* pemilik pada Gambar 4.36 digunakan untuk membantu pemilik memantau kondisi operasional dan performa bengkel secara ringkas. Pada bagian atas halaman ditampilkan beberapa informasi utama, seperti status operasional bengkel, jumlah mekanik aktif, mekanik terbaik, serta jumlah Vespa yang masuk pada hari tersebut. Informasi ini membantu pemilik mengetahui keadaan bengkel secara cepat tanpa harus membuka data satu per satu.

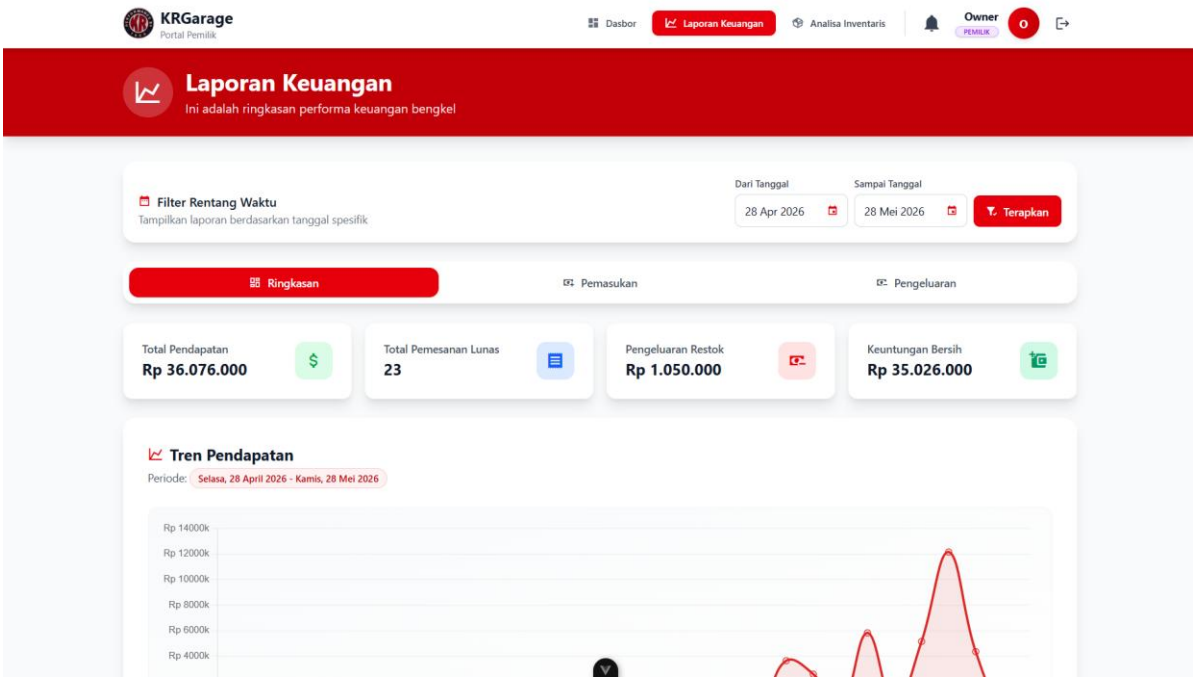
Selain menampilkan status operasional, halaman ini juga menyajikan ringkasan keuangan dalam bentuk kartu statistik. Data yang ditampilkan meliputi pendapatan hari ini, pendapatan bulan ini, pengeluaran bulan ini, dan keuntungan bulan ini. Khusus pada kartu

pendapatan hari ini, dilengkapi dengan *badge* indikator selisih nominal Rupiah harian untuk memberikan gambaran evaluasi pendapatan secara langsung bagi pemilik usaha. Penyajian informasi tersebut mendukung kebutuhan pemilik pada Tabel 4.1 dalam memantau perkembangan keuangan bengkel secara lebih terstruktur.

Di samping kartu statistik, halaman ini dilengkapi dengan *widget* Grafik Perbandingan Performa Harian berbasis grafik batang berdampingan. Grafik ini secara interaktif menyajikan komparasi visual tiga variabel finansial penting, yaitu Pendapatan, Pengeluaran, dan Keuntungan antara pencapaian Hari Ini dan pencapaian Hari Kemarin. Visualisasi grafik yang intuitif ini memudahkan pemilik dalam menganalisis perkembangan arus kas harian serta mengevaluasi efisiensi operasional bengkel secara efisien.

Pada bagian bawah halaman terdapat tabel Aktivitas Terbaru yang menampilkan beberapa pemesanan terakhir. Tabel ini menampilkan informasi kode pemesanan, tanggal, nama pelanggan, layanan, total biaya, status pengerjaan, dan status pembayaran. Fitur ini berfungsi untuk memberikan gambaran terbaru mengenai aktivitas servis yang sedang berlangsung maupun yang telah selesai. Dengan adanya halaman *dashboard* pemilik, pemilik bengkel dapat melakukan pemantauan operasional, keuangan, dan aktivitas pemesanan secara terpusat dalam satu tampilan.

b. Halaman Laporan Keuangan



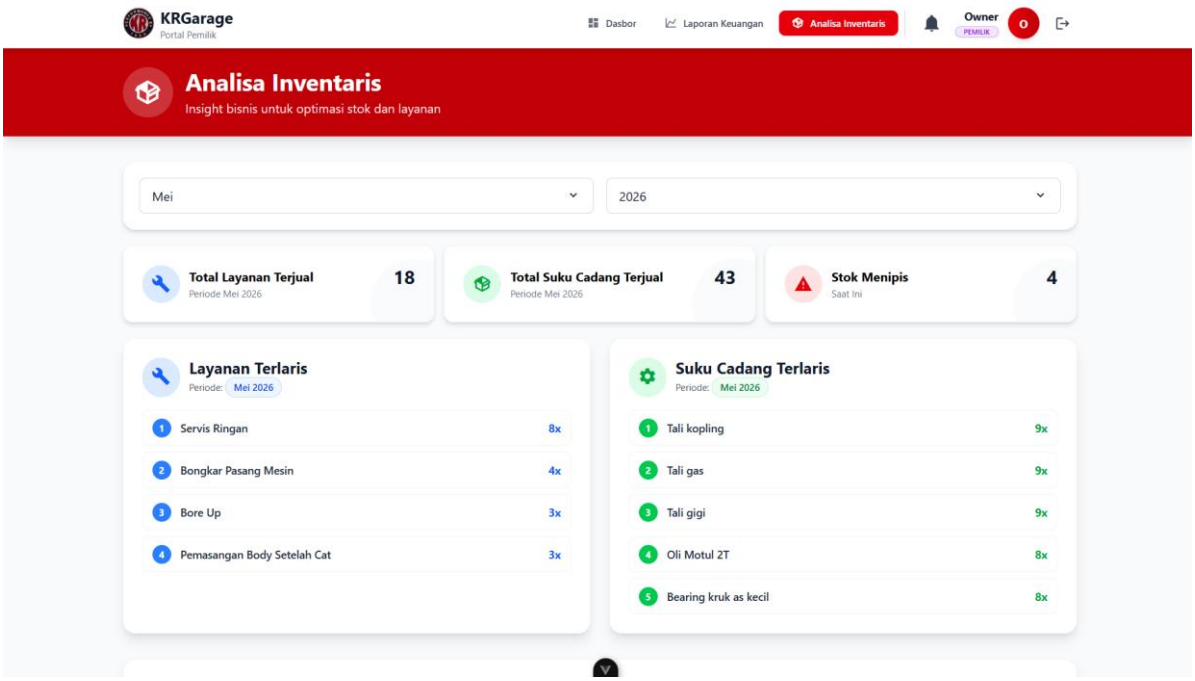
Gambar 4.37 Antarmuka Halaman Laporan Keuangan

Kebutuhan pemilik dalam memantau pendapatan, pengeluaran, dan keuntungan bengkel diwujudkan melalui halaman laporan keuangan seperti yang ditampilkan pada Gambar 4.37. Pada bagian atas halaman terdapat fitur filter rentang waktu yang memungkinkan pemilik menentukan tanggal awal dan tanggal akhir laporan. Setelah rentang tanggal dipilih, sistem akan menampilkan data keuangan sesuai periode yang ditentukan.

Halaman ini menyediakan tiga bagian utama, yaitu Ringkasan, Pemasukan, dan Pengeluaran. Pada tab Ringkasan, sistem menampilkan beberapa informasi penting seperti total pendapatan, total pemesanan yang sudah lunas, pengeluaran *restock*, dan keuntungan bersih. Data tersebut dihitung berdasarkan transaksi pemesanan serta pengeluaran *restock* suku cadang pada rentang waktu yang dipilih.

Selain ringkasan angka, halaman ini juga menampilkan grafik Tren Pendapatan yang menggambarkan perkembangan pendapatan bengkel dari waktu ke waktu. Grafik tersebut membantu pemilik melihat pola pemasukan dalam periode tertentu, sehingga dapat digunakan sebagai bahan evaluasi performa usaha. Sementara itu, tab Pemasukan digunakan untuk melihat daftar transaksi pemesanan, sedangkan tab Pengeluaran digunakan untuk melihat data pengeluaran *restock* suku cadang. Dengan adanya halaman laporan keuangan, pemilik dapat melakukan pemantauan pendapatan, pengeluaran, dan keuntungan bengkel secara lebih sistematis.

c. Halaman Analisis Inventaris



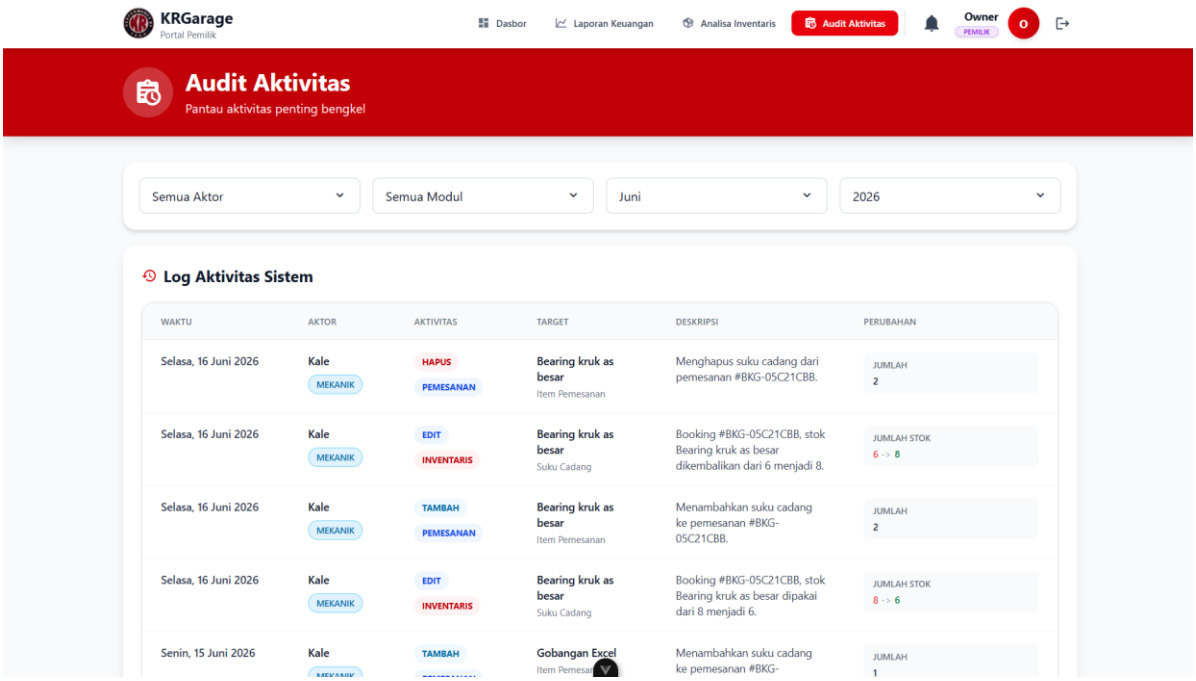
Gambar 4.38 Antarmuka Halaman Analisis Inventaris

Halaman analisis inventaris pada Gambar 4.38 digunakan untuk membantu pemilik memantau performa layanan dan penggunaan suku cadang. Halaman ini menyediakan filter berdasarkan bulan dan tahun, sehingga pemilik dapat melihat data analisis inventaris sesuai periode yang diinginkan. Dengan adanya filter tersebut, pemilik dapat membandingkan kebutuhan layanan dan suku cadang pada setiap periode secara lebih terarah.

Pada bagian ringkasan, sistem menampilkan total layanan terjual, total suku cadang terjual, serta jumlah stok yang menipis. Informasi ini membantu pemilik mengetahui tingkat permintaan layanan dan penggunaan komponen suku cadang selama periode tertentu. Selain itu, halaman ini juga menampilkan daftar Layanan Terlaris dan Suku Cadang Terlaris yang diurutkan berdasarkan frekuensi penggunaan atau penjualan pada periode yang dipilih. Informasi ini dapat digunakan sebagai bahan evaluasi untuk mengetahui tren permintaan pelanggan serta menentukan prioritas penyediaan stok suku cadang.

Halaman ini juga dilengkapi dengan informasi Stok Menipis, yaitu daftar suku cadang yang jumlah stoknya telah mencapai batas minimum yang telah ditentukan. Informasi tersebut berfungsi sebagai peringatan dini bagi pemilik untuk segera melakukan *restock* suku cadang sebelum stok habis dan mengganggu proses pelayanan bengkel. Dengan adanya fitur ini, pemilik dapat melakukan pengelolaan inventaris secara lebih proaktif dan menjaga ketersediaan suku cadang yang sering digunakan.

d. Halaman Audit Aktivitas



Gambar 4.39 Antarmuka Halaman Audit Aktivitas

Untuk mendukung pemantauan aktivitas pengguna, sistem menyediakan halaman audit aktivitas seperti yang ditampilkan pada Gambar 4.39. Halaman ini menampilkan riwayat aktivitas yang terjadi di dalam sistem, meliputi waktu aktivitas, nama pengguna, peran pengguna, modul, jenis aksi, serta deskripsi aktivitas yang dilakukan.

Fitur audit aktivitas membantu pemilik menelusuri perubahan data dan aktivitas operasional yang dilakukan oleh pengguna sistem. Dengan adanya halaman ini, proses pemantauan menjadi lebih transparan dan terdokumentasi, sehingga pemilik dapat mengetahui aktivitas penting yang terjadi pada sistem.

4.6 Pengujian Sistem

Setelah tahap implementasi antarmuka dan logika pemrograman selesai, dilakukan pengujian untuk mengetahui kesesuaian fungsi sistem dengan kebutuhan yang telah ditetapkan. Pengujian menggunakan metode *Black Box Testing*, yaitu pengujian yang berfokus pada masukan dan keluaran sistem tanpa menganalisis struktur internal kode program.

Pengujian dilakukan melalui antarmuka aplikasi dan pengiriman permintaan langsung ke API menggunakan Postman. Pengujian melalui antarmuka aplikasi bertujuan untuk memastikan fitur dapat digunakan oleh pengguna sesuai alur sistem. Sementara itu, pengujian melalui API bertujuan untuk memastikan bahwa validasi data tetap berjalan pada sisi *backend* meskipun permintaan tidak dikirim melalui antarmuka *frontend*.

Skenario pengujian mencakup registrasi, *login*, pembatasan hak akses, fitur pelanggan, fitur mekanik, fitur admin, dan fitur pemilik. Pengujian juga meliputi validasi *input*, pengelolaan data Vespa, pengelolaan pemesanan servis, penggunaan suku cadang, perubahan status pengerjaan dan pembayaran, pengelolaan persediaan, *restock* suku cadang, pengelolaan karyawan, laporan keuangan, analisis inventaris, serta audit aktivitas.

Pengujian *Black Box Testing* dilakukan dalam dua tahap, yaitu pengujian awal dan pengujian ulang setelah perbaikan. Pengujian awal dilakukan untuk mengetahui apakah setiap fitur sistem telah menghasilkan luaran yang sesuai dengan skenario pengujian. Pada tahap ini, sebagian besar skenario telah berjalan sesuai harapan, namun terdapat beberapa skenario yang belum menghasilkan luaran sesuai dengan hasil yang diharapkan, khususnya pada pengujian validasi *input* melalui API menggunakan Postman.

Skenario yang belum berhasil pada pengujian awal berkaitan dengan validasi data registrasi dan pengelolaan data Vespa. Pada proses registrasi, sistem pada sisi *backend* masih menerima beberapa *input* yang seharusnya ditolak, seperti nama yang terlalu pendek, registrasi tanpa email, nomor telepon dengan format tidak valid, dan *password* lemah. Sementara itu, pada pengelolaan data Vespa, *backend* masih menerima beberapa data tidak valid, seperti plat nomor dengan karakter khusus, plat nomor terlalu pendek atau terlalu panjang, model Vespa terlalu pendek, serta tahun produksi di luar rentang yang ditentukan.

Berdasarkan temuan tersebut, dilakukan perbaikan pada aturan validasi *backend* agar sesuai dengan ketentuan *input* yang telah ditetapkan. Setelah perbaikan dilakukan, skenario yang sebelumnya belum berhasil diuji kembali pada tahap pengujian ulang. Hasil pengujian ulang menunjukkan bahwa sistem telah mampu menolak *input* yang tidak valid dan menampilkan pesan validasi sesuai harapan. Dengan demikian, skenario yang sebelumnya belum berhasil dinyatakan berhasil setelah dilakukan perbaikan dan pengujian ulang.

Tabel 4.10 Hasil Pengujian Registrasi, *Login*, Profil, dan Hak Akses

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Registrasi pelanggan dengan data valid	Isi nama, email, nomor HP, dan *password* sesuai format, lalu klik daftar.	Akun pelanggan berhasil dibuat dan sistem menampilkan pesan registrasi berhasil.	Akun berhasil dibuat dengan *role* pelanggan dan sistem menampilkan pesan registrasi berhasil.	Berhasil
2	Registrasi dengan nama terlalu pendek melalui API.	Kirim data registrasi melalui Postman	Sistem menolak registrasi	Pada pengujian awal, API masih menerima nama	Berhasil setelah

		menggunakan nama A.	karena nama minimal 3 karakter.	satu karakter. Setelah validasi *backend* diperbaiki, sistem menolak registrasi dengan nama satu karakter dan menampilkan pesan validasi.	pengujian ulang
3	Registrasi dengan data kosong.	Kosongkan salah satu atau seluruh data registrasi, lalu klik daftar.	Sistem menampilkan pesan bahwa data wajib diisi dan akun tidak dibuat.	Sistem menampilkan pesan validasi pada data yang belum diisi dan akun tidak dibuat.	Berhasil
4	Registrasi dengan format email tidak valid.	Masukkan email seperti pelanggan@, lalu klik daftar.	Sistem menampilkan pesan bahwa format email tidak valid.	Sistem menampilkan pesan bahwa format email tidak valid dan proses registrasi tidak dilanjutkan.	Berhasil
5	Registrasi dengan email yang telah digunakan.	Masukkan email yang sudah terdaftar.	Sistem menolak registrasi karena email telah digunakan.	Sistem menolak registrasi dan menampilkan pesan bahwa email telah digunakan	Berhasil
6	Registrasi tanpa email melalui API.	Kirim data registrasi melalui Postman tanpa mengisi email.	Sistem menolak registrasi karena email wajib diisi.	Pada pengujian awal, API masih menerima registrasi tanpa email. Setelah validasi *backend* diperbaiki, sistem menolak registrasi tanpa email dan menampilkan pesan validasi bahwa email wajib diisi.	Berhasil setelah pengujian ulang
7	Registrasi dengan format nomor HP tidak valid.	Masukkan nomor HP seperti 12345 atau mengandung huruf.	Sistem menampilkan pesan bahwa format nomor HP tidak valid.	Sistem menampilkan pesan bahwa format nomor HP tidak valid dan akun tidak dibuat.	Berhasil

8	Registrasi dengan format nomor HP tidak valid melalui API.	Kirim data registrasi melalui Postman dengan nomor HP 12345.	Sistem menolak registrasi dan menampilkan validasi format nomor HP.	Pada pengujian awal, API masih menerima nomor HP dengan format tidak valid. Setelah validasi *backend* diperbaiki, sistem menolak nomor HP yang tidak sesuai format dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
9	Registrasi dengan nomor HP yang telah digunakan.	Masukkan nomor HP yang sudah terdaftar.	Sistem menolak registrasi karena nomor HP telah digunakan.	Sistem menolak registrasi dan menampilkan pesan bahwa nomor HP telah terdaftar.	Berhasil
10	Registrasi dengan *password* kurang dari 8 karakter.	Masukkan *password* seperti Abc123.	Sistem menampilkan pesan bahwa *password* minimal 8 karakter.	Sistem menampilkan pesan bahwa *password* minimal 8 karakter dan akun tidak dibuat.	Berhasil
11	Registrasi dengan *password* lemah melalui API.	Kirim data registrasi melalui Postman menggunakan *password* abcdefgh.	Sistem menolak registrasi karena *password* tidak memiliki huruf besar dan angka.	Pada pengujian awal, API masih menerima *password* yang hanya memenuhi panjang minimal karakter. Setelah validasi *backend* diperbaiki, sistem menolak *password* yang tidak memiliki kombinasi huruf besar, huruf kecil, dan angka.	Berhasil setelah pengujian ulang
12	Registrasi dengan format *password* tidak sesuai.	Masukkan *password* tanpa huruf besar, huruf kecil, atau angka.	Sistem menampilkan pesan mengenai ketentuan *password* yang belum terpenuhi.	Sistem menampilkan ketentuan *password* yang belum terpenuhi dan akun tidak dibuat.	Berhasil

13	*Login* dengan akun admin	Masukkan email dan *password* admin yang valid	Sistem masuk ke *dashboard* admin	Sistem menerima data *login* dan menampilkan *dashboard* admin	Berhasil
14	*Login* dengan akun pemilik	Masukkan email dan *password* pemilik yang valid	Sistem masuk ke *dashboard* pemilik	Sistem menerima data *login* dan menampilkan *dashboard* pemilik	Berhasil
15	*Login* dengan akun mekanik	Masukkan email dan *password* mekanik yang valid	Sistem masuk ke *dashboard* mekanik	Sistem menerima data *login* dan menampilkan *dashboard* mekanik	Berhasil
16	*Login* dengan akun pelanggan	Masukkan email dan *password* pelanggan yang valid	Sistem masuk ke *dashboard* pelanggan	Sistem menerima data *login* dan menampilkan *dashboard* pelanggan	Berhasil
17	*Login* dengan data salah	Masukkan email dan *password* yang salah	Sistem menampilkan pesan *error* dan tetap di halaman *login*	Sistem menampilkan pesan bahwa email atau *password* salah dan pengguna tetap berada di halaman *login*	Berhasil
18	Akses halaman *role* lain	*Login* sebagai pelanggan lalu akses URL admin	Sistem menolak akses atau mengarah ke halaman sesuai *role*	Sistem menolak akses pelanggan ke halaman admin dan mengarahkan pengguna ke halaman yang sesuai dengan *role*-nya	Berhasil
19	Mengubah profil dengan data valid	Klik profil, ubah nama, email, atau nomor HP menggunakan data valid, lalu klik simpan	Sistem menyimpan perubahan data profil	Perubahan profil berhasil disimpan dan ditampilkan pada sistem	Berhasil
20	Mengubah profil dengan nama terlalu pendek	Masukkan nama satu karakter, misalnya A, lalu klik simpan profil	Sistem menampilkan validasi bahwa nama minimal 3 karakter dan	Sistem menampilkan pesan bahwa nama minimal 3 karakter dan proses	Berhasil

			data tidak disimpan	penyimpanan tidak dilanjutkan	
21	Mengubah profil dengan nama terlalu pendek melalui API	Kirim perubahan profil melalui Postman menggunakan nama A	*Backend* menolak perubahan karena nama minimal 3 karakter	API menolak perubahan dengan status 422 *Unprocessable Content*, menampilkan pesan validasi nama, dan data profil tidak berubah	Berhasil
22	Mengubah profil dengan format nomor HP tidak valid	Masukkan nomor HP seperti 12345, lalu klik simpan profil	Sistem menampilkan validasi bahwa format nomor HP tidak valid	Sistem menampilkan pesan format nomor HP tidak valid dan data profil tidak disimpan	Berhasil
23	Mengubah profil dengan format nomor HP tidak valid melalui API	Kirim perubahan profil melalui Postman menggunakan nomor HP 12345	*Backend* menolak perubahan karena format nomor HP tidak valid	API menolak perubahan dengan status 422 *Unprocessable Content* lalu menampilkan pesan validasi nomor HP dan data profil tidak berubah	Berhasil
24	Mengubah profil menggunakan nomor HP yang telah digunakan melalui API	Kirim nomor HP yang telah digunakan oleh akun lain	*Backend* menolak perubahan karena nomor HP harus unik	API menolak perubahan dengan status 422 *Unprocessable Content*, menampilkan pesan bahwa nomor HP sudah digunakan, dan data profil tidak berubah	Berhasil
25	Mengubah nomor HP menggunakan format +628 melalui API	Kirim nomor HP valid seperti +6281234567890	Sistem menerima perubahan dan menormalisasi nomor menjadi format 08	API menerima perubahan dengan status 200 OK dan menyimpan nomor sebagai 081234567890	Berhasil
26	Mengganti *password* dengan data valid	Masukkan *password* lama yang benar,	Sistem berhasil mengganti *password*	Sistem menampilkan pesan bahwa	Berhasil

		password baru sesuai ketentuan, dan konfirmasi		*password* berhasil diganti dan *password* baru dapat digunakan untuk *login*	
27	Mengganti *password* dengan *password* lama yang salah	Masukkan *password* lama yang tidak sesuai dan *password* baru yang valid	Sistem menolak perubahan karena *password* lama tidak sesuai	Sistem menampilkan pesan bahwa *password* lama tidak sesuai dan *password* tidak diubah	Berhasil
28	Mengganti *password* kurang dari 8 karakter	Masukkan *password* baru kurang dari 8 karakter	Sistem menampilkan validasi bahwa *password* baru minimal 8 karakter	Sistem menampilkan pesan validasi dan perubahan *password* tidak diproses	Berhasil
29	Mengganti *password* tanpa huruf besar atau angka	Masukkan *password* baru seperti abcdefgh	Sistem menolak karena *password* tidak memenuhi ketentuan kombinasi karakter	Sistem menampilkan pesan bahwa *password* harus mengandung huruf besar dan angka	Berhasil
30	Mengganti *password* dengan konfirmasi tidak sesuai	Masukkan *password* baru dan konfirmasi *password* yang berbeda	Sistem menampilkan validasi bahwa konfirmasi *password* tidak cocok	Sistem menampilkan pesan bahwa konfirmasi *password* tidak cocok dan *password* tidak diubah	Berhasil
31	Mengganti *password* lemah melalui API	Kirim password baru abcdefgh	*Backend* menolak karena *password* tidak memiliki huruf besar dan angka	API menolak perubahan dengan status 422 *Unprocessable Content* dan *password* tidak diubah	Berhasil

Hasil pengujian pada Tabel 4.10 menunjukkan bahwa fungsi registrasi, *login*, pengelolaan profil, perubahan *password*, dan pembatasan hak akses telah berjalan sesuai kebutuhan setelah dilakukan pengujian awal dan pengujian ulang. Pada pengujian awal melalui API, ditemukan beberapa skenario yang belum berhasil pada proses registrasi, yaitu sistem masih menerima nama terlalu pendek, registrasi tanpa email, nomor HP tidak valid, dan *password* lemah. Kondisi tersebut menunjukkan bahwa validasi pada sisi *backend* belum sepenuhnya sesuai dengan ketentuan *input* yang ditetapkan.

Setelah temuan tersebut diperbaiki, skenario yang sebelumnya belum berhasil diuji kembali. Hasil pengujian ulang menunjukkan bahwa *backend* telah mampu menolak data yang tidak sesuai dan menampilkan pesan validasi sesuai harapan. Dengan demikian, seluruh skenario pada pengujian registrasi, *login*, profil, dan hak akses dinyatakan berhasil, baik pada skenario yang sejak awal sudah sesuai harapan maupun pada skenario yang berhasil setelah pengujian ulang.

Tabel 4.11 Hasil Pengujian Fitur Pelanggan

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menambah data Vespa	Isi model, tahun produksi, plat nomor, lalu simpan	Data Vespa berhasil ditambahkan	Data Vespa berhasil disimpan dan ditampilkan pada halaman Vespa Saya	Berhasil
2	Plat nomor otomatis kapital	Isi plat nomor dengan huruf kecil	Plat nomor tersimpan dan tampil dalam huruf kapital	Plat nomor yang dimasukkan menggunakan huruf kecil disimpan dan ditampilkan dalam huruf kapital	Berhasil
3	Menambah Vespa dengan data kosong	Kosongkan model, tahun produksi, atau plat nomor lalu simpan	Sistem menampilkan pesan validasi bahwa data Vespa wajib diisi	Sistem menampilkan pesan validasi pada data yang belum diisi dan data Vespa tidak disimpan	Berhasil
4	Menambah Vespa dengan plat nomor mengandung karakter tidak valid	Masukkan plat nomor seperti BM 1234@ABC, lalu simpan	Sistem menampilkan validasi bahwa plat nomor hanya boleh mengandung huruf dan angka	Sistem menampilkan pesan bahwa plat nomor hanya boleh mengandung huruf, angka, dan spasi sehingga data tidak disimpan	Berhasil
5	Menambah Vespa dengan karakter plat tidak valid melalui API	Kirim data Vespa melalui Postman dengan plat nomor BM 1234@ABC	Sistem menolak data karena plat nomor mengandung karakter khusus	Pada pengujian awal, API masih menerima plat nomor yang mengandung karakter khusus. Setelah validasi *backend* diperbaiki, sistem menolak plat nomor dengan karakter khusus dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
6	Menambah Vespa dengan	Masukkan plat nomor kurang	Sistem menampilkan validasi bahwa	Sistem menampilkan pesan bahwa plat nomor minimal 3	Berhasil

	plat nomor terlalu pendek	dari 3 karakter, misalnya BM	plat nomor minimal 3 karakter	karakter dan data tidak disimpan	
7	Menambah Vespa dengan plat nomor terlalu pendek melalui API	Kirim data Vespa melalui Postman dengan plat nomor BM	Sistem menolak data karena plat nomor kurang dari 3 karakter	Pada pengujian awal, API masih menerima plat nomor yang kurang dari 3 karakter. Setelah validasi *backend* diperbaiki, sistem menolak plat nomor terlalu pendek dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
8	Menambah Vespa dengan plat nomor terlalu panjang	Masukkan plat nomor lebih dari 15 karakter	Sistem menampilkan validasi bahwa plat nomor maksimal 15 karakter	Sistem menampilkan pesan bahwa plat nomor maksimal 15 karakter dan data tidak disimpan	Berhasil
9	Menambah Vespa dengan plat nomor terlalu panjang melalui API	Kirim plat nomor sepanjang 16 – 20 karakter melalui Postman	Sistem menolak data karena plat nomor melebihi 15 karakter	Pada pengujian awal, API masih menerima plat nomor yang melebihi batas maksimal karakter. Setelah validasi *backend* diperbaiki, sistem menolak plat nomor yang terlalu panjang dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
10	Menambah Vespa dengan plat nomor yang sudah terdaftar	Masukkan plat nomor yang sama dengan Vespa lain	Sistem menolak penyimpanan karena plat nomor harus unik	Sistem menolak penyimpanan dan menampilkan pesan bahwa plat nomor telah digunakan	Berhasil
11	Menambah Vespa dengan tahun produksi di bawah batas	Masukkan tahun produksi sebelum 1946, misalnya 1940	Sistem menampilkan validasi bahwa tahun produksi minimal 1946	Sistem menampilkan pesan bahwa tahun produksi minimal 1946 dan data tidak disimpan	Berhasil
12	Menambah Vespa dengan tahun produksi di bawah batas melalui API	Kirim data Vespa melalui Postman dengan tahun produksi 1940	Sistem menolak data karena tahun produksi kurang dari 1946	Pada pengujian awal, API masih menerima tahun produksi di bawah batas minimal. Setelah validasi *backend* diperbaiki, sistem menolak	Berhasil setelah pengujian ulang

				tahun produksi di bawah 1946 dan menampilkan pesan validasi.	
13	Menambah Vespa dengan tahun produksi melebihi batas	Masukkan tahun produksi lebih dari tahun berikutnya, misalnya 2028 pada pengujian tahun 2026	Sistem menampilkan validasi bahwa tahun produksi maksimal 2027	Sistem menampilkan pesan bahwa tahun produksi maksimal 2027 dan data tidak disimpan	Berhasil
14	Menambah Vespa dengan tahun produksi melebihi batas melalui API	Kirim data Vespa melalui Postman dengan tahun produksi 9999	Sistem menolak data karena tahun produksi melebihi batas	Pada pengujian awal, API masih menerima tahun produksi yang melebihi batas maksimal. Setelah validasi *backend* diperbaiki, sistem menolak tahun produksi yang melebihi batas tahun yang ditentukan dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
15	Menambah Vespa dengan format tahun bukan angka	Masukkan huruf atau karakter selain angka pada tahun produksi	Sistem tidak menerima karakter tersebut atau menampilkan validasi bahwa tahun harus berupa angka	Kolom tahun tidak menerima huruf atau karakter selain angka sehingga data tidak dapat disimpan	Berhasil
16	Menambah Vespa dengan model terlalu pendek	Masukkan model hanya satu karakter, misalnya P	Sistem menampilkan validasi bahwa model minimal 2 karakter	Sistem menampilkan pesan bahwa model minimal 2 karakter dan data tidak disimpan	Berhasil
17	Menambah Vespa dengan model terlalu pendek melalui API	Kirim data Vespa melalui Postman dengan model P	Sistem menolak data karena model kurang dari 2 karakter	Pada pengujian awal, API masih menerima model Vespa satu karakter. Setelah validasi *backend* diperbaiki, sistem menolak model Vespa yang terlalu pendek dan menampilkan pesan validasi.	Berhasil setelah pengujian ulang
18	Mengedit data Vespa	Ubah data Vespa lalu simpan	Data Vespa berhasil diperbarui	Perubahan data berhasil disimpan dan ditampilkan	Berhasil

				pada halaman Vespa Saya	
19	Menghapus data Vespa	Klik tombol hapus pada data Vespa	Data Vespa berhasil dihapus	Data Vespa berhasil dihapus dan tidak lagi ditampilkan	Berhasil
20	Membuat pemesanan servis	Pilih layanan, Vespa, dan jam	Pemesanan berhasil dibuat	Pemesanan berhasil dibuat dan ditampilkan pada riwayat pemesanan pelanggan	Berhasil
21	Membuat pemesanan servis dengan data valid melalui API	Kirim ID Vespa milik pelanggan, ID layanan yang valid, tanggal yang tersedia, dan jam operasional	Sistem menerima dan menyimpan pemesanan	API menerima permintaan, menyimpan pemesanan, dan mengembalikan *respons* berhasil	Berhasil
22	Membuat pemesanan dengan jadwal servis di luar jam operasional melalui API	Kirim pemesanan melalui Postman dengan jam_pemesanan pukul 23:00	Sistem menolak pemesanan karena jadwal servis berada di luar jam operasional bengkel	API menolak permintaan dengan status 422 lalu menampilkan pesan bahwa jam pemesanan tidak tersedia dan pemesanan tidak dibuat	Berhasil
23	Membuat pemesanan servis menggunakan layanan duplikat melalui API	Kirim pemesanan melalui Postman dengan ID layanan yang sama, misalnya [1, 1]	Sistem menolak pemesanan karena terdapat layanan yang sama	API menolak permintaan dengan status 422 lalu menampilkan pesan bahwa layanan tidak boleh dipilih lebih dari satu kali dan pemesanan tidak dibuat	Berhasil
24	Membuat pemesanan servis tanpa layanan	Buat pemesanan tanpa memilih layanan	Sistem menampilkan validasi layanan wajib dipilih	Sistem menampilkan validasi bahwa layanan wajib dipilih dan pemesanan tidak dibuat	Berhasil
25	Membuat pemesanan servis tanpa layanan melalui API	Kirim pemesanan melalui Postman dengan id_layanan kosong	Sistem menolak pemesanan karena layanan wajib dipilih	API menolak permintaan dengan status 422 lalu menampilkan pesan bahwa layanan wajib dipilih dan pemesanan tidak dibuat	Berhasil
26	Membuat pemesanan servis tanpa Vespa	Buat pemesanan tanpa memilih Vespa	Sistem menampilkan validasi Vespa wajib dipilih	Sistem menampilkan validasi bahwa Vespa wajib dipilih dan pemesanan tidak dibuat	Berhasil

27	Membuat pemesanan servis pada tanggal masa lalu	Pilih tanggal sebelum hari ini	Sistem menampilkan validasi bahwa tanggal tidak boleh di masa lalu	Sistem menolak tanggal sebelum hari pengujian dan pemesanan tidak dibuat	Berhasil
28	Membuat pemesanan servis pada hari libur	Pilih tanggal hari Jumat	Sistem menampilkan informasi bengkel libur	Sistem menampilkan informasi bahwa bengkel libur pada hari Jumat dan pemesanan tidak dapat dilanjutkan	Berhasil
29	Membuat pemesanan servis saat mekanik penuh	Pilih tanggal ketika seluruh mekanik sudah memiliki pekerjaan	Sistem menampilkan informasi bahwa semua mekanik sedang sibuk	Sistem menampilkan informasi bahwa seluruh mekanik sedang sibuk sehingga pemesanan tidak dapat dilanjutkan	Berhasil
30	Membuat pemesanan servis dengan catatan melebihi batas melalui API	Kirim pemesanan dengan catatan pelanggan lebih dari 1.000 karakter	Sistem menolak pemesanan karena catatan maksimal 1.000 karakter	API menolak permintaan dengan status 422 lalu menampilkan pesan bahwa catatan pelanggan melebihi batas maksimal dan pemesanan tidak dibuat	Berhasil
31	Melihat riwayat pemesanan servis	Buka menu riwayat	Sistem menampilkan daftar pemesanan pelanggan	Sistem menampilkan daftar pemesanan milik pelanggan beserta statusnya	Berhasil
32	Membatalkan pemesanan servis	Buka menu riwayat, pilih pemesanan aktif, lalu klik batalkan	Status pemesanan berubah menjadi batal	Status pemesanan aktif berubah menjadi Batal dan perubahan ditampilkan pada riwayat	Berhasil

Hasil pengujian pada Tabel 4.11 menunjukkan bahwa fitur pelanggan telah berjalan sesuai kebutuhan sistem. Pelanggan dapat menambah, mengedit, dan menghapus data Vespa, membuat pemesanan servis, melihat riwayat pemesanan, serta membatalkan pemesanan yang masih aktif. Pada pengujian melalui antarmuka aplikasi, sistem telah mampu melakukan validasi terhadap data yang tidak sesuai, seperti data Vespa kosong, plat nomor dengan karakter tidak valid, panjang plat nomor yang tidak sesuai, model Vespa terlalu pendek, serta tahun produksi di luar rentang yang ditentukan.

Namun, pada pengujian awal melalui API, ditemukan beberapa skenario pengelolaan data Vespa yang belum berhasil. *Backend* masih menerima beberapa *input* yang seharusnya

ditolak, seperti plat nomor dengan karakter khusus, plat nomor terlalu pendek, plat nomor terlalu panjang, tahun produksi di bawah batas, tahun produksi melebihi batas, dan model Vespa terlalu pendek. Setelah dilakukan perbaikan validasi pada sisi *backend*, seluruh skenario tersebut diuji kembali. Hasil pengujian ulang menunjukkan bahwa sistem telah mampu menolak data yang tidak valid sesuai hasil yang diharapkan. Dengan demikian, fitur pelanggan dinyatakan berhasil setelah melalui proses pengujian awal, perbaikan, dan pengujian ulang.

Tabel 4.12 Hasil Pengujian Fitur Mekanik

No	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melihat daftar pekerjaan	*Login* sebagai mekanik	Sistem menampilkan pekerjaan yang ditugaskan	Sistem menampilkan daftar pemesanan yang ditugaskan kepada mekanik yang sedang *login*	Berhasil
2	Tambah suku cadang ke pemesanan	Tambah suku cadang pada pemesanan aktif	Suku cadang berhasil ditambahkan ke pemesanan	Suku cadang berhasil ditambahkan ke pemesanan dan ditampilkan pada rincian pekerjaan	Berhasil
3	Tambah suku cadang melebihi stok	Pilih suku cadang dengan stok 2, lalu tambah jumlah menggunakan tombol +	Sistem membatasi jumlah maksimal sesuai stok tersedia	Tombol + hanya dapat menambah jumlah sampai 2 sehingga jumlah tidak melebihi stok	Berhasil
4	Tambah suku cadang dengan jumlah 0	Isi jumlah suku cadang dengan 0 lalu simpan	Sistem menampilkan validasi bahwa jumlah minimal 1	Sistem menampilkan validasi bahwa jumlah minimal 1 dan data tidak disimpan	Berhasil
5	Hapus suku cadang dari pemesanan	Hapus suku cadang yang sudah ditambahkan	Suku cadang dihapus dari pemesanan	Suku cadang berhasil dihapus dari rincian pemesanan	Berhasil
6	Menyelesaikan pekerjaan	Klik tombol selesaikan pekerjaan dan isi catatan mekanik	Status berubah menjadi selesai dan catatan mekanik tersimpan	Status pemesanan berubah menjadi Selesai dan catatan mekanik berhasil disimpan	Berhasil
7	Menyelesaikan pekerjaan tanpa catatan	Klik selesaikan pekerjaan tanpa mengisi catatan mekanik	Sistem menampilkan validasi bahwa catatan mekanik wajib diisi	Sistem menampilkan validasi bahwa catatan mekanik wajib diisi dan status tidak berubah	Berhasil

Hasil pengujian pada Tabel 4.12 menunjukkan bahwa fitur mekanik telah berjalan sesuai dengan kebutuhan proses pengerjaan servis. Seluruh skenario pada fitur mekanik berhasil pada pengujian awal, sehingga tidak diperlukan perbaikan maupun pengujian ulang. Mekanik dapat melihat daftar pekerjaan yang ditugaskan, menambah dan menghapus suku cadang yang digunakan pada pemesanan, serta menyelesaikan pekerjaan servis dengan mengisi catatan mekanik. Sistem juga berhasil membatasi jumlah suku cadang agar tidak melebihi stok yang tersedia. Selain itu, sistem mampu melakukan validasi terhadap *input* yang tidak sesuai, seperti jumlah suku cadang bernilai 0 dan penyelesaian pekerjaan tanpa catatan. Dengan demikian, fitur mekanik dapat mendukung proses pengerjaan servis secara lebih terstruktur dan terdokumentasi.

Tabel 4.13 Hasil Pengujian Fitur Admin

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melihat *dashboard* admin	*Login* sebagai admin	Statistik *dashboard* admin tampil sesuai data	Sistem menampilkan statistik *dashboard* sesuai data yang tersimpan	Berhasil
2	Konfirmasi pemesanan	Buka detail pemesanan, klik konfirmasi	Status pemesanan berubah menjadi "Dikonfirmasi"	Status pemesanan berubah dari Menunggu menjadi Dikonfirmasi	Berhasil
3	Membatalkan pemesanan	Buka detail pemesanan, klik batalkan	Status pemesanan berubah menjadi "Batal"	Status pemesanan berubah menjadi Batal	Berhasil
4	Menetapkan mekanik	Pilih mekanik pada detail pemesanan	Mekanik berhasil ditetapkan pada pemesanan	Mekanik yang dipilih berhasil ditetapkan pada pemesanan	Berhasil
5	Menetapkan mekanik tanpa memilih mekanik	Klik aksi mulai/penetapan tanpa memilih mekanik	Sistem menampilkan pesan pilih mekanik terlebih dahulu	Sistem menampilkan pesan untuk memilih mekanik dan proses penetapan tidak dilanjutkan	Berhasil
6	Mengubah status pembayaran	Ubah pembayaran menjadi "Lunas"	Status pembayaran berubah menjadi lunas	Status pembayaran berubah menjadi Lunas	Berhasil
7	Menambah layanan	Isi data layanan lalu simpan	Layanan berhasil ditambahkan	Data layanan berhasil disimpan dan ditampilkan pada daftar layanan	Berhasil

8	Menambah layanan dengan harga negatif	Masukkan harga kurang dari 0	Sistem menolak karena harga tidak boleh negatif	Sistem menampilkan pesan bahwa harga tidak boleh kurang dari 0 dan data layanan tidak disimpan	Berhasil
9	Mengedit layanan	Ubah data layanan lalu simpan	Layanan berhasil diperbarui	Perubahan data layanan berhasil disimpan dan ditampilkan	Berhasil
10	Menambah suku cadang	Isi data suku cadang lalu simpan	Suku cadang berhasil ditambahkan	Data suku cadang berhasil disimpan dan ditampilkan pada inventaris	Berhasil
11	Menambah suku cadang dengan harga atau stok negatif	Masukkan nilai negatif pada stok, harga beli, harga jual, atau batas minimal stok	Sistem menampilkan validasi dan data tidak disimpan	Sistem menampilkan pesan bahwa stok, harga beli, harga jual, dan batas minimal stok tidak boleh bernilai negatif sehingga data tidak disimpan	Berhasil
12	*Restock* suku cadang	Isi jumlah *restock* dan harga beli	Stok bertambah dan pengeluaran *restock* tercatat	Stok bertambah sesuai jumlah *restock* dan pengeluaran berhasil tercatat	Berhasil
13	*Restock* suku cadang dengan harga beli negatif	Masukkan harga beli kurang dari 0	Sistem menampilkan validasi dan *restock* tidak disimpan	Sistem menampilkan pesan bahwa harga beli tidak boleh bernilai negatif dan proses *restock* tidak dilanjutkan	Berhasil
14	*Restock* suku cadang dengan jumlah 0	Isi jumlah *restock* dengan 0 lalu simpan	Sistem menampilkan validasi jumlah *restock* harus lebih dari 0	Sistem menampilkan validasi bahwa jumlah harus lebih dari 0 dan *restock* tidak disimpan	Berhasil
15	Menambah karyawan dengan data valid	Isi nama, email, nomor HP, *password*, dan *role* sesuai ketentuan, lalu simpan	Data karyawan berhasil disimpan	Data karyawan berhasil disimpan dan ditampilkan pada halaman manajemen karyawan	Berhasil
16	Menambah karyawan dengan nama terlalu pendek	Masukkan nama satu karakter, lalu simpan	Sistem menampilkan validasi bahwa nama minimal 3 karakter	Sistem menampilkan pesan bahwa nama minimal 3 karakter	Berhasil

				dan data karyawan tidak disimpan	
17	Menambah karyawan dengan nomor telepon tidak valid	Masukkan nomor telepon kurang dari 10 angka, lalu simpan	Sistem menampilkan validasi format nomor telepon	Sistem menampilkan pesan format nomor telepon tidak valid dan data karyawan tidak disimpan	Berhasil
18	Menambah karyawan dengan *password* lemah	Masukkan *password* tanpa kombinasi huruf besar, huruf kecil, dan angka	Sistem menampilkan validasi ketentuan *password*	Sistem menampilkan pesan ketentuan *password* dan data karyawan tidak disimpan	Berhasil
19	Menambah karyawan dengan nomor telepon yang telah digunakan	Masukkan nomor telepon milik akun lain	Sistem menolak karena nomor telepon harus unik	Sistem menolak penyimpanan dan menampilkan pesan bahwa nomor telepon sudah digunakan	Berhasil
20	Menambah karyawan dengan email yang telah digunakan	Masukkan email milik akun lain	Sistem menolak karena email harus unik	Sistem menolak penyimpanan dan menampilkan pesan bahwa email sudah digunakan	Berhasil
21	Mengedit data karyawan	Ubah data karyawan, lalu simpan	Perubahan data karyawan berhasil disimpan	Perubahan data karyawan berhasil disimpan dan ditampilkan pada halaman manajemen karyawan	Berhasil
22	Menetapkan mekanik tanpa memilih mekanik melalui API	Kirim penetapan mekanik melalui Postman dengan id_mekanik kosong	API menolak permintaan karena mekanik wajib dipilih	API menolak permintaan dengan status 422 *Unprocessable Content* lalu menampilkan pesan bahwa mekanik wajib diisi dan data pemesanan tidak berubah	Berhasil

Pada Tabel 4.13, hasil pengujian menunjukkan bahwa fitur admin telah berjalan sesuai dengan kebutuhan operasional bengkel. Seluruh skenario pada fitur admin berhasil pada pengujian awal, sehingga tidak diperlukan perbaikan maupun pengujian ulang. Admin dapat melihat informasi pada *dashboard*, mengelola pemesanan, mengonfirmasi dan membatalkan pemesanan, menetapkan mekanik, mengubah status pembayaran, serta mengelola data layanan, suku cadang, *restock*, dan akun karyawan. Sistem berhasil memvalidasi *input* yang tidak sesuai,

seperti penetapan mekanik tanpa memilih mekanik, harga layanan bernilai negatif, stok dan harga suku cadang bernilai negatif, jumlah atau harga *restock* yang tidak memenuhi ketentuan, serta data karyawan yang tidak valid. Sistem juga mampu mencegah penggunaan email dan nomor telepon yang telah terdaftar. Dengan demikian, fitur admin telah mendukung pengelolaan operasional bengkel secara terstruktur serta menjaga ketepatan dan konsistensi data yang tersimpan.

Tabel 4.14 Hasil Pengujian Fitur Pemilik

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melihat *dashboard* pemilik	*Login* sebagai pemilik	Statistik dan aktivitas terbaru tampil	Sistem menampilkan statistik dan aktivitas terbaru bengkel	Berhasil
2	Melihat laporan keuangan	Buka menu laporan keuangan	Ringkasan keuangan dan grafik tampil	Sistem menampilkan ringkasan dan grafik keuangan	Berhasil
3	Filter laporan keuangan	Pilih rentang tanggal lalu terapkan filter	Data laporan tampil sesuai rentang tanggal	Sistem menampilkan data laporan sesuai rentang tanggal yang dipilih	Berhasil
4	Melihat tab pemasukan	Buka tab pemasukan	Sistem menampilkan daftar transaksi pemasukan	Sistem menampilkan daftar transaksi pemasukan	Berhasil
5	Melihat tab pengeluaran	Buka tab pengeluaran	Sistem menampilkan daftar pengeluaran *restock*	Sistem menampilkan daftar transaksi *restock*	Berhasil
6	Melihat analisis inventaris	Buka menu analisis inventaris	Sistem menampilkan ringkasan analisis inventaris	Sistem menampilkan ringkasan analisis inventaris	Berhasil
7	Melihat audit aktivitas	Buka menu audit aktivitas	Sistem menampilkan daftar aktivitas pengguna pada sistem	Sistem menampilkan daftar aktivitas pengguna yang tercatat	Berhasil

Hasil pengujian pada Tabel 4.14 menunjukkan bahwa fitur pemilik telah berjalan sesuai dengan kebutuhan *monitoring* dan evaluasi bengkel. Pemilik dapat melihat *dashboard*, laporan keuangan, data pemasukan, data pengeluaran *restock*, analisis inventaris, serta audit aktivitas pengguna. Fitur filter laporan keuangan juga berjalan sesuai dengan harapan karena sistem mampu menampilkan data berdasarkan rentang tanggal yang dipilih. Dengan demikian, fitur pemilik dapat membantu proses pemantauan kondisi bengkel dan mendukung pengambilan keputusan berdasarkan data yang tersedia.

Skenario yang memerlukan pengujian ulang hanya ditemukan pada Tabel 4.10 dan Tabel 4.11, yaitu pada validasi registrasi serta pengelolaan data Vespa melalui API. Sementara itu, skenario pada Tabel 4.12, Tabel 4.13, dan Tabel 4.14 berhasil pada pengujian awal, sehingga tidak memerlukan proses perbaikan maupun pengujian ulang.

Berdasarkan keseluruhan hasil pengujian *Black Box Testing*, fitur utama Sistem Informasi Manajemen Layanan Bengkel KRGarage telah berjalan sesuai kebutuhan fungsional. Sistem mampu menjalankan fungsi autentikasi, pembatasan hak akses, pengelolaan profil, perubahan *password*, pengelolaan pemesanan servis, pengelolaan data Vespa, pengerjaan mekanik, pengelolaan persediaan, pembayaran, laporan keuangan, analisis inventaris, audit aktivitas, dan *monitoring* pemilik sesuai dengan peran masing-masing pengguna. Dengan demikian, sistem dinyatakan layak untuk dilanjutkan ke tahap evaluasi penerimaan pengguna melalui *User Acceptance Testing* (UAT).

4.7 Evaluasi Sistem

Setelah tahap *Black Box Testing* selesai dilakukan dan hasil pengujian ulang menunjukkan bahwa fitur utama sistem berjalan sesuai kebutuhan fungsional, tahap selanjutnya adalah melakukan evaluasi menggunakan metode *User Acceptance Testing* (UAT). Evaluasi ini bertujuan untuk mengukur tingkat penerimaan pengguna terhadap sistem yang dikembangkan serta menilai kesesuaiannya dengan kebutuhan operasional bengkel KRGarage. Evaluasi dilakukan dengan melibatkan pengguna yang mewakili setiap aktor dalam sistem, yaitu pemilik, admin, mekanik, dan pelanggan.

4.7.1 Karakteristik Responden

Pengumpulan data dilakukan dengan menyebarkan instrumen kuesioner kepada para pengguna yang telah melakukan uji coba sistem secara langsung. Distribusi dan jumlah responden dalam pengujian UAT ini secara rinci dipaparkan pada Tabel 4.15.

Tabel 4.15 Distribusi Responden Pengujian UAT

Responden	Jumlah Responden	Metode Pengujian
Pemilik	1 orang	Akses penuh melalui perangkat seluler/laptop
Admin	1 orang	Akses panel administrasi melalui komputer/laptop
Mekanik	2 orang	Akses daftar pekerjaan melalui perangkat seluler
Pelanggan	5 orang	Akses pembuatan pemesanan servis melalui perangkat seluler
Total Responden	9 orang	

Responden pada pengujian UAT dipilih menggunakan teknik *purposive sampling*, yaitu pemilihan responden berdasarkan keterlibatan langsung dalam penggunaan sistem. Pemilik,

admin, mekanik, dan pelanggan dipilih karena mewakili seluruh aktor yang berinteraksi dengan fitur-fitur utama pada Sistem Informasi Manajemen Layanan Bengkel KRGarage. Dengan demikian, penilaian yang diberikan diharapkan dapat merepresentasikan tingkat penerimaan pengguna terhadap sistem yang dikembangkan.

4.7.2 Hasil Pengujian Kuesioner UAT

Penilaian instrumen kuesioner menggunakan kriteria Skala Likert lima tingkatan untuk mendapatkan data yang terukur, dengan rincian bobot skor: Sangat Setuju (SS = 5), Setuju (S = 4), Netral (N = 3), Tidak Setuju (TS = 2), dan Sangat Tidak Setuju (STS = 1).

Penyusunan butir pertanyaan kuesioner mengacu pada lima variabel evaluasi utama yang digunakan oleh Aliyah et al. (2025), yaitu fungsionalitas sistem, kinerja sistem, pengalaman dan tampilan antarmuka, efisiensi dan produktivitas, serta keamanan dan keandalan sistem. Namun, pada penelitian ini evaluasi UAT difokuskan pada empat variabel yang dapat dinilai secara langsung oleh pengguna saat melakukan uji coba sistem, yaitu fungsionalitas sistem, kinerja sistem, tampilan antarmuka, serta efisiensi dan produktivitas.

Variabel keamanan dan keandalan sistem tidak dijadikan variabel tersendiri dalam kuesioner UAT karena aspek tersebut telah diuji pada tahap *Black Box Testing*. Pengujian keamanan dilakukan melalui beberapa skenario, seperti proses *login*, validasi perubahan *password*, autentikasi pengguna, dan pembatasan hak akses berdasarkan peran pengguna. Dengan demikian, aspek keamanan tetap diuji dalam penelitian ini, tetapi tidak dihitung sebagai variabel persepsi pengguna pada kuesioner UAT. Berikut merupakan rekapitulasi perolehan skor aktual dari seluruh butir pertanyaan yang dijabarkan berdasarkan variabel pengujiannya.

Tabel 4.16 Rekapitulasi Skor Variabel Fungsionalitas Sistem

No	Aktor	Butir Pertanyaan Kuesioner	SS (5)	S (4)	N (3)	TS (2)	STS (1)	Total
1	Pemilik	Saya dapat melihat data pemesanan servis secara mudah	0	1	0	0	0	4
2	Pemilik	Saya dapat melihat informasi pelanggan dan Vespa secara jelas	0	1	0	0	0	4
3	Pemilik	Saya dapat melihat informasi kinerja operasional bengkel melalui fitur yang tersedia	1	0	0	0	0	5
4	Admin	Saya dapat melihat informasi pelanggan pada data pemesanan dengan mudah	0	1	0	0	0	4
5	Admin	Saya dapat melihat data Vespa pada detail pemesanan servis dengan jelas	1	0	0	0	0	5

6	Admin	Saya dapat mengelola data layanan servis dengan mudah	0	1	0	0	0	4
7	Admin	Saya dapat melihat dan memproses data pemesanan servis dengan jelas	0	1	0	0	0	4
8	Mekanik	Saya dapat melihat daftar pemesanan servis yang ditugaskan kepada saya	2	0	0	0	0	10
9	Mekanik	Saya dapat melihat detail Vespa yang akan diservis dengan jelas	2	0	0	0	0	10
10	Mekanik	Saya dapat melihat keluhan catatan pelanggan dengan mudah	2	0	0	0	0	10
11	Pelanggan	Saya dapat membuat pemesanan servis Vespa dengan mudah	5	0	0	0	0	25
12	Pelanggan	Form pemesanan servis menampilkan pilihan data yang sesuai dengan kebutuhan pelanggan	4	0	1	0	0	23
13	Pelanggan	Saya dapat melihat detail pemesanan servis secara lengkap	4	1	0	0	0	24
	Total	Sub-Total Skor Aktual Fungsionalitas Sistem						132

Rekapitulasi pada Tabel 4.16 menunjukkan bahwa variabel Evaluasi Fungsionalitas Sistem memperoleh nilai sub-total skor aktual sebesar 132. Hasil tersebut menunjukkan bahwa fitur-fitur utama pada sistem telah mampu mendukung kebutuhan operasional pengguna sesuai perannya masing-masing.

Tabel 4.17 Rekapitulasi Skor Variabel Kinerja Sistem

No	**Aktor**	**Butir Pertanyaan Kuesioner**	**SS (5)**	**S (4)**	**N (3)**	**TS (2)**	**STS (1)**	**Total**
1	Pemilik	Saya dapat melakukan login sebagai pemilik bengkel dengan mudah	1	0	0	0	0	5
2	Pemilik	Saya dapat memantau aktivitas layanan servis di bengkel melalui sistem	0	1	0	0	0	4
3	Admin	Saya dapat melakukan login sebagai admin dengan mudah	1	0	0	0	0	5
4	Admin	Saya dapat memperbarui status pemesanan servis dengan mudah	1	0	0	0	0	5
5	Mekanik	Saya dapat melakukan login sebagai mekanik dengan mudah	2	0	0	0	0	10
6	Mekanik	Saya dapat memperbarui progres atau status pengerjaan servis dengan mudah	2	0	0	0	0	10
7	Pelanggan	Saya dapat melakukan login ke sistem KRGarage dengan mudah	5	0	0	0	0	25
	Total	Sub-Total Skor Aktual Kinerja Sistem						64

Hasil pengumpulan data pada Tabel 4.17 menunjukkan bahwa variabel Kinerja Sistem memperoleh nilai sub-total skor aktual sebesar 64 dari tujuh butir pertanyaan yang diuji. Hasil tersebut menunjukkan bahwa sistem mampu mendukung aktivitas utama pengguna dengan

baik, mulai dari proses autentikasi, pemantauan aktivitas layanan, hingga pembaruan status pemesanan atau pengerjaan servis sesuai peran masing-masing pengguna.

Tabel 4.18 Rekapitulasi Skor Variabel Tampilan Antarmuka Sistem

No	Aktor	Butir Pertanyaan Kuesioner	SS (5)	S (4)	N (3)	TS (2)	STS (1)	Total
1	Pemilik	*Dashboard* pemilik menampilkan ringkasan informasi bengkel dengan jelas	1	0	0	0	0	5
2	Pemilik	Tampilan laporan atau rekap data mudah dipahami	1	0	0	0	0	5
3	Admin	*Dashboard* admin menampilkan informasi operasional bengkel dengan jelas	0	1	0	0	0	4
4	Mekanik	*Dashboard* mekanik menampilkan informasi pekerjaan servis dengan jelas	2	0	0	0	0	10
5	Pelanggan	Tampilan *dashboard* pelanggan mudah dipahami	3	1	1	0	0	22
6	Pelanggan	Saya dapat melihat informasi pemesanan servis mendatang dengan jelas	4	1	0	0	0	24
7	Pelanggan	Informasi status pemesanan servis mudah dipahami	4	1	0	0	0	24
8	Pelanggan	Catatan atau informasi tambahan pada detail pemesanan membantu saya memahami proses servis	3	1	1	0	0	22
	Total	Sub-Total Skor Aktual Tampilan Antarmuka Sistem						116

Merujuk pada data yang tersaji pada Tabel 4.18, variabel Evaluasi Tampilan Antarmuka Sistem dianalisis melalui delapan butir indikator pertanyaan dengan perolehan sub-total skor aktual sebesar 116. Hasil tersebut menunjukkan bahwa antarmuka sistem dinilai mudah dipahami dan mendukung pengguna dalam menjalankan aktivitas sesuai perannya masing-masing. Struktur navigasi, tata letak informasi, dan penyajian data dinilai membantu pengguna dalam mengakses informasi yang dibutuhkan secara efektif.

Tabel 4.19 Rekapitulasi Skor Variabel Efisiensi & Produktivitas

No	Aktor	Butir Pertanyaan Kuesioner	SS (5)	S (4)	N (3)	TS (2)	STS (1)	Total
1	Pemilik	Sistem membantu saya memantau status pengerjaan servis	1	0	0	0	0	5
2	Pemilik	Data yang ditampilkan sistem membantu saya dalam mengambil keputusan operasional	0	1	0	0	0	4
3	Pemilik	Secara keseluruhan, fitur pemilik pada KRGarage sudah sesuai dengan kebutuhan pemantauan bengkel	1	0	0	0	0	5
4	Admin	Saya dapat mengatur jadwal servis sesuai kebutuhan operasional bengkel	0	1	0	0	0	4

5	Admin	Informasi yang ditampilkan sistem membantu admin dalam mengelola aktivitas bengkel	1	0	0	0	0	5
6	Admin	Secara keseluruhan, fitur admin pada KRGarage sudah sesuai dengan kebutuhan pengelolaan bengkel	0	1	0	0	0	4
7	Mekanik	Fitur catatan servis membantu saya mencatat hasil pemeriksaan atau pengerjaan Vespa	1	1	0	0	0	9
8	Mekanik	Informasi jadwal servis membantu saya mengatur pekerjaan dengan lebih baik	2	0	0	0	0	10
9	Mekanik	Sistem membantu mekanik dalam memantau pekerjaan servis yang sedang berjalan	2	0	0	0	0	10
10	Mekanik	Secara keseluruhan, fitur mekanik pada KRGarage sudah sesuai dengan kebutuhan pengerjaan servis	2	0	0	0	0	10
11	Pelanggan	Sistem membantu saya memantau riwayat pemesanan servis Vespa	4	1	0	0	0	24
12	Pelanggan	Secara keseluruhan, fitur pelanggan pada KRGarage sudah sesuai dengan kebutuhan saya	3	2	0	0	0	23
	Total	Sub-Total Skor Aktual Efisiensi & Produktivitas						113

Pengolahan data kuesioner pada Tabel 4.19 menunjukkan bahwa variabel Evaluasi Efisiensi dan Produktivitas Sistem menghasilkan nilai sub-total skor aktual sebesar 113. Hasil tersebut menunjukkan bahwa sistem mampu membantu pengguna dalam meningkatkan efisiensi aktivitas operasional. Sistem dinilai dapat mengurangi proses pencatatan manual, membantu pengelolaan jadwal servis, serta mempermudah pemantauan operasional dan pengambilan keputusan oleh pemilik bengkel.

4.7.3 Analisis dan Perhitungan Indeks Persentase

Untuk memperoleh nilai tingkat penerimaan pengguna terhadap sistem yang dikembangkan, dilakukan perhitungan indeks persentase berdasarkan hasil kuesioner *User Acceptance Testing* (UAT). Perhitungan ini mengacu pada metode pengolahan data skala Likert yang digunakan dalam penelitian Aliyah et al. (2025), yaitu dengan menghitung nilai *mean* dari setiap butir pertanyaan, kemudian mengubahnya ke dalam bentuk persentase.

Nilai *mean* diperoleh dari total bobot penilaian pada setiap butir pertanyaan yang dibagi dengan jumlah responden yang menjawab pertanyaan tersebut. Karena instrumen UAT pada penelitian ini disusun berdasarkan peran pengguna, maka jumlah responden pada setiap butir pertanyaan menyesuaikan aktor yang menjawab, yaitu pemilik satu orang, admin satu orang,

mekanik dua orang, dan pelanggan lima orang. Selanjutnya, nilai persentase diperoleh dengan membagi nilai *mean* dengan bobot maksimum skala Likert, yaitu 5, kemudian dikalikan 100%.

Rumus yang digunakan dalam perhitungan adalah sebagai berikut:

$$Mean = \frac{Total\ Bobot\ Penilaian}{Jumlah\ Responden} \quad (4.1)$$

$$Persentase = \left(\frac{Nilai\ Mean}{Bobot\ Maksimum}\right).100\% \quad (4.2)$$

Setelah persentase setiap butir pertanyaan diperoleh, nilai persentase pada masing-masing variabel dihitung dengan mencari rata-rata dari seluruh persentase butir pertanyaan dalam variabel tersebut.

Berdasarkan perhitungan tersebut, diperoleh nilai rata-rata persentase untuk setiap variabel UAT sebagaimana ditunjukkan pada Tabel 4.20.

Tabel 4.20 Rekapitulasi Hasil UAT

Variabel	Persentase
Fungsionalitas Sistem	91,38%
Kinerja Sistem	97,14%
Tampilan Antarmuka Sistem	93,50%
Efisiensi dan Produktivitas	93,17%
Rata-rata Keseluruhan	93,80%

Hasil perhitungan UAT pada Tabel 4.20 menunjukkan bahwa variabel fungsionalitas sistem memperoleh nilai rata-rata sebesar 91,38%, kinerja sistem sebesar 97,14%, tampilan antarmuka 93,50%, serta efisiensi dan produktivitas sebesar 93,17%. Dari keempat variabel tersebut, diperoleh nilai rata-rata keseluruhan sebesar 93,80%. Nilai tersebut menunjukkan bahwa sistem memperoleh tingkat penerimaan yang tinggi dari pengguna.

4.7.4 Interpretasi Persentase Kelayakan Sistem

Untuk memudahkan interpretasi hasil *User Acceptance Testing* (UAT), nilai persentase yang diperoleh dikategorikan berdasarkan rentang interpretasi persentase kelayakan sistem. Kategori tersebut digunakan sebagai dasar dalam menentukan tingkat penerimaan dan kelayakan sistem yang dikembangkan. Adapun kategori interpretasi yang digunakan ditunjukkan pada Tabel 4.21.

Tabel 4.21 Interpretasi Persentase Kelayakan Sistem

Persentase	Kategori
0% - 20%	Sangat Tidak Layak
21% - 40%	Tidak Layak
41% - 60%	Cukup Layak
61% - 80%	Layak
81% - 100%	Sangat Layak

Nilai rata-rata keseluruhan UAT sebesar 93,80% berada pada rentang 81% - 100% sesuai kategori interpretasi pada Tabel 4.21. Dengan demikian, Sistem Informasi Manajemen Layanan Bengkel KRGarage termasuk dalam kategori Sangat Layak. Hasil ini menunjukkan bahwa sistem dapat diterima oleh pengguna dan dinilai sesuai untuk mendukung kebutuhan operasional bengkel.

4.7.5 Kesimpulan Evaluasi Sistem

Berdasarkan hasil perhitungan UAT, Sistem Informasi Manajemen Layanan Bengkel KRGarage memperoleh nilai rata-rata keseluruhan sebesar 93,80% dan termasuk dalam kategori Sangat Layak. Hasil tersebut menunjukkan bahwa sistem dapat diterima oleh pengguna dan dinilai mampu mendukung kebutuhan operasional bengkel, baik dari sisi pemilik, admin, mekanik, maupun pelanggan.

Variabel dengan nilai tertinggi adalah kinerja sistem sebesar 97,14%. Hal ini menunjukkan bahwa pengguna menilai sistem mudah digunakan dalam aktivitas utama, seperti proses masuk ke sistem, pemantauan aktivitas layanan, serta pembaruan status pemesanan atau pengerjaan servis. Sementara itu, variabel fungsionalitas sistem memperoleh nilai paling rendah dibandingkan variabel lainnya, yaitu sebesar 91,38%. Meskipun demikian, nilai tersebut tetap berada pada kategori Sangat Layak. Hal ini menunjukkan bahwa fitur-fitur utama sistem sudah dapat mendukung kebutuhan pengguna, namun aspek kelengkapan dan kesesuaian fungsi tetap dapat menjadi perhatian dalam pengembangan berikutnya.

Dengan demikian, hasil evaluasi menunjukkan bahwa sistem tidak hanya layak secara fungsional, tetapi juga dapat diterima oleh pengguna sebagai solusi untuk mendukung digitalisasi operasional bengkel. Sistem mampu membantu pengelolaan pemesanan servis, penugasan mekanik, pengelolaan inventaris, pencatatan transaksi, serta pemantauan laporan oleh pemilik.

4.8 Penerapan Awal Sistem

Tahap penerapan sistem pada penelitian ini difokuskan pada kesiapan awal sistem untuk digunakan secara bertahap di lingkungan KRGarage. Penerapan belum dilakukan dalam bentuk penggunaan operasional penuh, melainkan sebatas penyiapan sistem dan pengisian data master awal.

Data yang telah disiapkan meliputi data pengguna, data layanan servis, dan data suku cadang. Data pengguna mencakup akun dan *role* pengguna yang digunakan untuk

membedakan hak akses pada sistem. Sementara itu, data layanan servis dan data suku cadang digunakan sebagai data master awal agar fitur utama sistem dapat dijalankan sesuai alur yang telah dirancang.

Dengan adanya penyiapan tersebut, Sistem Informasi Manajemen Layanan Bengkel KRGarage dinyatakan siap diterapkan secara bertahap sebagai dasar digitalisasi pengelolaan layanan servis, data suku cadang, dan hak akses pengguna.

4.9 Pembahasan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi yang telah dilakukan, Sistem Informasi Manajemen Layanan Bengkel KRGarage menunjukkan bahwa sistem yang dikembangkan telah mampu menjawab permasalahan utama yang terjadi pada proses operasional bengkel. Permasalahan yang sebelumnya berkaitan dengan pencatatan manual, pencarian riwayat servis, pengelolaan pemesanan, perhitungan biaya, pengelolaan suku cadang, serta keterbatasan akses laporan bagi pemilik dapat diminimalkan melalui fitur-fitur yang telah diimplementasikan.

Pada sisi pelanggan, sistem menyediakan fitur pemesanan servis *online*, pengelolaan data Vespa, riwayat servis, status pengerjaan, serta pengingat servis berkala pada *dashboard* pelanggan. Fitur tersebut membantu pelanggan memperoleh informasi layanan secara lebih mudah dan transparan tanpa harus bergantung sepenuhnya pada komunikasi manual melalui WhatsApp. Selain itu, penyimpanan data Vespa dan riwayat servis secara digital juga memudahkan pelanggan untuk melihat kembali rekam jejak perawatan Vespa tanpa bergantung pada nota fisik.

Pada sisi admin dan mekanik, sistem membantu proses operasional bengkel menjadi lebih terstruktur. Admin dapat mengelola pemesanan, melakukan konfirmasi, menetapkan mekanik, mengelola layanan, mengelola suku cadang, mencatat *restock*, serta memperbarui status pembayaran. Sementara itu, mekanik dapat melihat daftar pekerjaan, membaca catatan keluhan pelanggan, memperbarui status pengerjaan, dan mencatat penggunaan suku cadang selama proses servis. Dengan adanya alur tersebut, proses kerja yang sebelumnya dilakukan secara manual dapat terdokumentasi dengan lebih baik di dalam sistem.

Pada sisi pemilik, sistem menyediakan *dashboard* pemilik, laporan keuangan, analisis inventaris, dan audit aktivitas. Fitur tersebut membantu pemilik dalam memantau pendapatan, pengeluaran, keuntungan, kondisi stok suku cadang, serta riwayat aktivitas pengguna pada sistem. Dengan demikian, sistem tidak hanya mendukung proses pelayanan pelanggan, tetapi

juga membantu pemilik dalam melakukan pemantauan operasional dan pengambilan keputusan berdasarkan data yang tersedia.

Hasil pengujian *Black Box Testing* menunjukkan bahwa fitur utama sistem telah berjalan sesuai harapan setelah dilakukan pengujian dan perbaikan terhadap validasi *backend*. Pengujian awal menemukan beberapa ketidaksesuaian pada validasi *input* melalui API, khususnya pada proses registrasi dan pengelolaan data Vespa. Setelah dilakukan penyesuaian validasi *backend* dan pengujian ulang, sistem mampu menolak data yang tidak valid sesuai ketentuan. Selanjutnya, hasil *User Acceptance Testing* menunjukkan bahwa sistem memperoleh persentase sebesar 93,80% dengan kategori Sangat Layak. Hasil tersebut menunjukkan bahwa sistem dapat diterima oleh pengguna dan dinilai mampu mendukung kebutuhan operasional bengkel KRGarage.

Tahap penerapan awal juga menunjukkan bahwa sistem telah disiapkan melalui pengisian data master dasar berupa data pengguna, data layanan servis, dan data suku cadang. Data tersebut menjadi dasar agar sistem dapat diterapkan secara bertahap sebelum digunakan secara penuh dalam kegiatan operasional bengkel.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa pengembangan Sistem Informasi Manajemen Layanan Bengkel KRGarage berhasil menghasilkan sistem berbasis *web* yang sesuai dengan kebutuhan pengguna. Sistem yang dibangun mampu mendukung digitalisasi proses pemesanan servis, pencatatan transaksi, pengelolaan suku cadang, pemantauan pengerjaan servis, serta penyajian laporan operasional dan keuangan. Selain itu, hasil pengujian dan evaluasi pengguna menunjukkan bahwa sistem telah memenuhi kebutuhan fungsional serta memperoleh tingkat penerimaan yang sangat baik. Dengan demikian, sistem ini dapat menjadi solusi untuk meningkatkan efisiensi, keteraturan, dan transparansi pengelolaan layanan bengkel KRGarage.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi, pengujian, dan evaluasi pada Sistem Informasi Manajemen Layanan Bengkel KRGarage yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

a. Penelitian ini berhasil membangun sistem informasi berbasis *web* menggunakan *framework* Laravel dan Vue.js dengan metode *Prototyping*. Sistem yang dikembangkan memiliki cakupan fitur yang lebih komprehensif dibandingkan dengan penelitian Silaban et al. (2025). Sistem ini mengintegrasikan pemesanan servis *online*, pengelolaan data Vespa, riwayat servis, manajemen suku cadang, dan pencatatan transaksi secara terpusat untuk mendukung peran pelanggan, admin, mekanik, dan pemilik bengkel.
b. Pengujian *Black Box Testing* menyatakan fitur utama sistem telah berjalan sesuai kebutuhan setelah perbaikan validasi *input*, sedangkan evaluasi *User Acceptance Testing* (UAT) memperoleh persentase kelayakan 93,80% dengan kategori Sangat Layak. Hasil UAT ini menunjukkan tingkat penerimaan yang lebih tinggi dibandingkan penelitian Silaban et al. (2025) dengan skor SUS 82,5, serta dilengkapi modul manajemen inventaris dan laporan keuangan yang melengkapi batasan pada penelitian tersebut.

5.2 Saran

Sistem Informasi Manajemen Layanan Bengkel KRGarage ini telah memenuhi kebutuhan fungsional utama, namun masih memiliki ruang untuk penyempurnaan. Oleh karena itu, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang.

a. Pengembangan sistem selanjutnya disarankan untuk mengintegrasikan layanan *payment gateway* agar pelanggan dapat melakukan pembayaran pemesanan secara langsung melalui sistem dengan berbagai metode.
b. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur *Application Programming Interface* (API) WhatsApp untuk mengirimkan notifikasi otomatis kepada pelanggan terkait status pemesanan, tagihan biaya, maupun pengingat jadwal servis berkala secara langsung ke ponsel pelanggan.

c. Meskipun fitur rekapitulasi pendapatan, pengeluaran *restock*, dan keuntungan operasional saat ini sudah fungsional, untuk mendukung skala bisnis yang lebih besar di masa depan, sistem dapat dilengkapi dengan modul akuntansi yang lebih kompleks, seperti pencatatan buku besar, depresiasi aset bengkel, dan penggajian karyawan.

DAFTAR PUSTAKA

Aliyah, Hartono, N., & Muin, A. A. (2025). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch: Jurnal Sains Dan Teknologi Informasi*, *3*(2), 42–58. https://doi.org/10.62951/switch.v3i1.330

Aripin, A. J., Febrina, Putra, S. S. D., & Hidayat. (2025). *Sistem informasi layanan servis pada bengkel mobil berkah jaya berbasis web*. *5*(2), 102–109. https://doi.org/https://doi.org/10.54840/jcstech.v5i2.344

Arizal, Puteri, A. N., Zakiyabarsi, F., & Priambodo, D. F. (2022). The Prototype Method in the Web-Based Final Project Management Information System. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, *10*(1). https://doi.org/https://doi.org/10.30646/tikomsin.v10i1.606 ISSN

Ginting, Y., & Pandia, H. (2025). Perancangan Sistem Informasi Aplikasi Jaringan Doa Untuk Umat Kristen Berbasis Web Menggunakan Vue.js Dan Laravel. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, *14*(1), 34. https://doi.org/10.35889/jutisi.v14i1.2658

Kurniawan, M. A., Pradana, A. I., & Ambarsari, D. (2022). Pengembangan Sistem Informasi Pelayanan Servis Sepeda Motor Dengan Menggunakan MySQL Dan C # Pada Bengkel Sepeda Motor Tiyas. *Prosiding Seminar Nasional Teknologi Informasi Dan Bisnis*, 122–130.

Liaotando, T. N., Santoso, A. J., & Dewi, F. K. S. (2024). Pembangunan Sistem Informasi CV Sembilan Matahari Sejahtera Berbasis Website. *Jurnal Informatika Atma Jogja*, *5*(1), 19–26. https://doi.org/10.24002/jiaj.v5i1.8687

Maulana, R. A., Ancela, M., & Vernanda, D. (2024). Perancangan Sistem Informasi Booking Service Bengkel Motor Pata Service Motor Berbasis Aplikasi. *Inventor: Jurnal Inovasi Dan Tren Pendidikan Teknologi Informasi*, *2*(2), 54–60. https://doi.org/10.37630/inventor.v2i2.1331

Nurlaila, D., & Mulyono, H. (2023). Sistem Informasi Manajemen Bengkel Berbasis Web Pada Bengkel Ikhsan Jaya Motor. *Jurnal MANAJEMEN SISTEM INFORMASI*, *8*(2), 207–217. http://repository.unama.ac.id/1716/

Rahmadi, L. A., Rapiyanta, P. T., & Pradiatiningtyas, D. (2020). Perancangan Sistem Informasi Pelayanan Surat Pengantar Dukuh Berbasis Website “Dukuhku Online .” *Indonesian*

Journal on Networking and Security, *9*(5), 337–343. https://doi.org/http://dx.doi.org/10.55181/ijns.v9i4.1674

Santoso, D. S., Shintawati, W. P., & Yunanto, A. A. (2024). Sistem Informasi Manajemen Bengkel Den Bagus Dinamo. *REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer*, *8*(3), 793–800. https://www.jurnal.polgan.ac.id/index.php/remik/article/view/13838

Saputri, M. I., Handayani, V. R., Rahmawati, E., & Kesuma, C. (2024). Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Service Pada Bengkel Sido Motor Berbasis Website. *Informatics and Computer Engineering Journal*, *4*(2), 39–47. https://jurnal.bsi.ac.id/index.php/ijec/

Silaban, E. P. M., Nugroho, M. H. D., & Tsani, M. R. (2025). *Pengembangan Website Layanan Bengkel Otomotif Dengan Integrasi Form Konsumen dan Sistem Admin Monitoring*. *5*(2), 269–280.

Sriyeni, Y., & Priatama, M. A. (2024). *Design and Built a Motorcycle Sales and Workshop Information System Using Prototype Method Rancang Bangun Sistem Informasi Penjualan Motor dan Bengkel Menggunakan Metode Prototipe*. *4*(January), 329–339.

Wahyudi, D., & Rahmi, A. N. (2022). Pembuatan Sistem Informasi Bengkel Berbasis Website Pada Bengkel Ferdi Motor. *Information System Journal*, *5*(2), 1–6. https://doi.org/10.24076/infosjournal.2022v5i2.932

Wuladari, C., & Fersellia. (2025). Implementasi Sistem Informasi Manajemen Bengkel Berbasis Web untuk Peningkatan Efisisensi Oprasional (Studi Kasus : Bengkel AA Motor). *INSOLOGI: Jurnal Sains Dan Teknologi*, *4*(3), 551–562. https://doi.org/10.55123/insologi.v4i3.5587

LAMPIRAN