

**RANCANG BANGUN SISTEM INFORMASI JASA PENYEWAAN
BERBASIS WEB PADA KOPERASI KONSUMEN HYDRO UTAMA
BUKITTINGGI**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Program Strata-1 Pada
Departemen Sistem Informasi Fakultas Teknologi Informasi Universitas Andalas

Oleh:

Syadza Intan Benya

2111521011

Pembimbing:

Afriyanti Dwi Kartika, MT

198904212019032024



**DEPARTEMEN SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG**

2025

HALAMAN PENGESAHAN

TUGAS AKHIR

**RANCANG BANGUN SISTEM INFORMASI JASA PENYEWAAN
BERBASIS WEB PADA KOPERASI KONSUMEN HYDRO UTAMA
BUKITTINGGI**

Oleh

Syadza Intan Benya

2111521011

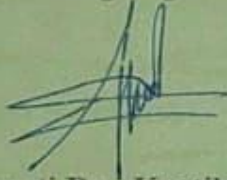
LULUS SIDANG TUGAS AKHIR

04 November 2025

Padang, 5 November 2025

Telah diperiksa dan disetujui oleh

Pembimbing Tugas Akhir



Afriyanti Dwi Kartika, MT

NIP. 198904212019032024

Mengetahui,

Ketua Departemen Sistem Informasi



Ricky Akbar, M.Kom

NIP. 198410062012121001

PERNYATAAN

Saya menyatakan bahwa laporan tugas akhir yang berjudul “Rancang Bangun Sistem Informasi Jasa Penyewaan Berbasis Web Pada Koperasi Konsumen Hydro Utama Bukittinggi” ini tidak terdapat karya yang pernah diajukan sebagai syarat menyelesaikan mata kuliah tugas akhir di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat saya yang pernah ditulis dan diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Padang, 05 November 2025

Penulis,



Syadza Intan Benya

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Rancang Bangun Sistem Informasi Jasa Penyewaan Berbasis Web pada Koperasi Konsumen Hydro Utama Bukittinggi” dengan baik. Penyusunan Tugas Akhir ini merupakan salah satu syarat akademik untuk menyelesaikan Program Strata-I pada Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. Dalam penyusunan laporan ini, penulis mendapat banyak dukungan, arahan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT, atas segala nikmat, petunjuk, serta kekuatan yang diberikan sehingga penulis mampu melalui setiap proses hingga Tugas Akhir ini selesai.
2. Kedua orang tua dan keluarga atas doa, yang senantiasa memberikan doa terbaik, dukungan tanpa batas, serta motivasi yang tiada henti. Setiap nasihat, perhatian, dan pengorbanan yang diberikan menjadi sumber kekuatan dan semangat bagi penulis untuk terus berusaha dan menyelesaikan Tugas Akhir ini hingga tuntas.
3. Bapak Ricky Akbar, M.Kom, selaku Ketua Departemen Sistem Informasi, Fakultas Teknologi Informasi Universitas Andalas, atas arahan dan dukungan akademik sehingga penulis dapat menjalankan dan menyelesaikan Tugas Akhir ini dengan baik.
4. Ibu Afriyanti Dwi Kartika, MT, selaku dosen Pembimbing Tugas Akhir yang telah meluangkan waktu untuk memberikan arahan, bimbingan, serta masukan berharga demi kesempurnaan laporan ini.
5. Ibu Dwi Welly Sukma Nirad, MT, selaku dosen pembimbing akademik, atas bimbingan, perhatian, dan motivasi yang diberikan selama perkuliahan.

6. Seluruh dosen Departemen Sistem Informasi, Fakultas Teknologi Informasi Universitas Andalas, yang telah memberikan ilmu, pengetahuan, serta bimbingan selama masa studi.
7. Ibu Rildawati selaku Manajer beserta pihak Koperasi Konsumen Hydro Utama Bukittinggi yang telah memberikan kesempatan, fasilitas, serta bimbingan selama pelaksanaan Tugas Akhir ini, sehingga penelitian dapat berjalan dengan lancar.
8. Rekan-rekan seperjuangan di Departemen Sistem Informasi yang selalu memberi dukungan, semangat, membantu dalam proses diskusi, maupun penyusunan Tugas Akhir ini.
9. Kepada seluruh pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan bantuan, dukungan, dan motivasi baik secara langsung maupun tidak langsung hingga Tugas Akhir ini dapat terselesaikan.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Apabila terdapat masukan lebih lanjut, dapat disampaikan melalui email di syadzaintan03@gmail.com. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang berkepentingan.

Padang, 05 November 2025

Penulis

Syadza Intan Benya

DAFTAR ISI

HALAMAN PENGESAHAN	ivi
PERNYATAAN	vi
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
ABSTRAK	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian.....	7
1.5 Manfaat Penelitian.....	7
1.6 Sistematika Penulisan.....	8
BAB II TINJAUAN PUSTAKA	9
2.1 Koperasi Konsumen Hydro Utama	9
2.2 Jasa Penyewaan	10
2.3 Sistem Informasi.....	12
2.4 Sistem Informasi Penyewaan	13
2.5 <i>Website</i>	13
2.6 Rancang Bangun Sistem.....	14
2.7 <i>Chatbot</i> dalam Sistem Informasi.....	15
2.8 Metode <i>Waterfall</i>	15
2.8.1 Analisis Kebutuhan (<i>Requirement Definition</i>)	16
2.8.2 Perancangan Sistem dan Perangkat Lunak (<i>System and Software Design</i>)	24
2.8.3 Implementasi dan Pengujian Unit (<i>Implementation and Unit Testing</i>)	26

2.8.4 Integrasi dan Pengujian Sistem (<i>Integration and System Testing</i>)	29
2.8.5 Pemeliharaan (<i>Operation and Maintenance</i>).....	30
2.9 Metode <i>Waterfall</i>	30
BAB III METODOLOGI PENELITIAN	35
3.1 Objek Penelitian	35
3.2 <i>Flowchart</i> Penelitian	35
3.2.1 Identifikasi Masalah dan Pengumpulan Data	37
3.2.2 Analisis Kebutuhan (<i>Requirement Definition</i>)	38
3.2.3 Perancangan Sistem dan Perangkat Lunak (<i>System and Software Design</i>)	38
3.2.4 Implementasi dan Pengujian Unit (<i>Implementation and Unit Testing</i>)	39
3.2.5 Integrasi dan Pengujian Sistem (<i>Integration and System Testing</i>)	39
3.2.6 Laporan	40
BAB IV ANALISIS DAN PERANCANGAN SISTEM	41
4.1 Identifikasi Masalah dan Pengumpulan Data	41
4.1.1 Wawancara.....	41
4.1.2 Analisis Dokumen.....	42
4.2 Analisis Sistem	42
4.2.1 Business Process and Model Nation (BPMN).....	43
4.2.3 Kebutuhan fungsional	53
4.2.4 <i>Use Case Diagram</i>	54
4.2.5 Deskripsi Tugas Aktor	55
4.2.6 <i>Activity Diagram</i>	60
4.2.7 <i>Sequence Diagram</i>	64
4.3 Perancangan Sistem	68
4.3.1 Perancangan Database	68
4.3.2 Struktur Tabel dan Basis Data	71
4.3.3 <i>Class Diagram</i>	74
BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM	79

5.1 Implementasi dan Pengujian Unit.....	79
5.1.1 Pengkodean Program.....	80
5.1.2 Implementasi Antarmuka Sistem.....	89
5.1.3 Pengujian Unit.....	96
5.2 Integrasi dan Pengujian Sistem.....	111
5.2.1 Fokus Pengujian.....	112
5.2.2 Kasus Uji dan Hasil Pengujian.....	115
5.2.3 Kesimpulan Hasil Pengujian.....	129
BAB VI PENUTUP	135
6.1 Kesimpulan.....	135
6.2 Saran.....	136
DAFTAR PUSTAKA.....	137
LAMPIRAN A.....	142
LAMPIRAN B.....	161
LAMPIRAN C.....	185
LAMPIRAN D.....	191
LAMPIRAN E.....	203
LAMPIRAN F.....	256
LAMPIRAN G.....	275
LAMPIRAN H.....	372

DAFTAR GAMBAR

Gambar 2.1 Metode <i>Waterfall</i> (Sommerville, 2020)	16
Gambar 2.2 Simbol <i>Class</i> (Ramdany et al., 2024).....	25
Gambar 2.3 Simbol <i>Association</i> (Ramdany et al., 2024).....	25
Gambar 2.4 Simbol <i>Aggregation</i> (Ramdany et al., 2024)	25
Gambar 2.5 Simbol Composition (Ramdany et al., 2024).....	26
Gambar 2.6 Simbol Generalization (Ramdany et al., 2024)	26
Gambar 2.7 Simbol <i>Dependency</i> (Ramdany et al., 2024).....	26
Gambar 3.1 Flowchart Penelitian.....	36
Gambar 4.1 BPMN Penyewaan Koperasi yang Sedang Berjalan.....	44
Gambar 4.2 BPMN Penyewaan Koperasi yang Diusulkan.....	47
Gambar 4.3 Use Case Diagram.....	55
Gambar 4.4 Activity Diagram lihat layanan dan melakukan pemesanan	61
Gambar 4.5 Activity Diagram Pelanggan Melakukan Pembayaran dan unggah bukti pembayaran	63
Gambar 4.6 Sequence Diagram Pelanggan lihat layanan dan melakukan pemesanan.....	65
Gambar 4.7 Sequence Diagram Pelanggan Melakukan Pembayaran dan Unggah Bukti Pembayaran	67
Gambar 4.8 ERD.....	70
Gambar 4.9 Class Diagram	75
Gambar 4.10 Halaman Antarmuka Pelanggan lihat detail layanan	76
Gambar 4.11 Halaman Antarmuka Pelanggan lihat konfirmasi pemesanan layanan	77
Gambar 5.1 Route Middleware Customer (Pelanggan)	81
Gambar 5.2 API Ketersediaan Pemesanan.....	81
Gambar 5.3 Route Pelanggan Melakukan Pemesanan.....	81
Gambar 5.4 Struktur Model Penyewaan	83
Gambar 5.5 Relasi Model Penyewaan	84
Gambar 5.6 Helper Methods untuk Status Penyewaan	85
Gambar 5.7 Method Create - Form Pemesanan	86

Gambar 5.8 Method Confirm - Validasi dan Konfirmasi	87
Gambar 5.9 Method GetAvailableSlots - API Ketersediaan	88
Gambar 5.10 Implementasi Antarmuka <i>Dashboard</i> Pelanggan	90
Gambar 5.11 Implementasi Antarmuka Detail Layanan.....	91
Gambar 5.12 Implementasi Antarmuka Detail Layanan (1).....	92
Gambar 5.13 Implementasi Antarmuka Detail Layanan (2).....	93
Gambar 5.14 Implementasi Antarmuka Konfirmasi Pemesanan Layanan	94
Gambar 5.15 Implementasi Antarmuka Detail Pembayaran.....	95
Gambar 5.16 Implementasi Antarmuka Riwayat Penyewaan.....	96
Gambar 5.17 Code Pengujian Unit User.....	103
Gambar 5.18 Code Pengujian Unit User (1)	104
Gambar 5.19 Code Pengujian Unit User (2)	105
Gambar 5.20 Code Pengujian Unit User (3)	106
Gambar 5.21 Berhasil Kasus Uji dan Hasil Pengujian Unit User.....	106
Gambar 5.22 Halaman <i>Dashboard</i> Pelanggan.....	117
Gambar 5.23 Tampilan Pelanggan Melakukan Pemesanan Dengan Tipe Harian	119
Gambar 5. 24 Bukti penambahan pelanggan baru untuk tipe harian	119
Gambar 5.25 Tampilan Pelanggan Mencoba Memesan Tanggal yang Sudah Dipesan Sebelumnya (Harian)	120
Gambar 5. 26 Tampilan Pelanggan Mencoba Memesan Tapi Tahapan Pemesanannya Tidak Lengkap	122
Gambar 5.27 Tampilan Pelanggan Melakukan Pemesanan Dengan Tipe Sesi Yang Masih Kosong	123
Gambar 5.28 Bukti penambahan pelanggan baru untuk tipe sesi	123
Gambar 5.29 Tampilan Pelanggan Melakukan Pemesanan Sesi Berbeda Pada Hari yang Sama	125
Gambar 5.30 Tampilan Pelanggan Melakukan Pemesanan Sesi Sama Pada Hari yang Sama.....	126
Gambar 5.31 Tampilan Pelanggan Melakukan Pemesanan dengan Konflik Antar Layanan Lapangan Golf dan lokasi foto pre wedding	127

Gambar 5.32 Tampilan Pelanggan Melakukan Pemesanan Pada Tanggal Libur 129



DAFTAR TABEL

Tabel 2.1 Simbol BPMN (Aprillianie et al., 2023)	18
Tabel 2.2 Simbol <i>Use Case Diagram</i> (Ramdany et al., 2024).....	20
Tabel 2.3 Simbol <i>Activity Diagram</i> (Nurdayati dkk, 2021).....	21
Tabel 2.4 Simbol <i>Sequence Diagram</i> (Fatoni & Zulfikar, 2020)	23
Tabel 2.5 Penelitian Terkait	31
Tabel 4.1 Hasil BPMN Simulation Penyewaan Koperasi.....	50
Tabel 4.2 Deskripsi Tugas Aktor	56
Tabel 4.3 Tabel Layanan.....	71
Tabel 4.4 Tabel Penyewaan	72
Tabel 4.5 Tabel Pembayaran.....	73
Tabel 5.1 Tabel Pengujian Unit User.....	98
Tabel 5. 2 Kesimpulan Hasil Pengujian Unit.....	107
Tabel 5.3 Tabel Fokus Pengujian Aplikasi	112
Tabel 5.4 Pengujian Pemesanan Tipe Harian	118
Tabel 5.5 Pengujian Pemesanan Harian Pada Tanggal yang Sama	119
Tabel 5.6 Pengujian Validasi Data Pemesanan pada Halaman Detail Layanan	121
Tabel 5.7 Pengujian Pemesanan Tipe Sesi.....	122
Tabel 5.8 Pengujian Pemesanan Sesi Berbeda pada Hari Sama	124
Tabel 5.9 Pengujian Pemesanan Sesi Sama pada Hari Sama.....	125
Tabel 5.10 Pengujian Pemesanan Dengan Konflik Antar Layanan.....	127
Tabel 5.11 Pengujian Pemesanan Pada Tanggal Libur	128
Tabel 5.12 Rekapitulasi Hasil Pengujian Aplikasi.....	129

ABSTRAK

Koperasi merupakan badan usaha yang menghimpun anggota untuk mencapai tujuan bersama melalui layanan yang dikelola secara kolektif. Salah satu koperasi yang bergerak di bidang jasa penyewaan adalah Koperasi Konsumen Hydro Utama Bukittinggi, yang menyediakan layanan penyewaan gedung pertemuan, lokasi foto pre-wedding, dan lapangan golf. Proses penyewaan sebelumnya dilakukan dengan datang langsung atau melalui komunikasi WhatsApp, serta pencatatan manual menggunakan nota kertas dan file Excel, sehingga menimbulkan kendala seperti keterlambatan pelayanan, potensi kesalahan pencatatan, dan keterbatasan akses informasi. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi jasa penyewaan berbasis web sebagai solusi terhadap permasalahan tersebut. Metode pengembangan yang digunakan adalah Waterfall, yang meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Sistem dikembangkan menggunakan PHP dan basis data MySQL. Fitur utama yang disediakan meliputi pemesanan layanan secara daring, pengecekan ketersediaan jadwal, unggah bukti pembayaran, pengelolaan data penyewaan, serta modul ulasan dan chatbot sederhana sebagai sarana informasi awal bagi pengguna. Hasil simulasi proses bisnis menggunakan Business Process Simulation (BPSim) menunjukkan peningkatan kinerja alur pelayanan, dengan penurunan rata-rata waktu penyewaan dari 139 menit menjadi 58,6 menit dan penurunan utilisasi Admin dari 63,34% menjadi 10,67%. Pengujian unit membuktikan bahwa seluruh skenario pada modul inti berjalan sesuai rancangan, sedangkan pengujian sistem dengan metode black-box menunjukkan bahwa fungsi utama, khususnya pemesanan layanan dan pengelolaan transaksi, dapat berjalan sesuai kebutuhan dan menghasilkan output yang tepat.

Kata kunci: Sistem Informasi, Koperasi, Penyewaan, Web, Digitalisasi.

BAB I

PENDAHULUAN

Bab ini menjelaskan dasar pelaksanaan penelitian yang meliputi latar belakang masalah, rumusan masalah, batasan penelitian, tujuan, manfaat, serta sistematika penulisan laporan tugas akhir.

1.1 Latar Belakang

Koperasi merupakan badan usaha yang beranggotakan perseorangan atau badan hukum, dan dijalankan berdasarkan prinsip-prinsip koperasi. Koperasi juga menjadi bagian dari gerakan ekonomi rakyat yang berlandaskan asas kekeluargaan (Rasid & Fitria, 2024). Seiring perkembangannya zaman, koperasi menjadi salah satu instrumen penting dalam perekonomian rakyat melalui berbagai unit usaha seperti simpan pinjam, perdagangan, maupun jasa. Namun, perkembangan teknologi informasi yang pesat dalam era digital saat ini menghadirkan tantangan baru. Koperasi dituntut untuk bertransformasi agar tetap relevan dengan kebutuhan zaman, memperluas jangkauan layanan, serta mampu memberikan pelayanan yang lebih cepat, terukur, dan adaptif terhadap kebutuhan anggota maupun pelanggan (Nurdany & Prajasari, 2020).

Salah satu koperasi yang melakukan transformasi adalah Koperasi Konsumen Hydro Utama Bukittinggi, yang mengelola beberapa unit usaha termasuk unit penyewaan dengan layanan yang disediakan lapangan golf, gedung pertemuan, dan lokasi foto *pre wedding*. Namun, kegiatan operasional pada unit ini masih dilakukan secara manual. Proses penyewaan dilakukan dengan cara pelanggan harus datang langsung ke kantor koperasi atau menghubungi PIC (*Person In Charge*) melalui WhatsApp untuk mengetahui ketersediaan layanan. Seluruh transaksi masih dicatat menggunakan file Excel, sedangkan bukti penyewaan diberikan dalam bentuk nota kertas. Selain itu, penyampaian informasi layanan masih mengandalkan media fisik seperti banner dan poster yang jangkauannya terbatas.

Kondisi tersebut menimbulkan berbagai masalah dalam proses penyewaannya. Pertama, pencatatan transaksi yang masih dilakukan menggunakan file Excel dan nota kertas rentan mengalami kesalahan input, duplikasi data, dan kehilangan arsip. Kedua,

pelanggan tidak dapat mengetahui ketersediaan layanan secara langsung karena informasi belum terhubung dalam satu sistem. Ketiga, pengarsipan dokumen yang masih mengandalkan dokumen kertas dan file Excel menyulitkan proses pencarian dan pengecekan data, terutama ketika jumlah transaksi meningkat. Permasalahan tersebut menyebabkan proses pelayanan menjadi lambat, kurang akurat, serta menurunkan tingkat kepuasan pelanggan. Jika kondisi ini dibiarkan, maka akan berdampak pada terhambatnya kinerja operasional dan menurunnya daya saing koperasi di tengah tuntutan digitalisasi. Untuk mengatasi berbagai kendala tersebut, koperasi membutuhkan sistem yang mampu mengintegrasikan seluruh proses penyewaan secara digital agar alur kerja menjadi lebih terorganisasi, transparan, dan mudah dipantau.

Oleh karena itu, perkembangan teknologi informasi menghadirkan solusi melalui penerapan sistem informasi berbasis web. Sistem ini memungkinkan layanan diakses secara lebih luas melalui jaringan internet, tanpa keterbatasan ruang maupun waktu. Melalui sistem berbasis web, pencatatan transaksi dapat dilakukan secara digital, data dapat dikelola secara lebih terstruktur, informasi jadwal tersedia secara langsung di sistem, serta arsip laporan tersimpan dengan lebih aman. Hal ini sejalan dengan pendapat (Siswidiyanto et al., 2020) yang menyatakan bahwa sistem informasi berbasis web merupakan strategi penting dalam mendukung digitalisasi layanan penyewaan dan dapat mengatasi berbagai keterbatasan pencatatan berbasis kertas maupun penggunaan file Excel.

Sejumlah penelitian sebelumnya juga telah membahas penerapan sistem informasi penyewaan berbasis web dengan fokus yang beragam. Menurut Wahyu (2024) mengembangkan sistem informasi penyewaan mobil berbasis web untuk meningkatkan efisiensi pemesanan dan pelaporan transaksi, namun belum dilengkapi dengan fitur interaktif serta sistem evaluasi layanan. Menurut Rahmadoni (2023) membangun sistem pemesanan lapangan futsal berbasis web yang memudahkan pelanggan melakukan reservasi secara daring dan membantu admin mengatur jadwal, namun sistem tersebut belum memiliki fitur umpan balik pelanggan dan integrasi pembayaran non-tunai. Sementara itu, menurut Harzeki (2025) mengembangkan sistem penyewaan lapangan badminton dan tenis berbasis *web* dan *mobile* dengan

integrasi pembayaran digital menggunakan API Midtrans. Walau memiliki fitur yang lebih kompleks, penelitian tersebut difokuskan pada sektor olahraga, bukan pada konteks koperasi yang memiliki karakteristik berbeda dari segi manajemen dan sumber daya.

Berdasarkan perbandingan tersebut, penelitian ini memiliki arah pengembangan baru pada dua hal. Pertama, sistem informasi penyewaan dikembangkan dengan menyesuaikan karakteristik koperasi sebagai lembaga ekonomi rakyat yang menekankan pelayanan anggota dan transparansi pengelolaan. Sistem dirancang berbasis web agar mudah diakses oleh pelanggan maupun pengurus koperasi. Kedua, sistem ini tidak hanya berfungsi untuk penyewaan, tetapi juga dilengkapi dengan fitur interaktif berupa chatbot sederhana yang membantu pelanggan memperoleh informasi layanan secara cepat, serta fitur umpan balik untuk menyampaikan saran atau keluhan secara langsung kepada pengelola. Kehadiran kedua fitur ini, bersama dengan penggabungan tiga jenis layanan penyewaan dalam satu sistem, menjadi pembeda utama penelitian ini dibandingkan penelitian sebelumnya yang umumnya hanya berfokus pada satu jenis layanan dan belum memanfaatkan interaksi digital dengan pengguna.

Saat ini, pencatatan transaksi pada unit penyewaan masih dilakukan menggunakan file Excel sebagai alat bantu utama. Setiap data penyewaan dan laporan masih dikelola secara terpisah sehingga pengurus perlu menelusuri data secara langsung ketika pelanggan ingin mengetahui jadwal atau status penyewaan. Proses komunikasi dengan pelanggan juga masih dilakukan melalui pesan pribadi, seperti WhatsApp, untuk menanyakan ketersediaan layanan atau menyampaikan bukti pembayaran. Cara kerja seperti ini membuat proses administrasi menjadi lebih lambat dan rawan terjadi kesalahan pencatatan. Oleh karena itu, dibutuhkan sistem yang dapat menyatukan seluruh proses penyewaan dalam satu platform digital agar pengelolaan data menjadi lebih mudah dan informasi dapat diakses secara langsung oleh pengurus maupun pelanggan.

Walaupun tingkat penyewaan pada unit usaha koperasi saat ini masih tergolong rendah, penerapan sistem digital tetap menjadi kebutuhan strategis untuk menghadapi

potensi pertumbuhan di masa mendatang. Proses manual yang terus digunakan berisiko menimbulkan kesulitan dalam pengelolaan data, verifikasi ketersediaan fasilitas, serta pelayanan kepada pelanggan apabila jumlah pemesan meningkat. Oleh karena itu, pengembangan sistem informasi ini tidak hanya ditujukan sebagai solusi terhadap permasalahan operasional saat ini, tetapi juga sebagai langkah proaktif membangun infrastruktur digital yang berkelanjutan. Dengan sistem yang terpusat, terstruktur, dan bersifat skalabel, koperasi dapat memperkuat kapasitas pengelolaan layanan dan siap beradaptasi dengan tuntutan perkembangan teknologi.

Pengembangan sistem informasi berbasis web terbukti menjadi solusi yang dapat menjawab persoalan penyewaan, mulai dari pencatatan data yang tidak terstruktur, keterbatasan akses layanan, hingga transparansi transaksi yang rendah. Hal ini memperkuat urgensi penerapan sistem serupa pada Koperasi Konsumen Hydro Utama Bukittinggi, yang hingga kini masih mengandalkan pencatatan pada nota kertas dan file Excel. Dengan adanya sistem berbasis web, pengelolaan data menjadi lebih teratur, pencatatan dan pelaporan dapat dilakukan secara cepat serta terdokumentasi dengan baik. Pelanggan juga memperoleh kemudahan dalam mengakses layanan kapan saja dan di mana saja. Selain itu, sistem juga dapat mendukung pengelola dalam mengatur jadwal maupun data transaksi secara digital dan informasi jadwal tersedia secara langsung di sistem, sehingga mendukung kelancaran operasional koperasi.

Sistem informasi penyewaan berbasis web yang dirancang dalam penelitian ini mendukung proses pemesanan secara online, pengecekan ketersediaan jadwal, serta mencatat setiap transaksi secara digital pelaporan yang terstruktur, hingga penyajian informasi secara langsung di sistem. Sistem juga dilengkapi dengan *chatbot* sebagai media komunikasi interaktif, yang memungkinkan pelanggan memperoleh jawaban cepat terkait layanan maupun jadwal penyewaan, serta memberikan umpan balik untuk meningkatkan kualitas layanan. Dengan demikian, sistem diharapkan mampu menghadirkan pengalaman pengguna yang lebih praktis, responsif.

Dalam tahap pengembangannya, penelitian yang ini menggunakan metode *Waterfall*, yaitu model pengembangan sistem yang bersifat berurutan, mulai dari tahap perencanaan, analisis, perancangan, implementasi, hingga pengujian (Badrul et al.,

2021). Pertimbangan utama pemilihan metode ini adalah karena kebutuhan sistem di koperasi cenderung jelas sejak awal dan tidak sering berubah. Dengan begitu, proses bisa dicatat secara rapi sehingga mudah ditelusuri kembali. Hal ini menjadi penting bagi koperasi yang nantinya membutuhkan pedoman jelas untuk memahami dan merawat sistem.

Menurut Syaputri et al., (2024) membandingkan metode *Waterfall* dengan *agile*, yang menegaskan bahwa metode *Waterfall* memiliki keunggulan berupa struktur yang teratur, dokumentasi yang jelas, serta mudah dikendalikan, sehingga cocok digunakan pada proyek dengan kebutuhan yang jarang berubah. Sebaliknya, meskipun *Agile* lebih fleksibel dalam menghadapi perubahan, penerapannya tidak sederhana karena memerlukan komunikasi intensif dan keterlibatan aktif pengguna. Kondisi ini sulit dilakukan pada koperasi yang memiliki keterbatasan tenaga dan sumber daya.

Sementara itu, menurut Pricillia (2021.) membandingkan beberapa metode, yaitu *Waterfall*, *Prototype*, dan RAD. *Waterfall* dinilai cocok untuk sistem yang sifatnya umum karena menghasilkan dokumentasi yang rapi dan kualitas perangkat lunak yang baik, meskipun prosesnya cenderung kaku dan sulit menyesuaikan perubahan kebutuhan. *Prototype* lebih menekankan keterlibatan aktif pengguna, tetapi berisiko menghasilkan rancangan yang kurang matang karena tahapan analisis dan perancangan berlangsung singkat. Sedangkan RAD mampu mempercepat proses pengembangan, namun biasanya membutuhkan tim besar dan lebih cocok untuk proyek berskala luas.

Berdasarkan pertimbangan, dari berbagai penelitian tersebut, *Waterfall* menjadi pilihan yang paling masuk untuk digunakan sebagai Kebutuhan koperasi yang relatif tetap, skala pengembangan yang tidak terlalu besar, serta keterbatasan sumber daya membuat *Waterfall* lebih sesuai dibandingkan model lainnya. Dengan pendekatan yang berurutan dan terdokumentasi dengan baik, sistem dapat dibangun secara terstruktur dan lebih mudah dikelola ke depannya.

Berdasarkan uraian tersebut, penelitian ini difokuskan pada perancangan dan pembangunan sistem informasi jasa penyewaan berbasis web di Koperasi Konsumen Hydro Utama Bukittinggi. Sistem ini diharapkan untuk mendukung pengelolaan

transaksi secara digital, penyajian informasi secara langsung di sistem, pencatatan serta pelaporan yang lebih terstruktur, dan memberikan akses layanan yang lebih mudah bagi pelanggan. Selain itu, sistem juga diharapkan mampu membantu koperasi dalam mengelola jadwal penyewaan, memperluas jangkauan layanan, dan beradaptasi dengan tuntutan digitalisasi sehingga mampu meningkatkan kualitas layanan serta memperkuat daya saing koperasi di era digital.

Dengan demikian, penelitian ini berfokus pada pengembangan sistem informasi penyewaan berbasis web yang dilengkapi fitur chatbot dan umpan balik pelanggan untuk mendukung pelayanan yang lebih responsif, transparan, dan mudah diakses pada Koperasi Konsumen Hydro Utama Bukittinggi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah yang akan dibahas dalam penelitian ini adalah, sebagai berikut:

1. Bagaimana merancang sistem informasi jasa penyewaan berbasis web untuk menggantikan proses tatap muka atau WhatsApp sehingga pelanggan lebih mudah mengakses layanan?
2. Bagaimana sistem dapat mendukung pencatatan data layanan, jadwal, dan transaksi penyewaan secara digital sehingga data lebih terstruktur, terintegrasi, dan mudah ditelusuri?
3. Bagaimana sistem dapat menyediakan fitur penyajian informasi layanan serta sarana komunikasi interaktif antara pelanggan dan koperasi untuk menunjang proses penyewaan secara online?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah disusun, agar penelitian ini lebih terarah dan tidak meluas, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Layanan yang dikelola terbatas pada penyewaan gedung pertemuan, lokasi foto *pre wedding*, dan lapangan golf.
2. Sistem informasi dikembangkan berbasis web menggunakan *framework* Laravel dengan bahasa pemrograman PHP dan basis data MySQL

3. Sistem hanya mencakup tiga jenis pengguna, yaitu pelanggan, admin, dan manajer.
4. Sistem informasi yang dikembangkan meliputi fitur registrasi, *login*, pemesanan layanan, pengecekan jadwal, pencatatan transaksi penyewaan secara digital pada basis data, penyajian *dashboard* laporan, serta fitur tambahan berupa chatbot untuk informasi layanan dan umpan balik pelanggan.
5. Penelitian dilakukan hingga tahap perancangan, implementasi, dan pengujian dengan metode *Waterfall*, tanpa mencakup tahap pemeliharaan sistem.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem informasi berbasis web pada Koperasi Konsumen Hydro Utama Bukittinggi. Berikut tujuan penelitiannya:

1. Mempermudah pelanggan dalam memperoleh informasi layanan, melakukan pemesanan, memeriksa ketersediaan jadwal, serta melakukan pembayaran secara digital.
2. Memberikan fasilitas bagi admin dalam mengelola data layanan dan melakukan verifikasi transaksi penyewaan.
3. Menyediakan sarana bagi manajer untuk memantau laporan penyewaan dan aktivitas operasional secara digital.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini dirancang untuk memberikan dampak positif bagi berbagai pihak yang terlibat, baik koperasi, pelanggan, maupun peneliti. Sistem informasi yang dikembangkan diharapkan dapat memperbaiki proses layanan, mempermudah akses informasi, dan mendukung kemajuan teknologi di koperasi. Berikut adalah manfaat yang diharapkan:

1. Membantu koperasi dalam menyederhanakan proses penyewaan, mengelola data secara digital, serta memperluas jangkauan layanan tanpa batasan waktu dan lokasi.
2. Memberikan kemudahan bagi pelanggan dalam mengakses informasi, melakukan pemesanan, dan mengecek jadwal penyewaan secara online.

3. Menambah pengalaman peneliti dalam merancang sistem informasi berbasis web serta menjadi referensi untuk penelitian selanjutnya di bidang teknologi koperasi dan jasa penyewaan.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terbagi menjadi 6 bab, sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi tentang pembahasan latar belakang masalah yang dimana menjadi dasar penelitian, perumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian. serta sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Membahas literatur dan teori-teori yang relevan dengan penelitian, meliputi konsep sistem informasi, koperasi, penyewaan berbasis web, serta teori pendukung lainnya.

BAB III: METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, tahapan penelitian, teknik pengumpulan data, serta metode analisis. Pada bagian ini juga dijelaskan tahapan pengembangan sistem mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian.

BAB IV: ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas analisis kebutuhan sistem, baik fungsional maupun non-fungsional, serta perancangan sistem yang mencakup desain arsitektur, alur kerja, basis data, dan antarmuka pengguna.

BAB V: IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini, menjelaskan tentang proses implementasi sistem informasi berbasis web yang telah dirancang, serta hasil pengujian untuk memastikan sistem berfungsi sesuai tujuan. Hasil pengujian dianalisis guna menilai keberhasilan implementasi.

BAB VI: PENUTUP

Bab ini berisi tentang kesimpulan hasil penelitian dan saran untuk pengembangan lebih lanjut. Selain itu, disampaikan refleksi penulis terhadap hasil implementasi dan kontribusi penelitian terhadap koperasi.

BAB II

TINJAUAN PUSTAKA

Bab ini membahas landasan teori yang mendukung penelitian, meliputi konsep, prinsip, dan hasil penelitian terdahulu yang relevan sebagai dasar dalam perancangan dan pelaksanaan penelitian.

2.1 Koperasi Konsumen Hydro Utama

Koperasi berasal dari istilah latin *cooperation*, yang dalam bahasa inggris disebut dengan *cooperation* memiliki arti kerja sama. Istilah ini terdiri dari dua kata, yaitu *co* yang berarti bersama dan *operation* yang berarti bekerja atau berusaha. Dengan demikian, koperasi dapat diartikan sebagai organisasi di mana para anggotanya bekerja sama untuk mencapai tujuan tertentu. Selain itu, koperasi juga dapat didefinisikan sebagai kelompok orang yang berfokus pada pemenuhan kebutuhan ekonomi anggotanya melalui pelayanan bersama (Maulana et al., 2020).

Menurut Pasal 6 Undang-Undang Republik Indonesia Nomor 17 Tahun 2012 tentang Perkoperasian, koperasi melaksanakan prinsip-prinsip sebagai berikut:

1. Keanggotaan Koperasi bersifat sukarela dan terbuka
2. Pengawasan oleh Anggota diselenggarakan secara demokratis
3. Anggota berpartisipasi aktif dalam kegiatan ekonomi Koperasi
4. Koperasi merupakan badan usaha swadaya yang otonom, dan independen
5. Koperasi menyelenggarakan pendidikan dan pelatihan bagi Anggota, Pengawas, Pengurus, dan karyawannya, serta memberikan informasi kepada masyarakat tentang jati diri, kegiatan, dan kemanfaatan Koperasi
6. Koperasi melayani anggotanya secara prima dan memperkuat Gerakan Koperasi, dengan bekerja sama melalui jaringan kegiatan pada tingkat lokal, nasional, regional, dan internasional; dan
7. Koperasi bekerja untuk pembangunan berkelanjutan bagi lingkungan dan masyarakatnya melalui kebijakan yang disepakati oleh Anggota.

Koperasi Konsumen Hydro Utama Bukittinggi awalnya berdiri pada tahun 1984 dengan nama awal Koperasi Pegawai Negeri PLN Sektor Bukittinggi, kemudian pada tahun 2022 berganti nama menjadi Koperasi Konsumen Hydro Utama

Bukittinggi. Dua tahun kemudian, pada 2024 manajemen UBP Bukittinggi resmi menyerahkan pengelolaan aset dan properti seperti lapangan golf, gedung pertemuan, dan lokasi foto *pre wedding* kepada koperasi. Penyerahan ini menandai lahirnya unit usaha penyewaan sebagai salah satu fokus pengembangan koperasi, yang menambah daftar unit koperasi selain unit pembiayaan dan pinjaman, unit toko, serta unit pengadaan barang/jasa.

Saat ini, koperasi memiliki 125 anggota, 121 terdiri dari pegawai aktif dan 4 anggota dari pensiunan PT PLN Indonesia Power Unit Pelaksana Pengendalian Pembangkit Bukittinggi. Koperasi memiliki visi dan misi sebagai berikut:

A. Visi

“Menjadi mitra kerja yang handal dalam menyediakan kebutuhan pelanggan”.

B. Misi

1. Menyelenggarakan pelayanan yang profesional kepada mitra kerja
2. Mengembangkan usaha-usaha untuk meningkatkan kesejahteraan anggota koperasi

Selain mengelola unit usaha simpan pinjam, toko, serta pengadaan barang dan jasa, koperasi juga memiliki unit penyewaan yang saat ini menjadi fokus pengembangan. Unit penyewaan tersebut meliputi:

1. Penyewaan gedung pertemuan yang berlokasi di Jalan Simpang Tiga, Padang Lua, Banuhampu, Agam.
2. Penyewaan lapangan Golf yang berada di Kecamatan Akabiluru, Kabupaten Lima Puluh Kota, Sumatera Barat.
3. Penyewaan lokasi foto *pre wedding* di sekitar kawasan lapangan golf.

2.2 Jasa Penyewaan

Jasa adalah aktivitas atau tindakan yang dilakukan individu maupun organisasi untuk memberikan manfaat bagi pelanggan. Walaupun dalam praktiknya sering melibatkan hal-hal berwujud, hakikat jasa tetap bersifat tidak berwujud (Wijaya, 2022). Penyewaan adalah sebuah kesepakatan dimana seseorang membayar untuk menggunakan barang atau properti milik orang lain dalam jangka waktu tertentu. Barang yang disewakan dapat beragam jenisnya dengan tarif dan durasi penyewaan

yang bervariasi sesuai kebutuhan (Luturlean, 2019). Penyewaan yang kata dasarnya berasal dari sewa, sehingga sewa-menyewa adalah perjanjian antara dua pihak yang saling bersepakat dan memiliki hak serta kewajiban masing-masing. Salah satu pihak memiliki hak untuk menerima pembayaran dan berkewajiban memberikan hak penggunaan atau manfaat atas suatu barang. Sementara pihak lainnya berhak menggunakan atau menikmati barang tersebut dan berkewajiban membayar sesuai kesepakatan (Sumriyah & Djulaeka, 2023). Beberapa istilah penting dalam konteks penyewaan adalah sebagai berikut:

1. Sewa, proses ketika penyewa memperoleh hak penggunaan barang atau jasa dalam periode tertentu dengan kewajiban membayar sesuai perjanjian.
2. Penyewaan, aktivitas atau layanan yang disediakan pengelola untuk memungkinkan pihak lain menggunakan barang atau jasa yang tersedia.
3. Pemesanan, tindakan formal yang dilakukan pelanggan untuk memastikan ketersediaan layanan atau fasilitas pada waktu tertentu, biasanya melibatkan konfirmasi maupun pembayaran.

Pada Koperasi Konsumen Hydro Utama Bukittinggi, kegiatan usaha dijalankan dalam bentuk jasa penyewaan, yang mencakup penyediaan layanan gedung pertemuan, lokasi foto *pre wedding*, serta lapangan golf. Dalam hal ini, koperasi berperan sebagai pengelola layanan penyewaan sekaligus penyedia pemesanan, meskipun tidak menjadi pemilik langsung dari seluruh aset tersebut. Saat ini, proses penyewaan masih dilakukan dengan calon penyewa mendatangi kantor koperasi atau berkomunikasi melalui WhatsApp untuk melakukan pemesanan, pengecekan jadwal, dan konfirmasi pembayaran. Transaksi pembayaran umumnya dilakukan secara tunai maupun melalui transfer bank, dengan bukti pembayaran yang dikirimkan kepada pengelola.

Pencatatan transaksi, jadwal, dan data penyewaan masih mengandalkan file Excel serta nota kertas. Cara ini berpotensi menimbulkan kesalahan pencatatan, duplikasi data, dan menyulitkan proses penelusuran ketika volume data semakin meningkat. Adapun kondisi penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi saat ini adalah sebagai berikut:

1. Jenis jasa penyewaan meliputi gedung pertemuan, lapangan golf, serta tempat lokasi foto *pre wedding*.
2. Proses pemesanan dan pengecekan jadwal masih dilakukan secara tatap muka atau melalui WhatsApp.
3. Pencatatan transaksi dan data penyewaan masih dilakukan menggunakan file Excel dan nota kertas, sehingga berisiko tidak akurat serta menyulitkan pengelolaan data di kemudian hari.

2.3 Sistem Informasi

Sistem informasi adalah sistem yang terdiri dari berbagai elemen yang saling berinteraksi untuk mencapai tujuan yang sudah ditetapkan. Hubungan antar elemen inilah yang membentuk satu kesatuan sehingga sistem dapat berjalan sesuai fungsinya (Asyari, 2021). Sistem informasi bertujuan untuk mengolah dan menyampaikan informasi yang jelas dan relevan kepada penerima informasi, agar tujuan sistem tersebut dapat tercapai dengan baik. Selain itu, sistem ini juga melibatkan pengumpulan, pengolahan, dan distribusi informasi yang mendukung pengambilan keputusan dan kegiatan operasional dalam organisasi (Sallaby & Kanedi, 2020). komponen-komponen utama dalam sistem informasi adalah sebagai berikut (Gede, 2022):

1. Perangkat Keras (*hardware*) merupakan perangkat fisik seperti komputer, server, dan perangkat lainnya yang digunakan untuk menyimpan dan memproses data.
2. Perangkat lunak (*software*) merupakan program dan aplikasi yang mengelola data serta mendukung operasional sistem informasi.
3. Data adalah informasi yang disimpan dan diolah dalam sistem, biasanya dalam bentuk basis data yang terstruktur
4. Brainware merupakan pengguna dan sumber daya manusia yang mengoperasikan, mengelola, dan memanfaatkan sistem informasi.
5. Prosedur adalah aturan serta proses yang harus diikuti dalam pengoperasian sistem untuk memastikan sistem berjalan sesuai dengan tujuan yang telah ditetapkan.

2.4 Sistem Informasi Penyewaan

Sistem Informasi Penyewaan adalah aplikasi berbasis teknologi yang dirancang untuk mempermudah proses penyewaan, seperti mencari, memesan, dan mengelola data penyewaan properti. Sistem ini menggantikan cara lama yang kurang optimal, seperti pemasangan iklan menggunakan selebaran kertas atau media cetak lainnya. Tujuan utama dari sistem ini adalah untuk mempermudah calon penyewa dalam mencari fasilitas yang sesuai dengan kebutuhan mereka, serta menghemat waktu dan tenaga (Siswidiyanto et al., 2020).

Sistem informasi penyewaan ini juga memudahkan proses transaksi melalui pemesanan online, akses jadwal secara langsung, dan pencatatan data yang lebih terorganisir. Dengan demikian, sistem ini mendukung kepuasan pelanggan, kelancaran operasional, serta membantu pengelola meningkatkan pendapatan dan kualitas layanan (Sonia & Cahyaningtyas, 2023). Dengan demikian, sistem informasi penyewaan berfungsi tidak hanya untuk mendigitalisasi seluruh proses operasional, tetapi juga sebagai sarana penghubung antara kebutuhan pelanggan dengan pihak pengelola. Hal ini memastikan bahwa interaksi dan koordinasi antara kedua pihak dapat berjalan dengan lebih terorganisir.

2.5 Website

Website merupakan sarana yang digunakan untuk menyajikan berbagai jenis informasi, seperti teks, audio, gambar, dan animasi, yang dapat diakses melalui jaringan internet menggunakan perangkat komputer. *Website* juga berfungsi sebagai media informasi berbasis jaringan yang dapat diakses kapan pun dan di mana pun (Firmansyah et al., 2021).

Web atau yang lebih dikenal dengan sebutan *World Wide Web* (WWW) adalah layanan yang bisa diakses oleh pengguna komputer yang terhubung ke internet (Noviana, 2022). Web page adalah dokumen HTML yang diakses lewat HTTP, halaman web ini bersifat statis atau dinamis dan saling terhubung dalam jaringan halaman (Noviantoro et al., 2022).

Perkembangan teknologi web dapat dibagi menjadi beberapa generasi, masing-masing dengan karakteristik dan kemampuan yang semakin maju (Sebastian et al., 2023):

1. Web 1.0 versi generasi awal yang hanya menampilkan informasi (read-only). Pengguna hanya dapat melihat konten tanpa memberikan kontribusi.
2. Web 2.0 versi yang memperkenalkan konsep read-write, memungkinkan pengguna memberikan umpan balik, berinteraksi, dan berkontribusi menambahkan konten pada platform.
3. Web 3.0 versi yang dikenal sebagai semantic web, yang berfokus pada kemampuan sistem yang mengenali dan memahami konten secara otomatis untuk mendukung proses yang lebih terstruktur.
4. Web 4.0 versi tahap lanjutan yang menghadirkan interaksi cerdas dengan kemampuan *read-write-execution-concurrency*, mendukung otomasi dan komunikasi yang lebih kompleks.

2.6 Rancang Bangun Sistem

Rancang bangun sistem adalah proses untuk merancang dan mengembangkan sebuah sistem atau aplikasi yang bertujuan untuk menyelesaikan masalah tertentu. Proses ini melibatkan berbagai langkah seperti memahami kebutuhan pengguna, merancang cara kerja sistem, membangun sistem menggunakan teknologi tertentu, dan memastikan sistem berjalan dengan baik melalui pengujian (Rahmatuloh & Revanda, 2022). Proses perancangan sistem dapat dibagi menjadi beberapa tahapan yang saling berkaitan, sebagai berikut (Azis, 2022) :

1. Pendefinisian kebutuhan fungsional serta persiapan untuk proses perancangan sistem.
2. Penyusunan gambaran awal sistem, mencakup pembuatan rancangan, perencanaan, serta sketsa awal yang menjadi acuan pengembangan.
3. Perancangan detail, meliputi analisis interaksi, pembuatan struktur basis data, dan perancangan antarmuka pengguna.
4. Implementasi serta pengujian, yaitu tahap menerapkan desain yang telah dibuat sekaligus memastikan sistem berjalan sesuai dengan yang direncanakan.

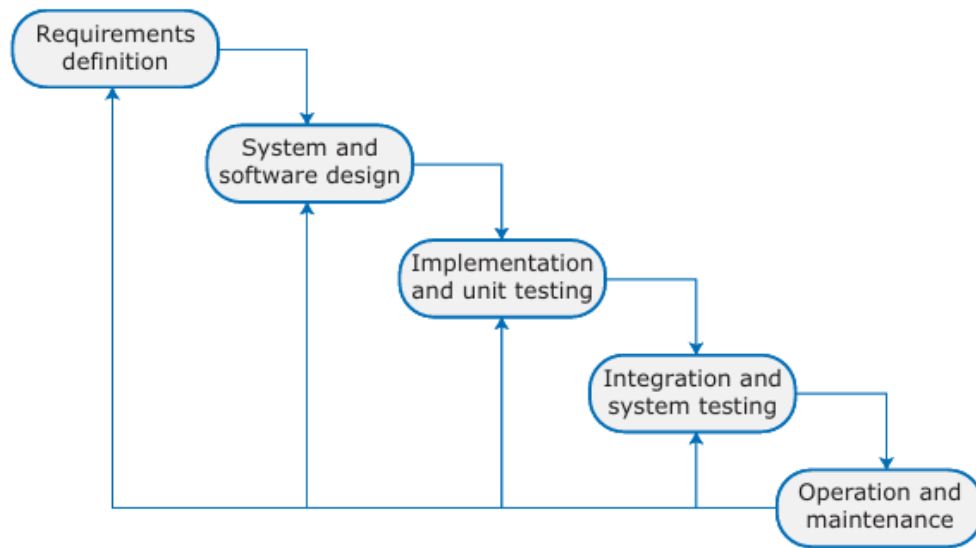
2.7 Chatbot dalam Sistem Informasi

Chatbot adalah sebuah program komputer yang dibuat untuk menirukan percakapan atau komunikasi interaktif dengan pengguna. Interaksi antara manusia dan komputer ini pada dasarnya merupakan hasil respon dari sistem yang sudah diprogram sebelumnya. Respon tersebut biasanya muncul dari proses pencarian kata kunci pada input pengguna, lalu sistem akan memberikan jawaban yang dianggap paling sesuai atau paling mendekati berdasarkan pola yang ada di dalam databasenya (Sugiyanto, 2020).

Chatbot berbasis aturan (*rule based*) bekerja dengan mencocokkan input dari pengguna dengan pola atau kata kunci yang telah ditetapkan sebelumnya. Ketika pengguna memberikan pertanyaan atau perintah, sistem akan mencari kesesuaian input tersebut dengan pola yang ada tersimpan di dalam basis data. Jika ada kecocokan, *chatbot* memberikan tanggapan yang sesuai dengan aturan yang sudah disusun. Namun, jika tidak ditemukan kecocokan, *chatbot* akan mengirimkan tanggapan bawaan atau memberitahukan bahwa ia tidak dapat memahami permintaan tersebut (Ananta et al., 2025).

2.8 Metode *Waterfall*

Menurut Badrul et al., (2021), metode *Waterfall* atau Air Terjun adalah suatu pendekatan pengembangan perangkat lunak yang dilakukan dengan cara berurutan, seperti aliran air terjun. Proses ini dimulai dengan tahap perencanaan, kemudian dilanjutkan dengan pemodelan, implementasi (pembangunan sistem), dan diakhiri pengujian. Model ini sesuai digunakan untuk proyek yang memiliki kebutuhan yang jelas dan stabil sejak awal, di mana setiap tahap dapat berjalan secara berurutan tanpa banyak perubahan besar (Pressman & Maxim, 2019).



Gambar 2.1 Metode *Waterfall* (Sommerville, 2020)

Tahapan dalam metode ini bisa dilihat pada Gambar 2.1 Metode *Waterfall* diatas yang memiliki beberapa tahapan akan dijelaskan lebih lanjut, sebagai berikut:

2.8.1 Analisis Kebutuhan (*Requirement Definition*)

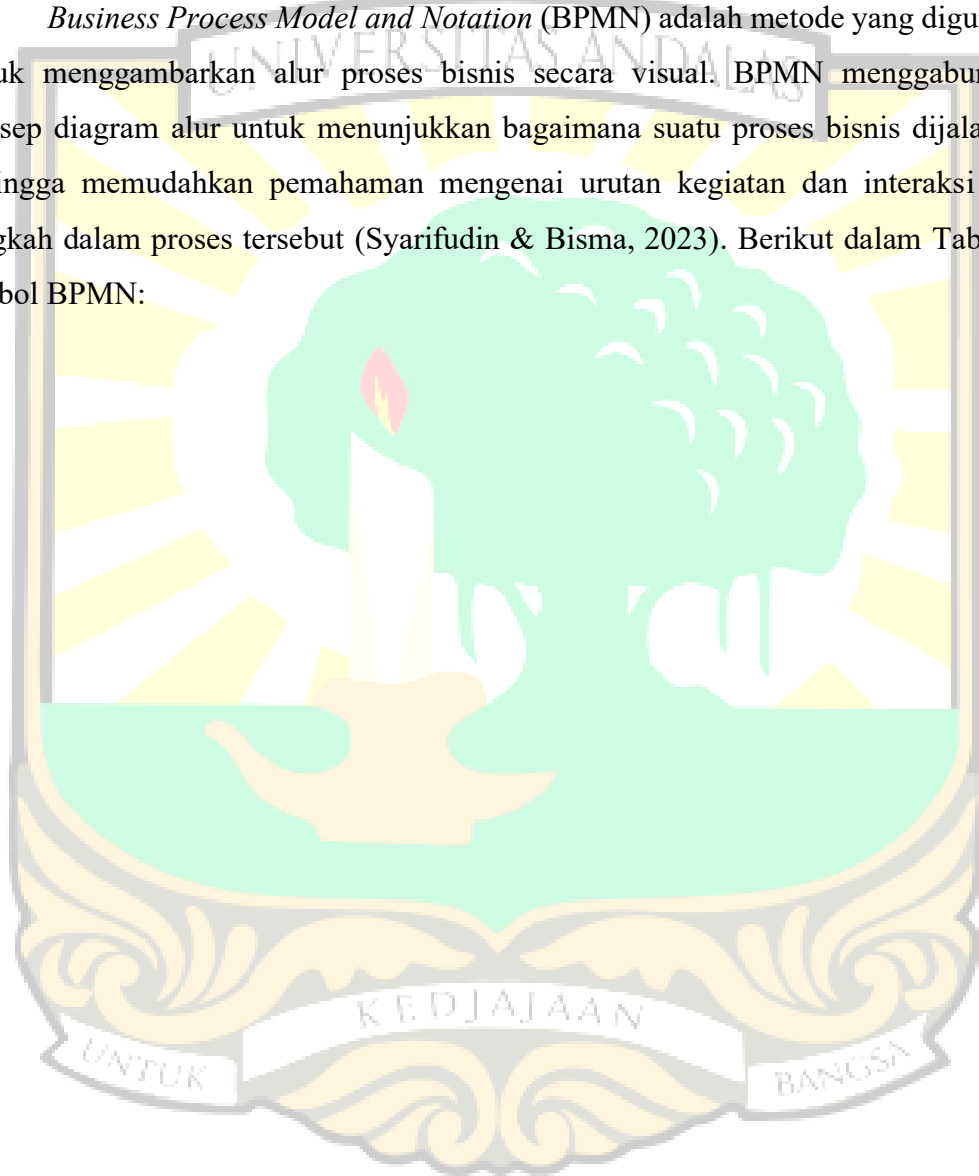
Pada tahap ini, tujuan utama adalah untuk memahami kebutuhan pengguna secara mendalam dan mendokumentasikan secara sistematis. Proses ini melibatkan dua langkah utama, yaitu pengumpulan kebutuhan dan analisis kebutuhan. Pengumpulan kebutuhan dilakukan melalui metode seperti wawancara, survei, atau observasi langsung dari pengguna untuk mendapatkan data yang relevan. Setelah itu, analisis kebutuhan dilakukan untuk memastikan bahwa semua kebutuhan tercakup secara lengkap dan tanpa adanya konflik, serta mendefinisikan spesifikasi kebutuhan dengan jelas (Mall, 2014).

Setelah kebutuhan dikumpulkan dan dianalisis, langkah berikutnya adalah menggambarkan alur proses bisnis akan dianalisis dan digambarkan menggunakan *Business Process Model and Notation* (BPMN), yang digunakan untuk memvisualisasikan langkah-langkah dan aktivitas dalam sistem. BPMN membantu menggambarkan interaksi antar aktivitas bisnis, serta hubungan antar langkah-langkah dalam sistem yang akan dibangun. Kebutuhan yang telah teridentifikasi kemudian dijelaskan lebih lanjut menggunakan diagram seperti *use case diagram*, *activity*

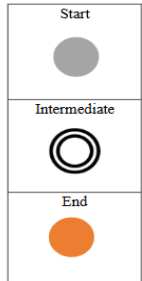
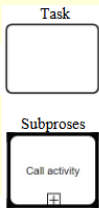
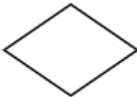
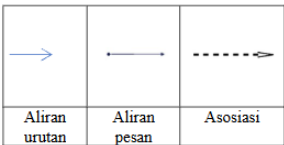
diagram, dan *sequence diagram* yang akan digunakan untuk menjelaskan bagaimana sistem akan berfungsi dan berinteraksi dengan pengguna.

2.8.1.1 Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN) adalah metode yang digunakan untuk menggambarkan alur proses bisnis secara visual. BPMN menggabungkan konsep diagram alur untuk menunjukkan bagaimana suatu proses bisnis dijalankan, sehingga memudahkan pemahaman mengenai urutan kegiatan dan interaksi antar langkah dalam proses tersebut (Syarifudin & Bisma, 2023). Berikut dalam Tabel 2.1 simbol BPMN:



Tabel 2.1 Simbol BPMN (Aprillianie & Muarie, 2023)

Simbol	Nama	Keterangan
	<i>Event</i>	Menggambarkan awal (<i>start</i>), perantara (<i>intermediate</i>), dan akhir (<i>end</i>).
	<i>Task/Activity</i>	Menunjukkan kegiatan yang harus diselesaikan. Subproses digunakan untuk menggambarkan detail.
	<i>Gateway</i>	Menunjukkan percabangan atau penggabungan jalur dalam alur proses.
	Menghubungkan an Objek	Terdiri dari aliran proses, aliran pesan, dan asosiasi. Aliran proses menghubungkan objek, aliran pesan menggambarkan komunikasi, dan asosiasi menunjukkan hubungan antar elemen.
	<i>Pool</i>	Untuk mempresentasikan aliran pesan antar proses atau alur kerja secara keseluruhan.

Simbol	Nama	Keterangan
	Jalur (<i>lane</i>)	Digunakan untuk mengelompokkan aktivitas dalam suatu proses sesuai fungsi atau peran tertentu.

BPMN *Simulation* merupakan pendekatan untuk memodelkan dan menguji alur proses bisnis di berbagai kondisi. Simulasi dilakukan dengan menetapkan parameter dan skenario tertentu sehingga dapat dianalisis bagaimana proses berjalan, hambatan yang muncul, serta potensi perbaikannya (Sparx Systems, 2022), berikut langkah-langkah dalam simulasi BPMN antara lain:

1. Analisis Proses Bisnis, membuat model BPMN untuk mengevaluasi durasi proses, mengidentifikasi hambatan, dan mengoptimalkan alur kerja.
2. Pengujian Skenario Berbeda, melalui konfigurasi BPSim, skenario alternatif dapat diuji untuk melihat dampak perubahan sumber daya atau kebijakan.
3. Penilaian Kinerja Proses, berdasarkan data simulasi, seperti *degree of utilisation* dan *average time*, sehingga kinerja dapat diukur dari penggunaan sumber daya dan durasi penyelesaian aktivitas.

Beberapa parameter utama yang digunakan untuk mengevaluasi kinerja dalam simulasi BPMN antara lain:

- 1) *Degree of Utilisation* adalah ukuran seberapa sering atau seberapa besar suatu sumber daya (seperti orang, mesin, atau sistem) digunakan dalam proses bisnis.
- 2) *Average Time* merupakan waktu rata-rata yang dibutuhkan untuk menyelesaikan suatu aktivitas atau proses, termasuk waktu tunggu dalam antrian.
- 3) *Mean of Processing Time* merupakan waktu rata-rata yang hanya digunakan untuk menjalankan suatu aktivitas tanpa menghitung waktu tunggu.
- 4) *Maximum Time* merupakan waktu terlama yang tercatat untuk menyelesaikan satu siklus proses selama simulasi berlangsung.
- 5) *Max of Processing Time* merupakan waktu pemrosesan terlama yang tercatat pada suatu aktivitas tertentu dalam periode simulasi.

- 6) *Minimum Time* merupakan waktu tercepat yang dibutuhkan untuk menyelesaikan satu siklus proses dari awal hingga akhir.
- 7) *Min of Processing Time* merupakan waktu pemrosesan paling singkat yang tercatat pada suatu aktivitas selama simulasi.
- 8) *Total Time Busy* merupakan total waktu suatu sumber daya atau aktivitas benar-benar digunakan (aktif bekerja) selama simulasi.

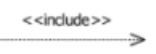
Dengan demikian, simulasi BPMN berperan penting dalam mengevaluasi kinerja proses bisnis, meningkatkan kualitas pelaksanaan, serta mengidentifikasi peluang perbaikan.

2.8.1.2 Use Case Diagram

Use case diagram ini adalah teknik untuk menggambarkan kebutuhan fungsional suatu sistem dengan menunjukkan fungsionalitas yang diharapkan dari sistem tersebut. Diagram ini berfokus kepada “apa” yang akan dilakukan sistem, bukan “bagaimana” cara melakukannya. Setiap *use case* menggambarkan interaksi antar aktor (pengguna atau sistem lain) dengan sistem. Sebuah *use case* biasanya menunjukkan tugas tertentu, misalnya *login* ke sistem, membuat daftar, dan lainnya. Aktor dapat berupa manusia atau perangkat lain yang berinteraksi dengan sistem untuk menjalankan tugasnya tersebut. Dengan demikian, *use case diagram* membantu pemangku kepentingan memahami dan mendefinisikan bagaimana sistem berfungsi dalam berbagai skenario (Hasanah & Untari, 2020). Berikut dapat dilihat pada Tabel 2. 2 Simbol *use case diagram*:

Tabel 2.2 Simbol *Use Case Diagram* (Ramdany et al., 2024)

Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili peran pengguna atau sistem eksternal yang berinteraksi dengan sistem.
	<i>Use Case</i>	Menggambarkan fungsi atau pekerjaan dilakukan yang dalam sistem
	<i>Association</i>	Penghubung antara aktor dengan <i>use case</i>

Simbol	Nama	Keterangan
	<i>Generalisasi</i>	Menggambarkan turunan dimana satu komponen menjadi bagian dari komponen lainnya.
	<i>Include Relationship</i>	Hubungan di mana suatu use case selalu menyertakan fungsi dari use case lain.
	<i>Extend Relationship</i>	Hubungan yang menunjukkan tambahan perilaku opsional pada use case utama.

Terdapat beberapa tahapan *use case diagram*, sebagai berikut (Ramdany et al., 2024):

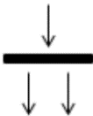
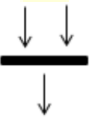
1. Menemukan aktor, mengidentifikasi siapa saja yang menggunakan atau berinteraksi dengan sistem (misalnya pelanggan, admin, atau perangkat eksternal).
2. Menemukan *use case*, menetapkan fungsi apa saja yang dapat dilakukan sistem untuk mendukung kebutuhan aktor.
3. Membuat skenario per *use case*, menggambarkan alur aktivitas antara aktor dan sistem saat menyelesaikan tugas tertentu.

2.8.1.3 Activity Diagram

Activity Diagram digunakan untuk memodelkan alur aktivitas atau proses bisnis dalam sistem, mulai dari titik awal hingga titik akhir. Diagram ini menunjukkan urutan langkah-langkah yang dilakukan, termasuk percabangan keputusan dan penggabungan alur kerja. Setiap aktivitas direpresentasikan dengan notasi sesuai fungsinya dijelaskan pada Tabel 2.3 Simbol *Activity Diagram*, sebagai berikut:

Tabel 2.3 Simbol *Activity Diagram* (Aliman, 2021)

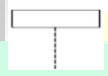
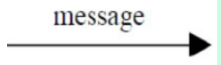
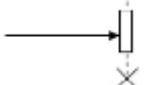
Simbol	Nama	Keterangan
	<i>Initial State</i>	Menunjukkan dimulainya alur aktivitas.

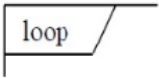
Simbol	Nama	Keterangan
		
	<i>Final State</i>	Menunjukkan berakhirnya alur aktivitas.
	<i>Activity</i>	Aktivitas atau kegiatan yang dilakukan dalam urutan kerja.
	<i>Fork</i>	Membagi satu alur menjadi dua atau lebih aktivitas secara paralel.
	<i>Join</i>	Menggabungkan beberapa alur menjadi satu alur kerja.
	<i>Decision</i>	Menyatakan titik pengambilan keputusan dengan kondisi tertentu.

2.8.1.4 Sequence Diagram

Sequence diagram adalah gambaran yang mengilustrasikan bagaimana objek-objek berkomunikasi satu sama lain. Diagram ini menunjukkan proses interaksi dan kelas-kelas yang terlibat dalam urutan pesan antar objek yang diperlukan untuk menjalankan fungsi tertentu (Honi & Iksari, 2023). *Sequence Diagram* adalah jenis diagram yang menggambarkan interaksi dinamis antar objek dalam sistem, serta urutan pesan yang dikirim di antara objek-objek tersebut saling berkomunikasi untuk menjalankan serangkaian peristiwa guna menghasilkan output tertentu. Diagram ini berguna untuk memvisualisasikan alur komunikasi antara objek selama eksekusi sistem. Berikut pada Tabel 2.4 Simbol *Sequence Diagram*:

Tabel 2.4 Simbol *Sequence Diagram* (Fatoni et al., 2020)

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan bagaimana orang berinteraksi dengan sistem
	<i>Boundary</i>	Menghubungkan aktor dengan antarmuka sistem, mewakili tampilan yang digunakan.
	<i>Control</i>	Objek yang ngatur alur proses, koordinasi, dan logika jalannya sistem supaya aktivitas bisa berjalan sesuai skenario
	<i>Entity</i>	Menunjukkan jenis informasi yang harus diambil, ditampilkan, dan disimpan oleh sistem (struktur data sistem).
	<i>Life Line</i>	Menggambarkan mulai dan berakhirnya suatu pesan
	<i>Message</i>	Menggambarkan bagaimana pesan dikirim dan diterima selama proses berlangsung
	<i>Activation</i>	Menunjukkan objek sedang melakukan aksi atau proses tertentu.
	Pesan tipe <i>Destroy</i>	Menyatakan bahwa suatu objek mengakhiri eksistensi objek lain.
	<i>Self Message</i>	Menunjukkan objek mengirim pesan ke dirinya sendiri (proses internal).

Simbol	Nama	Keterangan
	<i>Loop</i>	Menunjukkan aktivitas/pesan yang dijalankan berulang hingga kondisi terpenuhi.

2.8.2 Perancangan Sistem dan Perangkat Lunak (*System and Software Design*)

Tahap perancangan sistem dilakukan berdasarkan kebutuhan dan persyaratan yang telah dikumpulkan sebelumnya akan dibagi menjadi dua komponen utama, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Proses perancangan mencakup penyusunan arsitektur sistem secara menyeluruh untuk memastikan sistem dapat berfungsi sesuai kebutuhan pengguna (Sommerville, 2020b).

Perancangan sistem ini melibatkan pembuatan diagram seperti *Entity Relationship Diagram* (ERD) untuk basis data, *class diagram* untuk struktur perangkat lunak, serta desain *User Interface* (UI) untuk memudahkan interaksi pengguna dengan sistem. Tahap ini bertujuan untuk mempersiapkan sistem agar siap untuk implementasi dan diuji lebih lanjut.

2.8.2.1 Entity Relationship Diagram (ERD)

ERD adalah alat yang paling digunakan untuk pemodelan basis data, khususnya dalam sistem basis data relasional. ERD adalah sebuah diagram yang menggambarkan jenis-jenis entitas dalam suatu sistem, lengkap dengan atributnya, serta hubungan atau keterkaitan antar entitas tersebut. Dengan demikian, ERD memberikan gambaran abstrak mengenai bagaimana data disusun, disimpan, dan saling terhubung dalam sistem. Fokus utama ERD adalah memperlihatkan struktur data dan relasi antar data (Hasanah & Untari, 2020).

2.8.2.2 Class Diagram

Class Diagram adalah diagram yang menggambarkan struktur sistem melalui kelas-kelas yang dimiliki, atribut, metode, serta hubungan antar kelas. Diagram ini menunjukkan bagaimana komponen sistem terorganisasi dalam bentuk kelas dan bagaimana interaksi antar kelas terjadi (Ramdany, Aulia Kaidar, et al., 2024). Berikut Gambar simbol-simbol pada *class diagram*:

1. Kelas (*Class*)

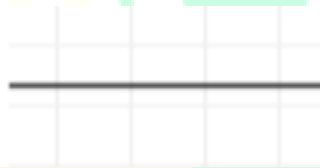
Digambarkan sebagai kotak yang terbagi menjadi tiga bagian: bagian pertama berisi nama kelas, bagian kedua berisi atribut (data yang dimiliki kelas), dan bagian ketiga berisi metode atau operasi yang dapat dijalankan kelas.



Gambar 2.2 Simbol *Class* (Ramdany, Aulia Kaidar, et al., 2024)

2. Asosiasi (*Association*)

Merepresentasikan hubungan antara dua kelas yang saling berinteraksi. Biasanya digambarkan dengan garis yang menghubungkan dua kelas, dan panah di ujung garis menunjukkan arah aliran informasi antara kelas tersebut.



Gambar 2.3 Simbol *Association* (Ramdany, Aulia Kaidar, et al., 2024)

3. Agregasi (*Aggregation*)

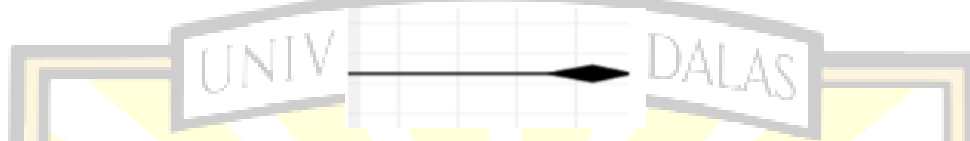
Menggambarkan hubungan bagian dan keseluruhan, di mana satu kelas merupakan bagian dari kelas lain, tetapi tetap dapat berdiri sendiri. Simbol ini ditunjukkan dengan garis yang di ujungnya terdapat diamond terbuka.



Gambar 2.4 Simbol *Aggregation* (Ramdany, Aulia Kaidar, et al., 2024)

4. Komposisi (*Composition*)

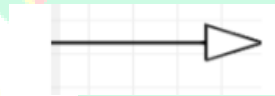
Menunjukkan hubungan yang lebih kuat daripada agregasi. Jika kelas utama dihapus, maka kelas yang bergantung juga ikut terhapus. Hubungan ini digambarkan dengan garis yang di ujungnya terdapat diamond padat



Gambar 2.5 Simbol Composition (Ramdany, Aulia Kaidar, et al., 2024)

5. Generalisasi (*Generalization*)

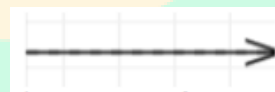
Menyatakan hubungan dari kelas khusus ke kelas umum. Kelas khusus mewarisi atribut dan metode dari kelas umum. Simbol ini digambarkan dengan garis berpanah menuju kelas umum.



Gambar 2.6 Simbol Generalization (Ramdany, Aulia Kaidar, et al., 2024)

6. Ketergantungan (*Dependency*)

Menggambarkan bahwa suatu kelas memerlukan kelas lain agar dapat berfungsi. Ditunjukkan dengan garis putus-putus berpanah dari kelas yang bergantung ke kelas yang digunakan.



Gambar 2.7 Simbol *Dependency* (Ramdany, Aulia Kaidar, et al., 2024)

2.8.3 Implementasi dan Pengujian Unit (*Implementation and Unit Testing*)

Tahap implementasi bertujuan untuk menerjemahkan desain perangkat lunak menjadi kode sumber yang dapat dijalankan. Proses pengkodean ini sering disebut fase implementasi, di mana setiap komponen desain diterjemahkan menjadi modul-modul program yang siap untuk diuji. Implementasi bertujuan untuk mewujudkan desain dalam bentuk sistem yang berfungsi (Mall, 2014).

Setelah proses pengkodean selesai, dilakukan pengujian unit untuk memastikan bahwa setiap bagian dari sistem berfungsi sesuai dengan yang diharapkan. Pengujian unit berfokus pada pengujian bagian-bagian kecil dari sistem secara individual untuk

memeriksa apakah mereka berjalan dengan benar. Pengujian unit bertujuan untuk memperbaiki kesalahan atau kekurangan dalam kode lebih awal, serta meminimalkan waktu debugging (Hasibuan & Dirgahayu, 2021).

2.8.3.1 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *open source* yang dirancang khusus untuk pengembangan web dan dapat disisipkan dalam dokumen HTML. Sintaksnya memiliki kemiripan dengan bahasa C, Java, dan Perl, sehingga relatif mudah dipelajari. PHP termasuk bahasa scripting sisi *server* (*server-side*), artinya proses pengolahan data dilakukan di server (Noviantoro, Silviana, et al., 2022). Dalam pengembangan web, PHP memiliki peran penting karena mampu memproses data, mengelola konten dinamis, dan berinteraksi dengan database seperti MySQL. Dengan PHP, *website* dapat dibuat menjadi lebih interaktif dan mampu menampilkan konten yang menyesuaikan kebutuhan pengguna. Selain itu, PHP dapat dipadukan dengan HTML serta *framework* seperti *Laravel* untuk mempercepat proses pengembangan dan mendukung pembuatan *website* yang lebih optimal (Arif, 2023).

2.8.3.2 Framework Laravel

Laravel adalah salah satu *framework* PHP yang menyediakan beragam fitur modern guna mempermudah pengembangan aplikasi web. *Framework* ini menyediakan berbagai fitur modern yang mendukung proses pembuatan aplikasi, seperti *Bundles*, *Eloquent ORM (Object-Relational Mapping)*, *Query Builder*, *Resource Controller*, *Blade*, *Migration*, *Middleware*, dan *Automatic Pagination* (Indah Melyani & Aji, 2023).

Dalam aplikasi berbasis *website*, umumnya terdapat tiga pembagian utama, yaitu pengolahan tampilan, pengolahan data, dan pengolahan logika kerja atau proses bisnis, yang dikenal dengan konsep *Model, View, Controller* (MVC). Laravel menggunakan pola pengembangan MVC yang memisahkan tampilan, data, dan logika aplikasi dengan tujuan mendukung prinsip *single responsibility* (SRP), yaitu pemisahan komponen berdasarkan fungsi masing-masing agar struktur sistem menjadi lebih teratur dan mudah dikelola (Ekohariadi et al., 2020).

2.8.3.3 MySql

MySQL adalah sistem manajemen basis data atau *Relational Database Management System* (RDBMS) yang menggunakan bahasa SQL untuk mengolah data. Aplikasi ini populer digunakan dalam pengembangan *web*. Sebagai RDBMS, MySQL menyimpan data dalam bentuk tabel yang terdiri dari kolom dan baris, sehingga memudahkan pengelolaan serta relasi antar data (Remawati, 2021).

Secara konsep, data di MySQL disimpan di dalam database, yang bisa dianggap sebagai wadah besar. Di dalam database terdapat tabel, yang berbentuk mirip lembar data dengan susunan baris dan kolom. Setiap baris (*row*) merepresentasikan satu data lengkap atau satu entitas, misalnya satu pelanggan atau satu buku. Sementara itu, kolom (*column/field*) berisi atribut dari data tersebut, seperti nama, alamat, atau tahun terbit. Titik pertemuan antara baris dan kolom disebut *field*, yaitu tempat nilai data disimpan. Selain itu, tabel dalam database bisa saling berhubungan. Misalnya, satu penulis dalam tabel penulis dapat memiliki banyak buku dalam tabel buku. Hal inilah yang membuat MySQL disebut sebagai sistem berbasis relasional (Silalahi, 2022).

2.8.3.4 PHPUnit

PHP Unit adalah *framework* untuk melakukan pengujian unit pada kode PHP. *Framework* ini menyediakan berbagai macam *assertion* untuk mengecek apakah kode berjalan sesuai dengan yang diharapkan. Dengan begitu, bug bisa lebih cepat ditemukan, dan kualitas perangkat lunak bisa tetap terjaga (Nur et al., 2024). Terdapat fungsi dan manfaat PHPUnit ini, sebagai berikut (Suasana, 2023):

1. Menjalankan pengujian unit untuk memeriksa bagian kecil dari aplikasi secara terpisah.
2. Menyediakan *assertion* seperti `assertEquals()` atau `assertTrue()` guna membandingkan hasil aktual dengan hasil yang diharapkan.
3. Menguji berbagai kondisi input, baik yang valid maupun tidak valid, agar aplikasi dapat menangani semua kemungkinan dengan benar.

Selain fungsi utama yang telah dijelaskan, penggunaan PHPUnit juga memberikan beberapa keuntungan bagi proses pengembangan perangkat lunak. Beberapa manfaat yang bisa dilihat, sebagai berikut:

1. Meningkatkan kualitas kode dengan mendeteksi bug lebih dini.
2. Mengurangi risiko error akibat perubahan kode dengan memastikan fungsionalitas tetap stabil setelah pembaruan.
3. Menyediakan dokumentasi eksplisit mengenai perilaku fungsi melalui skrip pengujian.

Pengujian unit dilakukan menggunakan *framework Pest* (berbasis PHPUnit) yang merupakan *library* bawaan Laravel untuk *automated testing*.

2.8.4 Integrasi dan Pengujian Sistem (*Integration and System Testing*)

Tahap integrasi dan pengujian sistem dilakukan dengan menggabungkan setiap unit atau modul program menjadi satu kesatuan. Setelah itu, dilakukan pengujian untuk memastikan semua kebutuhan perangkat lunak sudah terpenuhi. Jika pengujian berjalan dengan baik, sistem dapat diserahkan kepada pengguna untuk dioperasikan (Sommerville, 2020b) penelitian ini, pengujian dilakukan menggunakan metode *black box* Testing yaitu pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal atau kode program. Pengujian dilakukan dengan memberikan input tertentu ke sistem dan memeriksa apakah output yang dihasilkan sesuai dengan spesifikasi. Metode ini membantu memastikan sistem berjalan sebagaimana yang telah dirancang.

2.8.4.1 Black Box Testing

Metode *Black Box Testing* adalah teknik pengujian perangkat lunak yang berfokus pada fungsi dan spesifikasi sistem tanpa melihat kode program di dalamnya. Penguji memberikan berbagai macam input ke sistem dan kemudian memeriksa apakah output atau hasil yang ditampilkan sudah sesuai dengan yang diharapkan. Proses pengujiannya dilakukan dengan cara mencoba setiap form atau fitur yang ada dalam program. Tujuannya adalah untuk memastikan bahwa aplikasi yang dibuat benar-benar berfungsi sesuai dengan kebutuhan perusahaan (Shadiq et al., 2021). Pengujian Black Box pada penelitian ini dilakukan untuk memastikan bahwa setiap bagian program sudah sesuai dengan kebutuhan yang tertulis dalam spesifikasi. Dengan kata lain, pengujian ini dilakukan dengan cara menjalankan program sesuai skenario test case

yang telah dibuat, lalu mengamati apakah hasil yang muncul sesuai dengan yang diharapkan (Putra et al., 2020).

2.8.5 Pemeliharaan (*Operation and Maintenance*)

Tahap pemeliharaan merupakan fase terpanjang dalam siklus hidup perangkat lunak. Pada tahap ini, sistem dipasang dan mulai digunakan secara nyata. Aktivitas pemeliharaan meliputi perbaikan kesalahan yang tidak terdeteksi pada tahap sebelumnya, peningkatan kinerja sistem secara menyeluruh, serta penambahan atau perubahan fitur untuk memenuhi kebutuhan pengguna yang terus berkembang. Pemeliharaan dilakukan secara berkelanjutan agar sistem tetap berfungsi optimal dan mampu mendukung aktivitas pengguna dari waktu ke waktu (Sommerville, 2020b).

2.9 Penelitian Terkait

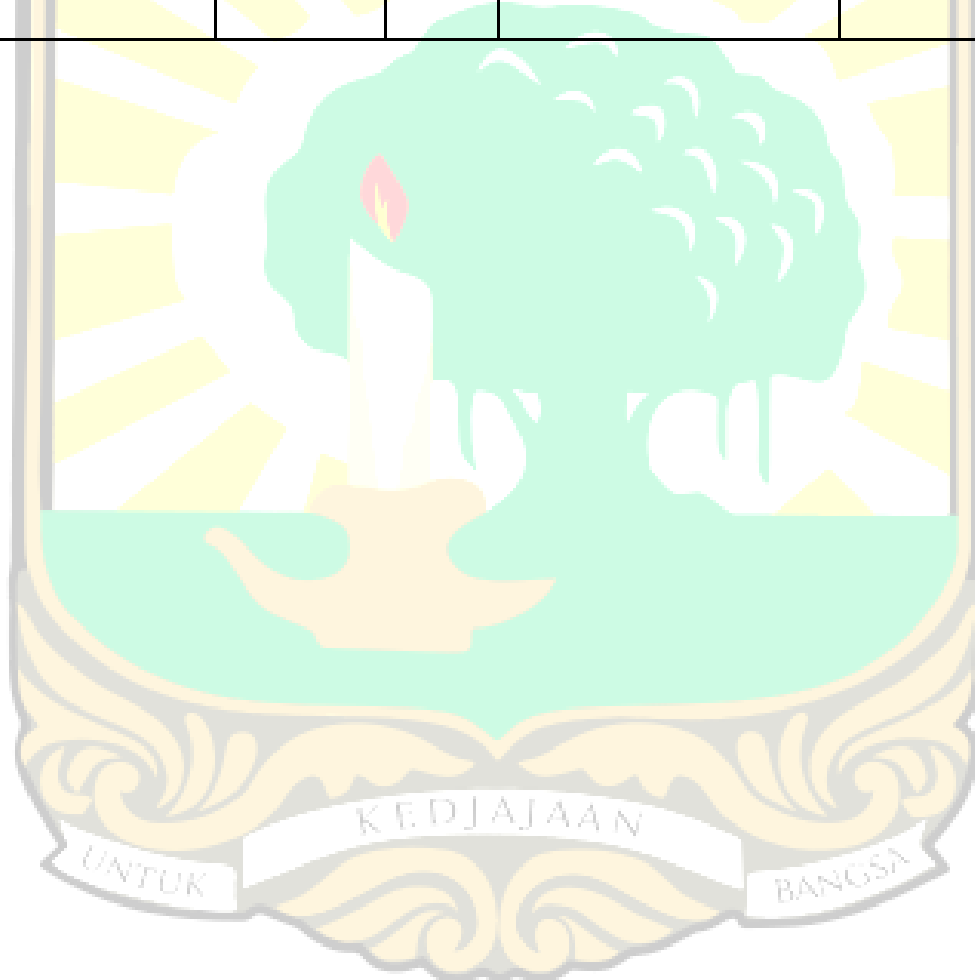
Penelitian terkait merupakan kajian terhadap karya ilmiah atau laporan yang telah dilakukan sebelumnya dan relevan dengan topik penelitian ini. Tujuannya adalah untuk memberikan gambaran mengenai sejauh mana penelitian serupa telah dilaksanakan, mengidentifikasi celah atau permasalahan yang belum terselesaikan, serta menjadi dasar untuk menunjukkan kontribusi dan keaslian penelitian yang sedang dilakukan. Berikut beberapa penelitian terkait yang dijelaskan dalam Tabel 2.5 Penelitian Terkait:

Tabel 2.5 Penelitian Terkait

No	Judul	Peneliti	Tahun	Masalah	Fitur	Metode	Kesimpulan
1	Rancang Bangun Sistem Penyewaan Baju dan Dekorasi Berbasis Web pada Nita Wedding Organizer	Tri Wahyuni, Indriyanti, Erni Ermawati, Haerul Fatah, Nurul Ichsan	2021	Nita Wedding Organizer Yogyakarta masih menggunakan pencatatan manual, menyebabkan kesalahan data dan laporan. Sistemnya belum otomatis, rentan kesalahan manusia.	Sistem berbasis web mempermudah pengelolaan dan mempercepat pembuatan laporan transaksi.	<i>Waterfall</i>	Sistem penyewaan berbasis web ini berhasil mempermudah Nita Wedding Organizer dalam mengelola data penyewaan dan membuat laporan transaksi lebih sistematis. Sistem ini meningkatkan kecepatan dan efisiensi dalam proses penyewaan.
2	Rancang Bangun Sistem Informasi Penyewaan Camping Ground Berbasis Web pada Lembah Permai Resor	Yesi Leony Julianti, Tati Mardiana, Ami Rahmawati	2022	Sistem manual membuat proses transaksi dan pemesanan lambat, serta distribusi informasi kepada pelanggan tidak merata	Pengguna: melihat paket, login/daftar, memesan, bayar, unggah bukti pembayaran, cetak invoice. Admin: kelola data camping ground, customer, pesanan, konfirmasi pembayaran, laporan	<i>Waterfall</i>	Sistem ini mempercepat transaksi, memudahkan informasi, dan promosi. Kelebihannya, akses cepat, data terkelola, promosi efisien. Kekurangannya, bergantung pada koneksi internet dan perlu pemeliharaan data yang rutin
3	Rancang Bangun Sistem Informasi	Agustina Tamo Ina	2023	Banyak gedung di Kota Kefamenanu yang belum	Daftar gedung yang tersedia, Informasi tentang	<i>Waterfall</i>	Sistem ini memudahkan penyewaan gedung secara online, mempercepat

No	Judul	Peneliti	Tahun	Masalah	Fitur	Metode	Kesimpulan
	Penyewaan Gedung Resepsi Pernikahan Berbasis Web	Bili, Yoseph P. K. Kelen, Siprianus Septian, Manek, Leonard P. Gelu		memiliki <i>website</i> untuk penyewaan. Pengguna masih harus datang langsung ke lokasi untuk mendapatkan informasi tentang ketersediaan gedung, yang memakan waktu, terutama bagi mereka yang sibuk.	fasilitas dan harga., Kemampuan untuk memilih dan memesan gedung secara online, Pengguna dapat melihat gambar dan detail gedung, Sistem notifikasi pemesanan dan konfirmasi.		transaksi, dan meningkatkan pendapatan pemilik. Namun, sistem memerlukan koneksi internet yang stabil dan belum optimal untuk menangani banyak permintaan sekaligus.
4	Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode <i>Prototype</i>	Siswidiyano, Ahmad Munif, Diah Wijayanti, Eko Haryadi	2020	Proses penyewaan rumah kontrakan masih dilakukan secara manual, mengandalkan selebaran dan pencarian langsung yang memakan waktu dan biaya.	Pencarian kontrakan berdasarkan lokasi, harga, dan fasilitas. Pemesanan dan pembayaran online. Manajemen data penyewa dan pemilik kontrakan. Forum diskusi dan halaman informasi kontak.	<i>prototype</i>	Sistem informasi penyewaan rumah kontrakan berbasis web mempermudah pencarian, pemesanan, dan pengelolaan data penyewa serta pemilik. Namun, sistem ini masih perlu perbaikan di bidang keamanan, fitur pemulihan akun, dan cadangan data untuk mencegah kerugian akibat kehilangan informasi.
5	Aplikasi Chatbot Persewaan Kost	Dicky Bagas Oktaviandia,	2023	Promosi kost masih manual, informasi terbatas, calon	<i>Chatbot</i> tanya-jawab otomatis, informasi kost	<i>Waterfall</i>	Aplikasi chatbot kost berbasis web mampu mengubah promosi manual

No	Judul	Peneliti	Tahun	Masalah	Fitur	Metode	Kesimpulan
	Berbasis Web Menggunakan Metode Waterfall (Studi Kasus: Kost Pak Purwanto)	Saiful Ulya, Achmad Ridwan, Avira Budianita		penyewa sulit mengetahui detail lokasi, fasilitas, biaya, dan kontak pemilik.	(lokasi, biaya, fasilitas, foto, kontak), antarmuka web, dukungan pembayaran transfer/tunai.		menjadi layanan digital terstruktur. Sistem ini memberi akses informasi cepat bagi calon penyewa sekaligus mendukung pemilik kost dalam pemasaran, meski masih terbatas pada respon yang diprogram.



Berdasarkan penelitian terdahulu, dapat disimpulkan bahwa sistem informasi berbasis web mampu mengatasi berbagai kendala dalam pengelolaan manual, seperti kesalahan pencatatan, keterbatasan distribusi informasi, dan proses operasional yang tidak teratur. Penerapan fitur pemesanan online, pencatatan transaksi digital, serta pembuatan laporan otomatis terbukti dapat meningkatkan keteraturan pengelolaan sekaligus memberikan kemudahan bagi pelanggan.

Penelitian ini memiliki beberapa perbedaan dibandingkan penelitian sebelumnya. Pertama, objek penelitian berfokus pada koperasi yang memiliki karakteristik berbeda dengan usaha komersial biasa, karena melibatkan anggota dan pengurus dalam pengelolaan. Kedua, sistem yang dikembangkan mengintegrasikan tiga jenis layanan penyewaan sekaligus gedung pertemuan, lokasi foto *pre wedding*, dan lapangan golf dalam satu sistem terpusat. Sementara penelitian terdahulu umumnya hanya mengelola satu jenis penyewaan. Ketiga, sistem ini dirancang untuk mendukung tiga peran pengguna yaitu admin, pelanggan, dan manajer, dengan fitur yang disesuaikan dengan kebutuhan masing-masing. Dengan demikian, penelitian ini diharapkan dapat menghasilkan sistem informasi yang lebih terintegrasi, mudah diakses, dan relevan dengan kebutuhan digitalisasi koperasi.

BAB III

METODOLOGI PENELITIAN

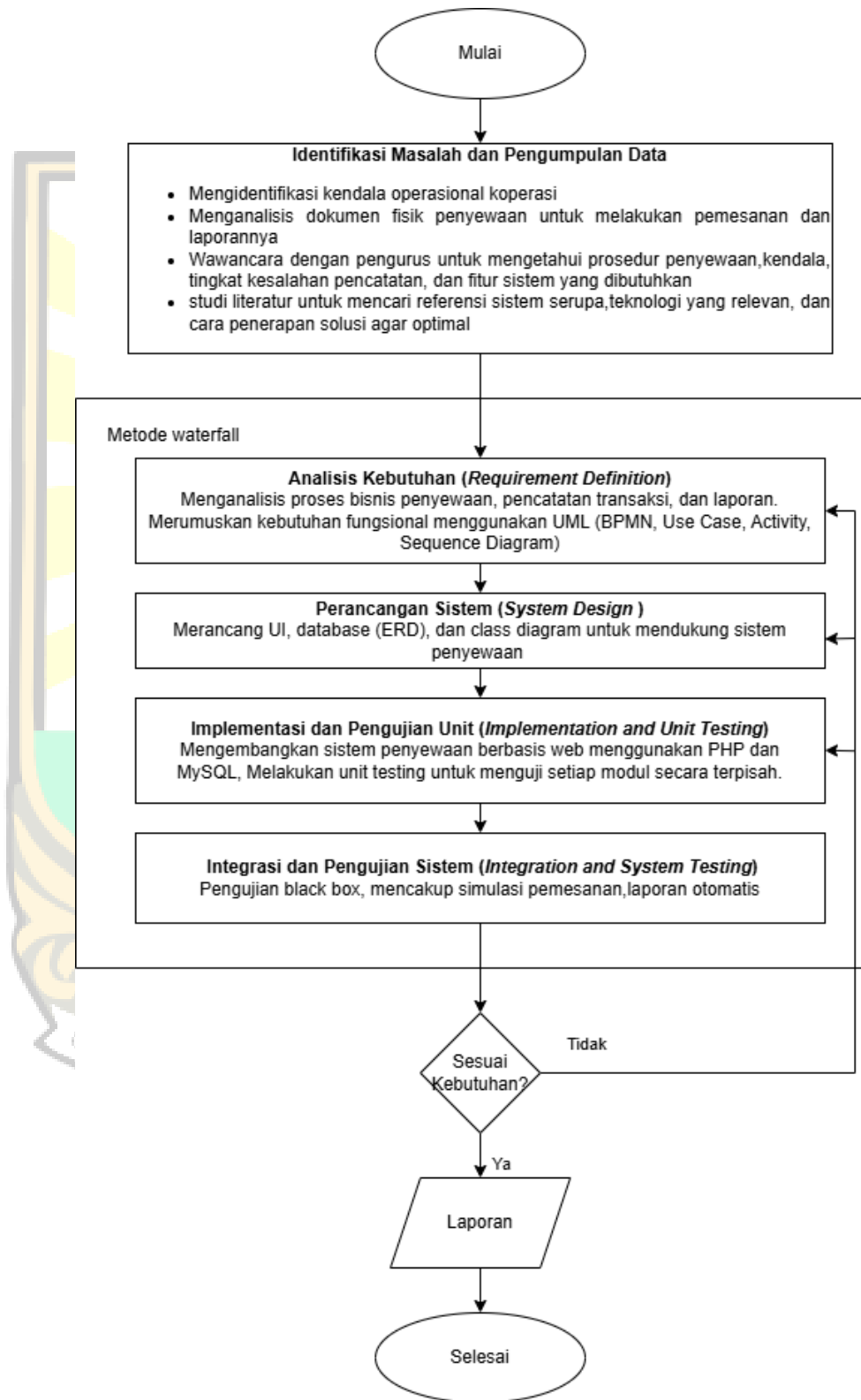
3.1 Objek Penelitian

Objek penelitian ini adalah Koperasi Konsumen Hydro Utama Bukittinggi, mengelola beberapa unit usaha, termasuk penyewaan gedung pertemuan, lapangan golf, dan lokasi foto *pre wedding*. Penelitian ini difokuskan pada pengembangan sistem informasi berbasis web untuk mendukung pengelolaan layanan penyewaan tersebut.

3.2 Flowchart Penelitian

Metode penelitian yang digunakan dalam pengembangan sistem ini mengikuti tahapan *Waterfall*, mulai dari identifikasi masalah, analisis kebutuhan, perancangan, implementasi, hingga pengujian sistem. Pada penelitian ini, tahapan tersebut dilengkapi dengan penyusunan laporan sebagai bentuk dokumentasi hasil penelitian. Seluruh tahapan divisualisasikan dalam *Flowchart* Penelitian yang memberikan gambaran menyeluruh mengenai proses pengembangan sistem.

Proses dimulai dari kegiatan identifikasi masalah dan pengumpulan data melalui analisis dokumen, wawancara, serta studi literatur. Tahap berikutnya adalah analisis kebutuhan untuk merumuskan proses bisnis dan kebutuhan fungsional sistem. Setelah itu dilakukan perancangan sistem, yang mencakup rancangan basis data, antarmuka, serta *diagram*. Hasil rancangan kemudian diimplementasikan menjadi sistem berbasis web, dilanjutkan dengan pengujian unit untuk memastikan setiap modul berfungsi dengan baik sebelum digabungkan. Tahap berikutnya adalah integrasi dan pengujian sistem secara keseluruhan menggunakan metode *Black Box Testing*. Apabila hasil pengujian belum sesuai kebutuhan, maka dilakukan perbaikan sedangkan jika sudah sesuai, penelitian dilanjutkan dengan penyusunan laporan akhir, sebagaimana ditunjukkan pada Gambar 3.1 *Flowchart* Penelitian



Gambar 3.1 Flowchart Penelitian

3.2.1 Identifikasi Masalah dan Pengumpulan Data

Pada tahap awal penelitian adalah mengidentifikasi permasalahan yang terjadi pada proses penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi. Beberapa permasalahan yang ditemukan antara lain proses penyewaan yang masih pelanggan harus datang langsung atau menghubungi penanggung jawab atau PIC melalui WhatsApp. Pencatatan transaksi yang masih dilakukan menggunakan nota kertas dan file Excel yang sederhana, sehingga rawan kesalahan, duplikasi, dan sulit ditelusuri kembali ketika data semakin banyak. Keterbatasan dalam penyampaian informasi karena hanya mengandalkan banner atau poster, sehingga pelanggan kesulitan mengakses informasi secara cepat. Untuk memastikan kesesuaian permasalahan dengan kondisi lapangan, dilakukan pengumpulan data menggunakan tiga metode:

- a. Wawancara dilakukan dengan pengurus koperasi untuk mengetahui prosedur penyewaan, kendala yang dialami, tingkat kesalahan pencatatan, serta fitur sistem yang diinginkan pada sistem baru. Topik pertanyaannya mencakup:
 - 1) Prosedur penyewaan unit usaha di koperasi saat ini.
 - 2) Kendala yang sering dihadapi dalam proses penyewaan dan pencatatan transaksi.
 - 3) Tingkat kesalahan dalam pengelolaan data penyewaan serta dampaknya bagi koperasi.
 - 4) Cara penyampaian informasi layanan kepada pelanggan saat ini serta keterbatasannya.
 - 5) Kebutuhan yang harus dipenuhi untuk mengembangkan sistem pencatatan, pengelolaan data, dan penyampaian informasi yang lebih baik.
- b. Analisis dokumen dilaksanakan dengan menelaah berbagai arsip yang dimiliki koperasi. Dokumen yang dianalisis mencakup nota kertas, file Excel pencatatan transaksi, dokumen profil koperasi, serta file promosi layanan penyewaan.
- c. Studi literatur dilakukan mengkaji buku, artikel ilmiah, dan penelitian terdahulu terkait sistem informasi penyewaan berbasis web, termasuk penggunaan *chatbot*. Langkah ini bertujuan memberi landasan teoritis yang memperkuat rancangan

sistem dan memastikan solusi yang diusulkan sejalan dengan perkembangan teknologi.

Hasil dari tahap ini menjadi bahan utama dalam menentukan kebutuhan sistem yang akan dikembangkan.

3.2.2 Analisis Kebutuhan (*Requirement Definition*)

Setelah identifikasi masalah dan pengumpulan data, tahap berikutnya adalah menganalisis kebutuhan sistem berdasarkan data yang sudah dikumpulkan. Analisis kebutuhan bertujuan untuk mendefinisikan fungsi yang harus dimiliki sistem agar mampu menyelesaikan masalah yang ada. Kebutuhan sistem dikelompokkan menjadi kebutuhan fungsional fitur yang tersedia, seperti registrasi pengguna, pemesanan online, pengecekan jadwal, unggah bukti pembayaran, *chatbot*, umpan balik, serta laporan dan kebutuhan non-fungsional seperti kemudahan penggunaan, keamanan data, serta kecepatan akses.

Untuk memperjelas kebutuhan, digunakan beberapa model pemodelan BPMN untuk menggambarkan alur bisnis penyewaan dari awal hingga akhir, baik kondisi saat ini maupun rancangan yang diusulkan. *Use case diagram* untuk menunjukkan interaksi antara aktor dengan sistem sesuai peran masing-masing. *Activity diagram* untuk memperlihatkan alur aktivitas dalam setiap proses pemesanan atau pengelolaan data. *Sequence diagram* untuk menggambarkan komunikasi antar objek dalam sistem secara berurutan. Hasil dari tahap ini berupa spesifikasi kebutuhan sistem dan model interaksi yang akan menjadi acuan pada tahap perancangan.

3.2.3 Perancangan Sistem dan Perangkat Lunak (*System and Software Design*)

Setelah kebutuhan sistem terdefinisi, tahap berikutnya adalah perancangan sistem. Tujuan utama tahap ini adalah menghasilkan rancangan yang dapat diterjemahkan menjadi kode program tanpa kehilangan makna dari kebutuhan yang sudah ditetapkan. Beberapa komponen perencanaan yang disusun antara lain, *Entity Relationship Diagram* (ERD), digunakan untuk memodelkan basis data. Dengan adanya ERD, struktur penyimpanan data lebih jelas dan mengurangi potensi inkonsistensi. *Class diagram* menggambarkan struktur perangkat lunak dalam bentuk kelas, atribut, metode, serta relasi antar kelas, *diagram* ini membantu pengembang

dalam membangun logika program yang sesuai dengan rancangan awal. Desain Antarmuka atau *user interface* (UI) disusun untuk menampilkan rancangan tampilan halaman sistem, Desain UI dirancang sederhana agar mudah digunakan pelanggan maupun pengurus koperasi. Perancangan ini menjadi pedoman utama pada tahap implementasi, sehingga meminimalkan perubahan besar di tahap berikutnya.

3.2.4 Implementasi dan Pengujian Unit (*Implementation and Unit Testing*)

Setelah desain sistem selesai, tahap berikutnya adalah implementasi dilakukan dengan menerjemahkan hasil perancangan ERD, *class diagram*, dan UI ke dalam kode program menggunakan bahasa pemrograman PHP dengan *framework* Laravel, serta basis data MySQL. Pemilihan Laravel didasarkan pada dukungan struktur Model, View, *Controller* (MVC) yang memudahkan pengembangan, pemeliharaan, serta pemisahan logika bisnis dengan tampilan antarmuka. Tahap implementasi meliputi Pembuatan modul antarmuka pengguna sesuai desain UI. Pembuatan modul pengelolaan data dan transaksi berdasarkan rancangan ERD dan *class diagram*. Integrasi fitur pendukung seperti umpan balik dan *chatbot* berbasis aturan (*rule based chatbot*) untuk memberikan jawaban otomatis sesuai kata kunci terkait layanan penyewaan.

Setelah implementasi selesai, dilakukan pengujian unit untuk memastikan setiap modul atau fungsi berjalan sesuai harapan. Pengujian ini menggunakan *framework* PHPUnit, yang memungkinkan penulisan skrip uji otomatis sehingga hasil lebih konsisten dan cepat didapatkan. Dengan pengujian unit, kesalahan pada level kecil dapat ditemukan lebih awal sebelum sistem diuji secara keseluruhan. Setelah tahap pengujian selesai, sistem siap untuk diuji lebih pada tahap integrasi.

3.2.5 Integrasi dan Pengujian Sistem (*Integration and System Testing*)

Setelah setiap bagian sistem diuji secara terpisah, tahap berikutnya adalah pengujian sistem secara keseluruhan. Jika semua modul lulus pengujian unit, tahap berikutnya adalah mengintegrasikan seluruh komponen sistem dan melakukan pengujian menyeluruh. Pengujian ini menggunakan pendekatan *Black Box Testing*, yaitu pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan kode internal. Metode ini dipilih karena sesuai dengan tujuan penelitian, yaitu memastikan

sistem dapat mendukung kebutuhan koperasi dari sisi pengguna. Beberapa skenario pengujian yang dilakukan antara lain, pemesanan layanan dengan jadwal tersedia, pemesanan dengan konflik jadwal, unggah bukti pembayaran, serta akses laporan penyewaan. Hasil pengujian menjadi dasar untuk menilai apakah sistem telah siap digunakan dan mampu menyelesaikan permasalahan yang dihadapi koperasi

3.2.6 Laporan

Tahap terakhir adalah penyusunan laporan penelitian. Laporan ini berfungsi sebagai dokumentasi resmi yang menjelaskan seluruh proses pengembangan, mulai dari identifikasi masalah, analisis kebutuhan, perancangan, implementasi, hingga pengujian. Selain itu, laporan juga memuat kendala yang dihadapi, strategi penyelesaiannya, serta rekomendasi untuk pengembangan lebih lanjut. Dokumen laporan tidak hanya menjadi bukti akademik, tetapi juga bisa dimanfaatkan koperasi sebagai bahan evaluasi dan dasar pengambilan keputusan ketika ingin mengembangkan sistem lebih lanjut.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas hasil analisis terhadap proses bisnis yang berjalan di Koperasi Konsumen Hydro Utama Bukittinggi, termasuk kendala yang dihadapi serta kebutuhan sistem informasi penyewaan berbasis *web*. Analisis dilakukan untuk memperoleh gambaran nyata mengenai kondisi operasional koperasi yang akan menjadi dasar perancangan sistem. Selanjutnya, rancangan sistem disajikan melalui berbagai diagram seperti *Business Process Model and Notation* (BPMN), *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Entity Relationship Diagram* (ERD), *Class Diagram*, dan desain antarmuka pengguna.

4.1 Identifikasi Masalah dan Pengumpulan Data

Tahap awal analisis difokuskan pada pemahaman kondisi lapangan pada unit penyewaan. Data dikumpulkan melalui wawancara dengan pengurus koperasi serta telaah terhadap dokumen transaksi yang digunakan dalam kegiatan operasional. Hasil analisis ini menjadi dasar dalam menentukan kebutuhan sistem yang akan dikembangkan.

4.1.1 Wawancara

Wawancara dilakukan dengan Manajer Koperasi Konsumen Hydro Utama Bukittinggi pada 23 Januari 2025. Berdasarkan hasil wawancara, diperoleh informasi bahwa koperasi mengelola empat unit usaha utama, yaitu unit toko, unit pembiayaan dan pinjaman, unit pengadaan barang/jasa, serta unit penyewaan. Pada unit penyewaan, koperasi menyediakan layanan penyewaan berbagai layanan, seperti lapangan golf, gedung pertemuan, lokasi foto *pre wedding*. Beberapa layanan juga dilengkapi dengan fasilitas tambahan yang dapat dipilih sesuai kebutuhan pelanggan.

Saat ini proses penyewaan saat ini masih secara manual, di mana pelanggan harus datang langsung ke kantor koperasi atau menghubungi pengurus (PIC) melalui WhatsApp. Transaksi dicatat menggunakan file Excel, sedangkan bukti pembayaran diberikan dalam bentuk nota kertas. Sistem ini menimbulkan berbagai kendala, seperti

potensi kesalahan pencatatan akibat data yang masih manual, risiko jadwal penyewaan tumpang tindih, keterbatasan akses informasi bagi pelanggan karena hanya bisa diperoleh melalui PIC, metode pembayaran yang masih terbatas pada tunai, serta arsip transaksi yang berpotensi berpotensi hilang dan sulit ditelusuri seiring meningkatnya jumlah data.

Berdasarkan kondisi tersebut, pihak koperasi mengharapkan adanya sistem berbasis *web* yang memungkinkan pemesanan dilakukan secara *online*, menyediakan metode pembayaran melalui transfer bank dengan unggah bukti pembayaran yang kemudian diverifikasi oleh admin, mengatur jadwal penyewaan secara terintegrasi, menampilkan informasi layanan yang dapat diakses langsung oleh pelanggan, serta menyediakan fitur ulasan pelanggan sebagai bahan evaluasi peningkatan kualitas layanan.

4.1.2 Analisis Dokumen

Selain wawancara, analisis juga dilakukan terhadap dokumen transaksi dan arsip operasional yang digunakan oleh koperasi. Hasil telaah dokumen menunjukkan bahwa pencatatan administrasi transaksi masih dilakukan menggunakan nota kertas dan file Excel. Kondisi ini menyebabkan data penyewaan mudah hilang, bercampur dengan data dari unit usaha lain, serta sulit ditelusuri kembali ketika dibutuhkan.

Dari sisi promosi, layanan penyewaan hanya dipasarkan melalui media sederhana, seperti poster atau banner yang dibuat menggunakan *template* PowerPoint. Penyebaran media promosi tersebut masih terbatas sehingga belum mampu menjangkau calon pelanggan secara luas.

Secara keseluruhan, hasil analisis dokumen menegaskan bahwa koperasi memerlukan sistem informasi berbasis *web* yang mampu mengintegrasikan pengelolaan data dan transaksi, memperluas jangkauan promosi layanan, serta menyediakan akses informasi digital yang terstruktur dan mudah digunakan oleh pengurus maupun pelanggan.

4.2 Analisis Sistem

Tahap ini membahas kondisi sistem yang saat ini diterapkan di Koperasi Konsumen Hydro Utama Bukittinggi serta rancangan sistem yang diusulkan. Analisis

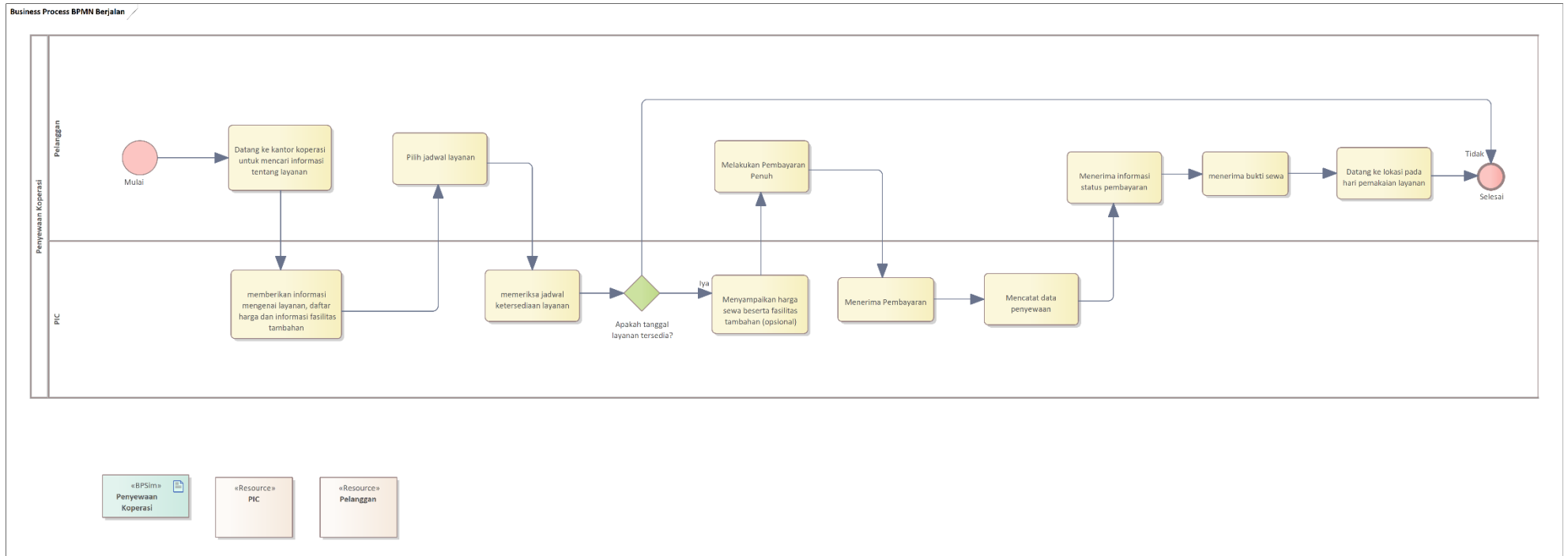
ini bertujuan untuk memahami kebutuhan pengguna dan memodelkan interaksi sistem dengan aktor yang terlibat melalui BPMN, *use case diagram*, *activity diagram*, dan *sequence diagram*. Selain itu, tahap ini juga mencakup identifikasi kebutuhan fungsional sebagai dasar perancangan sistem.

4.2.1 Business Process Model and Nation (BPMN)

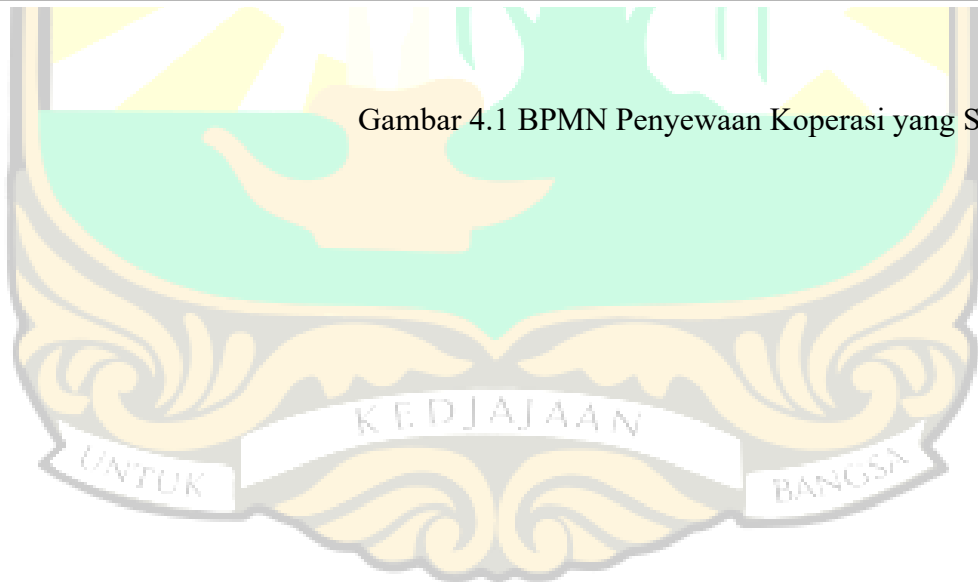
Pada tahap ini dilakukan pemodelan alur proses bisnis menggunakan BPMN. Pemodelan ini bertujuan untuk menggambarkan proses bisnis yang berjalan serta rancangan proses yang diusulkan di Koperasi Konsumen Hydro Utama Bukittinggi. Melalui BPMN, alur kegiatan, interaksi antar aktor, serta titik keputusan dapat divisualisasikan secara jelas dan menjadi dasar bagi pembuatan diagram lanjutan.

4.2.1.1 BPMN Sistem yang Sedang Berjalan

Proses penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi saat ini melibatkan dua aktor utama, yaitu pelanggan dan PIC (*Person In Charge*) yang bertanggung jawab mengelola penyewaan. BPMN ini menggambarkan tahapan prosedur penyewaan, mulai dari proses pemesanan oleh pelanggan hingga pengolahan data oleh PIC. Alur proses bisnis penyewaan yang sedang berjalan dapat dilihat pada Gambar 4.1.



Gambar 4.1 BPMN Penyewaan Koperasi yang Sedang Berjalan



Alur proses berdasarkan gambar dari BPMN di atas yang berjalan saat ini:

1. Pelanggan datang langsung ke kantor koperasi untuk memperoleh informasi terkait layanan penyewaan.
2. PIC memberikan penjelasan mengenai jenis layanan, daftar harga, jadwal yang tersedia, serta fasilitas tambahan yang dapat digunakan.
3. Setelah menerima informasi, pelanggan menentukan pilihan layanan dan jadwal yang diinginkan.
4. PIC memeriksa ketersediaan jadwal layanan:
 - a. Jika tidak tersedia, proses penyewaan dihentikan.
 - b. Jika tersedia, PIC menyampaikan harga sewa beserta opsi fasilitas tambahan, lalu proses berlanjut ke tahap pembayaran.
5. Pelanggan melakukan pembayaran sesuai kesepakatan.
6. PIC menerima pembayaran dan mencatat data penyewaan ke nota kertas serta mengarsipkan di Excel sebagai laporan.
7. Setelah transaksi selesai, pelanggan mendapatkan bukti penyewaan dan hadir ke lokasi sesuai tanggal penggunaan layanan.
8. Proses berakhir setelah pelanggan menggunakan layanan sesuai jadwal yang telah dipesan.

Berdasarkan hasil analisis, dapat disimpulkan bahwa sistem penyewaan yang berjalan saat ini memiliki beberapa kelemahan utama sebagai berikut:

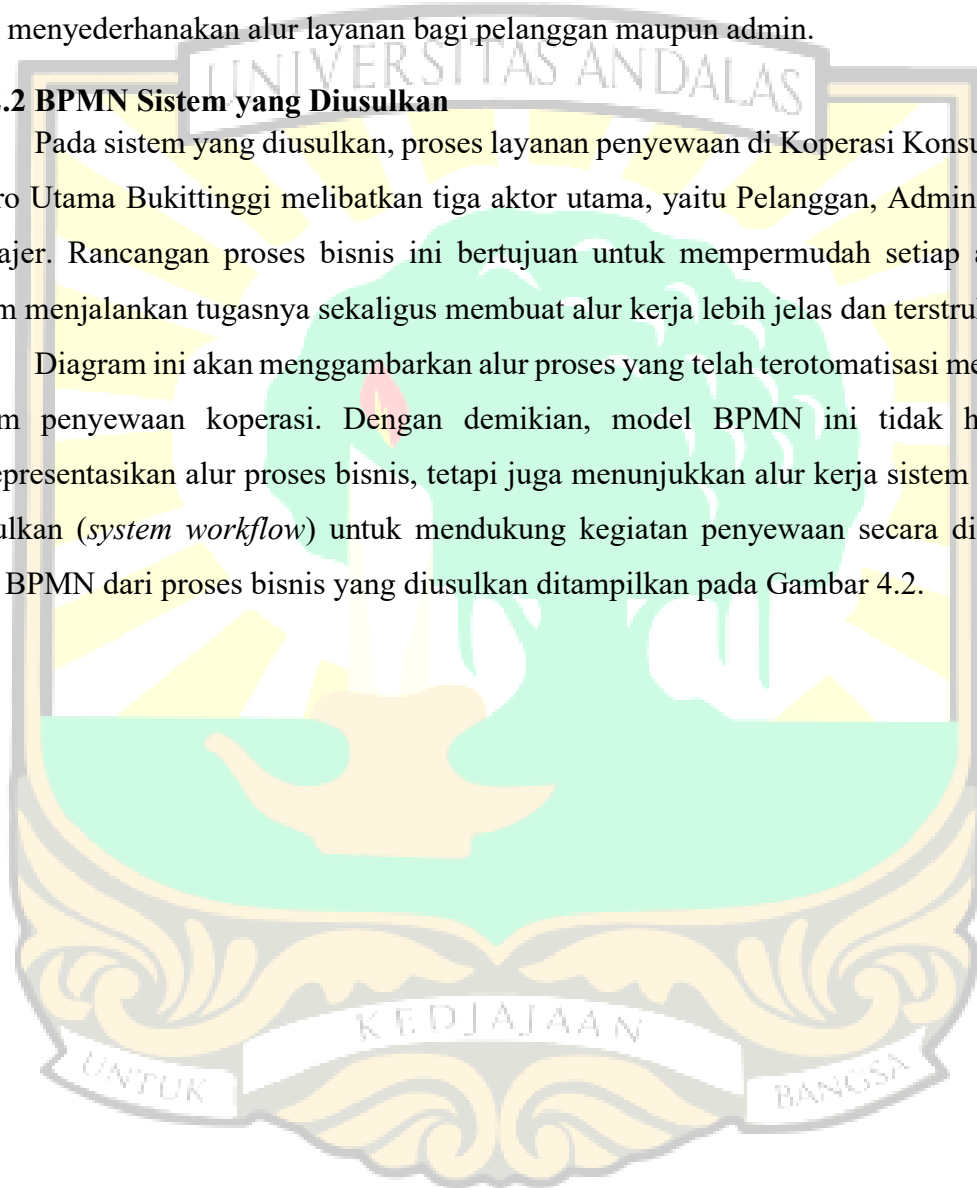
- 1) Pencatatan transaksi masih menggunakan nota kertas, sehingga rentan terjadi kehilangan data, serta pemantauan jadwal yang kurang akurat.
- 2) Data penyewaan dan pembayaran tersimpan pada media yang berbeda, sehingga penyusunan laporan menjadi lebih lama, pencarian data sulit dilakukan, dan analisis kinerja menjadi terhambat.
- 3) Proses pemesanan belum terhubung dengan sistem *online*. Pelanggan harus datang langsung ke koperasi atau menghubungi PIC melalui WhatsApp, yang memerlukan waktu lebih banyak dan membatasi kemudahan akses terhadap layanan.

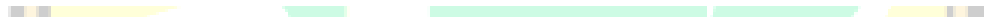
Berdasarkan kondisi tersebut, sistem yang diusulkan perlu mampu mengintegrasikan proses secara digital, menyatukan seluruh data dalam satu platform, serta menyederhanakan alur layanan bagi pelanggan maupun admin.

4.2.2.2 BPMN Sistem yang Diusulkan

Pada sistem yang diusulkan, proses layanan penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi melibatkan tiga aktor utama, yaitu Pelanggan, Admin, dan Manajer. Rancangan proses bisnis ini bertujuan untuk mempermudah setiap aktor dalam menjalankan tugasnya sekaligus membuat alur kerja lebih jelas dan terstruktur.

Diagram ini akan menggambarkan alur proses yang telah terotomatisasi melalui sistem penyewaan koperasi. Dengan demikian, model BPMN ini tidak hanya merepresentasikan alur proses bisnis, tetapi juga menunjukkan alur kerja sistem yang diusulkan (*system workflow*) untuk mendukung kegiatan penyewaan secara digital. Alur BPMN dari proses bisnis yang diusulkan ditampilkan pada Gambar 4.2.





Alur proses berdasarkan gambar dari BPMN di atas yang akan diusulkan:

1. Registrasi untuk pelanggan baru. Pelanggan melakukan pendaftaran akun, dan data diverifikasi secara otomatis oleh sistem koperasi.
2. *Login* pelanggan, setelah registrasi berhasil pelanggan dapat masuk ke sistem menggunakan akun yang telah terdaftar.
3. Pelanggan dapat melihat daftar layanan penyewaan yang disertai dengan informasi harga, fasilitas utama dan tambahan, serta jadwal ketersediaan.
4. Pelanggan kemudian memilih tanggal yang diinginkan.
 - a. Jika tanggal tidak tersedia, sistem akan menampilkan pemberitahuan jadwal tidak dapat dipesan, dan pelanggan dapat memilih untuk tidak melanjutkan pemesanan.
 - b. Jika tanggal tersedia, lanjut ke tahap pembayaran.
5. Pemilihan metode pembayaran, pelanggan meninjau rincian layanan dan jadwal, kemudian memilih metode pembayaran:
 - a. Pembayaran penuh, pelanggan melunasi seluruh biaya secara langsung.
 - b. Pembayaran uang muka DP (*Down Payment*), pelanggan membayar sebagian biaya terlebih dahulu.
6. Proses pembayaran dan unggah bukti, pelanggan melakukan transfer sesuai metode yang dipilih dan mengunggah bukti pembayaran melalui sistem.
 - a. Jika memilih DP, sistem mencatat status pembayaran serta menampilkan informasi mengenai jumlah dan tenggat pelunasan.
7. Verifikasi pembayaran oleh admin, admin koperasi memeriksa dan memvalidasi bukti pembayaran yang diunggah.
 - a. Jika tidak valid, pelanggan diminta memperbaiki atau mengunggah ulang.
 - b. Jika valid, status transaksi berubah menjadi berhasil (DP terkonfirmasi atau lunas penuh).
8. Pelanggan melunasi pembayaran DP, jika menggunakan DP pelanggan akan melakukan pelunasan sebelum jatuh tempo dan mengunggah bukti pelunasan. Admin kemudian memverifikasi kembali:

- 1) Jika valid, status berubah menjadi lunas, dan sistem menerbitkan bukti pemesanan.
 - 2) Jika tidak valid, pelanggan diminta untuk melakukan unggah ulang bukti pembayaran.
9. Setelah pembayaran terverifikasi, pelanggan menerima notifikasi status pembayaran dan memperoleh bukti penyewaan dari sistem.
 10. Pelanggan hadir dan menggunakan layanan sesuai jadwal yang telah dipesan.
 11. Setelah layanan selesai digunakan, pelanggan dapat memberikan ulasan atau *rating* sebagai umpan balik. Data ini tersimpan dalam sistem sebagai bahan evaluasi kualitas layanan.
 12. Admin dan manajer dapat mengakses data transaksi, laporan, serta hasil umpan balik pelanggan. Manajer memantau aktivitas penyewaan melalui *dashboard* sebagai bahan evaluasi dan dasar pengambilan keputusan strategis terkait pengembangan layanan koperasi.

Selain proses utama yang digambarkan dalam BPMN, sistem juga dilengkapi dengan fitur chatbot informatif untuk mendukung peningkatan akses informasi bagi pelanggan. Berdasarkan analisis tersebut, kebutuhan utama sistem yang diusulkan dapat dirumuskan sebagai berikut:

- 1) Digitalisasi dan integrasi data dalam satu sistem untuk mengurangi potensi kesalahan akibat pencatatan manual menggunakan nota kertas maupun file Excel.
- 2) Penyediaan akses informasi ketersediaan jadwal secara langsung agar pelanggan dapat mengetahui tanggal yang tersedia tanpa perlu menunggu konfirmasi dari admin.
- 3) Otomatisasi proses pemesanan dan pembayaran, termasuk verifikasi bukti pembayaran secara digital.
- 4) Monitoring dan pelaporan yang memudahkan manajer dalam evaluasi operasional.
- 5) Fitur umpan balik pelanggan untuk meningkatkan kualitas layanan dan kepuasan pelanggan.

Analisis kebutuhan ini menjadi dasar perancangan sistem, yang mencakup pembuatan *use case diagram*, *activity diagram*, dan *sequence diagram*, serta perancangan basis data dan antarmuka pengguna. Setelah pemodelan BPMN untuk proses penyewaan koperasi yang sedang berjalan saat ini dan diusulkan, dilakukan simulasi untuk membandingkan kinerja kedua model.

Simulasi ini menggunakan parameter seperti waktu eksekusi aktivitas, probabilitas keputusan pada *gateway*, serta keterlibatan aktor (pelanggan, admin, dan manajer) dalam alur proses. Tujuan adalah untuk menilai kelancaran alur layanan, mengukur tingkat pemanfaatan sumber daya (*degree of utilisation*), dan membandingkan rata-rata waktu penyelesaian proses antara model berjalan dan model usulan. Hasil simulasi tersebut disajikan pada Tabel 4.1

Tabel 4.1 Hasil BPMN *Simulation* Penyewaan Koperasi

Elemen	Parameter	Hasil	
		Berjalan	Diusulkan
Admin	<i>Degree Of Utilisation</i>	63.34%	10.67%
Pelanggan	<i>Degree Of Utilisation</i>	87.88%	97.33%
Penyewaan Koperasi	<i>Average Time</i>	139	58.6
Penyewaan Koperasi	<i>Mean Of Processing Time</i>	50	16.2
Penyewaan Koperasi	<i>Maximum Time</i>	162	71
Penyewaan Koperasi	<i>Max Of Processing Time</i>	65	21
Penyewaan Koperasi	<i>Minimum Time</i>	119	32.5
Penyewaan Koperasi	<i>Min Of Processing Time</i>	40	8.5
Penyewaan Koperasi	<i>Total Time Busy</i>	250	81

Berdasarkan hasil simulasi pada Tabel 4.1, diperoleh beberapa temuan sebagai berikut:

1. Utilisasi Admin mengalami penurunan dari 63,34% menjadi 10,67%. Kondisi ini menunjukkan bahwa sebagian besar pekerjaan administrasi, seperti pencatatan transaksi, pembuatan laporan, dan pengecekan jadwal, telah ditangani oleh sistem. Admin tidak perlu lagi melakukan penginputan pada nota

atau Excel, cukup memverifikasi data yang tersimpan otomatis pada menu penyewaan, pembayaran, dan layanan.

2. Utilisasi Pelanggan meningkat dari 87,88% menjadi 97,33%. Peningkatan ini menandakan bahwa pelanggan menjadi lebih aktif dalam penggunaan sistem. Meskipun harus melakukan registrasi dan login terlebih dahulu, pelanggan dapat secara mandiri memesan layanan, melihat jadwal, melakukan pembayaran, serta memantau riwayat penyewaan tanpa menunggu konfirmasi langsung dari admin.
3. Peran Manajer mulai terlibat pada sistem usulan, sedangkan pada proses berjalan sebelumnya belum terdapat aktivitas manajer. *Dashboard* manajer memungkinkan pemantauan langsung terhadap status pemesanan, riwayat penyewaan, dan umpan balik pelanggan, sehingga proses pengawasan tidak lagi bergantung pada laporan manual dari admin.
4. Rata-rata waktu proses penyewaan (*Average Time*) berkurang dari 139 menit menjadi 58,6 menit. Penurunan waktu ini menggambarkan bahwa alur penyewaan berlangsung lebih cepat karena proses pemesanan dan konfirmasi kini dapat dilakukan secara *online* melalui sistem *web*.
5. Waktu rata-rata per aktivitas (*Mean of Processing Time*) turun dari 50 menit menjadi 16,2 menit. Setiap aktivitas, seperti validasi data dan pencatatan transaksi, dapat diselesaikan dengan durasi yang lebih singkat karena sistem sudah mengatur alur kerja secara otomatis.
6. Waktu maksimum (*Maximum Time*) menurun dari 162 menit menjadi 71 menit. Hal ini menunjukkan bahwa durasi terlama dalam keseluruhan proses penyewaan berhasil dipangkas melalui penerapan sistem digital yang terintegrasi.
7. Waktu minimum (*Minimum Time*) berkurang dari 119 menit menjadi 32,5 menit. Penurunan ini menandakan bahwa skenario tercepat dalam proses penyewaan menjadi jauh lebih singkat karena pelanggan dapat menyelesaikan pemesanan hingga konfirmasi secara langsung melalui sistem.

8. Durasi terlama per aktivitas (*Max Of Processing Time*) menurun dari 65 menit menjadi 21 menit, yang menunjukkan bahwa aktivitas dengan beban proses paling tinggi kini dapat diselesaikan dalam waktu yang lebih terkendali.
9. Durasi tercepat per aktivitas (*Min Of Processing Time*) turun dari 40 menit menjadi 8,5 menit. Beberapa aktivitas inti, seperti pemesanan dan validasi, dapat diselesaikan jauh lebih cepat dibandingkan sebelumnya.
10. Total waktu sibuk (*Total Time Busy*) menurun dari 250 menit menjadi 81 menit. Perubahan ini memperlihatkan bahwa pembagian waktu proses menjadi lebih seimbang antara admin, pelanggan, dan sistem, sehingga tidak terjadi penumpukan beban kerja pada satu pihak tertentu.

Dari hasil simulasi diatas, sistem informasi layanan penyewaan yang diusulkan memberikan sejumlah perubahan yang dapat diamati sebagai berikut:

1. Proses penyewaan menjadi lebih cepat dan teratur. Waktu penyelesaian berkurang dari 139 menit menjadi 58,6 menit karena pelanggan dapat melakukan pemesanan dan konfirmasi tanpa harus datang langsung ke koperasi.
2. Penyimpanan dan pengelolaan data lebih terstruktur. Seluruh data pemesanan, pembayaran, jadwal, dan layanan tersimpan dalam basis data MySQL melalui model penyewaan, pembayaran, dan layanan, menggantikan pencatatan manual dengan nota atau Excel.
3. Pelanggan memperoleh kemudahan akses. Melalui sistem, pelanggan dapat melakukan pendaftaran, login, melihat jadwal, memesan layanan, serta mengunggah bukti pembayaran dengan lebih praktis dan transparan.
4. Beban kerja admin berkurang. Penurunan utilisasi admin mencerminkan bahwa sistem telah menangani sebagian besar aktivitas pencatatan dan pelaporan, sementara admin berfokus pada verifikasi serta pengawasan data.
5. Keterlibatan pelanggan meningkat. Peningkatan utilisasi pelanggan menunjukkan bahwa pengguna tidak lagi pasif menunggu admin, melainkan berinteraksi langsung dengan sistem untuk melakukan transaksi.

6. Pengawasan manajer menjadi lebih mudah. *Dashboard* yang disediakan memungkinkan manajer memantau aktivitas penyewaan, status transaksi, serta hasil umpan balik pelanggan secara langsung.
7. Konsistensi data lebih terjamin. Proses pencatatan dan validasi dilakukan secara langsung oleh sistem, sehingga meminimalkan kesalahan input dan kehilangan data yang sebelumnya sering terjadi pada pencatatan manual.

4.2.3 Kebutuhan fungsional

Tahap ini menjelaskan fungsi-fungsi yang harus tersedia pada sistem untuk mendukung seluruh aktivitas pengguna sesuai dengan peran masing-masing. Fungsi tersebut mencakup proses pembuatan akun, login, pemesanan layanan, pengecekan jadwal, pembayaran, pengunggahan bukti transaksi, serta pengelolaan data dan laporan oleh pihak koperasi.

Analisis kebutuhan fungsional diperoleh melalui wawancara dengan pihak koperasi, penelaahan dokumen terkait, yang menggambarkan alur proses bisnis penyewaan yang sedang berjalan. Berdasarkan hasil analisis tersebut, sistem informasi layanan penyewaan ini melibatkan tiga peran utama, yaitu Admin Koperasi, Manajer Koperasi, dan Pelanggan, dengan kebutuhan sebagai berikut:

1. Kebutuhan Fungsional Admin Koperasi

- a. *Login* sebagai admin
- b. *Logout* sebagai admin
- c. Mengubah profil akun
- d. Mengelola data layanan
- e. Mengatur jadwal libur layanan
- f. Memverifikasi dan mengkonfirmasi pembayaran pelanggan.
- g. Memverifikasi Request refund
- h. Mengelola data pengguna (manajer dan pelanggan)
- i. Mengelola data umpan balik pelanggan
- j. Melihat riwayat transaksi penyewaan

2. Kebutuhan Fungsional Manajer Koperasi

- a. *Login* sebagai manajer

- b. *Logout* sebagai manajer
 - c. Mengubah profil akun
 - d. Melihat *dashboard*
 - e. Melihat dan mengunduh laporan riwayat penyewaan pelanggan
 - f. Meninjau umpan balik pelanggan sebagai bahan evaluasi
3. Kebutuhan Fungsional Pelanggan
- a. Melakukan registrasi akun
 - b. *Login* sebagai pelanggan terdaftar.
 - c. *Logout* sebagai pelanggan
 - d. Mengubah profil akun
 - e. Mengakses chatbot untuk memperoleh informasi layanan
 - f. Melihat daftar layanan dan melakukan pemesanan.
 - g. Melakukan pembayaran serta unggah bukti pembayaran
 - h. Memberikan umpan balik terhadap layanan yang diterima
 - i. Melihat riwayat penyewaan

4.2.4 Use Case Diagram

Berdasarkan hasil analisis kebutuhan fungsional, disusun *use case diagram* untuk menggambarkan interaksi antara aktor dengan sistem. Diagram ini memvisualisasikan fungsi utama yang dapat diakses oleh masing masing aktor, yaitu admin, pelanggan, dan manajer serta menunjukkan bagaimana sistem mendukung aktivitas mereka dalam proses penyewaan. *Use case diagram* sistem informasi penyewaan koperasi dapat dilihat pada Gambar 4.3.



Gambar 4.3 Use Case Diagram

Use case diagram sistem informasi penyewaan koperasi dapat dilihat pada Gambar 4.3. Diagram tersebut memberikan gambaran visual mengenai hubungan antara aktor dan fungsi-fungsi utama sistem berbasis *web* yang telah dijabarkan pada sub bab sebelumnya.

4.2.5 Deskripsi Tugas Aktor

Setelah menampilkan *use case diagram* yang memperlihatkan interaksi antara aktor dan sistem pada sub bab sebelumnya, bagian ini menjelaskan secara lebih rinci tugas serta tanggung jawab masing-masing aktor. Penjelasan ini memberikan

gambaran mengenai hak akses dan aktivitas yang dapat dilakukan oleh setiap aktor dalam sistem informasi penyewaan berbasis *web*. Rincian tugas aktor disajikan pada Tabel 4.2 berikut.

Tabel 4.2 Deskripsi Tugas Aktor

Nama Aktor	Deskripsi Tugas
Admin	Login sebagai admin Admin dapat mengakses sistem dengan memasukkan email dan password yang telah terdaftar. Sistem akan memverifikasi kredensial dan memberikan akses ke <i>dashboard</i> admin dengan hak akses penuh untuk mengelola seluruh sistem
	Logout sebagai admin Admin dapat keluar dari sistem, mengakhiri sesi login
	Mengedit profil Admin dapat memperbarui data pribadi seperti nama, alamat, kata sandi, dan foto profil
	Mengelola layanan Admin bisa menambah, mengedit, dan mengatur status layanan yang tersedia di sistem. Termasuk mengatur harga, deskripsi, gambar, dan ketersediaan fasilitas yang dapat disewa oleh pelanggan
	Mengelola jadwal libur Admin dapat mengatur jadwal libur operasional koperasi, menentukan tanggal-tanggal tertentu layanan tidak tersedia
	Memverifikasi dan mengkonfirmasi pembayaran pelanggan. Admin memeriksa bukti pembayaran yang diunggah pelanggan. Setelah admin menyatakan valid, sistem secara otomatis memperbarui status transaksi menjadi terkonfirmasi (DP atau lunas)

Nama Aktor	Deskripsi Tugas
	Memverifikasi Refund Admin dapat memproses permintaan refund dari pelanggan, meninjau alasan pengembalian dana, menghitung nominal dan mengubah status refund menjadi "approved" berdasarkan kebijakan koperasi
	Mengelola akun (manajer dan pelanggan) Admin dapat membuat, mengedit, menonaktifkan, reset password dan memperpanjang sesuai periode untuk akun manajer bisa akses dan menonaktifkan dan aktifkan kembali pelanggan
	Melihat dan membalas umpan balik Admin dapat melihat, menanggapi, dan membalas umpan balik yang diberikan pelanggan. Memberikan respon terhadap keluhan, dan menggunakan umpan balik untuk perbaikan layanan
	Melihat riwayat penyewaan. Admin dapat mengakses dan melihat seluruh riwayat transaksi penyewaan yang telah dilakukan, termasuk detail pembayaran, status penyewaan, dan informasi pelanggan untuk keperluan audit dan pelaporan
Pelanggan	Register Pelanggan baru dapat mendaftar akun dengan mengisi formulir pendaftaran yang mencakup nama lengkap, email, password, alamat, jenis kelamin dan informasi kontak. Sistem akan memverifikasi email dan mengirim konfirmasi pendaftaran
	Login sebagai pelanggan terdaftar.

Nama Aktor	Deskripsi Tugas
	Pelanggan yang sudah terdaftar dapat masuk ke sistem menggunakan email dan password yang telah didaftarkan, mendapatkan akses ke fitur-fitur yang tersedia untuk pelanggan
	<i>Logout</i> sebagai pelanggan Pelanggan dapat keluar dari sistem, mengakhiri sesi login
	Mengedit profil Pelanggan dapat mengubah informasi profil pribadi seperti nama, password, alamat, dan nomor telepon. Perubahan ini akan tersimpan
	Menggunakan <i>chatbot</i> untuk mendapatkan informasi - Pelanggan yang belum memiliki akun dapat memanfaatkan <i>chatbot</i> sebagai media interaktif untuk memperoleh informasi dasar terkait layanan, prosedur pendaftaran, serta berbagai pertanyaan umum tanpa perlu melakukan proses registrasi terlebih dahulu. - Pelanggan yang telah terdaftar dapat menggunakan <i>chatbot</i> untuk mengakses informasi yang bersifat personal, seperti status pemesanan, riwayat pembayaran, serta detail layanan yang telah dipesan, dengan autentikasi melalui akun yang sudah login.
	Melihat daftar layanan dan melakukan pemesanan Mengakses katalog layanan, memilih jadwal, menentukan fasilitas tambahan (opsional), dan melakukan pemesanan
	Melakukan pembayaran dan unggah bukti pembayaran Mengunggah bukti transfer dan menunggu verifikasi dari admin
	Memberikan umpan balik

Nama Aktor	Deskripsi Tugas
	Menyampaikan ulasan dan penilaian setelah menggunakan layanan
	Melihat riwayat penyewaan Meninjau seluruh transaksi, status pembayaran, dan detail layanan yang pernah digunakan
Manajer	Login sebagai manajer Manajer dapat mengakses sistem dengan kredensial yang telah diberikan admin, mendapatkan akses ke <i>dashboard</i> manajer dengan hak akses terbatas sesuai dengan peran dan tanggung jawabnya
	Logout sebagai manajer Manajer dapat keluar dari sistem, mengakhiri sesi login
	Mengedit Profil Manajer dapat mengubah informasi profil pribadi seperti nama, password, dan informasi kontak lainnya sesuai dengan data yang terdaftar dalam sistem
	Melihat <i>dashboard</i> . Manajer dapat mengakses <i>dashboard</i> yang menampilkan ringkasan data penting seperti jumlah penyewaan, pendapatan, dan statistik lainnya untuk monitoring kinerja operasional
	Melihat riwayat penyewaan pelanggan dan download laporan Manajer dapat melihat dan menelusuri riwayat penyewaan yang dilakukan pelanggan, melakukan filter berdasarkan periode, layanan, atau status pembayaran, serta mengunduh laporan dalam format yang dapat digunakan untuk analisis dan pelaporan
	Meninjau umpan balik sebagai bahan evaluasi

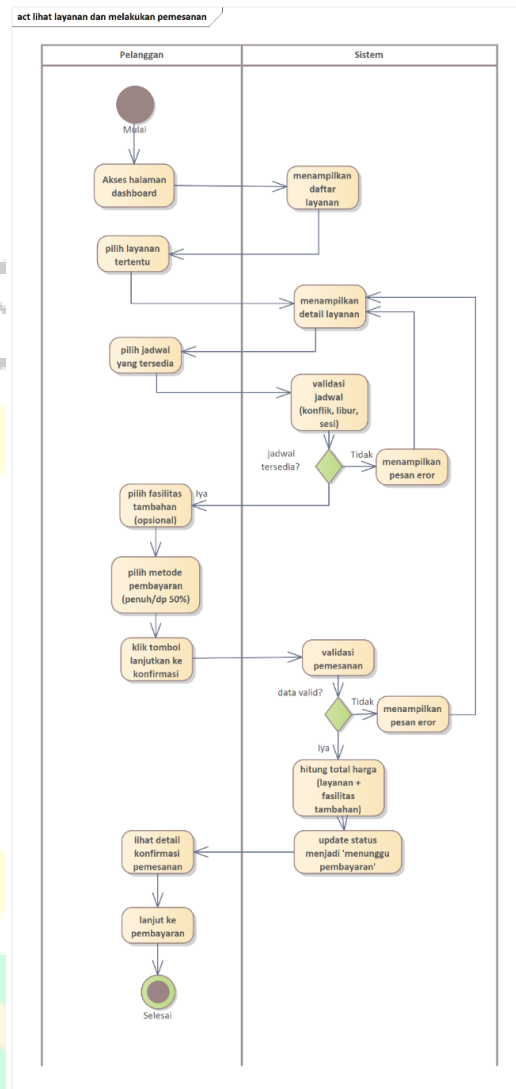
Nama Aktor	Deskripsi Tugas
	Manajer dapat melihat umpan balik Meninjau ulasan pelanggan sebagai bahan evaluasi dan pengambilan keputusan operasional

4.2.6 Activity Diagram

Activity diagram digunakan untuk memvisualisasikan alur aktivitas dalam sistem, termasuk interaksi antara admin, pelanggan, dan manajer. Diagram ini menggambarkan urutan proses mulai dari registrasi, *login*, pemesanan, pembayaran, hingga pengelolaan data. Pada subbab ini ditampilkan contoh *activity diagram* untuk proses pelanggan melihat layanan dan melakukan pemesanan, serta pelanggan melakukan pembayaran dan unggah bukti pembayaran. Diagram lengkap untuk keseluruhan proses sistem disajikan pada **Lampiran A**.

4.2.6.1 Pelanggan Melihat Layanan Dan Melakukan Pemesanan

Setelah berhasil *login*, pelanggan mengakses *dashboard* untuk melihat daftar layanan yang tersedia. Dari halaman tersebut, pelanggan memilih layanan sesuai kebutuhannya dan melanjutkan ke tahap pemesanan sebagaimana ditunjukkan pada Gambar 4.4.



Gambar 4.4 Activity Diagram lihat layanan dan melakukan pemesanan

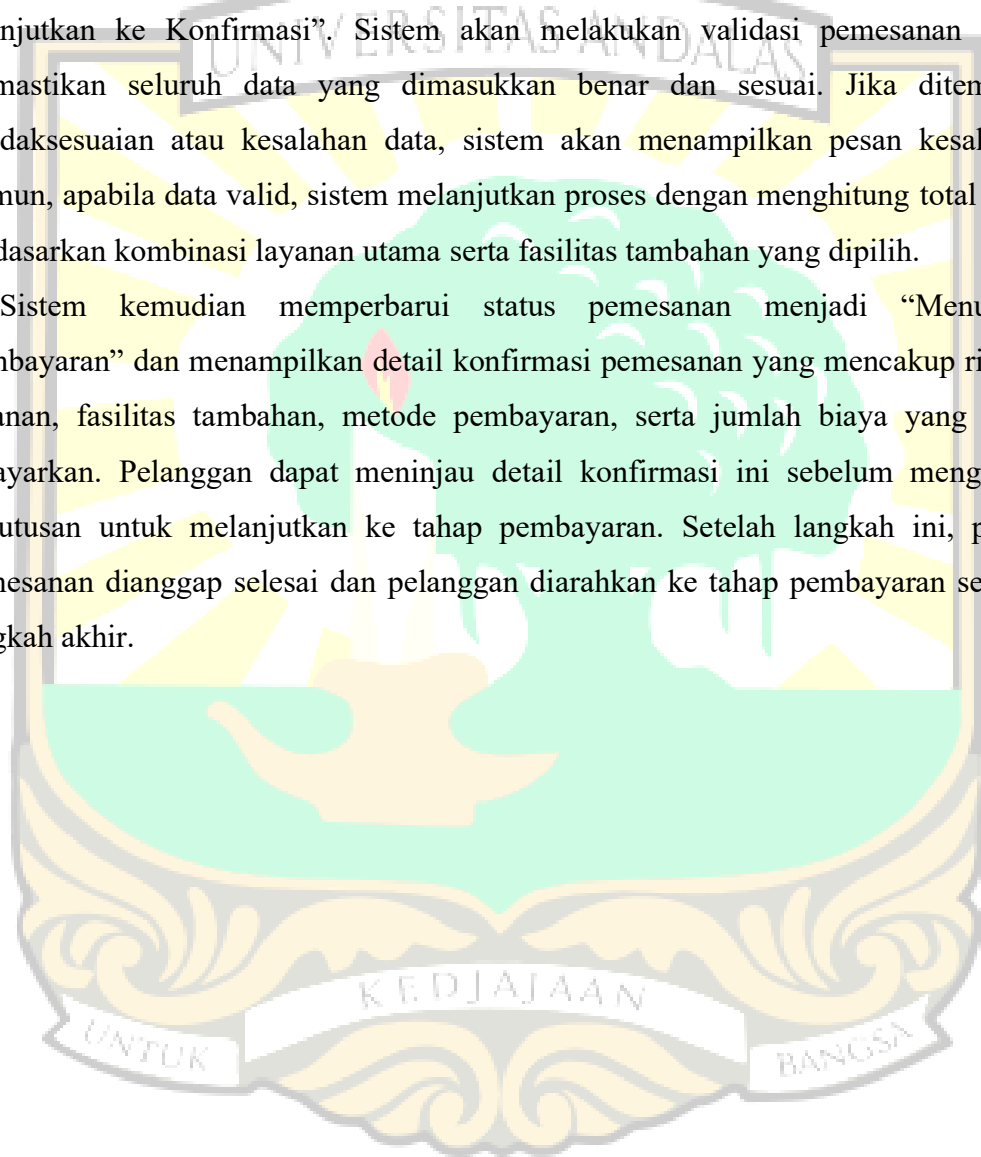
Pada Gambar 4.4, alur dimulai ketika pelanggan membuka halaman *dashboard*. Setelah halaman terbuka, sistem secara otomatis menampilkan daftar layanan yang tersedia. Pelanggan kemudian memilih salah satu layanan yang diinginkan, dan sistem akan menampilkan detail informasi salah satu layanan, dan sistem menampilkan detail informasi mencakup deskripsi layanan, kisaran harga, serta jadwal tersedia.

Tahap berikutnya, pelanggan menentukan jadwal layanan. Sistem memvalidasi jadwal dengan memeriksa potensi konflik jadwal, memastikan jadwal tidak jatuh pada hari libur, serta mengecek apakah sesi layanan masih tersedia. Jika jadwal yang dipilih tidak valid, sistem menampilkan pesan kesalahan yang menginformasikan pelanggan

untuk memilih jadwal alternatif. Apabila jadwal valid, pelanggan dapat melanjutkan proses dengan memilih fasilitas tambahan (jika diperlukan) sebagai opsi pelengkap terhadap layanan utama.

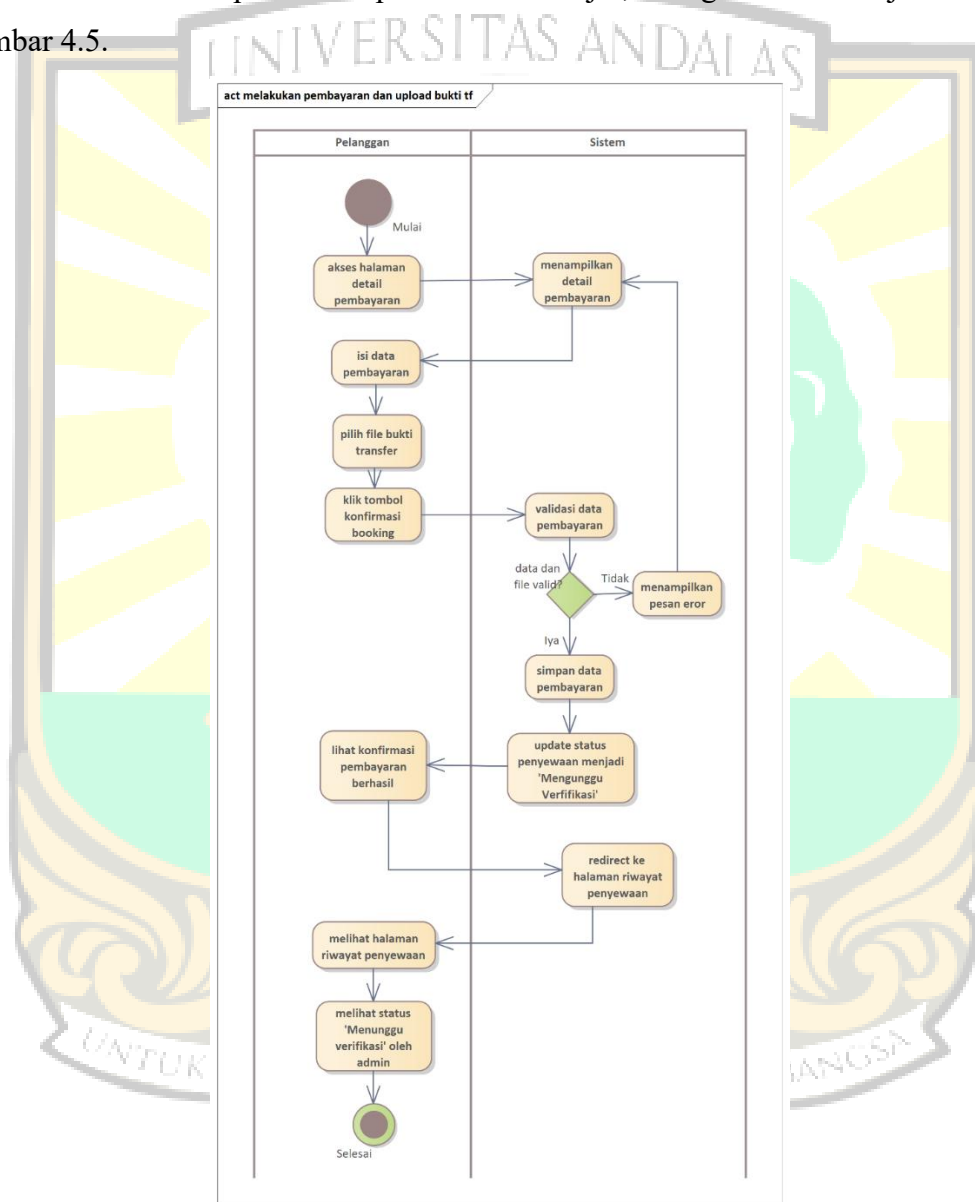
Selanjutnya, pelanggan memilih metode pembayaran, baik pelunasan penuh maupun pembayaran DP (Down Payment) sebesar 50%, lalu menekan tombol “Lanjutkan ke Konfirmasi”. Sistem akan melakukan validasi pemesanan untuk memastikan seluruh data yang dimasukkan benar dan sesuai. Jika ditemukan ketidaksesuaian atau kesalahan data, sistem akan menampilkan pesan kesalahan. Namun, apabila data valid, sistem melanjutkan proses dengan menghitung total biaya berdasarkan kombinasi layanan utama serta fasilitas tambahan yang dipilih.

Sistem kemudian memperbarui status pemesanan menjadi “Menunggu Pembayaran” dan menampilkan detail konfirmasi pemesanan yang mencakup rincian layanan, fasilitas tambahan, metode pembayaran, serta jumlah biaya yang harus dibayarkan. Pelanggan dapat meninjau detail konfirmasi ini sebelum mengambil keputusan untuk melanjutkan ke tahap pembayaran. Setelah langkah ini, proses pemesanan dianggap selesai dan pelanggan diarahkan ke tahap pembayaran sebagai langkah akhir.



4.2.6.2 Pelanggan Melakukan Pembayaran dan Unggah Bukti Pembayaran

Setelah pemesanan berhasil, pelanggan diarahkan ke halaman pembayaran. Pada tahap ini, pelanggan melakukan transfer sesuai nominal yang tercantum, lalu mengunggah bukti pembayaran sebagai konfirmasi. Bukti tersebut akan diverifikasi oleh admin sebelum pesanan diproses lebih lanjut, sebagaimana ditunjukkan pada Gambar 4.5.



Gambar 4.5 Activity Diagram Pelanggan Melakukan Pembayaran dan unggah bukti pembayaran

Pada Gambar 4.5, alur dimulai ketika pelanggan membuka halaman detail pembayaran. Sistem menyajikan informasi detail terkait pembayaran yang perlu dilakukan. Pelanggan kemudian mengisi data pembayaran sesuai instruksi yang tersedia. Setelah itu, pelanggan memilih file bukti transfer yang akan diunggah sebagai konfirmasi pembayaran. Bukti tersebut akan diverifikasi oleh admin sebelum pesanan diproses lebih lanjut.

Alur dimulai ketika pelanggan membuka halaman detail pembayaran. Sistem menyajikan informasi pembayaran yang perlu dilakukan. Pelanggan mengisi data sesuai instruksi dan mengunggah file bukti transfer. Sistem kemudian memvalidasi data dan file yang diunggah. Jika tidak valid, sistem menampilkan pesan kesalahan; jika valid, data pembayaran disimpan dalam basis data dan status penyewaan diperbarui menjadi “Menunggu Verifikasi”.

Setelah proses berhasil, sistem menampilkan konfirmasi bahwa pembayaran telah tersimpan dengan benar dan mengarahkan pelanggan ke halaman riwayat penyewaan untuk memantau status transaksi. Status awal ditampilkan sebagai Menunggu Verifikasi, yang menandakan bahwa pembayaran sedang diperiksa oleh admin.

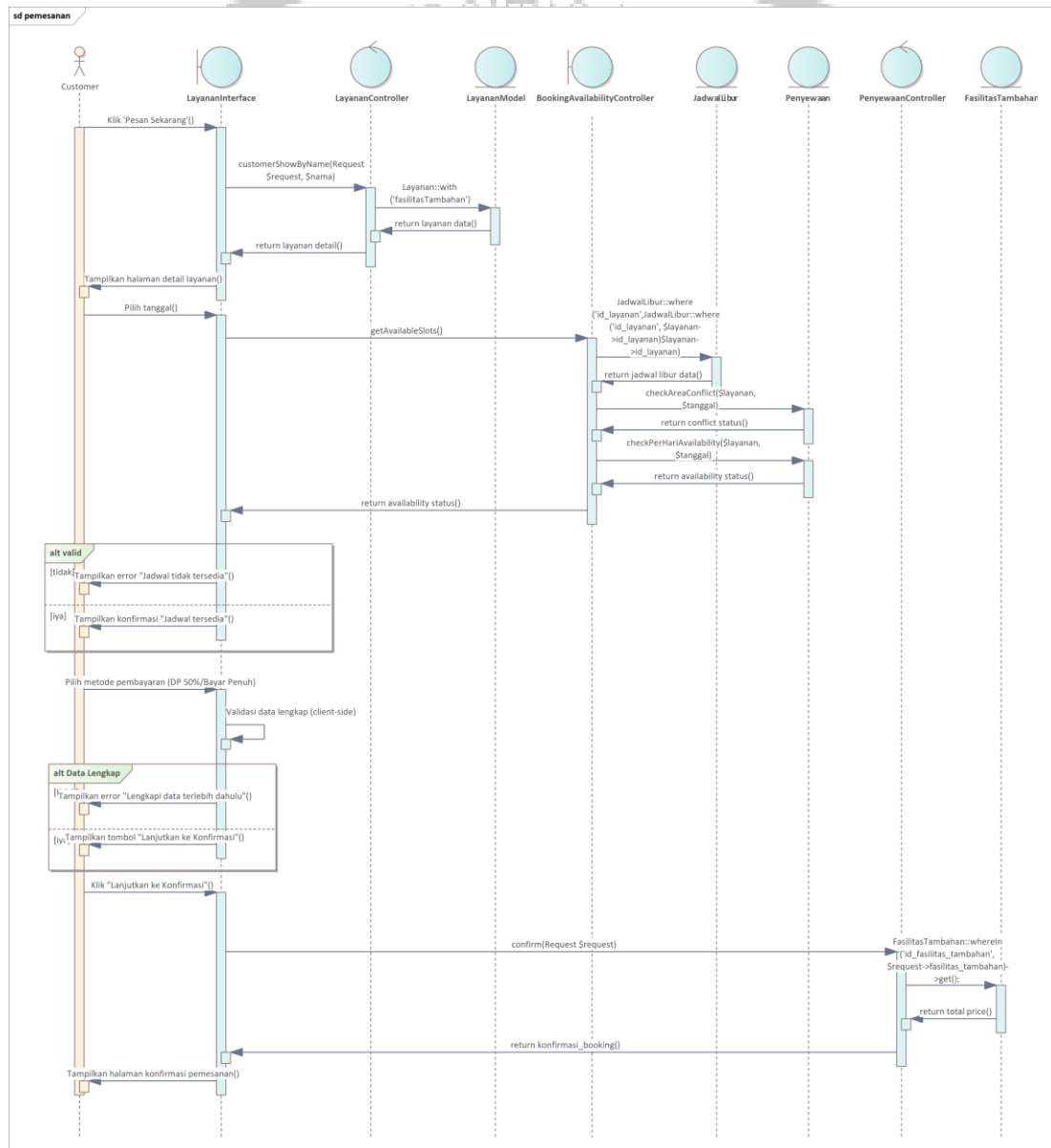
Dengan demikian, *activity diagram* membantu memvisualisasikan logika sistem secara menyeluruh, sehingga memudahkan pemahaman terhadap alur interaksi antar aktor dan sistem sesuai kebutuhan fungsional yang telah ditetapkan.

4.2.7 Sequence Diagram

Setelah alur aktivitas sistem dijelaskan melalui *activity diagram*, bagian ini menampilkan *sequence diagram* yang memperlihatkan urutan interaksi antar objek dalam sistem berdasarkan waktu. Diagram ini menunjukkan bagaimana admin, pelanggan, dan manajer berinteraksi dengan sistem mulai dari registrasi, *login*, pemesanan, pembayaran, hingga pengelolaan data. Subbab ini menampilkan contoh *sequence diagram* untuk proses pelanggan melihat layanan dan melakukan pemesanan, serta proses pembayaran dan unggah bukti pembayaran. Diagram lengkap untuk seluruh proses sistem dapat dilihat pada **Lampiran B**.

4.2.7.1 Pelanggan Lihat Layanan Dan Melakukan Pemesanan

Setelah berhasil login, pelanggan dapat mengakses *dashboard* untuk melihat daftar layanan yang tersedia. Dari *dashboard* tersebut, pelanggan memilih layanan sesuai kebutuhan dan melanjutkan ke tahap pemesanan dan konfirmasi. Alur interaksi ini divisualisasikan dalam *sequence diagram* pada Gambar 4.6.



Gambar 4.6 Sequence Diagram Pelanggan lihat layanan dan melakukan pemesanan

Pada Gambar 4.6, memperlihatkan interaksi antara aktor pelanggan (*customer*) dengan komponen sistem saat melakukan pemesanan layanan. Proses dimulai ketika

pelanggan menekan tombol "Pesan Sekarang", kemudian sistem mengarahkan ke halaman detail layanan melalui `LayananInterface`. Permintaan ini diteruskan ke `LayananController` untuk mengambil data dari `LayananModel`, termasuk relasi dengan `fasilitasTambahan`. Data yang berhasil diperoleh kemudian ditampilkan kembali pada halaman detail layanan.

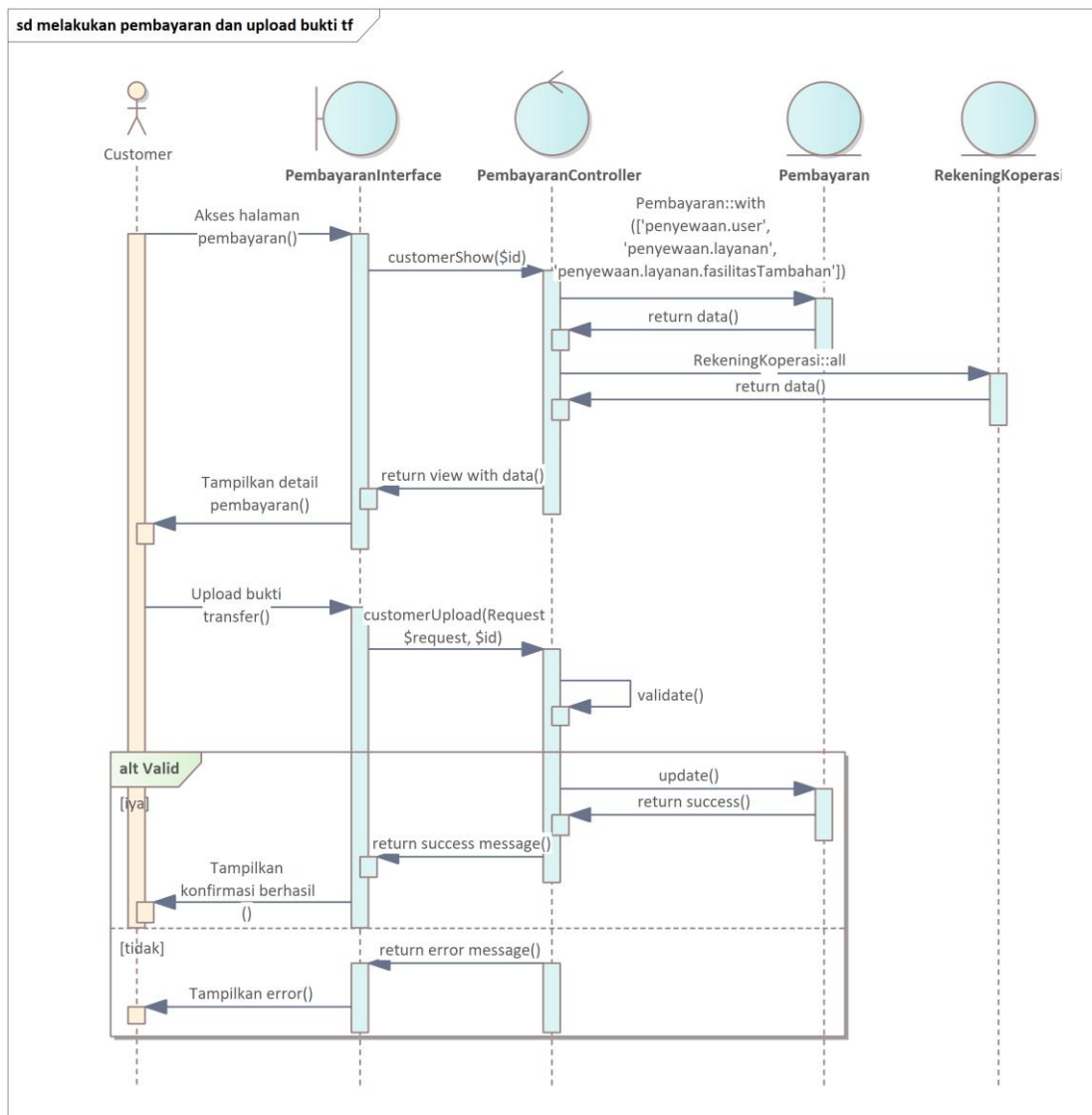
Selanjutnya pelanggan memilih tanggal layanan. Permintaan ini diproses oleh `BookingAvailabilityController` yang berinteraksi dengan `JadwalLiburModel` untuk memeriksa jadwal libur dan `PenyewaanModel` untuk mengecek konflik jadwal serta ketersediaan tanggal. Hasil dari pengecekan tersebut akan menentukan status ketersediaan layanan. Jika jadwal tidak tersedia, sistem menampilkan pesan kesalahan, sedangkan jika tersedia, sistem menampilkan pesan konfirmasi.

Berikutnya, pelanggan memilih metode pembayaran, yaitu *Down Payment* (DP) sebesar 50% atau pembayaran penuh. Pada tahap ini dilakukan validasi kelengkapan data secara *client-side*. Jika data belum lengkap, sistem menampilkan pesan kesalahan agar pelanggan melengkapi data terlebih dahulu. Apabila data sudah lengkap, sistem menampilkan tombol "Lanjutkan ke Konfirmasi".

Ketika pelanggan menekan tombol konfirmasi, permintaan diproses melalui `PenyewaanController` yang menghitung total harga dengan memanggil `FasilitasTambahanModel` untuk menjumlahkan biaya fasilitas tambahan yang dipilih. Hasil perhitungan digunakan untuk menyiapkan data konfirmasi pemesanan dan sistem kemudian menampilkan halaman konfirmasi berisi detail layanan, fasilitas tambahan, jadwal, serta total biaya.

4.2.7.2 Pelanggan Melakukan Pembayaran dan Unggah Bukti Pembayaran

Setelah konfirmasi pemesanan, sistem akan menampilkan instruksi unggah bukti pembayaran. Pelanggan kemudian melakukan pembayaran sesuai nominal yang ditentukan dan mengunggah bukti pembayaran melalui sistem. Proses interaksi ini divisualisasikan dalam *sequence diagram* yang dapat dilihat pada Gambar 4.7.



Gambar 4.7 Sequence Diagram Pelanggan Melakukan Pembayaran dan Unggah Bukti Pembayaran

Pada Gambar 4.7. menggambarkan alur interaksi antara pelanggan dan sistem proses pembayaran. Sistem melalui PembayaranController mengambil informasi detail pembayaran dari PembayaranModel, data rekening dari RekeningModel, serta data penyewaan dari PenyewaanModel. Seluruh informasi tersebut dikirim ke PembayaranInterface untuk ditampilkan kepada pelanggan.

Pelanggan kemudian melakukan transfer sesuai instruksi dan mengunggah bukti pembayaran. Data yang diunggah akan divalidasi oleh PembayaranController.

Jika ditemukan kesalahan, sistem menampilkan pesan peringatan agar pelanggan memperbaikinya. Jika valid, data pembayaran disimpan melalui PembayaranModel dan pelanggan menerima notifikasi bahwa proses unggah berhasil.

Tahap berikutnya, pelanggan menekan tombol “Konfirmasi Pembayaran”. Sistem kemudian memperbarui status pembayaran melalui PenyewaanModel mengarahkan pelanggan ke halaman riwayat penyewaan. Di halaman tersebut, pelanggan dapat melihat status transaksi terbaru, misalnya “Menunggu Verifikasi” atau status lain yang relevan. Dengan demikian, *sequence diagram* ini menggambarkan proses lengkap mulai dari tampilan detail pembayaran, unggah bukti transfer, validasi sistem, pembaruan data, hingga konfirmasi status pembayaran pada riwayat penyewaan.

4.3 Perancangan Sistem

Tahap ini membahas rancangan teknis dari sistem yang dikembangkan, meliputi basis data, struktur tabel, hubungan antar tabel, *class diagram*, serta rancangan antarmuka pengguna (*user interface*). Perancangan dilakukan untuk memastikan seluruh kebutuhan fungsional sistem dapat dipenuhi secara terstruktur dan sesuai dengan tujuan pengembangan.

4.3.1 Perancangan Database

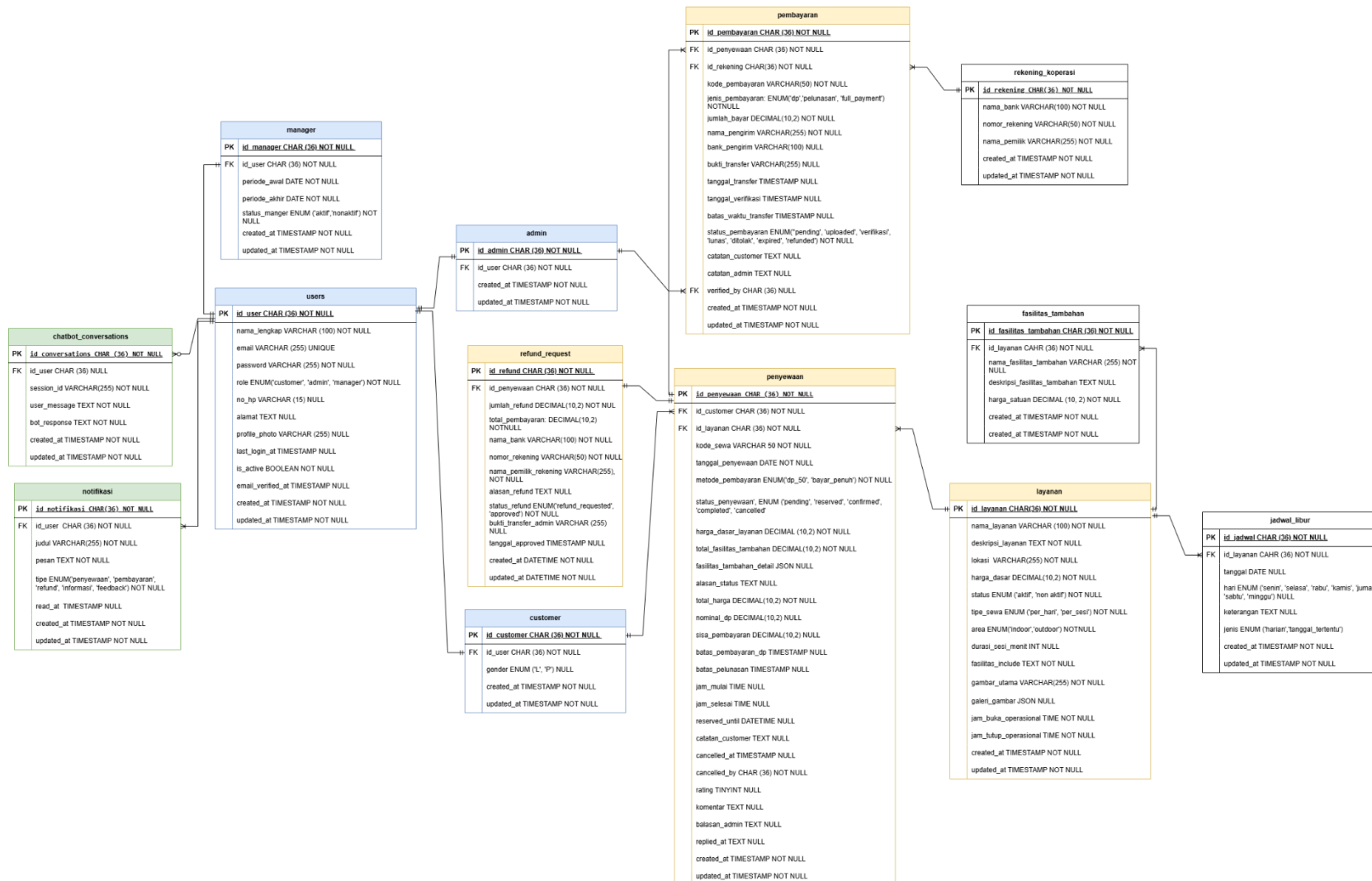
Entity Relationship Diagram (ERD) digunakan untuk memvisualisasikan struktur data serta hubungan antar entitas dalam Sistem Informasi Jasa Penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi. ERD ini dirancang untuk memastikan integrasi data yang konsisten antara bagian penyewaan, pembayaran, pelaporan, dan manajemen pengguna. Entitas pada sistem dibagi menjadi empat kelompok utama, yaitu:

1. Tabel inti, yang mencakup *user*, *customer*, *admin*, dan *manajer*.
2. Tabel bisnis yang langsung berkaitan dengan transaksi dan pendapatan meliputi *layanan*, *penyewaan*, *pembayaran*, dan *refund_requests*.
3. Tabel pendukung yang berfungsi menjaga kelancaran alur operasional, seperti *rekening_koperasi*, *fasilitas_tambahan*, dan *jadwal_layanan*.

4. Tabel sistem yang berperan dalam meningkatkan pengalaman pengguna, terdiri dari *notifications*, dan *chatbot_conversations*.

Dengan struktur tersebut, ERD mampu menggambarkan keterkaitan antar entitas secara menyeluruh, sekaligus menjadi dasar dalam perancangan *database* sistem. ERD penyewaan ditampilkan pada Gambar 4.8.





Gambar 4.8 ERD

4.3.2 Struktur Tabel dan Basis Data

Tahap selanjutnya setelah perancangan ERD adalah mendefinisikan struktur tabel dalam basis data. Perancangan tabel mencakup penentuan nama tabel, atribut, tipe data yang digunakan, *primary key*, *foreign key*, serta relasi antar tabel. Struktur setiap tabel dijabarkan melalui nama atribut, tipe data, dan constraint yang digunakan. Pada subbab ini ditampilkan tiga tabel utama yang menjadi pusat alur data sistem, yaitu tabel penyewaan, layanan, dan pembayaran. Penjelasan lebih rinci mengenai struktur tabel lainnya dapat dilihat secara lengkap pada **Lampiran C**.

4.3.2.1 Tabel Layanan

Tabel *layanan* digunakan untuk menyimpan data terkait setiap layanan penyewaan yang ditawarkan oleh koperasi. Struktur tabel tersebut dapat dilihat pada Tabel 4.3.

Tabel 4.3 Tabel Layanan

Nama Atribut	Tipe Data	Constraint
id_layanan	CHAR (36)	PK, NOT NULL
nama_layanan	VARCHAR (100)	NOT NULL
deskripsi_layanan	TEXT	NOT NULL
lokasi	TEXT	NOT NULL
harga_dasar	DECIMAL (10,2)	NOT NULL
tipe_sewa	Enum ('per_hari', 'per_sesi')	NOT NULL, DEFAULT 'per_hari'
area	ENUM('outdoor','indoor')	NOT NULL, DEFAULT 'indoor'
durasi_sesi_menit	int unsigned	NULL
fasilitas_include	TEXT	NOT NULL
gambar_utama	VARCHAR (255)	NOT NULL
galeri_gambar	JSON	NULL
jam_buka_operasional	TIME	NOT NULL
jam_tutup_operasional	TIME	NOT NULL

Nama Atribut	Tipe Data	Constraint
status	Enum ('aktif', 'nonaktif')	NOT NULL, DEFAULT 'aktif'
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

4.3.2.2 Tabel Penyewaan

Tabel *penyewaan* digunakan untuk menyimpan data hasil pemesanan layanan yang dilakukan oleh pelanggan. Struktur tabel tersebut disajikan pada Tabel 4.4.

Tabel 4.4 Tabel Penyewaan

Nama Atribut	Tipe Data	Keterangan
id_penyewaan	CHAR (36)	PK, NOT NULL
kode_sewa	VARCHAR (30)	Unique, NOT NULL
id_user	CHAR (36)	FK, NOT NULL
id_layanan	CHAR (36)	FK, NOT NULL
tanggal_penyewaan	DATE	NULL
jam_mulai	TIME	NULL
jam_selesai	TIME	NULL
metode_pembayaran	Enum ('dp_50', 'bayar_penuh')	NOT NULL
harga_dasar_layanan	DECIMAL (10,2)	NOT NULL
total_fasilitas_tambahan	DECIMAL (10,2)	NOT NULL
total_harga	DECIMAL (10,2)	NOT NULL
nominal_dp	DECIMAL (10,2)	NOT NULL
sisapembayaran	DECIMAL (10,2)	NOT NULL
batas_pembayaran_dp	DATE	NULL
batas_pelunasan	DATE	NULL
total_dibayar	DECIMAL(10,2)	DEFAULT 0

Nama Atribut	Tipe Data	Keterangan
status_penyewaan	Enum ('pending', 'reserved', 'confirmed', 'completed', 'cancelled')	NOT NULL
alasan_status	TEXT	NULL
reserved_until	TIMESTAMP	NULL
cancelled_at	TIMESTAMP	NOT NULL
cancelled_by	CHAR (36)	FK (users), NULL
rating	TINYINT	NULL
komentar	TEXT	NULL
balasan_admin	TEXT	NULL
replied_at	DATETIME	NULL
catatan_customer	TEXT	NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

4.3.2.3 Tabel Pembayaran

Tabel *pembayaran* digunakan untuk menyimpan informasi mengenai transaksi pembayaran yang dilakukan pelanggan atas penyewaan layanan. Struktur tabel tersebut dapat dilihat pada Tabel 4.5.

Tabel 4.5 Tabel Pembayaran

Nama Atribut	Tipe Data	Keterangan
id_pembayaran	CHAR (36)	PK, NOT NULL
id_penyewaan	CHAR (36)	FK, NOT NULL
id_rekening	CHAR (36)	FK, NOT NULL
kode_pembayaran	VARCHAR(30)	Unique, NOT NULL
jenis_pembayaran	Enum('dp', 'pelunasan', 'penuh')	NOT NULL
jumlah_bayar	DECIMAL(10,2)	NOT NULL

Nama Atribut	Tipe Data	Keterangan
nama_pengirim	VARCHAR(100)	NOT NULL
bank_pengirim	VARCHAR(50)	NULL
bukti_transfer	VARCHAR(255)	NULL
tanggal_transfer	DATETIME	NULL
tanggal_verifikasi	DATETIME	NULL
batas_waktu_transfer	DATETIME	NULL
status_pembayaran	ENUM ('pending', 'uploaded', 'lunas', 'ditolak', 'refunded')	NOT NULL
catatan_customer	TEXT	NULL
catatan_admin	TEXT	NULL
verified_by	CHAR (36)	FK (admin), NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

4.3.3 Class Diagram

Class diagram digunakan untuk memodelkan struktur sistem melalui kumpulan kelas yang berisi atribut, metode, serta relasi antar kelas. Diagram ini merepresentasikan rancangan logis dari komponen perangkat lunak yang nantinya akan diimplementasikan dalam kode program. Sistem Informasi Penyewaan di Koperasi Konsumen Hydro Utama Bukittinggi dirancang dengan arsitektur berbasis *role*, di mana *user* menjadi entitas autentikasi utama yang dapat memiliki peran sebagai pelanggan, admin, atau manajer.

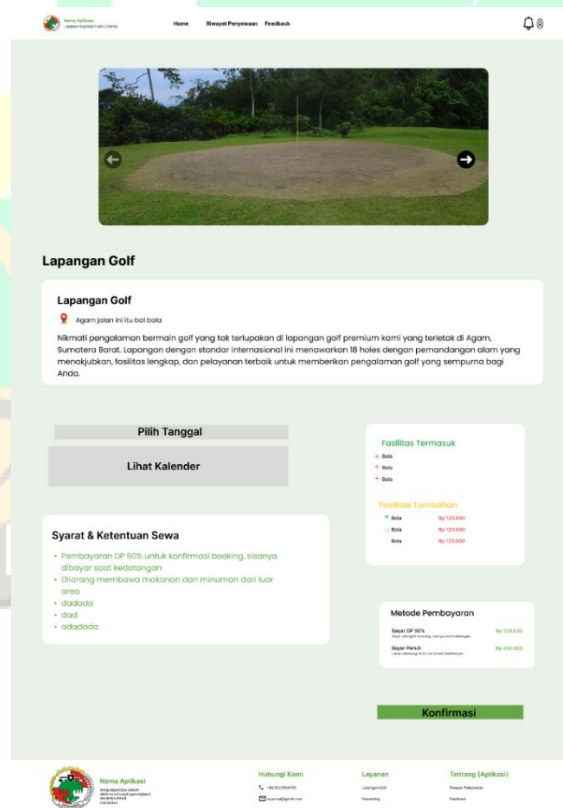
Setiap peran memiliki fungsi dan interaksi berbeda sesuai tanggung jawab dalam sistem. Selain itu, entitas *penyewaan* berperan sebagai pusat aktivitas bisnis yang menghubungkan proses pemesanan layanan, pembayaran, *refund*. Dengan demikian, *class diagram* menjadi acuan untuk memastikan rancangan sistem sesuai dengan kebutuhan fungsional dan tujuan pengembangan. Diagram tersebut dapat dilihat pada Gambar 4.9.

4.3.4 Perancangan Antarmuka

Perancangan antarmuka sistem bertujuan memastikan setiap aktor, yaitu admin, pelanggan, dan manajer, dapat menjalankan perannya sesuai kebutuhan masing-masing. Desain antarmuka dibuat berdasarkan fungsi utama yang telah didefinisikan pada tahap analisis kebutuhan, sehingga setiap menu dan tampilan difokuskan pada aktivitas yang paling relevan bagi pengguna. Bagian ini menampilkan rancangan antarmuka yang digunakan oleh pelanggan untuk melihat layanan, melakukan pemesanan, serta meninjau konfirmasi pemesanan. Rancangan antarmuka secara keseluruhan dapat dilihat pada **Lampiran D**.

4.3.4.1 Pelanggan Lihat Layanan Dan Melakukan Pemesanan

Antarmuka ini menampilkan halaman yang digunakan pelanggan untuk melihat detail layanan yang tersedia. Melalui halaman ini, pelanggan dapat memilih layanan sesuai kebutuhan dan melanjutkan ke proses pemesanan. Tampilan rancangan antarmuka tersebut dapat dilihat pada Gambar 4.10.



Gambar 4.10 Halaman Antarmuka Pelanggan lihat detail layanan

Pada Gambar 4.10. ditampilkan antarmuka yang digunakan pelanggan untuk melihat daftar layanan penyewaan yang tersedia. Informasi yang disajikan mencakup nama layanan, kategori, deskripsi singkat, lokasi, fasilitas yang termasuk dalam layanan, fasilitas tambahan, harga dasar, metode pembayaran, serta ketersediaan jadwal. Melalui halaman ini, pelanggan dapat memilih layanan tertentu dan melanjutkan ke tahap konfirmasi pemesanan.

4.3.4.1 Pelanggan Lihat Konfirmasi Pemesanan Layanan

Antarmuka ini digunakan pelanggan untuk memeriksa detail konfirmasi setelah melakukan pemesanan. Informasi yang ditampilkan meliputi layanan yang dipilih, jadwal ditunjukkan pada Gambar 4.11.

Konfirmasi Booking Lapangan Golf

Fasilitas Tambahan

Rental Golf Set
Set lengkap stick golf untuk pemula hingga mahir

Rp 120.000

Ringkasan Pesanan

Layanan:	Lapangan Golf
Tanggal:	12 Mei 2025
Waktu:	10.00 - 12.00 WIB
Lokasi:	Agam

Harga Base:	Rp 150.000
Rental Golf Set:	Rp 50.000
Total:	Rp xxx.xxx

Pembayaran Penuh
Bayar penuh Rp xxxxx sekarang.
Lebih praktis, tidak ada pembayaran di tempat.

[Lanjut Ke Pembayaran](#)

Nama Aplikasi
Layanan Koperasi Hydro Utama

Home **Profile Koperasi** **Riwayat Sewa**

Hubungi Kami
+62 81234567890
koperasi@gmail.com

Layanan
Lapangan Golf
Pewedding
Ruang Rapat

Tentang (Aplikasi)
Profile Koperasi
Feedback

Gambar 4.11 Halaman Antarmuka Pelanggan lihat konfirmasi pemesanan layanan

Pada Gambar 4.11 halaman ini, sistem menampilkan informasi lengkap mengenai layanan yang dipilih, termasuk jadwal penggunaan, status penyewaan, fasilitas tambahan (jika ada), serta total biaya yang harus dibayarkan. Setelah meninjau

seluruh detail dan memastikan data telah sesuai, pelanggan dapat melanjutkan ke tahap pembayaran melalui tombol “Lanjut ke Pembayaran” yang tersedia.



BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas proses implementasi Sistem Informasi Jasa Penyewaan berbasis *web* yang telah dirancang pada bab sebelumnya. Pembahasan mencakup tahap penerjemahan rancangan ke dalam kode program, penerapan antarmuka pengguna, serta integrasi antar modul sistem. Selain itu, bab ini juga menjelaskan hasil pengujian sistem yang dilakukan menggunakan metode *unit testing* untuk memastikan setiap modul berfungsi sesuai rancangan, serta *black box testing* untuk memverifikasi kesesuaian fungsionalitas sistem terhadap kebutuhan pengguna.

5.1 Implementasi dan Pengujian Unit

Pada tahap implementasi sistem, rancangan sistem diterapkan dalam bentuk kode program menggunakan bahasa pemrograman dan *framework* yang telah ditentukan. Proses implementasi mencakup penulisan kode, penerapan desain antarmuka pengguna sesuai rancangan, serta integrasi modul agar seluruh komponen dapat berfungsi aplikasi agar sesuai spesifikasi sistem. Pelaksanaan implementasi membutuhkan dukungan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

1. Spesifikasi perangkat keras (*hardware*) yang digunakan adalah sebagai berikut:
 - a. Prosesor Intel Core i5-10210U dengan kecepatan 1.60 GHz hingga 2.11 GHz
 - b. Memori RAM 8 GB
 - c. *Solid State Drive* (SSD) berkapasitas 500 GB
 - d. *Hard Disk Drive* (HDD) berkapasitas 1 TB
2. Spesifikasi perangkat lunak (*software*) yang digunakan adalah sebagai berikut:
 - a. Sistem Operasi: Windows 11
 - b. *Web Browser*: Google Chrome dan Microsoft Edge
 - c. *Code Editor*: Visual Studio Code
 - d. *Database*: MySQL
 - e. Bahasa Pemrograman & *Framework*: PHP 8.2+ dengan *Laravel* 12.0

5.1.1 Pengkodean Program

Tahap pengkodean merupakan proses menerjemahkan rancangan sistem ke dalam bahasa pemrograman dengan membangun modul, fungsi, dan logika aplikasi sesuai spesifikasi. Sistem dikembangkan menggunakan *framework laravel* dengan pola arsitektur *Model, View, Controller* (MVC), di mana setiap komponen memiliki peran masing-masing dalam alur kerja aplikasi.

Pada bagian ini ditampilkan contoh perwakilan kode inti sistem sebagai ilustrasi implementasi, terutama pada modul Penyewaan yang menjadi fitur utama aplikasi. Modul ini merepresentasikan data hasil dari Pemesanan yang dilakukan pelanggan, sehingga setiap pemesanan layanan akan tersimpan dalam tabel penyewaan. Contoh kode yang disajikan mencakup *routing, Controller, model*, dan *view* terkait proses pemesanan dan pengelolaan penyewaan. Kode lengkap keseluruhan aplikasi dapat dilihat secara detail pada **Lampiran E**.

5.1.1.1 Routing

Routing berfungsi sebagai pengatur jalur atau alur *Request* yang masuk ke aplikasi. Setiap permintaan dari pengguna, seperti membuka halaman pemesanan atau mengakses data penyewaan, diteruskan ke *Controller* yang sesuai melalui mekanisme *routing*. Dengan demikian, *routing* menentukan alur eksekusi *Request* sebelum diproses lebih lanjut.

Pada modul Penyewaan, *routing* digunakan untuk mengatur akses URL terkait proses pemesanan yang dilakukan pelanggan. Sistem menggunakan *route groups* dengan *middleware* autentikasi dan kontrol akses berbasis peran (*role-based access control*) untuk memastikan keamanan. Setiap route didefinisikan dengan metode HTTP sesuai fungsinya, yaitu:

1. *GET*, mengambil data dari server, misalnya menampilkan halaman pemesanan atau daftar penyewaan.
2. *POST*, mengirim data baru ke server, misalnya membuat pemesanan baru.
3. *PUT*, memperbarui data yang sudah ada, misalnya membatalkan pemesanan.

4. *DELETE*, menghapus data dari server (tidak digunakan pada modul penyewaan ini).

```
1 Route::middleware(['auth', 'verified', 'role:customer'])->prefix('customer')->name('customer.')->group(function () {
```

Gambar 5.1 *Route Middleware Customer (Pelanggan)*

Pada Gambar 5.1 menunjukkan contoh *route group* ini menggunakan *middleware auth* (harus *login*), *verified* (email terverifikasi), dan *role:customer* (hanya customer yang bisa akses). *Prefix customer* membuat semua URL dimulai dengan */customer/*, dan *name('customer.')* memberikan *prefix* pada penamaan *route*.

```
1 // Booking Availability API
2 Route::get('/booking/slots', [PenyewaanController::class, 'getAvailableSlots'])->name('booking.slots');
3 Route::get('/booking/availability', [PenyewaanController::class, 'getAvailability'])->name('booking.availability');
4 Route::get('/booking/check-availability', [PenyewaanController::class, 'checkAvailability'])->name('booking.check-availability');
```

Gambar 5.2 *API Ketersediaan Pemesanan*

Pada Gambar 5.2. menampilkan *route API* untuk pengecekan ketersediaan slot pemesanan. *Route* ini menggunakan metode *GET* karena hanya mengambil data tanpa melakukan perubahan pada basis data, serta dipanggil melalui *AJAX* untuk memvalidasi jadwal secara langsung pada saat permintaan dilakukan.

```
1 //Proses Pemesanan
2 Route::prefix('penyewaan')->name('penyewaan.')->group(function () {
3     Route::get('/', [PenyewaanController::class, 'customerHistory'])->name('index');
4     Route::get('/create', [PenyewaanController::class, 'create'])->name('create');
5     Route::post('/process', [PenyewaanController::class, 'store'])->name('process');
6     Route::get('/confirm', [PenyewaanController::class, 'showConfirm'])->name('confirm.show');
7     Route::post('/confirm', [PenyewaanController::class, 'confirm'])->name('confirm');
8     Route::get('/{id}/detail', [PenyewaanController::class, 'show'])->name('detail');
9     Route::put('/{id}/cancel', [PenyewaanController::class, 'cancel'])->name('cancel');
10 });
```

Gambar 5.3 *Route Pelanggan Melakukan Pemesanan*

Pada Gambar 5.3. memperlihatkan *route group* yang mengatur seluruh proses pemesanan oleh pelanggan. Fungsionalitas yang diatur meliputi melihat riwayat penyewaan, membuat pemesanan baru, konfirmasi, hingga pembatalan pemesanan.

5.1.1.4 Model

Model merupakan komponen yang berhubungan langsung dengan basis data. *Model* bertugas mengatur struktur data, menjalankan *query*, dan mengelola logika bisnis yang berkaitan dengan data. Dengan *model*, proses penyimpanan, pembacaan, perubahan, maupun penghapusan data dapat dilakukan secara terstruktur.

Dalam sistem ini, *Model* Penyewaan merepresentasikan catatan transaksi hasil dari proses Pemesanan. Setiap kali pelanggan memilih layanan, menentukan jadwal, dan mengajukan pemesanan, data tersebut tersimpan pada tabel penyewaan. Informasi yang dikelola meliputi identitas pelanggan, layanan yang dipilih, jadwal penggunaan, status penyewaan, serta keterkaitannya dengan pembayaran.

Model Penyewaan juga berperan sebagai pusat logika bisnis pada modul ini. Implementasinya menggunakan *UUID* sebagai *primary key*, dilengkapi relasi dengan model lain (*user*, layanan, pembayaran, fasilitas tambahan, *refund request*), serta menyediakan *helper methods* untuk menangani status penyewaan, kondisi pembatalan, dan perhitungan pembayaran. Dengan demikian, seluruh aktivitas pemesanan terdokumentasi sekaligus dapat diolah secara sistematis melalui model ini.

```

class Penyewaan extends Model
{
    use HasUuids;

    protected $table = 'penyewaan';
    protected $primaryKey = 'id_penyewaan';
    public $incrementing = false;
    protected $keyType = 'string';

    protected $fillable = [
        'tanggal_penyewaan',
        'alasan_status',
        'kode_sewa',
        'id_customer',
        'id_layanan',
        'metode_pembayaran',
        'harga_dasar_layanan',
        'total_fasilitas_tambahan',
        'fasilitas_tambahan_detail',
        'total_harga',
        'nominal_dp',
        'siswa_pembayaran',
        'batas_pembayaran_dp',
        'batas_pelunasan',
        'jam_mulai',
        'jam_selesai',
        'status_penyewaan',
        'reserved_until',
        'cancelled_at',
        'cancelled_by',
        'catatan_customer',
        'rating',
        'komentar',
        'balasan_admin',
        'replied_at'
    ];
}

```

Gambar 5.4 Struktur Model Penyewaan

Pada Gambar 5.4. kode ini mendefinisikan struktur dasar *model* Penyewaan dengan menggunakan *trait HasUuids* untuk menghasilkan *UUID* sebagai *primary key*. *Property \$table* menentukan nama tabel database yang digunakan, yaitu *penyewaan*. *Property \$primaryKey* mengatur *primary key* menjadi *id_penyewaan* dengan tipe data *string* dan tidak *auto-increment*. *Property \$fillable* mendefinisikan field-field yang dapat diisi secara mass assignment, meliputi data identitas pemesanan, *user*, layanan, metode pembayaran, harga, dan status penyewaan.

```

public function user()
{
    return $this->belongsTo(User::class, 'id_customer', 'id_user');
}

public function customer()
{
    return $this->belongsTo(Customer::class, 'id_customer', 'id_customer');
}

public function layanan()
{
    return $this->belongsTo(Layanan::class, 'id_layanan');
}

public function pembayaran()
{
    return $this->hasMany(Pembayaran::class, 'id_penyewaan');
}

public function refundRequest()
{
    return $this->hasOne(RefundRequest::class, 'id_penyewaan');
}

```

Gambar 5.5 Relasi Model Penyewaan

Pada Gambar 5.5. kode ini mendefinisikan relasi antar model dalam sistem. *Method user()* menggunakan relasi *belongsTo* untuk menghubungkan penyewaan dengan *user* yang melakukan pemesanan. *Method layanan()* menghubungkan penyewaan dengan layanan yang disewa. *Method pembayaran()* menggunakan relasi *hasMany* karena satu penyewaan dapat memiliki banyak pembayaran (DP dan pelunasan). *Method fasilitasTambahkan()* menghubungkan penyewaan dengan fasilitas tambahan yang dipesan. *Method refundRequest()* menggunakan relasi *hasOne* karena satu penyewaan hanya dapat memiliki satu permintaan *refund*.

```

public function isPending()
{
    return $this->status_penyewaan === 'pending';
}

public function isReserved()
{
    return $this->status_penyewaan === 'reserved';
}

public function isConfirmed()
{
    return $this->status_penyewaan === 'confirmed';
}

public function isActive()
{
    return $this->status_penyewaan === 'active';
}

public function isCompleted()
{
    return $this->status_penyewaan === 'completed';
}

public function isCancelled()
{
    return $this->status_penyewaan === 'cancelled';
}

public function isRefunded()
{
    return $this->status_penyewaan === 'refunded';
}

```

Gambar 5.6 Helper Methods untuk Status Penyewaan

Pada Gambar 5.6, kode ini mendefinisikan *helper methods* untuk menangani logika bisnis status dan pembayaran. *Method* `isPending()`, `isReserved()`, dan `isConfirmed()` mengecek status penyewaan yang spesifik. *Method* `canBeCancelled()` menentukan apakah penyewaan dapat dibatalkan berdasarkan status dan kondisi pembayaran. *Method* `totalPaid()` menghitung total pembayaran yang sudah lunas. *Method* `isFullyPaid()` mengecek apakah penyewaan sudah lunas dengan membandingkan total yang dibayar dengan total harga. Dengan adanya *Helper methods* ini memudahkan *controller* dan *view* dalam menangani logika bisnis yang kompleks.

5.1.1.3 Controller

Controller berfungsi sebagai penghubung antara *model* dan *view*, sekaligus menangani logika bisnis serta alur data dalam aplikasi. Pada modul Penyewaan, *controller* utama adalah *PenyewaanController* yang mengelola seluruh proses pemesanan layanan oleh pelanggan, mulai dari validasi input, pengecekan ketersediaan jadwal, hingga penyimpanan data ke basis data. *Controller* ini memanfaatkan *dependency injection* untuk mengakses *model* dan *service*, serta menerapkan validasi ketat guna menjaga integritas data.

```
public function create()
{
    $currentUser = Auth::user();
    $customerId = $currentUser->getCustomerId() ?: $currentUser->id_user;
    $unpaidBookings = $this->getTrulyUnpaidBookings($customerId);

    if ($unpaidBookings->count() > 0) {
        return $this->showUnpaidBookingWarning();
    }

    $pendingFeedback = Penyewaan::where('id_customer', Auth::id())
        ->where('status_penyewaan', 'completed')
        ->whereNull('rating')
        ->count();

    if ($pendingFeedback > 0) {
        return redirect()->route('customer.feedback.index')
            ->with('warning', "Anda memiliki {$pendingFeedback} penyewaan yang belum diberi feedback. Silakan berikan feedback terlebih dahulu sebelum melakukan booking baru.");
    }

    return redirect()->route('customer.dashboard')
        ->with('info', 'Silakan pilih layanan dari dashboard untuk melakukan booking.');
```

Gambar 5.7 Method *Create* - Form Pemesanan

Pada Gambar 5.7, method *create()* menampilkan form pemesanan baru. Sebelum ditampilkan, sistem mengecek apakah pelanggan masih memiliki pemesanan yang belum lunas dengan *getTrulyUnpaidBookings()*. Jika ada, pelanggan dialihkan ke halaman peringatan menggunakan *showUnpaidBookingWarning()*. Jika tidak, sistem mengambil daftar layanan aktif melalui *Layanan::where('status', 'active')->get()* dan menampilkan *form* pemesanan dengan data layanan yang tersedia.

```

public function confirm(Request $request)
{
    $request->validate([
        'layanan_id' => 'required|exists:layanan,id_layanan',
        'booking_date' => 'required|date|after_or_equal:today',
        'time_slot' => 'required|string',
        'metode_pembayaran' => 'required|in:dp_50,bayar_penuh',
        'fasilitas_tambahan' => 'nullable|array',
        'fasilitas_tambahan.*' => 'exists:fasilitas_tambahan,id_fasilitas_tambahan',
        'catatan' => 'nullable|string|max:500'
    ]);

    $currentUser = Auth::user();
    $customerId = $currentUser->getCustomerId() ?: $currentUser->id_user;
    $unpaidBookings = $this->getTrulyUnpaidBookings($customerId);

    if ($unpaidBookings->count() > 0) {
        $unpaidList = $unpaidBookings->map(function($booking) {
            $totalPaid = $booking->pembayaran()
                ->whereIn('status_pembayaran', ['verified'])
                ->sum('jumlah_bayar');
            $sisapembayaran = $booking->total_harga - $totalPaid;

            return $booking->layanan->nama_layanan . ' (' . $booking->tanggal_penyewaan->format('d/m/Y') . ')  

- Sisa: Rp ' . number_format($sisapembayaran, 0, ',', '.');
        })->join(', ');

        return back()->withErrors([
            'unpaid_booking' => "Anda masih memiliki booking yang belum lunas penuh: {$unpaidList}. Silakan  

lunasi terlebih dahulu sebelum melakukan booking baru."
        ]);
    }

    $layanan = Layanan::findOrFail($request->layanan_id);
    if ($layanan->tipe_sewa === 'per_hari') {
        $conflictingBooking = Penyewaan::join('layanan', 'penyewaan.id_layanan', '=', 'layanan.id_layanan')
            ->where('layanan.area', $layanan->area)
            ->where('penyewaan.tanggal_penyewaan', $request->booking_date)
            ->where('penyewaan.id_layanan', '!=', $request->layanan_id) // Exclude layanan yang

```

Gambar 5.8 Method *Confirm* - Validasi dan Konfirmasi

Pada Gambar 5.8, *method* `confirm()` melakukan validasi input menggunakan Laravel validation rules. Setelah itu, sistem memeriksa pemesanan yang belum lunas serta konflik jadwal dengan layanan lain. Untuk layanan harian, pengecekan dilakukan per tanggal; sedangkan untuk layanan berbasis sesi, pengecekan dilakukan per slot waktu. Proses ini menggunakan *query* gabungan untuk memastikan tidak ada konflik lokasi.

```

public function getAvailableSlots(Request $request)
{
    $request->validate([
        'id_layanan' => 'required|exists:layanan,id,layanan',
        'tanggal' => 'required|date|after_or_equal:today'
    ]);

    $layanan = Layanan::findOrFail($request->id_layanan);
    $tanggal = $request->tanggal;

    $jamBuka = Carbon::parse($layanan->jam_buka_operasional);
    $jamTutup = Carbon::parse($layanan->jam_tutup_operasional);
    $durasiSesi = (int) (config('layanan.durasi_sesi_default') ?? 0);
    $jedaSesi = (int) 0;

    $slots = [];
    $currentTime = $jamBuka->copy();
    $sesiCount = 1;

    while ($currentTime < $jamTutup) {
        $sendTime = $currentTime->copy()->addMinutes((int)$durasiSesi);
        if ($sendTime <= $jamTutup) {
            $jamMulai = $currentTime->format('H:i');
            $jamSelesai = $sendTime->format('H:i');
            if ($layanan->tipe_sewa === 'per_hari') {
                $existingBooking = Penyewaan::join('layanan', 'penyewaan.id_layanan', '=',
                    'layanan.id_layanan')
                    ->where('layanan.lokasi', $layanan->lokasi)
                    ->where('layanan.tipe_sewa', 'per_hari')
                    ->where('penyewaan.tanggal_penyewaan', $tanggal)
                    ->whereIn('penyewaan.status_penyewaan', ['pending', 'reserved', 'confirmed'])
                    ->with('user', 'layanan')
                    ->first();
            } else {
                $perHariConflict = Penyewaan::join('layanan', 'penyewaan.id_layanan', '=',
                    'layanan.id_layanan')
                    ->where('layanan.lokasi', $layanan->lokasi)
                    ->where('layanan.tipe_sewa', 'per_hari')
                    ->where('penyewaan.tanggal_penyewaan', $tanggal)
                    ->whereIn('penyewaan.status_penyewaan', ['pending', 'reserved', 'confirmed'])
                    ->with('user', 'layanan')
                    ->first();

                if ($perHariConflict) {
                    $existingBooking = $perHariConflict;
                } else {
                    $existingBooking = Penyewaan::join('layanan', 'penyewaan.id_layanan', '=',
                        'layanan.id_layanan')
                        ->where('layanan.lokasi', $layanan->lokasi)
                        ->where('layanan.tipe_sewa', 'per_sesi')
                        ->where('penyewaan.tanggal_penyewaan', $tanggal)
                        ->whereIn('penyewaan.status_penyewaan', ['pending', 'reserved', 'confirmed'])
                        ->where(function($query) use ($jamMulai, $jamSelesai) {
                            $query->where(function($q) use ($jamMulai, $jamSelesai) {
                                $q->where('jam_mulai', '<', $jamMulai);
                                ->where('jam_selesai', '>', $jamMulai);
                            });
                        });
                    ->with('user', 'layanan')
                    ->first();
                }
            }

            $status = 'available';
            $bookingInfo = null;

            if ($existingBooking) {
                if ($existingBooking->status_penyewaan === 'reserved') {
                    $status = 'reserved';
                    $remainingTime = 30;
                    $bookingInfo = [
                        'status' => 'reserved',
                        'remaining_minutes' => $remainingTime,
                        'message' => "Sedang diproses - ($remainingTime) menit lagi"
                    ];
                } else {
                    $status = 'booked';
                    $bookingInfo = [
                        'customer' => $existingBooking->user->getNamaLengkap(),
                        'status' => $existingBooking->status_penyewaan,
                        'layanan' => $existingBooking->layanan->nama_layanan,
                        'tipe_sewa' => $existingBooking->layanan->tipe_sewa,
                        'lokasi' => $existingBooking->layanan->lokasi
                    ];
                }
            }

            $slot = [
                'sesi' => $sesiCount,
                'jam_mulai' => $jamMulai,
                'jam_selesai' => $jamSelesai,
                'available' => $status === 'available',
                'status' => $status,
                'booking_info' => $bookingInfo
            ];
            if ($layanan->tipe_sewa === 'per_sesi') {
                $slots[] = $slot;
            } else {
                if ($status === 'available') {
                    $slots[] = $slot;
                }
            }
            $currentTime = $sendTime->addMinutes((int)$jedaSesi);
            $sesiCount++;
        } else {
            break;
        }
    }

    return response()->json([
        'success' => true,
        'slots' => $slots,
        'service_info' => [
            'nama_layanan' => $layanan->nama_layanan,
            'tipe_sewa' => $layanan->tipe_sewa,
            'jam_buka' => $jamBuka->format('H:i'),
            'jam_tutup' => $jamTutup->format('H:i')
        ]
    ]);
}

```

Gambar 5.9 Method *GetAvailableSlots* - API Ketersediaan

Pada Gambar 5.9. *method* `getAvailableSlots()` menangani API untuk mendapatkan slot waktu yang tersedia. Method ini menggunakan Carbon untuk parsing waktu operasional layanan dan generate slot waktu dengan durasi 2 jam per sesi. *Method* ini mengecek konflik berdasarkan tipe sewa: untuk layanan per hari akan memblokir seluruh hari jika ada booking di lokasi yang sama, sedangkan untuk per sesi akan mengecek overlap waktu dengan booking yang sudah ada. Method ini menggunakan join query yang kompleks untuk mengecek konflik dengan layanan lain di lokasi yang sama.

5.1.1.4 Views

View merupakan lapisan presentasi yang bertugas menampilkan data dari *controller* kepada pengguna melalui antarmuka web. Dalam Laravel, *view* dibangun menggunakan template *engine Blade*, yang memungkinkan kode PHP ditulis secara lebih bersih dan terintegrasi dengan HTML.

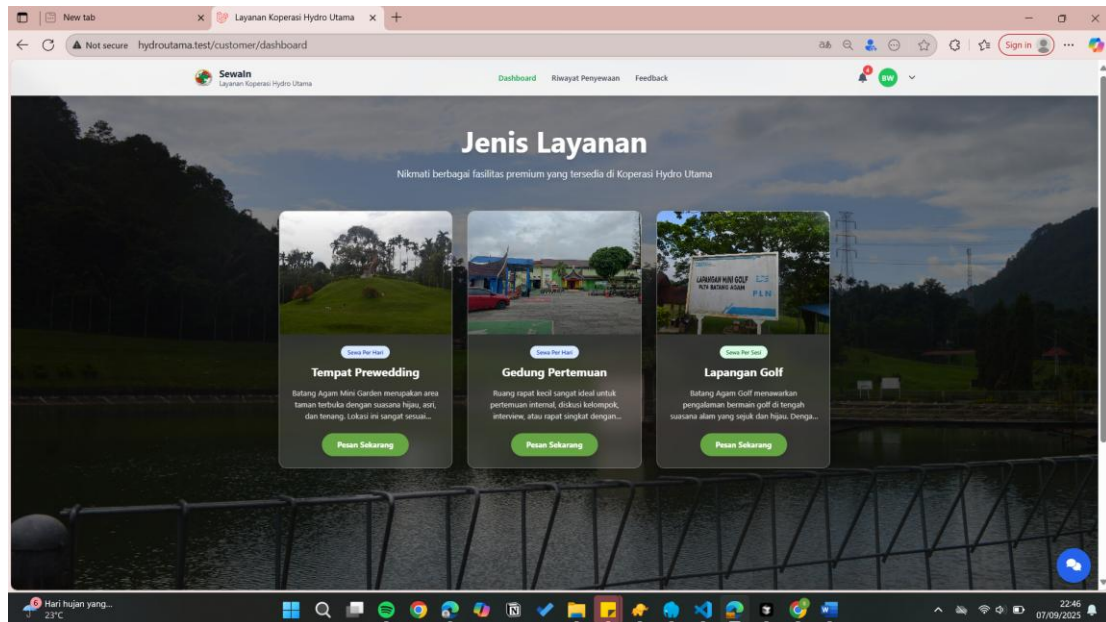
Pada modul Penyewaan, view digunakan untuk menampilkan halaman-halaman utama dalam proses pemesanan pelanggan, seperti detail pemesanan, halaman konfirmasi, riwayat pemesanan, dan detail penyewaan. *View* ini memanfaatkan komponen *Blade* yang dapat digunakan kembali serta menerapkan responsive design dengan *framework* CSS *Tailwind*, sehingga tampilan dapat menyesuaikan berbagai perangkat.

Setelah proses pengkodean selesai dilakukan, tahap berikutnya adalah membangun antarmuka pengguna yang sesuai dengan rancangan antarmuka lakukan sebelumnya.

5.1.2 Implementasi Antarmuka Sistem

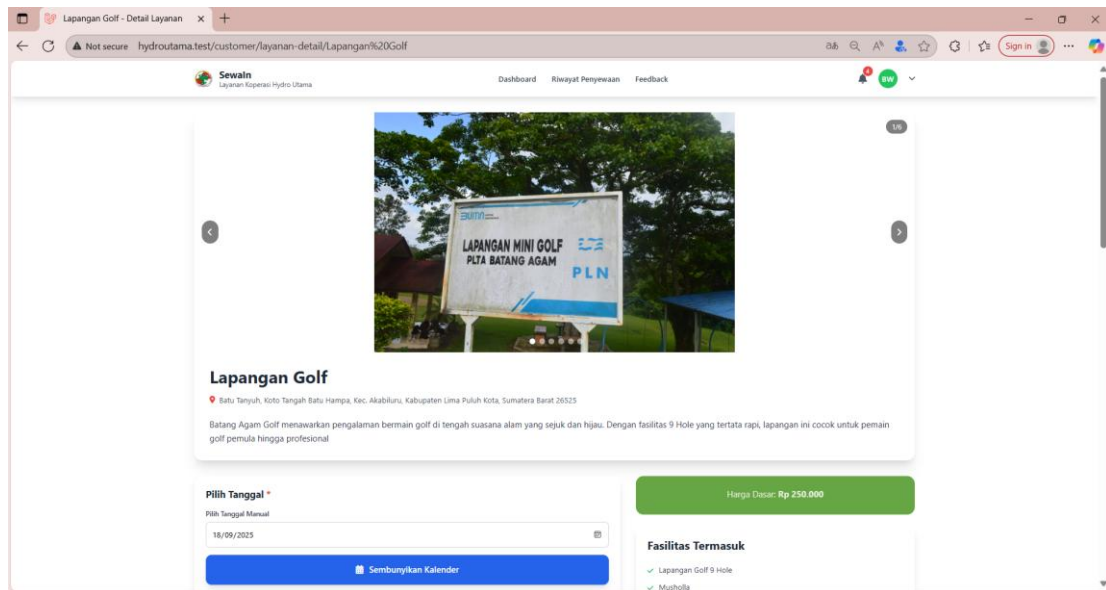
Setelah tahap pengkodean selesai dilakukan, langkah berikutnya adalah implementasi antarmuka pengguna (*user interface*) sesuai rancangan yang telah dibuat. Sistem dirancang untuk tiga jenis pengguna, yaitu Admin, Pelanggan, dan Manajer, dengan tampilan yang disesuaikan dengan kebutuhan masing-masing peran. Contoh implementasi antarmuka mencakup halaman *dashboard*, detail pemesanan layanan,

halaman pembayaran, serta riwayat penyewaan ditunjukkan pada Gambar 5.10 sampai dengan Gambar 5.16.



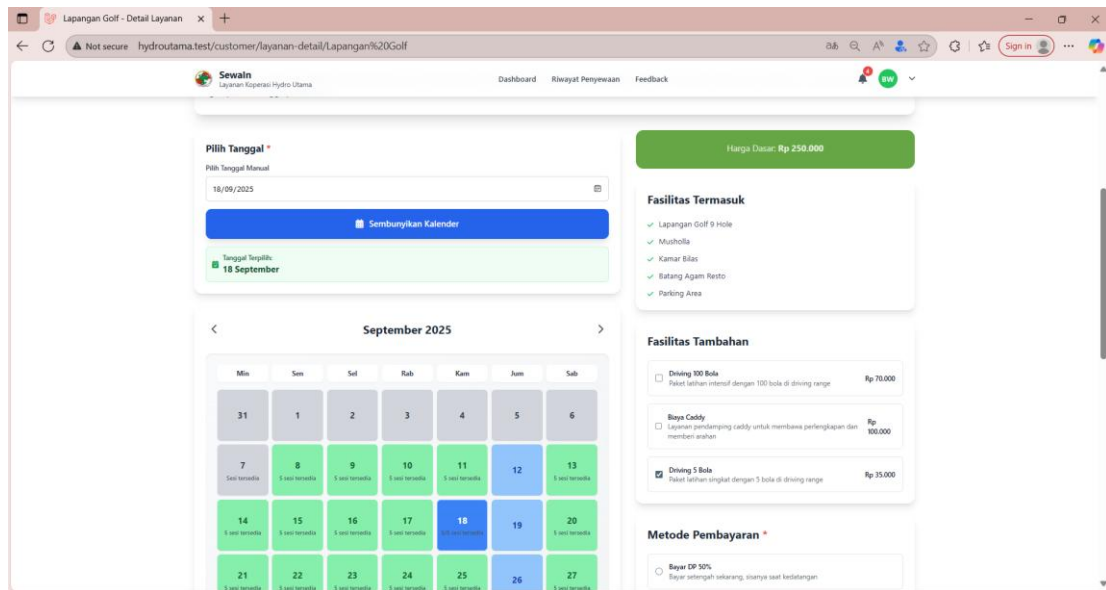
Gambar 5.10 Implementasi Antarmuka *Dashboard* Pelanggan

Pada Gambar 5.10 Halaman *dashboard* pelanggan ditampilkan setelah pelanggan melakukan login, setelah itu dapat melihat katalog layanan yang tersedia di Koperasi Konsumen Hydro Utama, antara lain ada layanan lokasi foto *pre wedding*, gedung pertemuan, dan lapangan golf. Setiap layanan disajikan dalam bentuk kartu (*card*) yang berisi foto, deskripsi singkat, serta tombol Pesan Sekarang. Tujuannya adalah mempermudah pelanggan mengenali pilihan layanan dan langsung diarahkan ke detail layanan yang diinginkan.



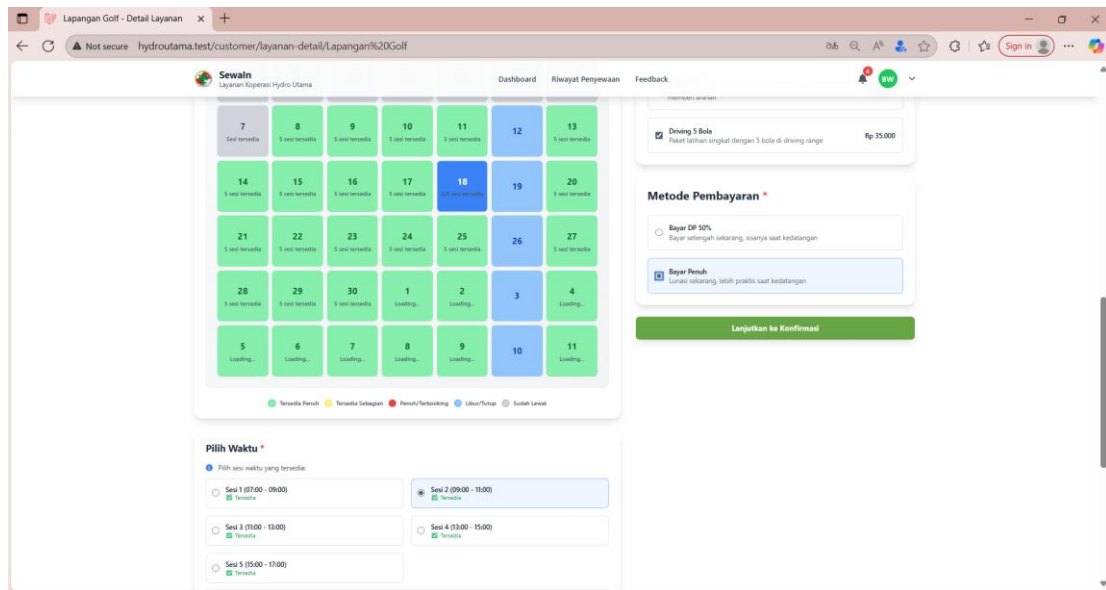
Gambar 5.11 Implementasi Antarmuka Detail Layanan

Pada Gambar 5.11 Halaman detail layanan muncul ketika pelanggan menekan tombol Pesan Sekarang yang menampilkan informasi lengkap dari layanan yang dipilih pengguna. Pada contoh gambar, layanan yang ditampilkan adalah Lapangan Golf. Antarmuka ini menyajikan deskripsi lokasi, fasilitas yang tersedia, dan harga dasar penyewaan. Dengan tampilan ini, pelanggan bisa meninjau kelayakan layanan sebelum melanjutkan ke tahap pemesanan. Form pemilihan tanggal sewa juga mulai ditampilkan pada bagian bawah halaman agar pelanggan bisa langsung melakukan input tanggal.



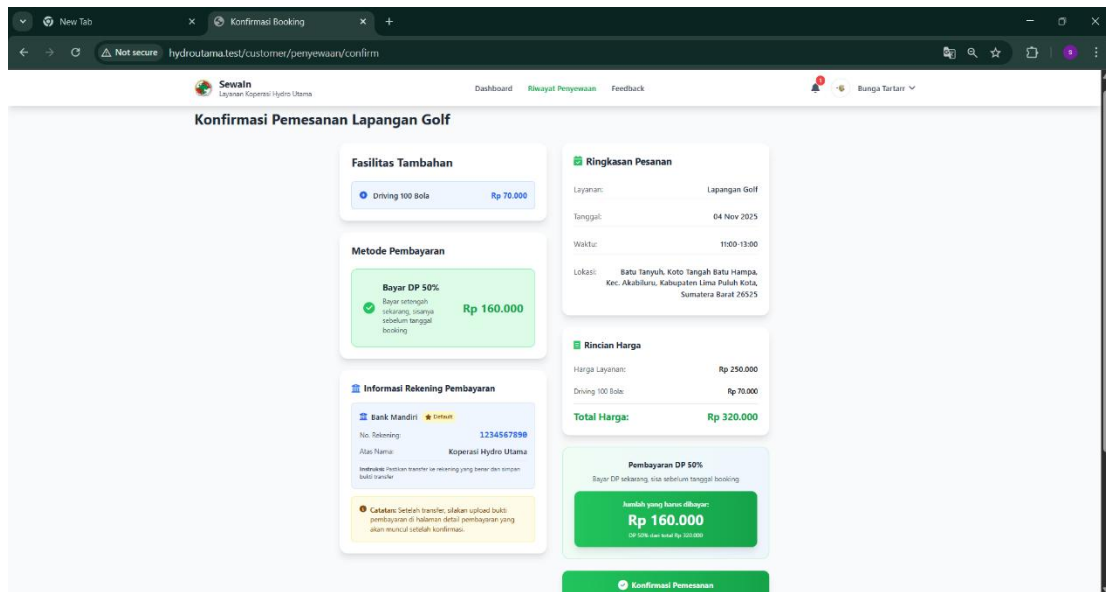
Gambar 5.12 Implementasi Antarmuka Detail Layanan (1)

Pada Gambar 5.12 lanjutan dari Halaman detail layanan memperlihatkan fitur kalender yang memungkinkan pelanggan melihat tanggal sewa. Kalender dilengkapi indikator warna untuk menunjukkan status ketersediaan, misalnya tersedia, penuh, atau lewat. Informasi tambahan seperti fasilitas utama juga ditampilkan di sisi kanan layar untuk membantu pelanggan memahami apa saja yang sudah termasuk dalam layanan. Dengan mekanisme ini, proses pemesanan menjadi lebih transparan dan meminimalkan potensi bentrok jadwal.



Gambar 5.13 Implementasi Antarmuka Detail Layanan (2)

Pada Gambar 5.13 Lanjutan Halaman detail layanan berikutnya, pelanggan dapat memilih opsional fasilitas tambahan yang tersedia, seperti Driving 100 Bola atau jasa Caddy, lengkap dengan harga masing-masing. Selain itu, sistem juga menyediakan opsi metode pembayaran, baik dengan DP 50% maupun pembayaran penuh. Di bagian kiri halaman, terdapat pilihan sesi waktu yang bisa dipilih sesuai ketersediaan. Antarmuka ini dirancang untuk memberikan fleksibilitas bagi pelanggan dalam menyesuaikan pemesanan sesuai kebutuhan.



Gambar 5.14 Implementasi Antarmuka Konfirmasi Pemesanan Layanan

Pada Gambar 5.14 Setelah semua data terisi, sistem menampilkan halaman konfirmasi pemesanan. Halaman ini berfungsi sebagai ringkasan pesanan yang berisi detail layanan, tanggal, sesi, fasilitas tambahan (jika memilih fasilitas tambahan), metode pembayaran, hingga total biaya yang harus dibayarkan. Dengan adanya tampilan ini, pelanggan dapat memverifikasi kembali kebenaran informasi sebelum mengirimkan pesan secara final. Tombol Konfirmasi Pemesanan menjadi langkah akhir untuk menyelesaikan proses pemesanan.

Detail Pembayaran

Booking berhasil dipesan! Silakan upload bukti pembayaran dalam 30 menit.

Informasi Pemesanan

Kode Booking: SEWA-20251030-5489

Layanan: Tempat Prewedding

Tanggal: 01 Nov 2025

Waktu: 07:00 - 17:00 (Sehari Penuh)

Informasi Pembayaran

Kode Pembayaran: BYR-PENUH-20251030-6710

Jenis Pembayaran: Bayar Penuh

Jumlah Bayar: Rp 1.000.000

Status: Menunggu Pembayaran

Batas Waktu: 30 Oct 2025 12:16

Upload Bukti Transfer

Bukti Transfer *

No file chosen

Format: JPG, PNG (Maks 2MB)

Nama Pengirim *

Nama yang tertera di bukti transfer

Bank Pengirim *

Ketik nama bank atau pilih dari daftar

Ketik nama bank atau pilih dari daftar yang tersedia

Tanggal transfer *

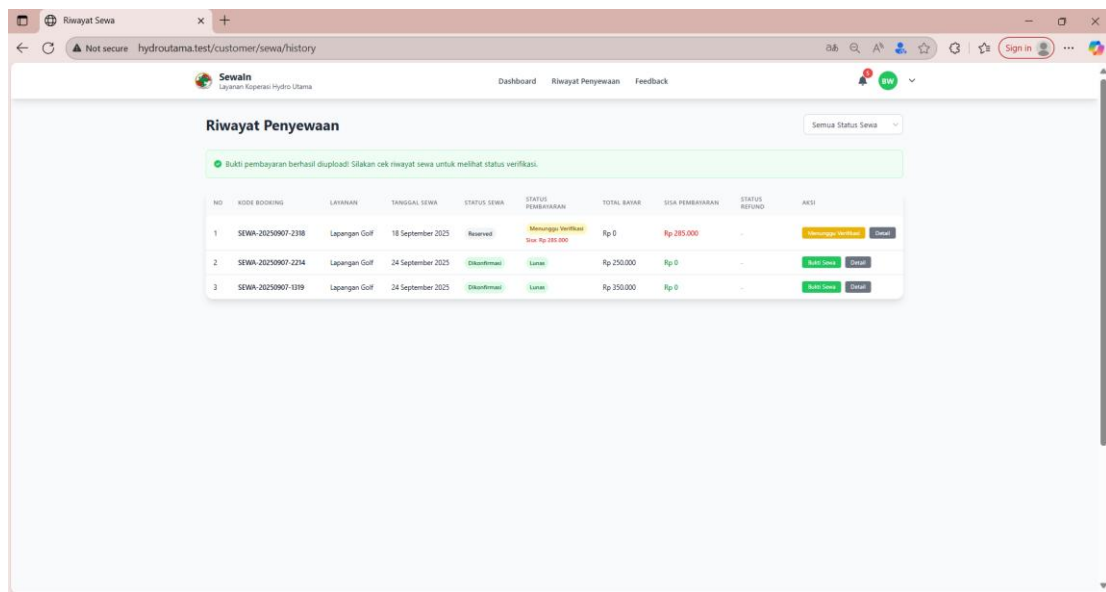
30/10/2025

Catatan (Optional)

Catatan tambahan untuk admin

Gambar 5.15 Implementasi Antarmuka Detail Pembayaran

Pada Gambar 5.15 Halaman detail pembayaran berfokus pada proses transaksi. Informasi pemesanan ditampilkan secara lengkap, mulai dari kode *booking*, jenis layanan, jadwal, hingga jumlah tagihan. Selain itu, ditampilkan juga rekening tujuan pembayaran. Pada sisi kanan halaman tersedia form untuk mengunggah bukti transfer dengan tambahan detail seperti nama pengirim, bank pengirim, tanggal transfer, dan catatan opsional. Dengan tampilan ini, sistem dapat mencatat pembayaran pelanggan secara lebih terstruktur dan mempermudah proses verifikasi oleh admin.



Gambar 5.16 Implementasi Antarmuka Riwayat Penyewaan

Pada Gambar 5.16 Halaman riwayat penyewaan menampilkan daftar seluruh transaksi yang sudah dilakukan pelanggan. Informasi yang disediakan mencakup kode *booking*, layanan yang dipesan, tanggal sewa, status penyewaan, status pembayaran, hingga nominal yang dibayarkan. Selain itu, tersedia tombol Detail dan Bukti Sewa agar pelanggan bisa meninjau lebih lanjut setiap transaksi. Antarmuka ini membantu pelanggan memantau histori pemesanan dengan jelas, sekaligus memberikan transparansi dalam status verifikasi pembayaran dan penggunaan layanan.

Setiap elemen tampilan, mulai dari menu navigasi, form input, hingga tombol aksi, dirancang konsisten agar alur interaksi dengan sistem mudah dipahami dan mendukung proses penyewaan secara terstruktur. Dokumentasi lengkap hasil implementasi antarmuka lainnya ditampilkan pada **Lampiran F**.

5.1.3 Pengujian Unit

Pengujian unit merupakan tahap awal pengujian perangkat lunak yang dilakukan pada bagian terkecil dari sistem, yaitu fungsi atau metode di dalam masing-masing modul. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap komponen berjalan sesuai dengan spesifikasi yang telah ditetapkan sebelum digabungkan ke tahap integrasi. Dengan pendekatan ini, kesalahan dapat dideteksi

lebih cepat dan diperbaiki tanpa mempengaruhi bagian lain dari sistem. Pada penelitian ini, pengujian unit dilakukan pada modul-modul inti sistem salah satunya modul *User*, sedangkan detail pengujian modul lain ditampilkan pada **Lampiran G**

1. Controller adalah komponen yang bertanggung jawab untuk mengelola permintaan dari pengguna dan mengembalikan respons yang sesuai. Dalam pengujian unit, controller diuji untuk memastikan bahwa alur pengolahan data dari permintaan pengguna berjalan dengan benar. Yang diuji antara lain adalah apakah controller dapat mengembalikan data yang sesuai, memproses data dengan tepat, dan melakukan pengalihan atau redirect dengan benar sesuai alur yang diinginkan.
2. Model berperan dalam mengelola data dan logika bisnis. Pengujian memastikan bahwa model memvalidasi data dengan benar dan menjalankan operasi seperti penyimpanan, pembaruan, atau penghapusan data sesuai desain.
3. Service berfungsi untuk menangani logika bisnis yang kompleks, termasuk interaksi dengan API eksternal. Pengujian memastikan bahwa proses bisnis dan respons dari API diproses dengan benar.
4. Helper adalah fungsi-fungsi pembantu untuk tugas-tugas umum seperti format tanggal atau manipulasi string. Pengujian memastikan fungsi memberikan hasil yang benar dan dapat menangani berbagai input dengan baik.

5.1.3.1 Rencana Pengujian

Rencana pengujian unit disusun untuk memastikan setiap fungsi pada sistem dapat bekerja sesuai spesifikasi sebelum digabungkan dengan modul lain. Rencana ini dituangkan dalam bentuk tabel yang berisi fokus pengujian, metode yang diuji, input yang diberikan, hasil yang diharapkan, serta keterangan singkat mengenai tujuan dari pengujian tersebut. Dengan adanya rencana pengujian, proses verifikasi menjadi lebih terarah dan memudahkan dalam pelacakan apabila ditemukan kesalahan. Sebagai contoh rencana pengujian ditampilkan pada Tabel 5.1

Tabel 5.1 Tabel Pengujian Unit User

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan
1	Profil by role	getProfil()	role=customer/admin/manager, relasi sesuai di-set	Mengembalikan relasi sesuai role	Set role → set relasi (customer/admin/manager) → panggil metode → verifikasi objek yang dikembalikan
2	Prioritas identitas	getNamaLengkap(), getNoHp(), getAlamat()	Profil ter-set; lalu override kolom di users	Mengambil dari users; fallback ke profil	Uji fallback ke profil, lalu override oleh kolom users
3	Cek role	isAdmin(), isManajer(), isCustomer()	role=admin/manager/customer	True sesuai role	Set role → panggil checker masing-masing → verifikasi boolean
4	Ambil id customer	getCustomerId()	role=customer, relasi customer.id_cust	"C-1"	Set role + relasi customer → panggil metode → cocokkan nilai

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan
			mer="C-1"		
5	Scopes query	scopeAdmins(), scopeCustomers(), scopeManagers(), scopeActive()	Mock Builder	Memanggil where yang sesuai, return builder	Verifikasi panggilan where('role',...) dan where('is_active',true)
6	Status verifikasi email	hasVerifiedEmail()	email_verified_at terisi / null	True jika terisi; False jika null	Set email_verified_at via Carbon (hindari DB) → cek boolean
7	Default UI	getDefaultPhoto(), getDefaultGreeting()	-	URL default photo; greeting default	Panggil metode → verifikasi string output
8	Avatar foto dengan fallback	getAvatar()	a) profil_photo di users; b) profil punya	a) URL storage users; b) URL storage profil; c) null	Uji tiga skenario berurutan, verifikasi masing-masing hasil

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan
			foto; c) kosong		
9	Inisial avatar	getAvatarInitials()	Profil nama_le ngkap="John Doe", "John", null	"JD", "JO", fallback_role "C"	Uji dua kata, satu kata, dan fallback role
10	Warna avatar	getAvatarColor()	role=admin/manager/customer	bg-gray-500 / bg-blue-500 / bg-green-500	Set role → panggil metode → verifikasi sesuai peta warna
11	Periode aktif manajer	getActivePeriod(), getManagerPeriodFormatted(), getManagerPeriodFormattedIndonesia()	role=manager, relasi Manajer dengan periode_akhir	Objek Manajer; format d/m/Y dan d M Y rentang tanggal	Set relasi manajer + tanggal via Carbon → verifikasi tiga output

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan
12	Total penyewaan	totalPenyewaan()	Mock relasi penyewaan()->count()=5	5	Stub relasi count → panggil metode → assert hasil
13	Periode admin	getAdminPeriodFormatted()	-	"Permanent"	Panggil metode → verifikasi string
14	Greeting admin/manager	getAdminGreeting(), getManagerGreeting()	-	Mengandung "Selamat"	Panggil dua metode → cek substring "Selamat"
15	Foto default admin	getAdminPhoto()	-	/images/admin.png	Panggil metode → verifikasi path default
16	Notifikasi belum dibaca	unreadNotifications()	Mock relasi notifications()->where('is_read',false)	Builder terfilter	Stub relasi → panggil metode → verifikasi argumen where

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan
17	Reset password notif	sendPasswordResetNotification()	Token "TOKEN123"	Memanggil notify() dengan notifikasi yang tepat	Mock notify(), panggil metode, verifikasi dipanggil
18	Tandai email terverifikasi	markEmailAsVerified()	Mock timestamp + save()	Return true; email_verified_at di-set	Mock freshTimestamp() dan save() → verifikasi hasil dan field

5.1.3.2 Kasus Uji dan Hasil Pengujian

Kasus uji merupakan implementasi dari rencana pengujian yang telah disusun sebelumnya. Pada tahap ini, setiap skenario uji dijalankan menggunakan *framework* Pest (Laravel Testing). Perbandingan antara hasil aktual dan ekspektasi menjadi acuan dalam menilai keberhasilan fungsi yang diuji. Jika ditemukan perbedaan, dilakukan analisis lebih lanjut serta perbaikan pada bagian kode yang terkait. Contoh hasil implementasi pengujian unit pada modul User dapat dilihat pada potongan kode dan Gambar 5.17

```

<?php

namespace App\Models {
    if (!function_exists('__NAMESPACE__ . '\\asset')) {
        function asset($path) {
            return 'http://localhost/' . ltrim($path, '/');
        }
    }
}

namespace {
    use App\Models\User;
    use App\Models\Manager;
    use App\Models\Customer;
    use App\Models\Admin;
    use Carbon\Carbon;
    use Illuminate\Database\Eloquent\Builder;
    use Mockery;

    test('1. getProfile() mengembalikan relasi sesuai role', function () {
        $user = new User();
        $user->role = 'customer';
        $customer = new Customer();
        $user->setRelation('customer', $customer);
        expect($user->getProfile())->toBe($customer);

        $user->role = 'admin';
        $admin = new Admin();
        $user->setRelation('admin', $admin);
        expect($user->getProfile())->toBe($admin);

        $user->role = 'manager';
        $manager = new Manager();
        $user->setRelation('manager', $manager);
        expect($user->getProfile())->toBe($manager);
    });

    test('2. getNamaLengkap/NoHp/Alamat prioritas users lalu profile', function () {
        $user = new User();
        $user->role = 'customer';
        $profile = new Customer();
        $profile->nama_lengkap = 'From Profile';
        $profile->no_hp = '0800';
        $profile->alamat = 'Profile Addr';
        $user->setRelation('customer', $profile);

        expect($user->getNamaLengkap())->toBe('From Profile');
        expect($user->getNoHp())->toBe('0800');
        expect($user->getAlamat())->toBe('Profile Addr');

        $user->nama_lengkap = 'From User';
        $user->no_hp = '0812';
        $user->alamat = 'User Addr';
        expect($user->getNamaLengkap())->toBe('From User');
        expect($user->getNoHp())->toBe('0812');
        expect($user->getAlamat())->toBe('User Addr');
    });

    test('3. isAdmin/isManager/isCustomer berdasarkan role', function () {
        $user = new User();
        $user->role = 'admin';
        expect($user->isAdmin())->toBeTrue();

        $user->role = 'manager';
        expect($user->isManager())->toBeTrue();

        $user->role = 'customer';
        expect($user->isCustomer())->toBeTrue();
    });

    test('4. getCustomerId() mengembalikan id_customer saat role customer', function () {
        $user = new User();
        $user->role = 'customer';
        $customer = new Customer();
        $customer->id_customer = 'C-1';
        $user->setRelation('customer', $customer);

        expect($user->getCustomerId())->toBe('C-1');
    });

    test('5. Scopes role dan active memanggil where yang tepat', function () {
        $builder = Mockery::mock(Builder::class);
        $builder->shouldReceive('where')->once()->with('role', 'admin')->andReturnSelf();
        $builder->shouldReceive('where')->once()->with('role', 'customer')->andReturnSelf();
        $builder->shouldReceive('where')->once()->with('role', 'manager')->andReturnSelf();
        $builder->shouldReceive('where')->once()->with('is_active', true)->andReturnSelf();

        $u = new User();
        expect($u->scopeAdmins($builder))->toBe($builder);
        expect($u->scopeCustomers($builder))->toBe($builder);
        expect($u->scopeManagers($builder))->toBe($builder);
        expect($u->scopeActive($builder))->toBe($builder);
    });
}

```

Gambar 5.17 Code Pengujian Unit User

```

test('6. hasVerifiedEmail true jika email_verified_at terisi', function () {
    $u = new User();
    $ref = new \ReflectionClass($u);
    $prop = $ref->getProperty('dateFormat');
    $prop->setAccessible(true);
    $prop->setValue($u, 'Y-m-d H:i:s');
    $u->email_verified_at = Carbon::now();
    expect($u->hasVerifiedEmail())->toBeTrue();

    $u->email_verified_at = null;
    expect($u->hasVerifiedEmail())->toBeFalse();
});

test('7. getDefaultPhoto dan getDefaultGreeting', function () {
    $u = new User();
    expect($u->getDefaultPhoto())->toBe('http://localhost/images/managerL.png');
    expect($u->getDefaultGreeting())->toBe('Selamat datang!');
});

test('8. getAvatar: gunakan users.profile_photo jika ada, fallback ke profile, selain itu null', function () {
    $u = new User();

    $u->profile_photo = 'avatars/u.jpg';
    expect($u->getAvatar())->toBe('http://localhost/storage/avatars/u.jpg');

    $u = new User();
    $u->role = 'customer';
    $profile = new Customer();
    $profile->profile_photo = 'profiles/c.png';
    $u->setRelation('customer', $profile);
    expect($u->getAvatar())->toBe('http://localhost/storage/profiles/c.png');

    $u = new User();
    expect($u->getAvatar())->toBeNull();
});

test('9. getAvatarInitials: dari profile.nama_lengkap atau fallback role', function () {
    $u = new User();
    $u->role = 'customer';
    $profile = new Customer();
    $profile->nama_lengkap = 'John Doe';
    $u->setRelation('customer', $profile);
    expect($u->getAvatarInitials())->toBe('JD');

    $profile->nama_lengkap = 'John';
    expect($u->getAvatarInitials())->toBe('JO');

    $profile->nama_lengkap = null;
    expect($u->getAvatarInitials())->toBe('C');
});

test('10. getAvatarColor berdasarkan role', function () {
    $u = new User();
    $u->role = 'admin';
    expect($u->getAvatarColor())->toBe('bg-gray-500');
    $u->role = 'manager';
    expect($u->getAvatarColor())->toBe('bg-blue-500');
    $u->role = 'customer';
    expect($u->getAvatarColor())->toBe('bg-green-500');
});

```

Gambar 5.18 Code Pengujian Unit User (1)

```

test('11. getActivePeriod & Manager period formatted', function () {
    $u = new User();
    $u->role = 'manager';

    $m = new Manager();
    $ref = new \ReflectionClass($m);
    $prop = $ref->getProperty('dateFormat');
    $prop->setAccessible(true);
    $prop->setValue($m, 'Y-m-d H:i:s');
    $m->periode_awal = Carbon::create(2025, 1, 1);
    $m->periode_akhir = Carbon::create(2025, 1, 31);
    $u->setRelation('manager', $m);

    expect($u->getActivePeriod()->toBe($m);
    expect($u->getManagerPeriodFormatted()->toBe('01/01/2025 - 31/01/2025'));
    expect($u->getManagerPeriodFormattedIndonesia()->toBe('01 Jan 2025 - 31 Jan 2025'));
});

test('12. totalPenyewaan menggunakan count() dari relasi (mock)', function () {
    $u = Mockery::mock(User::class)->makePartial();
    $u->shouldReceive('penyewaan')->andReturn(new class {
        public function count() { return 5; }
    });

    expect($u->totalPenyewaan()->toBe(5));
});

test('13. getAdminPeriodFormatted mengembalikan Permanent', function () {
    $u = new User();
    expect($u->getAdminPeriodFormatted()->toBe('Permanent'));
});

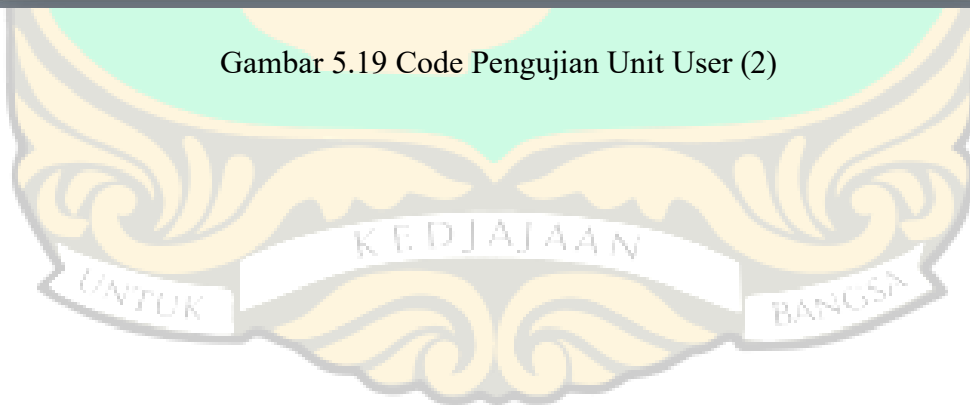
test('14. getAdminGreeting() dan getManagerGreeting()', function () {
    $u = new User();
    $u->role = 'admin';
    expect($u->getAdminGreeting()->toContain('Selamat'));

    $u->role = 'manager';
    expect($u->getManagerGreeting()->toContain('Selamat'));
});

test('15. getAdminPhoto() mengembalikan path default admin', function () {
    $u = new User();
    expect($u->getAdminPhoto()->toBe('/images/admin.png'));
});

```

Gambar 5.19 Code Pengujian Unit User (2)



```
test('16. unreadNotifications memanggil where is_read = false (mock)', function () {
    $u = Mockery::mock(User::class)->makePartial();
    $u->shouldReceive('notifications')->andReturn(new class {
        public array $called = [];
        public function where($key, $val) { $this->called = [$key, $val]; return $this; }
    });

    $rel = $u->unreadNotifications();
    expect($rel->called)->toBe(['is_read', false]);
});

test('17. sendPasswordResetNotification memanggil notify (mock)', function () {
    $u = Mockery::mock(User::class)->makePartial();
    $u->shouldReceive('notify')->once();
    >with(Mockery::type(\App\Notifications\CustomResetPasswordNotification::class));
    $u->sendPasswordResetNotification('TOKEN123');
    expect(true)->toBeTrue();
});

test('18. markEmailAsVerified menyetel email_verified_at dan memanggil save (mock)', function () {
    $u = Mockery::mock(User::class)->makePartial();
    $ref = new \ReflectionClass($u);
    $prop = $ref->getProperty('dateFormat');
    $prop->setAccessible(true);
    $prop->setValue($u, 'Y-m-d H:i:s');
    $u->shouldReceive('freshTimestamp')->andReturn(Carbon::create(2025,1,1,0,0,0));
    $u->shouldReceive('save')->once()->andReturn(true);

    $result = $u->markEmailAsVerified();
    expect($result)->toBeTrue();
    expect($u->email_verified_at)->not->toBeNull();
});
}
```

Gambar 5.20 Code Pengujian Unit User (3)

```
PASS Tests\Unit\Models\UserTest
✓ 1. getProfile() mengembalikan relasi sesuai role
✓ 2. getNamaLengkap/NoHp/Alamat prioritas users lalu profile
✓ 3. isAdmin/isManager/isCustomer berdasarkan role
✓ 4. getCustomerId() mengembalikan id_customer saat role customer
✓ 5. Scopes role dan active memanggil where yang tepat
✓ 6. hasVerifiedEmail true jika email_verified_at terisi
✓ 7. getDefaultPhoto dan getDefaultGreeting
✓ 8. getAvatar: gunakan users.profile_photo jika ada, fallback ke profile, selain itu null
✓ 9. getAvatarInitials: dari profile.nama_lengkap atau fallback role
✓ 10. getAvatarColor berdasarkan role
✓ 11. getActivePeriod & Manager period formatted
✓ 12. totalPenyewaan menggunakan count() dari relasi (mock)
✓ 13. getAdminPeriodFormatted mengembalikan Permanent
✓ 14. getAdminGreeting() dan getManagerGreeting()
✓ 15. getAdminPhoto() mengembalikan path default admin
✓ 16. unreadNotifications memanggil where is_read = false (mock)
✓ 17. sendPasswordResetNotification memanggil notify (mock)
✓ 18. markEmailAsVerified menyetel email_verified_at dan memanggil save (mock)

Tests: 18 passed (48 assertions)
Duration: 0.71s
```

Gambar 5.21 Berhasil Kasus Uji dan Hasil Pengujian Unit User

5.1.3.1 Kesimpulan Hasil Pengujian

Setelah dilakukan pengujian unit terhadap fungsi-fungsi pada Model User, seluruh skenario uji berhasil dijalankan dengan hasil yang sesuai dengan ekspektasi. Tabel berikut merangkum fokus pengujian, metode yang diuji, input yang diberikan, hasil yang diharapkan, serta status pengujian. Berdasarkan hasil uji, seluruh metode dinyatakan lulus, yang berarti fungsi pada modul ini dapat berjalan dengan benar sesuai spesifikasi. Contohnya dapat dilihat pada Tabel 5.2

Tabel 5. 2 Kesimpulan Hasil Pengujian Unit

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Hasil
1	Ambil profil sesuai role	getProfil()	role: customer/admin/manajer; relasi diset	Kembalikan instance profil sesuai role	Prioritas relasi yang cocok dengan nilai role.	Lulus
2	Data nama/HP/alamat (prioritas)	getNamaLengkap(), getNoHp(), getAlamat()	Kolom di users kosong lalu diisi; profil punya data	Ambil dari users jika ada; jika tidak, fallback ke profil	Menjamin tampilan konsisten meski data tersebar.	Lulus
3	Cek role	isAdmin(), isManajer(), isCustomer()	role='admin' / 'manajer' / 'customer'	True sesuai role	Pemeriksaan peran untuk otorisasi/fitur.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Hasil
4	ID customer	getCustomerId()	role='customer'; relasi customer.id = 'C-1'	"C-1"	Mengambil identifier profil customer dari relasi.	Lulus
5	Scopes role & aktif	scopeAdmins(), scopeCustomers(), scopeManagers(), scopeActive()	Builder mock where(...)	Kembali builder (chainable)	Memudahkan query filter berdasarkan role/aktif.	Lulus
6	Email terverifikasi	hasVerifiedEmail()	email_verified_at terisi/null	True jika terisi	Gunakan Carbon + reflection dateFormat agar tanpa DB.	Lulus
7	Default photo & greeting	getDefaultPhoto(), getDefaultGreeting()	-	"/images/manajerL.png"; "Selamat datang!"	Nilai default untuk UI saat data belum ada.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Hasil
8	Avatar URL prioritas	getAvatar()	users.profil _photo ada; lalu hanya di profil; lalu kosong	URL dari users; fallback ke profil; null jika tidak ada	Gunakan helper asset() (dimock) untuk URL storage.	Lulus
9	Avatar initials	getAvatarInitials()	role='customer'; nama "John Doe", "John", null	"JD"; "JO"; "C"	Ekstrak 2 huruf; fallback huruf role bila nama kosong.	Lulus
10	Warna avatar	getAvatarColor()	role=admin/ manajer/customer	bg-gray- 500/bg- blue- 500/bg- green-500	Warna konsisten per role untuk UI.	Lulus
11	Periode manajer aktif + format	getActivePeriod(), getManagerPeriodFormatted(), getManagerPeriodFor	role=manajer; periode 2025-01- 01..31	Kembalikan instance; "01/01/2025 - 31/01/2025" ; "01 Jan 2025 - 31 Jan 2025"	Ambil relasi manajer dan format tanggal Indonesia.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Hasil
		mattedIndonesia()				
12	Total penyewaan user	totalPenyewaan()	mock relasi penyewaan()->count()=5	5	Statistik sederhana untuk dashboard.	Lulus
13	Periode admin	getAdminPeriodFormatted()	-	"Permanent"	Admin dianggap permanen.	Lulus
14	Greetings role	getAdminGreeting(), getManagerGreeting()	role='admin' / 'manajer'	String mengandung "Selamat"	Salam dinamis sesuai waktu.	Lulus
15	Foto admin default	getAdminPhoto()	-	"/images/admin.png"	Path default jika admin tanpa foto.	Lulus
16	Notifikasi unread	unreadNotifications()	mock notifications()->where('is_read',false)	Relasi terfilter	Untuk badge jumlah unread.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Hasil
17	Kirim reset password	sendPasswordResetNotification()	mock notify(CustomResetPasswordNotification)	Pemanggilan notify terjadi	Menggunakan custom notification class.	Lulus
18	Verifikasi email	markEmailAsVerified()	mock save(); set timestamp	True; email_verified_at terisi	Set waktu sekarang dan simpan.	Lulus

Berdasarkan hasil pengujian unit pada *Model User* yang dirangkum pada Tabel 5.2, seluruh metode yang diuji berhasil menghasilkan keluaran sesuai dengan spesifikasi yang diharapkan. Hal ini menunjukkan bahwa fungsi-fungsi inti, seperti pengecekan peran pengguna, pewarnaan avatar, serta pesan sambutan (*greeting*), telah berjalan dengan benar. Dengan demikian, dapat disimpulkan bahwa *modul User* memenuhi kebutuhan fungsional pada tahap pengujian unit dan siap untuk diintegrasikan dengan modul lain dalam sistem.

5.2 Integrasi dan Pengujian Sistem

Tahap pengujian integrasi dilakukan untuk memastikan bahwa seluruh fungsi sistem bekerja sesuai kebutuhan dan spesifikasi ketika modul-modul digabungkan. Pengujian ini mencakup pemeriksaan alur logika program, interaksi antarmuka, serta validasi keluaran sistem terhadap masukan yang diberikan. Metode yang digunakan adalah *black box testing*, yaitu memeriksa perilaku sistem berdasarkan input dan output tanpa memperhatikan struktur kode internal. Setiap pengujian didokumentasikan sebagai bukti bahwa sistem telah divalidasi.

5.2.1 Fokus Pengujian

Tahap ini berfokus pada pengujian fungsi utama sistem sesuai kebutuhan pengguna. Uji coba dilakukan dengan memperhatikan kemudahan penggunaan, kejelasan informasi, serta kesesuaian fungsionalitas. Berikut adalah item-item yang diuji pada Tabel 5.3

Tabel 5.3 Tabel Fokus Pengujian Aplikasi

No	Item yang diuji	Pengguna	Detail
1	Melakukan registrasi	Pelanggan	Registrasi akun baru dengan nama, email, password, alamat, jenis kelamin dan no hp
2	Login dengan email dan password valid	Admin, pelanggan, manajer	Login ke sistem sesuai role
3	Melakukan reset password dengan email	Admin, pelanggan, manajer	Reset password via email
4	Logout	Admin, pelanggan, manajer	Keluar dari sistem, akhiri sesi login
5	Mengakses <i>dashboard</i> admin	Admin	Dapat akses semua menu admin
6	Mengakses <i>dashboard</i> pelanggan	Pelanggan	Dapat akses semua menu pelanggan

No	Item yang diuji	Pengguna	Detail
7	Mengakses <i>dashboard</i> manajer	Manger	Dapat akses semua menu manajer
8	Lihat Profil	Admin, pelanggan, manajer	Lihat profil
9	Update Profil	Admin, pelanggan, manajer	Update data yang ada di profil kecuali email
10	Edit Password	Admin, pelanggan, manajer	Edit password
11	Upload Foto	Admin, pelanggan, manajer	Upload foto profil
12	Lihat Daftar Layanan	pelanggan	Lihat katalog layanan, detail fasilitas, harga, jadwal
13	Melakukan pemesanan	pelanggan	Pilih layanan, tanggal, waktu, fasilitas tambahan, metode pembayaran
14	Cek Ketersediaan	pelanggan	Cek slot tersedia
15	Lihat Status pemesanan	pelanggan	Lihat status penyewaan

No	Item yang diuji	Pengguna	Detail
16	Lihat Riwayat Penyewaan	pelanggan	Lihat riwayat penyewaan
17	Upload Bukti	pelanggan	Unggah bukti pembayaran setelah pembayaran
18	Memberikan umpan balik	pelanggan	Berikan ulasan, rating setelah menggunakan layanan
19	Chat dengan pelanggan	pelanggan	Akses informasi personal dengan autentikasi
20	Chatbot untuk Guest	Guest	Dapat informasi dasar layanan tanpa registrasi
21	Lihat Riwayat Penyewaan	pelanggan	Lihat riwayat penyewaan, status pembayaran, detail transaksi
22	Lihat statistik <i>dashboard</i>	Manajer	Lihat ringkasan data penyewaan, pendapatan, umpan balik, statistik
23	Download laporan penyewaan	Manajer	Download laporan dalam format yang dapat dianalisis
24	Monitor Penyewaan	Manajer	Monitor Penyewaan
25	Lihat Umpan balik Pelanggan	manajer	Lihat Umpan balik Pelanggan

No	Item yang diuji	Pengguna	Detail
26	Kelola Layanan	Admin	Tambah, edit, hapus, atur status, harga, deskripsi, gambar
27	Kelola Jadwal Libur	Admin	Atur jadwal libur operasional koperasi
28	Verifikasi	Admin	Verifikasi bukti transfer, ubah status pembayaran
29	Kelola akun user manajer	Admin	Daftar manajer baru, nonaktifkan, reset password, perpanjang periode
30	Kelola akun user pelanggan	Admin	Aktifkan/nonaktifkan pelanggan
31	Kelola refund	Admin	Proses <i>refund Request</i> , approve, dan melakukan verifikasi <i>request</i> , serta melakukan transfer pengembalian uang pelanggan.
32	Lihat dan balas umpan balik pelanggan	Admin	Balas umpan balik pelanggan
33	Lihat riwayat penyewaan	Admin	Akses seluruh riwayat transaksi untuk audit dan pelaporan

5.2.2 Kasus Uji dan Hasil Pengujian

Pada tahap ini akan menampilkan hasil pengujian sistem yang dilaksanakan berdasarkan skenario uji (*test case*) yang telah disusun sebelumnya. Pengujian

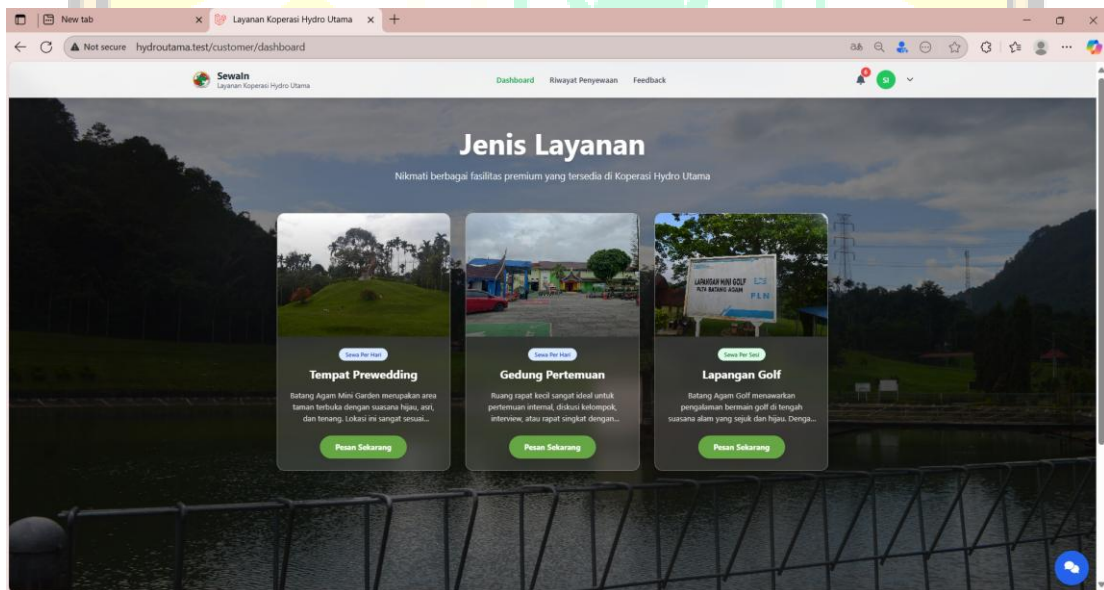
dilakukan dengan pendekatan *black box testing*, di mana penekanan diberikan pada hubungan antara data *input* dengan *output* yang dihasilkan oleh sistem tanpa memperhatikan struktur internal program. Setiap skenario pengujian mencakup data input, ekspektasi hasil, serta pencatatan terhadap hasil aktual sistem. Selain itu, bagian ini akan menjelaskan skenario pengujian utama yang mencakup proses pemesanan. Pada tahap pemesanan, pengguna dapat memilih sesi berdasarkan hari maupun jenis sesi yang tersedia.



5.2.2.1 Kasus Hasil Pengujian Melakukan Pemesanan Layanan

Kasus pengujian ini dibuat untuk memastikan bahwa alur pemesanan layanan sudah berjalan sesuai dengan kebutuhan yang ditetapkan. Pada tahap ini, pelanggan diberi opsi untuk memilih layanan yang mau dipilih dan tipe sewa yang tersedia, baik sewa harian maupun sewa per sesi. Pengujian difokuskan untuk mengecek apakah sistem dapat menerima input pemesanan, menampilkan pilihan tipe sewa dengan benar, serta menghasilkan output yang sesuai dengan skenario yang telah dirancang.

Pada Gambar 5.21 halaman ini ditampilkan katalog layanan yang tersedia di Koperasi Konsumen Hydro Utama. Setiap layanan ditampilkan dalam bentuk *card* informasi yang berisi deskripsi singkat serta tombol aksi untuk melakukan pemesanan. Jenis layanan yang ditawarkan mencakup lokasi foto *pre wedding*, Gedung Pertemuan, dan Lapangan Golf. Masing-masing layanan mendukung dua tipe sewa, yaitu berdasarkan sesi dan berdasarkan hari penuh. Dengan adanya opsi ini, pengguna dapat menyesuaikan kebutuhan pemakaian layanan, apakah hanya untuk periode waktu tertentu (per sesi) atau untuk keseluruhan hari.



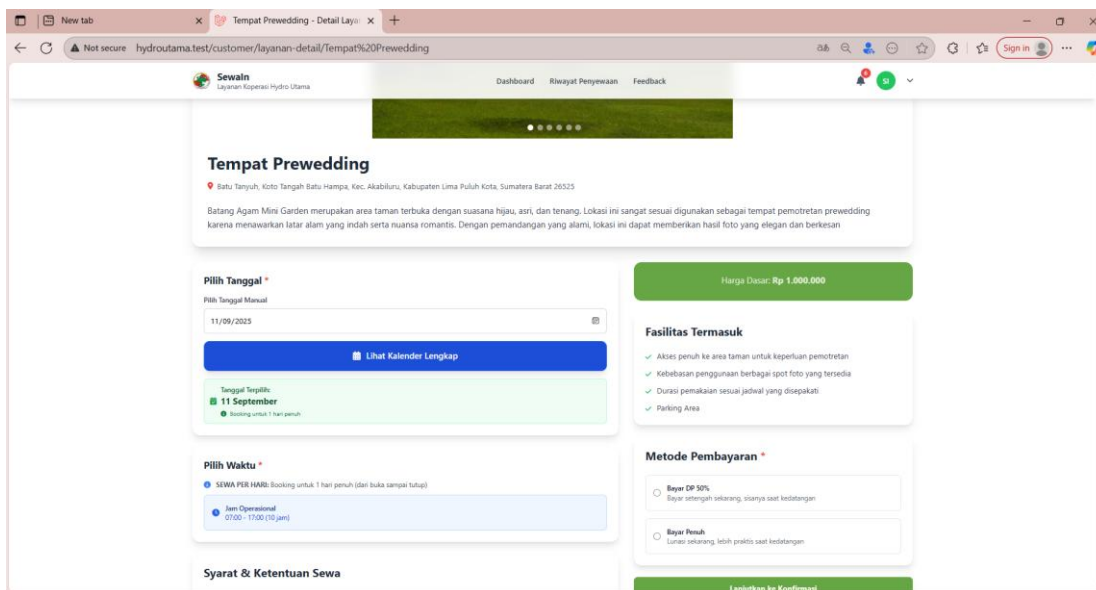
Gambar 5.22 Halaman *Dashboard* Pelanggan

Pada pengujian skenario ini bertujuan untuk memastikan sistem mampu memproses pemesanan dengan tipe sewa harian secara benar. Dalam kasus ini,

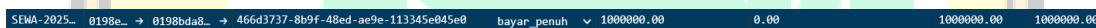
pengguna memilih tanggal tertentu untuk melakukan sewa harian. Sistem diharapkan dapat menerima input tanggal, memvalidasi ketersediaan layanan, dan menampilkan informasi apakah tanggal tersebut masih tersedia. Jika tersedia, pemesanan diproses dan dicatat ke dalam sistem. Pengujian ini penting untuk memastikan alur dasar pemesanan harian dapat berjalan tanpa kendala. Dapat dilihat pada Tabel 5.4, Gambar 5.22 dan Gambar 2.23.

Tabel 5.4 Pengujian Pemesanan Tipe Harian

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 11 September, Tipe Harian
Hasil Yang Diharapkan	Sistem menerima input, menampilkan ketersediaan, dan menyimpan pemesanan
Pengamatan	Sistem berhasil menampilkan tanggal tersedia dan pemesanan tercatat
Hasil	Sesuai



Gambar 5.23 Tampilan Pelanggan Melakukan Pemesanan Dengan Tipe Harian



Gambar 5. 24 Bukti penambahan pelanggan baru untuk tipe harian

Selanjutnya, pengujian dilakukan dengan skenario dirancang untuk menguji bagaimana sistem menangani kondisi ketika pelanggan mencoba melakukan pemesanan harian pada tanggal yang sudah pernah digunakan. Proses pengujian dilakukan dengan memasukkan tanggal yang sama untuk dua kali pemesanan. Sistem diharapkan dapat mendeteksi konflik tersebut, kemudian menolak permintaan atau memberikan pemberitahuan bahwa tanggal sudah tidak tersedia. Dapat dilihat pada Tabel 5.5 dan Gambar 5.24.

Tabel 5.5 Pengujian Pemesanan Harian Pada Tanggal yang Sama

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 11 September, Tipe Harian (sudah terisi)

Kasus dan Hasil Uji	
Hasil Yang Diharapkan	Sistem menolak pemesanan dan memberikan pemberitahuan “Tanggal sudah dibooking”
Pengamatan	Sistem memberikan pemberitahuan sesuai harapan
Hasil	Sesuai

The screenshot shows a web application interface for booking a prewedding venue. The main heading is 'Tempat Prewedding'. Below it, the venue name 'Batang Agam Mini Garden' is listed along with its address. A form for selecting a date and time is present. The date '11/09/2023' is selected, and the time '11 September' is highlighted. A message indicates that the selected date is already booked. The page also displays the venue name 'Batang Agam Mini Garden', the price 'Rp 1.000.000', and the payment method 'Bayar DP 50%'. A 'Lanjutkan ke Konfirmasi' button is visible at the bottom.

Gambar 5.25 Tampilan Pelanggan Mencoba Memesan Tanggal yang Sudah Dipesan Sebelumnya (Harian)

Selanjutnya, pada pengujian ini bertujuan memastikan sistem dapat melakukan pengecekan (validasi) terhadap kelengkapan data sebelum pelanggan menekan tombol konfirmasi layanan. Skenario dilakukan dengan cara mencoba menyimpan pemesanan ketika ada data yang belum terisi, misalnya tidak memilih metode pembayaran atau melewati pilih jadwal. Sistem diharapkan memberikan peringatan yang jelas mengenai

data yang belum dilengkapi, serta mencegah pemesanan diproses lebih lanjut sampai semua informasi diisi dengan benar. Dapat dilihat pada Tabel 5.6 dan Gambar 5.25.

Tabel 5.6 Pengujian Validasi Data Pemesanan pada Halaman Detail Layanan

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 24 September, Tipe Harian, Metode pembayaran tidak dipilih
Hasil Yang Diharapkan	Sistem menolak konfirmasi dan menampilkan validasi “Mohon Lengkapi data yang belum dipilih seperti metode pembayaran dahulu”
Pengamatan	Sistem menolak dan memberikan notifikasi validasi
Hasil	Sesuai

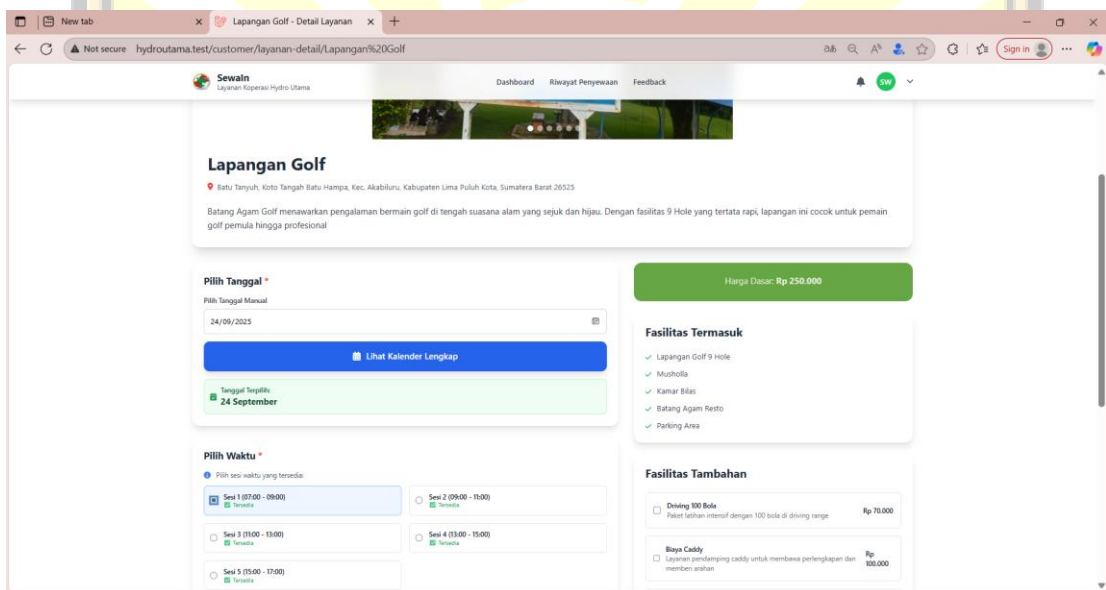
Gambar 5. 26 Tampilan Pelanggan Mencoba Memesan Tapi Tahapan Pemesanannya Tidak Lengkap

Kemudian, pada pengujian selanjutnya ini akan difokuskan pada layanan pemesanan per sesi, di mana pengguna dapat memilih waktu tertentu dalam satu hari. Sistem diuji apakah mampu menampilkan daftar sesi yang tersedia, menerima input dari pengguna, serta mencatat pemesanan sesuai sesi yang dipilih. Jika sesi masih kosong, sistem harus mengkonfirmasi keberhasilan pemesanan. Tabel 5.7, Gambar 5.26. dan Gambar 2.27.

Tabel 5.7 Pengujian Pemesanan Tipe Sesi

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 24 September, Tipe Sesi (Sesi 1)
Hasil Yang Diharapkan	Sistem menerima input, menampilkan ketersediaan sesi, dan mencatat pemesanan

Kasus dan Hasil Uji	
Pengamatan	Sistem berhasil menyimpan pemesanan untuk sesi 1
Hasil	Sesuai



Gambar 5.27 Tampilan Pelanggan Melakukan Pemesanan Dengan Tipe Sesi Yang Masih Kosong

layan	total_fasilitas_tambahan	total_harga	nominal_dp	sisa_pembayaran	batas_pembayaran_dp	batas_pelunasan	jam_mulai	jam_selesai
100000.00		350000.00	350000.00	0.00	NULL	NULL	07:00:00	09:00:00

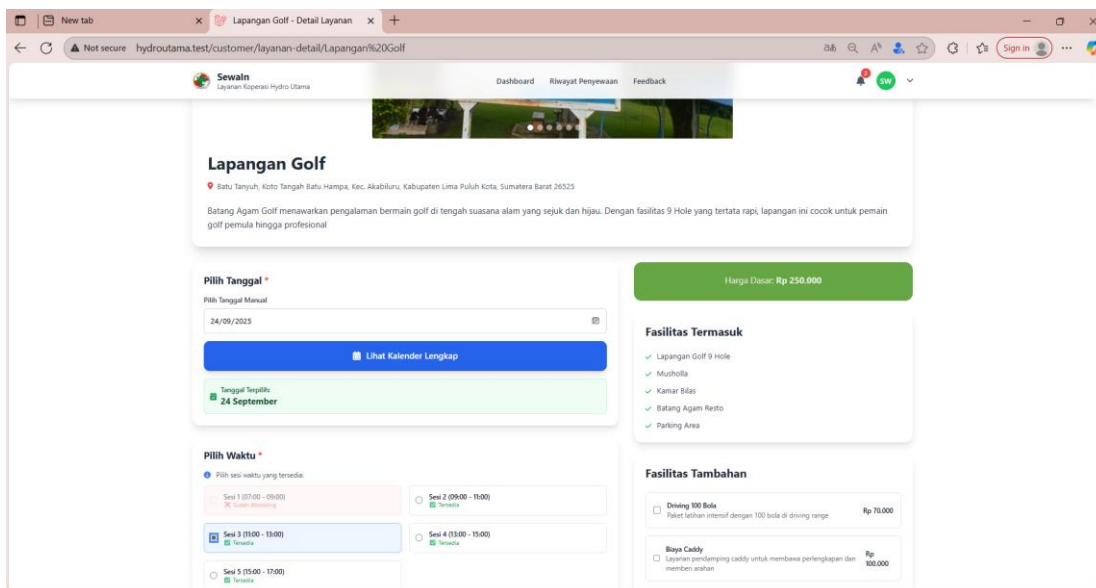
Gambar 5.28 Bukti penambahan pelanggan baru untuk tipe sesi

Selain itu, dilakukan pengujian untuk memverifikasi apakah sistem mengizinkan pengguna melakukan lebih dari satu pemesanan pada hari yang sama, selama sesi yang dipilih berbeda. Sistem hanya memperbolehkan satu pemesanan dalam satu transaksi (satu sesi per pesanan). Oleh karena itu, apabila pelanggan ingin melakukan pemesanan tambahan pada hari yang sama untuk sesi berbeda, pelanggan harus menyelesaikan transaksi pertama terlebih dahulu. Sistem diharapkan tetap

menerima pemesanan baru setelah transaksi pertama berhasil diselesaikan. Tabel 5.8 dan Gambar 5.28.

Tabel 5.8 Pengujian Pemesanan Sesi Berbeda pada Hari Sama

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 24 September, Tipe Sesi (Sesi 3) yang dimana sebelumnya udah melakukan pemesanan tanggal 24 September Tipe Sesi (Sesi 1)
Hasil Yang Diharapkan	Sistem menerima pemesanan kedua setelah transaksi pertama selesai
Pengamatan	Sistem berhasil mencatat sesi selanjutnya dengan transaksi terpisah
Hasil	Sesuai



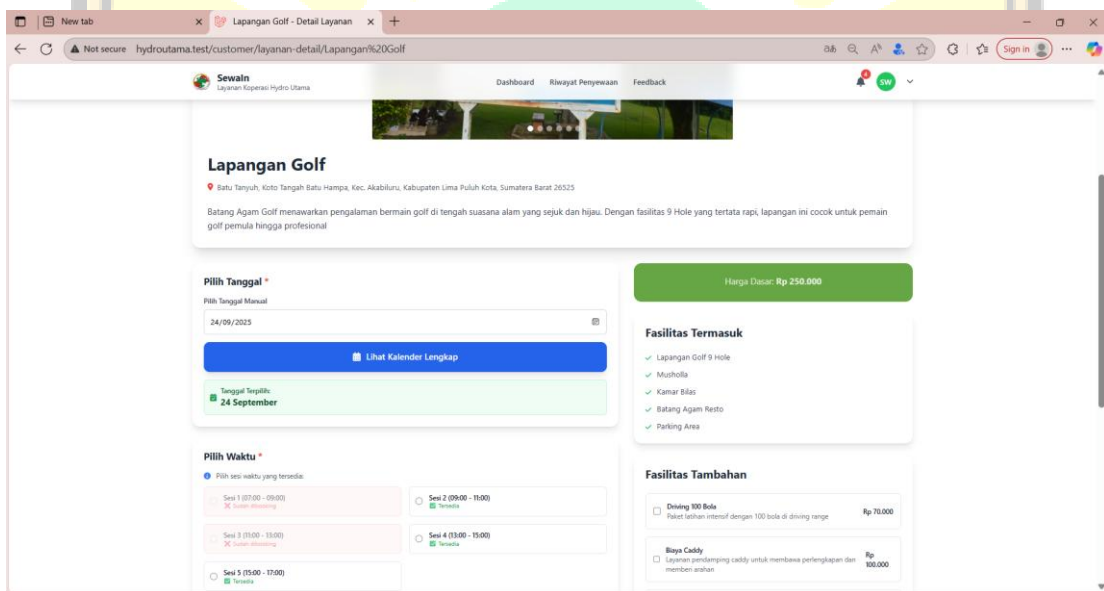
Gambar 5.29 Tampilan Pelanggan Melakukan Pemesanan Sesi Berbeda Pada Hari yang Sama

Berikutnya, pada pengujian ini dilakukan untuk memastikan sistem mampu mengantisipasi kondisi ketika pelanggan mencoba melakukan pemesanan pada hari dan sesi yang sebenarnya sudah terisi sebelumnya. Pada implementasinya, sistem tidak menunggu hingga proses pemesanan dijalankan untuk kemudian menolak, tetapi langsung melakukan pencegahan melalui antarmuka. Sesi yang sudah penuh akan otomatis ditandai sebagai tidak aktif (*disabled*), sehingga pengguna tidak dapat memilihnya. Dengan cara ini, sistem dapat mencegah terjadinya pemesanan ganda. Dapat dilihat Tabel 5.9 dan Gambar 5.29.

Tabel 5.9 Pengujian Pemesanan Sesi Sama pada Hari Sama

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 24 September, Tipe Sesi (Sesi 1 dan sesi 3) sudah dipesan
Hasil Yang Diharapkan	Sistem menonaktifkan opsi sesi yang sudah terisi sehingga tidak bisa dipilih

Kasus dan Hasil Uji	
Pengamatan	Sesi 1 dan 3 otomatis non aktif di tampilan, hanya sesi kosong yang bisa dipilih
Hasil	Sesuai



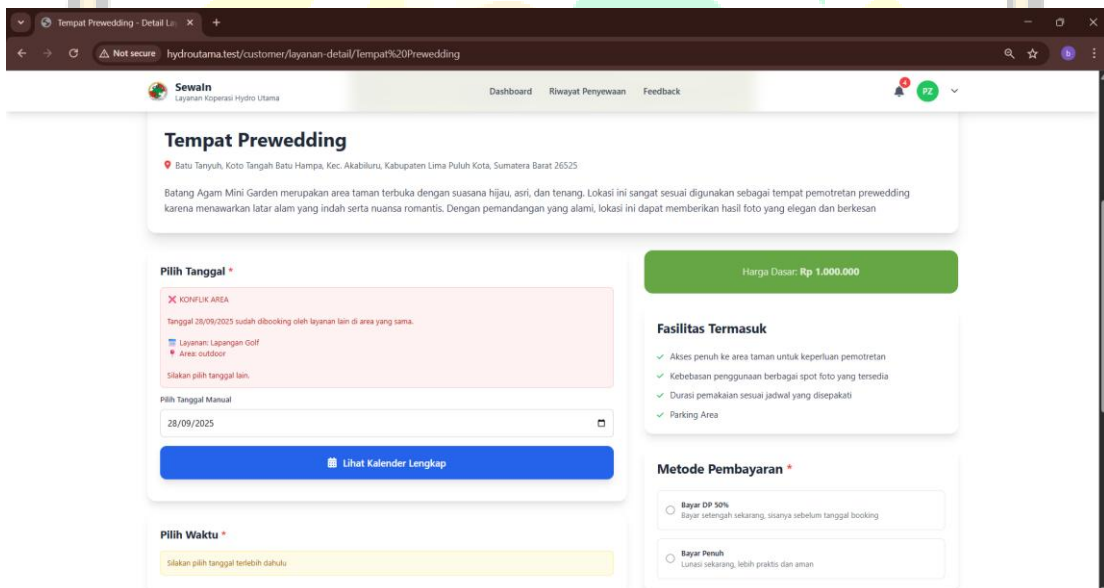
Gambar 5.30 Tampilan Pelanggan Melakukan Pemesanan Sesi Sama Pada Hari yang Sama

Selanjutnya, pada pengujian ini dirancang untuk memastikan sistem mampu mengantisipasi konflik ketika dua jenis layanan yang berbeda menggunakan area (*indoor* dan *outdoor*) yang sama pada tanggal yang sama. Contoh kasusnya adalah lapangan golf dan lokasi foto *pre wedding*. Apabila salah satu layanan sudah terjadwal, maka sistem harus mendeteksi adanya konflik sejak proses pemesanan, kemudian menolak pemesanan baru sekaligus memberikan validasi yang menjelaskan penyebab penolakan tersebut. Dengan mekanisme ini, sistem dapat menjamin bahwa fasilitas

hanya digunakan untuk satu layanan pada waktu tertentu sehingga tidak terjadi benturan jadwal maupun penggunaan ruang. Dapat dilihat Tabel 5.10 dan Gambar 5.30.

Tabel 5.10 Pengujian Pemesanan Dengan Konflik Antar Layanan

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 24 September, Tipe Sesi udah ada yang mencoba memesan lapangan golf
Hasil Yang Diharapkan	Sistem menolak pemesanan kedua dan menampilkan validasi konflik
Pengamatan	Sistem menolak pemesanan lokasi foto <i>pre wedding</i> dan hanya mengizinkan salah satu layanan
Hasil	Sesuai

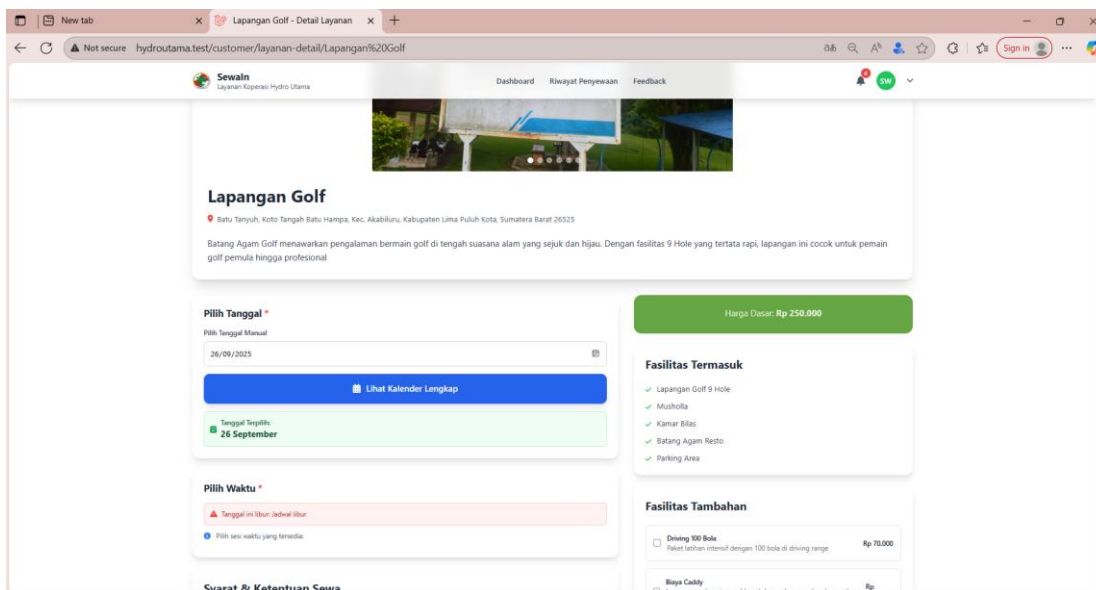


Gambar 5.31 Tampilan Pelanggan Melakukan Pemesanan dengan Konflik Antar Layanan Lapangan Golf dan lokasi foto *pre wedding*

Selanjutnya, pada pengujian ini difokuskan validasi pemesanan yang dilakukan pada tanggal yang telah ditandai sebagai libur oleh admin melalui pengaturan sistem. Penetapan tanggal libur ini biasanya berasal dari kebijakan koperasi, seperti hari libur nasional, pemeliharaan fasilitas, atau agenda internal tertentu lainnya. Sistem diharapkan dapat membaca konfigurasi tersebut dan secara otomatis menolak setiap pemesanan yang dilakukan pada tanggal yang sudah dinonaktifkan. Selain itu, sistem juga harus menampilkan validasi yang jelas agar pelanggan memahami bahwa tanggal tersebut memang tidak tersedia untuk digunakan. Dapat dilihat Tabel 5.11 dan Gambar 5.31.

Tabel 5.11 Pengujian Pemesanan Pada Tanggal Libur

Kasus dan Hasil Uji	
Data Masukkan	Pilih tanggal 26 September, pada jadwal libur (libur ditentukan admin koperasi)
Hasil Yang Diharapkan	Sistem menolak pemesanan dan memberikan pemberitahuan bahwa tanggal tersebut tidak tersedia
Pengamatan	Sistem menolak pemesanan dengan pesan error “Tanggal ini libur”
Hasil	Sesuai



Gambar 5.32 Tampilan Pelanggan Melakukan Pemesanan Pada Tanggal Libur

5.2.3 Kesimpulan Hasil Pengujian

Pengujian yang telah dilakukan bertujuan untuk memastikan bahwa setiap fungsi dalam sistem berjalan sesuai kebutuhan dan spesifikasi yang ditetapkan. Hasil pengujian ini menjadi dasar dalam menilai apakah aplikasi sudah memenuhi tujuan pengembangan dan siap digunakan oleh pengguna akhir. Dapat dilihat pada Tabel 5.12

Tabel 5.12 Rekapitulasi Hasil Pengujian Aplikasi

No	Item yang diuji	Pengguna	Detail	Hasil
1	Melakukan registrasi	Pelanggan	Registrasi akun baru dengan nama, email, password, alamat, jenis kelamin dan no hp	Sesuai
2	Login dengan email dan password valid	Admin, pelanggan, manajer	Login ke sistem sesuai role	Sesuai

No	Item yang diuji	Pengguna	Detail	Hasil
3	Melakukan reset password dengan email	Admin, pelanggan, manajer	Reset password via email	Sesuai
4	Logout	Admin, pelanggan, manajer	Keluar dari sistem, akhiri sesi login	Sesuai
5	Mengakses <i>dashboard</i> admin	Admin	Dapat akses semua menu admin	Sesuai
6	Mengakses <i>dashboard</i> pelanggan	Pelanggan	Dapat akses semua menu pelanggan	Sesuai
7	Mengakses <i>dashboard</i> manajer	Manajer	Dapat akses semua menu manajer	Sesuai
8	Lihat Profil	Admin, pelanggan, manajer	Lihat profil	Sesuai
9	Update Profil	Admin, pelanggan, manajer	Update data yang ada di profil kecuali email	Sesuai
10	Edit Password	Admin, pelanggan, manajer	Edit password	Sesuai

No	Item yang diuji	Pengguna	Detail	Hasil
11	Upload Foto	Admin, pelanggan, manajer	Upload foto profil	Sesuai
12	Lihat Daftar Layanan	pelanggan	Lihat katalog layanan, detail fasilitas, harga, jadwal	Sesuai
13	Melakukan pemesanan	pelanggan	Pilih layanan, tanggal, waktu, fasilitas tambahan, metode pembayaran	Sesuai
14	Cek Ketersediaan	pelanggan	Cek slot tersedia	Sesuai
15	Lihat Status pemesanan	pelanggan	Lihat status penyewaan	Sesuai
16	Batalan pemesanan	pelanggan	Batalan penyewaan	Sesuai
17	Upload Bukti	pelanggan	Unggah bukti pembayaran setelah pembayaran	Sesuai
18	Memberikan Umpan balik	pelanggan	Berikan ulasan, rating setelah menggunakan layanan	Sesuai

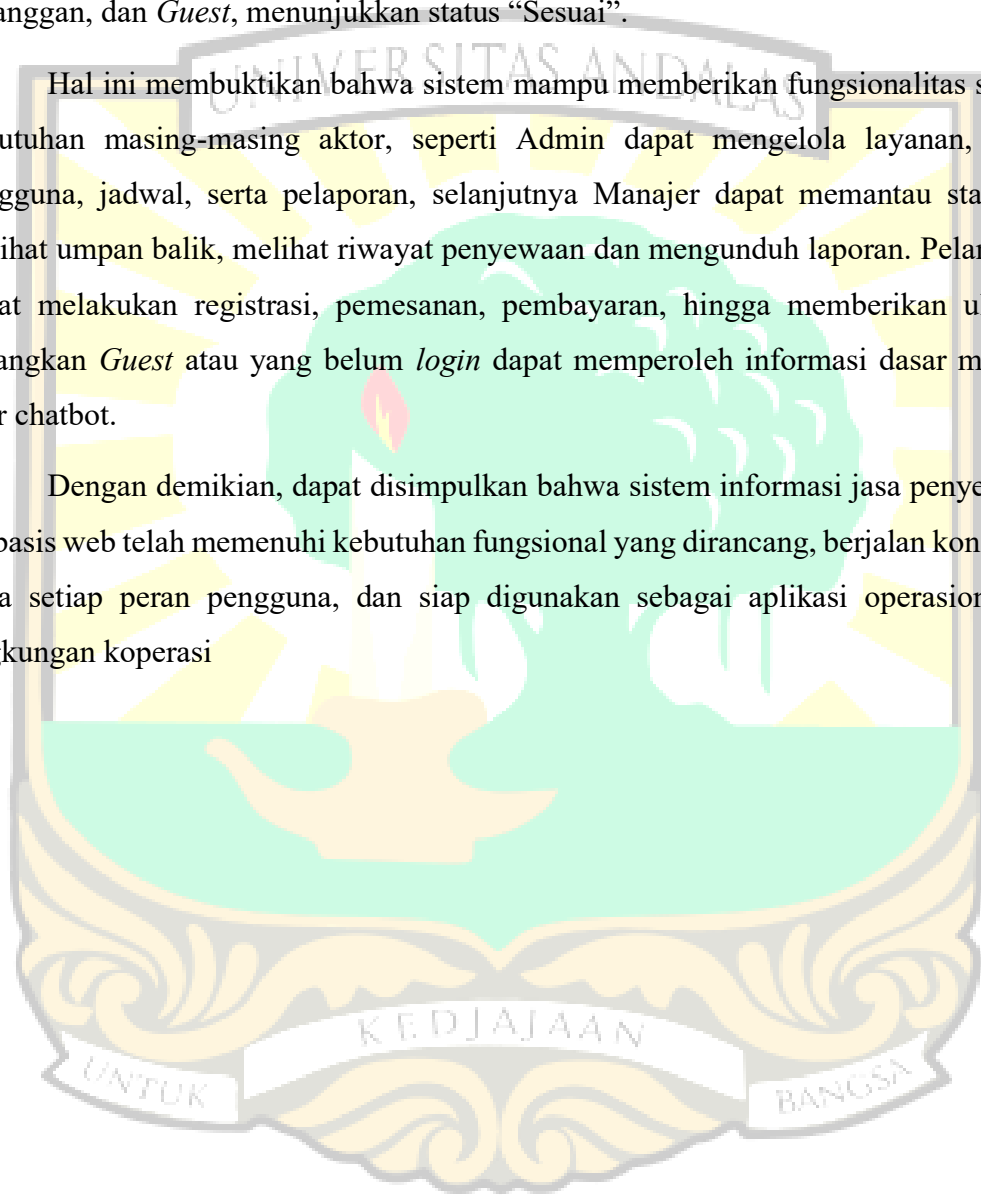
No	Item yang diuji	Pengguna	Detail	Hasil
19	Chat dengan pelanggan	pelanggan	Akses informasi personal dengan autentikasi	Sesuai
20	Chatbot untuk Guest	Guest	Dapat informasi dasar layanan tanpa registrasi	Sesuai
21	Lihat Riwayat Penyewaan	pelanggan	Lihat riwayat penyewaan, status pembayaran, detail transaksi	Sesuai
22	Lihat statistik dashboard	Manajer	Lihat ringkasan data penyewaan, pendapatan, umpan balik, statistik	Sesuai
23	Download laporan penyewaan	Manajer	Download laporan dalam format yang dapat dianalisis	Sesuai
24	Monitor Penyewaan	Manajer	Monitor Penyewaan	Sesuai
25	Lihat Umpan balik Pelanggan	Manajer	Lihat Umpan balik Pelanggan	Sesuai
26	Kelola Layanan	Admin	Tambah, edit, hapus, atur status, harga, deskripsi, gambar	Sesuai

No	Item yang diuji	Pengguna	Detail	Hasil
57	Kelola Jadwal Libur	Admin	Atur jadwal libur operasional koperasi	Sesuai
28	Verifikasi	Admin	Verifikasi bukti transfer, ubah status pembayaran	Sesuai
29	Kelola akun user manajer	Admin	Buat, edit, nonaktifkan, reset password, perpanjang periode	Sesuai
30	Kelola akun user pelanggan	Admin	Aktifkan/nonaktifkan pelanggan	Sesuai
31	Kelola refund	Admin	Proses refund <i>Request</i> , approve/reject, hitung jumlah refund	Sesuai
32	Lihat dan balas umpan balik pelanggan	Admin	Balas umpan balik pelanggan	Sesuai
33	Lihat riwayat penyewaan	Admin	Akses seluruh riwayat transaksi untuk audit dan pelaporan	Sesuai

Berdasarkan hasil pengujian pada Tabel 5.12, seluruh fungsi inti sistem, mulai dari autentikasi, pengelolaan profil, pemesanan layanan, pembayaran, hingga pelaporan, dapat dijalankan dengan hasil yang sesuai dengan spesifikasi. Setiap skenario uji yang melibatkan berbagai peran pengguna, yaitu Admin, Manajer, Pelanggan, dan *Guest*, menunjukkan status “Sesuai”.

Hal ini membuktikan bahwa sistem mampu memberikan fungsionalitas sesuai kebutuhan masing-masing aktor, seperti Admin dapat mengelola layanan, akun pengguna, jadwal, serta pelaporan, selanjutnya Manajer dapat memantau statistik, melihat umpan balik, melihat riwayat penyewaan dan mengunduh laporan. Pelanggan dapat melakukan registrasi, pemesanan, pembayaran, hingga memberikan ulasan, sedangkan *Guest* atau yang belum *login* dapat memperoleh informasi dasar melalui fitur chatbot.

Dengan demikian, dapat disimpulkan bahwa sistem informasi jasa penyewaan berbasis web telah memenuhi kebutuhan fungsional yang dirancang, berjalan konsisten pada setiap peran pengguna, dan siap digunakan sebagai aplikasi operasional di lingkungan koperasi



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan tahapan analisis, perancangan, implementasi, dan pengujian yang dilakukan, pengembangan sistem informasi penyewaan berbasis web pada Koperasi Konsumen Hydro Utama Bukittinggi telah berhasil direalisasikan sesuai kebutuhan pengguna. Sistem ini memungkinkan pelanggan melakukan registrasi, pemesanan layanan, pengecekan ketersediaan jadwal secara langsung melalui sistem, serta unggah bukti pembayaran tanpa harus hadir ke kantor koperasi. Dari sisi pengelolaan, sistem membantu admin dalam pencatatan transaksi, penjadwalan, penyimpanan data, dan pemantauan proses penyewaan secara terstruktur. Selain itu, manajer dapat mengakses informasi penyewaan dan umpan balik pelanggan melalui *dashboard* untuk mendukung proses pengawasan layanan.

Hasil simulasi menggunakan BPSim menunjukkan adanya peningkatan kinerja proses bisnis. Tingkat keterlibatan Admin turun dari 63,34% menjadi 10,67%, menandakan bahwa sebagian besar pekerjaan administratif telah dialihkan ke sistem. Sebaliknya, keterlibatan Pelanggan meningkat dari 87,88% menjadi 97,33%, yang menunjukkan bahwa pelanggan dapat terlibat lebih mandiri dalam mendapatkan layanan. Waktu rata-rata proses penyewaan juga berkurang dari 139 menit menjadi 58,6 menit, sedangkan rata-rata waktu pemrosesan turun dari 50 menit menjadi 16,2 menit, menunjukkan bahwa alur layanan berlangsung lebih singkat dibandingkan proses manual sebelumnya.

Pengujian unit menunjukkan seluruh fungsi pada modul utama berjalan sesuai rancangan dan pengujian sistem dengan metode *black box testing* menunjukkan bahwa seluruh fungsi inti, mulai dari pemesanan layanan, pengelolaan jadwal, hingga verifikasi pembayaran, berjalan sesuai spesifikasi dan dapat digunakan tanpa kendala berarti. Sistem juga menyediakan fitur chatbot sebagai sarana informasi mandiri bagi pengguna serta fitur umpan balik sebagai media evaluasi layanan.

Dengan demikian, tujuan penelitian telah tercapai, yaitu merancang dan membangun sistem informasi penyewaan berbasis web yang mampu memperluas akses layanan, mendukung pencatatan dan pengelolaan data secara terpusat, mengurangi beban kerja admin pada proses pencatatan manual, serta meningkatkan kualitas proses pelayanan melalui pengaturan alur layanan yang lebih terstruktur dan dapat dipantau secara langsung oleh manajer.

6.2 Saran

Sistem informasi penyewaan yang dibangun dalam penelitian ini telah mampu mendukung proses layanan penyewaan di koperasi, namun masih memiliki peluang pengembangan lebih lanjut agar dapat menyesuaikan dengan kebutuhan operasional ke depan. Salah satu pengembangan yang dapat dilakukan adalah integrasi pembayaran digital otomatis untuk mempercepat proses pencatatan transaksi dan mengurangi ketergantungan pada verifikasi manual. Saat ini koperasi masih menerapkan metode transfer manual karena volume transaksi belum besar dan membutuhkan masa adaptasi, sehingga verifikasi oleh admin dinilai lebih sesuai pada tahap awal.

Di samping itu, fitur chatbot yang tersedia saat ini dapat ditingkatkan menjadi layanan informasi yang lebih interaktif dan responsif, seperti integrasi kecerdasan buatan atau pemrosesan bahasa alami agar mampu menjawab pertanyaan pengguna secara lebih dinamis dan membantu proses layanan secara mandiri. Selain pengembangan fitur, diperlukan sosialisasi serta pelatihan kepada pengelola dan anggota koperasi untuk memastikan pemanfaatan sistem berjalan optimal. Pemeliharaan dan pembaruan sistem secara berkala juga perlu dilakukan untuk menjaga kestabilan sistem, meningkatkan keamanan data, serta memastikan sistem tetap relevan dengan perkembangan kebutuhan penyewaan di masa mendatang.

DAFTAR PUSTAKA

- Aliman, W. (2021). Perancangan Perangkat Lunak untuk Menggambar Diagram Berbasis Android. *Syntax Literate; Jurnal Ilmiah Indonesia*, 6(6), 3091. <https://doi.org/10.36418/syntax-literate.v6i6.1404>
- Ananta, Y. E., Yuniati, D., Rolliawati, D., Kunaefi, A., & Permadi, A. (2025). Pengembangan Aplikasi Sido Chatbot sebagai Aplikasi Pengenalan Objek Wisata Kediri Menggunakan Rule-Based Pattern Matching. *Jurnal Teknologi Dan Informasi*, 15(1), 40–53. <https://doi.org/10.34010/jati.v15i1.15219>
- Aprillianie, B., & Son Muarie, M. (2023). Pemodelan Proses Layanan Donasi Pada Dompot Dhuafa Menggunakan Business Process Modelling Notation (BPMN). *Prosiding Seminar Nasional Teknologi Komputer Dan Sains*, 1(1), 1–9. <https://prosiding.seminars.id/prosainteks>
- Arif, S. M. (2023). Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi 4.0 Internasional. *Journal of Information System, Applied, Management, Accounting*, 7(3), 789–796. <https://doi.org/10.52362/jisamar.v7i3.1168>
- Asyari, M. R. (2021). *SISTEM INFORMASI ARSIP SURAT MENYURAT*. 3, 31–2021. <https://doi.org/10.47233/jteksis.v3i1.172>
- Azis, N. (2022). *Analisis Perancangan Sistem Informasi*.
- Badrul, M., Ardy, R., Nusa Mandiri Jl Jatiwaringin Raya No, S., & Cipinang Melayu Jakarta Timur, K. (2021). Penerapan Metode Waterfall pada Perancangan Sistem Informasi Pendaftaran Siswa Baru. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 5, Issue 1).
- Bagas Oktaviandi, D., Ulya, S., Ridwan, A., & Budianita, A. (2023). APLIKASI CHATBOT PERSEWAAN KOST BERBASIS WEB MENGGUNAKAN METODE WATERFALL. In *Jurnal Bisnis Digital dan Sistem Informasi* (Vol. 46, Issue 1).
- Dicky, B. O., Ulya, S., Ridwan, A., & Budianita, A. (2023). APLIKASI CHATBOT PERSEWAAN KOST BERBASIS WEB MENGGUNAKAN METODE WATERFALL. In *Jurnal Bisnis Digital dan Sistem Informasi* (Vol. 46, Issue 1).
- Dody Firmansyah, M., Kom, S., & Kom, M. (2021). Analisa dan Perancangan Web E-Commerce Berbasis Website pada Toko Ida Shoes. *Journal of Information System and Technology*, 02(03), 62–76.
- Dr. Bachruddin Saleh Luturlean, S. E. , M. M. , dkk. (n.d.). *penyewaan*.

- Ekohariadi, M., Pd, R., Harimurti, S., Pd, M., Anistyasari, Y., Kom, S., Asmunin, M. K., & Kom, M. (2020). *PUSAT KARIR SMK 4.0 BERBASIS TEKNOLOGI INFORMASI*.
- Fatoni, A., Normalisa, & Zulfikar, A. F. (2020). Merancang Sistem Aplikasi Pendaftaran Kartu Kredit di Bank Panin Kantor Kas Permata Taman Palem. *Journal of Engineering, Technology, and Applied Science*, 2(1), 59–85. <https://doi.org/10.36079/lamintang.jetas-0201.95>
- Gede, W., Bratha, E., Program, M., Manajemen, M., Bhayangkara, U., Raya, J., & Penulis, K. (2022). *LITERATURE REVIEW KOMPONEN SISTEM INFORMASI MANAJEMEN: SOFTWARE, DATABASE DAN BRAINWARE*. 3(3). <https://doi.org/10.31933/jemsi.v3i3>
- Harriko Nur Harzeki. (2025). *PEMBANGUNAN APLIKASI PENYEWAAN LAPANGAN BADMINTON DAN TENIS BERBASIS WEB DAN MOBILE PADA GARUDA SAKTI SPORT HALL KOTA PADANG*.
- Hasanah, F. N., & Untari, R. S. (2020). *Buku Ajar Rekayasa Perangkat Lunak*.
- Hasibuan, A. N., & Dirgahayu, T. (2021). *Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna*.
- Honi, R. A., & Iksari, I. H. (2023). Sistem Informasi Sistem Informasi Penjualan Penjualan Aplikasi Aplikasi Kasir Berbasis Website Website Studi Kasus Toko Grosir Bogor. *Jurnal Riset Informatika Dan Inovasi*.
- Indah Melyani, R., & Aji, S. (2023). Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Framework Laravel dengan Metode Agile Software Development. *Jurnal Sistem Informasi Akuntansi*, 03(01). <http://jurnal.bsi.ac.id/index.php/jasika>
- Julianti, Y. L., Mardiana, T., & Rahmawati, A. (2022). Rancang Bangun Sistem Informasi Penyewaan Camping Ground Berbasis Web Pada Lembah Permai Resor. *Jurnal Pariwisata Bisnis Digital Dan Manajemen*, 1(2), 68–77. <https://doi.org/10.33480/jasdim.v1i2.3673>
- Luturlean, Dr. B. S. S. E. , M. M., Dr. Sukmadi, S. E. , M. M., Dr. ER Ummi Kalsum, MM. Par. , C., & Dr. Lien Maulina, S. Sos. , M. Pd. , A. Mad. Par. , C. (2020). *Manajemen Koperasi*.
- Mall, R. (2014). *RAJIB MALL Department of Computer Science and Engineering Indian Institute of Technology Kharagpur* (Vol. 3, Issue 2).
- Noviana, R. (2022). PEMBUATAN APLIKASI PENJUALAN BERBASIS WEB MONJA STORE MENGGUNAKAN PHP DAN MYSQL. *JTS*, 1(2).

- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103. <https://doi.org/10.56127/jts.v1i2.108>
- Nur, S. A., Ningsih, W., & Suartana, I. M. (2024). Implementasi Unit Testing pada Website Columbia Catanta Menggunakan Algoritma Generate dan Test Case. *Journal of Informatics and Computer Science*, 05.
- Nurdany, A., & Prajasari, A. C. (2020). DIGITALIZATION IN INDONESIAN COOPERATIVES: IS IT NECESSARY? *Journal of Developing Economies*, 5(2), 125. <https://doi.org/10.20473/jde.v5i2.19447>
- Pricillia, Z. T. (2021). Survey Paper: Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD).
- Pradana Putra, A., Andriyanto, F., Dewi Muji Harti, T., & Puspitasari, W. (2020). *PENGUJIAN APLIKASI POINT OF SALE BERBASIS WEB MENGGUNAKAN BLACK BOX TESTING*.
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering*.
- Rahmatuloh, M., & Revanda, M. R. (2022). Rancang Bangun Sistem Informasi Jasa Pengiriman Barang Pada Pt. Haluan Indah Transporindo Berbasis Web. *Jurnal Teknik Informatika*.
- Rahmadoni, M. H. A. (2025). PEMBANGUNAN APLIKASI BOOKING DAN PENYEWAAN STUDIO FOTO DI KOTA PADANG DENGAN MEMANFAATKAN MODEL BISNIS MARKETPLACE.
- Ramdany, S. W., Aulia Kaidar, S., Aguchino, B., Amelia, C., Putri, A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. In *Journal of Industrial and Engineering System* (Vol. 5, Issue 1).
- Rasid, N. M., & Fitria, D. (2024). PENILAIAN KINERJA KEUANGAN KOPERASI PEGAWAI NEGERI (KPN) SEPTA JAYA DESA PADANG HANGAT KABUPATEN KAUR. In *Journal of Management and Innovation Entrepreneursip (JMIE)* (Vol. 1, Issue 3).
- Remawati, D. (2021). *BUKU AJAR. WEB JSP DENGAN DATABASE MYSQL* Hendro Wijayanto STMIK Sinar Nusantara Surakarta. <https://www.researchgate.net/publication/352901412>
- Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. In *Jurnal Media Infotama*.

- Sebastian, D., Sembiring, I., Sedyono, E., & Hartomo, K. D. (2023). Literature Review dan Survey Trend Teknologi Pengembangan Website untuk Website Skala Kecil. *Educators and Professionals*, 7(2), 185–194.
- Shadiq, J., Safei, A., & Wahyudin Ratu Loly, R. (2021). INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing. *Information Management for Educators and Professionals*, 5(2), 97–110.
- Silalahi, F. D. (2022). *MySQL (Structured Query Language) Manajemen Database*.
- Siswidiyanto, S., Munif, A., Wijayanti, D., & Haryadi, E. (2020a). Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 15(1), 18–25. <https://doi.org/10.35969/interkom.v15i1.64>
- Sommerville, Ian. (2020). *Engineering software products : an introduction to modern software engineering*. Pearson Education, Inc.
- Sonia, S. D. F., & Cahyaningtyas, C. (2023). Perancangan Sistem Informasi Penyewaan Lapangan Futsal Berbasis Web. *JURNAL SAINS DAN KOMPUTER*, 7(02), 1–4. <https://doi.org/10.61179/jurnalinfact.v7i02.448>
- Sparx Systems. (2022). *BPSim Business Simulations How to simulate a business process? In Sparx Systems Enterprise Architect, create a process model in BPMN and a business process simulation (BPSim) configuration, and execute it using a BPSim simulation engine, comparing multiple simulation results. Enterprise Architect User Guide Series Author: Sparx Systems CREATED WITH.*
- Suasana, I. S., & Kom, M. (2023). *P H P Framework*.
- Sugiyanto, R. M. (2020.). *BAB 2 LANDASAN TEORI*.
- Sumriyah, S. H., & S. H. Djulaeka. (2023). *Kapita Selektta Hukum Perjanjian*.
- Syaputri, L., Gunawan Putra, E., Syahrani, E., Dwian, E., Purwani, F., informasi, S., dan Teknologi, S., & Islam Negri Raden Fatah Palembang, U. (2024). ARTICLE INFO ABSTRACT Correspondent. *Journal of Scientech Research and Development*, 6(2). <https://idm.or.id/JSCR/in>
- Syarifudin, M. R., & Bisma, R. (2023). Analisa Proses Bisnis Pengadilan Agama Surabaya Menggunakan Metode Business Procees Modelling Notation (BPMN). In JEISBI (Vol. 04).
- Tamo, A., Bili, I., Kelen, Y. P. K., Manek, S. S., & Gelu, L. P. (2023). RANCANG BANGUN SISTEM INFORMASI PENYEWAAN GEDUNG RESEPSI

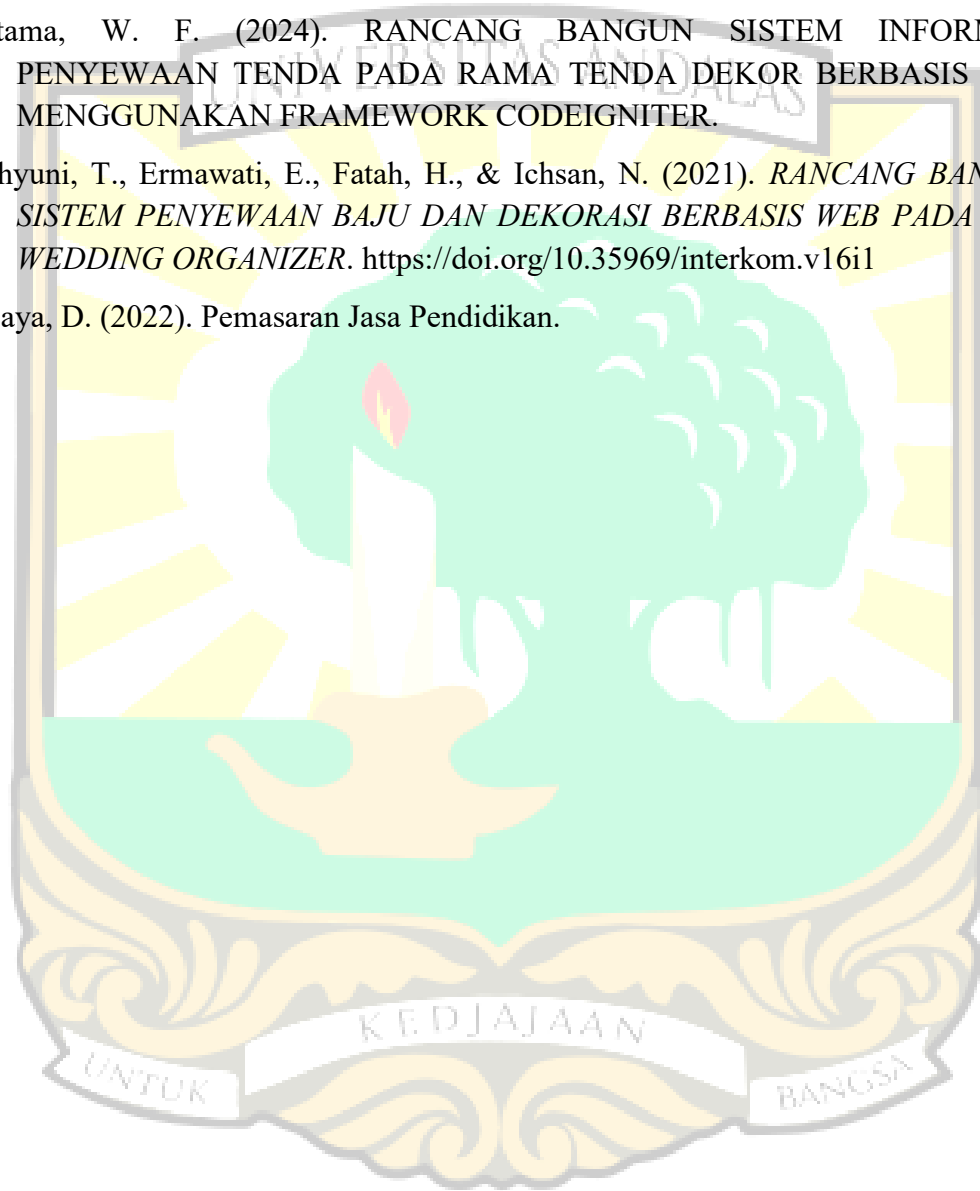
PERNIKAHAN MENGGUNAKAN METODE WATERFALL DI KOTA KEFAMENANU BERBASIS WEB. *Jurnal Manajemen Dan Teknologi Informasi (JMTI)*, 13(2), 10–23. <https://doi.org/10.59819>

Pricillia, Z. T. (2021). Survey Paper: Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD).

Pratama, W. F. (2024). RANCANG BANGUN SISTEM INFORMASI PENYEWAAN TENDA PADA RAMA TENDA DEKOR BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER.

Wahyuni, T., Ermawati, E., Fatah, H., & Ichsan, N. (2021). *RANCANG BANGUN SISTEM PENYEWAAN BAJU DAN DEKORASI BERBASIS WEB PADA NITA WEDDING ORGANIZER*. <https://doi.org/10.35969/interkom.v16i1>

Wijaya, D. (2022). Pemasaran Jasa Pendidikan.



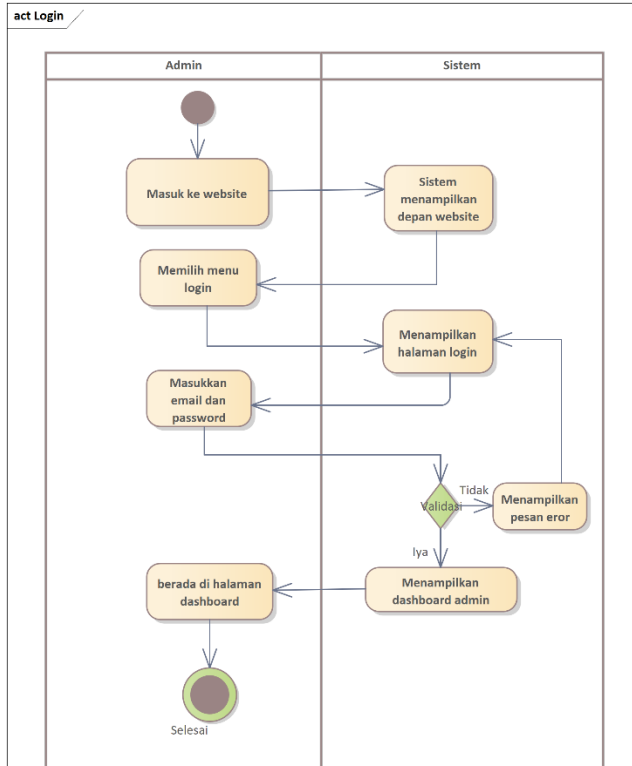


LAMPIRAN A

(Activity Diagram – Lanjutan)

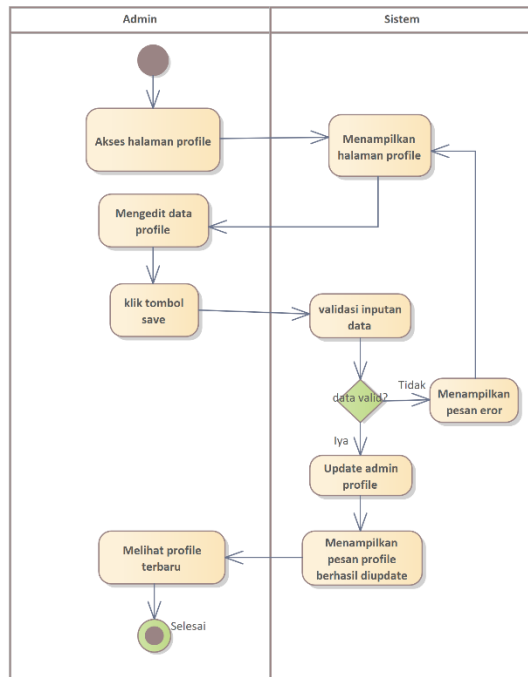
A. Admin

1. Login



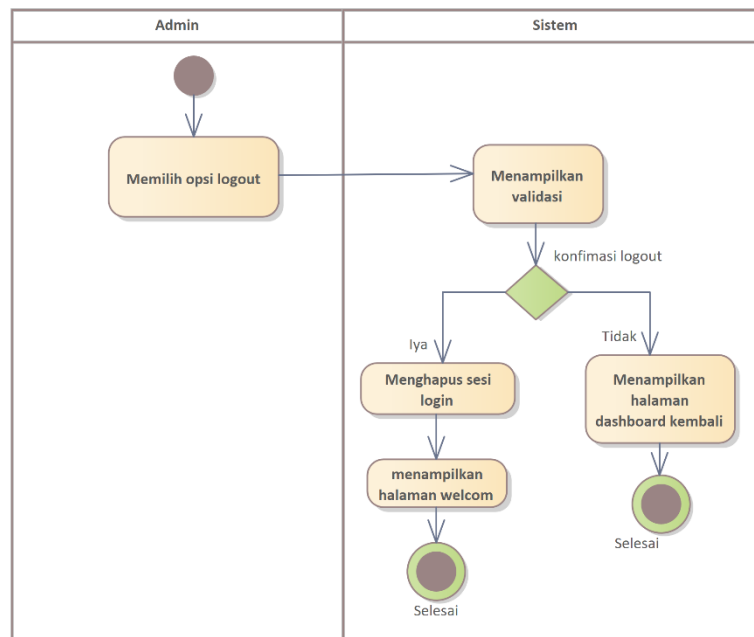
2. Edit profil

act Edit Profile

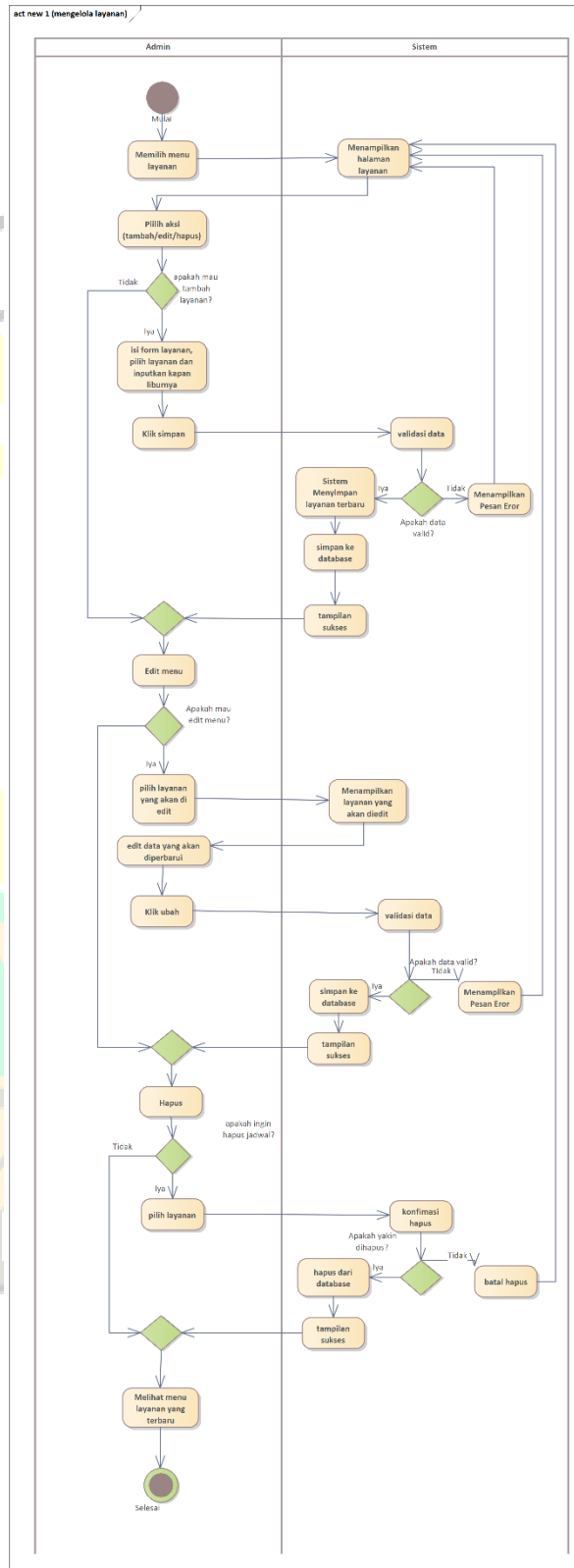


3. Logout

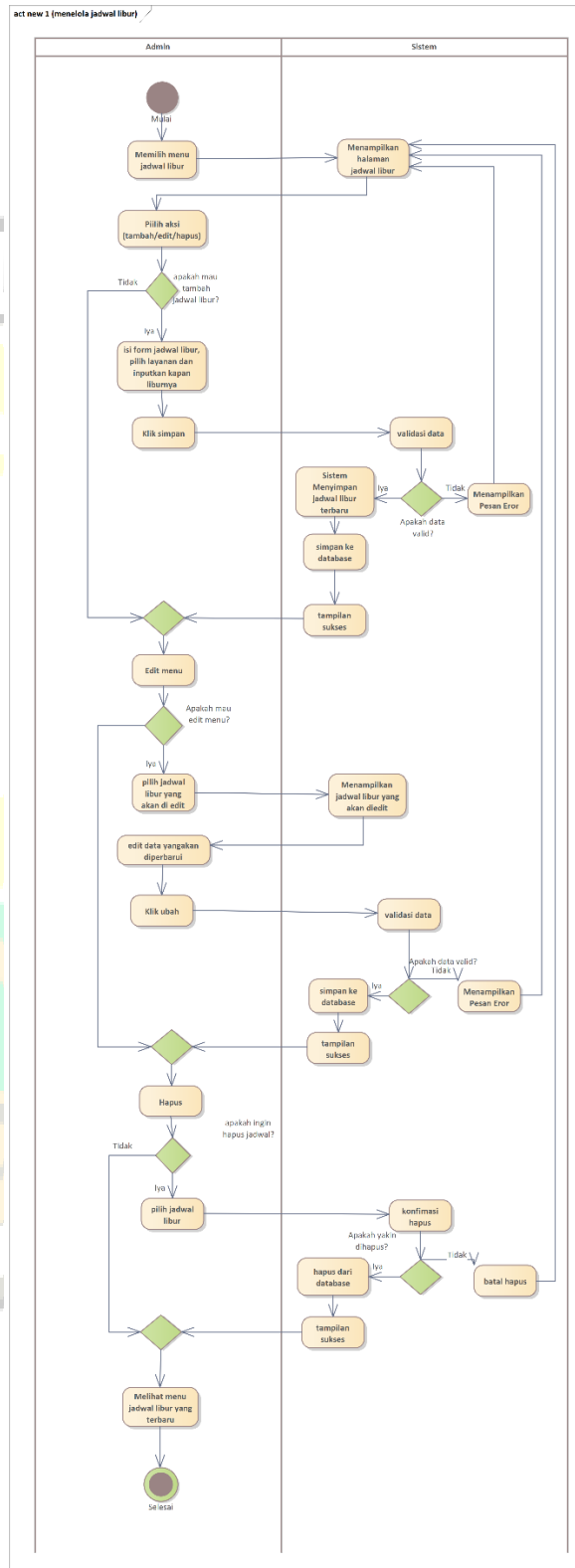
act Logout



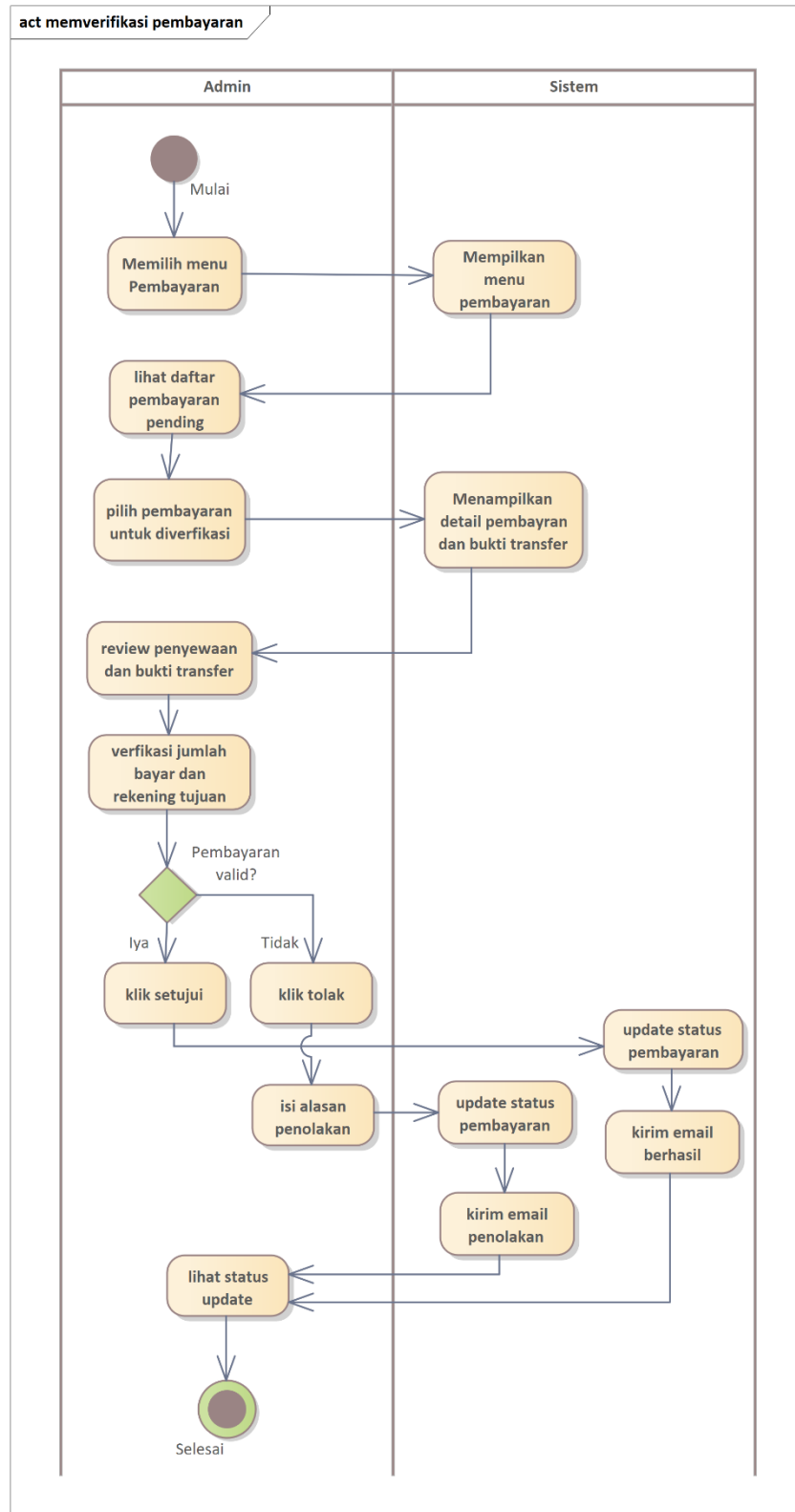
4. Mengelola layanan



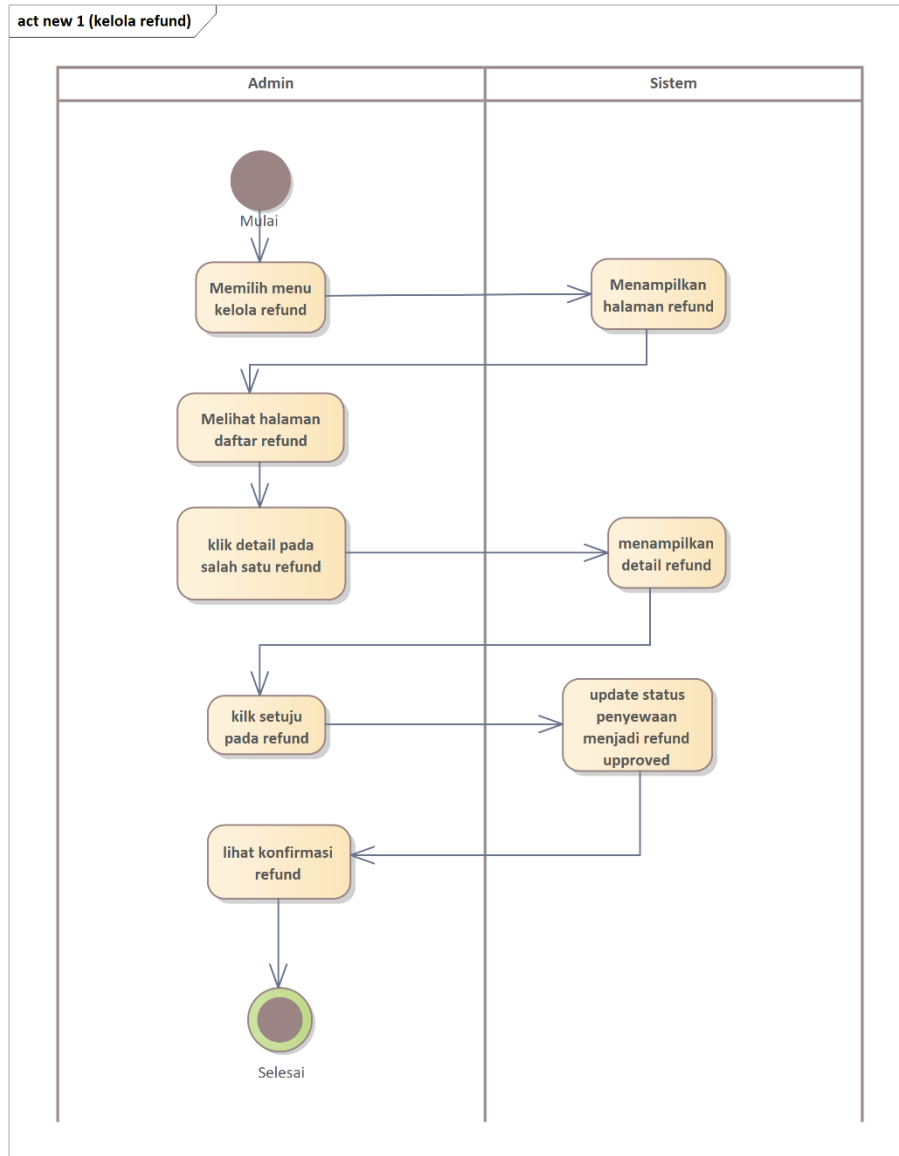
5. Mengelola jadwal libur



6. Memverifikasi pembayaran pelanggan



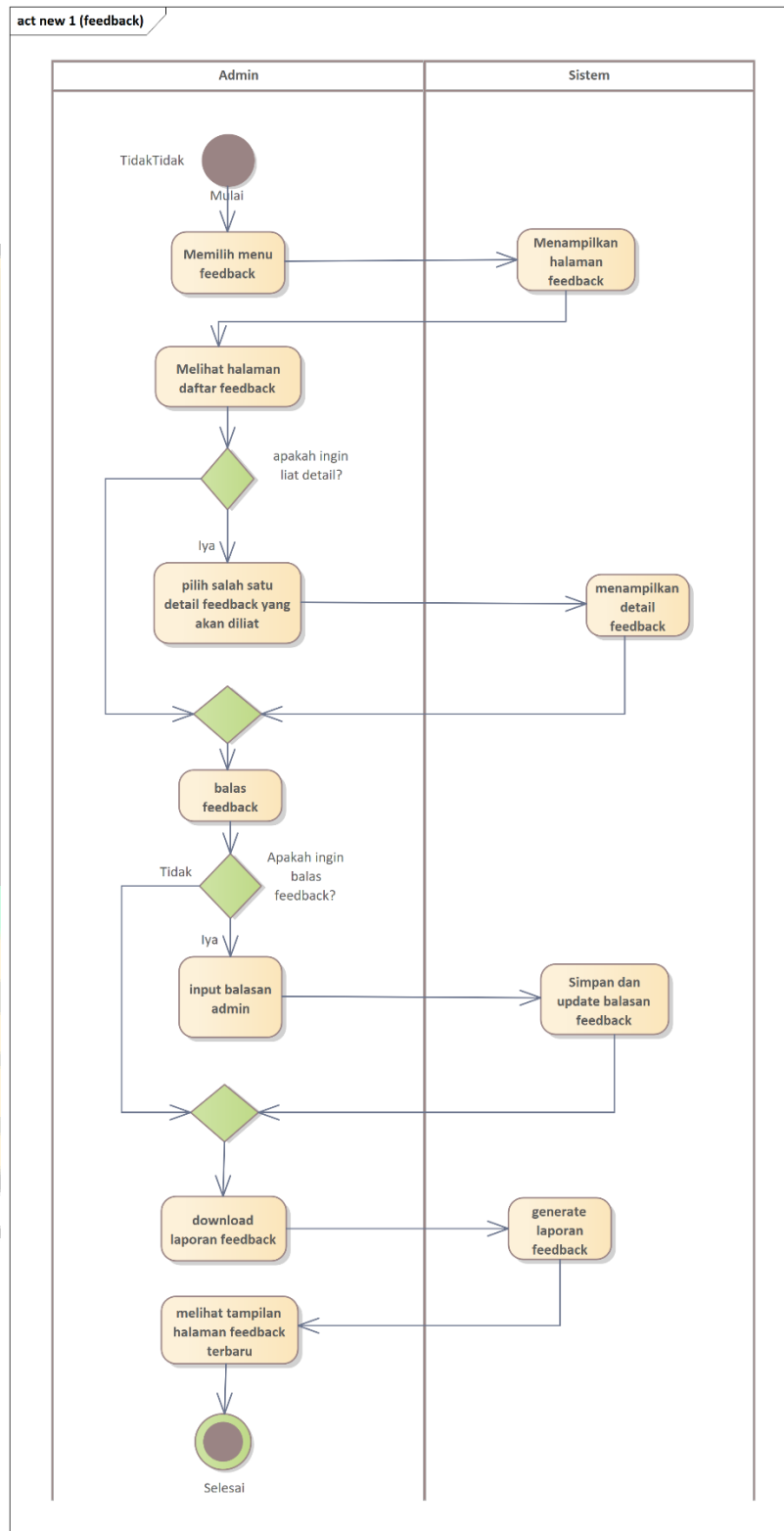
7. Memverifikasi refund pelanggan



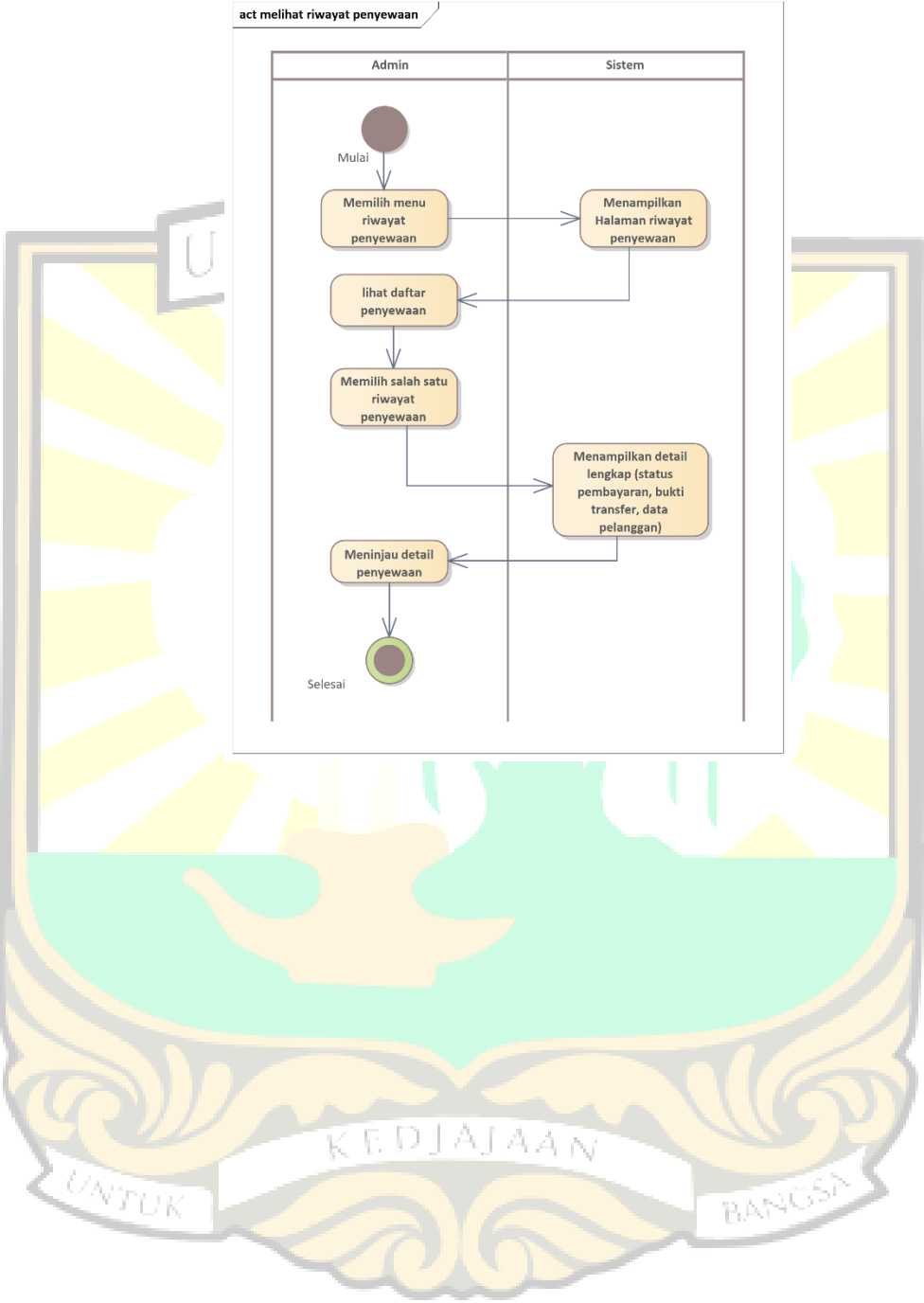
```

graph TD
    Start([Mulai]) --> Login([Masuk ke sistem manager])
    Login --> Logout([Keluar sistem manager])
    Logout --> End([Selesai])
    Login --> AddData([Tambah data manager])
    AddData --> Validasi1{Validasi data}
    Validasi1 -- Ya --> Simpan1([Simpan data manager])
    Validasi1 -- Tidak --> AddData
    Simpan1 --> TampilData([Tampilkan data manager])
    TampilData --> Validasi2{Validasi data}
    Validasi2 -- Ya --> Simpan2([Simpan data manager])
    Validasi2 -- Tidak --> AddData
    Simpan2 --> EditData([Edit data manager])
    EditData --> Validasi3{Validasi data}
    Validasi3 -- Ya --> Simpan3([Simpan data manager])
    Validasi3 -- Tidak --> AddData
    Simpan3 --> HapusData([Hapus data manager])
    HapusData --> Validasi4{Validasi data}
    Validasi4 -- Ya --> Simpan4([Simpan data manager])
    Validasi4 -- Tidak --> AddData
    Simpan4 --> BackupData([Backup data manager])
    BackupData --> End
  
```

9. Melihat dan membalas umpan balik

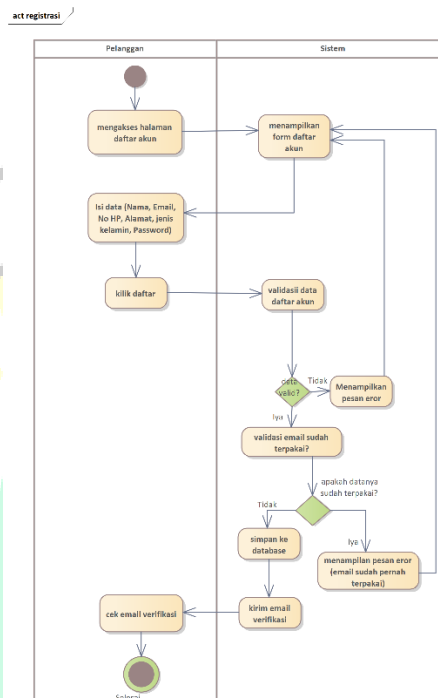


10. Melihat Riwayat penyewaan

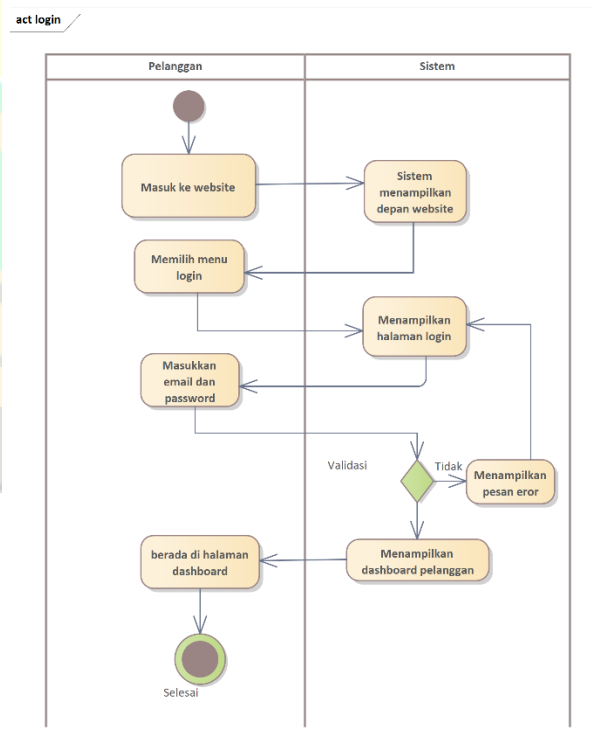


B. Pelanggan (Customer)

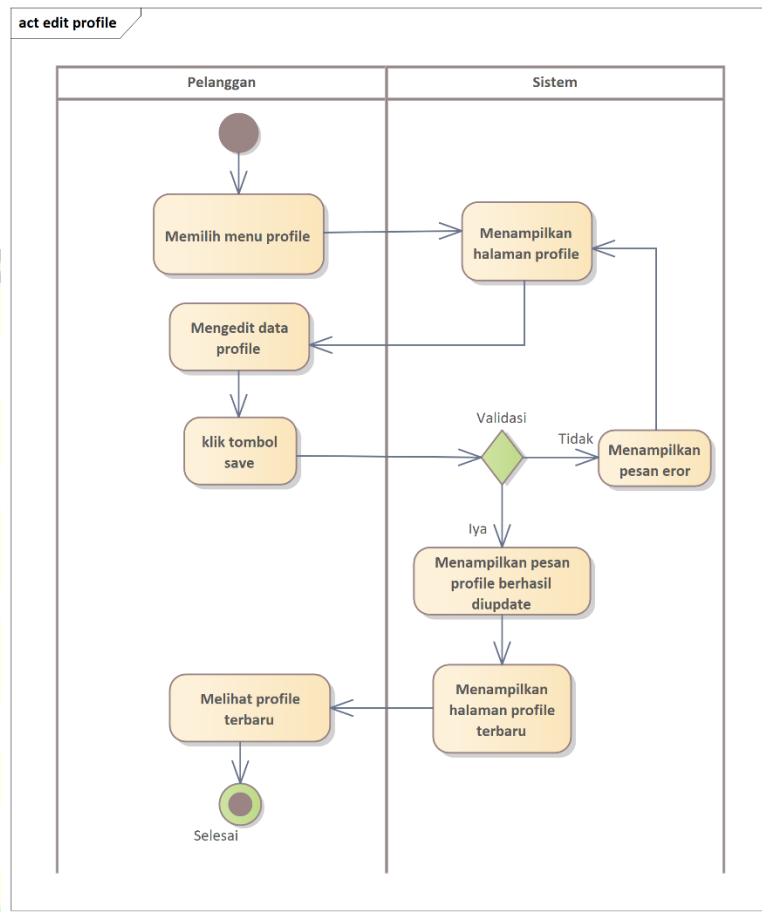
1. Registrasi



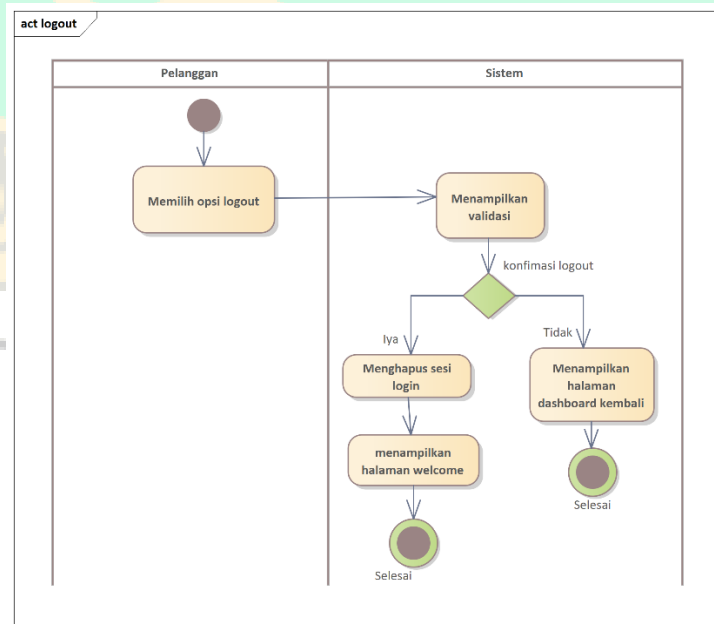
2. Login



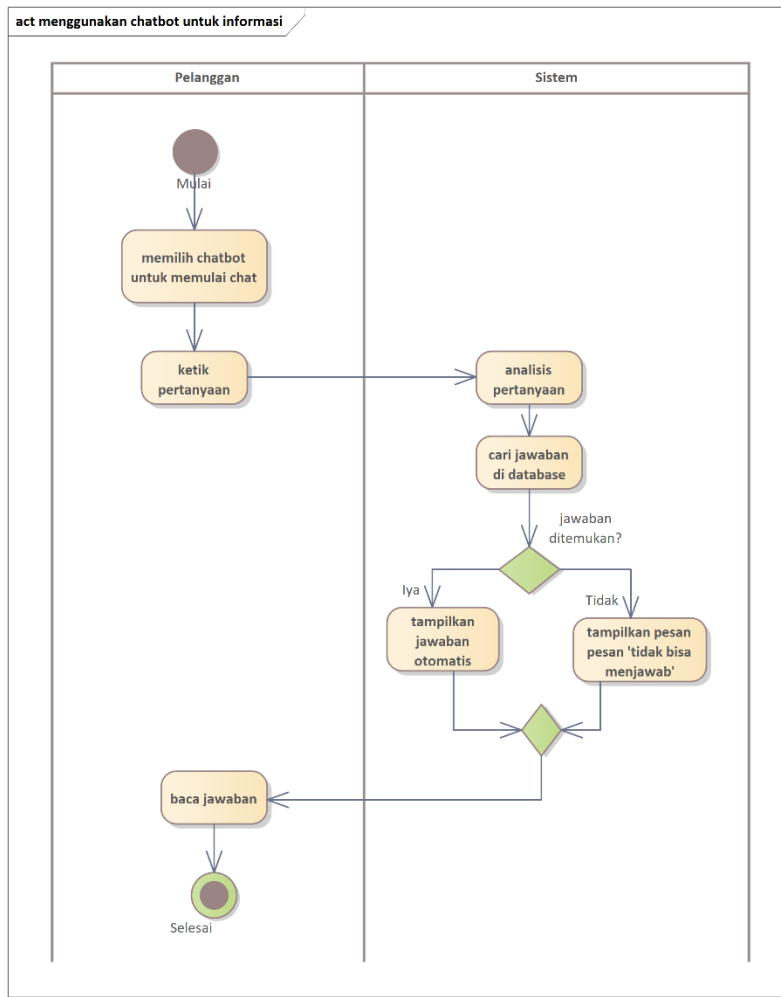
3. Edit profil



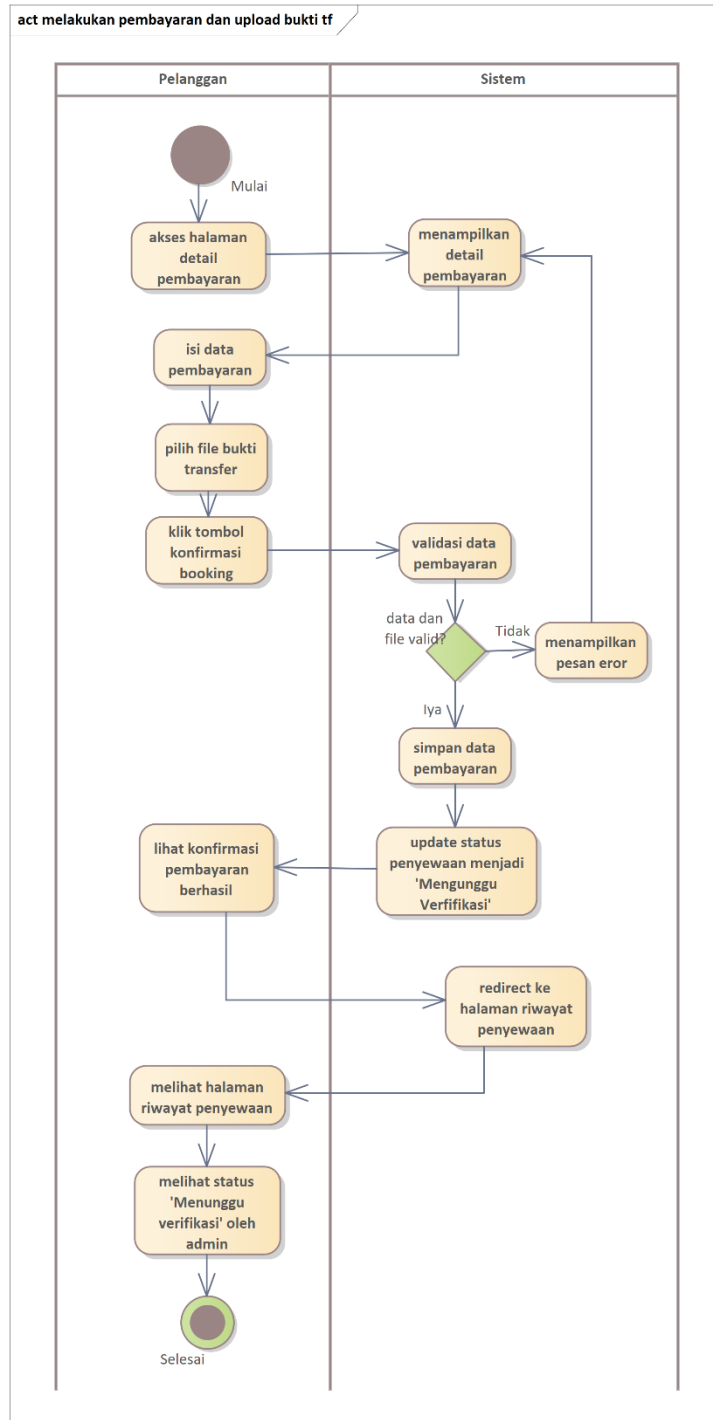
4. Logout



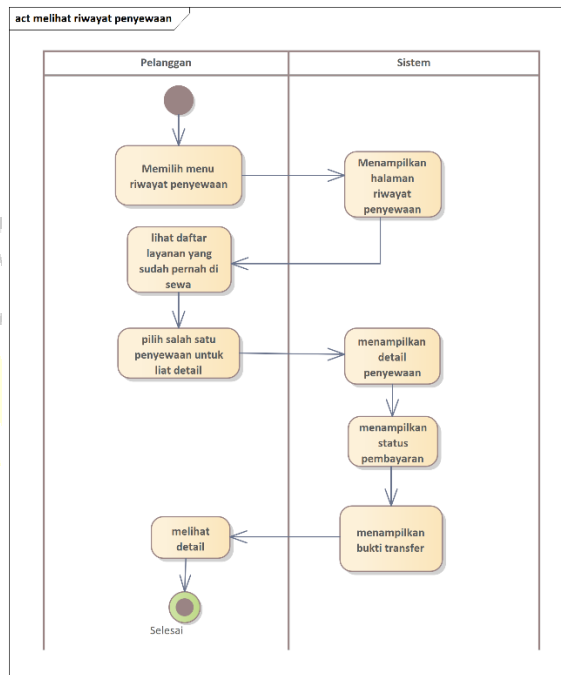
5. Menggunakan *chatbot* untuk informasi



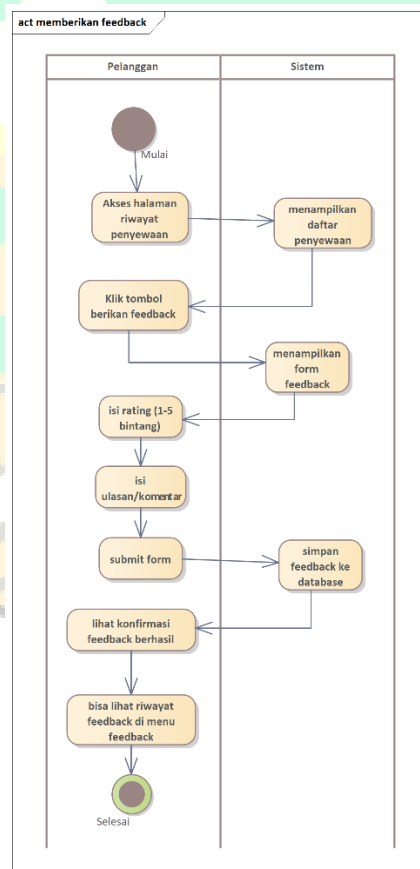
6. Melakukan pembayaran dan unggah bukti pembayaran



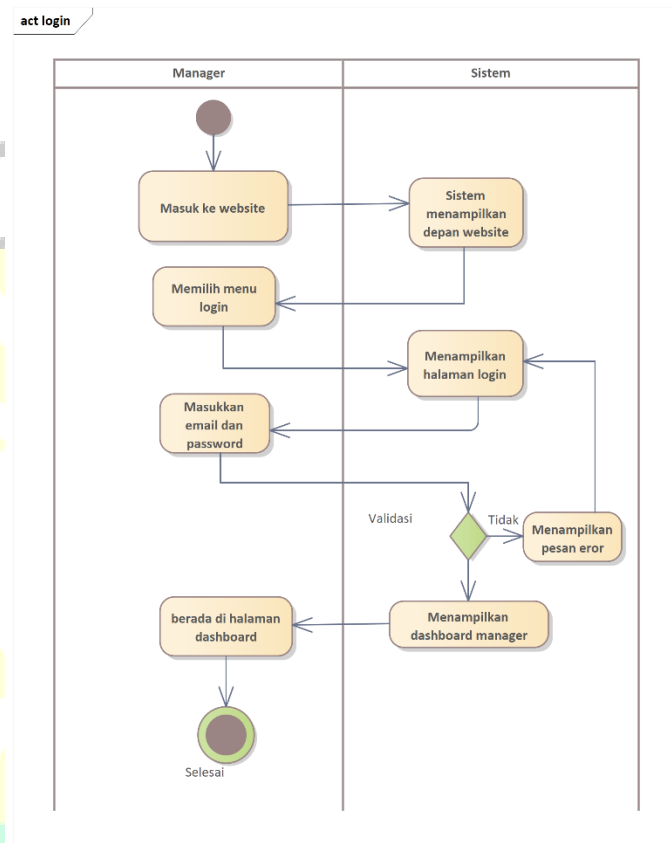
7. Melihat Riwayat penyewaan



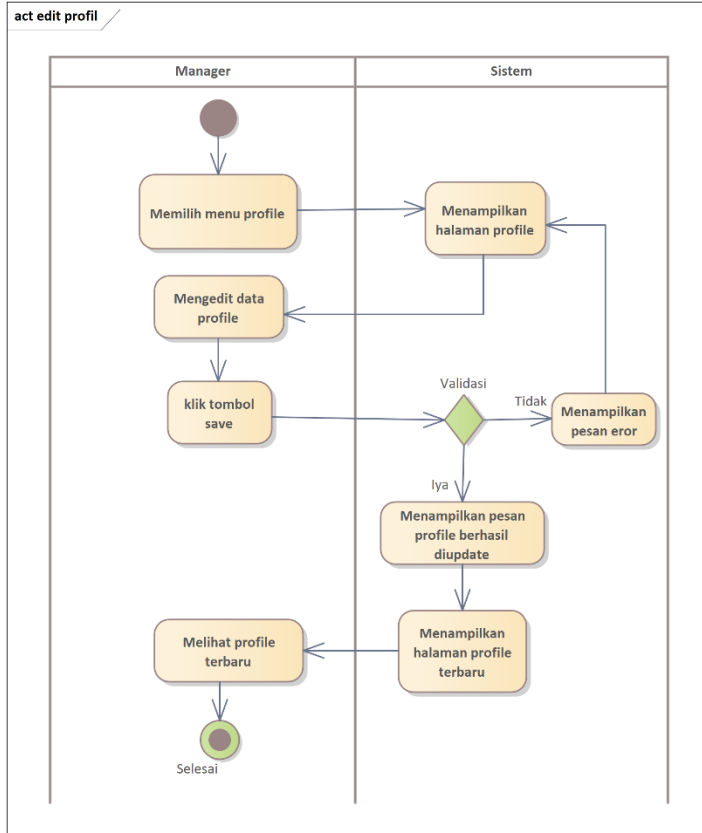
8. Memberikan umpan balik



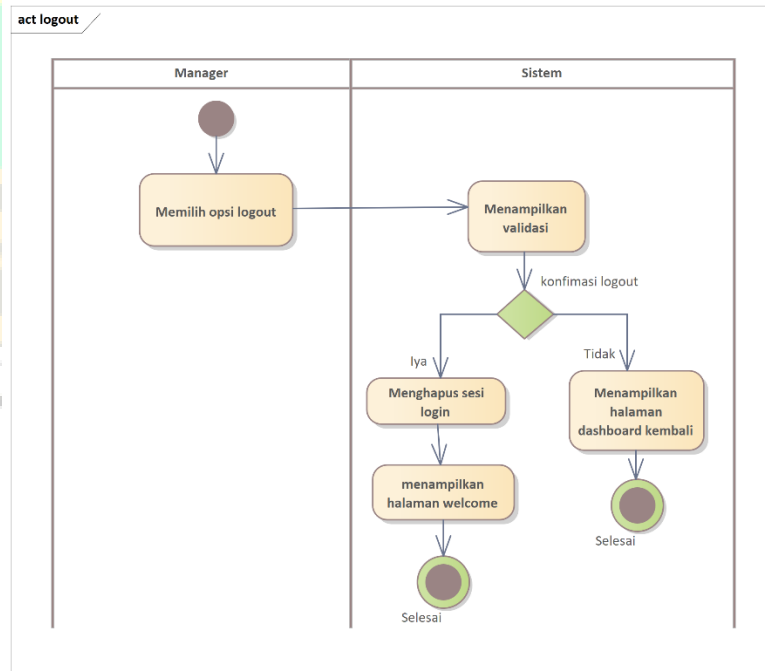
C. Manajer
1. Login



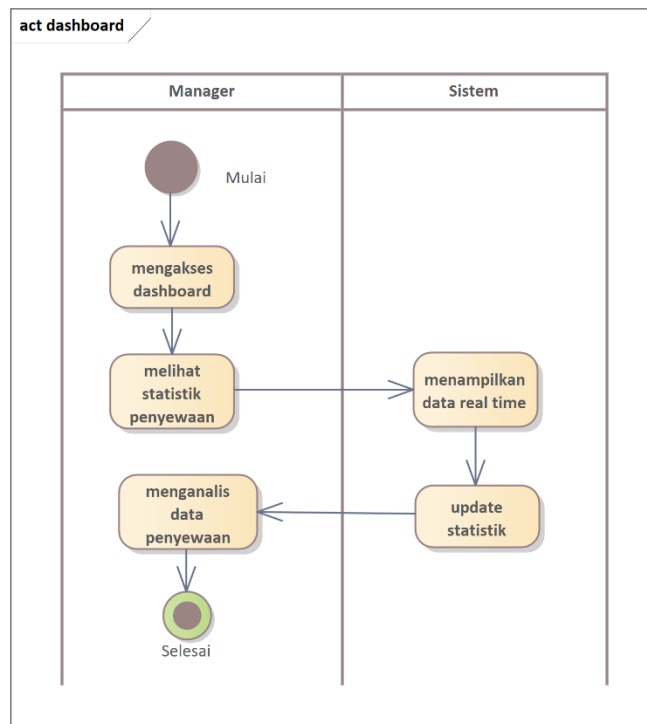
2. Edit profil



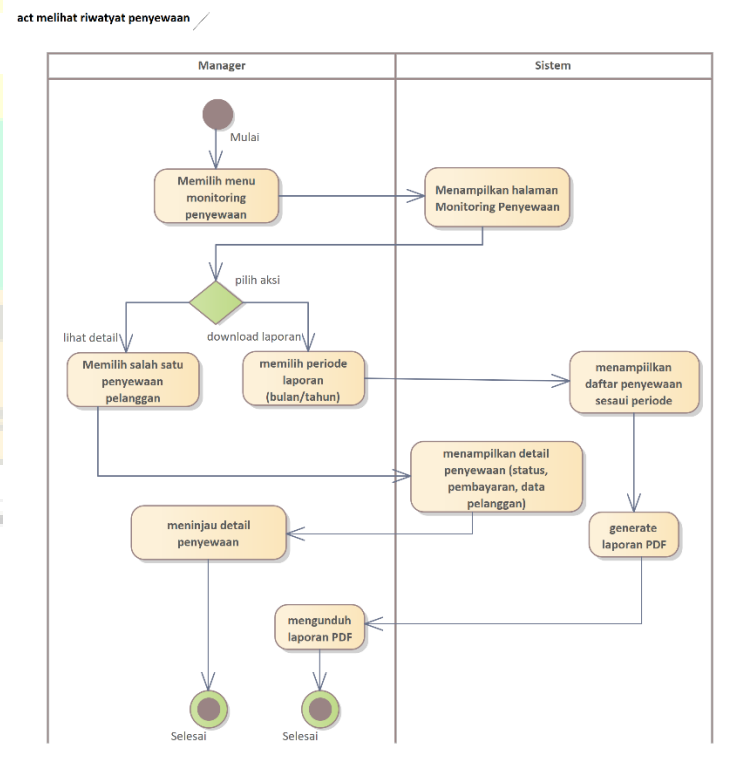
3. Logout



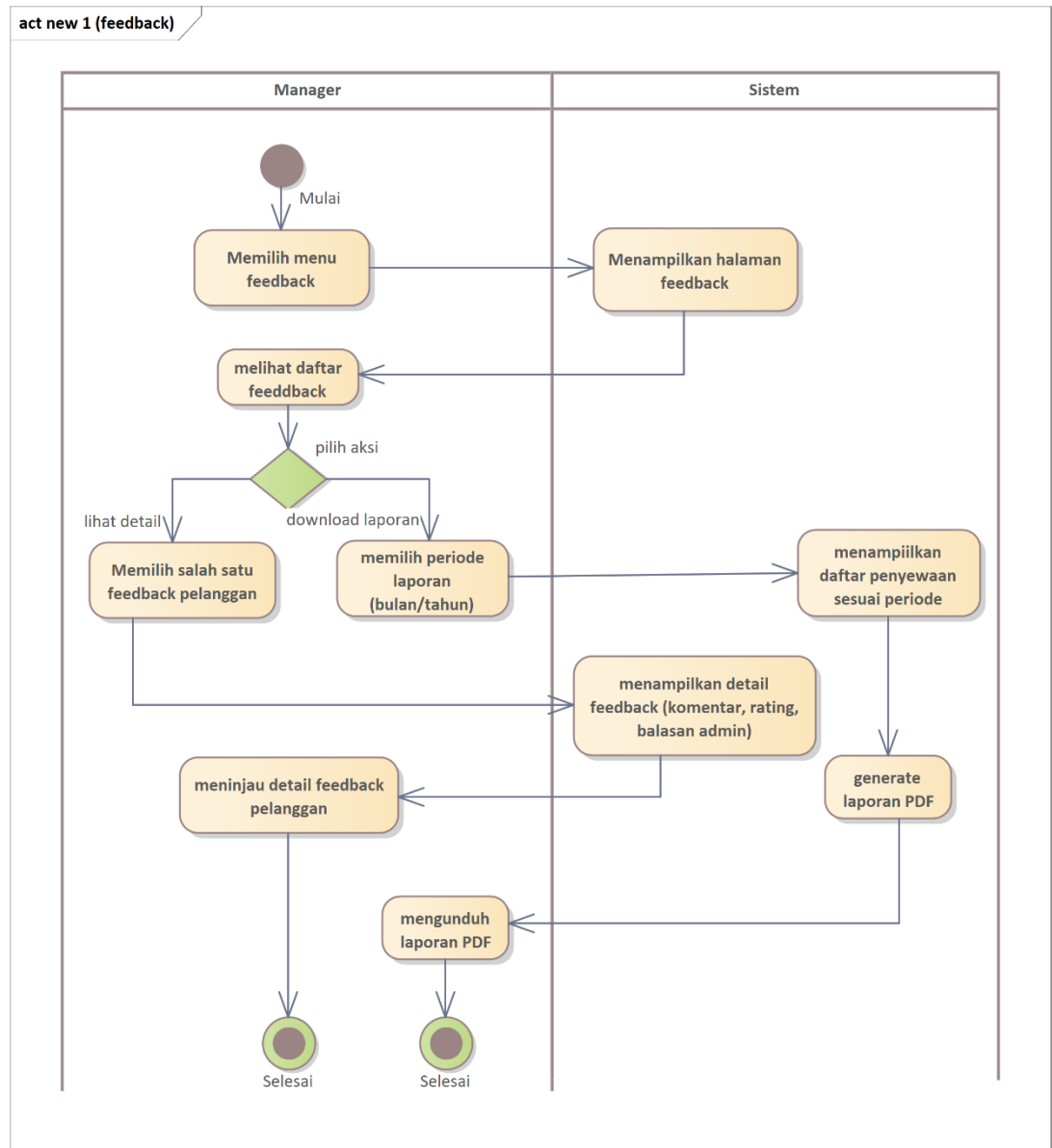
4. Dashboard manajer



5. Melihat riwayat penyewaan pelanggan dan download laporan



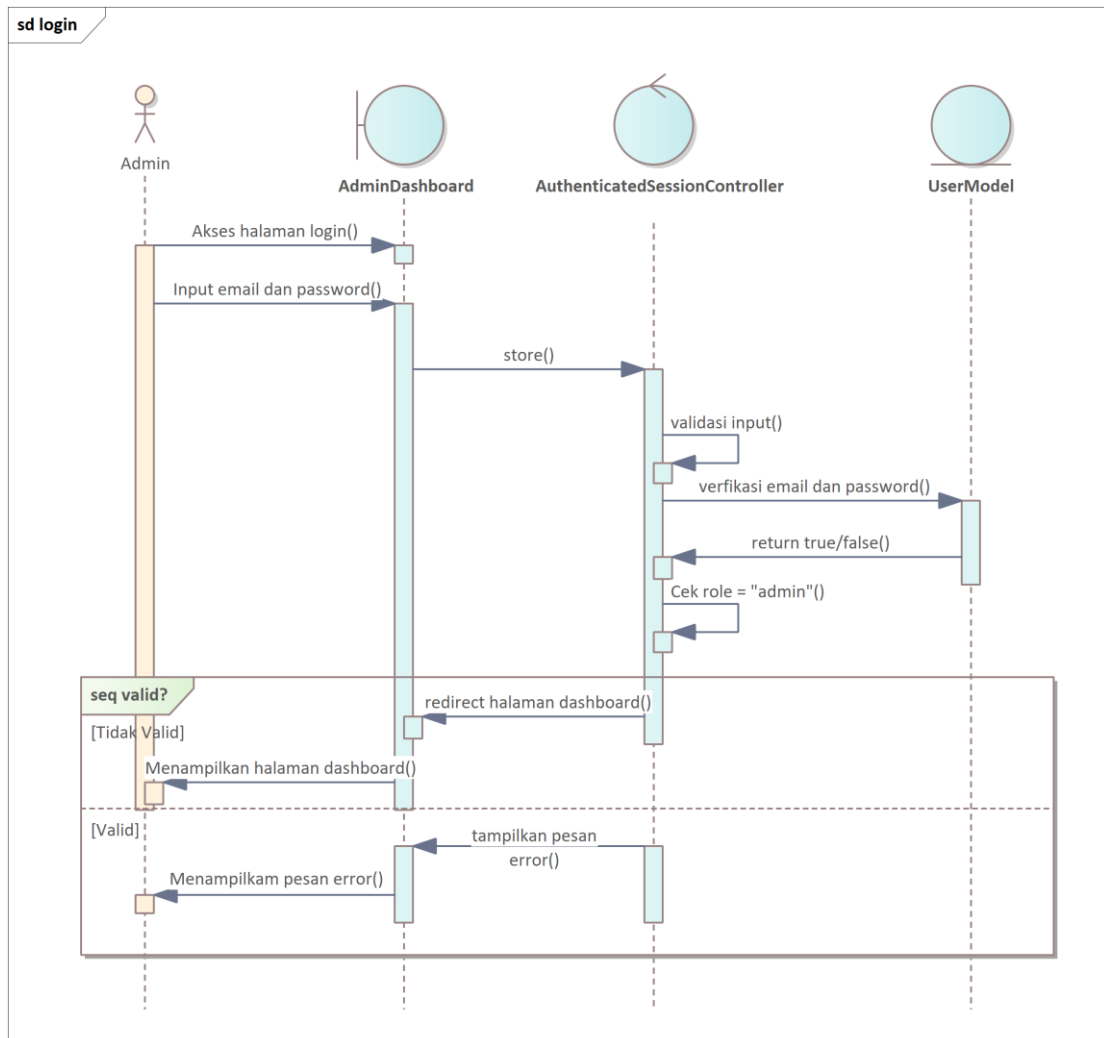
6. Meninjau Umpan balik sebagai bahan evaluasi



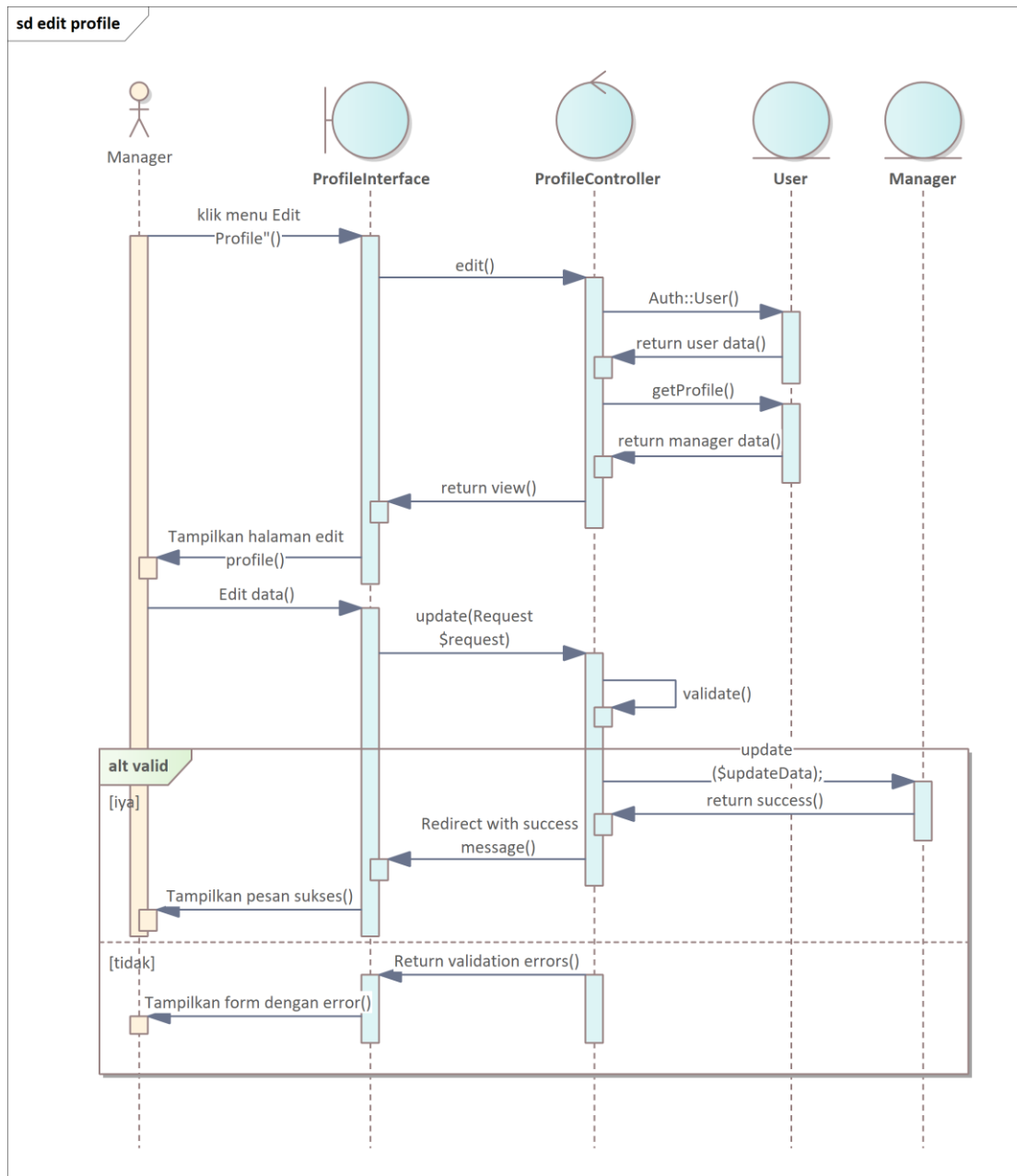


A. Admin

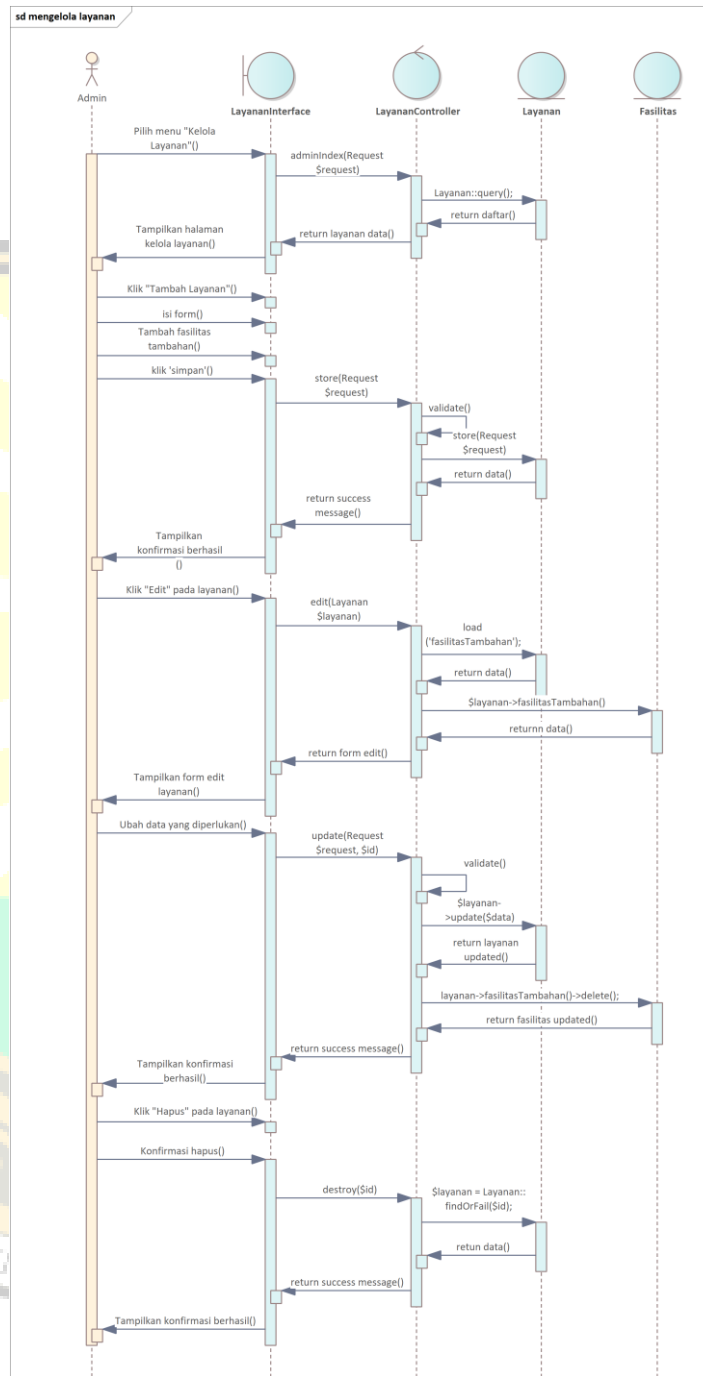
1. Login



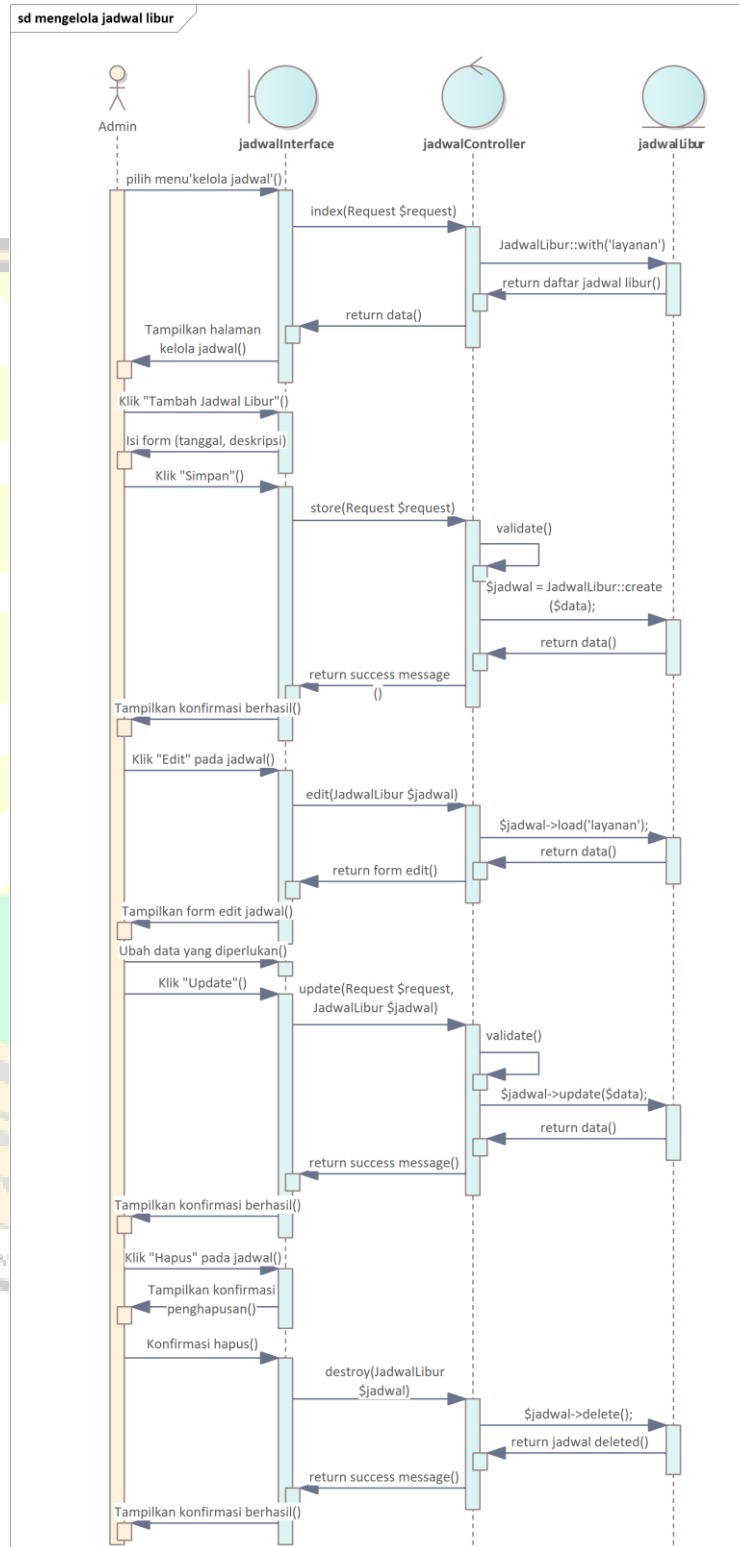
2. Edit profil



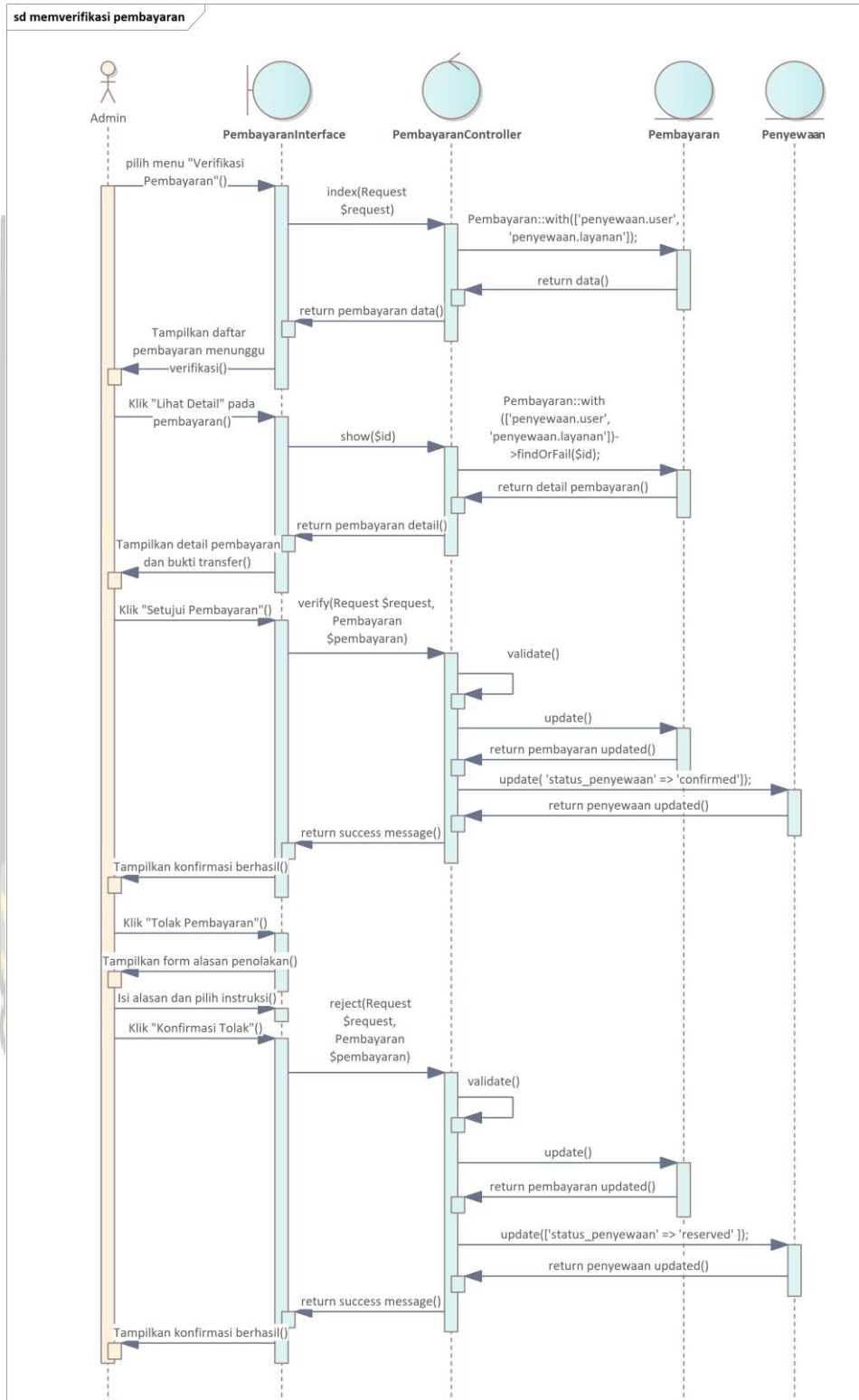
3. Mengelola layanan



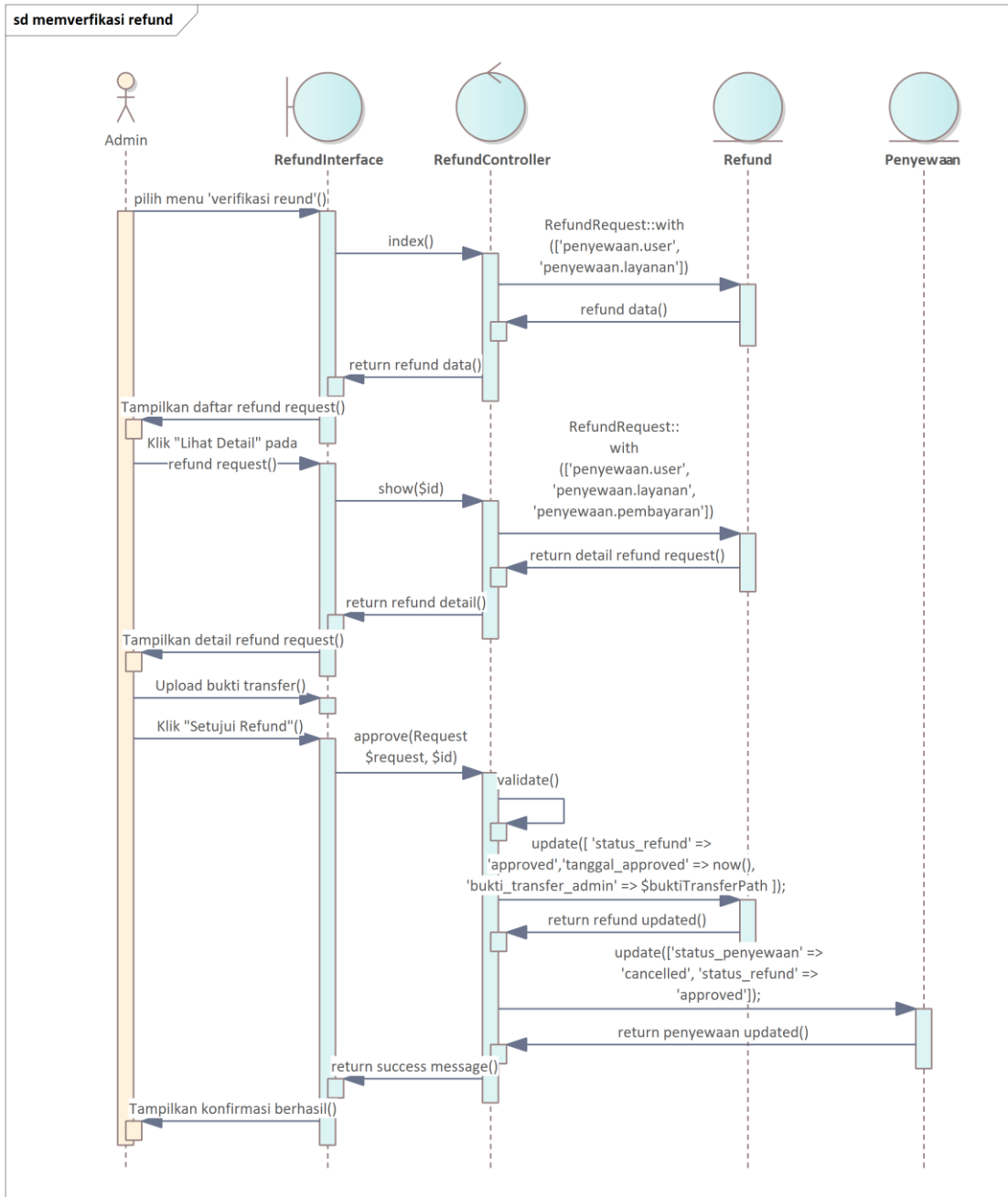
4. Mengelola jadwal libur



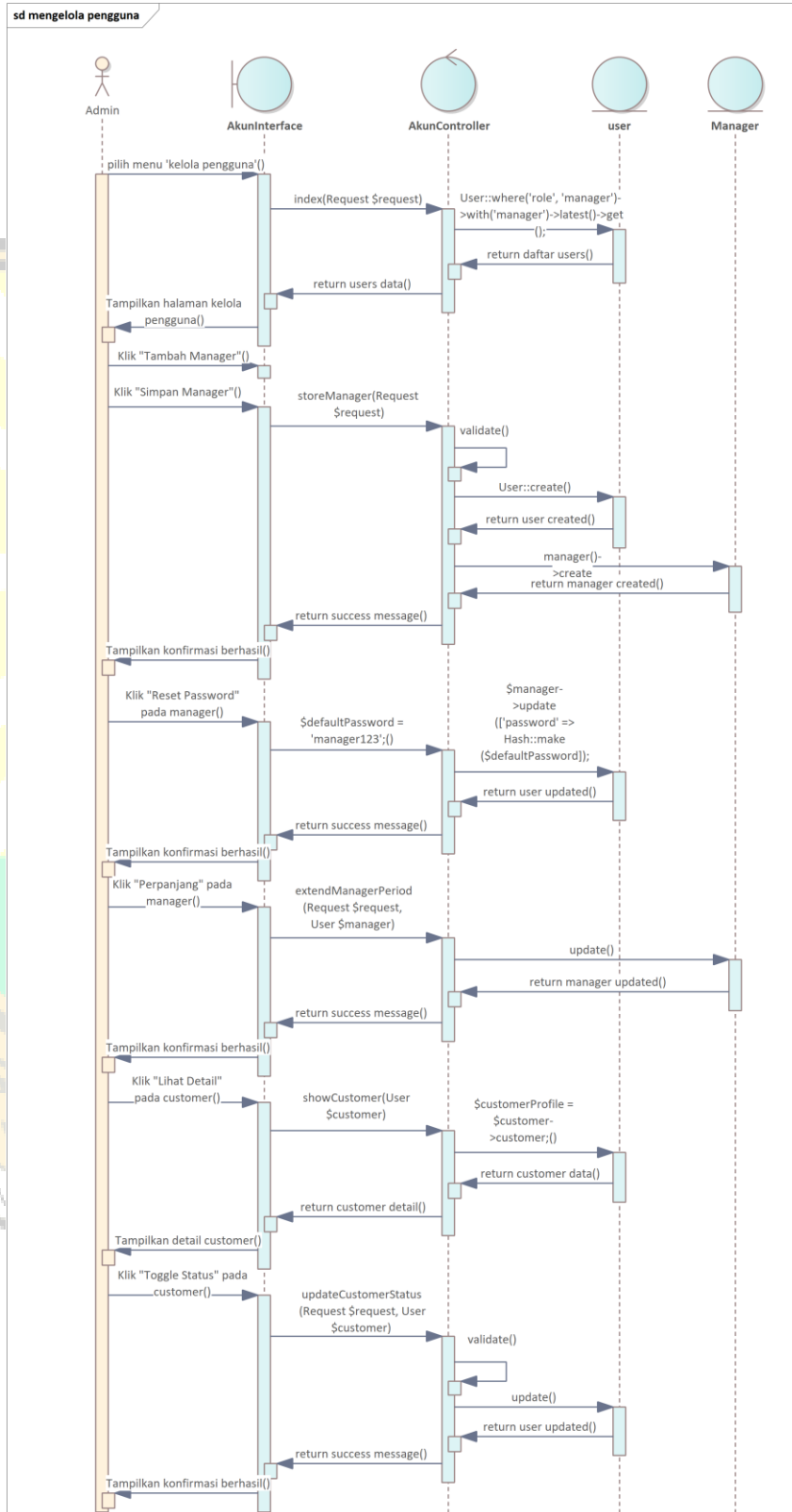
5. Memverifikasi pembayaran pelanggan



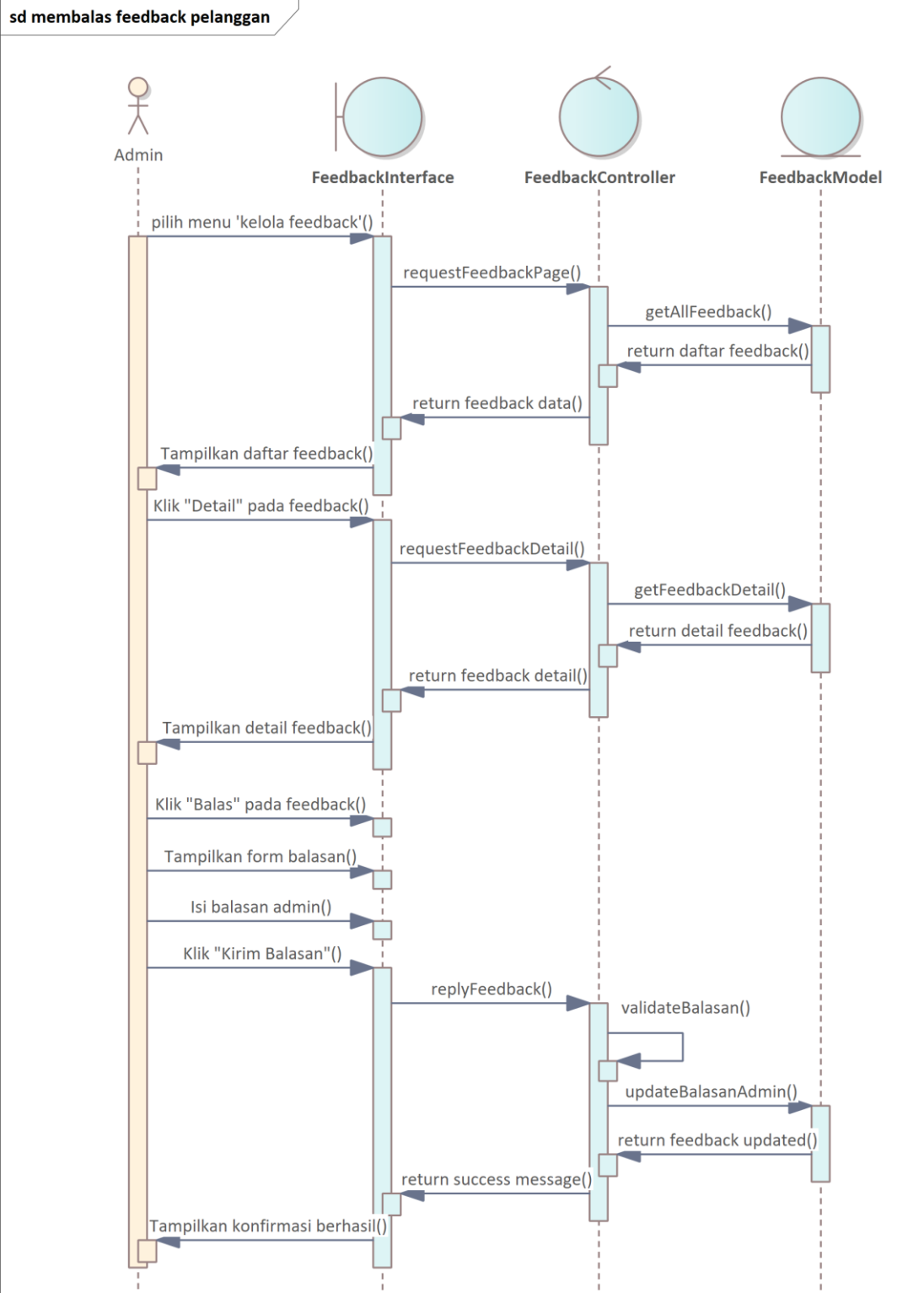
6. Memverifikasi Refund



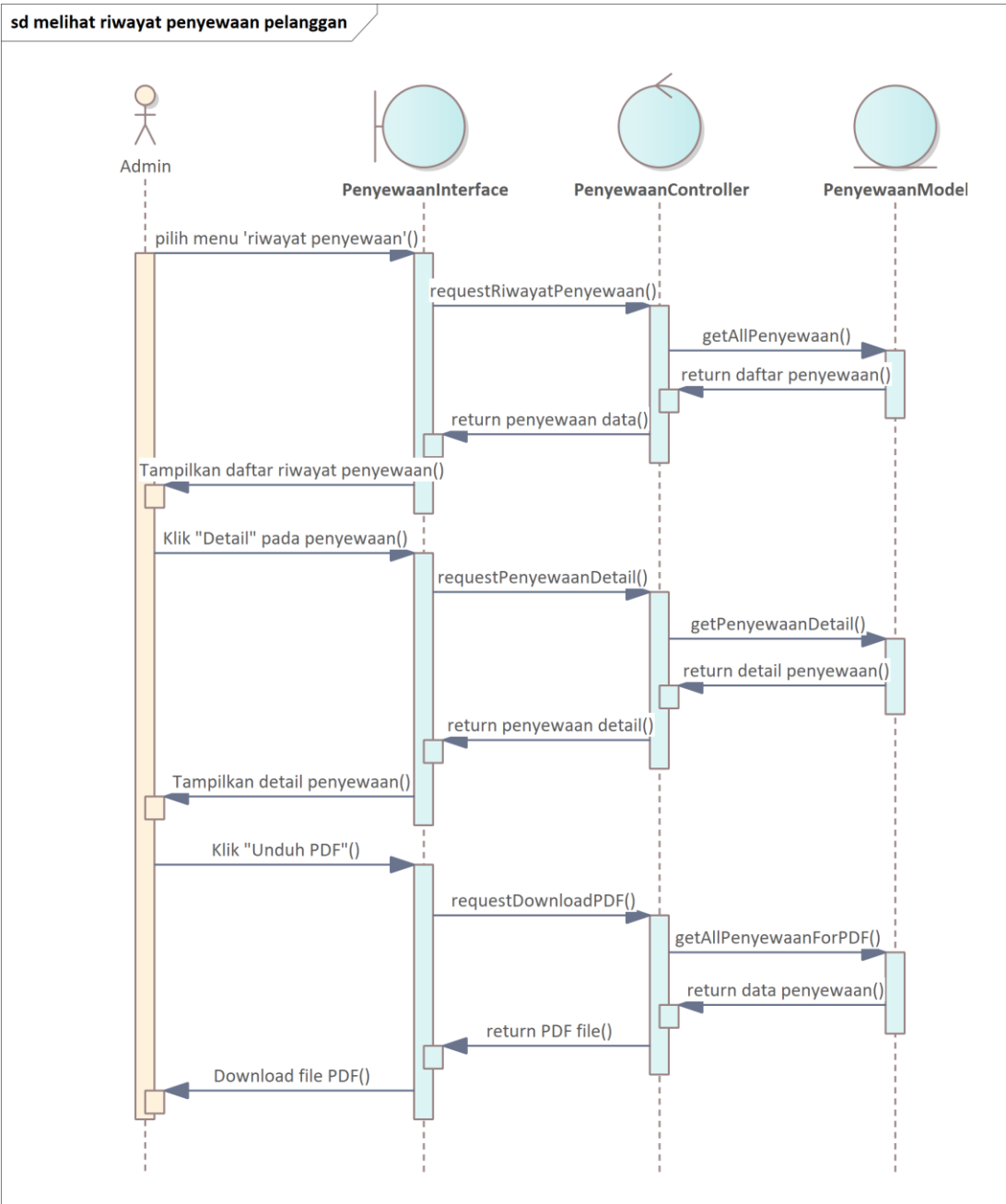
7. Mengelola akun



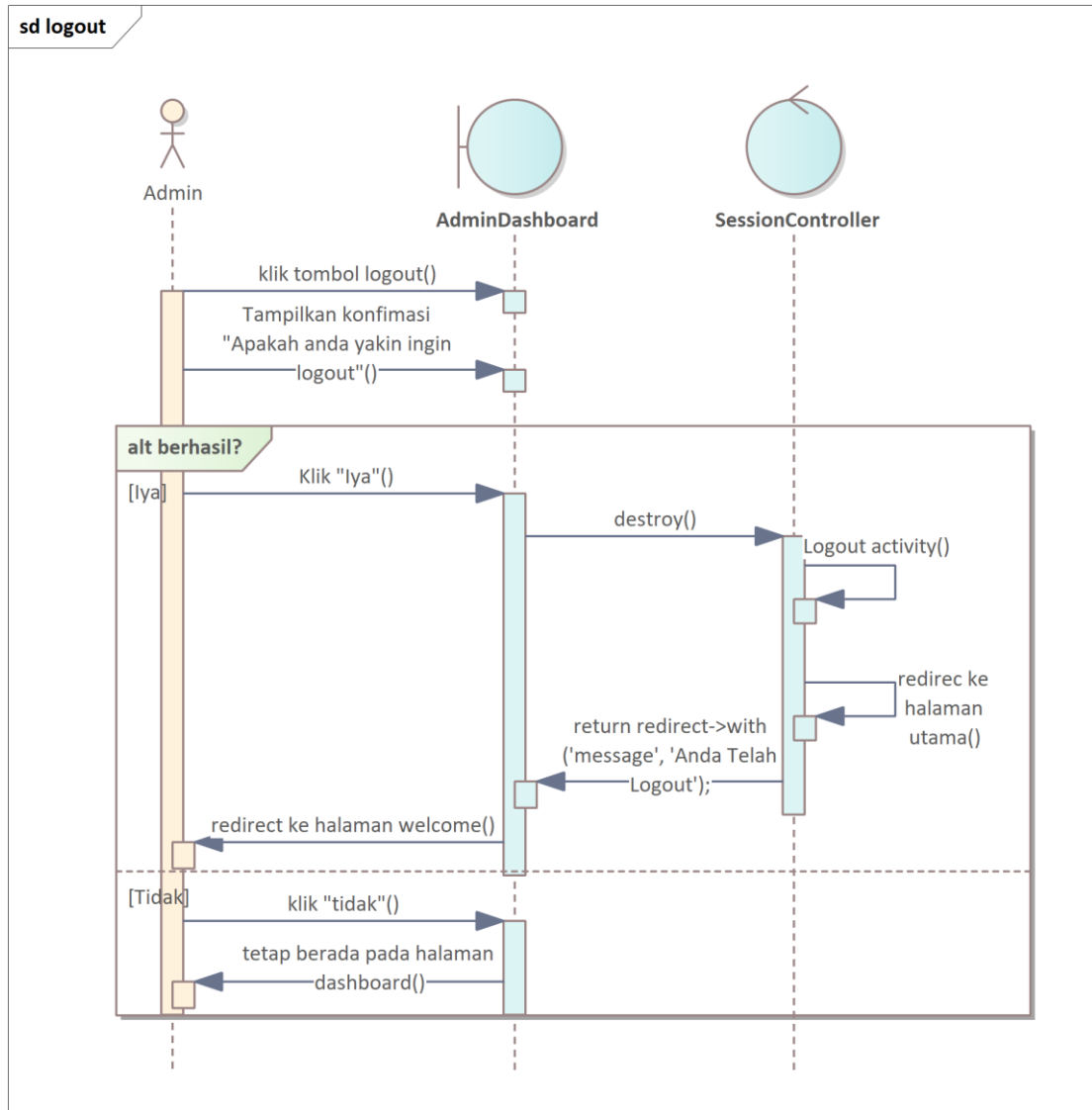
8. Melihat dan membalas umpan balik



9. Melihat Riwayat penyewaan

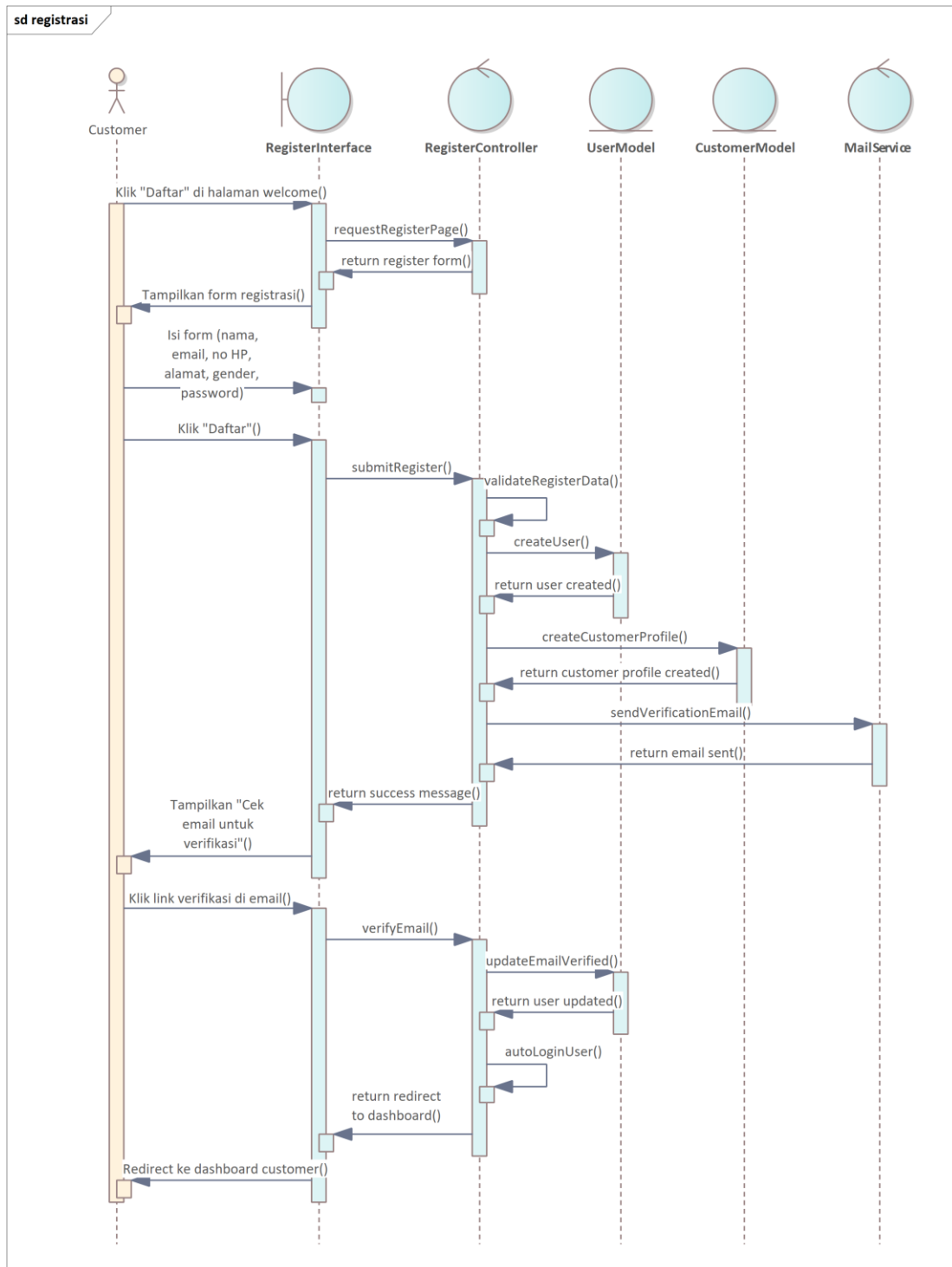


10. Logout

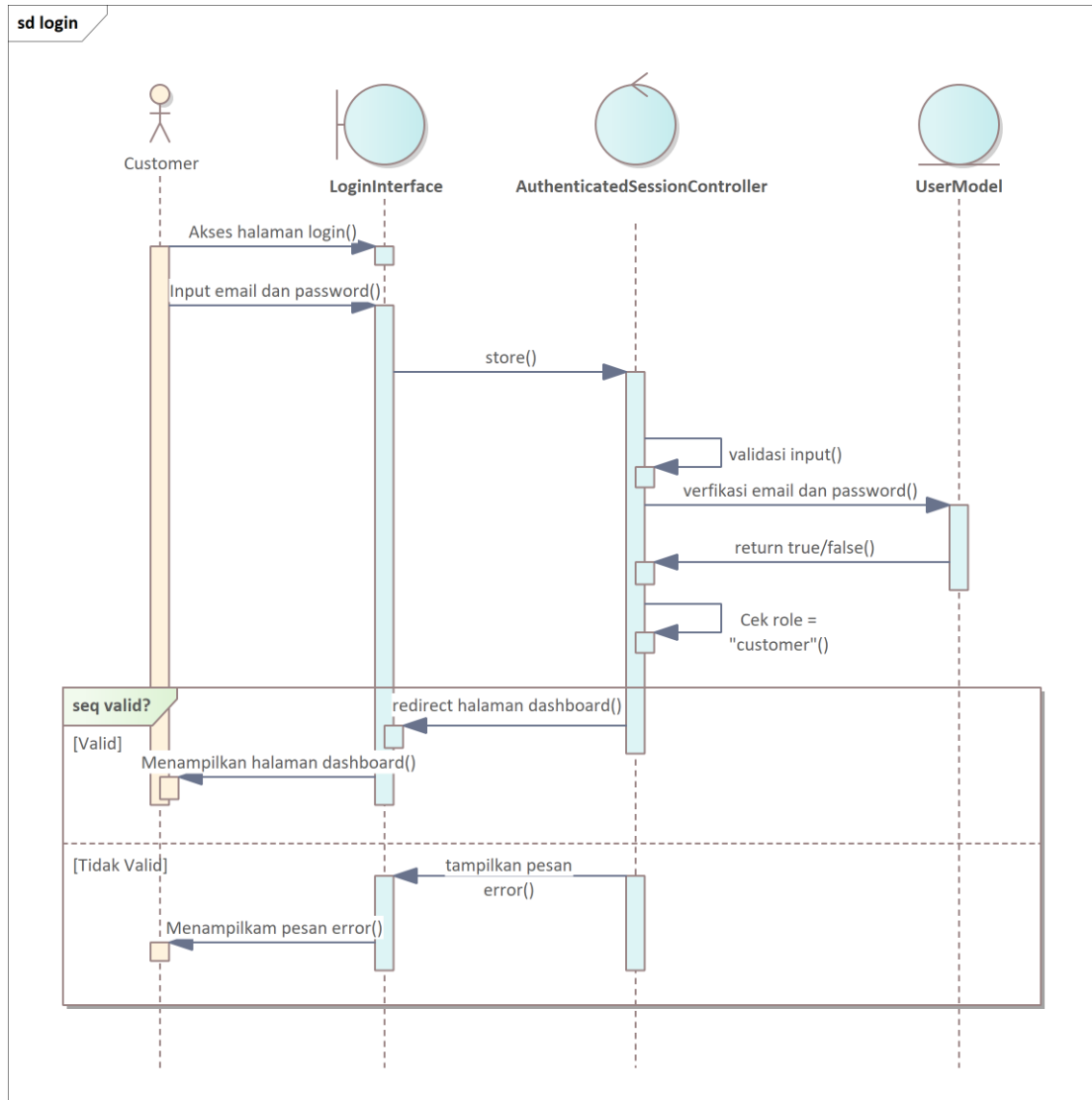


B. Pelanggan (Customer)

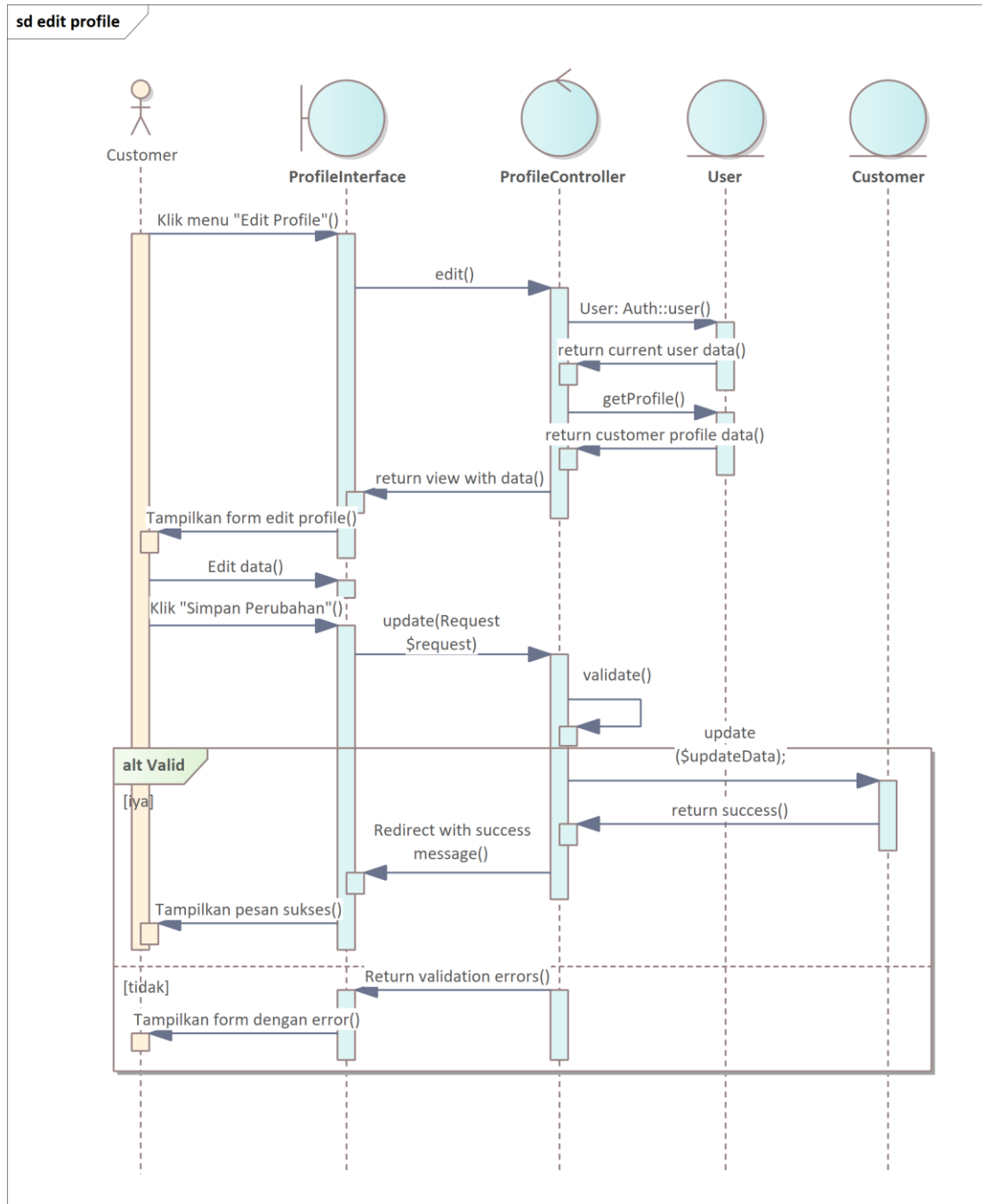
1. Registrasi



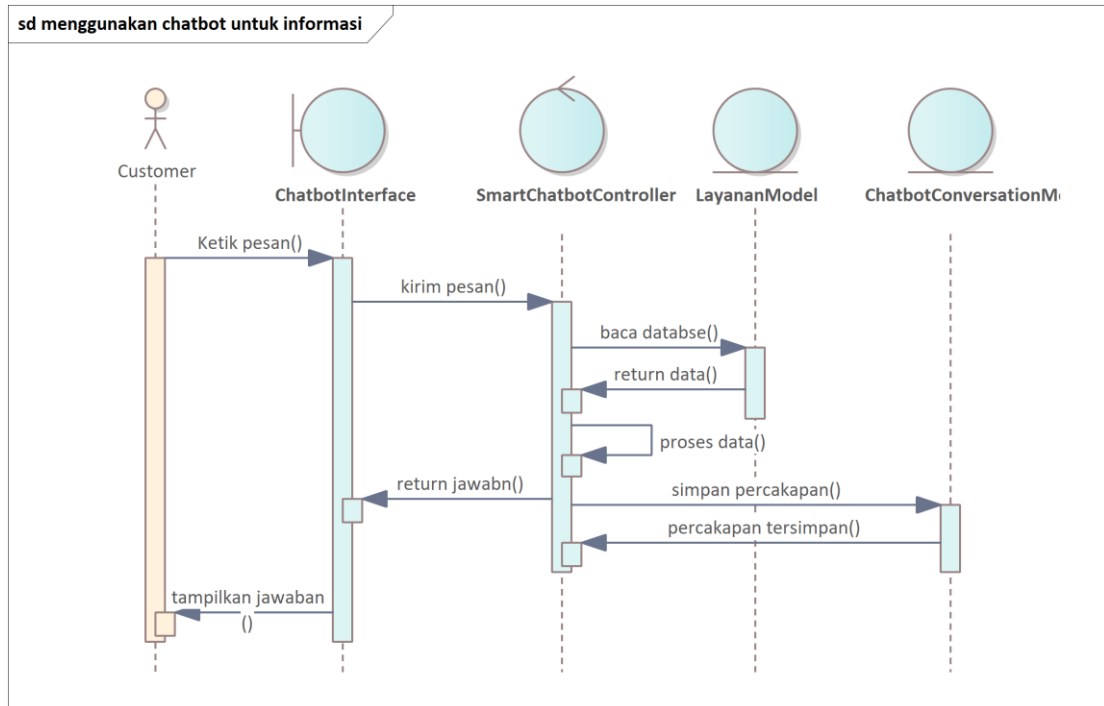
2. Login



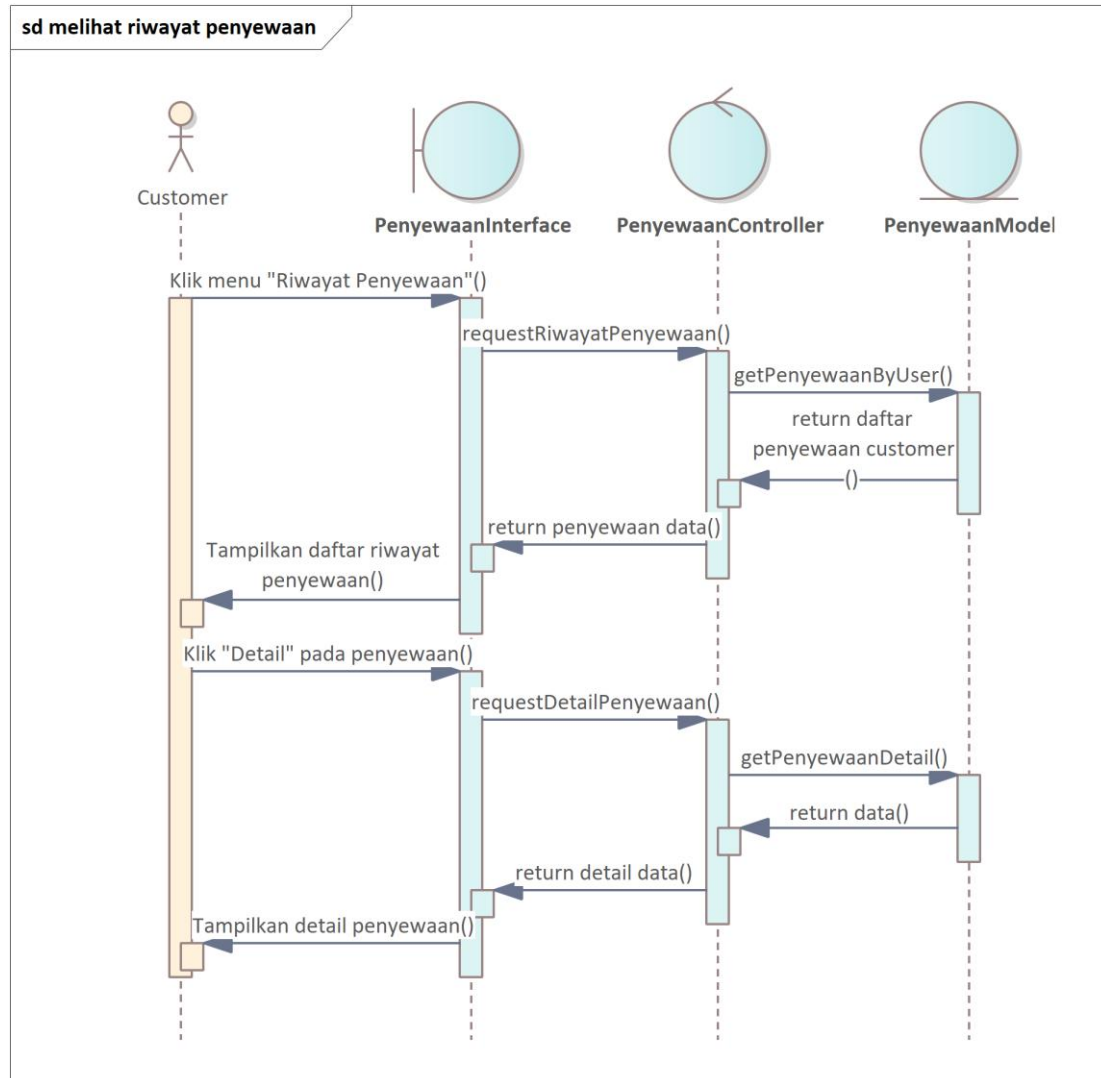
3. Edit profil



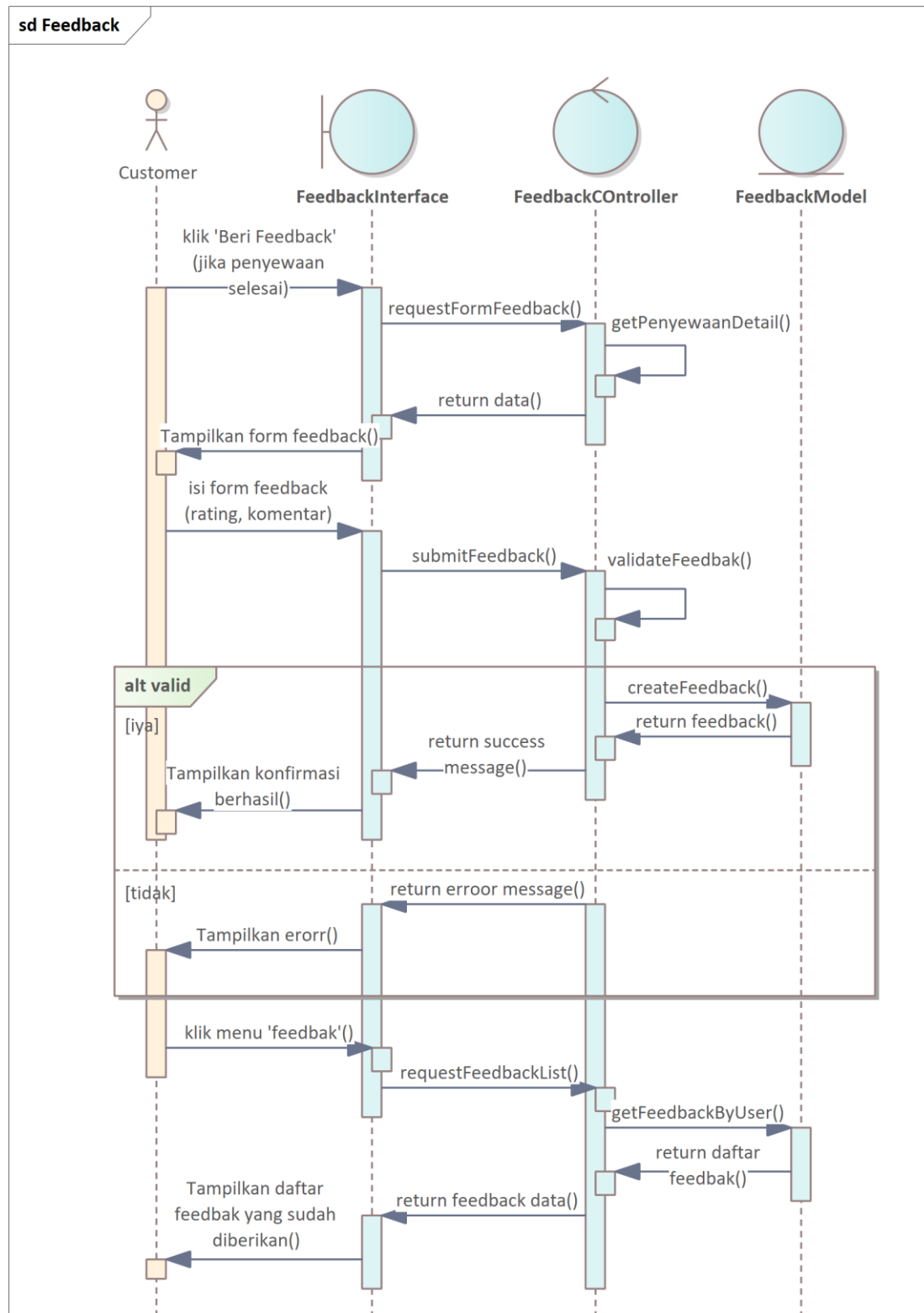
4. Menggunakan *chatbot* untuk informasi



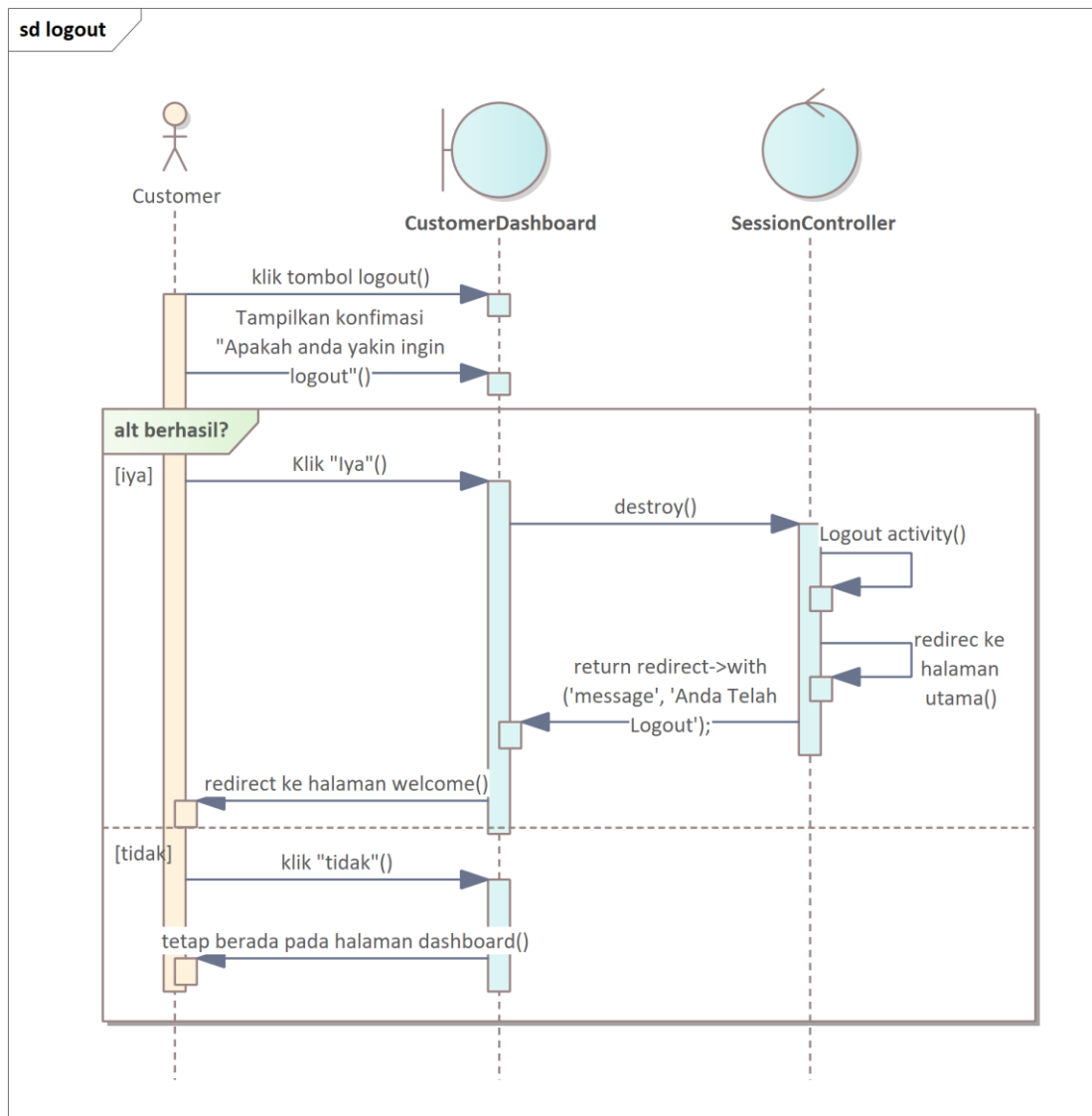
5. Melihat Riwayat penyewaan



6. Memberikan umpan balik

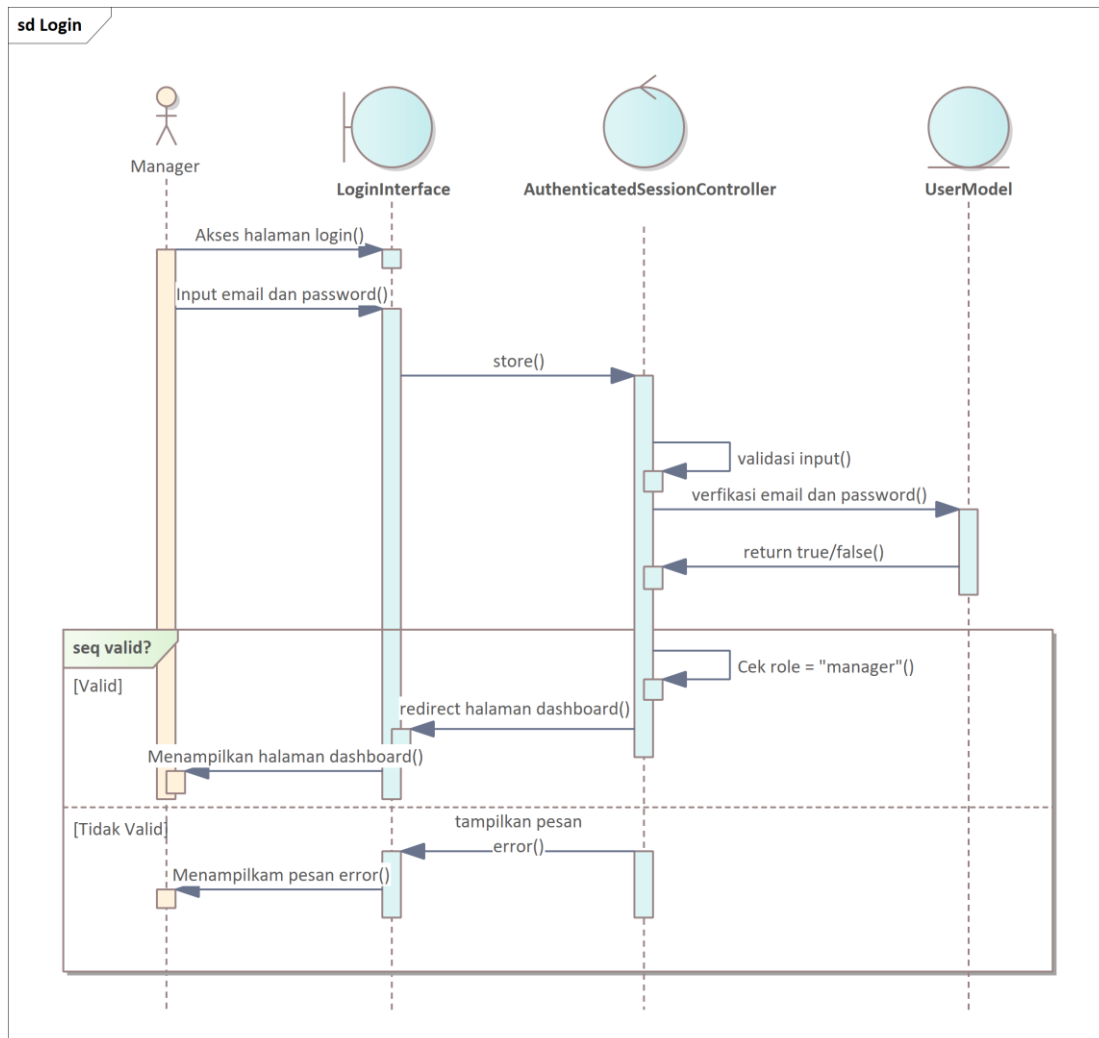


7. Logout

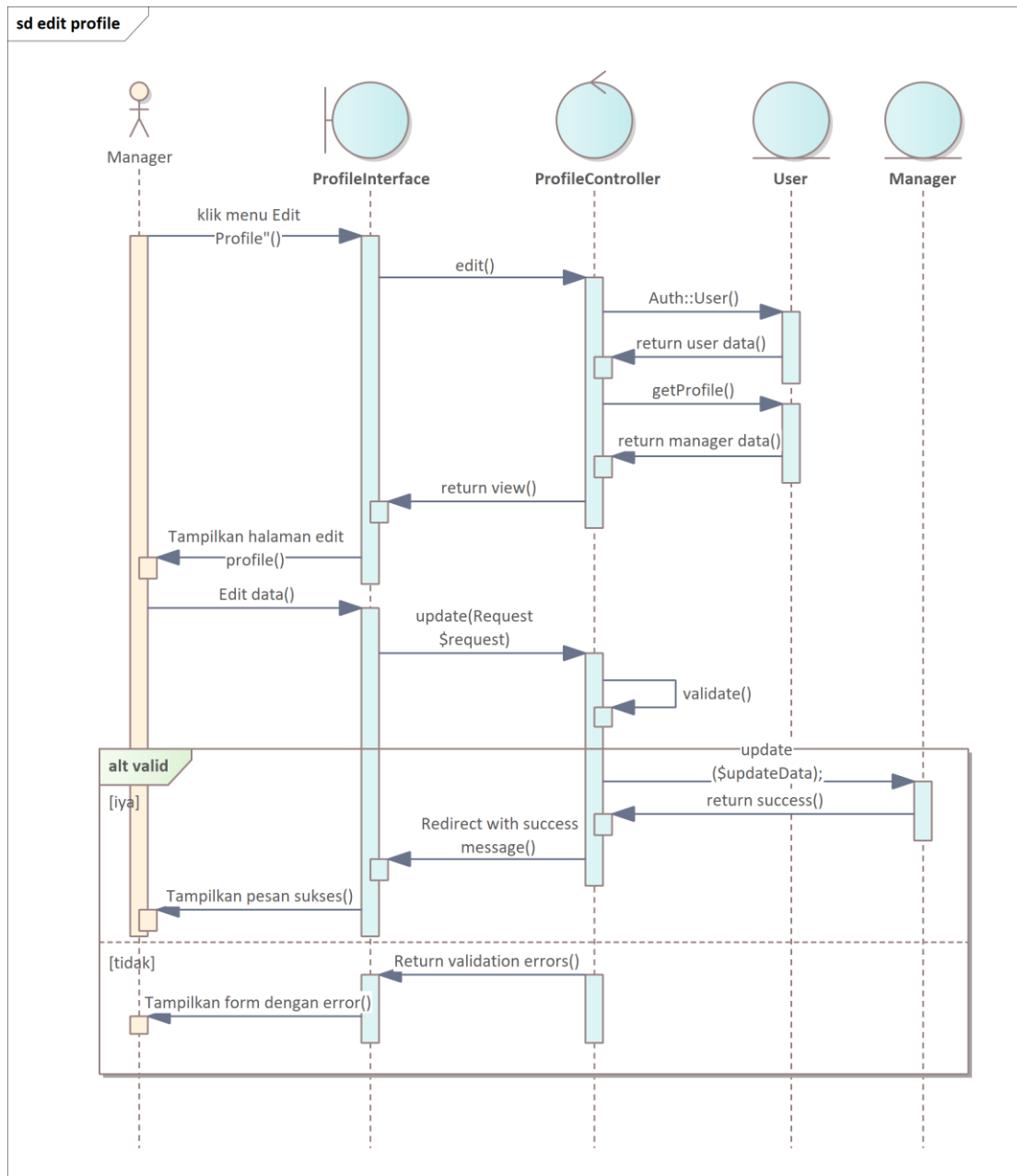


C. Manajer

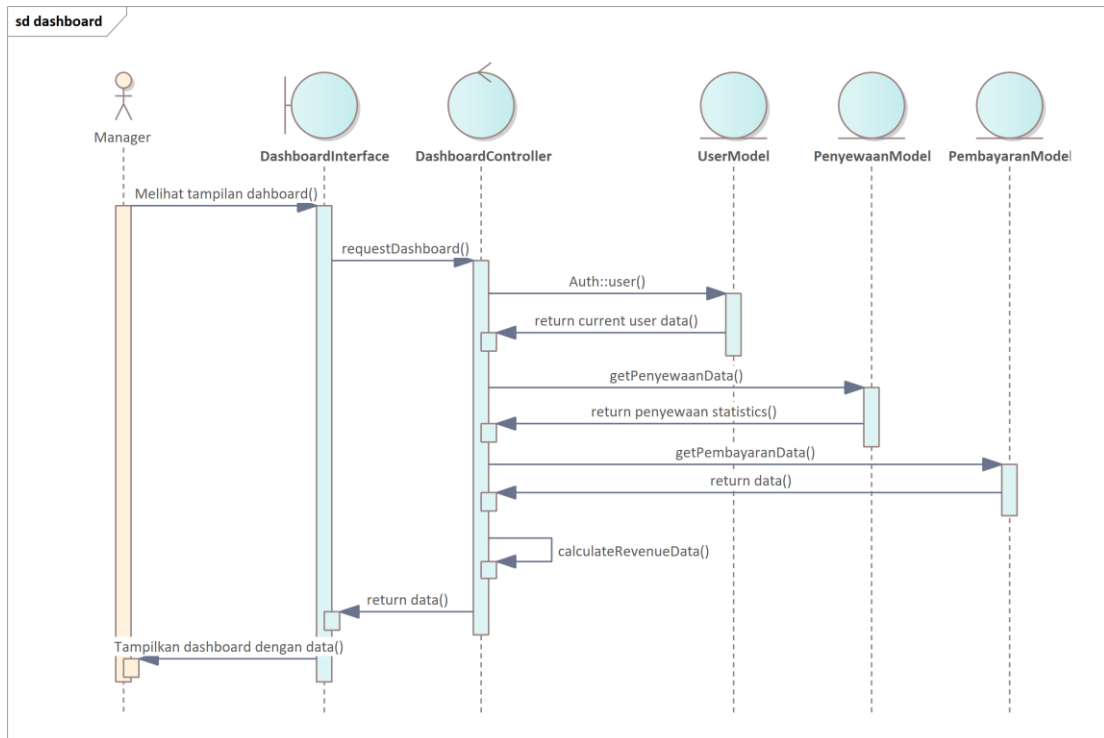
1. Login



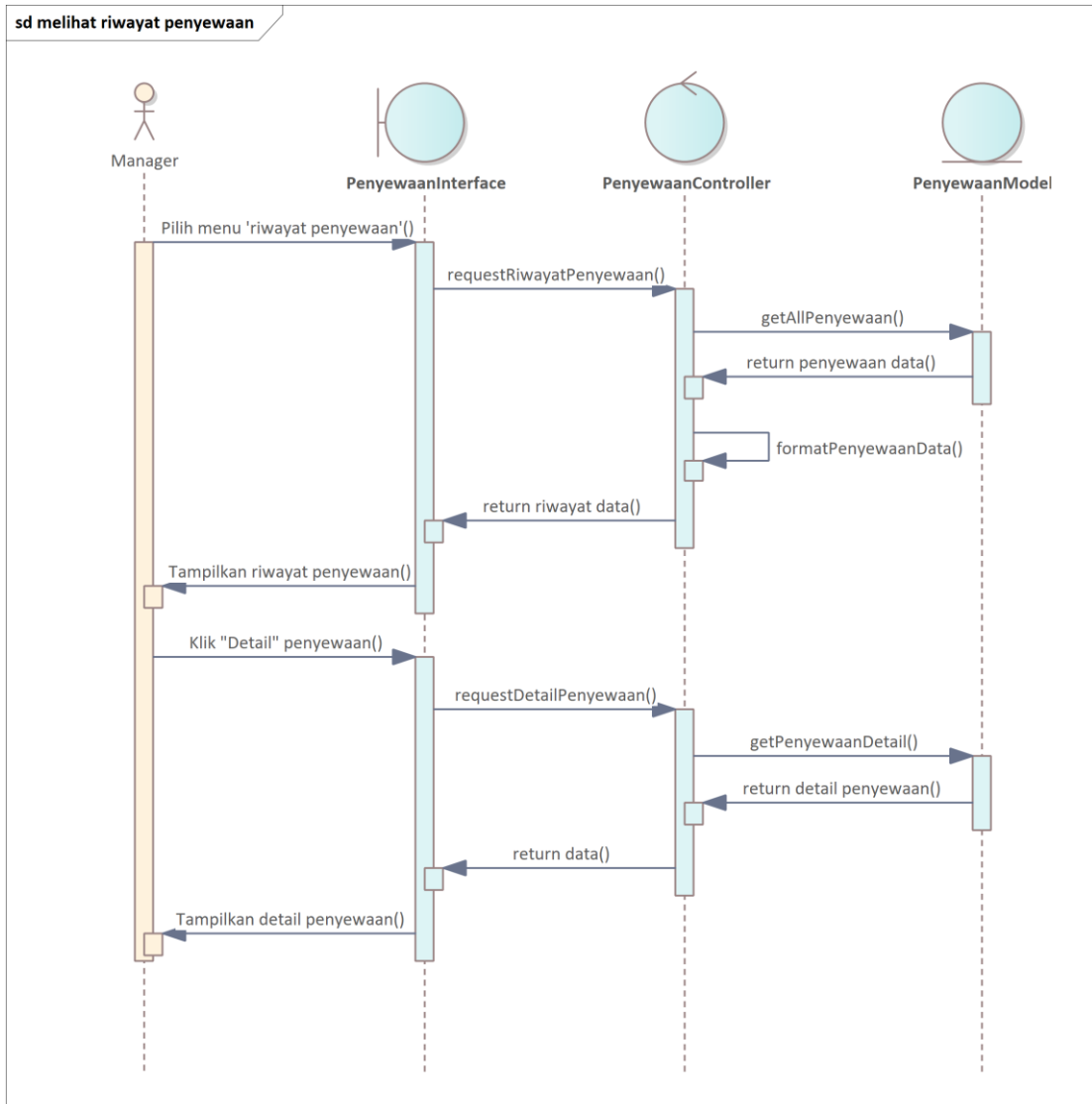
2. Edit profil



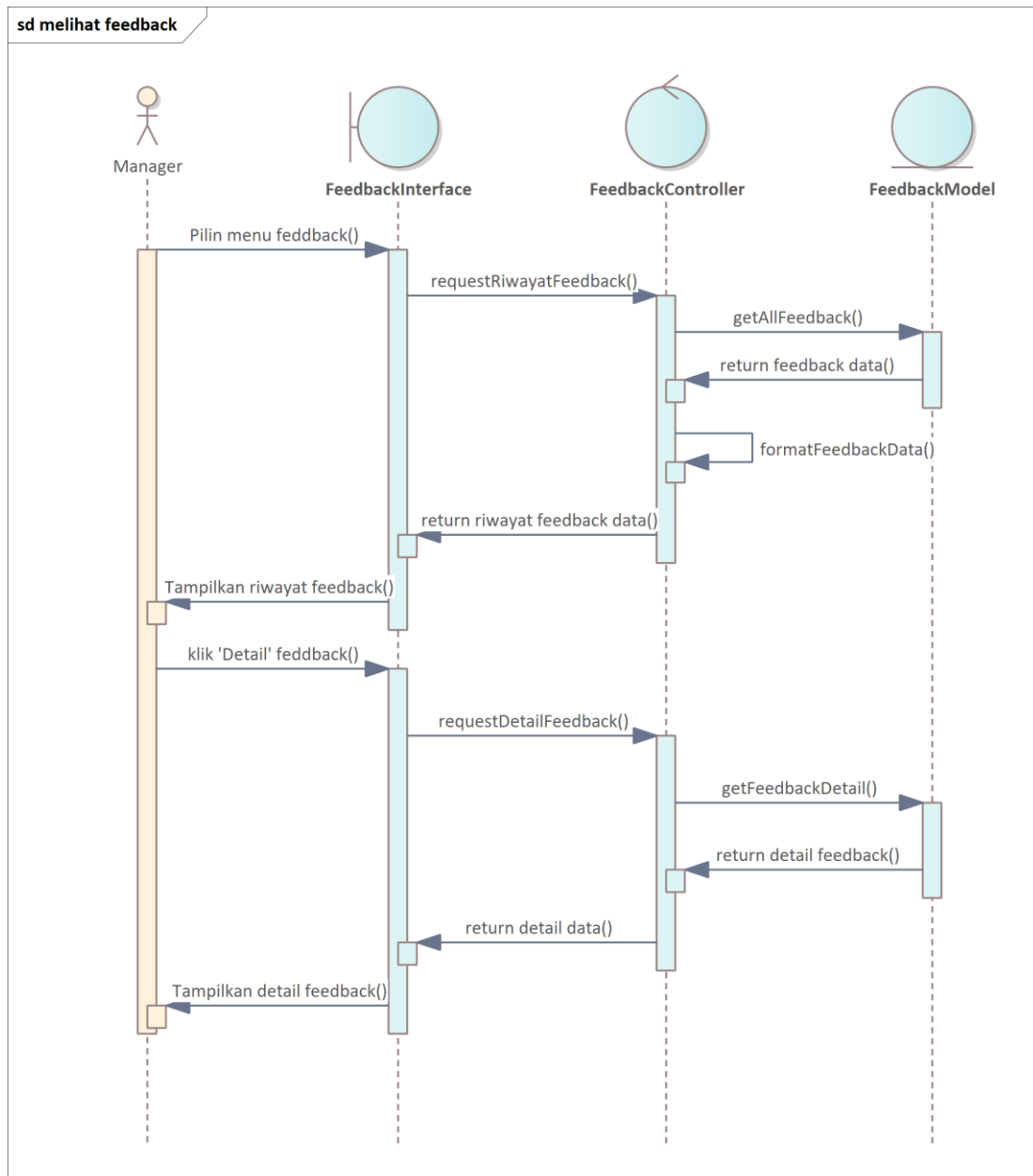
3. Dashboard manajer



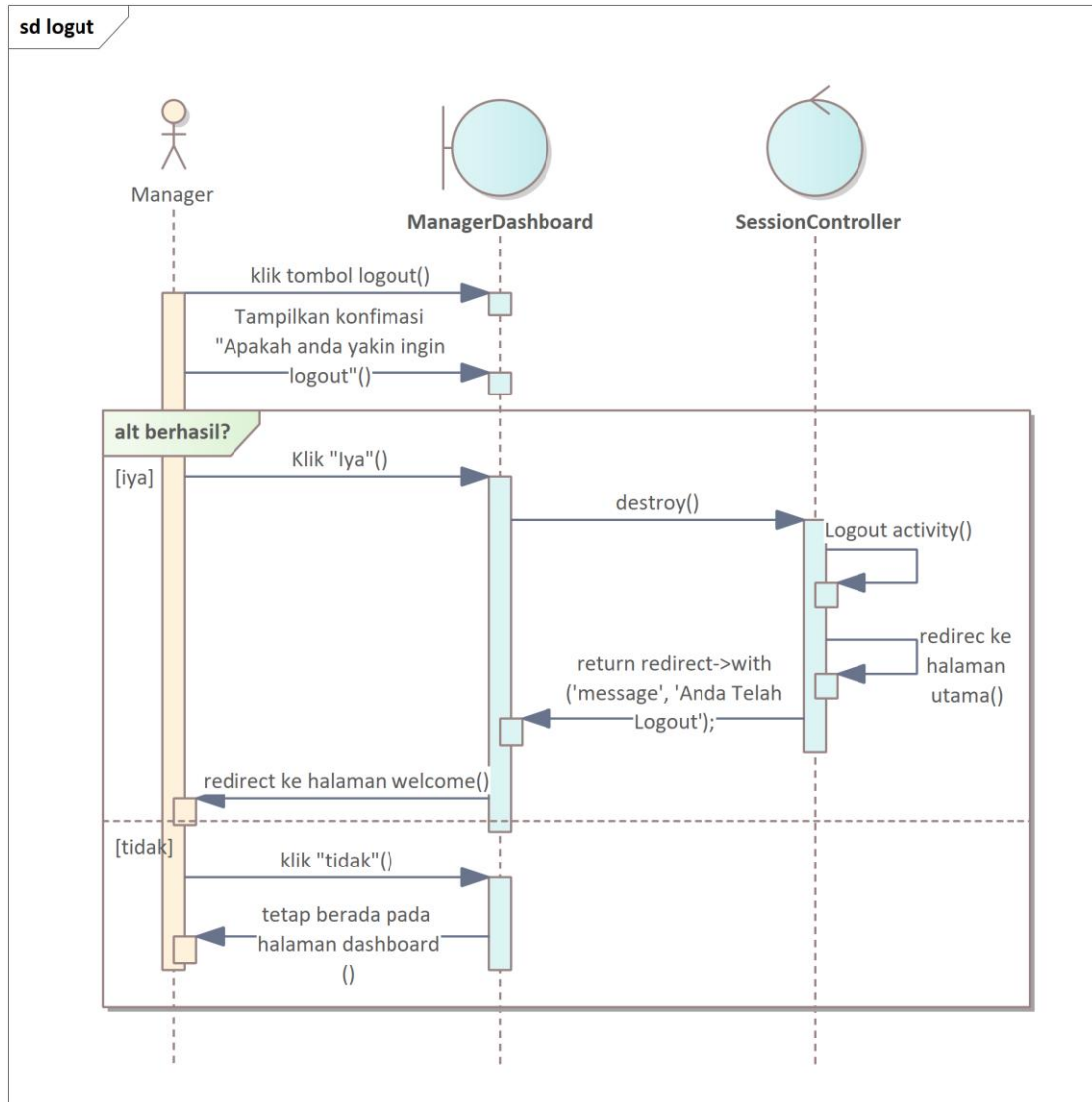
4. Melihat Riwayat Penyewaan



5. Melihat Riwayat Umpan balik



6. Logout





LAMPIRAN C

(Struktur Tabel – Lanjutan)

1. Tabel User

Nama Atribut	Tipe Data	Keterangan
id_user	CHAR (36)	PK, NOT NULL
nama_lengkap	VARCHAR(100)	NOT NULL
email	VARCHAR(100)	Unique, NOT NULL
no_hp	VARCHAR(20)	NOT NULL
role	Enum ('customer', 'admin', 'manajer')	NOT NULL, DEFAULT 'customer'
Password	VARCHAR (255)	NOT NULL
last_login	TIMESTAMP	NULL
alamat	TEXT	NOT NULL
is_active	BOOLEAN	NOT NULL, DEFAULT true
email_verified_at	TIMESTAMP	NULL
email_verification_token	VARCHAR (100)	NULL
profil_photo	VARCHAR(255)	NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

2. Tabel Customer (Pelanggan)

Nama Atribut	Tipe Data	Keterangan
id_customer	CHAR (36)	PK, NOT NULL
id_user	CHAR (36)	FK, NOT NULL
gender	Enum ('L', 'P')	NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

3. Tabel Admin

Nama Atribut	Tipe Data	Keterangan
id_admin	CHAR (36)	PK, NOT NULL
id_user	CHAR (36)	FK, NOT NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

4. Tabel Manajer

Nama Atribut	Tipe Data	Keterangan
id_manajer	CHAR (36)	PK, NOT NULL
id_user	CHAR (36)	FK, NOT NULL
periode_awal	DATE	NULL
periode_akhir	DATE	NULL
status_manajer	Enum ('aktif', 'nonaktif')	NOT NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

5. Tabel Fasilitas Tambahan

Nama Atribut	Tipe Data	Keterangan
id_fasilitas_tambahan	CHAR (36)	PK, NOT NULL
id_layanan	CHAR (36)	FK, NOT NULL
nama_fasilitas_tambahan	VARCHAR(255)	NOT NULL
deskripsi_fasilitas_tambahan	TEXT	NULL
harga_satuan	DECIMAL(10,2)	NOT NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

6. Tabel Notification

Nama Atribut	Tipe Data	Keterangan
id_notification	CHAR (36)	PK, NOT NULL
id_user	CHAR (36)	FK, NOT NULL
judul	VARCHAR (200)	NOT NULL
pesan	TEXT	NOT NULL
tipe	ENUM ('penyewaan', 'pembayaran', 'refund', 'informasi', 'umpan balik')	NOT NULL
is_read	BOOLEAN	DEFAULT false
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

7. Tabel *Chatbot_Conversations* (History)

Nama Atribut	Tipe Data	Keterangan
id_conversation	CHAR (36)	PK, NOT NULL
id_user	CHAR (36)	FK, NULL
session_id	VARCHAR(255)	NOT NULL
user_message	TEXT	NOT NULL
bot_response	TEXT	NOT NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

8. Tabel Refund *Requests*

Nama Atribut	Tipe Data	Keterangan
id_refund	CHAR (36)	PK, NOT NULL
id_penyewaan	CHAR (36)	FK, NOT NULL
jumlah_refund	DECIMAL(10,2)	NOT NULL
total_pembayaran	DECIMAL(10,2)	NOT NULL
nama_bank	VARCHAR(255)	NOT NULL

nomor_rekening	VARCHAR(255)	NOT NULL
nama_pemilik_rekening	VARCHAR(255)	NOT NULL
alasan_refund	TEXT	NULL
status_refund	ENUM (refund_requested', 'approved')	NOT NULL
tanggal_approved	TIMESTAMP	NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL
bukti_transfer_admin	VARCHAR(255)	NULL

9. Tabel Jadwal_Libur

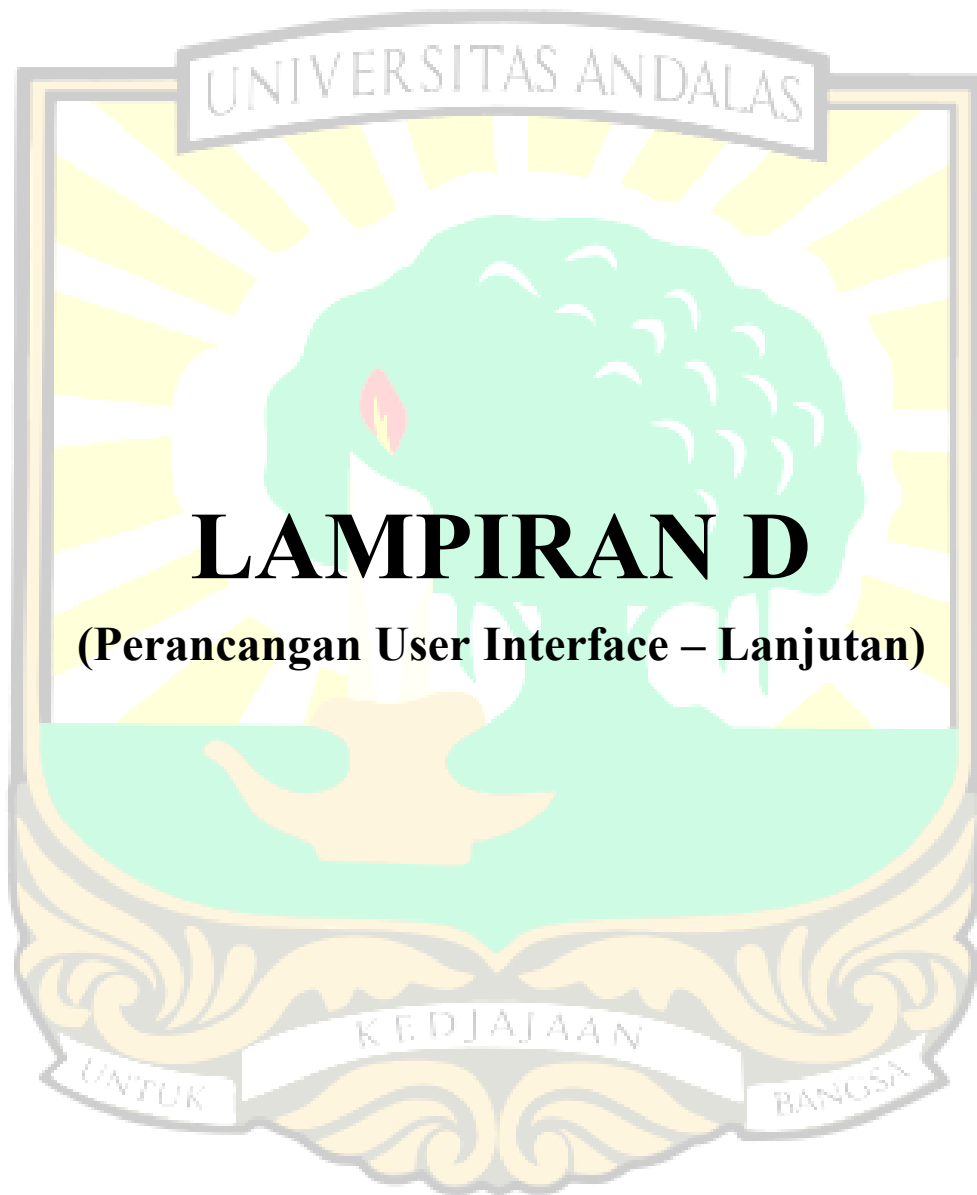
Nama Atribut	Tipe Data	Keterangan
id_jadwal	CHAR (36)	PK, NOT NULL
id_layanan	CHAR (36)	FK, NOT NULL
tanggal	DATE	NULL
hari	ENUM ('senin', 'selasa', 'rabu', 'kamis', 'jumat', 'sabtu', 'minggu')	NULL
jenis	ENUM ('harian', tanggal_tertentu')	NOT NULL
keterangan	TEXT	NULL
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL

10. Tabel Rekening Bank

Nama Atribut	Tipe Data	Keterangan
id_rekening	CHAR (36)	PK, NOT NULL
nama_bank	VARCHAR(50)	NOT NULL
nomor_rekening	VARCHAR(30)	NOT NULL
nama_pemilik	VARCHAR(100)	NOT NULL

instruksi_transfer	TEXT	NULL
is_default	BOOLEAN	DEFAULT false
created_at	TIMESTAMP	NULL
updated_at	TIMESTAMP	NULL



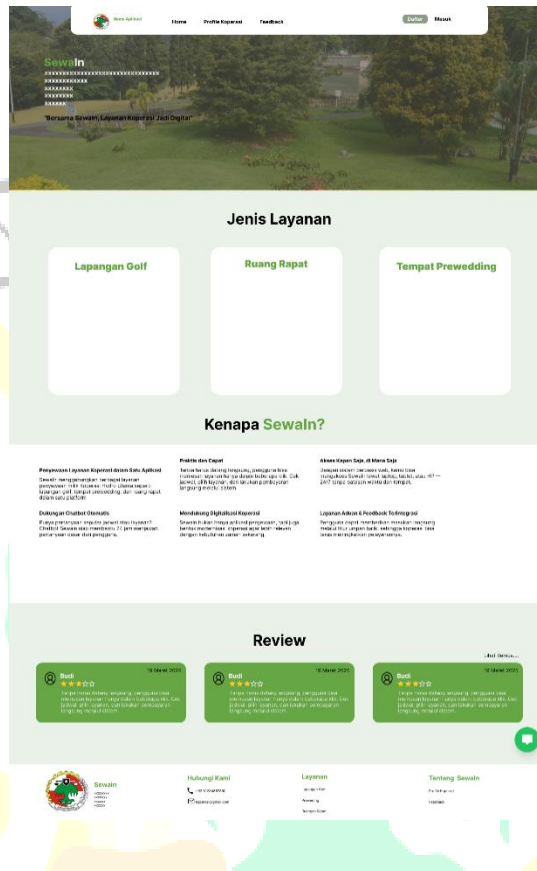


LAMPIRAN D

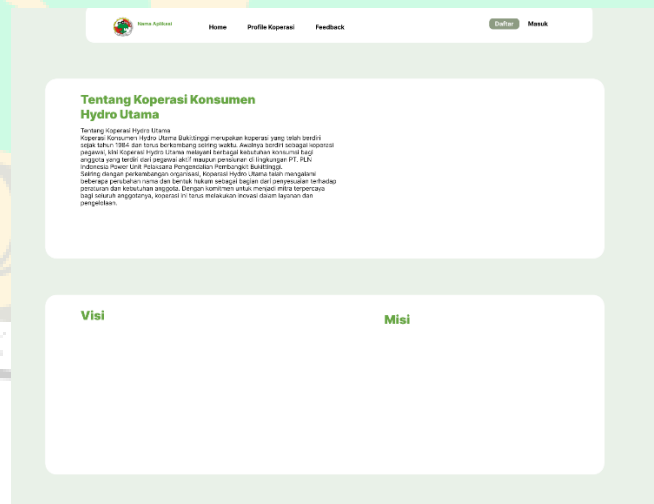
(Perancangan User Interface – Lanjutan)

A. Utama

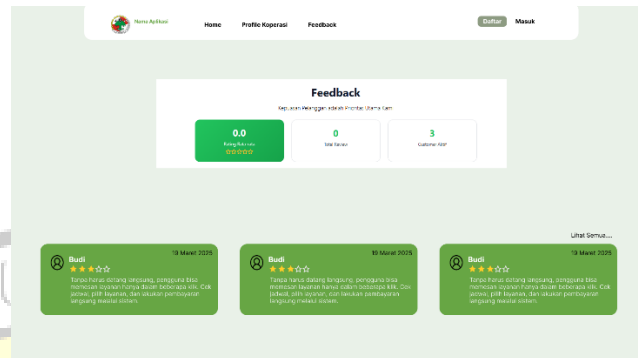
1. Welcome Page



2. Profil Organisasi

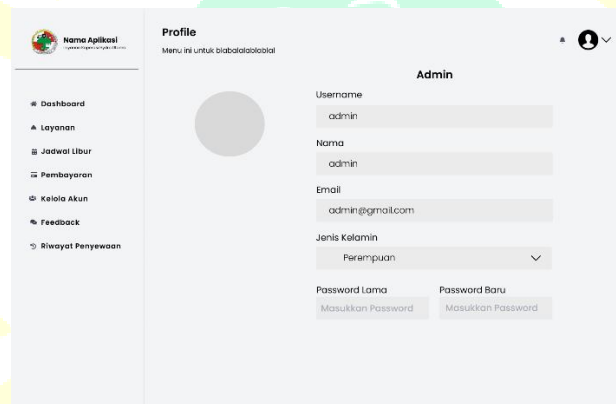


3. Umpan balik

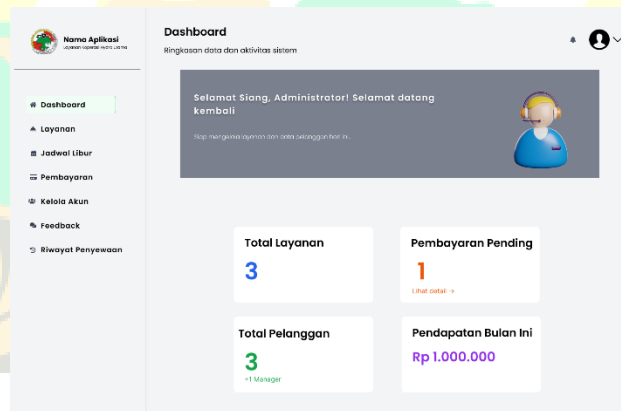


B. Admin

1. Edit Profil

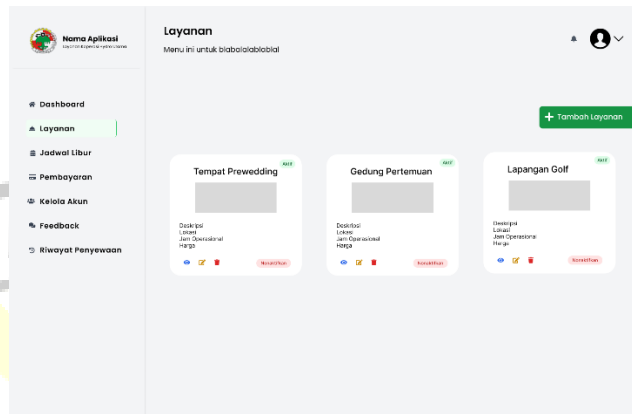


2. Dashboard

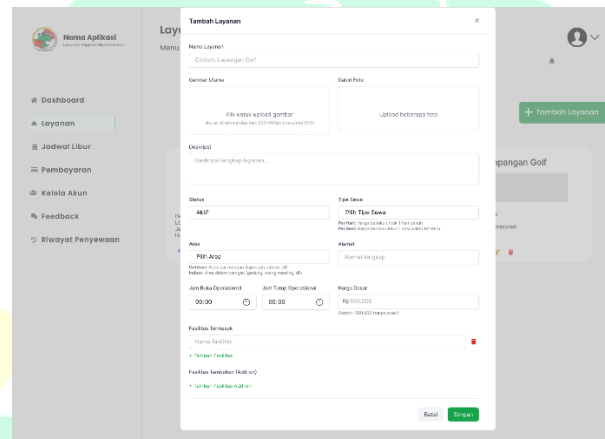


3. Layanan

1) Tambah Layanan



2) Edit Layanan



[illegible]

Nama Aplikasi

Universitas Pendidikan Indonesia

Dashboard

Layanan

Jadwal libur

Pembayaran

Kelola Akun

Feedback

Riwayat Penyewaan

Refund

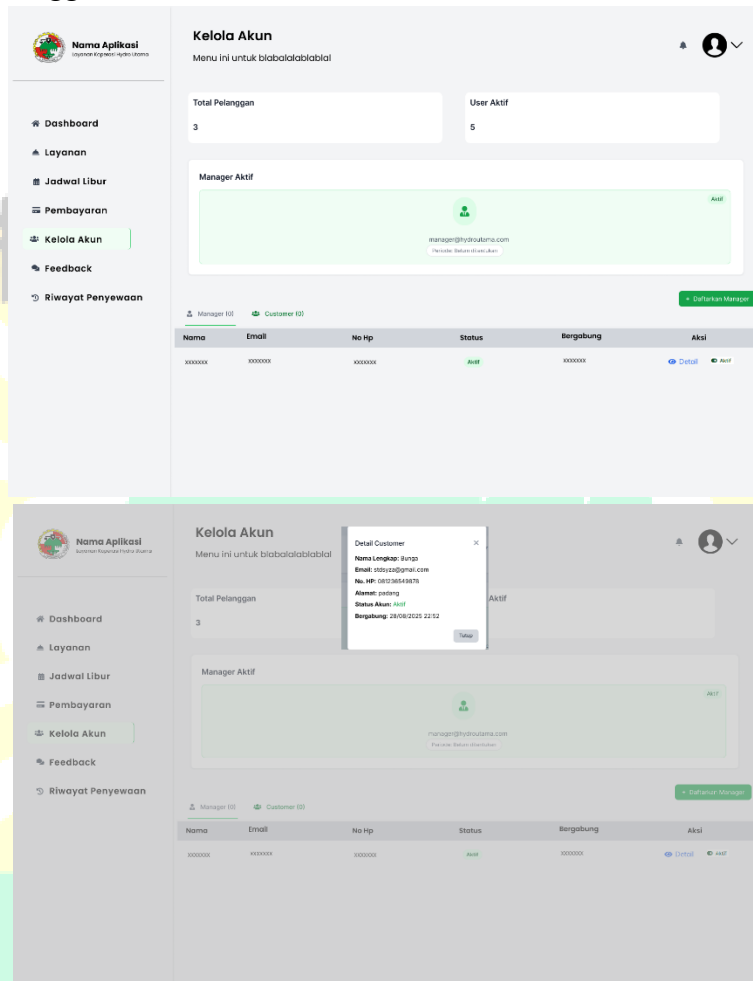
Menu ini untuk klaim refund

Pembayaran

Refund Request

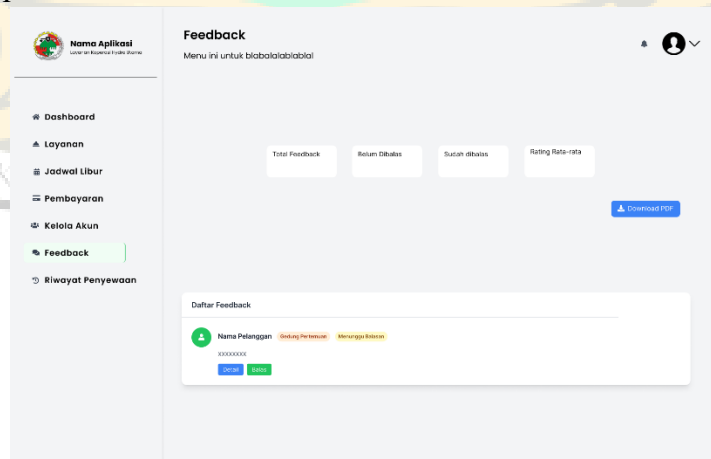
ID Refund	Pekangan	Layanan	Jumlah Refund	Status	Tanggal	Aksi
XXXXX	Ayo XXXX	Layanan Dof	Rp 100.000	PENDING	01/04/2025	

2) Akun Pelanggan



8. Umpan balik

1) Daftar umpan balik



The screenshot displays a web application interface. On the left is a sidebar menu with the following items: 'Nama Aplikasi' (Logo of Universitas Negeri Semarang), 'Dashboard', 'Layanan', 'Jadwal Laburan', 'Pembayaran', 'Kelola Akun', 'Feedback' (highlighted in green), and 'Biwayat Penyewaan'. The main content area is titled 'Detail Feedback' and contains the text 'Menu ini untuk bla bla bla bla bla!'. On the right side of the main content area, there is a 'Kembali' button. Below the main content area, there is a detailed view of a feedback entry for 'Gedung Pertemuan'. This entry includes the feedback ID '01090500-0000-715ae-9b2a-b0dca779c10f', the date and time '01 Sep 2025 20:46', the customer name 'Syazza Intan Banya', the email 'yzazzyy@gmail.com', a rating of 4.5 stars, and a comment 'Fasilitasnya oke'. At the bottom of the feedback entry, there is a 'Status Feedback' section indicating that the feedback has been processed by an administrator.

Nama Aplikasi
Layanan Pemasaran Digital Jarak Jauh

🔔

- 🏠 Dashboard
- ▲ Layanan
- 📅 Jadwal Libur
- 👤 Pembayaran
- 👤 Kelola Akun
- 📢 Feedback
- [📖 Riwayat Penyewaan](#)

Riwayat Penyewaan

Menu ini untuk biablabalablabla!

Filter Layanan
Filter Bulan
Filter Tahun

LAYANAN	POLARIGGAN	TANGGAL BERA	STATUS	TOTAL BAKER	TINDAKAN
Gelang Perampuan	Syadza Iman Ganga	28 Agustus 2023	Lunas	Rp 11.000.000	0

1. Edit Profil

1. Edit Profil

2. Dashboard

3. Riwayat Penyewaan



PT. KALAMATI

Unit of Rental Services

[Home](#)
[Profil Koperasi](#)
[Riwayat Sewa](#)




Riwayat Sewa

[Pin Layanan](#)

No	Layanan	Tanggal Sewa	Status	Total Bayar	Bukti Sewa	Aksi
1	Lapangan Golf	14 Maret 2025	Pesos	Rp. 500.000	Rp. 300.000	Detail
2	Ruang Rapat	14 Maret 2025	Selesai	Rp. 500.000	Rp. 300.000	Detail
3	Panasonic	14 Maret 2025	Batal	Rp. 500.000	Rp. 300.000	Detail

D. Manajer

1. Edit Profil

1. Edit Profil

2. Dashboard



Nama Aplikasi

aplikasi kerja dan manajemen

Dashboard

Riwayat Feedback

Riwayat Penyewaan



Dashboard

Ringkasan data dan aktivitas sistem

Selamat Sore, Manager! Selamat datang kembali

Pantau dan evaluasi semua aktivitas penyewaan untuk menjaga kualitas layanan



Grafik Pendapatan Bulanan



3. Riwayat penyewaan



Nama Aplikasi

Dashboard

Riwayat Feedback

Riwayat Penyewaan

Riwayat Penyewaan

Ringkasan data dan aktivitas sistem

Pilih Layanan

Rentang Tanggal

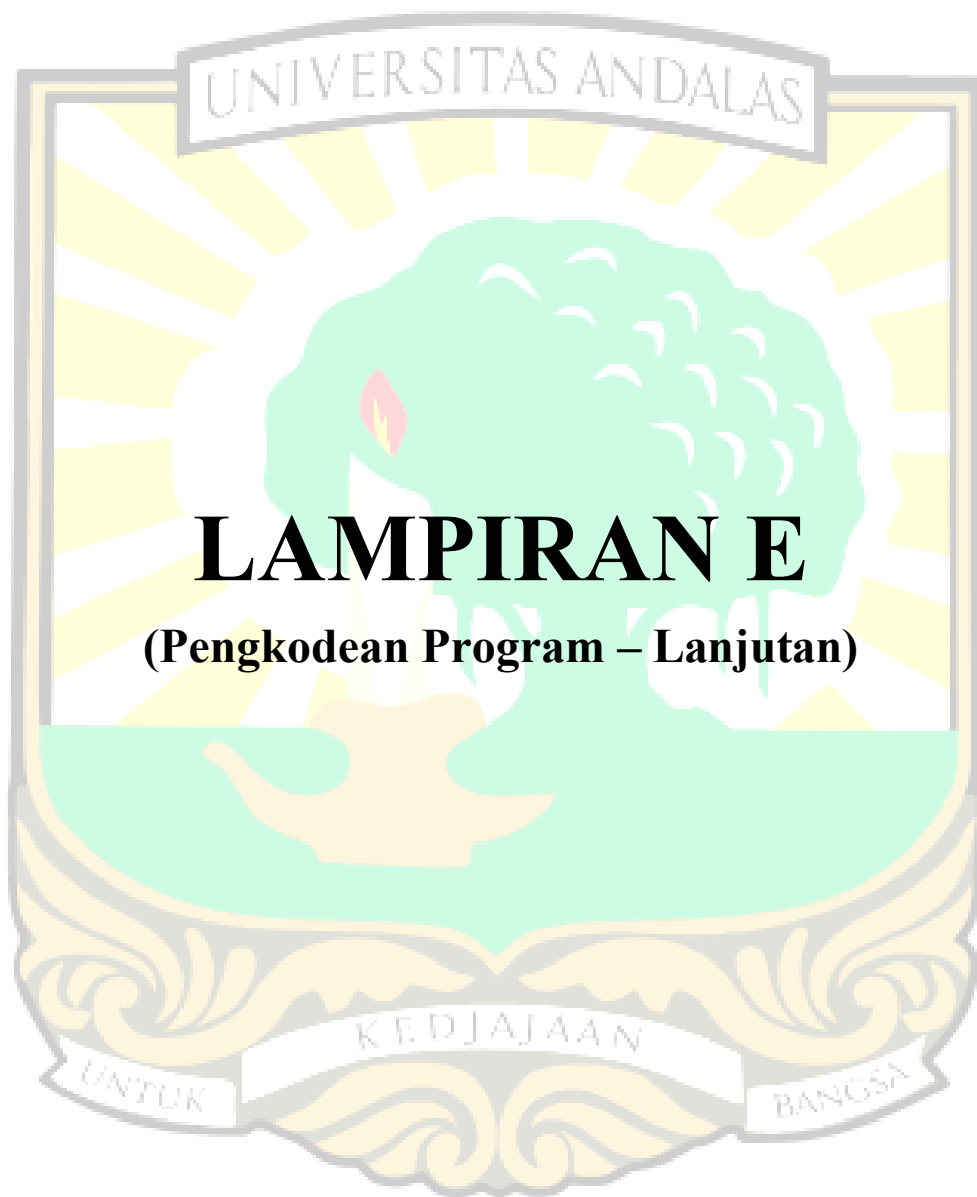
S/D

Download Laporan

Layanan	Nama	Email	Tanggal	Total Bayar	Status	Aksi
Lapangan Golf	Jiya xxxxx	xyab@gmail.com	05 Juni 2025	Rp 100.000	Lunas	

4. Umpan balik





LAMPIRAN E

(Pengkodean Program – Lanjutan)

A. Controller

1. Layanan.php

```
<?php

namespace App\Http\Controllers;

use Carbon\Carbon;
use App\Models\Layanan;
use App\Models\Penyewaan;
use App\Models\Pembayaran;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use App\Models\JadwalLibur;

class LayananController extends Controller
{
    private function getActiveBookings($userId)
    {
        return Penyewaan::where('id_user', $userId)
            ->whereIn('status_penyewaan', ['pending', 'reserved'])
            ->with(['pembayaran', 'layanan'])
            ->get()
            ->filter(function($booking) {
                // Skip booking yang sudah dibatalkan karena refund
                if ($booking->status_penyewaan === 'cancelled') {
                    return false;
                }
                return true;
            });
    }

    private function getUnpaidBookings($userId)
    {
        return $this->getActiveBookings($userId)->filter(function($booking) {
            $totalPaid = $booking->pembayaran()
                ->where('status_pembayaran', 'lunas')
                ->sum('jumlah_bayar');

            return $totalPaid < $booking->total_harga;
        });
    }

    public function index(Request $request)
    {
        if (Auth::user()->role === 'admin') {
            return $this->adminIndex($request);
        } else {
```

```

        return redirect()->route('customer.dashboard');
    }
}

public function adminIndex(Request $request)
{
    $query = Layanan::query();

    // Search functionality
    if ($request->filled('search')) {
        $search = $request->search;
        $query->where(function ($q) use ($search) {
            $q->where('nama_layanan', 'like', '%' . $search . '%')
                ->orWhere('lokasi', 'like', '%' . $search . '%')
                ->orWhere('deskripsi_layanan', 'like', '%' . $search . '%');
        });
    }

    if ($request->filled('status')) {
        $query->where('status', $request->status);
    }

    // Filter by price range
    if ($request->filled('harga_min')) {
        $query->where('harga_dasar', '>=', $request->harga_min);
    }
    if ($request->filled('harga_max')) {
        $query->where('harga_dasar', '<=', $request->harga_max);
    }

    $layanans = $query->orderBy('created_at', 'desc')->paginate(12)-
    >withQueryString();

    $stats = [
        'total_layanan' => Layanan::count(),
        'layanan_aktif' => Layanan::where('status', 'aktif')->count(),
        'layanan_nonaktif' => Layanan::where('status', 'nonaktif')->count(),
        'rata_rata_harga' => round(Layanan::avg('harga_dasar'), 0)
    ];

    return view('admin.layanan.index', compact('layanans', 'stats'));
}

public function create()
{
    return view('admin.layanan.create');
}

public function store(Request $request)
{
    try {
        $validationRules = [

```

```

        'nama_layanan' => 'required|string|max:100',
        'deskripsi_layanan' => 'required|string|min:10',
        'lokasi' => 'required|string',
        'jam_buka_operasional' => 'required|date_format:H:i',
        'jam_tutup_operasional' =>
'required|date_format:H:i|after:jam_buka_operasional',
        'harga_dasar' => 'required|numeric|min:0',
        'tipe_sewa' => 'required|in:per_hari,per_sesi',
        'area' => 'required|in:outdoor,indoor',
        'gambar_utama' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:20480',
        'galeri_foto.*' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:20480',
        'fasilitas_include' => 'required|array',
        'fasilitas_include.*' => 'required|string',
        'fasilitas_addon.*.nama' => 'nullable|string|max:100',
        'fasilitas_addon.*.harga' => 'nullable|numeric|min:0',
        'fasilitas_addon.*.deskripsi' => 'nullable|string'
    ];

    if ($request->tipe_sewa === 'per_sesi') {
        // $validationRules['durasi_sesi_menit'] = 'required|integer|in:60';
    }

    $request->validate($validationRules);
} catch (\Illuminate\Validation\ValidationException $e) {
    return response()->json([
        'success' => false,
        'message' => 'Validasi gagal: ' . collect($e->errors())->flatten()-
>implode(', '),
    ], 422);
}

DB::beginTransaction();
try {
    // Cek apakah storage public sudah ada dan writable
    if (!Storage::disk('public')->exists('layanan')) {
        try {
            Storage::disk('public')->makeDirectory('layanan/main');
            Storage::disk('public')->makeDirectory('layanan/gallery');
        } catch (\Exception $e) {
            throw new \Exception('Tidak bisa membuat direktori storage: ' . $e-
>getMessage());
        }
    }

    // Map form fields to database fields
    $data = [
        // Data dasar (untuk semua tipe sewa)
        'nama_layanan' => $request->nama_layanan,
        'deskripsi_layanan' => $request->deskripsi_layanan,
        'lokasi' => $request->lokasi,
        'jam_buka_operasional' => $request->jam_buka_operasional,
        'jam_tutup_operasional' => $request->jam_tutup_operasional,
        'harga_dasar' => $request->harga_dasar,

```

```

'tipe_sewa' => $request->tipe_sewa,
'area' => $request->area,

'durasi_sesi_menit' => $request->tipe_sewa === 'per_sesi' ? 120 : null,

// Data lainnya
'fasilitas_include' => is_array($request->fasilitas_include) ? implode(', ',
array_filter($request->fasilitas_include)) : $request->fasilitas_include,

'status' => $request->status ?? 'aktif',
'created_at' => now(),
'updated_at' => now()
];

if ($request->tipe_sewa === 'per_hari') {
    $data['durasi_sesi_menit'] = null;
}

if ($request->hasFile('gambar_utama')) {
    try {
        $file = $request->file('gambar_utama');

        if (!$file->isValid()) {
            throw new \Exception('File gambar utama tidak valid atau
corrupt');
        }

        $size = $file->getSize();
        if ($size > 20 * 1024 * 1024) {
            throw new \Exception('Ukuran gambar utama terlalu besar (maks
20MB)');
        } elseif ($size > 8 * 1024 * 1024) {
            Log::warning('Gambar utama berukuran besar: '.$file-
>getClientOriginalName().'. ('.number_format($size/1024/1024,2).' MB)');
        }

        $allowedMimes = ['image/jpeg', 'image/jpg', 'image/png', 'image/gif'];
        if (!in_array($file->getMimeType(), $allowedMimes)) {
            throw new \Exception('Format gambar tidak didukung. Gunakan
JPG, PNG, atau GIF');
        }

        $data['gambar_utama'] = $file->store('layanan/main', 'public');
    } catch (\Exception $e) {
        throw new \Exception('Error upload gambar utama: '. $e-
>getMessage());
    }
} else {
    // Gunakan gambar default sesuai database constraint
    $data['gambar_utama'] = 'images/default-service.jpg';
}

if ($request->hasFile('galeri_foto')) {

```

```

try {
    $galeri = [];
    $files = $request->file('galeri_foto');

    if (count($files) > 15) {
        throw new \Exception('Maksimal 15 foto untuk galeri');
    }

    foreach ($files as $index => $file) {
        if (!$file->isValid()) {
            throw new \Exception("Foto ke-{$index} tidak valid atau
corrupt");
        }

        $size = $file->getSize();
        if ($size > 20 * 1024 * 1024) {
            throw new \Exception("Foto ke-{$index} terlalu besar (maks
20MB)");
        } elseif ($size > 8 * 1024 * 1024) {
            Log::warning('Galeri foto ukuran besar: '.$file-
>getClientOriginalName().' ('.number_format($size/1024/1024,2).' MB)');
        }

        $allowedMimes = ['image/jpeg', 'image/jpg', 'image/png',
'image/gif'];
        if (!in_array($file->getMimeType(), $allowedMimes)) {
            throw new \Exception("Foto ke-{$index} format tidak didukung.
Gunakan JPG, PNG, atau GIF");
        }

        $galeri[] = $file->store('layanannya/gallery', 'public');
    }

    $data['galeri_gambar'] = json_encode($galeri);
} catch (\Exception $e) {
    throw new \Exception('Error upload galeri foto: ' . $e-
>getMessage());
}
} else {
    $data['galeri_gambar'] = json_encode([$data['gambar_utama']]);
}

$layanan = Layanan::create($data);

if ($request->filled('fasilitas_addon')) {
    $fasilitasData = [];
    foreach ($request->fasilitas_addon as $fasilitas) {
        if (!empty($fasilitas['nama']) && !empty($fasilitas['harga'])) {
            $fasilitasData[] = [
                'id_fasilitas_tambahan' => Str::uuid(),
                'id_layanan' => $layanan->id_layanan,
                'nama_fasilitas_tambahan' => $fasilitas['nama'],
                'harga_satuan' => $fasilitas['harga'],
            ];
        }
    }
}

```

```

        'deskripsi_fasilitas_tambahan' => $fasilitas['deskripsi'] ?? ",
        'created_at' => now(),
        'updated_at' => now()
    ];
    }
}

if (!empty($fasilitasData)) {
    DB::table('fasilitas_tambahan')->insert($fasilitasData);
}

DB::commit();

return response()->json([
    'success' => true,
    'message' => 'Layanan berhasil ditambahkan!',
    'data' => [
        'id' => $layanan->id_layanan,
        'nama' => $layanan->nama_layanan
    ]
]);
} catch (\Exception $e) {
    DB::rollback();
    Log::error('Layanan store error: ' . $e->getMessage());

    return response()->json([
        'success' => false,
        'message' => 'Terjadi kesalahan saat menyimpan data: ' . $e-
->getMessage()
    ], 500);
}

}

public function edit(Layanan $layanan)
{
    try {

        $layanan->load('fasilitasTambahan');

        // Format jam operasional untuk form
        $jam_buka = $layanan->jam_buka_operasional ?
        \Carbon\Carbon::parse($layanan->jam_buka_operasional)->format('H:i') : "";
        $jam_tutup = $layanan->jam_tutup_operasional ?
        \Carbon\Carbon::parse($layanan->jam_tutup_operasional)->format('H:i') : "";

        return response()->json([
            'success' => true,
            'layanan' => [
                'id_layanan' => $layanan->id_layanan,
                'nama_layanan' => $layanan->nama_layanan,
                'deskripsi_layanan' => $layanan->deskripsi_layanan,
                'lokasi' => $layanan->lokasi,
            ]
        ]);
    } catch (\Exception $e) {
        Log::error('Layanan edit error: ' . $e->getMessage());

        return response()->json([
            'success' => false,
            'message' => 'Terjadi kesalahan saat mengedit data: ' . $e->getMessage()
        ], 500);
    }
}

```

```

        'jam_buka_operasional' => $jam_buka,
        'jam_tutup_operasional' => $jam_tutup,
        'harga_dasar' => $layanan->harga_dasar,
        'tipe_sewa' => $layanan->tipe_sewa,
        'area' => $layanan->area,
        'durasi_sesi_menit' => $layanan->durasi_sesi_menit,
        'jeda_antar_sesi' => $layanan->jeda_antar_sesi,
        'max_sesi_per_hari' => $layanan->max_sesi_per_hari,
        'konfigurasi_sesi' => $layanan->konfigurasi_sesi,
        'status' => $layanan->status,
        'fasilitas_include' => $layanan->fasilitas_include,
        'gambar_utama' => $layanan->gambar_utama,
        'galeri_gambar' => $layanan->galeri_gambar,
        'fasilitas_tambahan' => $layanan->fasilitasTambahan
    ]
    );
} catch (\Exception $e) {
    Log::error('Edit layanan error: ' . $e->getMessage());
    Log::error('Stack trace: ' . $e->getTraceAsString());

    return response()->json([
        'success' => false,
        'message' => 'Gagal memuat data layanan: ' . $e->getMessage()
    ], 500);
}

}

public function update(Request $request, $id)
{
    try {
        // Log incoming request data for debugging
        Log::info('Layanan update request data:', [
            'id' => $id,
            'all_data' => $request->all(),
            'tipe_sewa' => $request->tipe_sewa,
            'tipe_sewa_old' => Layanan::find($id)->tipe_sewa ?? 'unknown'
        ]);

        $layanan = Layanan::findOrFail($id);

        if ($request->tipe_sewa !== $layanan->tipe_sewa) {
            Log::info('Tipe sewa berubah:', [
                'from' => $layanan->tipe_sewa,
                'to' => $request->tipe_sewa
            ]);

            // Cek apakah ada penyewaan aktif untuk layanan ini
            $activeBookings = $layanan->penyewaan()
                ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
                ->exists();

            if ($activeBookings) {

```

```

aktif, [
    Log::warning('Tipe sewa tidak bisa diubah karena ada penyewaan
    'layanan_id' => $id,
    'tipe_sewa_old' => $layanan->tipe_sewa,
    'tipe_sewa_new' => $request->tipe_sewa
]);

return response()->json([
    'success' => false,
    'message' => 'Tidak dapat mengubah tipe sewa karena layanan ini
    masih memiliki penyewaan aktif. Silakan batalkan atau selesaikan semua
    penyewaan terlebih dahulu.'
    ], 422);
}
}

$validationRules = [
    'nama_layanan' => 'required|string|max:100',
    'deskripsi_layanan' => 'required|string|min:10',
    'lokasi' => 'required|string',
    'jam_buka_operasional' => 'required|date_format:H:i',
    'jam_tutup_operasional' =>
    'required|date_format:H:i|after:jam_buka_operasional',
    'harga_dasar' => 'required|numeric|min:0',
    'tipe_sewa' => 'required|in:per_hari,per_sesi',
    'area' => 'required|in:outdoor,indoor',
    'gambar_utama' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:20480',
    'galeri_foto.*' => 'image|mimes:jpeg,png,jpg,gif|max:20480',
    'fasilitas_include' => 'required|array|min:1',
    'fasilitas_include.*' => 'string',
    'status' => 'required|in:aktif,nonaktif'
];

if ($request->tipe_sewa === 'per_sesi') {
    try {
        $request->validate($validationRules);
    } catch (\Illuminate\Validation\ValidationException $e) {
        Log::error('Validation failed in update:', [
            'errors' => $e->errors(),
            'request_data' => $request->all()
        ]);
        return response()->json([
            'success' => false,
            'message' => 'Validasi gagal: ' . collect($e->errors())->flatten()-
            >implode(', '),
            'errors' => $e->errors()
        ], 422);
    }

    if ($request->tipe_sewa === 'per_sesi') {
        try {

```

```

$jam_buka = Carbon::parse($request->jam_buka_operasional);
$jam_tutup = Carbon::parse($request->jam_tutup_operasional);
$durasi_sesi = 120; // TETAP 120 menit

$total_menit_operasional = $jam_buka->diffInMinutes($jam_tutup);
$max_sesi_yang_muat = floor($total_menit_operasional /
$durasi_sesi);
} catch (\Exception $e) {
    Log::error('Error parsing time in update:', [
        'jam_buka' => $request->jam_buka_operasional,
        'jam_tutup' => $request->jam_tutup_operasional,
        'error' => $e->getMessage()
    ]);

    return response()->json([
        'success' => false,
        'message' => 'Format waktu tidak valid. Pastikan format jam:menit
(contoh: 08:00)'
    ], 422);
}

DB::beginTransaction();
try {
    /
    $data = [
        'nama_layanan' => $request->nama_layanan,
        'deskripsi_layanan' => $request->deskripsi_layanan,
        'lokasi' => $request->lokasi,
        'jam_buka_operasional' => $request->jam_buka_operasional,
        'jam_tutup_operasional' => $request->jam_tutup_operasional,
        'harga_dasar' => $request->harga_dasar,
        'tipe_sewa' => $request->tipe_sewa,
        'area' => $request->area,

        'durasi_sesi_menit' => $request->tipe_sewa === 'per_sesi' ? 120 :
null,

        'fasilitas_include' => implode(',', $request->fasilitas_include),
        'status' => $request->status
    ];

    if ($request->tipe_sewa === 'per_hari') {
        $data['durasi_sesi_menit'] = null;
    }
    if ($request->hasFile('gambar_utama')) {
        if ($layanan->gambar_utama) {
            Storage::disk('public')->delete($layanan->gambar_utama);
        }
        $data['gambar_utama'] = $request->file('gambar_utama')-
>store('layanan/main', 'public');
    }
}

```

```

if ($request->hasFile('galeri_foto')) {
    if ($layanannya->galeri_gambar) {
        $oldGaleri = json_decode($layanannya->galeri_gambar, true);
        if (is_array($oldGaleri)) {
            foreach ($oldGaleri as $foto) {
                Storage::disk('public')->delete($foto);
            }
        }
    }
}

$galeri = [];
foreach ($request->file('galeri_foto') as $file) {
    $galeri[] = $file->store('layanannya/gallery', 'public');
}

$data['galeri_gambar'] = json_encode($galeri);
}

$layanannya->update($data);

$layanannya->fasilitasTambahannya->delete();
if ($request->filled('fasilitas_addon')) {
    foreach ($request->fasilitas_addon as $fasilitas) {
        if (!empty($fasilitas['nama']) && !empty($fasilitas['harga'])) {
            $layanannya->fasilitasTambahannya->create([
                'id_layanannya' => $layanannya->id_layanannya,
                'nama_fasilitas_tambahan' => $fasilitas['nama'],
                'harga_satuan' => $fasilitas['harga'],
                'deskripsi_fasilitas_tambahan' => $fasilitas['deskripsi'] ?? '',
                'status' => 'aktif'
            ]);
        }
    }
}

DB::commit();

return response()->json([
    'success' => true,
    'message' => 'Layanan berhasil diperbarui!'
]);
} catch (\Exception $e) {
    DB::rollback();
    Log::error('Update layanan error: ' . $e->getMessage());

    return response()->json([
        'success' => false,
        'message' => 'Terjadi kesalahan: ' . $e->getMessage()
    ], 500);
}
} catch (\Exception $e) {
    Log::error('Error in update method: ' . $e->getMessage());
    return response()->json([
        'success' => false,

```

```

        'message' => 'Terjadi kesalahan saat memperbarui layanan: ' . $e-
->getMessage()
    ], 500);
    }
}
public function toggleStatus($id)
{
    try {
        $layanan = Layanan::findOrFail($id);
        $layanan->status = $layanan->status === 'aktif' ? 'nonaktif' : 'aktif';
        $layanan->save();

        return response()->json([
            'success' => true,
            'message' => 'Status layanan berhasil diubah!',
            'status' => $layanan->status
        ]);
    } catch (\Exception $e) {
        return response()->json([
            'success' => false,
            'message' => 'Gagal mengubah status: ' . $e->getMessage()
        ], 500);
    }
}
public function show($id)
{
    try {

        $layanan = Layanan::with('fasilitasTambahan')->findOrFail($id);

        return response()->json([
            'success' => true,
            'layanan' => [
                'id_layanan' => $layanan->id_layanan,
                'nama_layanan' => $layanan->nama_layanan,
                'deskripsi_layanan' => $layanan->deskripsi_layanan,
                'lokasi' => $layanan->lokasi,
                'jam_buka_operasional' => $layanan->jam_buka_operasional ?
Carbon::parse($layanan->jam_buka_operasional)->format('H:i') : "",
                'jam_tutup_operasional' => $layanan->jam_tutup_operasional ?
Carbon::parse($layanan->jam_tutup_operasional)->format('H:i') : "",
                'harga_dasar' => $layanan->harga_dasar,
                'tipe_sewa' => $layanan->tipe_sewa,
                'area' => $layanan->area,
                'durasi_sesi_menit' => $layanan->durasi_sesi_menit,
                'jeda_antar_sesi' => $layanan->jeda_antar_sesi,
                'max_sesi_per_hari' => $layanan->max_sesi_per_hari,
                'konfigurasi_sesi' => $layanan->konfigurasi_sesi,
                'status' => $layanan->status,
                'fasilitas_include' => $layanan->fasilitas_include,
                'gambar_utama' => $layanan->gambar_utama,
                'galeri_gambar' => $layanan->galeri_gambar,
            ]
        ]);
    }
}

```

```

        'fasilitas_tambahan' => $layanan->fasilitasTambahan
    ]
    );
} catch (\Exception $e) {
    Log::error('Error in layanan show method', [
        'id' => $id,
        'error' => $e->getMessage(),
        'trace' => $e->getTraceAsString()
    ]);

    return response()->json([
        'success' => false,
        'message' => 'Gagal memuat detail layanan: ' . $e->getMessage()
    ], 500);
}

public function checkBookingStatus($id)
{
    try {
        $layanan = Layanan::findOrFail($id);

        // Cek penyewaan aktif
        $activeBookings = $layanan->penyewaan()
            ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
            ->get();

        $hasActiveBookings = $activeBookings->count() > 0;
        $activeBookingsCount = $activeBookings->count();

        return response()->json([
            'success' => true,
            'hasActiveBookings' => $hasActiveBookings,
            'activeBookingsCount' => $activeBookingsCount,
            'activeBookings' => $activeBookings->map(function($booking) {
                return [
                    'id' => $booking->id_penyewaan,
                    'kode_sewa' => $booking->kode_sewa,
                    'tanggal_penyewaan' => $booking->tanggal_penyewaan-
>format('d/m/Y'),
                    'status' => $booking->status_penyewaan,
                    'customer' => $booking->user->name ?? 'N/A'
                ];
            })
        ]);
    } catch (\Exception $e) {
        Log::error('Error checking booking status: ' . $e->getMessage());

        return response()->json([
            'success' => false,
            'message' => 'Gagal mengecek status penyewaan: ' . $e->getMessage()
        ], 500);
    }
}

```

```

    }

    public function destroy($id)
    {
        try {
            $layanannya = Layanan::findOrFail($id);

            if ($layanannya->gambar_utama && !Str::startsWith($layanannya->gambar_utama, 'images/')) {
                Storage::disk('public')->delete($layanannya->gambar_utama);
            }

            if ($layanannya->galeri_gambar) {
                $galeri = json_decode($layanannya->galeri_gambar, true);
                foreach ($galeri as $image) {
                    if (!Str::startsWith($image, 'images/')) {
                        Storage::disk('public')->delete($image);
                    }
                }
            }

            $layanannya->delete();

            return response()->json([
                'success' => true,
                'message' => 'Layanan berhasil dihapus!'
            ]);
        } catch (\Exception $e) {
            return response()->json([
                'success' => false,
                'message' => 'Tidak dapat menghapus layanan: ' . $e->getMessage()
            ], 500);
        }
    }

    public function customerShow(Request $request, $id)
    {
        $layanannya = Layanan::with('fasilitasTambahan')
            ->where('status', 'aktif')
            ->findOrFail($id);

        $unpaidBookings = Penyewaan::where('id_user', Auth::id())
            ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
            ->with(['pembayaran', 'layanannya'])
            ->get()
            ->filter(function($booking) {
                // Skip booking yang sudah dibatalkan
                if ($booking->status_penyewaan === 'cancelled') {
                    return false;
                }

                // Skip booking yang sudah expired (tanggal sudah lewat)
                if (Carbon::parse($booking->tanggal_penyewaan)->isPast()) {

```

```

        return false;
    }

    // Hitung total yang sudah dibayar (HANYA status 'lunas')
    $totalPaid = $booking->pembayaran()
        ->where('status_pembayaran', 'lunas')
        ->sum('jumlah_bayar');

    // Cek apakah ada pembayaran yang menunggu verifikasi admin
    $hasUploadedPayment = $booking->pembayaran()
        ->where('status_pembayaran', 'uploaded')
        ->exists();

    $hasRemainingPayment = $totalPaid < $booking->total_harga;

    if ($hasUploadedPayment) {
        return true;
    }

    return $hasRemainingPayment;
});

if ($unpaidBookings->count() > 0) {
    $unpaidList = $unpaidBookings->map(function($booking) {
        $totalPaid = $booking->pembayaran()
            ->where('status_pembayaran', 'lunas')
            ->sum('jumlah_bayar');
        $sisapembayaran = $booking->total_harga - $totalPaid;

        $statusText = "";
        if ($booking->pembayaran()->where('status_pembayaran', 'uploaded')-
            >exists()) {
            $statusText = ' (Menunggu Verifikasi Admin)';
        } elseif ($sisapembayaran > 0) {
            $statusText = ' - Sisa: Rp ' . number_format($sisapembayaran, 0, ',',
                '.');
        }
    });

    return $booking->layanan->nama_layanan . ' (' . $booking-
        >tanggal_penyewaan->format('d/m/Y') . ') ' . $statusText;
    }->join(', ');

    return redirect()->route('customer.penyewaan.unpaid-warning')
        ->with('error', "Anda masih memiliki booking yang belum lunas atau
        menunggu verifikasi: {$unpaidList}. Silakan lunasi atau tunggu verifikasi terlebih
        dahulu sebelum melakukan booking baru.");
}

$unavailableDates = [];

if ($layanan->tipe_sewa === 'per_hari') {

```

```

// Hanya untuk per_hari, blok tanggal yang sudah dibooking
$unavailableDates = Penyewaan::where('id_layanan', $id)
->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
->where('tanggal_penyewaan', '>=', now())
->where('tanggal_penyewaan', '<=', now()->addMonths(3))
->pluck('tanggal_penyewaan')
->map(function ($date) {
    return Carbon::parse($date)->format('Y-m-d');
})
->unique()
->values()
->toArray();
}

$holidayDates = $this->getHolidayDates($id);

$selectedDate = $request->get('date');
$timeSlots = [];

if ($selectedDate) {
    $timeSlots = $this->getAvailableTimeSlots($id, $selectedDate);
}

$relatedLayanan = Layanan::where('status', 'aktif')
->where('id_layanan', '!=', $id)
->whereBetween('harga_dasar', [
    $layanan->harga_dasar * 0.8,
    $layanan->harga_dasar * 1.2
])
->take(4)
->get();

return view('customer.layanan.detail', compact(
    'layanan',
    'unavailableDates',
    'holidayDates',
    'timeSlots',
    'relatedLayanan'
));
}

public function customerShowByName(Request $request, $nama)
{
    $layanan = Layanan::with('fasilitasTambahan')
->where('status', 'aktif')
->where('nama_layanan', 'like', '%'. $nama . '%')
->first();

    if (!$layanan) {
        abort(404, 'Layanan tidak ditemukan');
    }

    $id = $layanan->id_layanan;

```

```

        $unpaidBookings = Penyewaan::where('id_user', Auth::id())
        ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
        ->with(['pembayaran', 'layanan'])
        ->get()
        ->filter(function($booking) {
            // Skip booking yang sudah dibatalkan
            if ($booking->status_penyewaan === 'cancelled') {
                return false;
            }

            // Skip booking yang sudah expired (tanggal sudah lewat)
            if (Carbon::parse($booking->tanggal_penyewaan)->isPast()) {
                return false;
            }

            $totalPaid = $booking->pembayaran()
                ->where('status_pembayaran', 'lunas')
                ->sum('jumlah_bayar');

            $hasUploadedPayment = $booking->pembayaran()
                ->where('status_pembayaran', 'uploaded')
                ->exists();

            return ($totalPaid < $booking->total_harga) || $hasUploadedPayment;
        });

        if ($unpaidBookings->count() > 0) {
            $unpaidList = $unpaidBookings->map(function($booking) {
                $totalPaid = $booking->pembayaran()
                    ->where('status_pembayaran', 'lunas')
                    ->sum('jumlah_bayar');
                $sisapembayaran = $booking->total_harga - $totalPaid;

                $statusText = "";
                if ($booking->pembayaran()->where('status_pembayaran', 'uploaded')-
                    >exists()) {
                    $statusText = ' (Menunggu Verifikasi Admin)';
                } elseif ($sisapembayaran > 0) {
                    $statusText = ' - Sisa: Rp '. number_format($sisapembayaran, 0, ',',
                        '.');
                }

                return $booking->layanan->nama_layanan . ' (' . $booking-
                    >tanggal_penyewaan->format('d/m/Y') . ')'. $statusText;
            }->join(', ');

            return redirect()->route('customer.penyewaan.unpaid-warning')
                ->with('error', "Anda masih memiliki booking yang belum lunas atau
                menunggu verifikasi: {$unpaidList}. Silakan lunasi atau tunggu verifikasi terlebih
                dahulu sebelum melakukan booking baru.");
        }
    }

```

```

        $unavailableDates = [];

        if ($layanan->tipe_sewa === 'per_hari') {
            // Hanya untuk per_hari, blok tanggal yang sudah dibooking
            $unavailableDates = Penyewaan::where('id_layanan', $id)
                ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
                ->where('tanggal_penyewaan', '>=', now())
                ->pluck('tanggal_penyewaan')
                ->map(function ($date) {
                    return Carbon::parse($date)->format('Y-m-d');
                })
                ->unique()
                ->values()
                ->toArray();
        }

        $holidayDates = $this->getHolidayDates($id);

        $selectedDate = $request->get('date');
        $timeSlots = [];

        if ($selectedDate) {
            $timeSlots = $this->getAvailableTimeSlots($id, $selectedDate);
        }

        $relatedLayanan = Layanan::where('status', 'aktif')
            ->where('id_layanan', '!=', $id)
            ->whereBetween('harga_dasar', [
                $layanan->harga_dasar * 0.8,
                $layanan->harga_dasar * 1.2
            ])
            ->take(4)
            ->get();

        return view('customer.layanan.detail', compact(
            'layanan',
            'unavailableDates',
            'holidayDates',
            'timeSlots',
            'relatedLayanan'
        ));
    }

    /**
     * Get available time slots for specific date
     */
    private function getAvailableTimeSlots($layananId, $date)
    {
        try {
            $layanan = Layanan::find($layananId);
            if (!$layanan) {
                \Log::warning("Layanan not found: {$layananId}");
                return [];
            }
        }
    }

```

```

    }

    /**
     * API endpoint untuk mendapatkan available time slots
     */
    public function getAvailableSlotsApi(Request $request)
    {
        $request->validate([
            'id_layanan' => 'required|integer',
            'tanggal' => 'required|date'
        ]);

        $layananId = $request->id_layanan;
        $date = $request->tanggal;

        try {
            $slots = $this->getAvailableTimeSlots($layananId, $date);

            return response()->json([
                'success' => true,
                'slots' => $slots,
                'debug' => [
                    'total_slots' => count($slots),
                    'available_slots' => count(array_filter($slots, fn($s) =>
                        $s['available'])),
                    'booked_slots' => count(array_filter($slots, fn($s) => !$s['available']))
                ]
            ]);
        } catch (\Exception $e) {
            \Log::error("Error in getAvailableSlotsApi: " . $e->getMessage());

            return response()->json([
                'success' => false,
                'message' => 'Terjadi kesalahan: ' . $e->getMessage(),
                'slots' => []
            ], 500);
        }
    }

    public function getHolidayDates($layananId)
    {
        $sholidayDates = [];

        // Get specific dates (no time limit)
        $specificHolidays = JadwalLibur::where('id_layanan', $layananId)
            ->where('jenis', 'tanggal_tertentu')
            ->where('tanggal', '>=', now())
            ->get();

        foreach ($specificHolidays as $sholiday) {
            $sholidayDates[Carbon::parse($sholiday->tanggal)->format('Y-m-d')] = [

```

```

        'keterangan' => $holiday->keterangan,
        'jenis' => $holiday->jenis,
        'hari' => $holiday->hari
    ];
}

// Get weekly holidays (harian)
$weeklyHolidays = JadwalLibur::where('id_layanan', $layananId)
->where('jenis', 'harian')
->whereNotNull('hari')
->get();

$startDate = now();
$endDate = now()->addYears(2);

foreach ($weeklyHolidays as $holiday) {

    $currentDate = $startDate->copy();

    while ($currentDate <= $endDate) {
        $dayName = strtolower($currentDate->format('l')); // Monday, Tuesday,
etc.
        $indonesianDay = [
            'monday' => 'senin',
            'tuesday' => 'selasa',
            'wednesday' => 'rabu',
            'thursday' => 'kamis',
            'friday' => 'jumat',
            'saturday' => 'sabtu',
            'sunday' => 'minggu'
        ];
        if ($indonesianDay[$dayName] === $holiday->hari) {
            $dateKey = $currentDate->format('Y-m-d');
            if (!isset($holidayDates[$dateKey])) {
                $holidayDates[$dateKey] = [
                    'keterangan' => $holiday->keterangan,
                    'jenis' => $holiday->jenis,
                    'hari' => $holiday->hari
                ];
            }
        }
        $currentDate->addDay();
    }

    return $holidayDates;
}

public function getStats()
{
    return [
        'total_layanan' => Layanan::count(),

```

```

'layanat_aktif' => Layanat::where('statu', 'aktif')->count(),
'layanat_populer' => Layanat::withCount(['pembayaran' => function
($query) {
    $query->where('statu', 'verified');
}])->orderBy('pembayaran_count', 'desc')->take(5)->get(),
'rata_rata_harga' => round(Layanat::avg('harga_dasar'), 0)
    ];
}
}

```



2. bookingAvailability.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Layanan;
use App\Models\Penyewaan;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Carbon\Carbon;

class BookingAvailabilityController extends Controller
{
    public function getAvailableSlots(Request $request)
    {
        $request->validate([
            'id_layanan' => 'required|uuid|exists:layanan,id_layanan',
            'tanggal' => 'required|date|after_or_equal:today',
        ]);

        $layanan = Layanan::findOrFail($request->id_layanan);
        $tanggal = Carbon::parse($request->tanggal);

        $slots = $this->generateTimeSlots($layanan, $tanggal);

        return response()->json($slots);
    }

    private function generateTimeSlots($layanan, $tanggal)
    {
        if ($layanan->tipe_sewa !== 'per_sesi') {
            return [
                'tipe_sewa' => 'per_hari',
                'message' => 'Layanan ini menggunakan sistem per hari, bukan per sesi'
            ];
        }

        $jamBuka = Carbon::parse($layanan->jam_buka_operasional);
        $jamTutup = Carbon::parse($layanan->jam_tutup_operasional);
        $durasiSesi = $layanan->durasi_sesi_menit ?? 120;
        $jedaAntarSesi = $layanan->jeda_antar_sesi ?? 0;

        $slots = [];
        $currentTime = $jamBuka->copy();

        while ($currentTime < $jamTutup) {
            $jamMulai = $currentTime->format('H:i');
            $jamSelesai = $currentTime->copy()->addMinutes($durasiSesi)->format('H:i');

            if ($currentTime->copy()->addMinutes($durasiSesi) > $jamTutup) {
                break;
            }
        }
    }
}
```

```

$existingBooking = Penyewaan::where('id_layanan', $layanan->id_layanan)
->where('tanggal_penyewaan', $tanggal->format('Y-m-d'))
->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
->where('jam_mulai', '<', $jamSelesai)
->where('jam_selesai', '>', $jamMulai)
->first();

$slots[] = [
    'jam_mulai' => $jamMulai,
    'jam_selesai' => $jamSelesai,
    'available' => !$existingBooking,
    'booking_info' => $existingBooking ? [
        'kode_sewa' => $existingBooking->kode_sewa,
        'customer' => $existingBooking->user->getNamaLengkap(),
        'status' => $existingBooking->status_penyewaan
    ] : null
];

// Pindah ke slot berikutnya
$currentTime->addMinutes($durasiSesi + $jedaAntarSesi);
}

return [
    'tipe_sewa' => 'per_sesi',
    'tanggal' => $tanggal->format('Y-m-d'),
    'durasi_sesi' => $durasiSesi,
    'jeda_antar_sesi' => $jedaAntarSesi,
    'jam_operasional' => [
        'buka' => $layanan->jam_buka_operasional,
        'tutup' => $layanan->jam_tutup_operasional
    ],
    'slots' => $slots,
    'total_slots' => count($slots),
    'available_slots' => count(array_filter($slots, fn($slot) => $slot['available']))
];
}

public function checkPerHariAvailability($layanan, $tanggal)
{
    $existingBooking = Penyewaan::where('id_layanan', $layanan->id_layanan)
->where('tanggal_penyewaan', $tanggal->format('Y-m-d'))
->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
->first();

    $areaConflict = $this->checkAreaConflict($layanan, $tanggal);

    $isAvailable = !$existingBooking && !$areaConflict['has_conflict'];

    return [
        'available' => $isAvailable,
        'message' => $isAvailable ? 'Tanggal tersedia untuk booking' :
        ($areaConflict['has_conflict'] ? $areaConflict['message'] : 'Tanggal sudah dibooking'),
    ];
}

```

```

        'existing_booking' => $existingBooking ? [
            'kode_sewa' => $existingBooking->kode_sewa,
            'customer' => $existingBooking->user->getNamaLengkap(),
            'status' => $existingBooking->status_penyewaan
        ] : null,
        'area_conflict' => $areaConflict,
        'tipe_sewa' => 'per_hari',
        'tanggal' => $tanggal->format('Y-m-d')
    ];
}

private function checkAreaConflict($layanannya, $tanggal)
{
    // Cek apakah ada booking di area yang sama pada tanggal yang sama
    $conflictingBooking = Penyewaan::join('layanannya', 'penyewaan.id_layanannya', '=',
'layanannya.id_layanannya')
        ->where('layanannya.area', $layanannya->area)
        ->where('penyewaan.tanggal_penyewaan', $tanggal->format('Y-m-d'))
        ->where('penyewaan.id_layanannya', '!=', $layanannya->id_layanannya)
        ->whereIn('penyewaan.status_penyewaan', ['pending', 'reserved', 'confirmed'])
        ->with('layanannya')
        ->first();

    if ($conflictingBooking) {
        return [
            'has_conflict' => true,
            'message' => "Tanggal ini sudah diboooking oleh layanan '{$conflictingBooking->layanannya->nama_layanannya}' di area yang sama",
            'conflicting_layanannya' => $conflictingBooking->layanannya->nama_layanannya,
            'conflicting_area' => $conflictingBooking->layanannya->area
        ];
    }

    return [
        'has_conflict' => false,
        'message' => null
    ];
}

public function getMonthlyAvailability(Request $request)
{
    $request->validate([
        'id_layanannya' => 'required|uuid|exists:layanannya,id_layanannya',
        'bulan' => 'required|integer|between:1,12',
        'tahun' => 'required|integer|min:2024'
    ]);

    $layanannya = Layanannya::findOrFail($request->id_layanannya);
    $bulan = $request->bulan;
    $tahun = $request->tahun;

    $startDate = Carbon::create($tahun, $bulan, 1);
    $endDate = $startDate->copy()->endOfMonth();

```

```

$availability = [];

for ($date = $startDate->copy(); $date->lte($endDate); $date->addDay()) {
    $dateStr = $date->format('Y-m-d');

    // Cek apakah tanggal sudah lewat
    if ($date->isPast()) {
        $availability[$dateStr] = [
            'color' => 'gray-soft',
            'tooltip' => 'Tanggal sudah lewat',
            'available' => false
        ];
        continue;
    }

    $jadwalLibur = JadwalLibur::where('id_layanan', $layanan->id_layanan)
        ->where('tanggal', $dateStr)
        ->where('status', 'libur')
        ->first();

    if ($jadwalLibur) {
        $availability[$dateStr] = [
            'color' => 'blue',
            'tooltip' => 'Libur: ' . $jadwalLibur->keterangan,
            'available' => false
        ];
        continue;
    }

    if ($layanan->tipe_sewa === 'per_hari') {
        $existingBooking = Penyewaan::where('id_layanan', $layanan->id_layanan)
            ->where('tanggal_penyewaan', $dateStr)
            ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
            ->first();

        if ($existingBooking) {
            $availability[$dateStr] = [
                'color' => 'red',
                'tooltip' => 'Sudah diboooking oleh customer lain',
                'available' => false
            ];
            continue;
        }
    }

    $areaConflict = $this->checkAreaConflict($layanan, $date);
    if ($areaConflict['has_conflict']) {
        $availability[$dateStr] = [
            'color' => 'red',
            'tooltip' => $areaConflict['message'],
            'available' => false,
            'area_conflict' => true,

```

```

        'conflicting_layanan' => $areaConflict['conflicting_layanan']
    ];
    continue;
}

if ($layanan->tipe_sewa === 'per_hari') {
    $availability[$dateStr] = [
        'color' => 'green-light',
        'tooltip' => 'Tersedia untuk booking (1 hari penuh)',
        'available' => true,
        'display_text' => '1 hari'
    ];
} else {
    // Untuk per_sesi, cek slot availability
    $slots = $this->generateTimeSlots($layanan, $date);
    $availableSlots = count(array_filter($slots['slots'], fn($slot) => $slot['available']));
    $totalSlots = count($slots['slots']);

    if ($availableSlots === 0) {
        $availability[$dateStr] = [
            'color' => 'red',
            'tooltip' => 'Semua sesi sudah dibooking',
            'available' => false
        ];
    } else if ($availableSlots < $totalSlots) {
        $availability[$dateStr] = [
            'color' => 'yellow-soft',
            'tooltip' => "Tersedia {$availableSlots} dari {$totalSlots} sesi",
            'available' => true,
            'display_text' => "{$availableSlots}/{$totalSlots} sesi",
            'total_sessions' => $totalSlots,
            'available_sessions' => $availableSlots
        ];
    } else {
        $availability[$dateStr] = [
            'color' => 'green-light',
            'tooltip' => "Tersedia {$totalSlots} sesi",
            'available' => true,
            'display_text' => "{$totalSlots} sesi",
            'total_sessions' => $totalSlots,
            'available_sessions' => $availableSlots
        ];
    }
}
}

return response()->json([
    'success' => true,
    'availability' => $availability,
    'layanan' => [
        'nama' => $layanan->nama_layanan,
        'tipe_sewa' => $layanan->tipe_sewa,
        'area' => $layanan->area
    ]
]);

```

```

    ]
  });
}
}

```

3. Pembayaran.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Layanan;
use App\Models\Pembayaran;
use App\Models\Penyewaan;
use Illuminate\Http\Request;
use App\Models\RekeningKoperasi;
use Illuminate\Support\Facades\Auth;
use App\Services\NotificationService;
use App\Events\PembayaranStatusUpdated;
use Illuminate\Support\Facades\Log;
use App\Http\Controllers\NotificationController;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;

class PembayaranController extends Controller
{
    public function index(Request $request)
    {
        $query = Pembayaran::with(['penyewaan.user', 'penyewaan.layanan']);

        if ($request->filled('search')) {
            $search = $request->search;
            $query->where(function($q) use ($search) {
                $q->where('kode_pembayaran', 'like', "%{$search}%")
                ->orWhereHas('penyewaan.user', function($userQuery) use ($search) {
                    $userQuery->where('nama_lengkap', 'like', "%{$search}%")
                    ->orWhere('email', 'like', "%{$search}%");
                });
            });
        }

        if ($request->filled('status')) {
            $query->where('status_pembayaran', $request->status);
        }

        if ($request->filled('layanan_id')) {
            $query->whereHas('penyewaan', function($q) use ($request) {
                $q->where('id_layanan', $request->layanan_id);
            });
        }

        $pembayarans = $query->latest()->paginate(15);
    }
}

```

```

$layanans = Layanan::where('status', 'aktif')->orderBy('nama_layanan', 'asc')->get();

$pendingVerificationCount = Pembayaran::whereIn('status_pembayaran', ['uploaded',
'verifikasi'])->count();

$filterInfo = [];
if ($request->filled('layanan_id')) {
    $layanan = Layanan::find($request->layanan_id);
    $filterInfo['layanan'] = $layanan ? $layanan->nama_layanan : 'Unknown';
}
if ($request->filled('status')) {
    $statusLabels = [
        'pending' => 'Menunggu Pembayaran',
        'uploaded' => 'Menunggu Verifikasi',
        'verifikasi' => 'Sedang Diverifikasi',
        'lunas' => 'Lunas',
        'ditolak' => 'Ditolak',
        'expired' => 'Kedaluwarsa'
    ];
    $filterInfo['status'] = $statusLabels[$request->status] ?? $request->status;
}
if ($request->filled('search')) {
    $filterInfo['search'] = $request->search;
}

return view('admin.pembayaran.index', compact('pembayarans', 'layanans', 'filterInfo',
'pendingVerificationCount'));
}

public function show($id)
{
    $pembayaran = Pembayaran::with(['penyewaan.user', 'penyewaan.layanan'])->findOrFail($id);

    return view('admin.pembayaran.show', compact('pembayaran'));
}

/**
 * Admin: Stream/preview bukti transfer tanpa bergantung pada symlink publik
 */
public function showBukti($id)
{
    $pembayaran = Pembayaran::findOrFail($id);

    if (!$pembayaran->bukti_transfer) {
        abort(404);
    }

    $relativePath = 'bukti_transfer/' . $pembayaran->bukti_transfer;
    $publicDisk = Storage::disk('public');

    if ($publicDisk->exists($relativePath)) {
        return $publicDisk->response($relativePath, null, [

```

```

        'Cache-Control' => 'private, max-age=31536000',
        'Content-Disposition' => 'inline',
    );
}

$legacyPath = storage_path('app/private/public/bukti_transfer/' . $pembayaran->bukti_transfer);
if (file_exists($legacyPath)) {
    return response()->file($legacyPath, [
        'Content-Type' => mime_content_type($legacyPath) ?: 'application/octet-stream',
        'Content-Disposition' => 'inline; filename="' . basename($legacyPath) . '"',
    ]);
}

// 3) Fallback lain: jika ada symlink dan file tersalin ke public/storage
$publicSymlinkPath = public_path('storage/bukti_transfer/' . $pembayaran->bukti_transfer);
if (file_exists($publicSymlinkPath)) {
    return response()->file($publicSymlinkPath, [
        'Content-Type' => mime_content_type($publicSymlinkPath) ?: 'application/octet-stream',
        'Content-Disposition' => 'inline; filename="' . basename($publicSymlinkPath) . '"',
        'Cache-Control' => 'private, max-age=31536000',
    ]);
}

Log::warning('Bukti transfer tidak ditemukan', [
    'pembayaran_id' => $pembayaran->id_pembayaran,
    'file' => $pembayaran->bukti_transfer,
]);

abort(404);
}

public function verify(Request $request, Pembayaran $pembayaran)
{
    $request->validate([
        'verification_note' => 'nullable|string|max:1000'
    ]);

    $pembayaran->update([
        'status_pembayaran' => 'lunas',
        'catatan_admin' => $request->verification_note,
        'tanggal_verifikasi' => now(),
        'verified_by' => Auth::id()
    ]);

    if ($pembayaran->jenis_pembayaran === 'full_payment') {
        // Jika bayar penuh, langsung confirmed
        $pembayaran->penyewaan->update([
            'status_penyewaan' => 'confirmed'
        ]);
    }
}

```

```

NotificationService::sendPaymentSuccess($pembayaran);
} elseif ($pembayaran->jenis_pembayaran === 'pelunasan') {
    $totalPaid = $pembayaran->penyewaan->pembayaran()
        ->where('status_pembayaran', 'lunas')
        ->sum('jumlah_bayar');

    if ($totalPaid >= $pembayaran->penyewaan->total_harga) {
        $pembayaran->penyewaan->update([
            'status_penyewaan' => 'confirmed'
        ]);
        NotificationService::sendPaymentSuccess($pembayaran);
    }
} else {
    NotificationService::sendDpAccepted($pembayaran);
    NotificationService::sendFinalPaymentReminder($pembayaran->penyewaan);
}
event(new PembayaranStatusUpdated($pembayaran));

return redirect()
    ->route('admin.pembayaran.show', $pembayaran->id_pembayaran)
    ->with('success', 'Pembayaran berhasil diverifikasi!');
}

public function reject(Request $request, Pembayaran $pembayaran)
{
    $request->validate([
        'instruksi_customer' => 'required|in:upload_ulang,refund_only',
        'catatan_admin' => 'required|string|max:1000'
    ]);

    $pembayaran->update([
        'status_pembayaran' => 'ditolak',
        'catatan_admin' => $request->catatan_admin
    ]);

    if ($request->instruksi_customer === 'upload_ulang') {
        $pembayaran->penyewaan->update([
            'status_penyewaan' => 'reserved'
        ]);

        $pembayaran->update([
            'batas_waktu_transfer' => now()->addHours(24)
        ]);

        NotificationController::sendPaymentNotification($pembayaran,
            'rejected_upload_ulang');
    } else {
        $pembayaran->penyewaan->update([
            'status_penyewaan' => 'cancelled',
            'cancelled_by' => Auth::id(), // Set admin sebagai yang membatalkan
            'cancelled_at' => now(),

```

```

        'alasan_pembatalan' => 'Pembayaran ditolak oleh admin - refund'
    ]);

    // Kirim notifikasi refund
    NotificationController::sendPaymentNotification($pembayaran, 'rejected_refund');
}

event(new PembayaranStatusUpdated($pembayaran));

return redirect()
    ->route('admin.pembayaran.show', $pembayaran->id_pembayaran)
    ->with('success', 'Pembayaran berhasil ditolak!');
}
/**
 * Customer: Halaman pelunasan untuk DP yang sudah dibayar
 */
public function pelunasan($id_penyewaan)
{
    $penyewaan = Penyewaan::with(['layanan', 'pembayaran'])->findOrFail($id_penyewaan);

    if ($penyewaan->id_user !== Auth::id()) {
        abort(403, 'Anda tidak memiliki akses ke halaman ini');
    }

    $totalPaid = $penyewaan->pembayaran()->where('status_pembayaran', 'lunas')->sum('jumlah_bayar');
    $sisapembayaran = $penyewaan->total_harga - $totalPaid;

    if ($sisapembayaran <= 0) {
        return redirect()->route('customer.penyewaan.detail', $penyewaan->id_penyewaan)
            ->with('info', 'Pembayaran sudah lunas penuh');
    }

    $existingPelunasan = $penyewaan->pembayaran()
        ->where('jenis_pembayaran', 'pelunasan')
        ->whereIn('status_pembayaran', ['pending', 'uploaded'])
        ->first();

    if ($existingPelunasan) {
        return redirect()->route('customer.pembayaran.show', $existingPelunasan->id_pembayaran)
            ->with('info', 'Anda sudah memiliki pembayaran pelunasan yang sedang diproses');
    }

    $defaultRekening = \App\Models\RekeningKoperasi::where('is_default', true)->first();
    if (!$defaultRekening) {
        $defaultRekening = \App\Models\RekeningKoperasi::where('is_default', true)->first();
    }

    $pembayaran = Pembayaran::create([
        'id_pembayaran' => Str::uuid(),
        'id_penyewaan' => $penyewaan->id_penyewaan,

```

```

        'id_rekening' => $defaultRekening->id_rekening,
        'kode_pembayaran' => 'LUNAS-' . date('Ymd') . '-' . str_pad(rand(1, 9999), 4, '0',
STR_PAD_LEFT),
        'jenis_pembayaran' => 'pelunasan',
        'jumlah_bayar' => $sisapembayaran,
        'status_pembayaran' => 'pending',
        'batas_waktu_transfer' => now()->addDays(3), // 3 hari untuk pelunasan
    ]);

    return redirect()->route('customer.pembayaran.show', $pembayaran->id_pembayaran)
        ->with('success', 'Silakan lakukan pelunasan sebesar Rp ' .
number_format($sisapembayaran, 0, ',', '.'));
    }

    public function customerShow($id)
    {
        $pembayaran = Pembayaran::with(['penyewaan.user', 'penyewaan.layanan',
'penyewaan.layanan.fasilitas Tambahan'])
        ->where('id_pembayaran', $id)
        ->whereHas('penyewaan', function($q) {
            $q->where('id_user', Auth::id());
        })
        ->firstOrFail();

        $rekeningList = RekeningKoperasi::all();

        return view('customer.pembayaran.show', compact('pembayaran', 'rekeningList'));
    }

    public function customerUpload(Request $request, $id)
    {
        $request->validate([
            'bukti_transfer' => 'required|image|mimes:jpeg,png,jpg|max:2048',
            'nama_pengirim' => 'required|string|max:100',
            'bank_pengirim' => 'required|string|max:50',
            'bank_lainnya' => 'nullable|string|max:50',
            'tanggal_transfer' => 'required|date',
            'catatan_customer' => 'nullable|string|max:500'
        ]);

        try {
            $pembayaran = Pembayaran::where('id_pembayaran', $id)
            ->whereHas('penyewaan', function($q) {
                $q->where('id_user', Auth::id());
            })
            ->firstOrFail();
        } catch (\Exception $e) {
            Log::error('Pembayaran not found or access denied', [
                'id' => $id,
                'user_id' => Auth::id(),
                'error' => $e->getMessage()
            ]);
        }
    }

```

```

        return redirect()->back()->withErrors(['error' => 'Pembayaran tidak ditemukan atau akses ditolak']);
    }

    $bankPengirim = $request->bank_pengirim;
    if ($request->bank_pengirim === 'lainnya' && $request->bank_lainnya) {
        $bankPengirim = $request->bank_lainnya;
    }

    try {
        $file = $request->file('bukti_transfer');
        $fileName = 'bukti_' . time() . '.' . $file->getClientOriginalExtension();
        $file->storeAs('public/bukti_transfer', $fileName);
    } catch (\Exception $e) {
        Log::error('File upload failed', [
            'error' => $e->getMessage()
        ]);
        return redirect()->back()->withErrors(['error' => 'Gagal mengupload file. Silakan coba lagi.']);
    }

    try {
        $pembayaran->update([
            'bukti_transfer' => $fileName,
            'nama_pengirim' => $request->nama_pengirim,
            'bank_pengirim' => $bankPengirim,
            'tanggal_transfer' => $request->tanggal_transfer,
            'catatan_customer' => $request->catatan_customer,
            'status_pembayaran' => 'uploaded'
        ]);
    } catch (\Exception $e) {
        Log::error('Failed to update pembayaran', [
            'pembayaran_id' => $pembayaran->id_pembayaran,
            'error' => $e->getMessage()
        ]);
        return redirect()->back()->withErrors(['error' => 'Gagal menyimpan data pembayaran. Silakan coba lagi.']);
    }

    try {
        NotificationController::sendPaymentNotification($pembayaran, 'uploaded');
    } catch (\Exception $e) {
        Log::error('Failed to send payment upload notification', [
            'pembayaran_id' => $pembayaran->id_pembayaran,
            'error' => $e->getMessage()
        ]);
    }

    return redirect()->route('customer.sewa.history')
        ->with('success', 'Bukti pembayaran berhasil diupload! Silakan cek riwayat sewa untuk melihat status verifikasi.');
```

4. Layanan

```
<?php

namespace App\Http\Controllers;

use Carbon\Carbon;
use App\Models\Layanan;
use App\Models\Penyewaan;
use App\Models\Pembayaran;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Str;
use App\Models\JadwalLibur;

class LayananController extends Controller
{
    /**
     * Helper method untuk mendapatkan active bookings (bukan cancelled atau completed)
     */
    private function getActiveBookings($userId)
    {
        return Penyewaan::where('id_customer', $userId)
            ->whereIn('status_penyewaan', ['pending', 'reserved'])
            ->with(['pembayaran', 'layanan'])
            ->get()
            ->filter(function($booking) {
                // Skip booking yang sudah dibatalkan karena refund
                if ($booking->status_penyewaan === 'cancelled') {
                    return false;
                }
                return true;
            });
    }

    /**
     * Helper method untuk mendapatkan unpaid bookings
     */
    private function getUnpaidBookings($userId)
    {
        return $this->getActiveBookings($userId)->filter(function($booking) {
            $totalPaid = $booking->pembayaran()
                ->where('status_pembayaran', 'lunas')
                ->sum('jumlah_bayar');

            return $totalPaid < $booking->total_harga;
        });
    }
}
```

method - redirect to appropriate index based on role

```

*/
public function index(Request $request)
{
    if (Auth::user()->role === 'admin') {
        return $this->adminIndex($request);
    } else {
        return redirect()->route('customer.dashboard');
    }
}

public function adminIndex(Request $request)
{
    $query = Layanan::query();

    // Search functionality
    if ($request->filled('search')) {
        $search = $request->search;
        $query->where(function ($q) use ($search) {
            $q->where('nama_layanan', 'like', '%' . $search . '%')
                ->orWhere('lokasi', 'like', '%' . $search . '%')
                ->orWhere('deskripsi_layanan', 'like', '%' . $search . '%');
        });
    }

    if ($request->filled('status') && $request->status !== "") {
        $query->where('status', $request->status);
    }

    if ($request->filled('harga_min')) {
        $query->where('harga_dasar', '>=', $request->harga_min);
    }
    if ($request->filled('harga_max')) {
        $query->where('harga_dasar', '<=', $request->harga_max);
    }

    $layanans = $query->orderBy('created_at', 'desc')->paginate(12)->withQueryString();

    // Debug: Log hasil query
    ]);

    $stats = [
        'total_layanan' => Layanan::count(),
        'layanan_aktif' => Layanan::where('status', 'aktif')->count(),
        'layanan_nonaktif' => Layanan::where('status', 'nonaktif')->count(),
        'rata_rata_harga' => round(Layanan::avg('harga_dasar'), 0)
    ];

    return view('admin.layanan.index', compact('layanans', 'stats'));
}

public function create()

```

```

{
    return view('admin.layanan.create');
}

public function store(Request $request)
{
    try {

        $validationRules = [
            'nama_layanan' => 'required|string|max:100',
            'deskripsi_layanan' => 'required|string|min:10',
            'lokasi' => 'required|string',
            'jam_buka_operasional' => 'required|date_format:H:i',
            'jam_tutup_operasional' =>
            'required|date_format:H:i|after:jam_buka_operasional',
            'harga_dasar' => 'required|numeric|min:0',
            'tipe_sewa' => 'required|in:per_hari,per_sesi',
            'area' => 'required|in:outdoor,indoor',
            'gambar_utama' =>
            'nullable|image|mimes:jpeg,png,jpg,gif|max:5120|dimensions:max_width=2000,max_height=2000',
            'galeri_foto.*' =>
            'nullable|image|mimes:jpeg,png,jpg,gif|max:5120|dimensions:max_width=2000,max_height=2000',
            'fasilitas_include' => 'nullable|array',
            'fasilitas_include.*' => 'nullable|string',
            'fasilitas_addon.*.nama' => 'nullable|string|max:100',
            'fasilitas_addon.*.harga' => 'nullable|numeric|min:0',
            'fasilitas_addon.*.deskripsi' => 'nullable|string'
        ];

        // VALIDASI KHUSUS UNTUK PER SESI SAJA
        if ($request->tipe_sewa === 'per_sesi') {

            $request->validate($validationRules);
        } catch (\Illuminate\Validation\ValidationException $e) {
            return response()->json([
                'success' => false,
                'message' => 'Validasi gagal: ' . collect($e->errors())->flatten()->implode(', '),
            ], 422);
        }

        DB::beginTransaction();
        try {
            // Cek apakah storage public sudah ada dan writable
            if (!Storage::disk('public')->exists('layanan')) {
                try {
                    Storage::disk('public')->makeDirectory('layanan/main');
                    Storage::disk('public')->makeDirectory('layanan/gallery');
                } catch (\Exception $e) {
                    throw new \Exception('Tidak bisa membuat direktori storage: ' . $e->getMessage());
                }
            }
        }
    }
}

```

```

    }
}

// Map form fields to database fields
$data = [
    // Data dasar (untuk semua tipe sewa)
    'nama_layanan' => $request->nama_layanan,
    'deskripsi_layanan' => $request->deskripsi_layanan,
    'lokasi' => $request->lokasi,
    'jam_buka_operasional' => $request->jam_buka_operasional,
    'jam_tutup_operasional' => $request->jam_tutup_operasional,
    'harga_dasar' => $request->harga_dasar,
    'tipe_sewa' => $request->tipe_sewa,
    'area' => $request->area,

    // Data sesi (hanya untuk per_sesi, null untuk per_hari)
    // Durasi dari config default
    'durasi_sesi_menit' => $request->tipe_sewa === 'per_sesi' ?
config('layanan.durasi_sesi_default') : null,

    // Data lainnya
    'fasilitas_include' => is_array($request->fasilitas_include) ? implode(', ',
array_filter($request->fasilitas_include)) : ($request->fasilitas_include ?? ''),

    'status' => $request->status ?? 'aktif',
    'created_at' => now(),
    'updated_at' => now()
];

// Pastikan data sesi null untuk per_hari
if ($request->tipe_sewa === 'per_hari') {
    $data['durasi_sesi_menit'] = null;
}

// Handle main image upload - OPTIMIZED dengan error handling
if ($request->hasFile('gambar_utama')) {
    try {
        $file = $request->file('gambar_utama');

        // Validasi file exists dan valid
        if (!$file->isValid()) {
            throw new \Exception('File gambar utama tidak valid atau corrupt');
        }

        // Ukuran file: warning > 8MB, hard limit 20MB
        $size = $file->getSize();
        if ($size > 20 * 1024 * 1024) {
            throw new \Exception('Ukuran gambar utama terlalu besar (maks 20MB)');
        } elseif ($size > 8 * 1024 * 1024) {
            Log::warning('Gambar utama berukuran besar: ' . $file->
getClientOriginalName() . ' (' . number_format($size/1024/1024, 2) . ' MB)');
        }
    }
}

```

```

// Validasi format file
$allowedMimes = ['image/jpeg', 'image/jpg', 'image/png', 'image/gif'];
if (!in_array($file->getMimeType(), $allowedMimes)) {
    throw new \Exception('Format gambar tidak didukung. Gunakan JPG, PNG,
atau GIF');
}

$data['gambar_utama'] = $file->store('layanannya/main', 'public');
} catch (\Exception $e) {
    throw new \Exception('Error upload gambar utama: ' . $e->getMessage());
}
} else {
    // Gunakan gambar default sesuai database constraint
    $data['gambar_utama'] = 'images/default-service.jpg';
}

// Handle gallery images upload - OPTIMIZED dengan error handling
if ($request->hasFile('galeri_foto')) {
    try {
        $galeri = [];
        $files = $request->file('galeri_foto');

        // Validasi jumlah file (max 15 foto)
        if (count($files) > 15) {
            throw new \Exception('Maksimal 15 foto untuk galeri');
        }

        // Batch upload untuk mempercepat
        foreach ($files as $index => $file) {
            // Validasi file exists dan valid
            if (!$file->isValid()) {
                throw new \Exception("Foto ke-{$index} tidak valid atau corrupt");
            }

            // Ukuran file: warning > 8MB, hard limit 20MB
            $size = $file->getSize();
            if ($size > 20 * 1024 * 1024) {
                throw new \Exception("Foto ke-{$index} terlalu besar (maks 20MB)");
            } elseif ($size > 8 * 1024 * 1024) {
                Log::warning('Galeri foto ukuran besar: ' . $file->getClientOriginalName() . ' (' . number_format($size/1024/1024, 2) . ' MB)');
            }

            // Validasi format file
            $allowedMimes = ['image/jpeg', 'image/jpg', 'image/png', 'image/gif'];
            if (!in_array($file->getMimeType(), $allowedMimes)) {
                throw new \Exception("Foto ke-{$index} format tidak didukung.
Gunakan JPG, PNG, atau GIF");
            }

            $galeri[] = $file->store('layanannya/gallery', 'public');
        }
    }
}

```

```

        $data['galeri_gambar'] = json_encode($galeri);
    } catch (\Exception $e) {
        throw new \Exception('Error upload galeri foto: ' . $e->getMessage());
    }
} else {
    // Gunakan gambar utama sebagai galeri default
    $data['galeri_gambar'] = json_encode([$data['gambar_utama']]);
}

// Create layanan dengan data lengkap
$layanan = Layanan::create($data);

// Handle fasilitas tambahan - OPTIMIZED dengan batch insert
if ($request->filled('fasilitas_addon')) {
    $fasilitasData = [];
    foreach ($request->fasilitas_addon as $fasilitas) {
        if (!empty($fasilitas['nama']) && !empty($fasilitas['harga'])) {
            $fasilitasData[] = [
                'id_fasilitas_tambahan' => Str::uuid(),
                'id_layanan' => $layanan->id_layanan,
                'nama_fasilitas_tambahan' => $fasilitas['nama'],
                'harga_satuan' => $fasilitas['harga'],
                'deskripsi_fasilitas_tambahan' => $fasilitas['deskripsi'] ?? '',
                'created_at' => now(),
                'updated_at' => now()
            ];
        }
    }

    // Batch insert untuk mempercepat - hanya jika ada data
    if (!empty($fasilitasData)) {
        DB::table('fasilitas_tambahan')->insert($fasilitasData);
    }

    DB::commit();

    // Response cepat tanpa delay
    return response()->json([
        'success' => true,
        'message' => 'Layanan berhasil ditambahkan!',
        'data' => [
            'id' => $layanan->id_layanan,
            'nama' => $layanan->nama_layanan
        ]
    ]);
} catch (\Exception $e) {
    DB::rollback();
    Log::error('Layanan store error: ' . $e->getMessage());

    return response()->json([
        'success' => false,
        'message' => 'Terjadi kesalahan saat menyimpan data: ' . $e->getMessage()
    ]);
}

```

```

    ], 500);
  }
}

/**
 * Admin: Edit layanan (return JSON for modal)
 */
public function edit(Layanan $layanan)
{
    try {
        // Check if this is an AJAX request
        if (!request()->ajax() && !request()->wantsJson()) {
            // If not AJAX, redirect to admin layanan index
            return redirect()->route('admin.layanan.index');
        }

        $layanan->load('fasilitasTambahkan');

        // Format jam operasional untuk form
        $jam_buka = $layanan->jam_buka_operasional ? \Carbon\Carbon::parse($layanan->jam_buka_operasional)->format('H:i') : "";
        $jam_tutup = $layanan->jam_tutup_operasional ? \Carbon\Carbon::parse($layanan->jam_tutup_operasional)->format('H:i') : "";

        return response()->json([
            'success' => true,
            'layanan' => [
                'id_layanan' => $layanan->id_layanan,
                'nama_layanan' => $layanan->nama_layanan,
                'deskripsi_layanan' => $layanan->deskripsi_layanan,
                'lokasi' => $layanan->lokasi,
                'jam_buka_operasional' => $jam_buka,
                'jam_tutup_operasional' => $jam_tutup,
                'harga_dasar' => $layanan->harga_dasar,
                'tipe_sewa' => $layanan->tipe_sewa,
                'area' => $layanan->area,
                'durasi_sesi_menit' => $layanan->durasi_sesi_menit,
                'jeda antar sesi' => $layanan->jeda antar sesi,
                'max_sesi_per_hari' => $layanan->max_sesi_per_hari,
                'konfigurasi_sesi' => $layanan->konfigurasi_sesi,
                'status' => $layanan->status,
                'fasilitas_include' => $layanan->fasilitas_include,
                'gambar_utama' => $layanan->gambar_utama,
                'galeri_gambar' => $layanan->galeri_gambar,
                'fasilitas_tambahan' => $layanan->fasilitasTambahkan-
                >map(function($fasilitas) {
                    return [
                        'nama' => $fasilitas->nama_fasilitas_tambahan,
                        'deskripsi' => $fasilitas->deskripsi_fasilitas_tambahan,
                        'harga' => $fasilitas->harga_satuan
                    ];
                })->toArray()
            ]
        ])
    }
}

```

```

    });
    } catch (\Exception $e) {
        Log::error('Edit layanan error: ' . $e->getMessage());
        Log::error('Stack trace: ' . $e->getTraceAsString());

        return response()->json([
            'success' => false,
            'message' => 'Gagal memuat data layanan: ' . $e->getMessage()
        ], 500);
    }
}

/**
 * Admin: Update layanan
 */
public function update(Request $request, $id)
{
    try {
        // Log incoming request data for debugging
        Log::info('Layanan update request data:', [
            'id' => $id,
            'all_data' => $request->all(),
            'tipe_sewa' => $request->tipe_sewa,
            'tipe_sewa_old' => Layanan::find($id)->tipe_sewa ?? 'unknown'
        ]);

        $layanan = Layanan::findOrFail($id);
        if ($request->tipe_sewa !== $layanan->tipe_sewa) {
            Log::info('Tipe sewa berubah:', [
                'from' => $layanan->tipe_sewa,
                'to' => $request->tipe_sewa
            ]);

            // Cek apakah ada penyewaan aktif untuk layanan ini
            $activeBookings = $layanan->penyewaan()
                ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
                ->exists();

            if ($activeBookings) {
                Log::warning('Tipe sewa tidak bisa diubah karena ada penyewaan aktif', [
                    'layanan_id' => $id,
                    'tipe_sewa_old' => $layanan->tipe_sewa,
                    'tipe_sewa_new' => $request->tipe_sewa
                ]);

                return response()->json([
                    'success' => false,
                    'message' => 'Tidak dapat mengubah tipe sewa karena layanan ini masih
memiliki penyewaan aktif. Silakan batalkan atau selesaikan semua penyewaan terlebih
dahulu.'
                ], 422);
            }
        }
    }
}

```

```

$validationRules = [
    'nama_layanan' => 'required|string|max:100',
    'deskripsi_layanan' => 'required|string|min:10',
    'lokasi' => 'required|string',
    'jam_buka_operasional' => 'required|date_format:H:i',
    'jam_tutup_operasional' =>
'required|date_format:H:i|after:jam_buka_operasional',
    'harga_dasar' => 'required|numeric|min:0',
    'tipe_sewa' => 'required|in:per_hari,per_sesi',
    'area' => 'required|in:outdoor,indoor',
    'gambar_utama' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:20480',
    'galeri_foto.*' => 'image|mimes:jpeg,png,jpg,gif|max:20480',
    'fasilitas_include' => 'required|array|min:1',
    'fasilitas_include.*' => 'string',
    'status' => 'required|in:aktif,nonaktif'
];

}

try {
    $request->validate($validationRules);
} catch (\Illuminate\Validation\ValidationException $e) {
    Log::error('Validation failed in update:', [
        'errors' => $e->errors(),
        'request_data' => $request->all()
    ]);

    return response()->json([
        'success' => false,
        'message' => 'Validasi gagal: ' . collect($e->errors()->flatten()->implode(', '),
        'errors' => $e->errors()
    ], 422);
}

// Validasi khusus untuk layanan per sesi
if ($request->tipe_sewa === 'per_sesi') {
    try {
        $jam_buka = Carbon::parse($request->jam_buka_operasional);
        $jam_tutup = Carbon::parse($request->jam_tutup_operasional);
        $durasi_sesi = config('layanan.durasi_sesi_default'); // Durasi dari config
    ]
        $total_menit_operasional = $jam_buka->diffInMinutes($jam_tutup);
        $max_sesi_yang_muat = floor($total_menit_operasional / $durasi_sesi);
    } catch (\Exception $e) {
        Log::error('Error parsing time in update:', [
            'jam_buka' => $request->jam_buka_operasional,
            'jam_tutup' => $request->jam_tutup_operasional,
            'error' => $e->getMessage()
        ]);

        return response()->json([
            'success' => false,

```

```

        'message' => 'Format waktu tidak valid. Pastikan format jam:menit (contoh:
08:00)'
    ], 422);
    }
}

DB::beginTransaction();
try {
    $data = [
        // Data dasar (untuk semua tipe sewa)
        'nama_layanan' => $request->nama_layanan,
        'deskripsi_layanan' => $request->deskripsi_layanan,
        'lokasi' => $request->lokasi,
        'jam_buka_operasional' => $request->jam_buka_operasional,
        'jam_tutup_operasional' => $request->jam_tutup_operasional,
        'harga_dasar' => $request->harga_dasar,
        'tipe_sewa' => $request->tipe_sewa,
        'area' => $request->area,

        // Data sesi (hanya untuk per_sesi, null untuk per_hari)
        // Durasi dari config default
        'durasi_sesi_menit' => $request->tipe_sewa === 'per_sesi' ?
config('layanan.durasi_sesi_default') : null,

        // Data lainnya
        'fasilitas_include' => implode(', ', $request->fasilitas_include),
        'status' => $request->status
    ];

    // Pastikan data sesi null untuk per_hari
    if ($request->tipe_sewa === 'per_hari') {
        $data['durasi_sesi_menit'] = null;
    }

    // Handle main image upload
    if ($request->hasFile('gambar_utama')) {
        // Delete old image
        if ($layanan->gambar_utama) {
            Storage::disk('public')->delete($layanan->gambar_utama);
        }
        $data['gambar_utama'] = $request->file('gambar_utama')->store('layanan/main', 'public');
    }

    // Handle gallery images upload
    if ($request->hasFile('galeri_foto')) {
        // Delete old gallery images
        if ($layanan->galeri_gambar) {
            $oldGaleri = json_decode($layanan->galeri_gambar, true);
            if (is_array($oldGaleri)) {
                foreach ($oldGaleri as $foto) {
                    Storage::disk('public')->delete($foto);
                }
            }
        }
    }
}

```

```

    }
}

$galeri = [];
foreach ($request->file('galeri_foto') as $file) {
    $galeri[] = $file->store('layanan/gallery', 'public');
}
$data['galeri_gambar'] = json_encode($galeri);
}

$layanan->update($data);

// Update fasilitas tambahan
$layanan->fasilitasTambahkan()->delete();
if ($request->filled('fasilitas_addon')) {
    foreach ($request->fasilitas_addon as $fasilitas) {
        if (!empty($fasilitas['nama']) && !empty($fasilitas['harga'])) {
            $layanan->fasilitasTambahkan()->create([
                'id_layanan' => $layanan->id_layanan,
                'nama_fasilitas_tambahan' => $fasilitas['nama'],
                'harga_satuan' => $fasilitas['harga'],
                'deskripsi_fasilitas_tambahan' => $fasilitas['deskripsi'] ?? '',
                'status' => 'aktif'
            ]);
        }
    }
}

DB::commit();

return response()->json([
    'success' => true,
    'message' => 'Layanan berhasil diperbarui!'
]);
} catch (\Exception $e) {
    DB::rollback();
    Log::error('Update layanan error: ' . $e->getMessage());

    return response()->json([
        'success' => false,
        'message' => 'Terjadi kesalahan: ' . $e->getMessage()
    ], 500);
} catch (\Exception $e) {
    Log::error('Error in update method: ' . $e->getMessage());
    return response()->json([
        'success' => false,
        'message' => 'Terjadi kesalahan saat memperbarui layanan: ' . $e->getMessage()
    ], 500);
}

: Toggle status layanan
*/

```

```

public function toggleStatus($id)
{
    try {
        $layanannya = Layanan::findOrFail($id);
        $layanannya->status = $layanannya->status === 'aktif' ? 'nonaktif' : 'aktif';
        $layanannya->save();

        return response()->json([
            'success' => true,
            'message' => 'Status layanan berhasil diubah!',
            'status' => $layanannya->status
        ]);
    } catch (\Exception $e) {
        return response()->json([
            'success' => false,
            'message' => 'Gagal mengubah status: '. $e->getMessage()
        ], 500);
    }
}

return response()->json([
    'success' => true,
    'layanannya' => [
        'id_layanan' => $layanannya->id_layanan,
        'nama_layanan' => $layanannya->nama_layanan,
        'deskripsi_layanan' => $layanannya->deskripsi_layanan,
        'lokasi' => $layanannya->lokasi,
        'jam_buka_operasional' => $layanannya->jam_buka_operasional ?
Carbon::parse($layanannya->jam_buka_operasional)->format('H:i') : '',
        'jam_tutup_operasional' => $layanannya->jam_tutup_operasional ?
Carbon::parse($layanannya->jam_tutup_operasional)->format('H:i') : '',
        'harga_dasar' => $layanannya->harga_dasar,
        'tipe_sewa' => $layanannya->tipe_sewa,
        'area' => $layanannya->area,
        'durasi_sesi_menit' => $layanannya->durasi_sesi_menit,
        'jeda_antar_sesi' => $layanannya->jeda_antar_sesi,
        'max_sesi_per_hari' => $layanannya->max_sesi_per_hari,
        'konfigurasi_sesi' => $layanannya->konfigurasi_sesi,
        'status' => $layanannya->status,
        'fasilitas_include' => $layanannya->fasilitas_include,
        'gambar_utama' => $layanannya->gambar_utama,
        'galeri_gambar' => $layanannya->galeri_gambar,
        'fasilitas_tambahan' => $layanannya->fasilitasTambahan-
>map(function($fasilitas) {
            return [
                'nama' => $fasilitas->nama_fasilitas_tambahan,
                'deskripsi' => $fasilitas->deskripsi_fasilitas_tambahan,
                'harga' => $fasilitas->harga_satuan
            ];
        })->toArray()
    ]
]);

```

```

    } catch (\Exception $e) {
        Log::error('Error in layanan show method', [
            'id' => $id,
            'error' => $e->getMessage(),
            'trace' => $e->getTraceAsString()
        ]);

        return response()->json([
            'success' => false,
            'message' => 'Gagal memuat detail layanan: ' . $e->getMessage()
        ], 500);
    }
}

public function checkBookingStatus($id)
{
    try {
        $layanan = Layanan::findOrFail($id);

        // Cek penyewaan aktif
        $activeBookings = $layanan->penyewaan()
            ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
            ->get();

        $hasActiveBookings = $activeBookings->count() > 0;
        $activeBookingsCount = $activeBookings->count();

        return response()->json([
            'success' => true,
            'hasActiveBookings' => $hasActiveBookings,
            'activeBookingsCount' => $activeBookingsCount,
            'activeBookings' => $activeBookings->map(function($booking) {
                return [
                    'id' => $booking->id_penyewaan,
                    'kode_sewa' => $booking->kode_sewa,
                    'tanggal_penyewaan' => $booking->tanggal_penyewaan->format('d/m/Y'),
                    'status' => $booking->status_penyewaan,
                    'customer' => $booking->user->name ?? 'N/A'
                ];
            })
        ]);
    } catch (\Exception $e) {
        Log::error('Error checking booking status: ' . $e->getMessage());

        return response()->json([
            'success' => false,
            'message' => 'Gagal mengecek status penyewaan: ' . $e->getMessage()
        ], 500);
    }
}

public function customerShow(Request $request, $id)
{
    $layanan = Layanan::with('fasilitas Tambahan')
        ->where('status', 'aktif')

```

```

->findOrFail($id);

$ids = [];
if (Auth::check()) {
    $u = Auth::user();
    foreach ([ $u->getCustomerId(), $u->id_user ] as $salt) {
        if ($salt) $ids[] = $salt;
    }
}
$unpaidBookings = Penyewaan::whereIn('id_customer', $ids)
->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
->with(['pembayaran', 'layanan'])
->get()
->filter(function($booking) {
    if ($booking->status_penyewaan === 'cancelled') {
        return false;
    }

    // Skip booking yang sudah expired (tanggal sudah lewat)
    if (Carbon::parse($booking->tanggal_penyewaan)->isPast()) {
        return false;
    }

    // Hitung total yang sudah dibayar (HANYA status 'lunas')
    $totalPaid = $booking->pembayaran()
        ->where('status_pembayaran', 'lunas')
        ->sum('jumlah_bayar');

    // Cek apakah ada pembayaran yang menunggu verifikasi admin
    $hasUploadedPayment = $booking->pembayaran()
        ->where('status_pembayaran', 'uploaded')
        ->exists();

    $hasRemainingPayment = $totalPaid < $booking->total_harga;

    // Jika ada pembayaran yang menunggu verifikasi admin, dianggap unpaid
    if ($hasUploadedPayment) {
        return true; // Menunggu verifikasi admin = tidak bisa booking baru
    }

    // Jika masih ada sisa pembayaran, dianggap unpaid
    return $hasRemainingPayment;
});

if ($unpaidBookings->count() > 0) {
    $unpaidList = $unpaidBookings->map(function($booking) {
        $totalPaid = $booking->pembayaran()
            ->where('status_pembayaran', 'lunas')
            ->sum('jumlah_bayar');
        $sisapembayaran = $booking->total_harga - $totalPaid;

        $statusText = "

```

```

        if ($booking->pembayaran()->where('status_pembayaran', 'uploaded')->exists())
        {
            $statusText = ' (Menunggu Verifikasi Admin)';
        } elseif ($sisapembayaran > 0) {
            $statusText = ' - Sisa: Rp ' . number_format($sisapembayaran, 0, ',', '.');
        }

        return $booking->layanan->nama_layanan . ' (' . $booking->tanggal_penyewaan-
        >format('d/m/Y') . ') ' . $statusText;
    }->join(', ');

    return redirect()->route('customer.penyewaan.unpaid-warning')
        ->with('error', "Anda masih memiliki booking yang belum lunas atau menunggu
    verifikasi: {$unpaidList}. Silakan lunasi atau tunggu verifikasi terlebih dahulu sebelum
    melakukan booking baru.");
}

$unavailableDates = [];

if ($layanan->tipe_sewa === 'per_hari') {
    // Hanya untuk per_hari, blok tanggal yang sudah diboooking
    $unavailableDates = Penyewaan::where('id_layanan', $id)
        ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
        ->where('tanggal_penyewaan', '>=', now())
        ->where('tanggal_penyewaan', '<=', now()->addMonths(3))
        ->pluck('tanggal_penyewaan')
        ->map(function ($date) {
            return Carbon::parse($date)->format('Y-m-d');
        })
        ->unique()
        ->values()
        ->toArray();
}

$holidayDates = $this->getHolidayDates($id);

$selectedDate = $request->get('date');
$timeSlots = [];

if ($selectedDate) {
    $timeSlots = $this->getAvailableTimeSlots($id, $selectedDate);
}

// Get related layanan (similar price range)
$relatedLayanan = Layanan::where('status', 'aktif')
    ->where('id_layanan', '!=', $id)
    ->whereBetween('harga_dasar', [
        $layanan->harga_dasar * 0.8,
        $layanan->harga_dasar * 1.2
    ])
    ->take(4)
    ->get();

```

```

return view('customer.layanan.detail', compact(
    'layanannya',
    'unavailableDates',
    'holidayDates',
    'timeSlots',
    'relatedLayanan'
));
}

public function customerShowByName(Request $request, $nama)
{
    // Convert dash back to space for search
    $searchName = str_replace('-', ' ', $nama);

    $layanannya = Layanan::with('fasilitas Tambahan')
        ->where('status', 'aktif')
        ->where('nama_layanan', 'like', '%' . $searchName . '%')
        ->first();

    if (!$layanannya) {
        abort(404, 'Layanan tidak ditemukan');
    }

    $id = $layanannya->id_layanan;

    $sids = [];
    if (Auth::check()) {
        $u = Auth::user();
        foreach ($u->getCustomerId(), $u->id_user as $salt) {
            if ($salt) $sids[] = $salt;
        }
    }

    $unpaidBookings = Penyewaan::whereIn('id_customer', $sids)
        ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
        ->with(['pembayaran', 'layanannya'])
        ->get()
        ->filter(function($booking) {
            if ($booking->status_penyewaan === 'cancelled') return false;
            if (Carbon::parse($booking->tanggal_penyewaan)->isPast()) return false;
            $totalPaid = $booking->pembayaran()->where('status_pembayaran', 'lunas')-
                >sum('jumlah_bayar');
            $hasUploadedPayment = $booking->pembayaran()->where('status_pembayaran',
                'uploaded')->exists();
            return ($totalPaid < $booking->total_harga) || $hasUploadedPayment;
        });

    if ($unpaidBookings->count() > 0) {
        $unpaidList = $unpaidBookings->map(function($booking) {
            $totalPaid = $booking->pembayaran()->where('status_pembayaran', 'lunas')-
                >sum('jumlah_bayar');
            $sisapembayaran = $booking->total_harga - $totalPaid;
            $statusText = "";

```

```

        if ($booking->pembayaran()->where('status_pembayaran', 'uploaded')->exists())
        {
            $statusText = ' (Menunggu Verifikasi Admin)';
        } elseif ($sisapembayaran > 0) {
            $statusText = ' - Sisa: Rp ' . number_format($sisapembayaran, 0, ',', '.');
        }
        return $booking->layanan->nama_layanan . ' (' . $booking->tanggal_penyewaan-
        >format('d/m/Y') . ') ' . $statusText;
    }->join(', ');
    return redirect()->route('customer.penyewaan.unpaid-warning')
    ->with('error', "Anda masih memiliki booking yang belum lunas atau menunggu
    verifikasi: {$SunpaidList}. Silakan lunasi atau tunggu verifikasi terlebih dahulu sebelum
    melakukan booking baru.");
}

sesi tertentu yang unavailable
$unavailableDates = [];

if ($layanan->tipe_sewa === 'per_hari') {
    // Hanya untuk per_hari, blok tanggal yang sudah diboooking
    $unavailableDates = Penyewaan::where('id_layanan', $id)
    ->whereIn('status_penyewaan', ['pending', 'reserved', 'confirmed'])
    ->where('tanggal_penyewaan', '>=', now())
    ->pluck('tanggal_penyewaan')
    ->map(function ($date) {
        return Carbon::parse($date)->format('Y-m-d');
    })
    ->unique()
    ->values()
    ->toArray();
}
// Untuk per_sesi, $unavailableDates tetap array kosong

// Get holiday schedule data using new method
$holidayDates = $this->getHolidayDates($id);

// Get availability for specific date (if requested)
$selectedDate = $request->get('date');
$timeSlots = [];

if ($selectedDate) {
    $timeSlots = $this->getAvailableTimeSlots($id, $selectedDate);
}

// Get related layanan (similar price range)
$relatedLayanan = Layanan::where('status', 'aktif')
->where('id_layanan', '!=', $id)
->whereBetween('harga_dasar', [
    $layanan->harga_dasar * 0.8,
    $layanan->harga_dasar * 1.2
])
->take(4)
->get();

```

```

return view('customer.layanan.detail', compact(
    'layanan',
    'unavailableDates',
    'holidayDates',
    'timeSlots',
    'relatedLayanan'
));
}

private function getAvailableTimeSlots($layananId, $date)
{
    try {
        $layanan = Layanan::find($layananId);
        if (!$layanan) {
            \Log::warning("Layanan not found: {$layananId}");
            return [];
        }

        return $slots;
    } catch (\Exception $e) {
        \Log::error("Error in getAvailableTimeSlots: " . $e->getMessage());
        return [];
    }
}

/*
untuk mendapatkan available time slots
*/
public function getAvailableSlotsApi(Request $request)
{
    $request->validate([
        'id_layanan' => 'required|integer',
        'tanggal' => 'required|date'
    ]);

    $layananId = $request->id_layanan;
    $date = $request->tanggal;

    try {
        $slots = $this->getAvailableTimeSlots($layananId, $date);

        return response()->json([
            'success' => true,
            'slots' => $slots,
            'debug' => [
                'total_slots' => count($slots),
                'available_slots' => count(array_filter($slots, fn($s) => $s['available'])),
                'booked_slots' => count(array_filter($slots, fn($s) => !$s['available']))
            ]
        ]);
    } catch (\Exception $e) {
        \Log::error("Error in getAvailableSlotsApi: " . $e->getMessage());

        return response()->json([

```

```

        'success' => false,
        'message' => 'Terjadi kesalahan: ' . $e->getMessage(),
        'slots' => []
    ], 500);
    }
}

public function getHolidayDates($layananId)
{
    $SholidayDates = [];

    // Get specific dates (no time limit)
    $specificHolidays = JadwalLibur::where('id_layanan', $layananId)
        ->where('jenis', 'tanggal_tertentu')
        ->where('tanggal', '>=', now())
        ->get();

    foreach ($specificHolidays as $sholiday) {
        $SholidayDates[Carbon::parse($sholiday->tanggal)->format('Y-m-d')] = [
            'keterangan' => $sholiday->keterangan,
            'jenis' => $sholiday->jenis,
            'hari' => $sholiday->hari
        ];
    }

    $weeklyHolidays = JadwalLibur::where('id_layanan', $layananId)
        ->where('jenis', 'harian')
        ->whereNotNull('hari')
        ->get();

    $startDate = now();
    $endDate = now()->addYears(2);

    foreach ($weeklyHolidays as $sholiday) {
        $currentDate = $startDate->copy();

        while ($currentDate <= $endDate) {
            $dayName = strtolower($currentDate->format('l')); // Monday, Tuesday, etc.
            $indonesianDay = [
                'monday' => 'senin',
                'tuesday' => 'selasa',
                'wednesday' => 'rabu',
                'thursday' => 'kamis',
                'friday' => 'jumat',
                'saturday' => 'sabtu',
                'sunday' => 'minggu'
            ];

            if ($indonesianDay[$dayName] === $sholiday->hari) {
                $dateKey = $currentDate->format('Y-m-d');
                if (!isset($SholidayDates[$dateKey])) {
                    $SholidayDates[$dateKey] = [

```

```

        'keterangan' => $holiday->keterangan,
        'jenis' => $holiday->jenis,
        'hari' => $holiday->hari
    ];
    }
}

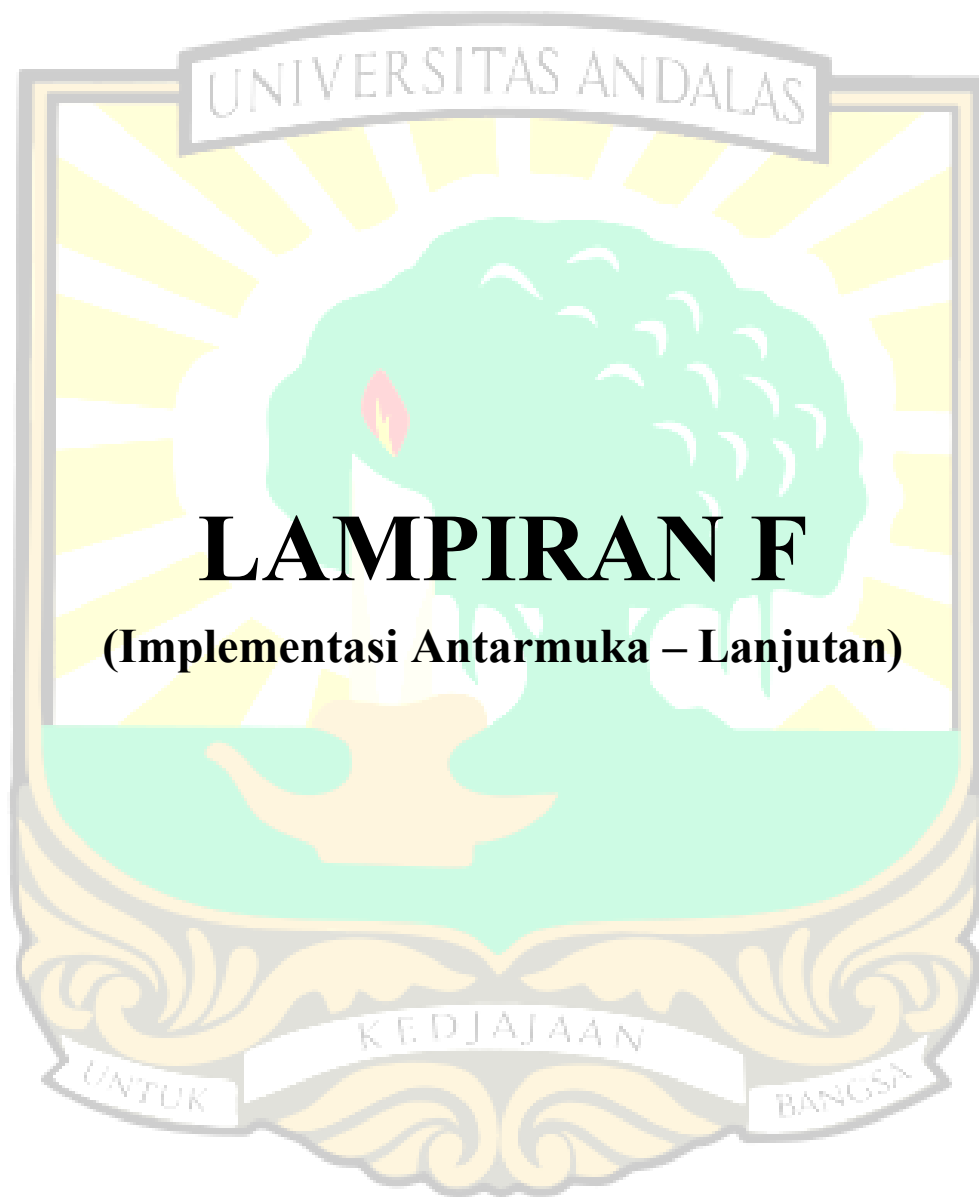
$currentTime->addDay();
}
}

return $holidayDates;
}

public function getStats()
{
    return [
        'total_layanan' => Layanan::count(),
        'layanan_aktif' => Layanan::where('status', 'aktif')->count(),
        'layanan_populer' => Layanan::withCount(['pembayaran' => function ($query) {
            $query->where('status', 'verified');
        }])->orderBy('pembayaran_count', 'desc')->take(5)->get(),
        'rata_rata_harga' => round(Layanan::avg('harga_dasar'), 0)
    ];
}
}
}

```



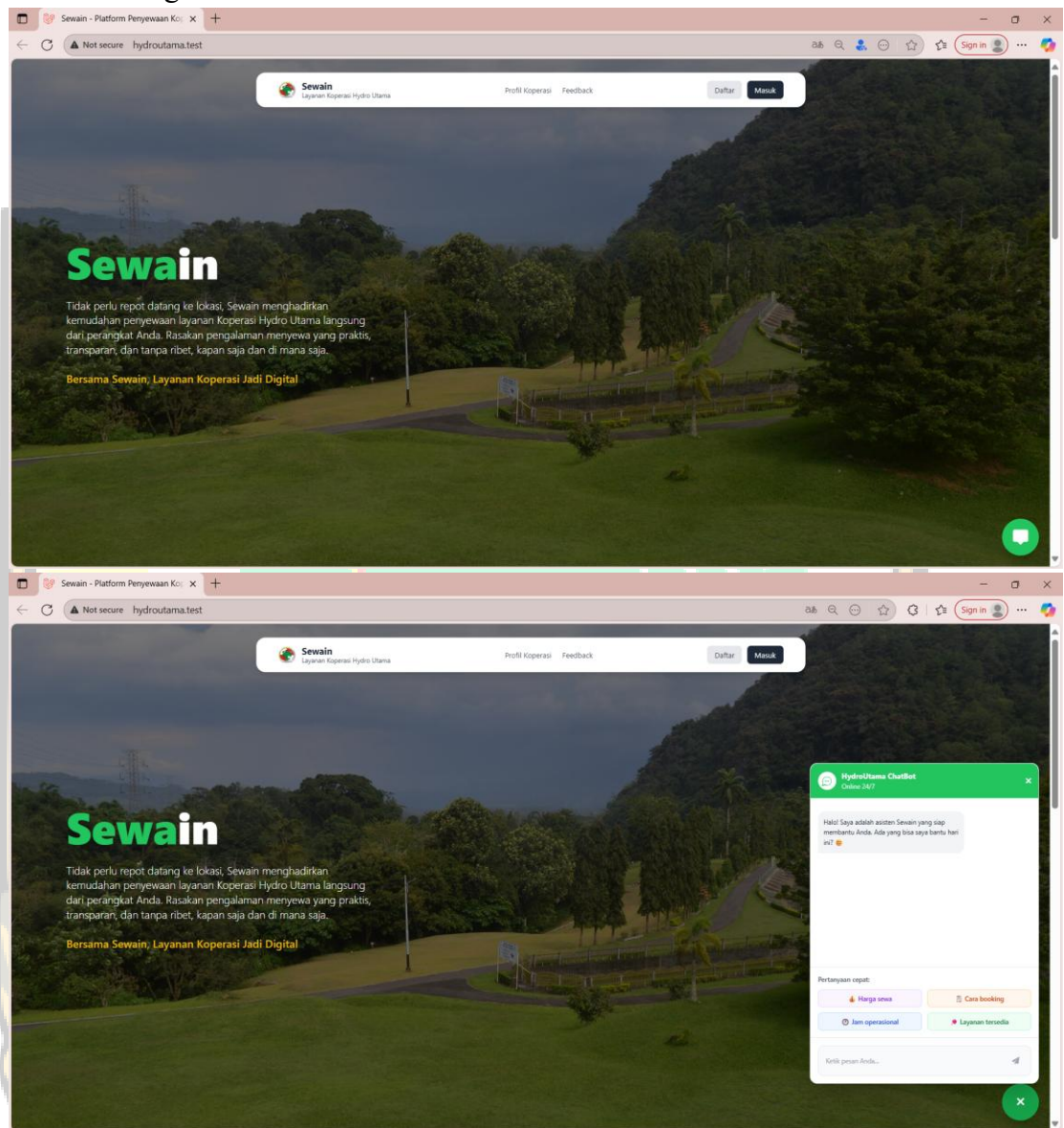


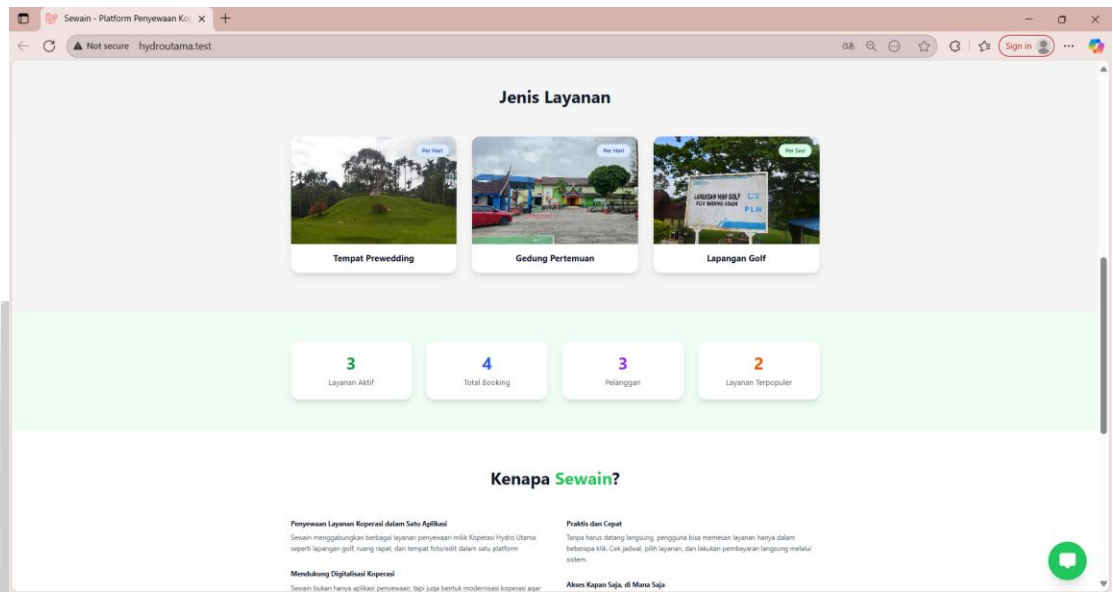
LAMPIRAN F

(Implementasi Antarmuka – Lanjutan)

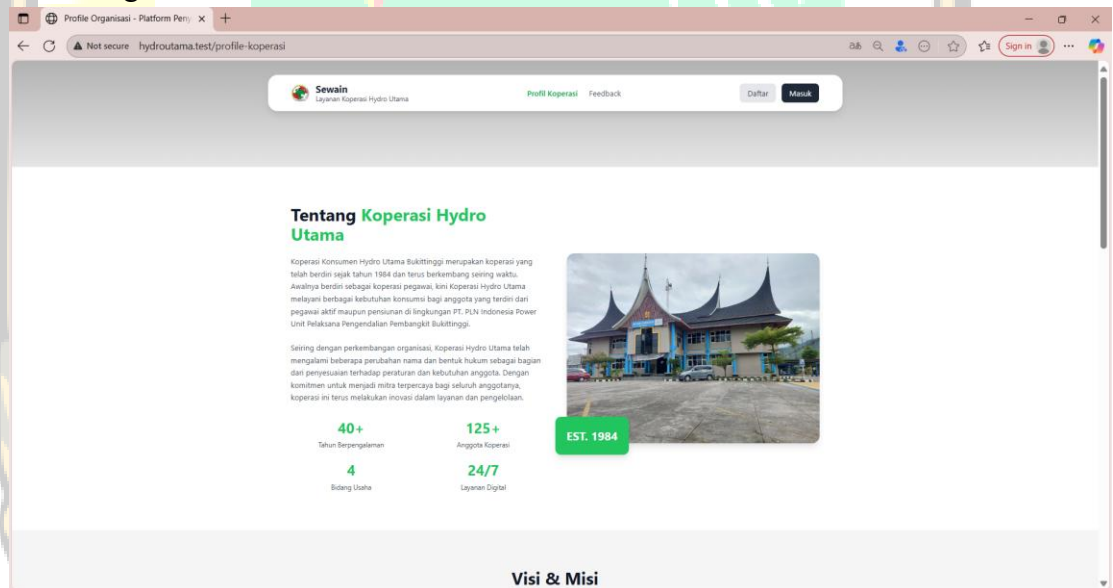
A. Utama

1. Welcome Page

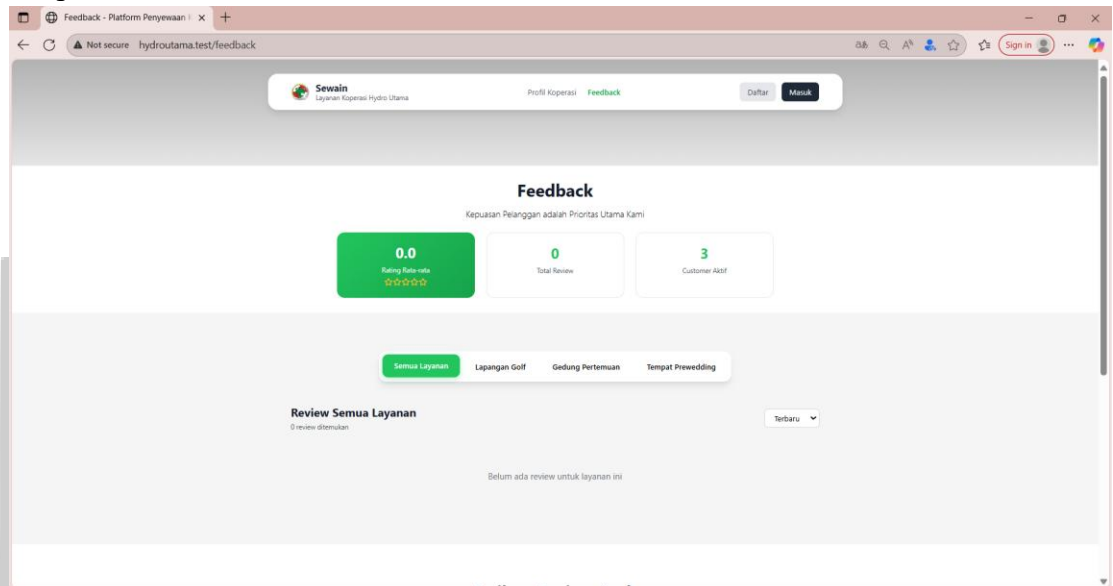




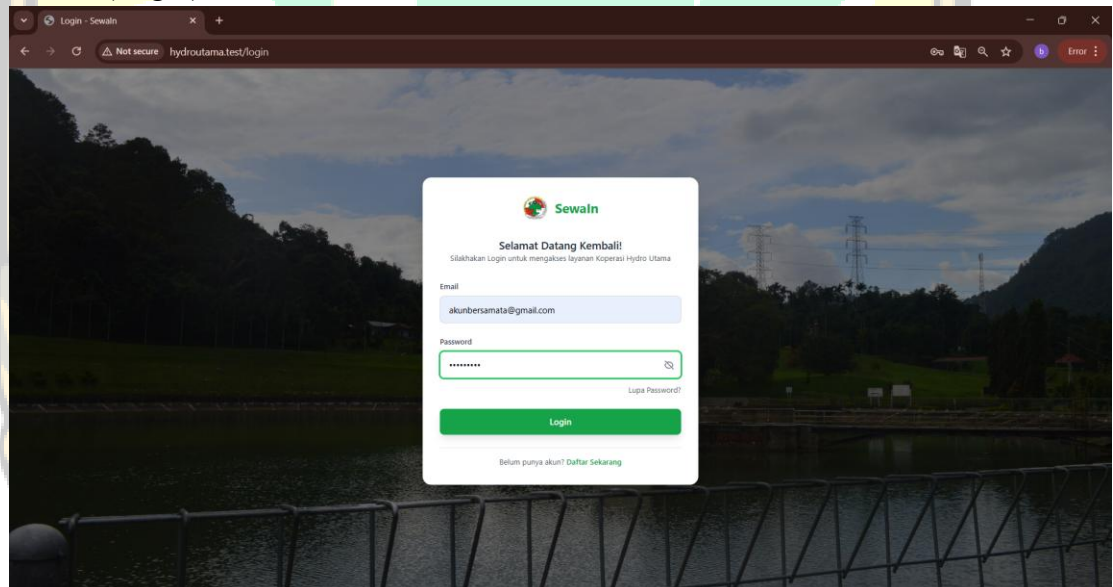
2. Profil Organisasi



3. Umpan balik



4. Masuk (Login)



5. Daftar Akun (untuk pelanggan)

The screenshot shows a web browser window with the address bar displaying "hydroutama.test/register". The page features a registration form for "Sewain". The form includes the following fields and options:

- Nama Lengkap** (Full Name): Input field containing "Putra Xavier".
- Email**: Input field containing "akunbersamata@gmail.com".
- No. HP** (Phone Number): Input field containing "081245789632".
- Jenis Kelamin** (Gender): Dropdown menu with "Laki-laki" (Male) selected.
- Password**: Input field with masked characters "*****".
- Konfirmasi Password** (Confirm Password): Input field with masked characters "*****".
- Alamat** (Address): Input field containing "Padang Panjang".

Below the form is a green button labeled "Daftar" (Register). At the bottom of the form, there is a link: "Sudah punya akun? [Masuk Sekarang](#)" (Already have an account? [Login Now](#)).

The screenshot shows a web browser window with the address bar displaying "hydroutama.test/verify-email". The page features an email verification confirmation message from "Sewain". The message includes the following content:

- Verifikasi Email Anda** (Verify Your Email).
- A green envelope icon.
- Text: "Kami telah mengirim email verifikasi ke: **akunbersamata@gmail.com**".
- Text: "Silahkan cek email Anda dan klik link aktivasi untuk mengaktifkan akun." (Please check your email and click the activation link to activate your account).
- A green button: "Akun berhasil dibuat! Silahkan cek email untuk verifikasi" (Account created successfully! Please check email for verification).
- Text: "Tidak menerima email?" (Did not receive email?).
- A green button: "Kirim Ulang Email" (Resend Email).
- Text: "Kembali ke Login" (Back to Login).
- Tips:** (Tips) section with a light blue background:
 - Cek folder Spam/Junk email
 - Pastikan email address benar
 - Tunggu beberapa menit untuk email masuk

B. Admin

1. Edit Profil

Edit Profile
Kata informasi profil Anda

Informasi Personal

Nama Lengkap
Kadaryanti Ningsih

Email
hydroutama2025@gmail.com

Nomor Telepon
081236547896

Alamat

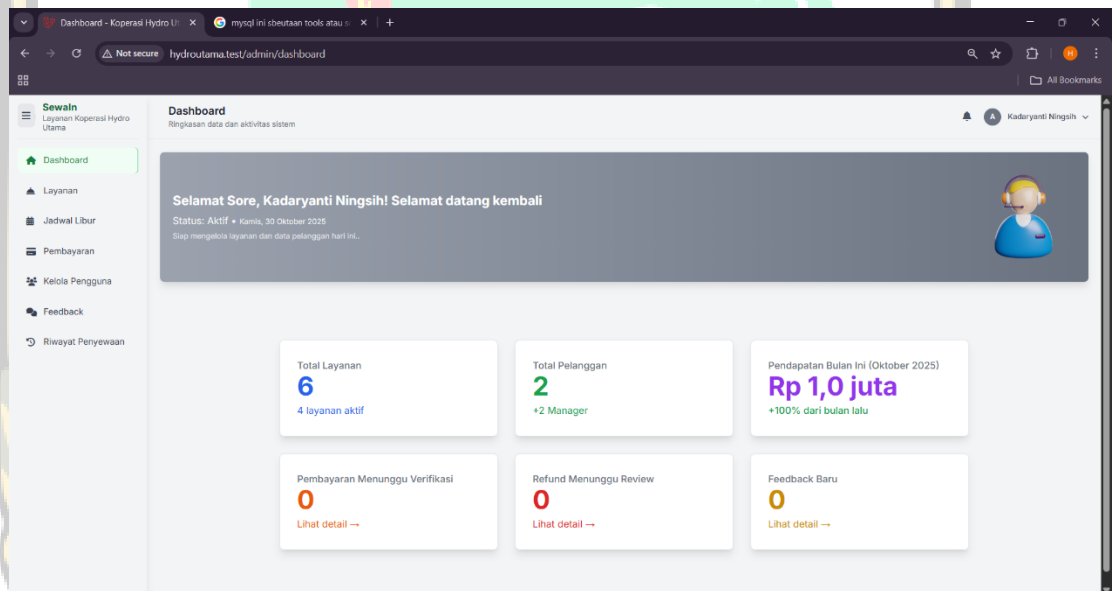
Buktinggi Pin

Keamanan & Pengaturan

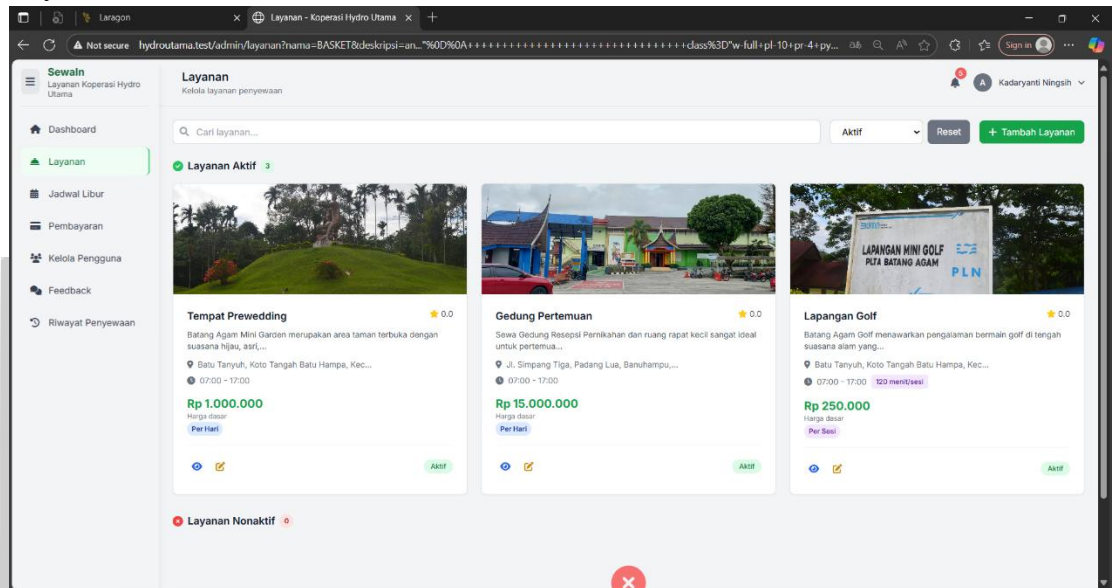
Ubah Password
Password default dari sistem, ubah untuk keamanan

Informasi Akun
Akun dibuat: 18 Agustus 2025
Login terakhir: 41 kali yang lalu

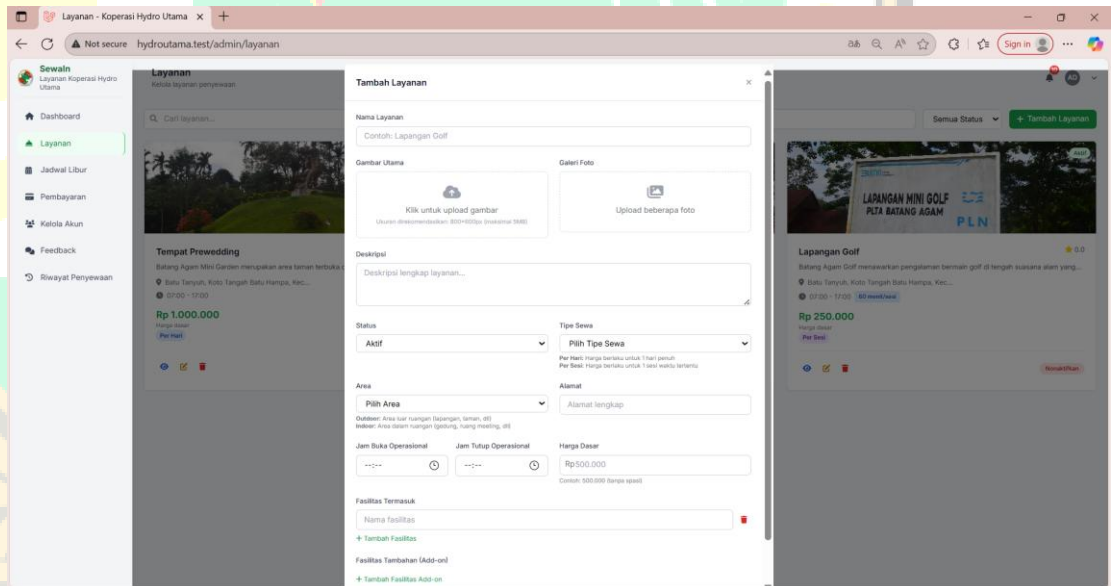
2. Dashboard



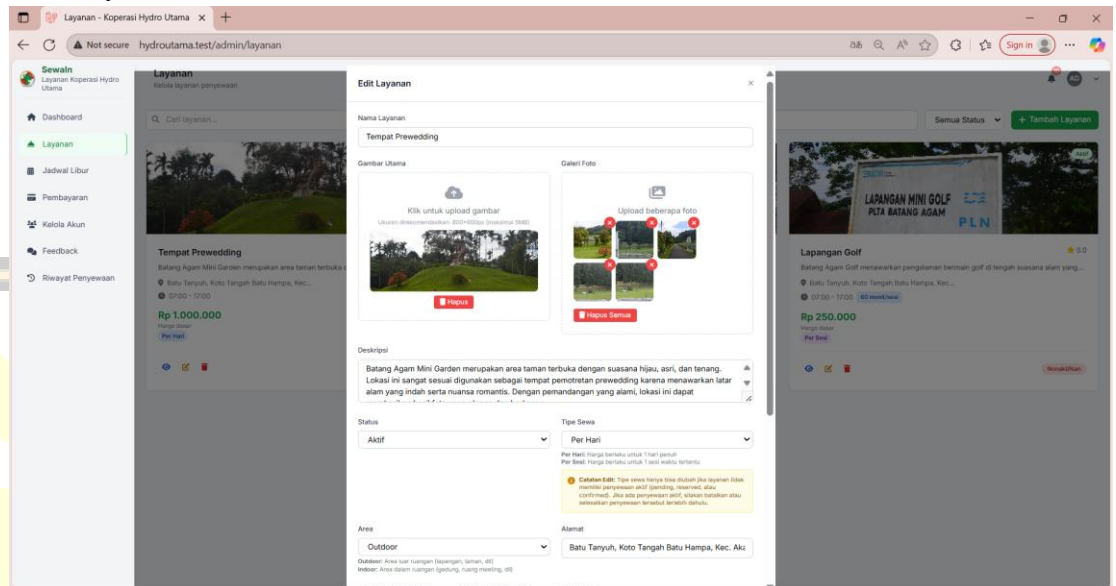
3. Layanan



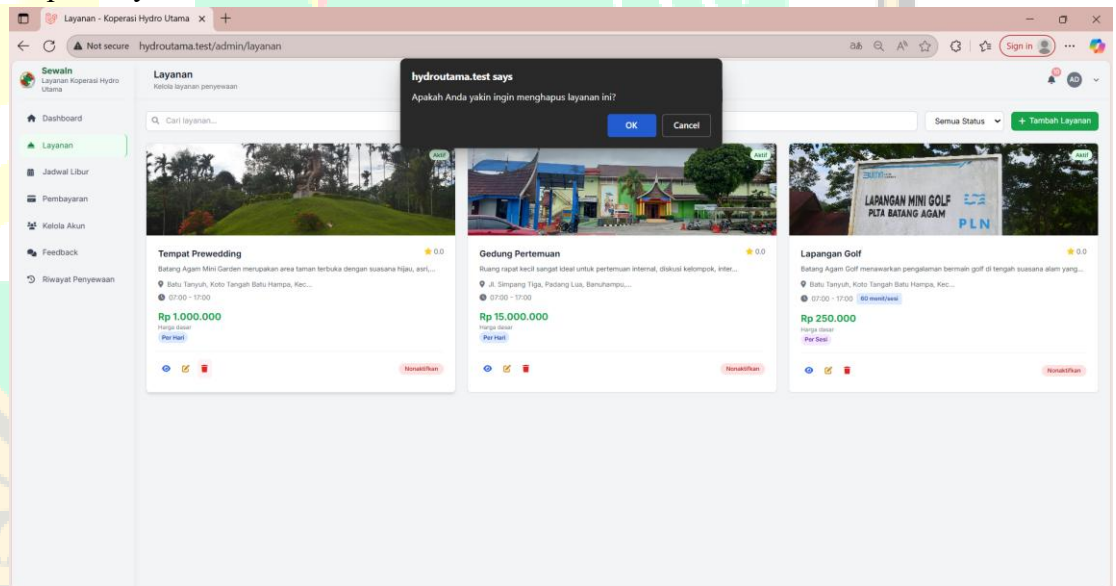
1) Tambah Layanan



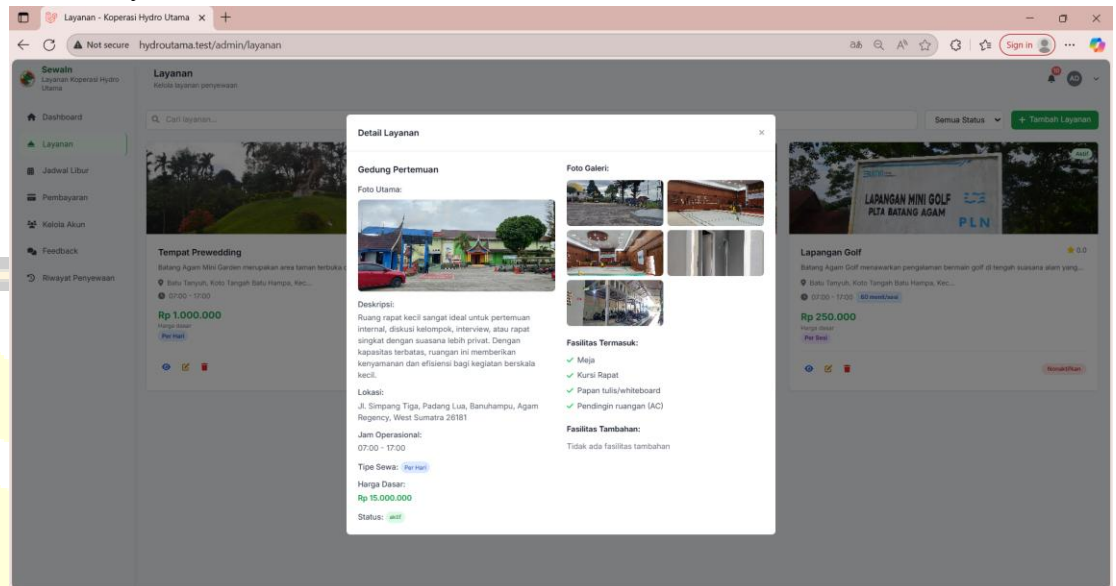
2) Edit Layanan



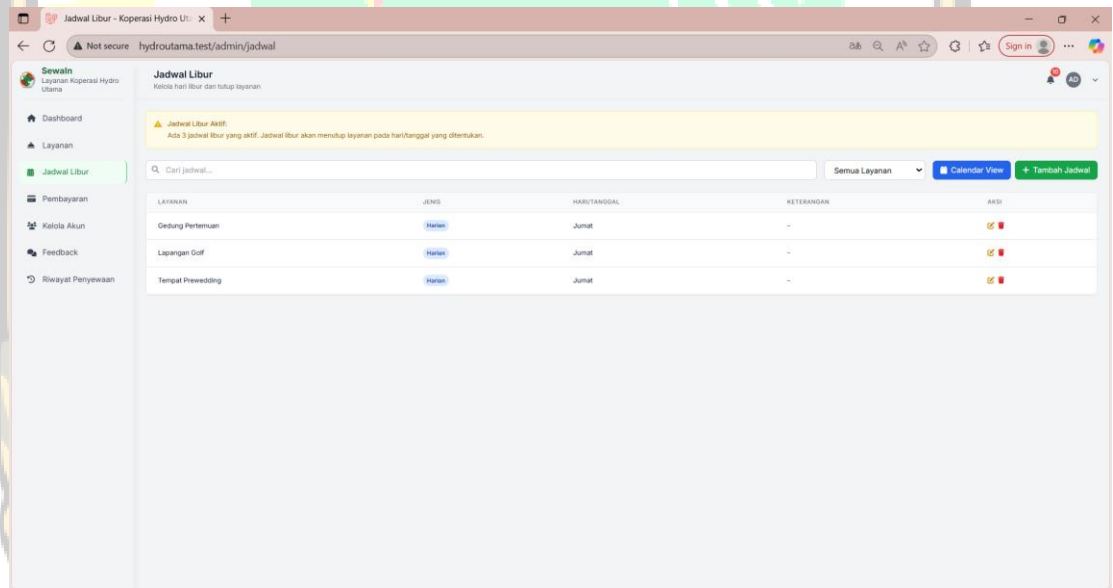
3) Hapus Layanan



4) Detail Layanan



4. Jadwal Libur



1) Tambah Jadwal Libur

The screenshot shows the 'Tambah Jadwal' (Add Holiday Schedule) form in the 'Sewain' application. The form is titled 'Tambah Jadwal' and contains the following fields:

- Layanan ***: A dropdown menu with 'Pilih Layanan' selected.
- Jenis Jadwal ***: A dropdown menu with 'Pilih Jenis' selected.
- Keterangan**: A text area with the placeholder 'Keterangan opsional...'. There is a small icon of a document with a pencil in the bottom right corner of the text area.

At the bottom of the form are two buttons: 'Batal' (Cancel) and 'Simpan' (Save).

Background information visible in the interface:

- Header: 'Sewain Layanan Koperasi Hydro Utama'
- Left sidebar: Dashboard, Layanan, Jadwal Libur (selected), Pembayaran, Kelola Akun, Feedback, Riwayat Penyesuaian.
- Main content area: 'Jadwal Libur' section with a search bar 'Cari jadwal...', a table of existing holidays, and a '+ Tambah Jadwal' button.
- Table headers: 'LAYANAN', 'JENIS', 'KETERANGAN', 'AKSI'.
- Table rows: 'Gedung Pertemuan', 'Lapangan Golf', 'Tempat Prewedding', each with a 'Hapus' button.
- Top right: 'Semua Layanan', 'Calendar View', and 'Tambah Jadwal' button.

2) Edit Jadwal Libur

The screenshot shows the 'Edit Jadwal Libur' (Edit Holiday Schedule) form in the 'Sewain' application. The form is titled 'Edit Jadwal Libur' and contains the following fields:

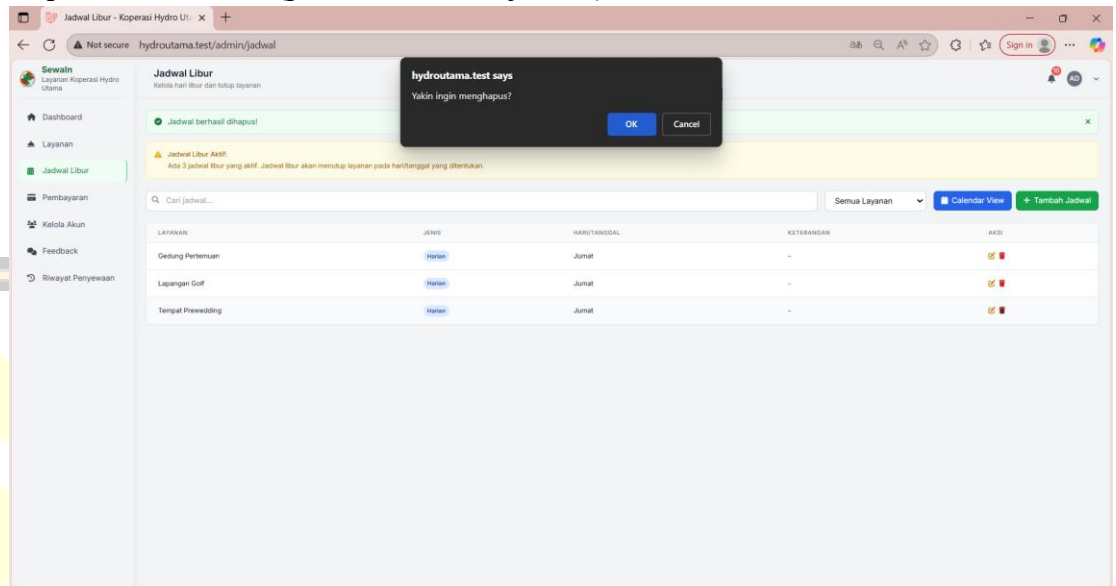
- Layanan ***: A dropdown menu with 'Gedung Pertemuan' selected.
- Jenis Jadwal ***: A dropdown menu with 'Harian (Setiap Hari)' selected.
- Hari ***: A dropdown menu with 'Jumat' selected.
- Keterangan**: A text area with the placeholder 'Keterangan opsional...'. There is a small icon of a document with a pencil in the bottom right corner of the text area.

At the bottom of the form are two buttons: 'Batal' (Cancel) and 'Update' (Update).

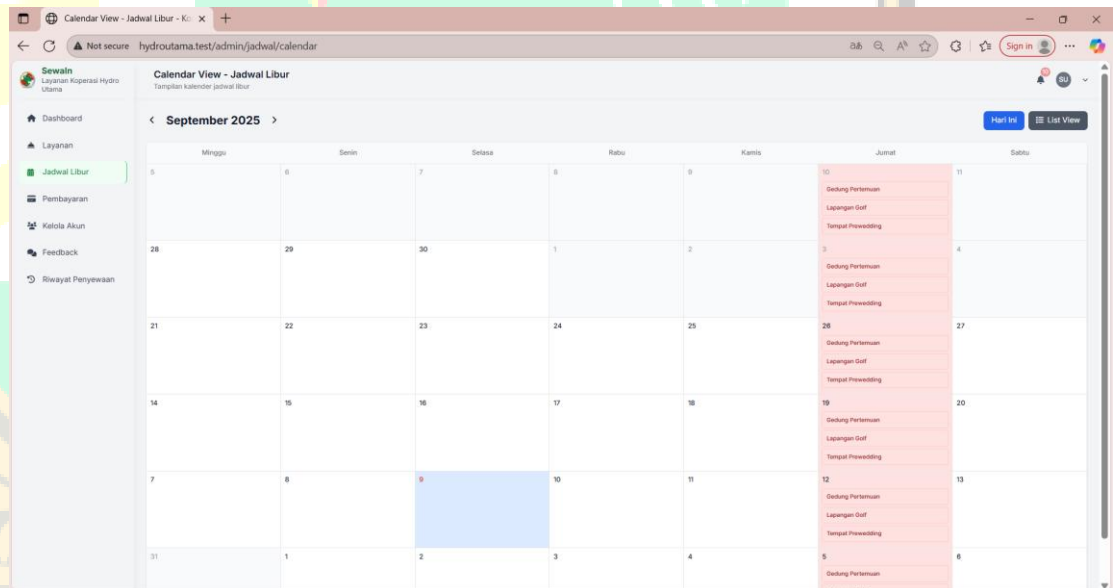
Background information visible in the interface:

- Header: 'Sewain Layanan Koperasi Hydro Utama'
- Left sidebar: Dashboard, Layanan, Jadwal Libur (selected), Pembayaran, Kelola Akun, Feedback, Riwayat Penyesuaian.
- Main content area: 'Edit Jadwal Libur' section with a search bar 'Cari jadwal...', a table of existing holidays, and an 'Edit Jadwal' button.
- Table headers: 'LAYANAN', 'JENIS', 'KETERANGAN', 'AKSI'.
- Table rows: 'Gedung Pertemuan', 'Lapangan Golf', 'Tempat Prewedding', each with a 'Hapus' button.
- Top right: 'Semua Layanan', 'Calendar View', and 'Tambah Jadwal' button.

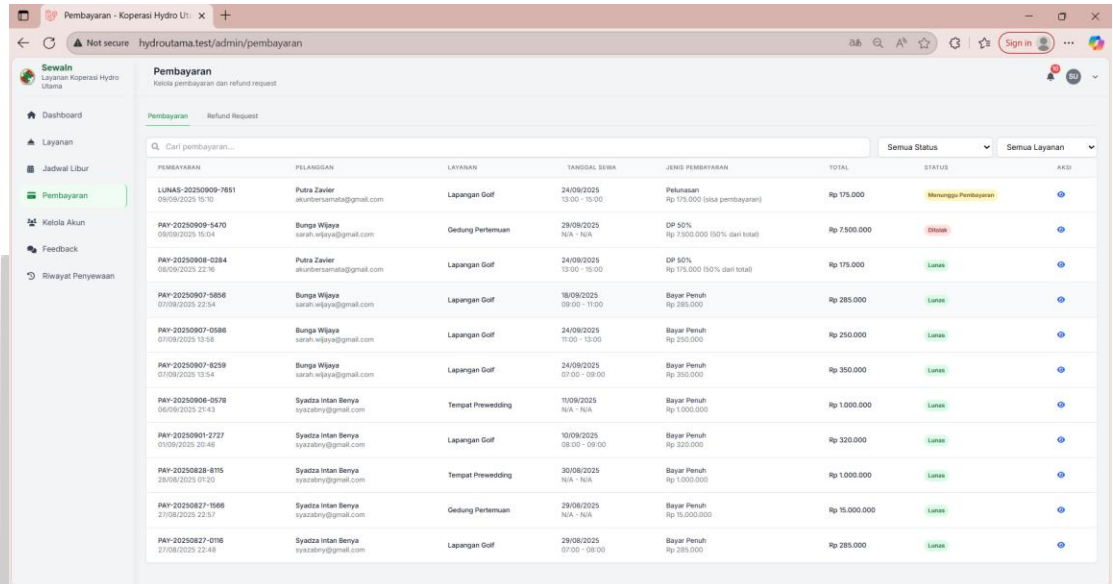
3) Hapus Jadwal Libur (pilih disalah satu jadwal)



4) Calendar View

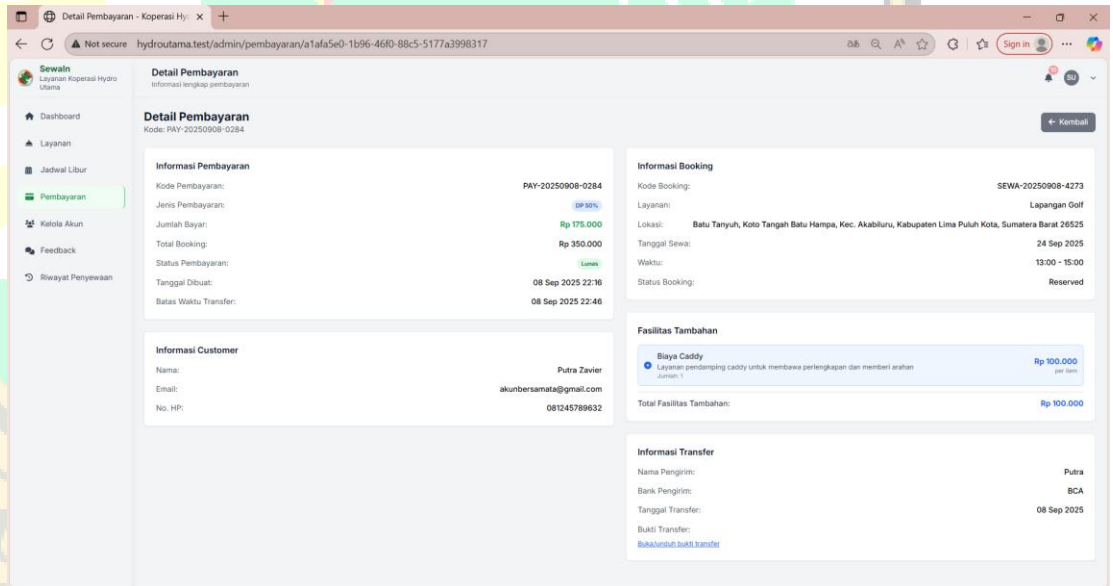


5. Verifikasi Pembayaran



PEMBAYARAN	PELANGGAN	LAYANAN	TANGGAL SEWA	JENIS PEMBAYARAN	TOTAL	STATUS	Aksi
LUNAS-20250909-7651 09/09/2025 16:10	Putra Xavier akunbersamata@gmail.com	Lapangan Golf	24/09/2025 13:00 - 19:00	Pelunasan Rp 175.000 (sisa pembayaran)	Rp 175.000	Menunggu Pembayaran	
PAY-20250909-1470 09/09/2025 15:04	Bunga Wijaya sarah.wijaya@gmail.com	Gedung Pertemuan	28/09/2025 N/A - N/A	DP 50% Rp 7.500.000 (50% dari total)	Rp 7.500.000	Dibayar	
PAY-20250908-0284 08/09/2025 22:16	Putra Xavier akunbersamata@gmail.com	Lapangan Golf	24/09/2025 13:00 - 19:00	DP 50% Rp 175.000 (50% dari total)	Rp 175.000	Lunas	
PAY-20250907-5858 07/09/2025 22:54	Bunga Wijaya sarah.wijaya@gmail.com	Lapangan Golf	18/09/2025 08:00 - 11:00	Bayar Penuh Rp 285.000	Rp 285.000	Lunas	
PAY-20250907-0586 07/09/2025 13:56	Bunga Wijaya sarah.wijaya@gmail.com	Lapangan Golf	24/09/2025 10:00 - 13:00	Bayar Penuh Rp 250.000	Rp 250.000	Lunas	
PAY-20250907-8259 07/09/2025 13:54	Bunga Wijaya sarah.wijaya@gmail.com	Lapangan Golf	24/09/2025 07:00 - 09:00	Bayar Penuh Rp 350.000	Rp 350.000	Lunas	
PAY-20250906-0578 06/09/2025 21:43	Syafza Intan Benya syafzabny@gmail.com	Tempat Prewedding	10/09/2025 N/A - N/A	Bayar Penuh Rp 1.000.000	Rp 1.000.000	Lunas	
PAY-20250901-2727 01/09/2025 20:46	Syafza Intan Benya syafzabny@gmail.com	Lapangan Golf	10/09/2025 08:00 - 09:00	Bayar Penuh Rp 320.000	Rp 320.000	Lunas	
PAY-20250828-8155 28/08/2025 07:20	Syafza Intan Benya syafzabny@gmail.com	Tempat Prewedding	30/08/2025 N/A - N/A	Bayar Penuh Rp 1.000.000	Rp 1.000.000	Lunas	
PAY-20250827-1568 27/08/2025 22:57	Syafza Intan Benya syafzabny@gmail.com	Gedung Pertemuan	28/08/2025 N/A - N/A	Bayar Penuh Rp 15.000.000	Rp 15.000.000	Lunas	
PAY-20250827-0196 27/08/2025 22:48	Syafza Intan Benya syafzabny@gmail.com	Lapangan Golf	29/08/2025 07:00 - 19:00	Bayar Penuh Rp 285.000	Rp 285.000	Lunas	

1) Detail Salah Satu Pelanggan



Detail Pembayaran
Informasi lengkap pembayaran
Kode: PAY-20250908-0284

Informasi Pembayaran

Kode Pembayaran: PAY-20250908-0284
Jenis Pembayaran: DP 50%
Jumlah Bayar: Rp 175.000
Total Booking: Rp 350.000
Status Pembayaran: Lunas
Tanggal Dibuat: 08 Sep 2025 22:16
Batas Waktu Transfer: 08 Sep 2025 22:46

Informasi Customer

Nama: Putra Xavier
Email: akunbersamata@gmail.com
No. HP: 081245789632

Informasi Booking

Kode Booking: SEWA-20250908-4273
Layanan: Lapangan Golf
Lokasi: Batu Tanyuh, Koto Tangah Batu Hampa, Kec. Akabikuru, Kabupaten Lima Puluh Kota, Sumatera Barat 26525
Tanggal Sewa: 24 Sep 2025
Waktu: 13:00 - 15:00
Status Booking: Reserved

Fasilitas Tambahan

Biaya Caddy
Layanan pendamping caddy untuk membawa perlengkapan dan memberi arahan
Jumlah: 1
Rp 100.000 per hari

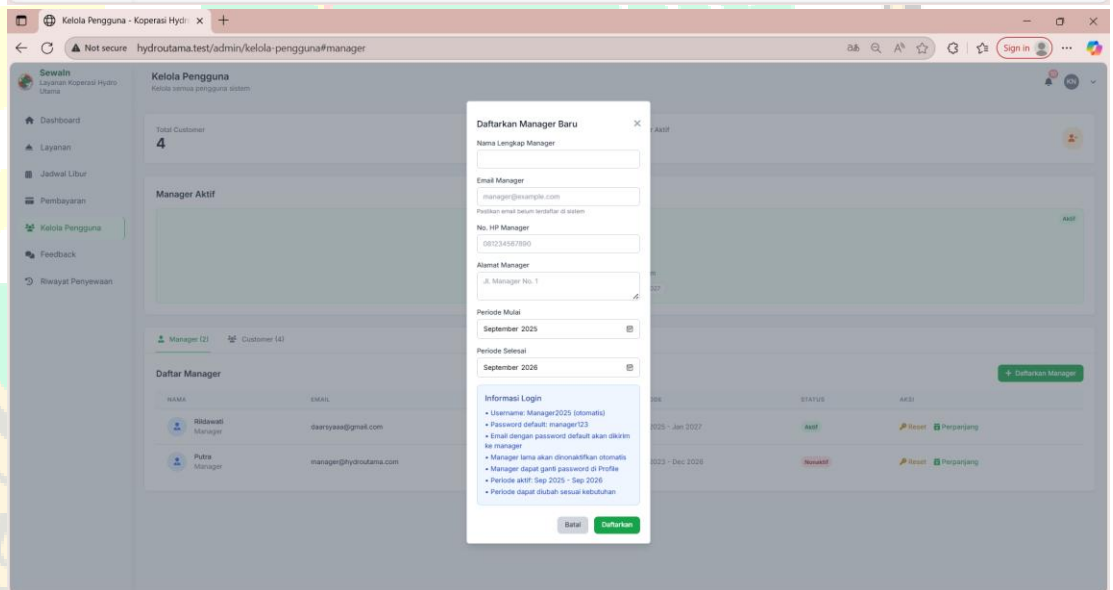
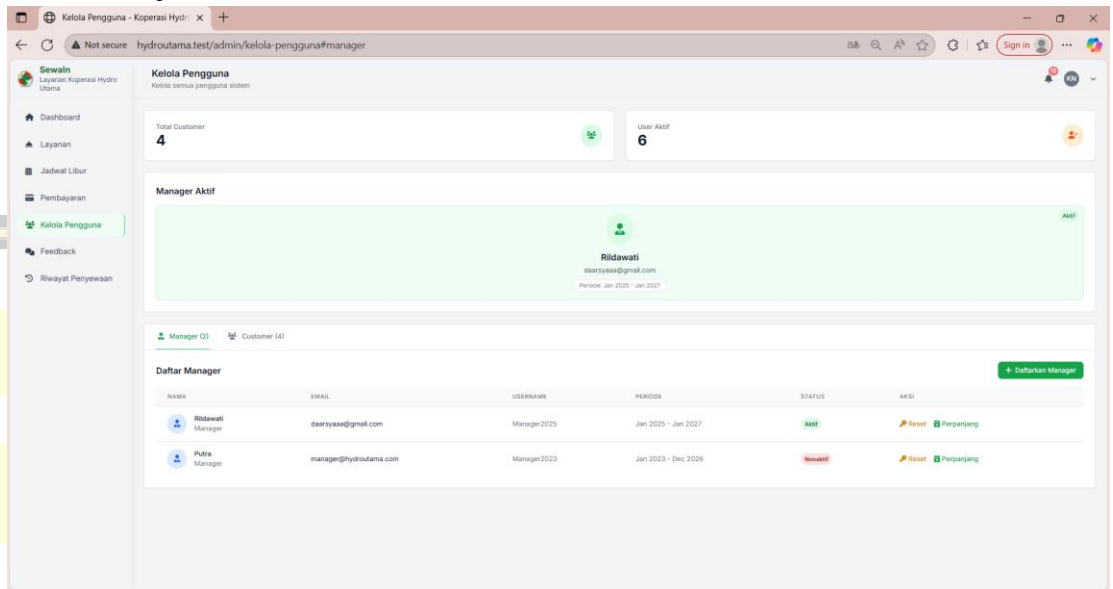
Total Fasilitas Tambahan: Rp 100.000

Informasi Transfer

Nama Pengirim: Putra
Bank Pengirim: BCA
Tanggal Transfer: 08 Sep 2025
Bukti Transfer: BuktiTransferBuktiTransfer

6. Akun

1) Akun Manajer



2) Akun Pelanggan

Kelola Pengguna
Kelola semua pengguna sistem

Total Customer: 4 User Aktif: 6

Manager Aktif

Rildawati
doarsyasa@gmail.com
Periode: Jan 2025 - Jan 2027

Informasi Sistem Status Customer

- Aktif: Customer bisa login dan melakukan booking
- Nonaktif: Customer tidak bisa login, tapi data tetap tersimpan
- Tipe: Gantikan simbol "Aktif/Nonaktif" untuk mengontrol akses customer. Data customer tidak akan dihapus untuk menjaga integritas sistem.

Daftar Customer

NAMA	EMAIL	NO. HP	STATUS	BERGABUNG	AKSI
Piri Permata Customer	syatzairan03@gmail.com	081254875214	Aktif	22/08/2025	Detail Aktif
Putra Xavier Customer	akunbersamut@gmail.com	081245789632	Aktif	08/08/2025	Detail Aktif
Syatza Intan Benya Customer	syatzabny@gmail.com	081254852121	Aktif	27/06/2025	Detail Aktif

7. Umpan balik

Kelola Feedback
Kelola dan balas feedback dari customer

Total Feedback: 1 Belum Dibalas: 1 Sudah Dibalas: 0 Rating Rata-rata: 4.0

Daftar Feedback

Syatza Intan Benya
Gedung Perikanan Menunggu Balasan
1 minggu yang lalu
Rating: 4.0 (4/5)

8. Riwayat Penyewaan

Riwayat Penyewaan
Menampilkan daftar penyewaan dari semua pelanggan

Filter Layanan: Semua Layanan | Filter Bulan: Semua Bulan | Filter Tahun: Semua Tahun | Unduh PDF

NO	LAYANAN	PELANGGAN	TANGGAL SEWA	STATUS	TOTAL BAYAR	TINDAKAN
1	Gedung Pertemuan	Bunga Wijaya	29 September 2025	Ditolak	Rp 15.000.000	
2	Lapangan Golf	Bunga Wijaya	24 September 2025	Lunas	Rp 350.000	
3	Lapangan Golf	Putra Xavier	24 September 2025	Mundur	Rp 350.000	
4	Lapangan Golf	Bunga Wijaya	24 September 2025	Lunas	Rp 250.000	
5	Lapangan Golf	Bunga Wijaya	18 September 2025	Lunas	Rp 285.000	
6	Tempat Prewedding	Syadza Intan Benya	11 September 2025	Lunas	Rp 1.000.000	
7	Lapangan Golf	Syadza Intan Benya	10 September 2025	Lunas	Rp 320.000	
8	Tempat Prewedding	Syadza Intan Benya	30 Agustus 2025	Lunas	Rp 1.000.000	
9	Gedung Pertemuan	Syadza Intan Benya	29 Agustus 2025	Lunas	Rp 15.000.000	
10	Lapangan Golf	Syadza Intan Benya	29 Agustus 2025	Lunas	Rp 285.000	

C. Pelanggan

1. Edit Profil

Edit Profile
Kelola informasi profil Anda

Informasi Personal

Nama Lengkap: Bunga Wijaya

Email: sarah.wijaya@gmail.com
Email tidak dapat diubah untuk keamanan akun

Jenis Kelamin: Perempuan

Nomor Telepon: 081234567892

Alamat: J. Pelanggan No. 1, Jakarta Selatan

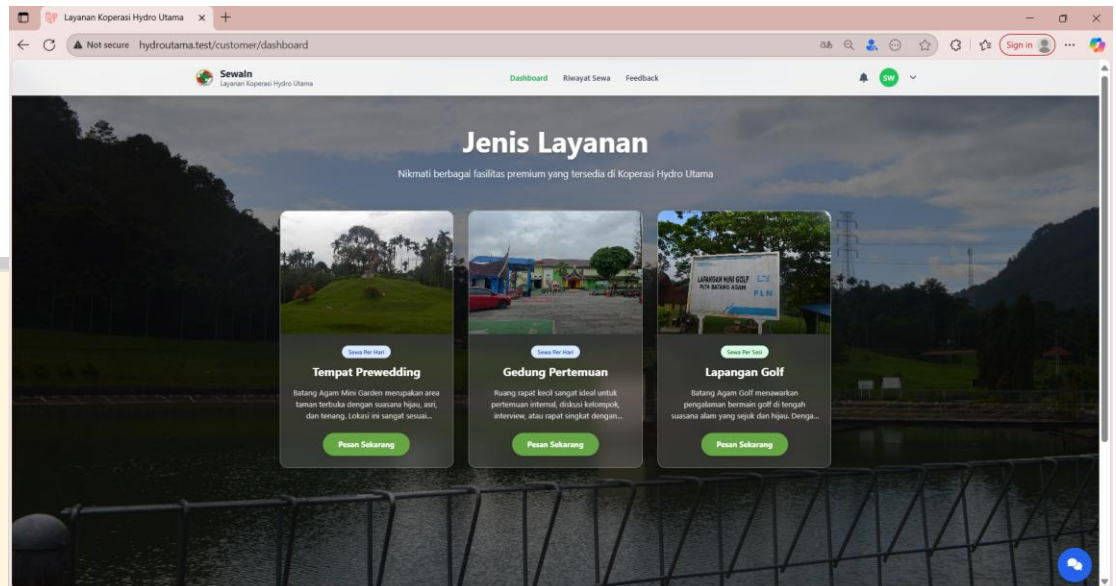
Keamanan & Pengaturan

Ubah Password: Password default dari sistem, ubah untuk keamanan

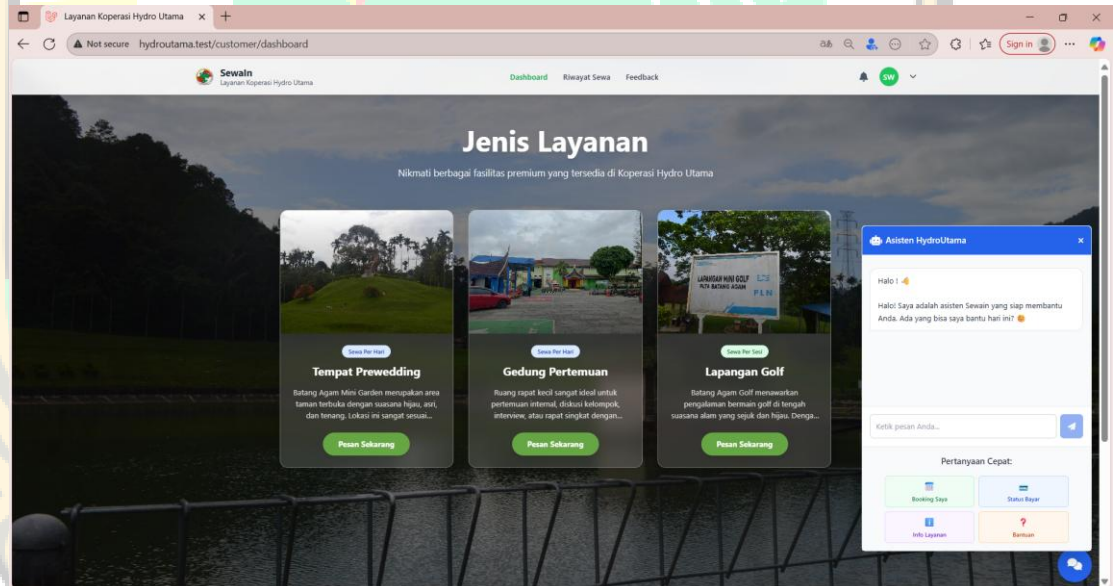
Ubah Password

Informasi Akun

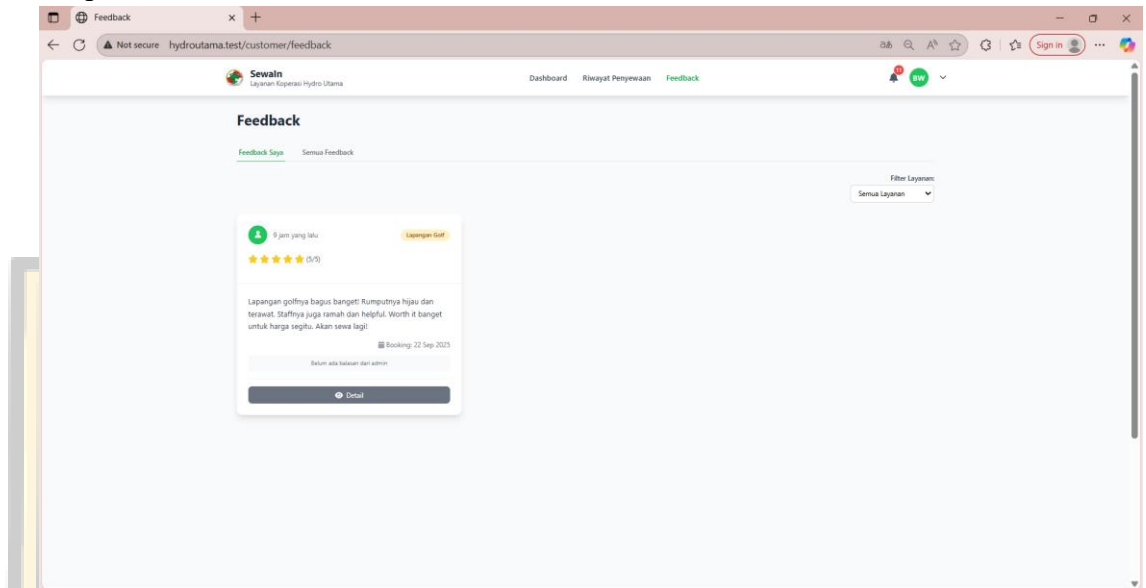
2. Dashboard



3. Chatbot

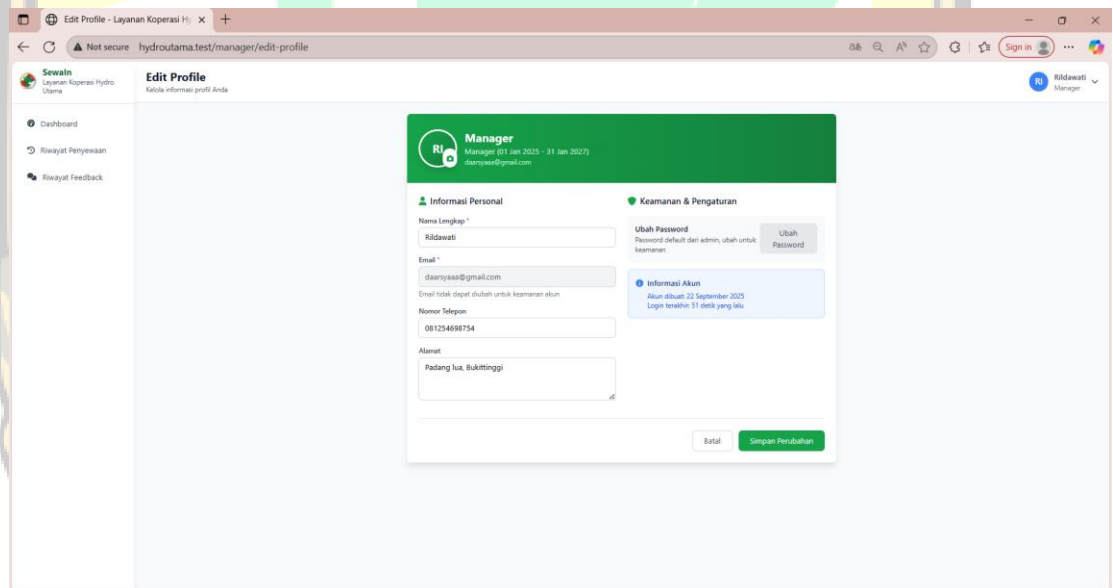


4. Umpan balik

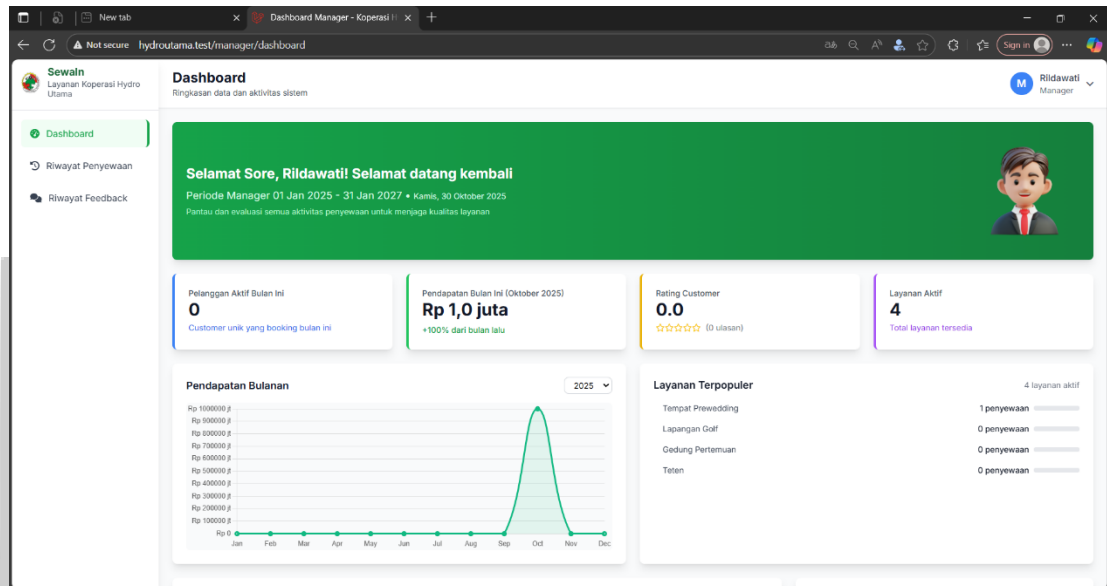


D. Manajer

1. Edit Profil



2. Dashboard



3. Riwayat penyewaan

Riwayat Penyewaan
Menu ini menampilkan data penyewaan yang telah dilakukan oleh pelanggan

Pilih Layanan: Semua Layanan | Rentang Tanggal: 31/08/2025 - 29/09/2025 | Generate Laporan

ID	LAYANAN	NAMA	EMAIL	TANGGAL	TOTAL HARGA	TOTAL BAYAR	STATUS	TINDAKAN
1	Lapangan Golf	Rutha Xavier	akunbersamata@gmail.com	29 September 2025	Rp 420.000	Rp 420.000	Lunas	Detail
2	Gedung Pertemuan	Bunga Wijaya	sarahvijaya@gmail.com	29 September 2025	Rp 15.000.000	Rp 15.000.000	Lunas	Detail
3	Lapangan Golf	Syadza Intan Benya	syadzany@gmail.com	29 September 2025	Rp 320.000	Rp 320.000	Menunggu	Detail
4	Lapangan Golf	Peti Permata	syadzawati3@gmail.com	28 September 2025	Rp 330.000	Rp 330.000	Lunas	Detail
5	Lapangan Golf	Syadza Intan Benya	syadzany@gmail.com	28 September 2025	Rp 250.000	Rp 250.000	Lunas	Detail
6	Lapangan Golf	Bunga Wijaya	sarahvijaya@gmail.com	24 September 2025	Rp 330.000	Rp 330.000	Lunas	Detail
7	Lapangan Golf	Rutha Xavier	akunbersamata@gmail.com	24 September 2025	Rp 350.000	Rp 350.000	Lunas	Detail
8	Lapangan Golf	Bunga Wijaya	sarahvijaya@gmail.com	24 September 2025	Rp 280.000	Rp 280.000	Lunas	Detail
9	Lapangan Golf	Bunga Wijaya	sarahvijaya@gmail.com	18 September 2025	Rp 285.000	Rp 285.000	Lunas	Detail
10	Tempat Prewedding	Rutha Xavier	akunbersamata@gmail.com	13 September 2025	Rp 1.000.000	Rp 1.000.000	Dibatal	Detail

Menampilkan 1 sampai 10 dari 15 hasil

4. Umpan balik

Riwayat Feedback

Mengumpulkan daftar feedback dari pelanggan terkait layanan yang digunakan

Pilih Layanan: Semua Layanan

Rentang Tanggal: 01/09/2025 to 30/09/2025

Generate Laporan

NO	LAYANAN	NAMA	TANGGAL	KOMENTAR	RATING	TINDAKAN
1	Lapangan Golf	Syaida Intan Beriya	22 September 2025	Lapangan golfnya oke, rumputnya hijau dan terawat. Tapi ada beberapa area yan...	★★★★☆	●
2	Gedung Pertemuan	Syaida Intan Beriya	22 September 2025	Ruang rapatnya nyaman dan fasilitasnya lengkap. Projector dan sound system b...	★★★★☆	●
3	Lapangan Golf	Bunga Wijaya	22 September 2025	Lapangan golfnya bagus banget! Rumputnya hijau dan terawat. Staffnya juga ra...	★★★★★	●





LAMPIRAN G

(Pengujian Unit dan Pengujian Aplikasi – Lanjutan)

A. Pengujian Unit

1. Model

a. Customer (Pelanggan)

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	getFullName	user.nama_lengkap = "John Doe"	"John Doe"	"John Doe"	Mengambil nama lengkap dari user. Pastikan metode getFullName() mengembalikan nama yang sesuai dengan input nama_lengkap.	Lulus
2	getPhoneNumber	user.no_hp = "08123456789"	"08123456789"	"08123456789"	Mengambil nomor telepon dari user. Verifikasi bahwa nomor yang dikembalikan sesuai dengan input no_hp.	Lulus
3	getAvatarInitials	user.nama_lengkap = "John Doe"	"JD"	"JD"	Mengambil inisial avatar berdasarkan nama lengkap. Verifikasi apakah dua huruf pertama yang diambil sesuai dengan input nama lengkap.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
4	getAvatarInitials	user.nama_1 engkap = "John"	"JO"	"JO"	Mengambil inisial avatar dengan satu kata nama. Verifikasi bahwa inisial diambil dari dua huruf pertama nama.	Lulus
5	getAvatarInitials	user.nama_1 engkap = null	"C"	"C"	Inisial avatar default saat nama kosong. Verifikasi bahwa jika nama kosong, maka inisial yang dikembalikan adalah "C".	Lulus
6	getAvatarColor	-	"bg-green-500"	"bg-green-500"	Mengembalikan warna default avatar. Verifikasi apakah warna avatar sesuai dengan warna default yang diharapkan.	Lulus
7	getGenderLabelAttribute	customer.gender = 'L'	"Laki-laki"	"Laki-laki"	Mengembalikan label gender sesuai nilai L. Pastikan hasilnya adalah	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					"Laki-laki" sesuai dengan nilai gender yang diberikan.	
8	getGenderLabelAttribute	customer.gender = 'P'	"Perempuan"	"Perempuan"	Mengembalikan label gender sesuai nilai P. Pastikan hasilnya adalah "Perempuan" sesuai dengan nilai gender yang diberikan.	Lulus
9	getFormattedPhoneAttribute	user.no_hp = '08123456789'	"+628123456789"	"+628123456789"	Mengembalikan nomor HP yang diformat dengan benar. Pastikan bahwa nomor HP diformat dengan tanda "+" di depan, menggantikan angka 0.	Lulus
10	getFormattedPhoneAttribute	user.no_hp = '628123456789'	"628123456789"	"628123456789"	Mengembalikan nomor HP yang diformat tanpa angka 0. Pastikan hasilnya sesuai dengan input yang	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					diberikan, tanpa mengubah nomor yang sudah dimulai dengan angka 6.	
11	getFormattedPhoneNumberAttribute	user.no_hp = null	""	""	Mengembalikan string kosong saat nomor HP kosong. Verifikasi bahwa fungsi mengembalikan hasil yang benar saat nomor HP null.	Lulus
12	getProfilePhoto	user.profile_photo = 'avatars/customer.jpg'	'http://localhost/storage/avatars/customer.jpg'	'http://localhost/storage/avatars/customer.jpg'	Mengembalikan URL foto profil jika ada. Pastikan URL yang dikembalikan adalah yang benar dan sesuai dengan data yang ada pada profil_photo.	Lulus
13	getProfilePhoto	user.profile_photo = null	null	null	Mengembalikan null jika foto profil tidak ada. Verifikasi bahwa	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					jika foto profil tidak ada, maka hasil yang dikembalikan adalah null.	
14	scopeActive	Mock whereHas('user', is_active=true)	Builder terfilter dengan kondisi is_active = true	Builder terfilter dengan kondisi is_active = true	Menguji scope untuk menambahkan filter pada is_active yang bernilai true. Pastikan builder dikembalikan dengan kondisi filter yang benar.	Lulus
15	scopeByGender	Mock where gender = 'L'	Builder terfilter sesuai gender 'L'	Builder terfilter sesuai gender 'L'	Menguji scope untuk menambahkan filter pada gender sesuai parameter yang diberikan. Verifikasi apakah builder dikembalikan dengan kondisi filter yang benar.	Lulus

b. Admin

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	getFullName()	user.nama_lengkap = "Admin Utama"	"Admin Utama"	"Admin Utama"	Verifikasi bahwa nama lengkap admin sesuai dengan input user.nama_lengkap.	Lulus
2	getPhoneNumber()	user.no_hp = "08123456789"	"08123456789"	"08123456789"	Verifikasi bahwa nomor HP admin sesuai dengan input user.no_hp.	Lulus
3	getAddress()	user.alamat = "Jl. Admin No. 1"	"Jl. Admin No. 1"	"Jl. Admin No. 1"	Verifikasi alamat admin sesuai dengan input user.alamat.	Lulus
4	getAvatarInitials()	user.nama_lengkap = "Admin Utama"	"AU"	"AU"	Verifikasi bahwa dua huruf pertama dari nama lengkap diambil sebagai inisial avatar.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
5	getAvatarInitials()	user.nama_le ngkap = "Admin"	"AD"	"AD"	Verifikasi bahwa dua huruf pertama diambil dari nama pendek sebagai inisial avatar.	Lulus
6	getAvatarInitials()	user.nama_le ngkap = null	"A"	"A"	Verifikasi bahwa default inisial adalah "A" jika nama kosong.	Lulus
7	getAvatarColor()	—	"bg-gray-500"	"bg-gray-500"	Verifikasi bahwa warna avatar default adalah bg-gray-500.	Lulus
8	getFormattedPhoneAttribute()	user.no_hp = "08123456789"	"+628123456789"	"+628123456789"	Verifikasi bahwa nomor HP diformat dengan benar, mengubah angka 0 menjadi +62.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
9	getFormattedPhoneAttribute()	user.no_hp = "628123456789"	"628123456789"	"628123456789"	Verifikasi bahwa nomor HP tanpa angka 0 tetap dikembalikan sesuai input.	Lulus
10	getAdminGreeting()	—	Mengandung "Selamat"	Mengandung "Selamat"	Verifikasi bahwa greeting admin mengandung kata "Selamat".	Lulus
11	getAdminPeriodFormatted()	—	"Permanent"	"Permanent"	Verifikasi bahwa periode admin selalu "Permanent".	Lulus
12	getProfilePhoto()	user.profil_photo = 'avatars/admin.jpg'	'http://localhost/storage/avatars/admin.jpg'	'http://localhost/storage/avatars/admin.jpg'	Verifikasi bahwa URL foto profil dikembalikan dengan benar jika ada foto.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
13	getProfilPhoto()	user.profil_photo = null	null	null	Verifikasi bahwa foto profil adalah null jika tidak ada foto.	Lulus
14	getAdminPhoto()	user.profil_photo = null	'http://localhost/images/admin.png'	'http://localhost/images/admin.png'	Verifikasi bahwa foto admin default dikembalikan jika tidak ada foto pengguna.	Lulus
15	getAdminPhoto()	user.profil_photo = 'profil/a.png'	'http://localhost/storage/profil/a.png'	'http://localhost/storage/profil/a.png'	Verifikasi bahwa URL foto pengguna dikembalikan jika ada foto pengguna.	Lulus
16	getTotalPembayaranVerified()	Mock relation count (3)	3	3	Verifikasi bahwa total pembayaran yang diverifikasi sesuai dengan hasil mock.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
17	getTotalRefundHandled()	Mock relation count (2)	2	2	Verifikasi bahwa total refund yang di-handle sesuai dengan hasil mock.	Lulus
18	scopeActive()	Mock whereHas('user', is_active=true)	Builder terfilter dengan kondisi is_active = true	Builder terfilter dengan kondisi is_active = true	Verifikasi bahwa query scope is_active bekerja sesuai dengan parameter yang diberikan.	Lulus

c. Manajer

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Nama lengkap manajer	getFullName()	user.nama_lengkap='Manajer Utama'	"Manajer Utama"	Ambil nama dari relasi user yang diset manual.	Lulus
2	Nomor HP manajer	getPhoneNumber()	user.no_hp='08123456789'	"08123456789"	Ambil no_hp dari relasi user.	Lulus
3	Alamat manajer	getAddress()	user.alamat='Jl. Manajer No. 1'	"Jl. Manajer No. 1"	Ambil alamat dari relasi user.	Lulus
4	Inisial avatar (2 kata)	getAvatarInitials()	user.nama_lengkap='Manajer Utama'	"MU"	Ekstrak huruf pertama tiap kata.	Lulus
5	Inisial avatar (1 kata)	getAvatarInitials()	user.nama_lengkap='Manajer'	"MA"	Fallback 2 huruf dari satu kata.	Lulus
6	Inisial default	getAvatarInitials()	user.nama_lengkap=null	"M"	Default untuk role manajer.	Lulus
7	Warna avatar	getAvatarColor()	-	"bg-blue-500"	Warna khusus role manajer.	Lulus
8	Format no HP (awalan 0)	getFormattedPhoneAttribute()	user.no_hp='08123456789'	" +628123456789"	0 → +62 (format internasional).	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
9	Format no HP (tanpa 0)	getFormattedPhoneAttribute()	user.no_hp='628123456789'	"628123456789"	Tidak diubah jika sudah internasional.	Lulus
10	Nomor HP kosong	getFormattedPhoneAttribute()	user.no_hp=null	""	Kembalikan string kosong.	Lulus
11	URL foto profil	getProfilPhoto()	user.profil_photo='avatars/manajer.jpg'	http://localhost/storage/avatars/manajer.jpg	Mock helper asset() di namespace model.	Lulus
12	Foto profil null	getProfilPhoto()	user.profil_photo=null	null	Tidak ada foto → null.	Lulus
13	Foto default manajer	getManajerPhoto()	user.profil_photo=null	http://localhost/images/manajerL.png	Gunakan gambar default.	Lulus
14	Foto dari user	getManajerPhoto()	user.profil_photo='profil/m.png'	http://localhost/storage/profil/m.png	Prioritas foto user.	Lulus
15	Greeting	getManajerGreeting()	-	Mengandung "Selamat"	Salam dinamis berdasar waktu.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
16	Periode kosong	getPeriodeFormatted()	periode_awal=null; periode_akhir=null	"Tidak ada periode"	Teks fallback ketika periode tak diset.	Lulus
17	Format periode (DD/MM/YY)	getPeriodeFormatted()	2025-01-01 s.d 2025-01-31	"01/01/2025 - 31/01/2025"	Set dateFormat via reflection agar tanpa DB.	Lulus
18	Format periode Indonesia	getPeriodeFormattedIndonesia()	2025-01-01 s.d 2025-01-31	"01 Jan 2025 - 31 Jan 2025"	Varian label bulan Indonesia.	Lulus
19	Durasi periode (hari)	getPeriodeDuration()	2025-01-01 s.d 2025-01-31	31.0	Perhitungan diffInDays inklusif.	Lulus
20	Status periode relatif waktu	isPeriodeActive/Expired/Upcoming()	now=2025-01-15; periode Jan 2025	Active=true; Expired/Upcoming=false	Gunakan Carbon::setTestNow untuk deterministik.	Lulus
21	Status aktif manajer	isActive()	status_manajer='aktif'	True	Cek simple flag status_manajer.	Lulus
22	Status non	isNonActive()	status_manajer='nonaktif'	True	Kebalikan dari aktif.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	aktif manajer					
23	Label status	getStatusLabelAttribute()	status_manajer='aktif'	"Aktif"	Normalisasi label untuk UI.	Lulus
24	Warna status	getStatusColorAttribute()	status_manajer='aktif'	"text-green-600 bg-green-100"	Mapping status → kelas Tailwind.	Lulus
25	Scope user aktif	scopeActive()	Builder mock	Builder sama (chainable)	whereHas('user', where is_active=true).	Lulus
26	Scope by status	scopeByStatus()	Builder mock + 'aktif'	Builder sama	where('status_manajer','aktif').	Lulus
27	Scope active/nonactive periode	scopeActivePeriod(), scopeNonActivePeriod()	Builder mock	Builder sama	Kedua scope memanggil where status_manajer sesuai.	Lulus
1	Nama lengkap manajer	getFullName()	user.nama_lengkap='Manajer Utama'	"Manajer Utama"	Ambil nama dari relasi user yang diset manual.	Lulus

d. Penyewaan

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status pending	isPending()	status_penyewaan='pending'→'confirmed'	True, lalu False	Validasi state awal booking.	Lulus
2	Status reserved	isReserved()	status_penyewaan='reserved'→'pending'	True, lalu False	Reserved biasanya sesudah DP.	Lulus
3	Status confirmed	isConfirmed()	'confirmed'→'pending'	True, lalu False	Booking terkonfirmasi.	Lulus
4	Status active	isActive()	'active'→'pending'	True, lalu False	Sesi sedang berlangsung.	Lulus
5	Status completed	isCompleted()	'completed'→'pending'	True, lalu False	Penyewaan selesai.	Lulus
6	Status cancelled	isCancelled()	'cancelled'→'pending'	True, lalu False	Pembatalan oleh user/admin.	Lulus
7	Status refunded	isRefunded()	'refunded'→'pending'	True, lalu False	Dana telah dikembalikan.	Lulus
8	Sudah dibayar ?	isPaid()	status: reserved/confirmed/active/completed/pending	True kecuali pending	Status tertentu dianggap sudah ada transaksi.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
9	Perlu DP	needsDP()	metode_pembayaran='dp_50'→'lunas'	True, lalu False	Menentukan skema pembayaran.	Lulus
10	Alasan pembatalan (format)	getFormattedCancellationReason()	alasan_status='Customer membatalkan'→null	"Customer membatalkan"→"-"	Normalisasi tampilan alasan.	Lulus
11	Alasan status (format)	getFormattedStatusReason()	alasan_status='Pembayaran diterima'→null	"Pembayaran diterima"→"-"	Idem untuk alasan umum.	Lulus
12	Hitung DP 50%	calculateDP()	payment_method='dp_50'; total_harga=200000	nominal_dp=100000; sisa=100000	Mengisi nominal_dp & sisa_pembayaran.	Lulus
13	Hitung Lunas	calculateDP()	payment_method='lunas'; total=200000	nominal_dp=200000; sisa=0	Skema tanpa DP.	Lulus
14	Total terbayar	totalPaid()	mock_pembayaran_verified sum=250000	250000	Agregasi hanya status verified.	Lulus
15	Ada pembayaran?	hasPayment()	sum=1000 dan 0	True; lalu False	Cek eksistensi pembayaran bermakna.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
16	Layak refund?	canBeRefunded()	status reserved/confirmed/active + hasPayment	True, True, lalu False	Hanya sebelum aktif dan ada pembayaran.	Lulus
17	Kadaluarsa DP	isExpired()	reserved; batas_pembayaran_dp lewat/masa depan/null	True/False/False	Menggunakan batas waktu DP.	Lulus
18	Lunas penuh	isFullyPaid()	total_harga=200000; sum verified=250000	True	Terbayar $\geq$ total.	Lulus
19	Payment complete	isPaymentComplete()	mock isFullyPaid()=true	True	Delegasi ke isFullyPaid().	Lulus
20	Remaining payment: refund approved	getRemainingPayment()	refundRequest.status='approved'	0	Refund penuh menutup kewajiban.	Lulus
21	Remaining payment	getRemainingPayment()	ada pembayaran status 'refunded'	0	Jika ada transaksi	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	t: ada refunded				refunded, sisa 0.	
22	Remaining payment: pakai uploaded	getRemainingPayment()	latest uploaded jumlah_bayar=120000; total=500000	380000	Pakai pembayaran uploaded terbaru bila ada.	Lulus
23	Remaining payment: fallback totalPaid	getRemainingPayment()	totalPaid=200000; total=500000	300000	Jika tidak ada uploaded, gunakan totalPaid.	Lulus
24	Batal booking	cancel()	mock update status/alasan/cancelled_at	Update terpenggil	Menetapkan payload pembatalan lengkap.	Lulus
25	Update status booking	updateStatus()	status='active', alasan='ok'	Update terpenggil	Mutasi status terkontrol.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
26	Overdue DP & pelunasan	isDPOverdue()/isPelunasanOverdue()	batas lewat/belum	True/False per kondisi	Gunakan waktu kini (Carbon).	Lulus
27	Set waktu sesi (per_sesi)	setSessionTime()	layanan per_sesi durasi=90; jam_mulai='09:00:00'	jam_mulai='09:00:00'; jam_selesai='10:30:00'	Menghitung jam selesai dari durasi sesi.	Lulus
28	Set waktu hari penuh	setSessionTime()	layanan per_hari 08:00–17:00	jam_mulai=08:00; jam_selesai=17:00	Mengikuti jam operasional layanan.	Lulus
29	Format tanggal batal/status	getFormattedCancellationDate()/getFormattedStatusUpdateDate()	cancelled_at 2025-01-02 09:15; updated_at 2025-01-03 10:30	"02 Jan 2025 09:15"; "03 Jan 2025 10:30"	Format Indonesia konsisten untuk UI.	Lulus

e. Pembayaran

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status pending	isPending()	status_pembayaran='pending' lalu 'lunas'	True, lalu False	Validasi status awal sebelum ada bukti pembayaran.	Lulus
2	Status uploaded	isUploaded()	status_pembayaran='uploaded' lalu 'pending'	True, lalu False	Tanda bukti transfer sudah diunggah; menunggu verifikasi.	Lulus
3	Status verified	isVerified()	status_pembayaran='verified' lalu 'pending'	True, lalu False	Setelah admin verifikasi, pembayaran sah.	Lulus
4	Status ditolak	isDitolak()	status_pembayaran='ditolak' lalu 'pending'	True, lalu False	Bukti tidak valid; akan ada tindak lanjut.	Lulus
5	Kadaluarsa	isExpired()	status='ditolak'; batas_waktu_transfer lewat/belum	True jika lewat, False jika belum	Menggunakan tanggal batas untuk menentukan expired.	Lulus
6	Status refunded	isRefunded()	status='refunded' lalu 'pending'	True, lalu False	Dana telah dikembalikan (final).	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
7	Jenis DP	isDP()	jenis_pembayara n='dp' lalu 'pelunasan'	True, lalu False	Identifikasi uang muka.	Lulus
8	Jenis pelunasan	isPelunasan()	jenis_pembayara n='pelunasan' lalu 'dp'	True, lalu False	Pembayaran sisa setelah DP.	Lulus
9	Jenis penuh	isPenuh()	jenis_pembayara n='penuh' lalu 'dp'	True, lalu False	Bayar 100% tanpa DP.	Lulus
10	Label status	getStatusDisplayAttribute()	variasi status + kondisi batas waktu	"Menunggu Pembayaran", "Menunggu Verifikasi", "Verified DP", "Ditolak", "Kadaluarsa"	Mapping status ke label UI; khusus ditolak+lewat waktu → "Kadaluarsa".	Lulus
11	Label jenis bayar	getJenisPembayaranDisplayAttribute()	'dp','pelunasan',' penuh'	"DP 50%","Pelunasan","Bayar Penuh"	Normalisasi teks untuk tampilan.	Lulus
12	Boleh diverifikasi	canBeVerified()	status 'uploaded','pending','lunas'	True hanya untuk 'uploaded'	Tombol verifikasi hanya muncul jika bukti sudah diunggah.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
13	Boleh ditolak	canBeRejected()	status 'uploaded','pending','lunas'	True hanya untuk 'uploaded'	Penolakan hanya relevan setelah ada bukti.	Lulus
14	Bisa expired	canBeExpired()	status 'pending','uploaded','lunas','ditolak'	True untuk pending/uploaded; False lainnya	Auto-expire hanya untuk yang belum final.	Lulus
15	Generate kode default	generateKodePembayaran()	-	Prefix 'BYR-PAY' + tanggal	Kode unik berisi tanggal (Ymd) untuk tracking.	Lulus
15	Generate kode DP	generateKodePembayaran('dp')	-	Prefix 'BYR-DP' + tanggal	Membedakan jenis dari kode.	Lulus
15	Generate kode pelunasan	generateKodePembayaran('pelunasan')	-	Prefix 'BYR-PELUNASAN' + tanggal	Idem.	Lulus
15	Generate kode penuh	generateKodePembayaran('penuh')	-	Prefix 'BYR-PENUH' + tanggal	Idem.	Lulus
16	Tandai uploaded	markAsUploaded()	mock update status + tanggal_transfer	Update terpanggil	Menyimpan waktu upload untuk audit trail.	Lulus

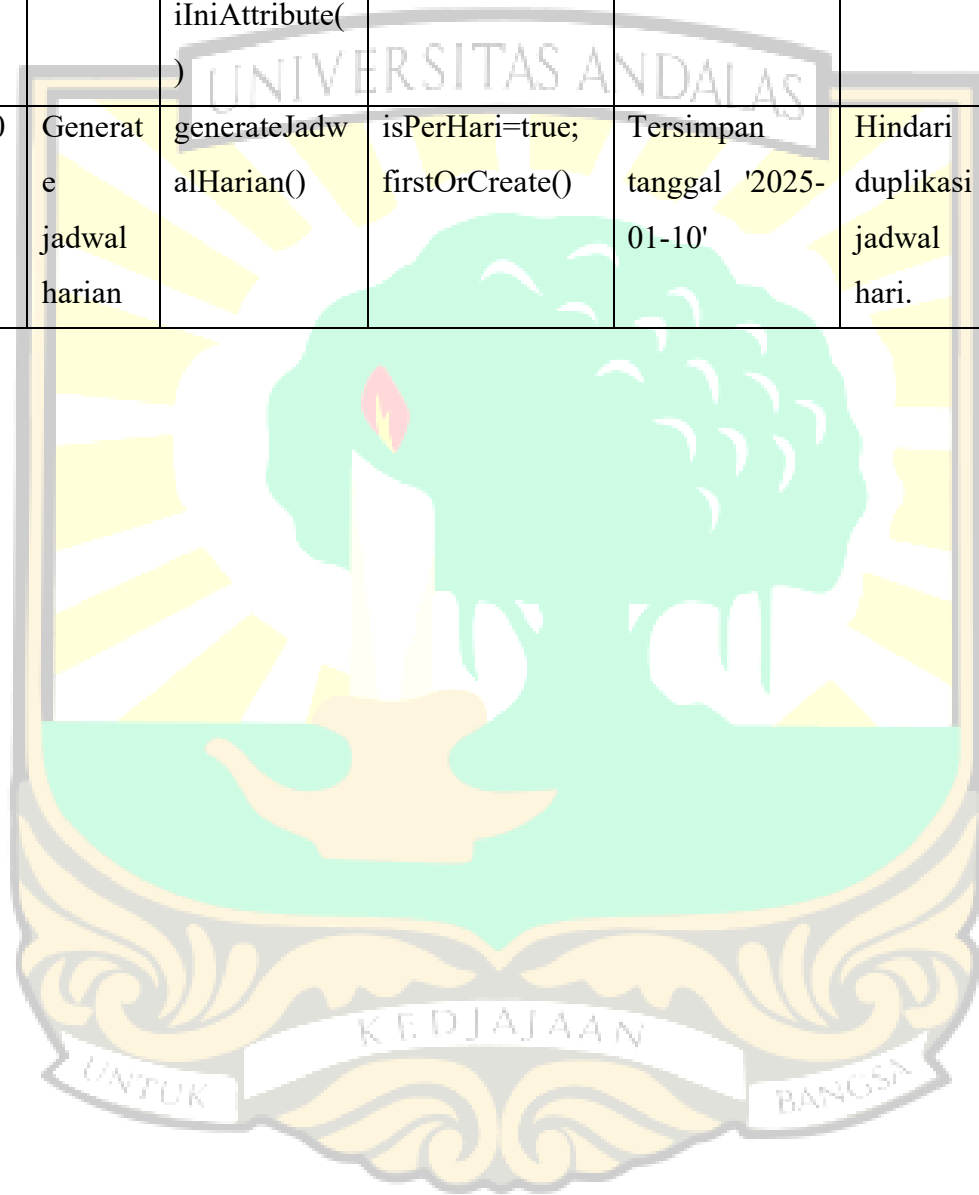
f. Layanan

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status aktif layanan	isActive()	status='aktif' lalu 'nonaktif'	True, lalu False	Validasi flag status operasional layanan.	Lulus
2	Tipe sewa per hari	isPerHari()	tipe_sewa='per_hari' lalu 'per_sesi'	True, lalu False	Membedakan mode perhitungan/p enjadwalan.	Lulus
3	Tipe sewa per sesi	isPerSesi()	tipe_sewa='per_sesi' lalu 'per_hari'	True, lalu False	Kebalikan dari per_hari.	Lulus
4	Area outdoor	isOutdoor()	area='outdoor' lalu 'indoor'	True, lalu False	Kategori lokasi mempengaruhi tampilan/filter	Lulus
5	Area indoor	isIndoor()	area='indoor' lalu 'outdoor'	True, lalu False	Kebalikan dari outdoor.	Lulus
6	Label area	getAreaTextAttribute()	area='outdoor', 'indoor'	"Outdoor", "Indoor"	Normalisasi label untuk UI.	Lulus
7	Durasi sesi jam	getDurasiSesiJamAttribute()	120, 90, null	2.0, 1.5, null	Konversi menit→jam desimal; handle null.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
8	Format harga dasar	getFormattedHargaDasarAttribute()	100000; 1500000	"Rp 100.000"; "Rp 1.500.000"	Format Rupiah sesuai standar ID.	Lulus
9	Badge status	getStatusBadgeAttribute()	'aktif','nonaktif','unknown'	Kelas hijau/merah/abu	Mapping status→kelas Tailwind untuk UI.	Lulus
10	Badge tipe sewa	getTipeSewaBadgeAttribute()	'per_hari','per_sesi'	Kelas biru/ungu	Diferensiasi visual tipe sewa.	Lulus
11	Info perhitungan sesi	getPerhitunganSesiAttribute()	per_sesi; durasi=90; jam 08-17; max_sesi_per_hari=5	total=540, muat=6, aktif=5, jam="08:00 - 17:00"	Hitung menit operasional, jumlah sesi muat, dan pembatasan by setting.	Lulus
12	Max sesi yang muat	getMaxSesiYangMuatAttribute()	per_sesi; durasi=60; 09-12	3.0	180/60; validasi aritmetika slot.	Lulus
13	Sesi per hari	getSesiPerHariAttribute()	7 lalu null	7 lalu null	Ambil nilai setting apa adanya.	Lulus
14	Jam sesi text	getJamSesiTextAttribute()	07:30-15:45	"07:30 - 15:45"	Format rentang jam untuk UI.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
15	Generate jadwal sesi	generateJadwalSesi()	per_sesi; durasi=120; 08-14; max=10; tgl '2025-01-02'	3 slot; pertama 08:00-10:00	Bangun slot berurutan sepanjang jam operasional.	Lulus
16	Statistik rating/feedback/pemesanan	averageRating(), totalFeedbacks(), totalPemesanan(), getRatingAttribute()	Mock relasi penyewaan	4.2; 12; 27; rating attr fallback 0	Agregasi via relasi; fallback null→0.	Lulus
17	Total pendapatan	totalPendapatan()	Mock whereHas(pembayaran verified)->sum(total_harga)	1500000	Hanya hitung penyewaan dengan pembayaran verified.	Lulus
18	Relasi fasilitas aktif	fasilitasAktif()	Mock fasilitasTambah()	Return relasi	Akses relasi untuk daftar fasilitas layanan.	Lulus
19	Statistik jadwal	getTotalJadwalAttribute(), getJadwalTersediaCountAttribute(),	Mock relasi jadwal*	8; 5; list hari ini (1 item)	Hitung total/tersedia/hari-ini via relasi.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
		getJadwalHar iIniAttribute()				
20	Generat e jadwal harian	generateJadw alHarian()	isPerHari=true; firstOrCreate()	Tersimpan tanggal '2025- 01-10'	Hindari duplikasi jadwal hari. per	Lulus



g. Refund

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status refund diminta	isRefundRequested()	status_refund='refund_requested' → 'approved'	True, lalu False	Memastikan flag awal pengajuan refund terdeteksi sebelum diproses admin.	Lulus
2	Status approved	isApproved()	status_refund='approved' → 'pending'	True, lalu False	Menandai refund yang sudah disetujui admin sebagai final state.	Lulus
3	Badge status requested	getStatusBadgeAttribute()	status_refund='refund_requested'	Mengandung 'bg-yellow-100' dan 'Menunggu Persetujuan'	Mapping status ke badge Tailwind + teks untuk UI (pending/review).	Lulus
4	Format jumlah refund	getFormattedJumlahRefundAttribute()	100000; 1500000	"Rp 100.000"; "Rp 1.500.000"	Format Rupiah titik ribuan untuk konsistensi	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					tampilan laporan.	
5	Approve refund	approve()	mock update({status='approved', tanggal_approved})	Berhasil update	Validasi side-effect perubahan status dan timestamp approval.	Lulus
6	Badge default saat status tak dikenal	getStatusBadgeAttribute()	'unknown_status'	Badge kuning + "Menunggu Persetujuan"	Fallback aman ketika nilai status tidak valid/baru.	Lulus
7	Format jumlah 0	getFormattedJumlahRefundAttribute()	0	"Rp 0"	Edge case nol tetap diformat jelas, tanpa error.	Lulus
8	Pembulatan desimal	getFormattedJumlahRefundAttribute()	100000.50	"Rp 100.001"	Nilai desimal dibulatkan (ceiling) agar akuntabel.	Lulus
9	Approved false untuk	isApproved()	'refund_requester'	False	Proteksi logika agar hanya 'approved'	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	status lain				yang dianggap disetujui.	
10	Format tanggal dibuat	getFormatted TanggalAttribute()	created_at=2025-01-02 14:30	"02/01/2025 14:30"	Format Indonesia dd/mm/YYYY HH:mm untuk dokumen/riwayat.	Lulus
11	Scope requested	scopeRefundRequested()	Builder mock	Kembali builder setelah where('status_refund','refund_requested')	Enkapsulasi filter query untuk list pending.	Lulus
12	Scope approved	scopeApproved()	Builder mock	Kembali builder setelah where('status_refund','approved')	Enkapsulasi filter query untuk list approved.	Lulus

h. Notification

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status sudah dibaca	isRead()	is_read=true lalu false	True, lalu False	Pemeriksaan flag boolean yang menandai notifikasi sudah dibuka user.	Lulus
2	Status belum dibaca	isUnread()	is_read=false lalu true	True, lalu False	Kebalikan dari isRead(); memudahkan filter unread.	Lulus
3	Ikon tipe pembayaran	getNotificationIcon()	tipe='pembayaran'	'fas fa-credit-card'	Mapping tipe→ikon Font Awesome untuk visual di UI.	Lulus
4	Ikon tipe penyewaan	getNotificationIcon()	tipe='penyewaan'	'fas fa-calendar-check'	Ikon kalender cek untuk event booking.	Lulus
5	Ikon tipe refund	getNotificationIcon()	tipe='refund'	'fas fa-undo'	Ikon undo merepresentasikan pengembalian dana.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
6	Ikon tipe informasi	getNotificationIcon()	type='informasi'	'fas fa-info-circle'	Ikon info untuk pesan general.	Lulus
7	Ikon tipe feedback	getNotificationIcon()	type='feedback'	'fas fa-comment'	Ikon komentar untuk ulasan/feedback.	Lulus
8	Ikon default	getNotificationIcon()	type='unknown'	'fas fa-bell'	Fallback aman saat tipe baru/tidak dikenali.	Lulus
9	Warna tipe pembayaran	getNotificationColor()	type='pembayaran'	'bg-blue-50 border-blue-200'	Warna tema biru untuk kategori pembayaran.	Lulus
10	Warna tipe penyewaan	getNotificationColor()	type='penyewaan'	'bg-green-50 border-green-200'	Warna hijau untuk status booking.	Lulus
11	Warna tipe refund	getNotificationColor()	type='refund'	'bg-orange-50 border-orange-200'	Orange sebagai perhatian/alert.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
12	Warna tipe informasi	getNotificationColor()	type='informasi'	'bg-gray-50 border-gray-200'	Netral/abu untuk informasi umum.	Lulus
13	Warna tipe feedback	getNotificationColor()	type='feedback'	'bg-purple-50 border-purple-200'	Ungu untuk diferensiasi feedback.	Lulus
14	Warna default	getNotificationColor()	type='unknown'	'bg-gray-50 border-gray-200'	Fallback konsisten saat tipe tak dikenali.	Lulus
15	Tandai sebagai dibaca	markAsRead()	mock update(['is_read' =>true])	Berhasil memanggil update	Memastikan perubahan status unread→read dilakukan via update.	Lulus

i. Jadwal Libur

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Cek jenis harian	isHarian()	jenis='harian' lalu 'tanggal_tertentu'	True, lalu False	Validasi enum jenis. True hanya untuk nilai 'harian'; memastikan pembedaan tipe jadwal.	Lulus
2	Cek jenis tanggal tertentu	isTanggalTerentu()	jenis='tanggal_tertentu' lalu 'harian'	True, lalu False	Kebalikan dari isHarian; mengunci logika dua mode jadwal libur.	Lulus
3	Label jenis: Harian	getJenisTextAttribute()	jenis='harian'	"Harian"	Accessor mengubah nilai enum menjadi label user-friendly untuk UI.	Lulus
4	Label jenis: Tanggal Tertentu	getJenisTextAttribute()	jenis='tanggal_tertentu'	"Tanggal Tertentu"	Memastikan mapping enum → label konsisten di tampilan.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
4b	Default label saat jenis tak dikenal	getJenisTextAttribute()	jenis='abc'	"Tanggal Tertentu"	Fallback aman jika data tidak valid/baru; UI tetap terbaca.	Lulus
5	Konversi nama hari	getHariTextAttribute()	hari='senin'..'minggu'	"Senin".."Minggu"	Normalisasi title case untuk semua 7 hari agar konsisten di UI/filtering.	Lulus
6	Hari tak dikenal	getHariTextAttribute()	hari='unknown'	"unknown"	Tidak memaksa mapping jika nilai tidak dikenal; memudahkan debugging/data import.	Lulus
7	Konstanta daftar hari	HARI_VALUES	-	['senin','selasa','rabu','kamis','jumat','sabtu','minggu']	Menjamin sumber kebenaran untuk validasi input/dropdown.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
8	Scope jadwal harian	scopeHarian()	Builder mock	Kembali builder setelah where('jenis','harian')	Memastikan filter query di level DB, memungkinkan chaining efisien.	Lulus
9	Scope tanggal tertentu	scopeTanggal Tertentu()	Builder mock	Kembali builder setelah where('jenis','tanggal_tertentu')	Konsistensi scope untuk query reusable pada laporan/fitur kalender.	Lulus

j. Fasilitas Tambahan

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Total pendapatan fasilitas	totalPendapatan()	id_fasilitas_tambahan='FT-1'; mock Penyewaan: :whereNotNull('fasilitas_tambahan_detail')->get() → 3 item: (1) {FT-1:10000, FT-2:5000}, (2) {FT-1:15000}, (3) null	25000	Mengakumulasi pendapatan untuk satu fasilitas berdasarkan id milik instance. Langkah: (a) ambil semua penyewaan dengan fasilitas_tambahan_detail != null; (b) iterasi; (c) jika ada key FT-1, tambah harga ke total; (d) abaikan entry null/fasilitas lain. Total: 10000 + 15000 = 25000. Pemanggilan statik Penyewaan dimock via alias, jadi tanpa DB.	lulus

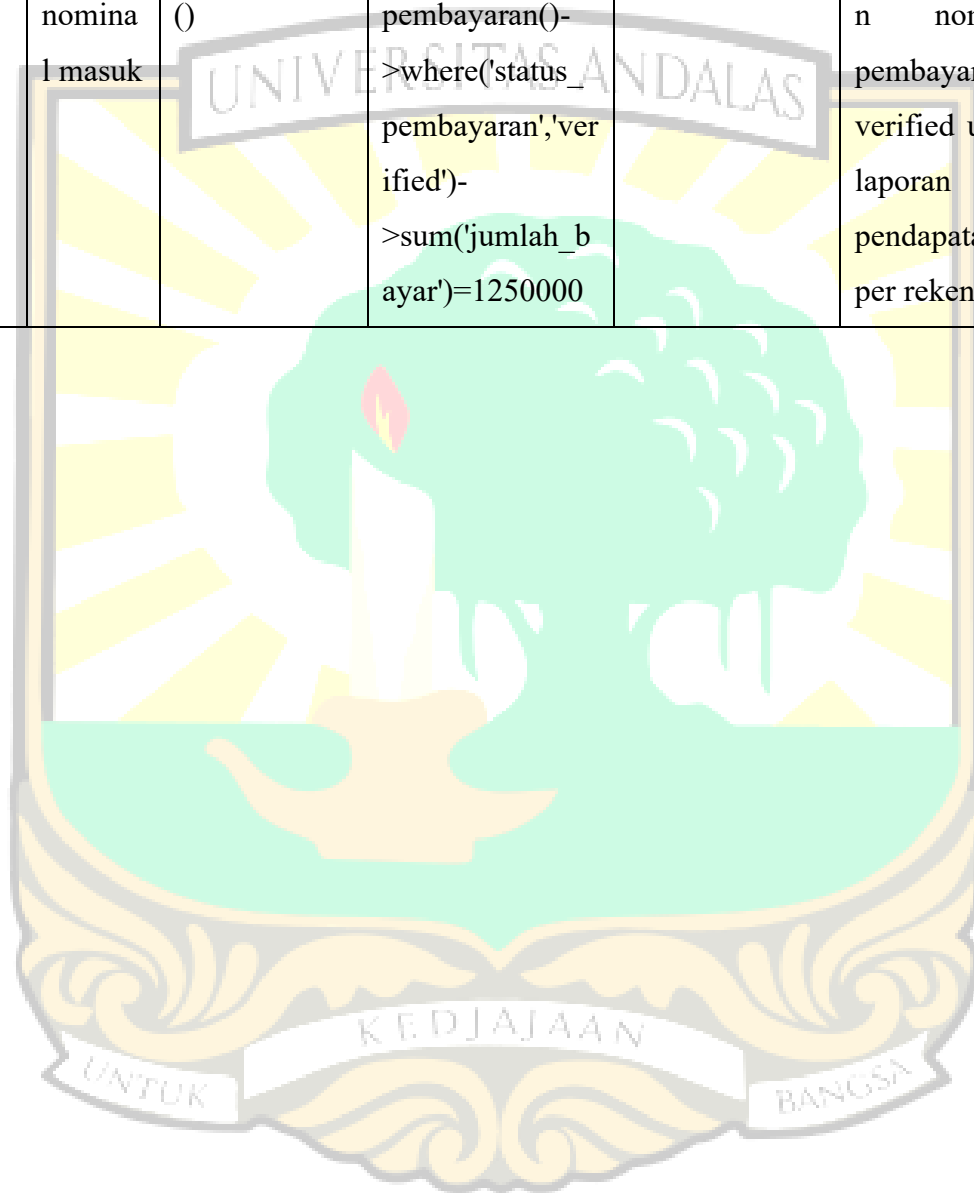
k. ChatbotConversation

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Deteksi dari tamu	isFromGuest())	id_user=null lalu id_user='user-1'	True saat null, False saat ada	Cek dua kondisi	Lulus
2	Deteksi dari user	isFromUser()	id_user='user-1' lalu id_user=null	True saat ada, False saat null	Cek dua kondisi	Lulus
3	from user saat null	isFromUser()	id_user=null	False	Kasus eksplisit	Lulus
4	from guest saat ada	isFromGuest())	id_user='user-1'	False	Kasus eksplisit	Lulus

1. Rekening

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Status rekening aktif	isActive()	status='aktif' lalu 'nonaktif'	True, lalu False	Validasi flag aktif/non aktif untuk menentukan bisa dipakai sebagai tujuan transfer.	Lulus
2	Format info bank	getFormatBankAttribute()	nama_bank='BCA', nomor_rekening='1234567890', nama_pemilik='Koperasi Hydro Utama'	"BCA - 1234567890 (Koperasi Hydro Utama)"	Helper untuk menampilkan nama bank + nomor + pemilik dalam satu string rapi untuk UI/instruksi transfer.	Lulus
3	Total transaksi terverifikasi	totalTransaksi()	mock relasi pembayaran()->where('status_pembayaran','verified')->count()=7	7	Menghitung jumlah pembayaran berstatus verified yang masuk ke rekening ini. Menggunakan mock agar tanpa DB.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
4	Total nominal masuk	totalNominal()	mock pembayaran()->where('status_pembayaran','verified')->sum('jumlah_bayar')=1250000	1250000	Menjumlahkan nominal pembayaran verified untuk laporan pendapatan per rekening.	Lulus



2. Helper

a. DateHelper

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Format tanggal Indonesia	formatIndonesia()	2025-08-21 15:45:00; includeTime: false/true	"21 Agustus 2025"; string yang mengandung "21 Agustus 2025"	Memformat tanggal ke lokal ID dengan nama bulan penuh; jika includeTime=true menambahkan jam-menit.	Lulus
2	Format ringkas/numerik/DB/penuh	formatShort(), formatNumeric(), formatDatabase(), formatFull()	"2025-08-21 10:00:00"	"21 Aug 2025"; "21/08/2025"; "2025-08-21"; string memuat "2025"	Kumpulan utilitas untuk tampilan ringkas, angka (dd/mm/YYYY), untuk penyimpanan DB (YYYY-MM-DD), dan format lengkap.	Lulus
3	Format khusus report	formatReport(), formatInvoice()	"2025-08-21 10:00:00"	"21 Agustus 2025"; "21/08/2025"	Preset untuk laporan (Indonesia) dan invoice	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	& invoice				(numeric) agar konsisten di dokumen resmi.	
4	Penanganan null/kosong	semua formatter	null / "	"-" untuk semua kecuali formatDatabase() → null	Menghindari error/format aneh saat data tanggal belum ada; database formatter tetap mengembalikan null untuk keperluan query.	Lulus

b. FormaterHelper

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Format rupiah	formatCurrency()	0; 100000; 1500000	"Rp 0"; "Rp 100.000"; "Rp 1.500.000"	Memformat angka ke Rupiah dengan pemisah ribuan titik; cocok untuk	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					tampilan harga.	
2	Format tanggal	formatDate(); formatDateTime()	"2025-01-02"; "2025-01-02 14:30:00"	"02 January 2025"; "02 January 2025, 14:30" (atau custom "d/m/Y")	Utilitas serbaguna untuk tanggal saja dan tanggal+jam, mendukung format kustom.	Lulus
3	Badge status (kelas CSS)	getStatusBadge()	'proses','selesai','batal','unknown'	'bg-yellow-100 text-yellow-800'; 'bg-green-100 text-green-800'; 'bg-red-100 text-red-800'; 'bg-gray-100 text-gray-800'	Mapping status → kelas Tailwind untuk konsistensi warna di UI.	Lulus
4	HTML bintang rating	getStars()	3	String HTML berisi 5 ikon bintang (3 filled)	Menghasilkan markup dengan total 5 bintang; jumlah ikon diverifikasi	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					via count 'fa-star'.	
5	Warna layanan deterministik	getLayananColor()	"Lapangan A" dipanggil 2x	Nilai warna sama pada kedua panggilan	Menjamin warna konsisten per nama layanan untuk chart/legend UI.	Lulus



3. Service

a. RefundService

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Full refund jika dibatalkan koperasi	calculateRefundAmount()	penyewaan.cancelled_by=admin ; total=200000	200000	Override rule: pembatalan oleh admin selalu 100% refund; pembayaran/DP tidak mengurangi.	Lulus
2	Persentase & status refund by waktu	getRefundPercentage(), getRefundStatus()	H-8, H-5, H-2	100%/"Full Refund"; 50%/"Partial Refund"; 0%/"Tidak Ada Refund"	Aturan waktu: $\geq$ H-7 full; H-4.6 50%; $\leq$ H-3 0; waktu dibekukan via Carbon::setTestNow.	Lulus
3	Alasan penolakan refund	getRefundRejectionReason()	lewat tanggal; $\leq$ H-3; status tak memenuhi	String alasan relevan	Beri pesan "sudah lewat", " $\leq$ 3 hari", atau "Status penyewaan ...".	Lulus
4	Hitung nilai refund by	calculateRefundAmount()	H-7, H-5, H-2; total=200000	200000; 100000; 0	Mengalikan total dengan persentase	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	aturan waktu				dari aturan waktu.	
5	Kelayakan refund komprehensif	canBeRefunded()	Dibatalkan admin (true); dibatalkan customer H-2 (false)	True; False	Admin override true, sedangkan customer tunduk pada batas waktu ($\geq H-4$) dan status.	Lulus
6	Ringkasan info refund	getRefundInfo()	H-7; total=200000	{can_refund:true, amount:200000, percentage:100, status:'Full Refund'}	Satu API merangkum semua info untuk UI/keputusan.	Lulus

b. NotificationService

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Notifikasi sukses pembayaran (tanpa	sendPaymentSuccess()	Pembayaran dengan relasi: penyewaan.layanan.nama_layanan,	Objek notifikasi dibuat (customer + semua admin)	Cek duplikat via where(...)->whereDate(..)->first(); jika null, create 1	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	duplikasi)		penyewaan.cust omer.id_user		untuk customer + N admin (User::where('role','admin')) ; seluruh call dimock alias.	
2	Notif sukses pembayaran (duplikat)	sendPayment Success()	Pembayaran sama, existing notification ditemukan	Notifikasi existing di-update, tidak membuat baru	where(...)->first() mengembalikan object dengan update(); create() tidak boleh terpanggil.	Lulus
3	Notif pembayaran expired	sendPayment Expired()	Pembayaran yang melewati batas	Objek notifikasi dibuat	Notifikasi tipe informasi ke customer; create() dimock.	Lulus
4	Notif DP diterima	sendDpAccepted()	Pembayaran DP; ambil 2 admin	Tercipta 3 notifikasi (1 customer + 2 admin)	Pesan memuat kode pembayaran/sewa, sisa pembayaran,	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					batas pelunasan; prevent-duplicate ringan.	
5	Notif ditolak (variasi tindak lanjut)	sendPaymentRejected()	Tindak lanjut: upload_ulang, refund_only, atau null	Judul "Pembayaran Ditolak", isi pesan sesuai tindak lanjut	Tiga skenario memvalidasi konten pesan yang tepat, termasuk alasan jika ada.	Lulus
6	Notif pelunasan diterima	sendPartialPaymentAccepted()	Pembayaran pelunasan	3 notifikasi (customer + admin)	Mirip DP diterima, untuk pelunasan/sisa pembayaran.	Lulus
7	Reminder DP	sendDPReminder()	Penyewaan punya pembayaran jenis 'dp'	Objek notifikasi dibuat	Jika tidak ada dp: kembalikan null; relasi penyewaan dipermudah dengan stub.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
8	Reminder pelunasan	sendFinalPaymentReminder()	Penyewaan milik customer	Objek notifikasi dibuat	Pengingat final payment; create() dimock.	Lulus
9	Hitung unread	getUnreadCount()	id_user	Angka unread (int)	Chain where(...)->count() dimock; mengembalikan 3.	Lulus
10	Tandai satu notifikasi read	markAsRead()	id_notif	Notifikasi ter-update	Notification::find()->update(['is_read'=>true]) dimock; return instance.	Lulus
11	Tandai semua read	markAllAsRead()	id_user	Jumlah baris diupdate	where(...)->update(['is_read'=>true]) mengembalikan 1.	Lulus

c. Pdf Service

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Generate struk pembayaran	generatePaymentReceipt()	Penyewaan stub + Pembayaran	Array {pdf, filename, content}	DomPDF facade dimock: loadView()->setPaper()->output()='%PDF-FAKE%'; filename mengandung prefix "Struk_".	Lulus
2	Simpan struk pembayaran	savePaymentReceipt()	Sama	Path diawali "receipts/"	Storage::put() dimock; memastikan penyimpanan di folder receipts.	Lulus
3	Attachement email	getPaymentReceiptAttachment()	Sama	{content, filename, mime='application/pdf'}	Struktur siap kirim email/response download.	Lulus
4	Generate struk DP	generateDPReceipt()	Sama	Array {pdf, filename, content}	Versi khusus DP; struktur sama, konten template DP.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
5	Simpan struk DP	saveDPReceipt()	Sama	Path diawali "receipts/dp/"	Penyimpanan tersegmentasi untuk DP.	Lulus



4. Controller

a. Akun

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Buat akun manajer baru	storeManajer()	Admin login; email unik; periode awal-akhir	{success:true}	Validasi; cek email unik (aktif); nonaktifkan manajer lama; create User role manajer + profil Manajer; hash password; kirim ManajerAccountMail; semua via alias mock.	Lulus
2	Reset password manajer	resetManajerPassword()	Admin; target user role manajer	{success:true}	Hash password baru; update user; proteksi role admin-only.	Lulus
3	Update status customer	updateCustomerStatus()	Admin; is_active=true/false	{success:true}	Validasi admin; update flag aktif; log error bila gagal.	Lulus
4	Reset password admin	resetAdminPassword()	Admin; target role admin	{success:true}	Mirip reset manajer; hanya untuk admin.	Lulus
5	Statistik user	getStats()	-	Objektif metrik	Serangkaian where(...)->count()	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
				total/aktif/inaktif/verified/harian/bulanan	alias-mock pada User.	
6	Aksi massal user	bulkAction()	action: activate/deactivate/delete; user_ids[]	{success:true}	Activate/deactivate: update flag; Delete: hanya customer tanpa penyewaan; hindari hapus diri sendiri/admin.	Lulus

b. BookingAvailability

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Cek keterseediaan per hari	checkPerHariAvailability()	Layanan per_hari; tanggal	available=true	Mock Penyewaan:: where/join mengembalikan null (tidak bentrok area maupun booking aktif).	Lulus
2	Hitung slot per sesi	getAvailableSlots()	Layanan per_sesi 08–10; 60 menit	total_slots=2; available_slots=1	Where()->first() dibuat booked untuk slot pertama;	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					slot kedua kosong.	
3	Keterse- diaan bulanan	getMonthlyA- vailability()	id_layanan, bulan, tahun	{success:true, availability:...}	Alias Layanan/Jadw- alLibur/Penye- waan untuk cek libur dan bentrok per tanggal.	Lulus

c. *Dashboard*

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Routing <i>dashbo- ard</i> admin + hitung statistik	index()	Auth user role=a- dmin	Method tersedia dan siap hitung stats	Dimock: Pembayaran (sum=5.000.000, count=15), Layanan (count=8, list), User (customers=50, manajers=5), Penyewaan (avg rating=4.75, feedback baru=3), Notification (unread=5), RefundRequest (pending=2), ChatbotConversation (count=120). Verifikasi	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					method privat adminDashboard ada.	
2	Data revenue bulanan (12 bulan)	getRevenueData(Request)	Request.get('year') default 2024	Method publik menerima 1 parameter Request	Pembayaran::where()->whereYear()->whereMonth()->sum() dimock pakai array [1.000.000..1.600.000]; setiap bulan return nilai berbeda secara siklik. Verifikasi signature dan akses publik.	Lulus
3	Daftar method publik	index, getRevenueData	-	Keduanya ada	Cek via method_exists.	Lulus
4	Daftar method privat	adminDashboard, manajer Dashboard, customer Dashboard	-	Ada dan private	Cek via ReflectionClass -> isPrivate().	Lulus
5	Inheritance controller	extends Controller	-	Parent class 'App\Http\Controllers\Controller'	Verifikasi get_parent_class().	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
6	Validasi tipe parameter	getRevenueData(Request)	-	Param pertama bertipe Illuminate\Http\Request	Cek tipe parameter via refleksi.	Lulus

d. Feedback

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Statistik feedback global	getStatistics()	-	Rata-rata rating 4.75; total 150	Hitung pada Penyewaan yang punya rating: whereNotNull('rating')->avg('rating') dan count(). Dipakai untuk ringkasan performa layanan.	Lulus
2	Listing feedback + pagination	getFeedbacks()	Filter opsional (layanan, rating, tanggal, search)	Koleksi terpaginate berisi feedback terformat	Query chain: with(customer, layanan), whereNotNull('rating'), whereHas(...),	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					orderBy(...), lalu paginate(). Item dipetakan (rating, komentar, tanggal, relasi).	
3	Balasan admin ke feedback	adminReply()	id penyewaan, balasan_admin	Field balasan_admin dan replied_at terupdate	Validasi admin, ambil penyewaan ber-rating via findOrFail, update kolom balasan dan timestamp.	Lulus
4	Keterse- diaan method publik	index, getStatistics, getFeedbacks , getServices, getServiceRa- ting, adminIndex, adminShow, adminReply,	-	Semua method ada	Verifikasi eksistensi method publik untuk seluruh peran (admin/manaj- er/customer).	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
		manajerIndex , manajerShow , manajerDownload, customerIndex, customerCreate, customerStore				
5	Inheritance Controller	extends Controller	-	Parent = App\Http\Controllers\Controller	Memastikan arsitektur mengikuti base controller.	Lulus
6	Signature getFeedbacks	getFeedbacks (Request)	-	Param 1 bertipe Request	Cek type-hint dan jumlah parameter via refleksi.	Lulus
7	Signature adminReply	adminReply(Request, id)	-	Param 1 Request, param 2 bernama 'id'	Verifikasi method menerima 2 parameter dengan	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
12	Daftar feedback admin	adminIndex()	Filter/pencarian opsional	Paginator list feedback	Query with(...), filter by layanan/customer, orderBy, paginate()->withQueryString().	Lulus
13	Detail feedback admin	adminShow()	id penyewaan	Data feedback + relasi	with(customer, layanan)->whereNotNull('rating')->findOrFail(id).	Lulus
14	Daftar feedback customer	customerIndex()	Auth customer	List feedback milik user	Filter by id_customer dari Auth; gabung layanan untuk filter.	Lulus
15	Submit feedback customer	customerStore()	id_penyewaan, rating, komentar	Penyewaan terupdate (rating, komentar)	Validasi, cek kepemilikan (id_customer = Auth), update	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					rating/komentar.	

e. Jadwal

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Eksistensi method index	index()	-	Method publik tersedia	Cek via ReflectionClass bahwa index ada dan public.	Lulus
2	Eksistensi method edit	edit()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
3	Eksistensi method store	store()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
4	Eksistensi method update	update()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
5	Eksistensi method destroy	destroy()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
6	Eksistensi method calendar	calendar()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
7	Eksistensi method getJadwalData	getJadwalData()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
8	Eksistensi method checkAvailability	checkAvailability()	-	Method publik tersedia	Verifikasi method exist + public.	Lulus
9	Method util privat	checkDuplikates()	-	Method ada dan private	Menjamin utilitas internal anti-duplikasi tersedia dan tidak diekspos publik.	Lulus
10	Inheritance Controller	extends Controller	-	Parent class = App\Http\Controllers\Controller	Menjamin mengikuti pola base controller Laravel.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
11	Tipe parameter index	index(Request)	-	Param 1 bertipe Illuminate\Http\Request	Verifikasi type-hint via refleksi.	Lulus
12	Tipe parameter edit	edit(JadwalLibur)	-	Param 1 bertipe App\Models\JadwalLibur	Verifikasi type-hint via refleksi.	Lulus
13	Tipe parameter store	store(Request)	-	Param 1 bertipe Request	Verifikasi type-hint via refleksi.	Lulus
14	Tipe parameter update	update(Request, JadwalLibur)	-	Param 1 Request, Param 2 JadwalLibur	Verifikasi type-hint via refleksi.	Lulus
15	Tipe parameter destroy	destroy(JadwalLibur)	-	Param 1 JadwalLibur	Verifikasi type-hint via refleksi.	Lulus
16	Tipe parameter calendar	calendar(Request)	-	Param 1 Request	Verifikasi type-hint via refleksi.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
17	Tipe parameter checkAvailability	checkAvailability(Layanan, date)	-	Param 1 Layanan; Param 2 bernama 'date'	Menjamin kontrak endpoint: layanan yang dicek + tanggal.	Lulus
18	Signature method privat	checkDuplicates()	-	Tanpa parameter, tanpa return type	Pemeriksaan bentuk util internal sesuai harapan.	Lulus
19	Import dependensi	File controller memuat import wajib	-	Semua import ditemukan	Dicek isi file: JadwalLibur, Layanan, Request, Log, Carbon terimport.	Lulus
20	Daftar method publik minimal	index, edit, store, update, destroy, calendar, getJadwalData, checkAvailability	-	Semua ada	Verifikasi satu per satu via method_exists.	Lulus

f. Layanan

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Routing index sesuai role	index()	Auth user role=admin	Terarah ke alur adminIndex (method tersedia)	Mock Auth; siapkan query Layanan (all/query/where/whereHas/paginate). Tujuan: halaman index untuk admin.	Lulus
2	Listing admin terfilter	adminIndex()	Request filter opsional	Paginator items sesuai koleksi	Chain: query()->where/whereHas()->orderBy()->paginate(); items diambil dari koleksi dummy.	Lulus
3	Buat layanan baru	store()	Payload valid + upload gambar utama	Layanan dibuat; file tersimpan	Mock Storage (disk/putFileAs/makeDirectory), DB transaction (begin/commit/rollback), Log warning/error; return sukses.	Lulus
4	Update layanan	update()	Payload valid (tanpa	Data layanan terbaru	Mock findOrFail()->update(); Storage+DB dimock; status diambil dari input.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
			upload)			
5	Toggle status layanan	toggleStatus()	id_layanan valid	Status berubah ('aktif' ↔ 'nonaktif')	Mock findOrFail()->update(); verifikasi method tersedia.	Lulus
6	Detail layanan (admin)	show()	id_layanan	Detail + hitung booking	Mock Penyewaan::where()->whereIn()->count() untuk statistik pemesanan.	Lulus
7	Detail layanan (customer)	customerShow()	id_layanan; tanggal opsional	Detail + data booking hari ini + daftar libur	Mock Penyewaan::where()->whereDate()->pluck(); JadwalLibur::where()->whereIn()->get().	Lulus
8	API slot tersedia	getAvailableSlotsApi()	layanan_id, tanggal	Daftar slot tersedia	Layanan per_sesi (jam buka/tutup, durasi). Penyewaan::where()->pluck() untuk slot terpakai.	Lulus
9	Statistik layanan	getStats()	-	total layanan, total pemesanan, pendapatan	Layanan::count(); Layanan::where()->sum(harga);	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					Penyewaan::whereHas()->count().	
10	Cek status booking cepat	checkBookingStatus()	id_layanan	count=0 (tidak bentrok)	Penyewaan::where()->whereDate()->whereIn()->count().	Lulus
11	Customer show by name	customerShowByName()	nama_layanan	Detail layanan ditemukan	Layanan::where()->first(); pengambilan data booking/libur sama seperti customerShow.	Lulus
12	Tanggal libur layanan	getHolidayDates()	id_layanan	List tanggal libur	JadwalLibur::where()->whereIn()->get() mengembalikan tanggal.	Lulus
13	Form edit	edit()	id_layanan	Method publik tersedia	findOrFail() dimock; hanya verifikasi eksistensi/aksesibilitas.	Lulus
14	Form create	create()	-	Method publik tersedia	Verifikasi eksistensi dan visibility public.	Lulus
15	Struktur method publik	Daftar public methods	-	Semua method publik ada	index, adminIndex, create, store, edit, update, toggleStatus, show, checkBookingStatus,	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					customerShow, customerShowByName, getAvailableSlotsApi, getHolidayDates, getStats.	
16	Method util privat	getActiveBookings, getUnpaidBookings, getAvailableTimeSlots	-	Ada dan private	Cek via ReflectionClass isPrivate() untuk ketiga helper internal.	Lulus
17	Inheritance Controller	extends Controller	-	Parent = App\Http\Controllers\Controller	Verifikasi get_parent_class().	Lulus

g. Notification

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Daftar notifikasi customer	customerIndex()	Auth id_user='USER1'	List notifikasi + terfilter paginate	Filter by id_user, urut terbaru; hitung unread. Query chain dimock.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
2	Tandai 1 notif dibaca	markAsRead()	id_notification='NOTIF1' milik user	JSON sukses; is_read=true	Cek ownership, set is_read=true, hapus cache counter.	Lulus
3	Tandai semua dibaca	markAllAsRead()	Auth id_user	Kembalikan jumlah baris diupdate	where(id_user)->where(is_read=false)->update(...); clear cache.	Lulus
4	Hapus notifikasi	destroy()	id_notification='NOTIF1' milik user	JSON sukses	Validasi kepemilikan; delete; hapus cache.	Lulus
5	Hitung unread	getUnreadCount()	Auth id_user	Integer (mis. 3)	where(id_user)->where(is_read=false)->count().	Lulus
6	Buat notifikasi + email	createNotification() (static)	{id_user, title, message, type}	Notifikasi tersimpan; email terkirim	Notification::create(); ambil email user; Mail::to()->send(); clear	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					cache; log error bila gagal.	
7	Struktur method publik	customerIndex, adminIndex, markAsRead, markAllAsRead, destroy, getUnreadCount, bulkAction	-	Semua tersedia	Verifikasi method_exists.	Lulus
8	Struktur method statis	createNotification, sendEmailNotification, sendBookingNotification, sendPaymentNotification, sendSystemNotification, sendFeedbackReminder, sendDailyReminders,	-	Ada dan static	Cek via Reflection bahwa method static tersedia.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
		sendCancellationNotification, sendRefundNotification				
9	Inheritance Controller	extends Controller	-	Parent = App\Http\Controllers\Controller	Menjamin arsitektur konsisten.	Lulus
10	Signature markAsRead	markAsRead(Request, id)	-	Param 1 Request; param 2 'id'	Verifikasi tipe dan nama parameter via refleksi.	Lulus

h. Pembayaran

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Listing pembayaran admin	index()	Filter opsional	Paginator items	Query: query()- >with('penyewa')- >where/where Has()- >orderBy()- >paginate();	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					verifikasi items.	
2	Verifikasi pembayaran	verify()	id_pembayaran='PAY-001', status='verified', catatan	Status pembayaran jadi verified; penyewaan ter-update bila perlu; notifikasi dibuat	Auth admin; Pembayaran::findOrFail(); Penyewaan::findOrFail(); Notification::create().	Lulus
3	Tolak pembayaran	reject()	id_pembayaran='PAY-001', alasan_penolakan	Status jadi 'rejected'; notifikasi dibuat	Auth admin; update status dengan alasan.	Lulus
4	Buat pelunasan akhir	pelunasan()	id_penyewaan='SEW-001' (sudah ada DP)	Rekam pembayaran sisa; notifikasi dibuat	Hitung kekurangan dari total_harga - sum(verified); create Pembayaran baru.	Lulus
5	Upload bukti pembayaran	customerUpload()	id_pembayaran='PAY-001'; file bukti	Simpan file; status → 'uploaded'	Storage::disk('')->putFileAs(); update path	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	customer				bukti dan status.	
6	Detail pembayaran	show()	id_pembayaran= 'PAY-001'	Data pembayaran + relasi	Pembayaran::findOrFail() dimock.	Lulus
7	Lihat bukti pembayaran	showBukti()	id_pembayaran= 'PAY-001'	URL storage bukti	Storage::disk()->exists(url()); kembalikan URL.	Lulus
8	Detail customer (own)	customerShow()	id_pembayaran milik CUST1	Data ditampilkan jika milik user	Auth id_user diperiksa terhadap penyewaan.id_customer.	Lulus
9	Struktur method publik	-	-	Semua method publik ada	index, show, showBukti, verify, reject, pelunasan, customerShow, customerUpload.	Lulus
10	Method util privat	-	-	Ada dan private	generateAndSendDPReceipt,	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					generateAndSendFinalReceipt, generateKodePembayaran.	

i. Penyewaan

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Riwayat penyewaan customer	customerHistory()	Auth id_user='CUST1'	Paginator berisi data milik customer	Filter by id_customer, with relasi, order, paginate.	Lulus
2	Create booking (validasi + create)	store()	Auth CUST1; id_layanan; tanggal; jam; jenis_pembayaran	Tercipta Penyewaan + Pembayaran awal; notifikasi dibuat	Layanan::findOrFail; Penyewaan::create; Pembayaran::create; Notification::create.	Lulus
3	Listing admin	adminIndex()	Filter pencarian opsional	Paginator items	query()->with()->where/where	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					Has()- >orderBy()- >paginate().	
4	Update status booking	updateStatus()	id_penyewaan; status baru; catatan	Status terbaru; notifikasi dibuat	findOrFail()- >update(); Notification::create.	Lulus
5	Ambil slot tersedia (per_sesi)	getAvailable Slots()	layanan_id; tanggal	Daftar slot tersedia	Layanan per_sesi; exclude jam yang sudah terpakai (pluck); cek libur.	Lulus
6	Cek keterseediaan (per_hari)	checkAvailability()	layanan_id; tanggal	available=true jika bebas bentrok	Hitung existing booking dan libur di tanggal tersebut.	Lulus
7	Get availability (per_hari)	getAvailability()	layanan_id; tanggal	Struktur hasil ketersediaan	Mirip checkAvailability dengan bentuk response terstruktur.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
8	Rincian sesi pada tanggal	getDateSesi onDetails()	layanan_id; tanggal	Detail slot sesi berdasarkan durasi	Hitung rentang slot dan exclude slot terpakai.	Lulus
9	Form pembuatan booking	create()	Auth CUST1	View data layanan tersedia	Layanan::where()->get().	Lulus
10	Peringatan booking belum bayar	showUnpaid BookingWarning()	Auth CUST1	List booking unpaid	whereIn(status) + with(relasi).	Lulus
11	Tampilkan halaman konfirmasi	showConfirm()	layanan_id; booking_date	View konfirmasi	Validasi request; Layanan::findOrFail.	Lulus
12	Konfirmasi booking (create + DP)	confirm()	layanan_id; booking_date; time_slot; metode_pembayaran;	Buat Penyewaan + Pembayaran (DP bila notifikasi	Cek konflik (where/join); create penyewaan + pembayaran;	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
			fasilitas_tambahan		Notification::create.	
13	Listing manajer	manajerIndex()	Filter opsional	Paginator items	query()->with()->where/whereHas()->paginate().	Lulus
14	Detail penyewaan	show()	id_penyewaan	Data + relasi	findOrFail().	Lulus
15	Detail penyewaan (ringkasan)	getPenyewaanDetail()	id_penyewaan	Object detail	findOrFail().	Lulus
16	Cek availability (alias)	getAvailability()	layanan_id; tanggal	available true/false	Validasi cepat untuk UI.	Lulus
17	Cek update status terbaru	checkUpdates()	Auth CUST1; last_check	List perubahan	whereIn(statuses)->get() setelah waktu last_check.	Lulus
18	Unduh laporan	manajerDownload()	rentang tanggal opsional	PDF dihasilkan	with()->filter rentang->orderBy()-	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	(manajer)				>get(); PDF dimock.	
19	Unduh laporan (admin)	adminDownload()	rentang tanggal opsional	PDF dihasilkan	Mirip manajerDownload untuk admin.	Lulus
20	Riwayat penyewaan admin	adminRiwayatPenyewaan()	Filter tahun/bulan/rentang	Paginator items	with()->whereBetween/year/month/filter->paginate().	Lulus
21	Unduh single penyewaan	downloadSinglePenyewaan()	id_penyewaan	PDF 1 penyewaan	findOrFail(); generate PDF dimock.	Lulus
22	Unduh penyewaan manajer (tahun)	manajerDownloadPenyewaan()	tahun	PDF daftar	with()->whereYear()->get().	Lulus
23	Unduh penyewaan manajer (range)	manajerDownloadPenyewaanRange()	start_date, end_date	PDF daftar	with()->whereBetween()->get().	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
24	Lihat bukti sewa	showBuktiSewa()	id_penyewaan	View bukti	findOrFail().	Lulus
25	Unduh bukti sewa	downloadBuktiSewa()	id_penyewaan	PDF bukti	findOrFail(); generate PDF dimock.	Lulus
26	Lihat struk DP	showDPReceipt()	id_penyewaan (ada pembayaran dp verified)	View struk DP	findOrFail(); filter pembayaran dp verified.	Lulus
27	Unduh struk DP	downloadDPReceipt()	id_penyewaan (ada pembayaran dp verified)	PDF struk DP	findOrFail(); generate PDF dimock.	Lulus
28	Struktur method publik	-	-	Semua method publik ada	Daftar 27+ method publik diverifikasi dengan method_exists .	Lulus
29	Method util privat	getTrulyUnpaidBookings, generateKode Pembayaran	-	Ada dan private	Cek via ReflectionClass isPrivate().	Lulus
30	Inheritance	extends Controller	-	Parent = App\Http\Cont	Konsistensi arsitektur.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	Controller			rollers\Controll er		

j. Profil

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Edit profil	edit()	Auth user	Method public tersedia	Menampilkan form edit sesuai role.	Lulus
2	Update profil	update(Request)	nama_lengkap, no_hp, alamat (+foto opsional)	Profil terupdate; URL foto bila ada	Storage dimock; Log info/error; tanpa upload pada test.	Lulus
3	Update password	updatePassword(Request)	current_password, password baru	Password terganti	Hash::check lalu Hash::make; update via User::where() ->update().	Lulus
4	Hapus foto profil	removeProfilePhoto()	-	Foto dihapus; kolom dibersihkan	Storage::exists/delete; update user.	Lulus
5	Tampilkan profil	show()	id user (owner)	Data profil	Validasi role; User::findOrFail().	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
6	Hapus akun	destroy(Request)	password + konfirmasi "DELETE"	Akun dinonaktifkan; logout	Verifikasi password; tidak ada pembayaran aktif; hapus foto; update user; logout.	Lulus
7	Persentase kelengkapan	getCompletionPercentage()	-	Persentase 0–100	Hitung dari field penting terisi.	Lulus
8	Upload avatar	uploadAvatar(Request)	file	Simpan ke storage; update avatar	Storage::disk()->url(); update kolom avatar.	Lulus
9	Struktur & inherit	Daftar method publik; extends Controller	-	Semua method ada; parent benar	Verifikasi method_exists dan get_parent_class.	Lulus

k. Refund

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Tampilkan popup form refund	showPopupForm()	id penyewaan/refund	Method public tersedia	Verifikasi eksistensi dan visibility; digunakan untuk memunculkan form pengajuan.	Lulus
2	Halaman daftar refund customer	customerIndex()	-	Method public tersedia	Endpoint listing refund milik customer; di sini hanya dicek keberadaan method.	Lulus
3	Submit pengajuan refund	submitRefundRequest()	Request	Method public tersedia dan menerima Request	Validasi signature: 1 parameter bertipe Request. Proses nyata (validasi, simpan, notifikasi)	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					diuji di integration.	
4	Daftar refund admin	index()	-	Method public tersedia	Endpoint listing refund untuk admin (review/approval).	Lulus
5	Detail refund	show()	id	Method public tersedia; param wajib 'id'	Digunakan untuk melihat detail satu refund request.	Lulus
6	Approval refund	approve()	Request, id	Method public tersedia; param 1 Request, param 2 'id'	Flow bisnis: set approved, kirim mail, update pembayaran/penyewaan; di unit test difokuskan pada kontrak method.	Lulus
7	Pewarisan controller	extends Controller	-	Parent = App\Http\Controllers\Controller	Menjamin konsistensi arsitektur.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
8	Signature method	showPopupForm(id), submitRefundRequest(Request), show(id), approve(Request, id)	-	Parameter dan tipe sesuai	Divalidasi via ReflectionClass.	Lulus
9	Import dependensi	Berbagai model/facade/service	-	Semua import wajib ditemukan	Dicek isi file: Request, Auth, Log, Mail, Str, model-model (Penyewaan, Pembayaran, RefundRequest, User, Admin, Notification), mailer (RefundRequestNotification, RefundCompletedCustomerMail), service	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					RefundCalculationService, dan NotificationController.	

1. Rekening

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Halaman pengaturan rekening	showSetting()	-	List rekening tersortir; ensure default	Ambil daftar dengan orderBy; jika belum ada default dapat dibuat.	Lulus
2	Update pengaturan (admin)	updateSetting()	Admin; data rekening	Rekening terupdate; default diatur benar	Nonaktifkan default lama via where()->update bila perlu; validasi input.	Lulus
3	Daftar method publik	showSetting, updateSetting, manajerIndex	-	Semua ada	Verifikasi method_exists.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
		manajerCreate, manajerStore, manajerEdit, manajerUpdate, manajerDestroy				
4	Inheritance	extends Controller	-	Parent = App\Http\Controllers\Controller	Konsistensi arsitektur.	Lulus
5	Daftar rekening manajer	manajerIndex()	-	List rekening	orderBy()->get(); siap untuk view manajer.	Lulus
6	Tambah rekening (manajer)	manajerStore()	Payload valid	Rekening dibuat; handle default flag	Jika is_default true, nonaktifkan yang lain; kirim notifikasi/email jika ada.	Lulus
7	Ubah rekening	manajerUpdate()	id + payload	Rekening terupdate; kirim email	Atur default; update field; email via	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
	(manajer)				Mail::to()->send().	
8	Hapus rekening (manajer)	manajerDestroy()	id	Rekening dihapus	Pastikan bukan default sebelum delete; kirim email jika perlu.	Lulus

m. SmartChatbot

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Endpoint chat tamu	guestChat(Request)	Request	Return type JsonResponse	Cek method exist, visibility, tipe parameter, tipe return.	Lulus
2	Endpoint chat customer	customerChat(Request)	Request	Return type JsonResponse	Verifikasi signature dan visibility.	Lulus
3	Method privat NLP	detectIntent()	message:string	Return string	Method ada dan private; parameter bertipe string.	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
4	Generate respons tamu	generateGuestResponse()	intent:string	Return string	Method private; parameter bertipe string.	Lulus
5	Generate respons customer	generateCustomerResponse()	intent:string, userId:string	Return string	Method private; 2 parameter bertipe benar.	Lulus
6	Login percakapan	logConversation()	request:Request, userMessage:string, botResponse:string	Return void	Method private; parameter/return type diverifikasi.	Lulus
7	Inheritance	extends Controller	-	Parent = App\Http\Controllers\Controller	Arsitektur konsisten.	Lulus
8	Import dependensi	Model+facade	-	Import ChatbotConversation, Fasilitas Tambahan, Layanan, Penyewaan, User; Log, Validator	Dicek isi file.	Lulus

n. Welcome

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
1	Halaman landing	index()	-	View dengan data ringkasan	Layanan aktif, testimoni terakhir, metrik (total penyewaan, pendapatan bulan ini, total user). Semua via mock.	Lulus
2	Statistik ringkas	getStatistics()	-	{layanan:8, penyewaan:150, pendapatan:5.000.000, user:50}	Aggregasi via Layanan/Penyewaan/User; whereHas/whereMonth/sum/count dimock.	Lulus
3	Detail layanan	getServiceDetails()	id_layanan	Data layanan lengkap	Layanan::findOrFail(); log error bila exception.	Lulus
4	Slot tersedia (public)	getAvailableSlots(Request)	layanan_id, tanggal	Daftar slot	Layanan per_sesi; exclude jam sudah dipesan; gunakan	Lulus

No	Fokus Pengujian	Metode yang Diuji	Input	Hasil yang Diharapkan	Keterangan	Status
					Penyewaan:: where()- >pluck().	
5	Struktur & inherit	index, getServiceDetails, getAvailableSlots, getStatistics, logConversation; extends Controller	-	Semua method ada; parent benar	Verifikasi eksistensi method dan inheritance.	Lulus

B. Pengujian Sistem

1. Manajer

Lembaran Pengujian Aplikasi

Nama Aplikasi : Sistem Informasi Jasa Penyewaan (Sewain)

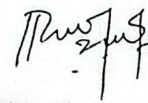
Nama Penguji : Rildawati

Hak Akses : Manager

NO	Fungsional	Skenario	Hasil Yang Diharapkan	Hasil	
				Sesuai	Tidak Sesuai
1	Login web	Mengisi <i>email</i> dan <i>password</i> valid, lalu klik login	Sistem menampilkan halaman dashboard	✓	
2	Melihat Dashboard	Melihat halaman dashboard manager	Sistem menampilkan halaman dashboard manager dan ringkasan data penyewaan	✓	
3	Edit Profile	Mengubah data pada form profil (kecuali email) dan klik simpan	Sistem menyimpan perubahan	✓	
4	Logout	Klik opsi logout	Sistem mengarahkan kembali ke halaman welcome	✓	
5	Monitor penyewaan	Klik menu Penyewaan pada menu	Sistem menampilkan daftar penyewaan	✓	
		Pilih salah satu transaksi pelanggan untuk melihat detail	Sistem menampilkan data transaksi yang lengkap	✓	
		Klik download laporan	Sistem menampilkan laporan dalam format	✓	

			tabel yang bisa diunduh		
6	Monitor feedback	Klik menu feedback	Sistem menampilkan semua feedback pelanggan beserta status (dibalas/belum)	✓	

Bukittingi, 12 Agustus 2025



Rildawati



2. Admin

Lembaran Pengujian Aplikasi

Nama Aplikasi : Sistem Informasi Jasa Penyewaan (Sewain)

Nama Penguji : Kadaryati Ningsih

Hak Akses : Admin

NO	Fungsional	Skenario	Hasil Yang Diharapkan	Hasil	
				Sesuai	Tidak Sesuai
1	Login web	Mengisi <i>email</i> dan <i>password</i> valid, lalu klik login	Sistem menampilkan halaman dashboard	✓	
2	Edit Profile	Mengubah data pada form profil (kecuali <i>email</i>) dan klik simpan	Sistem menyimpan perubahan	✓	
3	Logout	Klik opsi logout	Sistem mengarahkan kembali ke halaman welcome	✓	
4	Mengelola Layanan	Klik tambah layanan, isi form lengkap, simpan	Sistem menyimpan data layanan baru	✓	
		Pilih layanan, ubah data, simpan	Sistem menyimpan perubahan	✓	
		Pilih layanan, klik hapus, konfirmasi	Sistem menghapus data layanan dari database	✓	
		Klik salah satu untuk liat detail layanan	Sistem menampilkan data lengkap layanan	✓	

5	Mengelola Jadwal Libur	Tambah	Isi data jadwal libur, simpan	✓	
		Edit	Ubah data jadwal libur, simpan	✓	
		Hapus	Hapus data jadwal libur	✓	
		Lihat Detail	Klik detail jadwal libur	✓	
6	Memverifikasi Pembayaran	Pilih transaksi, klik verifikasi	Sistem mengubah status transaksi menjadi terverifikasi	✓	
7	Mengelola Refund	Pilih transaksi refund, proses pengembalian dana	Sistem menandai status refund berhasil	✓	
8	Mengelola Akun (manager dan pelanggan)	Klik tombol tambah akun, isi form (nama, email, password, periode jabatan), klik simpan	Sistem memperbarui database akun sesuai aksi	✓	
		Pilih akun Manager, klik reset password	Sistem mengirim password baru ke email Manager	✓	
		Pilih akun pelanggan yang nonaktif, klik aktifkan	Sistem mengubah status akun menjadi aktif	✓	

		Pilih akun pelanggan, klik nonaktifkan	Sistem mengubah status akun menjadi nonaktif	✓	
9	Lihat dan balas feedback	Klik menu feedback, balas pesan	Sistem menampilkan pesan sebagai "dibalas"	✓	
10	Melihat Riwayat penyewaan	Klik menu riwayat penyewaan	Sistem menampilkan daftar riwayat lengkap	✓	

Bukittingi, 12 Agustus 2025



Kadaryati Ningsih



3. Pelanggan

Lembaran Pengujian Aplikasi

Nama Aplikasi : Sistem Informasi Jasa Penyewaan (Sewain)

Nama Penguji : Nadini Annisa Byant

Hak Akses : Pelanggan

NO	Fungsional	Skenario	Hasil Yang Diharapkan	Hasil	
				Sesuai	Tidak Sesuai
	Registrasi Akun	Mengisi form registrasi dengan data lengkap dan valid	Sistem membuat akun baru dengan status menunggu verifikasi	✓	
	Login web	Mengisi <i>email</i> dan <i>password</i> yang valid	Sistem menampilkan halaman dashboard pelanggan	✓	
	Edit Profile	Mengubah data profil (kecuali email), klik simpan	Sistem menyimpan perubahan	✓	
	Logout	Klik opsi logout	Sistem mengarahkan kembali ke halaman welcome	✓	
	Melihat Layanan	Klik menu layanan pada dashboard pelanggan	Sistem menampilkan daftar layanan yang tersedia beserta detail harga dan jadwal	✓	
	Pemesanan Layanan	melakukan pemesanan,	Sistem menyimpan pemesanan dan	✓	

		pilih layanan, tanggal, dan metode pembayaran	menampilkan ringkasan konfirmasi		
	Melakukan pembayaran dan Upload Bukti Pembayaran	Mengisi data untuk pembayaran dan mengunggah bukti pembayaran setelah transfer	Sistem menyimpan bukti pembayaran dan memperbarui status transaksi menjadi "Menunggu Verifikasi"	✓	
	Melihat Status Pemesanan	Klik menu riwayat pemesanan	Sistem menampilkan daftar pemesanan beserta status (Menunggu, Disetujui, Ditolak, Selesai)	✓	
	Chatbot	Mengakses chatbot untuk menanyakan informasi layanan atau status pemesanan	Sistem memberikan jawaban sesuai database secara otomatis	✓	
	Memberikan Feedback	Mengisi form feedback setelah layanan selesai digunakan	Sistem menyimpan feedback dan menampilkannya pada dashboard admin/manager	✓	

Padang, 12 Agustus 2025



Nadini Annisa Byant



LAMPIRAN H

(Dokumen)

1. Hasil Wawancara

BERITA ACARA WAWANCARA

Pada hari Kamis, 23 Januari 2025 telah dilaksanakan wawancara oleh:

1. Pewawancara

Nama : Syadza Intan Benya

NIM : 2111521011

2. Yang diwawancara

Nama : Rildawati

Jabatan : Manager Koperasi Konsumen Hydro Utama Bukittinggi

Hasil Wawancara

1.	Pertanyaan	Sebagai langkah awal, mohon perkenan Ibu untuk menjelaskan secara singkat mengenai unit-unit usaha yang saat ini dikelola oleh Koperasi Konsumen Hydro Utama Bukittinggi
	Jawaban	Saat ini, Koperasi Konsumen Hydro Utama Bukittinggi mengelola empat unit usaha utama, yaitu: 1. Unit Simpan Pinjam, menyediakan pembiayaan dan pinjaman bagi anggota koperasi 2. Unit Usaha Toko, melayani kebutuhan fotokopi, penjiilidan, dan perlengkapan arsip 3. Unit Pengadaan Barang dan Jasa, menangani proyek internal dari PT PLN Indonesia Power, seperti pengadaan ATK, material, dan sewa komputer 4. Unit Penyewaan, menyewakan fasilitas seperti gedung pertemuan, lapangan golf, tempat prewedding, dan penginapan
2.	Pertanyaan	Dari beberapa unit usaha tersebut, apakah terdapat yang masih mengalami kendala dalam proses operasional, terutama yang masih dijalankan secara manual dan belum terintegrasi sistem digital?
	Jawaban	Ya, masih terdapat beberapa unit usaha yang menghadapi kendala operasional karena belum terintegrasi dengan sistem digital. Salah satunya adalah unit penyewaan fasilitas seperti gedung pertemuan, lapangan golf, tempat prewedding. Proses reservasi

		masih dilakukan secara manual, baik langsung maupun lewat telepon dan dicatat menggunakan buku atau Excel. ini berpotensi berisiko menimbulkan kesalahan pencatatan, jadwal tumpang tindih, serta menyulitkan pemantauan ketersediaan fasilitas. Selain itu, pelanggan juga belum dapat mengakses informasi secara real-time maupun melakukan reservasi dan pembayaran secara digital, sehingga menghambat efisiensi layanan
3.	Pertanyaan	Secara khusus, apakah unit usaha penyewaan termasuk yang membutuhkan prioritas pengembangan sistem digital? Jika ya, mohon dijelaskan alasannya.
	Jawaban	Benar, unit penyewaan sangat memerlukan sistem digital karena proses reservasi dan pembayaran saat ini masih manual. Dengan adanya sistem digital, pelayanan akan menjadi lebih efisien dan akses informasi bagi pelanggan lebih mudah
4.	Pertanyaan	Bagaimana alur penyewaan layanan di koperasi saat ini? Apakah masih dilakukan secara manual atau sudah menggunakan sistem tertentu?
	Jawaban	Saat ini, proses penyewaan masih dilakukan secara manual. Dimana pelanggan perlu menghubungi PIC secara langsung melalui WhatsApp atau datang ke kantor Koperasi di Jl. Sentral KM 1 Padang Luar Kec. Banuhampu Kab. Agam untuk reservasi
5.	Pertanyaan	Apa saja kendala utama yang dihadapi koperasi dalam pengelolaan layanan penyewaan?
	Jawaban	Beberapa kendala utama yang dihadapi dalam pengelolaan layanan penyewaan ini di antaranya adalah 1. Proses penyewaan yang masih manual, jadi anggota harus menghubungi PIC langsung atau datang ke kantor. 2. Selain itu, sistem pembayaran juga masih tunai, belum bisa transfer atau digital. 3. Koperasi juga belum punya sistem yang bisa membantu mengatur data dan jadwal penyewaan secara digital, jadi pengelolaannya masih dilakukan secara manual.

6.	Pertanyaan	Apakah terdapat perbedaan harga untuk setiap jenis layanan, misalnya berdasarkan waktu, durasi, atau kategori pelanggan?
	Jawaban	Ya, ada perbedaan harga untuk setiap jenis layanan. Misalnya, tarif untuk sewa lapangan golf, gedung, atau prewedding memang berbeda sesuai dengan jenis layanannya. Tapi, untuk saat ini, tarif tersebut masih bersifat tetap dan belum dibedakan lagi berdasarkan durasi penggunaan atau kategori pelanggan
7.	Pertanyaan	Apa saja syarat dan ketentuan bagi pelanggan yang ingin menyewa layanan, seperti lapangan golf, tempat prewedding, ruang rapat, dan lainnya?
	Jawaban	Setiap layanan memiliki syarat dan ketentuan masing-masing, yang secara umum, ketentuan penyewaan mencakup, - Reservasi dilakukan terlebih dahulu melalui PIC atau langsung ke kantor koperasi. - Pembayaran dilakukan di muka atau sesuai kesepakatan sebelum penggunaan layanan. - Penyewa bertanggung jawab atas kebersihan dan keamanan fasilitas selama masa sewa. - Waktu penggunaan disesuaikan dengan kesepakatan awal, baik untuk sewa harian maupun jam tertentu. - Untuk tempat prewedding, penyewa wajib membawa perlengkapan sendiri (kamera, lighting, dll). - Untuk sewa gedung, penggunaan berlaku untuk satu hari penuh sesuai waktu yang disepakati.
8.	Pertanyaan	Apakah ada jam operasional khusus untuk setiap layanan?
	Jawaban	Ya, sebagian besar layanan dibuka setiap hari mulai pukul 07.00 hingga 17.00 WIB, kecuali hari Jumat tutup.
9.	Pertanyaan	Apakah koperasi menyediakan fasilitas tambahan seperti katering, dekorasi, atau peralatan pendukung lainnya untuk mendukung kegiatan penyewaan?
	Jawaban	Ya, koperasi menyediakan beberapa fasilitas tambahan, tergantung jenis layanan yang disewa. Misalnya: 1. Untuk lapangan golf, tersedia fasilitas tambahan seperti:

		- Green Fee (9 hole): Rp250.000 - Biaya Caddy: Rp100.000 - Driving Range: Rp35.000 (5 bola) / Rp70.000 (100 bola) Namun, untuk kebutuhan lain seperti katering, dekorasi, dokumentasi, atau peralatan tambahan (sound system, kursi, tenda, dan sebagainya), umumnya pelanggan mengelola secara mandiri atau membicarakannya langsung dengan pengelola. Beberapa layanan bisa difasilitasi berdasarkan permintaan khusus dan ketersediaan, sehingga perlu dikonfirmasi saat proses pemesanan.
10.	Pertanyaan	Bagaimana ketentuan pembayaran untuk layanan penyewaan? Apakah pelanggan harus membayar penuh di awal atau cukup memberikan uang muka (DP)?
	Jawaban	Untuk ketentuan pembayaran, pelanggan diwajibkan melakukan pembayaran penuh di awal sebelum kegiatan atau layanan penyewaan berlangsung. Hal ini diterapkan untuk memastikan adanya komitmen dari pihak penyewa dan agar proses administrasi dapat segera diselesaikan. Pembayaran saat ini masih dilakukan secara manual, melalui transfer atau langsung ke kantor koperasi
11.	Pertanyaan	Apakah koperasi memiliki kebijakan terkait pembatalan penyewaan atau pengembalian dana? Jika ya, bagaimana ketentuannya?
	Jawaban	Jika pembatalan dilakukan sebelum pembayaran, tidak dikenakan biaya. Namun, jika sudah melakukan pembayaran, dana yang telah dibayarkan tidak dapat dikembalikan
12.	Pertanyaan	Seberapa sering koperasi memperbarui informasi layanan, seperti harga maupun ketersediaan?
	Jawaban	Saat ini, koperasi belum memiliki jadwal rutin dalam memperbarui informasi layanan seperti harga dan ketersediaan. Namun, pembaruan dilakukan secara berkala sesuai kebutuhan atau ketika terdapat perubahan signifikan

13.	Pertanyaan	Jika sistem digital akan dikembangkan, data apa saja yang menurut Ibu perlu dicatat dari pelanggan? (Contoh: nama, nomor telepon, jenis layanan, dll.)
	Jawaban	Jika sistem digital akan dikembangkan, data yang perlu dicatat dari pelanggan mencakup: - Nama lengkap, untuk identifikasi pelanggan. - Nomor telepon/WhatsApp yang aktif, sebagai sarana komunikasi dan konfirmasi pemesanan. - Jenis layanan yang disewa, untuk mencatat transaksi dan kebutuhan pelanggan. - Tanggal dan waktu penyewaan, penting untuk pengaturan jadwal layanan dan ketersediaan. - Jumlah atau durasi sewa, menyesuaikan dengan tarif dan alokasi sumber daya. - Ulasan atau penilaian pelanggan, sebagai bahan evaluasi untuk peningkatan layanan. Data ini akan sangat berguna untuk pengelolaan layanan yang lebih efisien dan untuk membangun relasi jangka panjang dengan pelanggan.
14.	Pertanyaan	Siapa saja yang direncanakan akan menjadi pengguna dari sistem tersebut? Apakah hanya admin koperasi atau juga pelanggan dan pihak lainnya?
	Jawaban	Pengguna utama dari sistem yang direncanakan adalah admin koperasi, yang akan bertanggung jawab atas pengelolaan data layanan, reservasi, dan laporan operasional. Selain itu, sistem juga akan dibuka aksesnya bagi pelanggan, untuk melihat info layanan dan melakukan pemesanan mandiri, serta pihak manajemen koperasi atau pengelola fasilitas untuk keperluan monitoring dan evaluasi.
15.	Pertanyaan	Apa harapan koperasi terhadap sistem digital yang akan dikembangkan? Fitur atau layanan apa saja yang dianggap penting untuk diakomodasi dalam sistem tersebut?
	Jawaban	Koperasi berharap sistem digital yang dikembangkan dapat:



		- Mempermudah proses pemesanan layanan secara online, - Mendukung pembayaran secara online, dan - Menyediakan fitur ulasan atau penilaian dari pelanggan untuk bahan evaluasi dan peningkatan kualitas pelayanan.
--	--	---

Pewawancara

Manager Koperasi Konsumen Hydro Utama



Syadza Intan Benya



Rildawati

Berita wawancara lanjutan

16	Pertanyaan	Bagaimana dengan metode pembayaran yang berlaku di koperasi? Apakah semua pelanggan diwajibkan membayar penuh di awal?
	Jawaban	Saat ini masih dengan tunai. Koperasi menerapkan dua metode pembayaran, yaitu pembayaran penuh (100%) dan pembayaran uang muka sebesar 50% (DP). Pembayaran penuh biasanya diterapkan untuk layanan jangka pendek seperti penyewaan harian atau per jam, sedangkan pembayaran 50% digunakan untuk layanan yang memerlukan persiapan seperti penyewaan gedung atau lapangan. Sisa pembayaran dilunasi maksimal satu hari sebelum kegiatan berlangsung.
17	Pertanyaan	Apakah koperasi memiliki ketentuan atau aturan khusus terkait pengembalian dana (refund) jika terjadi pembatalan penyewaan?

	Jawaban	Saat ini belum ada, tetapi untuk dalam jangka panjang koperasi memiliki kebijakan refund yang diatur berdasarkan waktu pembatalan. Jika pembatalan dilakukan minimal H-7 sebelum tanggal penyewaan, pelanggan berhak mendapatkan pengembalian sebesar 100%. Jika dilakukan antara H-4 hingga H-6, pengembalian yang diberikan sebesar 50%. Namun, jika pembatalan dilakukan kurang dari H-4, maka dana tidak dapat dikembalikan (0%). Apabila pembatalan dilakukan oleh pihak koperasi atau admin karena alasan internal, maka pelanggan akan mendapatkan refund penuh (100%).
18	Pertanyaan	Apakah sistem digital yang akan dikembangkan nantinya juga akan mencakup fitur pembayaran DP dan refund otomatis?
	Jawaban	Ya, dalam rencana pengembangan sistem digital, koperasi berharap fitur transaksi dapat mencakup pembayaran uang muka (DP), pelunasan, serta proses refund. Tujuannya agar seluruh transaksi lebih transparan, akurat, dan dapat dipantau secara langsung oleh admin maupun pelanggan.

2. Dokumen

KWITANSI PEMBAYARAN

No : /KKHU/XII/2024

Terima dari : Gita Pratiwi
 Terbilang : Lima ratus ribu rupiah
 Untuk Pembayaran : Sewalapangan golf u/ acara photo prewedding

Rp. 500.000,-

Bukittinggi, 04 Desember 2024



KOPERASI KONSUMEN HYDRO UTAMA
SADAN HUKUM NO. 1576/SH-XVII
 TANGGAL 24 FEBRUARI TH. 1996

No : KK 1010

BUKTI PENGELUARAN KAS

DIBAYARKAN KEPADA : PT. PLN IP UBP BUKITTINGGI

Tunai	Rp.	500.000	Jumlah dalam huruf Lima ratus ribu rupiah
Cek	Rp.		
Jumlah	Rp.	500.000	

Untuk : Pembayaran biaya prewedding Hardianggie Syakira Laua

Bkt. 12/01/25
u/dibayarkan

AHMAD AFANDI **RILDAWATI** **KELVIN JUNIRO**
Manajer

Diisi oleh Bagian Pembukuan	Nomor Perkiraan	Debet	Kredit
Tgl. Pembukuan :			Rp 500.000
Hal Buku Harian :			
Paraf :			

KOPERASI KONSUMEN HYDRO UTAMA
SADAN HUKUM NO. 1576/SH-XVII
 TANGGAL 24 FEBRUARI TH. 1996

No : KK 1910

BUKTI PENGELUARAN KAS

DIBAYARKAN KEPADA : PT. PLN IP UBP BUKITTINGGI

Tunai	Rp.	500.000	Jumlah dalam huruf Lima ratus ribu rupiah
Cek	Rp.		
Jumlah	Rp.	500.000	

Untuk : Pembayaran biaya prewedding Wendri Okta Firmando

Bkt. 12/01/25
u/dibayarkan

AHMAD AFANDI **RILDAWATI** **KELVIN JUNIRO**
Manajer

Diisi oleh Bagian Pembukuan	Nomor Perkiraan	Debet	Kredit
Tgl. Pembukuan :			Rp 500.000
Hal Buku Harian :			
Paraf :			



Batang Agam Golf

Fasilitas & Pelayanan di Lapangan Golf :

- ☑ 9 Hole
- ☑ Batang Agam Resto, kamar bilas
- ☑ Musholla
- ☑ Buka setiap hari jam 07.00 wib s.d 17.00 wib kecuali hari Jum'at - TUTUP

Tarif

Green Fee :

☑ 9 Hole - Rp. 250.000,-

☑ Biaya Caddy - Rp. 100.000,-

Driving :

☑ 5 bola - Rp. 35.000,-

☑ 100 bola - Rp. 70.000,-

Reservasi Hubungi :
Sunarto ☎ 0813-7478-7275



Batang Agam Mini Garden

☺ PREWEDDING PACKAGE

Price , Only :
Rp. 1 Juta per seasion

- ☑ Perangkat disediakan sendiri
- ☑ Buka setiap hari jam 07.00 wib s.d 17.00 wib kecuali hari Jum'at - TUTUP



Reservasi Hubungi :
Sunarto ☎ 0813-7478-7275



Syadza TA

Laporan TA_Syadza Intan Benya_2111521011.pdf



My Files



My Files



Universitas Andalas

Document Details

Submission ID

trn:oid::3618:119940324

Submission Date

Nov 5, 2025, 12:36 PM GMT+7

Download Date

Nov 5, 2025, 12:50 PM GMT+7

File Name

Laporan TA_Syadza Intan Benya_2111521011.pdf

File Size

16.7 MB

396 Pages

43,424 Words

294,527 Characters

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Top Sources

- 9%  Internet sources
- 6%  Publications
- 15%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.