

LAPORAN KULIAH KERJA

**Perancangan Sistem Informasi Manajemen Bengkel AC Berbasis Web (Studi Kasus
Bengkel AC Adi Jaya Teknik)**

Diajukan untuk memenuhi sebagian persyaratan
menyelesaikan mata kuliah Kerja Program Studi
Teknik Informatika



Disusun Oleh :

Riski keliolan

312210213

Laurensius rivaldo

nahak

3123107277

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PELITA BANGSA

BEKASI

2025

KATA PENGANTAR

Dengan memanjangkan puji dan syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Karunia-Nya sehingga penulis dapat menyelesaikan Laporan Kuliah dengan judul **“Perancangan Sistem Informasi Manajemen Bengkel AC Berbasis Web (Studi Kasus Bengkel AC Adi Jaya Teknik)”** secara khusus disusun dengan maksud untuk memenuhi syarat kelulusan

Program Studi Teknik Informatika Universitas Pelita Bangsa.

Dalam penulisan laporan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, maka dengan rendah hati penulis menerima semua kritik dan saran dari semua pihak demi penyempurnaan laporan ini di kemudian hari.

Penulis,

Tugas kelompok

DAFTAR ISI

LEMBAR PERSETUJUAN KERJA PRAKTEK.....	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	v
DAFTAR LAMPIRAN.....	viii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
DAFTAR SINGKATAN	xi
Abstrak.....	xii
BAB I.....	13
PENDAHULUAN	13
1.1 Latar Belakang	13
1.2 Identifikasi Masalah	14
1.3 Batasan Masalah.....	14
1.4 Rumusan Masalah	15
1.5 Tujuan Kerja Praktek.....	15
1.6 Manfaat Kerja Praktek.....	15
1.7 Sistematika Penulisan.....	15
BAB II.....	17
TINJAUAN PUSTAKA DAN LANDASAN TEORI	17
2.1 Tinjauan Pustaka	17
2.2 Landasan Teori	18
2.2.1 Pengertian Perancangan	18

2.2.2 Sistem Informasi	19
2.2.3 Digitalisasi Layanan.....	19
2.2.4 Efisiensi Operasional.....	20
2.2.5 Layanan Pelanggan.....	21
2.2.6 Pengertian <i>Website</i>	22
2.2.7 Pengertian Basis Data.....	23
2.2.8 Pengertian <i>XAMPP</i>	23
2.2.9 <i>HTML (Hypertext Markup Language)</i>	24
2.2.10 Pengertian <i>UML (Unified Modeling Language)</i>	24
2.2.11 Pengertian <i>Use Case Diagram</i>	25
2.2.12 Pengertian <i>Activity Diagram</i>	27
2.2.14 Pengertian <i>Class Diagram</i>	30
2.2.15 Desain Web Responsif dan <i>Bootstrap</i>	31
BAB III	33
HASIL DAN PEMBAHASAN.....	33
3.1 Sejarah Singkat dan Profil Perusahaan.....	33
3.2 Visi dan Misi	35
3.2.1 Visi	35
3.2.2 Misi	35
3.3 Struktur Organisasi.....	35
3.4 Sistem Berjalan	37
3.5 Analisis Sistem.....	38
3.5.1 Analisis Proses Bisnis (Sistem Berjalan)	38
3.5.2 Permasalahan.....	39
3.5.3 Kebutuhan Sistem	40
3.6 Perancangan Sistem.....	41

3.6.1	<i>Use Case Diagram</i>	42
3.6.2	<i>Activity Diagram</i> Pendaftaran dan <i>Login</i> Pelanggan.....	45
3.6.3	<i>Activity Diagram</i> Pemesanan Servis oleh Pelanggan	47
3.6.4	<i>Activity Diagram</i> Melihat Status Pesanan oleh Pelanggan	49
3.6.5	<i>Activity Diagram</i> Melihat Riwayat Servis.....	51
3.6.6	<i>Activity Diagram</i> Kelola Pesanan oleh Admin.....	52
3.6.7	<i>Activity Diagram</i> Mengisi Riwayat Servis oleh Admin/Teknisi	54
3.6.8	<i>Activity Diagram</i> Melihat Data Pelanggan	56
3.6.9	<i>Activity Diagram</i> Kelola Data Barang.....	58
3.6.10	<i>Activity Diagram</i> Kelola Data Layanan.....	60
3.6.11	<i>Sequence Diagram</i> Pendaftaran Akun.....	62
3.6.12	<i>Sequence Diagram</i> <i>Login</i> Akun	65
3.6.13	<i>Sequence Diagram</i> Pemesanan Layanan oleh Pelanggan.....	67
3.6.14	<i>Sequence Diagram</i> Melihat Status Pesanan	69
3.6.15	<i>Sequence Diagram</i> Melihat Riwayat Servis oleh Pelanggan	70
3.6.16	<i>Sequence Diagram</i> Mengelola Pesanan oleh Admin/Teknisi.....	72
3.6.17	<i>Sequence Diagram</i> Mengelola Data Barang oleh Admin/Teknisi.	74
3.6.18	<i>Sequence Diagram</i> Pengelolaan Data Layanan (Admin)	76
3.6.19	<i>Class Diagram</i>	78
3.6.20	Perancangan Tabel <i>Database</i>	80
3.6.21	Tampilan Penguna (User Interface).....	83
BAB IV	93
KESIMPULAN DAN SARAN	93
4.1	Kesimpulan.....	93
4.2	Saran.....	93
DAFTAR PUSTAKA	95

DAFTAR GAMBAR

Gambar 3. 1 Logo Usaha	33
Gambar 3. 2 Dokumentasi Pekerjaan	34
Gambar 3. 3 Struktur Organisasi	36
Gambar 3. 4 Alur Pemesanan	37
Gambar 3. 5 Alur Usulan Sistem.....	40
Gambar 3. 6 Use case Diagram	42
Gambar 3. 7 Activity Diagram Pendaftaran dan Login Pelanggan	46
Gambar 3. 8 Activity Diagram Pemesanan Servis oleh Pelanggan.....	48
Gambar 3. 9 Activity Diagram Melihat Status Pesanan oleh Pelanggan.....	50
Gambar 3. 10 Activity Diagram Melihat Riwayat Servis oleh Pelanggan	51
Gambar 3. 11 Activity Diagram Kelola Pesanan oleh Admin	53
Gambar 3. 12 Activity Diagram Mengisi Riwayat Servis oleh Admin	55
Gambar 3. 13 Activity Diagram Melihat Data Pelanggan.....	57
Gambar 3. 14 Activity Diagram Kelola Data Barang.....	59
Gambar 3. 15 Activity Diagram Kelola Data Layanan	61
Gambar 3. 16 Sequence Diagram Pendaftaran Akun	63
Gambar 3. 17 Sequence Diagram Login	65
Gambar 3. 18 Sequence Diagram Pemesanan Layanan	67
Gambar 3. 19 Sequence Diagram Melihat Status Pesanan.....	69
Gambar 3. 20 Sequence Diagram Melihat Riwayat Servis	71
Gambar 3. 21 Sequence Diagram Melihat Daftar Pesanan	72
Gambar 3. 22 Sequence Diagram Memperbarui Status Pesanan	73
Gambar 3. 23 Sequence Diagram Mengelola Data Barang.....	75
Gambar 3. 24 Sequence Diagram Mengelola Data Layanan.....	77
Gambar 3. 25 Class Diagram Sistem.....	79
Gambar 3. 26 Perancangan Antarmuka Halaman Login.....	83
Gambar 3. 27 Perancangan Antarmuka Halaman Registrasi.....	84
Gambar 3. 28 Perancangan Antarmuka Dashboard Pelanggan.....	85
Gambar 3. 29 Perancangan Antarmuka Formulir Pemesanan Layanan	87
Gambar 3. 30 Perancangan Antarmuka Halaman Status Pesanan.....	89
Gambar 3. 31 Perancangan Antarmuka Dashboard Admin	90

DAFTAR TABEL

Tabel 2. 1 Daftar Simbol Use Case Diagram	26
Tabel 2. 2 Daftar Simbol Activity Diagram	28
Tabel 2. 3 Daftar Simbol Sequace Diagram	29
Tabel 2. 4 Daftar Simbol Class Diagram.....	31
Tabel 3. 1 Database Pengguna.....	81
Tabel 3. 2 Database Barang.....	81
Tabel 3. 3 Database Layanan.....	82
Tabel 3. 4 Database Pesanan	82

DAFTAR SINGKATAN

AC = Air Conditioner

AJT = Adi Jaya Teknik

API = Application Programming Interface

CRM = Customer Relationship Management

CRUD = Create, Read, Update, Delete

CSS = Cascading Style Sheets

DBMS = Database Management System (Sistem Manajemen Basis Data)

ERD = Entity-Relationship Diagram

HTML = Hypertext Markup Language K

PHP = Hypertext Preprocessor

Abstrak

Bengkel AC Adi Jaya Teknik menghadapi tantangan efisiensi operasional karena proses bisnisnya yang masih berjalan secara manual. Pencatatan pesanan melalui aplikasi WhatsApp dan catatan pribadi menyebabkan data pelanggan tidak terstruktur, sulit dilacak, dan berisiko menimbulkan kesalahan penjadwalan yang dapat mempengaruhi kepuasan pelanggan. Tujuan dari kerja praktek ini adalah merancang sebuah sistem informasi layanan berbasis web untuk mendigitalisasi proses bisnis inti, mulai dari pencatatan pesanan hingga pengelolaan data pelanggan dan teknisi secara terpadu guna meningkatkan efisiensi operasional. Perancangan sistem ini menggunakan metode pemodelan visual *Unified Modeling Language (UML)*. Alur kerja, interaksi pengguna, dan arsitektur sistem dimodelkan secara detail melalui *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* yang menjadi landasan untuk perancangan basis data serta antarmuka pengguna. Hasil dari perancangan ini adalah sebuah prototipe sistem informasi yang komprehensif, mencakup fitur pendaftaran, pemesanan layanan, pengelolaan pesanan oleh admin, serta pelacakan status servis secara *real-time*. Sistem yang dirancang ini menyediakan sebuah cetak biru (*blueprint*) yang logis dan solid untuk tahap implementasi, dengan potensi besar untuk meningkatkan efisiensi operasional dan transparansi layanan secara signifikan di Bengkel AC Adi Jaya Teknik.

Bengkel AC Adi Jaya Teknik faces operational efficiency challenges due to its reliance on manual business processes. Order recording through WhatsApp and personal notes leads to unstructured customer data, tracking difficulties, and the risk of scheduling errors that can affect customer satisfaction. The objective of this practical work is to design a web-based service information system to digitize core business processes, from order entry to integrated management of customer and technician data, to improve operational efficiency. This system design utilizes the *Unified Modeling Language (UML)* visual modeling method. Workflows, user interactions, and system architecture are modeled in detail through *Use Case*, *Activity*, *Sequence*, and *Class Diagrams*, which form the foundation for the database and user interface design. The result of this design is a comprehensive information system prototype that includes features for registration, service ordering, order management by an admin, and real-time service status tracking. The designed system provides a logical and solid blueprint for the implementation phase, with significant potential to improve operational efficiency and service transparency at Bengkel AC Adi Jaya Teknik.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan informasi teknologi, banyak pelaku usaha, terutama di sektor Usaha Kecil dan Menengah (UKM), mulai mempertimbangkan digitalisasi sebagai cara untuk mencapai layanan dan meningkatkan kualitas operasional. Salah satu sektor yang terdampak adalah layanan servis Air Conditioner (AC), yang umumnya masih bergantung pada metode konvensional seperti pemesanan melalui aplikasi pesan instan *WhatsApp* dan penyebaran informasi secara lisan.



Gambar 1. 1 Penerima Pesanan Melalui WhatsApp

Usaha bengkel AC Adi Jaya Teknik selama ini menerima pesanan secara manual, sehingga berpotensi menimbulkan kendala dalam pengelolaan data pelanggan, penjadwalan teknisi, dan pembacaan proses servis. Hal ini dapat mempengaruhi kepuasan pelanggan serta menghambat pengembangan usaha ke arah yang lebih profesional dan terstruktur. Digitalisasi melalui pengembangan sistem informasi berbasis *website* yang dapat diakses oleh pelanggan dan admin melalui perangkat desktop maupun *mobile* (browser) untuk operasional internal menjadi solusi yang relevan. Sistem ini diharapkan mampu meningkatkan efisiensi

pengelolaan layanan, memperbaiki alur komunikasi antara pelanggan dan teknisi, serta memberikan pengalaman layanan yang lebih baik bagi pelanggan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat diidentifikasi beberapa permasalahan dalam proses layanan servis di Bengkel AC Adi Jaya Teknik sebagai berikut:

1. Permintaan layanan servis masih dicatat secara manual , seperti melalui pesan *WhatsApp* atau catatan pribadi, tanpa dokumentasi yang terstruktur. Hal ini menyebabkan data pelanggan sering tercecer, sulit dilacak kembali, dan berisiko tidak terdokumentasi dengan baik.
2. Informasi pelanggan dan teknisi belum terkelola secara terpadu, sehingga proses penjadwalan teknisi menjadi kurang efisien. Dalam beberapa kasus, terjadi tumpang tindih jadwal atau keterlambatan penanganan karena data tidak tersusun rapi.
3. Tidak adanya pelacakan status servis secara *real-time*, sehingga pelanggan tidak mengetahui sampai tahap mana penggerjaan dilakukan. Hal ini mengurangi kepercayaan pelanggan dan dapat berdampak pada kepuasan layanan secara keseluruhan.

1.3 Batasan Masalah

Berdasarkan permasalahan yang telah diidentifikasi, pembahasan dalam laporan ini dibatasi hanya pada masalah layanan inti yang berkaitan dengan pencatatan permintaan servis di Bengkel AC Adi Jaya Teknik. Pembahasan fokus ditujukan pada kendala yang timbul akibat pencatatan manual, seperti kurangnya keterpaduan data antara pelanggan dan teknisi, serta tidak tersedianya informasi status penggerjaan servis secara terbuka dan dapat dicari oleh pelanggan. Laporan ini tidak membahas aspek lain seperti pengelolaan suku cadang, keuangan bengkel, maupun kegiatan pemasaran, karena berada di luar fokus permasalahan yang dibahas.

1.4 Rumusan Masalah

Berdasarkan batasan masalah yang telah ditetapkan, maka rumusan masalah dalam laporan ini adalah, Bagaimana merancang sistem informasi layanan bengkel AC yang dapat mencatat permintaan servis secara terstruktur, serta memfasilitasi pengelolaan informasi pelanggan dan petugas secara efisien?

1.5 Tujuan Kerja Praktek

Tujuan dari pelaksanaan praktek kerja ini adalah untuk merancang sistem layanan informasi bengkel AC yang mampu mendokumentasikan permintaan servis secara sistematis, mengelola data pelanggan dan teknisi secara terpadu, serta meningkatkan efisiensi proses pelayanan yang sebelumnya masih dilakukan secara manual.

1.6 Manfaat Kerja Praktek

Kegiatan Kerja Praktik ini diharapkan dapat memberikan kontribusi nyata yang dapat dirasakan oleh berbagai pihak khususnya bagi mahasiswa, dan umumnya bagi semua pihak yang terlibat dalam penulisan laporan ini, antara lain:

1. Bagi Pelanggan: Memberikan kemudahan dalam pemesanan layanan serta akses terhadap informasi status servis secara transparan.
2. Bagi Bengkel AC: Mendukung pengelolaan operasional yang lebih efisien, profesional, dan terdokumentasi dengan baik.
3. Bagi Akademik: Memberikan kontribusi terhadap pengembangan sistem informasi berbasis teknologi sebagai solusi digital untuk UKM.

1.7 Sistematika Penulisan

Laporan kerja praktik ini disusun secara sistematis agar memudahkan pembaca dalam memahami proses perancangan sistem informasi manajemen bengkel AC yang berbasis digital. Adapun isi laporan ini dibagi beberapa menjadi bab sebagai berikut

Bab I PENDAHULUAN

berisi penjabaran mengenai latar belakang permasalahan yang melandasi kegiatan kerja praktik, identifikasi masalah, batasan masalah, rumusan masalah, tujuan dan manfaat kerja praktik, serta sistematika penulisan sebagai panduan pembaca terhadap isi laporan.

Bab II TINJAUAN PUSTAKA DAN LANDASAN TEORI

memuat pembahasan teori-teori yang mendasari penyusunan laporan, seperti konsep sistem informasi manajemen, digitalisasi layanan UMKM, teknologi aplikasi berbasis *web*, serta referensi dari penelitian terdahulu yang relevan dengan topik kerja praktik.

Bab III HASIL DAN PEMBAHSAN

berisi uraian mengenai profil Bengkel AC Adi Jaya Teknik sebagai objek kajian, analisis terhadap sistem yang berjalan saat ini, mengidentifikasi kebutuhan pengguna, serta merancang sistem informasi sebagai solusi dari permasalahan yang dihadapi. Perancangan mencakup diagram, struktur data, dan antarmuka pengguna yang dirancang dalam bentuk prototype *web*.

Bab IV PENUTUP

memuat kesimpulan dari kegiatan kerja praktik berdasarkan hasil yang diperoleh, serta saran-saran yang dapat diberikan untuk pengembangan sistem lebih lanjut maupun implementasi di masa mendatang.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Empat penelitian berikut memuat topik penelitian ini dalam identifikasi pendekatan digitalisasi layanan dan pengelolaan informasi berbasis sistem informasi di sektor usaha kecil dan menengah, khususnya pada layanan bengkel.

1. Menurut penelitian yang dilakukan oleh (Susinah Kuntadi dan Dwi Puryati) yang berjudul “**Implementasi Digitalisasi Manajemen Usaha Melalui Digital Marketing dan Pembukuan Digital pada UMKM Bengkel Vespa Kota Bandung**” pada tahun 2024. Penelitian ini menekankan proses transformasi digital dalam manajemen bengkel melalui pemasaran digital (*digital marketing*) dan sistem pembukuan elektronik. Studi ini menekankan pentingnya keterpaduan antara layanan digital dan pencatatan keuangan untuk mendukung profesionalisme usaha bengkel[1].
2. Menurut penelitian yang dilakukan oleh (Lilis Setyowati dan Untung Lasiyono) yang berjudul “**Optimalisasi Perencanaan dan Penjadwalan Produksi untuk Meningkatkan Efisiensi Operasional**” pada tahun 2024. Penelitian ini membahas pentingnya perencanaan dan penjadwalan produksi yang optimal dalam meningkatkan efisiensi operasional perusahaan. Penjadwalan yang efektif dapat mengurangi waktu tunggu, meningkatkan pemanfaatan sumber daya, dan menurunkan biaya operasional. Hasil temuan ini relevan dengan sistem pelayanan jasa seperti bengkel, yang membutuhkan ketepatan jadwal dalam menangani permintaan servis pelanggan[2].
3. Menurut penelitian yang dilakukan oleh (Giovan Meidy, Sandy Kosas, David, Susanti M. Kuway.) yang berjudul “**Perancangan Mobile Web Bengkel Mobil Mizu Pontianak**” pada tahun 2024. Penelitian ini merancang sistem informasi berbasis *mobile web* di Bengkel Mobil Mizu, Kubu Raya, dengan *framework Laravel* dan *Bootstrap*. Sistem ini

menggantikan proses manual dalam pengelolaan layanan dan inventaris. Hasil penelitian menunjukkan sistem tersebut mampu meningkatkan efisiensi operasional dan memudahkan akses pelanggan dalam melakukan pemesanan layanan[3].

4. Menurut penelitian yang dilakukan oleh (Dwi Vernanda, Zahran Zain Aprilia Putri, dan Setiawan Budi) yang berjudul “**Sistem Informasi Bengkel Online Berbasis Aplikasi**” pada tahun 2024. Penelitian ini membahas implementasi dan pengembangan sistem informasi bengkel online berbasis aplikasi. Sistem ini dirancang untuk mempermudah proses pemesanan layanan, pemantauan status perbaikan, dan komunikasi antara pelanggan dan bengkel. Hasil pengujian menunjukkan bahwa sistem ini berhasil meningkatkan efisiensi operasional bengkel dan memperluas jangkauan layanan melalui platform digital[4].

2.2 Landasan Teori

2.2.1 Pengertian Perancangan

Perancangan sistem informasi merupakan tahap penting dalam siklus pengembangan sistem yang bertujuan untuk merancang struktur dan komponen utama sistem secara menyeluruh. Menurut R. Empu Cahyoari Yuwono (2023), perancangan sistem dipahami sebagai aktivitas yang bertujuan menentukan alur sistem kerja serta bagaimana perangkat lunak akan dibangun berdasarkan perencanaan teknis yang matang. Pada tahap ini, dilakukan penggambaran alur sistem, desain antarmuka pengguna, serta pembuatan model basis data yang digunakan sebagai dasar implementasi sistem[5].

Sementara itu, Daniel Riano Kaparang, Raynaldi Ilyas, Stralen Pratasik (2022), menyatakan bahwa perancangan adalah bentuk aktivitas sistem pengembangan yang harus dilakukan dengan perencanaan yang tepat, guna memastikan seluruh elemen perangkat lunak terintegrasi sesuai kebutuhan pengguna. Penekanan utama dalam penelitian ini adalah bahwa perancangan bukan hanya soal teknis, namun juga berkaitan erat dengan pemahaman terhadap

kebutuhan pengguna serta penyesuaian terhadap konteks lingkungan kerja yang dituju[6].

Pemahaman tentang perancangan menjadi landasan awal dalam membangun sistem informasi yang sesuai kebutuhan pengguna. Untuk itu, diperlukan pengetahuan mengenai komponen dan fungsi sistem informasi secara keseluruhan. perbedaan utama dari kedua pandangan tersebut terletak pada fokus pendekatannya. Yuwono lebih menekankan aspek teknis dan alur proses dalam perancangan sistem, sementara Kaparang menyoroti pentingnya perencanaan menyeluruh dan pemahaman konteks pengguna sebagai dasar keberhasilan perancangan informasi.

2.2.2 Sistem Informasi

Sistem informasi adalah suatu sistem yang terintegrasi antara teknologi, prosedur, data, dan manusia untuk mendukung aktivitas operasional dan pengambilan keputusan dalam suatu organisasi. Menurut Wahyudin dan Rahayu (2020), sistem informasi merupakan kombinasi dari berbagai komponen teknologi informasi yang saling berinteraksi untuk menghasilkan informasi, serta membentuk jalur komunikasi yang mendukung proses kerja dalam organisasi. Penelitian ini menyatakan bahwa sistem informasi bukan hanya perangkat lunak dan perangkat keras, tetapi juga mencakup proses bisnis dan sumber daya manusia yang memanfaatkannya untuk mencapai tujuan tertentu[7].

Sistem informasi menjadi landasan penting dalam proses digitalisasi layanan karena mampu mengotomatisasi kegiatan manual, mempercepat distribusi informasi, dan mempermudah pemantauan operasional. Dengan demikian, keberadaan sistem informasi memberikan kontribusi yang signifikan dalam meningkatkan efisiensi dan kualitas layanan di berbagai sektor, termasuk sektor jasa seperti bengkel AC.

2.2.3 Digitalisasi Layanan

Digitalisasi layanan merupakan proses mengubah layanan tradisional menjadi layanan berbasis teknologi digital dengan memanfaatkan teknologi

informasi untuk meningkatkan efisiensi, aksesibilitas, dan kualitas pelayanan. Penelitian Dian Anggraeni dan Willy Arafah (2023) pada layanan perbankan digital menemukan bahwa kualitas digitalisasi termasuk responsivitas, kemudahan penggunaan (*ease of use*), dan transparansi biaya mempengaruhi kepuasan dan niat pelanggan untuk bertahan di platform layanan digital[8].

Di sektor e-commerce, Arfi Fajar dan Pipit Sundari (2024) menegaskan bahwa digitalisasi layanan pelanggan, seperti penggunaan chatbot, personalisasi berbasis data, dan integrasi saluran layanan (*omnichannel*) berperan signifikan dalam meningkatkan pengalaman dan loyalitas pelanggan[9],

Selain itu, Tri Wahyuningjati dkk. (2024) membahas tantangan dan peluang digitalisasi dalam manajemen jasa di era Industri 4.0 yakni efisiensi operasional yang lebih baik, layanan yang dipersonalisasi, dan pengambilan keputusan berbasis data walaupun hal ini masih terkendala oleh resistensi budaya organisasi dan infrastruktur teknologi yang belum optimal[10].

Dari berbagai temuan tersebut terlihat bahwa digitalisasi layanan tidak hanya meningkatkan kualitas dan kecepatan pelayanan, tetapi juga menjadi strategi penting untuk membangun hubungan jangka panjang dengan pelanggan. Implementasi digitalisasi dalam sistem bengkel AC seperti pelacakan status servis secara *real-time*, pelaporan digital, atau interaksi pelanggan berbasis web/mobile dapat mengadopsi prinsip-prinsip ini untuk menciptakan nilai lebih dan memperkuat kepercayaan pelanggan.

2.2.4 Efisiensi Operasional

Efisiensi operasional adalah kemampuan organisasi dalam memaksimalkan output menggunakan sumber daya seminimal mungkin, sambil mempertahankan atau meningkatkan kualitas layanan. Supriyadi (2024) dalam penelitiannya pada industri manufaktur di Cilegon menunjukkan bahwa integrasi Sistem Informasi Manajemen SDM (HRMIS) mampu meningkatkan efisiensi operasional melalui penghematan waktu, akurasi data, dan dukungan pengambilan keputusan yang lebih cepat[11].

Selain itu, Sudianto dan Sutopo (2025) juga menekankan bahwa sistem informasi manajemen berbasis cloud berperan penting dalam menghubungkan alur proses bisnis, meminimalkan kesalahan manual, serta memfasilitasi kolaborasi antar departemen melalui akses data yang keripik dan dapat diakses kapan pun. Dengan demikian, penggunaan teknologi cloud tidak hanya mendukung efisiensi secara teknis, tetapi juga strategis dalam pengelolaan operasional organisasi.[12]

Yulianti dkk. (2024) dalam penelitiannya menyebutkan bahwa implementasi ERP berbasis cloud memberikan dampak positif terhadap efisiensi rantai pasok hingga mencapai 51,6% pada sektor industri manufaktur. Hal ini terjadi karena sistem ERP yang terintegrasi mampu mengurangi redundansi data, mempercepat aliran informasi antar divisi, dan meningkatkan koordinasi secara real-time[13].

Lebih lanjut, penelitian Fikri dkk. (2024) pada implementasi arsip elektronik di SMK Gelora Jaya Nusantara Medan mengungkap bahwa digitalisasi dokumen mampu memangkas waktu pencarian dan penyimpanan, serta mengurangi biaya fisik dan meningkatkan keamanan data[14].

2.2.5 Layanan Pelanggan

Layanan pelanggan (customer service) merupakan perhatian dan dukungan yang diberikan perusahaan kepada pelanggan sebelum, selama, dan setelah pembelian, guna memastikan kebutuhan dan ekspektasi mereka terpenuhi. Menurut Felicia dkk. (2024), kualitas layanan adalah evaluasi pelanggan terhadap kesesuaian layanan yang diterima dibandingkan dengan apa yang diharapkannya. Penelitian ini menemukan bahwa kualitas layanan yang baik termasuk aspek kehandalan (keandalan), ketanggapan (daya tanggap), dan empati (empati) bersama pengalaman pelanggan secara langsung mempengaruhi loyalitas[15].

Diyah Ayu Patmawati dan Andjarwati (2023) menyoroti bahwa layanan pelanggan yang solid akan meningkatkan kepuasan pelanggan. Penelitian mereka pada sektor jasa menemukan bahwa kualitas layanan dan pengalaman pelanggan memiliki hubungan positif yang signifikan terhadap loyalitas. Temuan ini

memperkuat pentingnya membangun layanan pelanggan yang responsif dan konsisten untuk menciptakan hubungan jangka panjang[16].

Dengan demikian, layanan pelanggan bukan sekadar menyajikan bantuan teknis, melainkan sebuah strategi layanan menyeluruh yang harus mengutamakan kepuasan, ketepatan waktu, empati, dan pengalaman positif. Implementasi sistem informasi layanan bengkel yang mampu mengelola aspek-aspek ini secara digital diharapkan dapat meningkatkan kepuasan dan loyalitas pelanggan.

2.2.6 Pengertian *Website*

Website merupakan sebuah sistem berbasis teknologi internet yang menyajikan kumpulan halaman yang saling terhubung dan dapat diakses melalui *web browser* oleh pengguna dari berbagai lokasi dan perangkat. Dalam konteks pengembangan sistem informasi, *website* tidak hanya berfungsi sebagai sarana penyampaian informasi, tetapi juga sebagai platform interaktif yang memungkinkan pertukaran data antara pengguna dan sistem.

Menurut Qori Imanda (2022), *website* didefinisikan sebagai sistem digital yang dirancang untuk memfasilitasi pengelolaan data secara lebih efisien, singkat, dan dapat diakses oleh berbagai pihak melalui jaringan internet. Dalam penelitiannya, *website* digunakan untuk mengelola proses administrasi akademik sekolah, mulai dari input data hingga pelaporan, menggunakan teknologi *PHP*, *Laravel*, dan *MySQL*. Sistem ini terbukti mampu mengurangi ketergantungan pada metode manual serta meningkatkan akurasi data dan transparansi informasi bagi pengguna (guru, siswa, dan admin sekolah) [17].

Lebih jauh lagi, *website* dinilai sebagai salah satu bentuk implementasi sistem informasi yang paling fleksibel karena dapat diakses dari mana saja dan kapan saja tanpa perlu instalasi khusus di perangkat pengguna. Dalam ranah bisnis, termasuk sektor jasa seperti bengkel AC, penggunaan *website* memungkinkan pemilik usaha untuk menyediakan layanan seperti pemesanan servis online, pelacakan status servis, dan manajemen pelanggan secara lebih profesional. Hal ini menjadikan *website* sebagai salah satu sarana digitalisasi layanan yang mendukung

efisiensi operasional, meningkatkan kepuasan pelanggan, dan pencitraan usaha yang lebih modern.

2.2.7 Pengertian Basis Data

Basis data atau *database* merupakan komponen penting dalam sistem informasi yang berfungsi untuk menyimpan, mengelola, dan mengorganisir data secara sistematis agar mudah diakses, dikelola, serta diperbarui. Dalam konteks sistem informasi layanan bengkel AC, basis data digunakan untuk menyimpan informasi pelanggan, teknisi, jadwal servis, serta status pekerjaan yang dilakukan. Menurut Khoirunnisa (2023), basis data adalah kumpulan data atau informasi yang disusun secara sistematis dan disimpan dalam komputer untuk kemudian diolah melalui perangkat lunak pengelola basis data seperti sistem manajemen basis data (DBMS)[18].

Fahzirah dan Nasution (2024) juga menyatakan bahwa keberadaan basis data sangat berpengaruh dalam mendukung efisiensi operasional karena mampu meminimalisir duplikasi data, mempercepat pengambilan keputusan, serta menjaga konsistensi dan integritas informasi. Dalam sistem informasi berbasis web, penggunaan basis data memungkinkan integrasi antarmuka pengguna dengan informasi *backend* secara *real-time*. Dengan demikian, basis data menjadi dasar utama dalam membangun sistem layanan digital yang andal, terstruktur, dan mampu memenuhi kebutuhan operasional secara optimal[19].

2.2.8 Pengertian XAMPP

XAMPP adalah paket perangkat lunak *local server* yang menyediakan lingkungan pengembangan lengkap untuk aplikasi web, mencakup server web *Apache*, database *MySQL/MariaDB*, serta interpretasi bahasa pemrograman seperti *PHP* dan *Perl*. *XAMPP* memudahkan pengembang dengan menyediakan satu paket lengkap, sehingga pengembangan web lebih cepat, mudah, dan aman[20].

Menurut Chandra dan Setyaningsih (2025), *XAMPP* digunakan sebagai wadah uji coba lokal (*local development environment*) yang memudahkan pengembang dalam melakukan benchmarking kinerja server dibandingkan platform

lain seperti *MAMP* dan *Laragon*, sehingga membantu dalam menentukan alat terbaik untuk pengembangan web secara cepat dan efisien[21].

Dalam konteks sistem informasi bengkel AC, *XAMPP* menjadi pilihan tepat untuk pengembangan prototipe web karena mempermudah proses instalasi, menyediakan lingkungan *database*, dan mendukung eksekusi script *PHP* sehingga mendukung pengujian fungsionalitas sistem secara offline sebelum diimplementasikan secara online.

2.2.9 *HTML (Hypertext Markup Language)*

HTML adalah bahasa markup standar yang digunakan untuk membuat struktur dasar halaman web. Melalui penggunaan tag-tag, *HTML* memungkinkan pengembang menyusun elemen-elemen seperti teks, gambar, tautan, dan multimedia ke dalam format yang dapat ditampilkan oleh peramban web (web browser). Menurut Dirgantara dkk. (2024), *HTML* berfungsi sebagai kerangka dasar komunikasi antara sistem dan pengguna, karena memungkinkan penyusunan informasi teks dan gambar secara terstruktur dan mudah diakses secara online[22].

Sementara itu, Mardiansyah dkk. (2024), menyebutkan bahwa *HTML* dan *CSS* menjadi fondasi penting bagi pengembangan web, dengan *HTML* bertugas membangun struktur halaman, sementara *CSS* mengatur tata letak dan tampilan visual[23].

Dengan demikian, *HTML* tidak hanya berperan sebagai bahasa markup teknis, tetapi juga sebagai medium komunikasi yang menyampaikan konten dengan cara yang terstruktur dan mudah dipahami. Dalam konteks sistem informasi bengkel AC berbasis web, *HTML* menjadi komponen utama yang harus dikuasai karena menetapkan kerangka kerja halaman depan (*front-end*) di mana pelanggan dan teknisi berinteraksi dengan sistem.

2.2.10 Pengertian *UML (Unified Modeling Language)*

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan secara luas dalam rekayasa perangkat lunak untuk merancang, mendokumentasikan, dan memvisualisasikan sistem informasi secara terstruktur.

UML menyediakan kumpulan notasi diagram yang dapat menggambarkan berbagai aspek sistem baik dari sisi proses bisnis, struktur data, hingga interaksi pengguna sehingga memudahkan pengembang sistem dalam memahami, mengembangkan, dan memelihara perangkat lunak secara efisien.

Menurut Setiaji dkk. (2024), *UML* digunakan sebagai alat bantu perancangan sistem yang mampu menjembatani kebutuhan fungsional dengan desain teknis. *UML* memungkinkan perancang sistem untuk menuangkan alur sistem secara detail dalam bentuk diagram seperti *use case*, *sequence*, dan *Activity diagram*, yang mudah dipahami oleh berbagai pihak termasuk pengembang, analis, maupun pengguna non-teknis. Dalam penelitian tersebut, *UML* terbukti mampu meningkatkan efektivitas sistem pengembangan karena alur kerja dan struktur sistem sudah terdokumentasi secara sistematis sejak awal[24].

Pendapat serupa juga disampaikan oleh Oktarino dkk. (2024) melalui penelitian yang menunjukkan bahwa penggunaan *UML* sangat membantu dalam proses analisis kebutuhan dan desain sistem informasi sekolah. *UML* digunakan untuk memodelkan proses pendaftaran, pengelolaan data siswa, dan laporan akademik dalam bentuk visual diagram yang terstandarisasi. Hal ini tidak hanya mempermudah proses desain, tetapi juga mempercepat sistem pengembangan karena diagram yang jelas dan mudah diterjemahkan ke dalam kode program[25].

Secara umum, *UML* menjadi alat yang sangat penting dalam pengembangan sistem berbasis digital, seperti yang diterapkan pada sistem layanan bengkel AC berbasis web dan mobile. Dengan *UML*, tim pengembang dapat berkomunikasi secara visual dan efisien, serta mengurangi risiko kesalahan pemahaman dalam merancang sistem.

2.2.11 Pengertian *Use Case Diagram*

Use Case adalah metode pemodelan fungsional dalam pengembangan sistem yang menggambarkan interaksi antara *aktor* (pengguna atau sistem lain) dengan sistem untuk mencapai tujuan tertentu. Menurut Tita Ayu Rospricilia dan Mochamad Nizar Palefi Ma'ady (2024) *Use Case Diagram* digunakan untuk menjelaskan secara rinci interaksi yang mungkin terjadi antara aktor dan sistem,

terutama dalam integrasi fungsional antar sistem. Diagram ini menampilkan skenario yang mencakup alur utama (*aliran utama*) dan alur alternatif (*aliran alternatif*) sehingga memudahkan analis dan pengembang dalam memahami kebutuhan fungsional sistem secara terpadu[26].

Tabel 2. 1 Daftar Simbol Use Case Diagram

Simbol	Nama	Keterangan
	Association	Sebuah jalur komunikasi antara seorang actor dengan satu <i>use case</i> dimana actor tersebut berpartisipasi
	Aktor	Merepresentasikan satu peranan yang dimainkan oleh orang luar, unit kerja, atau hal-hal yang berinteraksi dengan sistem, subsistem atau kelas
	Extensi	Penyisipan sebuah fungsionalitas tambahan ke dalam sebuah <i>use case</i> dasar yang bersifat opsional
	Generalisasi	Sebuah hubungan antara sebuah <i>use case</i> umum dengan <i>use case</i> yang lebih spesifik dimana <i>use case</i> yang lebih spesifik mewarisi <i>use case</i> umum tersebut
	Include	Penyisipan sebuah fungsionalitas tambahan ke dalam sebuah <i>use case</i> dasar dimana fungsionalitas tersebut bersifat mandatory (wajib)

Sumber[27]

2.2.12 Pengertian *Activity Diagram*

Activity Diagram merupakan salah satu jenis diagram perilaku dalam *UML* yang menggambarkan alur kerja (*workflow*) sistem secara visual, termasuk aliran kendali (*control flow*) dan data (*data flow*). Diagram ini sangat efektif digunakan untuk memodelkan proses bisnis berurutan, paralel, dan kondisional, sekaligus memperlihatkan interaksi antaraktivitas dalam system, Menurut Elmansouri, Meghzili, dan Chaoui (2021),*activity diagram* digunakan untuk memodelkan aliran proses dengan cara yang terstruktur, sehingga memudahkan verifikasi formal dan validasi behavior sistem sebelum implementasi. Diagram ini menggambarkan langkah-langkah aktivitas yang membentuk alur kerja sistem secara keseluruhan[28].

Sementara itu, Siewe dan Ngounou (2025),menjelaskan bahwa *activity diagram* tidak hanya digunakan untuk desain awal, tetapi juga dapat dijalankan secara runtime untuk verifikasi semantic menggunakan teknik model checking agar diagram dapat dieksekusi, diuji, dan diverifikasi secara otomatis. secara garis besar, *Activity Diagram* memungkinkan developer dan analis sistem untuk memvisualisasikan logika proses (baik berurutan maupun paralel), menangkap kondisi dan pengambilan keputusan, serta menyediakan dasar untuk pengujian dan validasi behavior sistem[29].

Dalam kasus pengembangan aplikasi bengkel AC berbasis web, *activity diagram* dapat digunakan untuk merancang alur pemesanan servis, penjadwalan teknisi, hingga pelacakan status penggerjaan dengan jelas dan sistemati,secara keseluruhan, penggunaan diagram aktivitas dalam pengembangan aplikasi bengkel AC memberikan manfaat besar dalam menganalisis kebutuhan sistem, mendeteksi potensi hambatan alur kerja , serta memastikan tidak ada langkah-langkah bisnis yang terlewat selama proses implementasi sistem informasi berbasis web ini.

Tabel 2. 2 Daftar Simbol Activity Diagram

Simbol	Nama	Keterangan
	Status Awal	Sebuah diagram aktifitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya di awali dengan kata kerja.
	Percabangan atau decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Generalisasi	Asosiasi penggabungan dimana lebih dari satu aktivitas di gabungkan menjadi satu.
	Status akhir	Sebuah diagram aktivitas memiliki sebuah status akhir.
	Swimlane	Memisahkan organisasi bisnis yang tertanggung jawab terhadap aktivitas yang terjadi.

Sumber[27]

2.2.13 Pengertian Sequace Diagram

Sequence Diagram adalah jenis diagram interaksi dalam *Unified Modeling Language (UML)* yang menggambarkan secara visual urutan pesan (*messages*) yang dikirim dan diproses antar objek selama periode waktu tertentu. Diagram ini membantu memodelkan behavior sistem berdasarkan skenario dari *use case*,

dengan menampilkan bagaimana event atau aktivitas dijalankan oleh objek-objek terlibat. Menurut Al-Fedaghi (2021), *Sequence Diagram* merupakan diagram *UML* paling umum kedua yang menampilkan interaksi objek serta pertukaran pesan secara kronologis dari waktu ke waktu. Diagram ini menunjukkan lifeline dari setiap objek, pesan yang dikirim (*synchronous/asynchronous*), serta kondisi atau fragment logika seperti *guard conditions* dan *loops*, sehingga membantu dalam memahami dan memvalidasi skenario sistem sebelum dilakukan implementasi[30].

Sequence Diagram menjadi sangat berguna dalam tahap perancangan sistem karena memperjelas bagaimana objek saling berkolaborasi untuk menyelesaikan suatu aksi atau alur proses, serta menyediakan dasar bagi pengembangan skenario pengujian (*test cases*). Dengan menggambarkan interaksi secara runtut, *Sequence Diagram* memudahkan analisis desain terutama untuk memastikan bahwa setiap objek dan pesan yang diperlukan telah terdefinisi dengan tepat.

Tabel 2. 3 Daftar Simbol Sequace Diagram

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya

	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Sumber[27]

2.2.14 Pengertian *Class Diagram*

Class Diagram adalah diagram statistik dalam *UML* yang berfungsi untuk memodelkan struktur sistem melalui representasi kelas (*class*), atribut (*atribut*), metode (*operasi*), dan hubungan antar kelas seperti asosiasi, pewarisan (*warisan*), agregasi, atau komposisi. Menurut Ramdany (2024) dalam jurnal “*Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web*”, *Class Diagram* digunakan untuk mendesain sistem informasi perpustakaan, termasuk entitas seperti Peminjam, Buku, dan Transaksi, beserta atribut dan relasi yang jelas serta struktur logika yang mudah dipahami oleh pengembang dan pemangku kepentingan[31].

Penggunaan *Class Diagram* memudahkan pengembang dalam memahami kebutuhan dan struktur data sistem sebelum implementasi, sekaligus menjadi acuan bagi penulisan kode dengan pendekatan berorientasi objek. Diagram ini juga membantu dalam mendeteksi inkonsistensi model, merancang *database* (memetakan ke tabel), serta memperjelas tanggung jawab masing-masing kelas dalam sistem. Dalam konteks pengembangan sistem bengkel AC berbasis web/mobile, *Class Diagram* sangat krusial untuk memodelkan entitas seperti Pelanggan, Teknisi, PesananServis, serta relasi dan fungsionalitasnya, sehingga mendukung konsistensi desain dan meminimalkan potensi kesalahan saat implementasi.

Tabel 2. 4 Daftar Simbol Class Diagram

Gambar	Nama	Keterangan
	Class	Kelas dalam struktur sistem
	Interface	Sama dengan konsep interface dalam pemrograman interface
	Association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Directed Association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. Asosiasi biasanya juga disertai dengan multiplicity
	Aggregation	Relasi antar kelas dengan makna semua bagian (whole-part)
	Dependency	Relasi antar kelas dengan makna kebergantungan antar kelas

Sumber[27]

2.2.15 Desain Web Responsif dan *Bootstrap*

Responsive Web Design (RWD) adalah pendekatan pengembangan web yang memungkinkan tampilan dan fungsi sebuah situs menyesuaikan diri secara otomatis dengan berbagai perangkat dan ukuran layar, baik desktop, tablet, maupun

smartphone. Penelitian oleh Hartanto dkk. (2024) dalam *Jurnal Pengabdian Masyarakat untuk Ilmu MIPA dan Terapannya* menunjukkan bahwa pelatihan pembuatan website responsif dengan *Flask dan Bootstrap 4.5* terbukti memungkinkan peserta membangun halaman web yang konsisten dan user-friendly di berbagai perangkat, meningkatkan efisiensi akses dan visibilitas online UMKM[32].

Bootstrap sendiri adalah *framework CSS* populer yang mendukung pembuatan desain web responsif melalui sistem grid dan komponen siap pakai. Putra.M (2020) dalam *InfoTekJar* menyebutkan bahwa *Bootstrap* menyediakan kelas-kelas responsif yang memudahkan pengembang dalam menyusun layout dinamis tanpa perlu menulis banyak kode *CSS* manual . Penelitian di *Jurnal Nasional Ilmu Komputer* juga mencatat keberhasilan penggunaan *Bootstrap* dalam pengembangan *webstore online*, yang memberikan tampilan responsif dan mempermudah interaksi pengguna melalui berbagai perangkat[33].

Dalam konteks sistem manajemen bengkel AC berbasis web, penerapan RWD dan *Bootstrap* memungkinkan aplikasi untuk dijalankan di desktop dan HP tanpa perlu pengembangan aplikasi *mobile* khusus. Hal ini memberi banyak keuntungan, seperti:

1. Konsistensi tampilan antardevice , tanpa penambahan waktu dan biaya untuk dua platform
2. Pengalaman pengguna yang baik melalui akses responsif terhadap fitur penting seperti pemesanan servis, status pelacakan, dan dashboard pekerja
3. , karena satu dasar *kode frontend* untuk semua perangkat

BAB III

HASIL DAN PEMBAHASAN

3.1 Sejarah Singkat dan Profil Perusahaan

Bengkel Adi Jaya Teknik merupakan sebuah usaha yang bergerak di bidang layanan perbaikan, pemasangan, dan perawatan perangkat pendingin udara (*Air Conditioner / AC*) serta beberapa layanan teknis lainnya seperti instalasi listrik dan CCTV. Bengkel ini Didirikan pada tahun 2014 oleh Bapak Adi, seorang dokter berpengalaman yang melihat peluang besar dalam menyediakan layanan servis AC yang lebih cepat, profesional, dan terjangkau di wilayah Cikarang dan sekitarnya. Berbekal keahlian di bidang pendingin udara dan pengalaman kerja di berbagai proyek instalasi, beliau memulai usaha ini dengan visi sederhana menghadirkan solusi layanan AC yang praktis dan terpercaya bagi masyarakat.

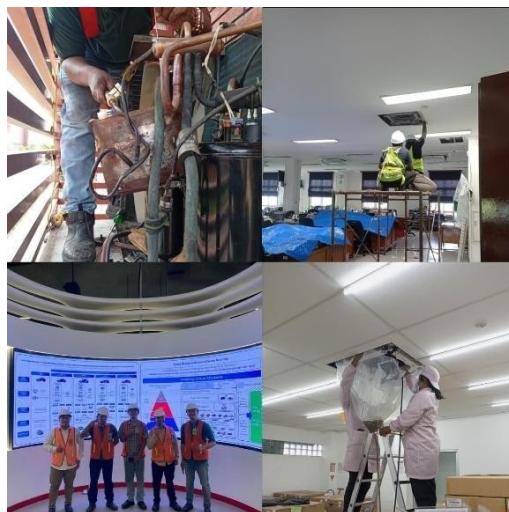
Sejak awal berdirinya, Adi Jaya Teknik beroperasi dengan sumber daya yang terbatas. Proses pemesanan layanan dilakukan secara manual melalui aplikasi pesan instan seperti WhatsApp atau melalui telepon langsung. Seluruh pencatatan data pelanggan, jadwal petugas, dan riwayat servis masih dilakukan menggunakan catatan buku dan riwayat chat yang ada di WhatsApp, yang sering kali menyulitkan dalam hal pelacakan data ketika jumlah pelanggan meningkat. Kondisi ini tidak jarang menimbulkan tantangan seperti keterlambatan pelayanan, tumpang tindih jadwal teknisi, hingga hilangnya data penting pelanggan.



Gambar 3. 1 Logo Usaha

Saat ini, Adi Jaya Teknik beroperasi di alamat sementara Kampung Sinyar Barat, Kontrakan Haji Jefri, Cikarang Timur . Lokasi ini dipilih sebagai tempat operasional sementara sambil menunggu rencana relokasi ke ruko permanen yang

lebih strategis untuk mendukung perkembangan usaha di masa depan. Meski dengan keterbatasan lokasi, Adi Jaya Teknik terus berupaya memberikan layanan terbaik kepada pelanggan. usaha ini memiliki dua karyawan inti , yaitu pemilik sekaligus manajer teknis, dan seorang asisten tetap yang bertugas membantu pekerjaan. Selain itu, untuk mengantisipasi permintaan terutama di musim panas atau periode tertentu dengan permintaan tinggi, Adi Jaya Teknik juga menjamin kerja sama dengan beberapa teknisi cadangan *freelance* yang dapat dipanggil sewaktu-waktu sesuai kebutuhan.



Gambar 3. 2 Dokumentasi Pekerjaan

Selain layanan perbaikan dan pemasangan AC, Adi Jaya Teknik juga menyediakan berbagai jasa lainnya untuk memenuhi kebutuhan pelanggan. Layanan yang ditawarkan meliputi bongkar pasang AC baru maupun bekas, pembersihan serta servis berkala perangkat AC guna memastikan kinerja yang optimal, hingga perbaikan AC dan kulkas untuk berbagai tipe dan merek. Tidak hanya itu, bengkel ini juga melayani instalasi jaringan listrik rumah dan kantor, pemasangan serta pemeliharaan sistem CCTV, serta penjualan beli unit AC bekas jika stok tersedia.

Adi Jaya Teknik menerapkan jam operasional yang fleksibel, yaitu setiap hari mulai pukul 08.00 hingga 17.00. Namun, seringkali jam operasional ini disesuaikan dengan kebutuhan pelanggan di lapangan. Dalam kondisi tertentu, terutama untuk kebutuhan layanan yang mendesak, tim teknisi siap melayani

hingga malam hari untuk memastikan kepuasan pelanggan dengan komitmen tinggi terhadap kualitas layanan, profesionalitas teknisi, serta harga yang kompetitif, Adi Jaya Teknik terus membangun kepercayaan pelanggan. Bengkel ini berupaya untuk menjadi pilihan utama masyarakat Cikarang Timur dan sekitarnya dalam layanan perbaikan AC. Ke depan, transformasi menuju layanan berbasis digital juga menjadi salah satu fokus pengembangan, agar proses pemesanan hingga pelaporan dapat dilakukan dengan lebih cepat, efisien, dan transparan.

3.2 Visi dan Misi

3.2.1 Visi

Menjadi bengkel AC dan layanan teknis terpercaya di wilayah Cikarang dan sekitarnya dengan pelayanan cepat, profesional, serta berbasis teknologi digital untuk mendukung efisiensi dan transparansi layanan.

3.2.2 Misi

1. Memberikan layanan bongkar pasang, servis, dan pemeliharaan AC dengan standar kualitas terbaik untuk kepuasan pelanggan.
2. Meningkatkan efisiensi operasional melalui digitalisasi layanan agar proses pemesanan, layanan pelacakan, dan komunikasi menjadi lebih cepat dan transparan.
3. Berpikir tentang usaha ke arah perusahaan berbadan hukum (PT) yang mampu membuka lapangan pekerjaan baru bagi masyarakat sekitar.
4. Memperluas cakupan layanan tidak hanya pada pendingin udara tetapi juga merambah ke bidang mesin industri dan teknologi robotik untuk perusahaan dan instansi di masa depan.
5. Menjaga profesionalitas dan kepercayaan pelanggan melalui pelayanan yang responsif, harga yang kompetitif, dan inovasi berkelanjutan.

3.3 Struktur Organisasi

Adi Jaya Teknik menerapkan struktur organisasi yang sederhana dan efisien, sesuai dengan skala usahanya sebagai unit layanan UMKM di bidang pendingin udara dan layanan teknis lainnya. Saat ini, pengelolaan operasional

bengkel dilakukan langsung oleh pemilik usaha dibantu oleh satu orang asisten petugas.

Pemilik usaha sekaligus bertindak sebagai Manajer Utama , yang memegang peran sentral dalam pengambilan keputusan, manajemen pelanggan, hingga pengelolaan jadwal layanan. Dalam kesehariannya, beliau juga turun langsung ke lapangan sebagai teknisi utama, memastikan kualitas layanan yang diberikan kepada pelanggan. asisten teknisi memiliki peran membantu pekerjaan teknis di lapangan, mulai dari pemasangan AC, perawatan berkala, hingga penanganan instalasi listrik atau CCTV. Selain itu, Adi Jaya Teknik juga memiliki beberapa tenaga tambahan atau tenaga lepas yang dapat dipanggil sewaktu-waktu jika ada peningkatan volume pekerjaan atau permintaan layanan mendesak dari pelanggan. struktur organisasi ini memungkinkan Adi Jaya Teknik untuk tetap fleksibel dalam menghadapi permintaan layanan, sekaligus menjaga kualitas dan efisiensi operasional.



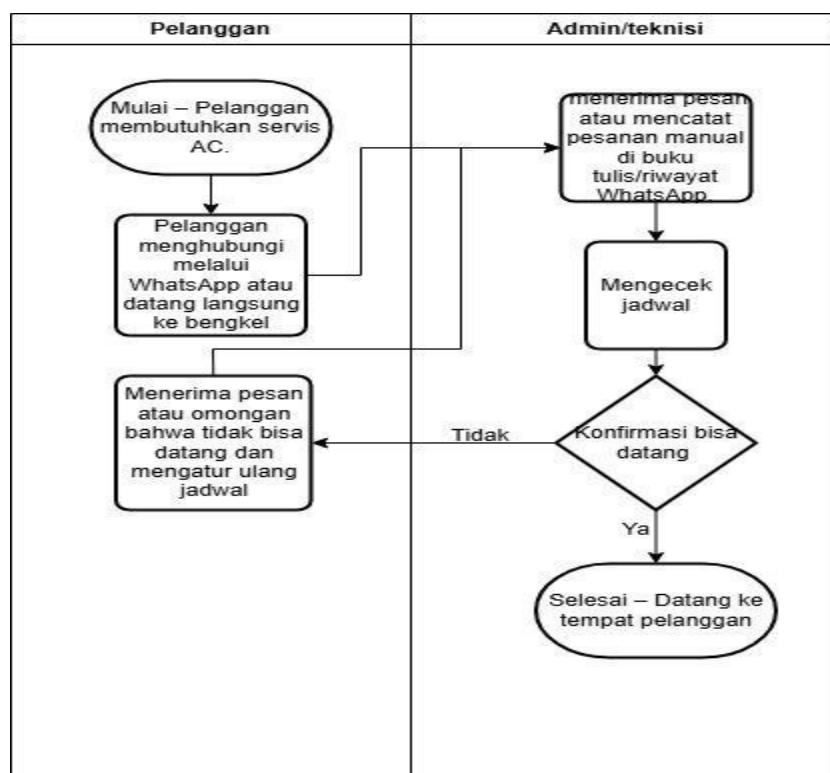
Gambar 3. 3 Struktur Organisasi

Gambar di atas menampilkan susunan hirarki vertikal sederhana yang diterapkan oleh Adi Jaya Teknik. Posisi Pemilik Usaha/Manajer Utama berada di puncak sebagai pengambil keputusan sekaligus teknisi utama. Di bawahnya terdapat Asisten/Teknisi Utama yang mendukung pelaksanaan layanan teknis sehari-hari.

Selain itu, terdapat Teknisi Tambahan (Cadangan) yang berperan sebagai tenaga pembuka ketika terjadi penyiaran pekerjaan atau kebutuhan mendesak dari pelanggan. model organisasi ini dirancang untuk memastikan alur kerja yang cepat dan koordinasi yang efektif dengan jumlah tenaga kerja minimal, yang sangat sesuai untuk skala UMKM.

3.4 Sistem Berjalan

Saat ini sistem operasional Adi Jaya Teknik masih berjalan secara manual tanpa dukungan aplikasi atau sistem informasi berbasis digital. Pemesanan layanan servis umumnya dilakukan melalui WhatsApp , di mana pelanggan menghubungi langsung pemilik bengkel untuk menyampaikan keluhan atau permintaan servis. Selain melalui pesan instan, beberapa pelanggan yang tinggal di sekitar bengkel terkadang datang langsung ke lokasi operasional untuk memesan layanan.seluruh pesanan yang masuk dicatat secara manual oleh pemilik dalam buku tulis sebagai pengingat, meskipun seringkali juga hanya mengandalkan riwayat percakapan WhatsApp . Metode pencatatan ini menyebabkan data pelanggan dan riwayat layanan menjadi kurang terstruktur dan rawan tercecer.



Gambar 3. 4 Alur Pemesanan

Dalam hal penjadwalan teknisi, pemilik bengkel sekaligus bertindak sebagai teknisi utama, sementara asisten teknisi membantu pekerjaan di lapangan. Adanya keterbatasan jumlah tenaga kerja tetap, serta tidak adanya sistem yang mendukung manajemen jadwal, sering lupa menyebabkan jadwal pelayanan dan bahkan

keterlambatan penanganan terhadap pelanggan. selain itu, proses pelaporan status servis kepada pelanggan juga masih dilakukan secara manual melalui pesan WhatsApp ketika diminta. Tidak ada sistem pelacakan real-time yang memungkinkan pelanggan memantau perkembangan servis secara mandiri. kondisi ini membuat alur operasional kerja cenderung tidak efisien, terutama saat volume pesanan meningkat. Oleh karena itu, diperlukan solusi berbasis digital untuk memperbaiki pencatatan, penjadwalan, dan transparansi layanan demi meningkatkan kepuasan pelanggan serta profesionalitas bengkel.

3.5 Analisis Sistem

Analisis sistem dilakukan untuk memahami kelemahan pada proses operasional yang berjalan saat ini di Bengkel AC Adi Jaya Teknik, serta untuk merumuskan kebutuhan sistem informasi yang akan dikembangkan agar dapat memberikan solusi yang efektif, efisien, dan sesuai dengan kebutuhan pengguna.

3.5.1 Analisis Proses Bisnis (Sistem Berjalan)

Proses bisnis layanan di Bengkel AC Adi Jaya Teknik saat ini masih bersifat manual. Pelanggan memesan layanan melalui aplikasi *WhatsApp* atau datang langsung ke bengkel. Semua pesanan dicatat di buku tulis atau mengandalkan riwayat chat.

Alur kerja sistem berjalan dapat diartikan sebagai berikut: Pelanggan yang menghubungi → Admin/teknisi mencatat pesanan → Pengerjaan layanan → Pelanggan menghubungi kembali untuk memperbarui status.

Kelemahan dari proses ini adalah ketergantungan pada memori teknisi dan catatan manual yang rawan hilang, serta tidak adanya sistem pengingat untuk layanan jadwal. dari hasil analisis, kelemahan-kelemahan pada sistem manual saat ini antara lain:

1. Tidak memikirkannya data layanan menyebabkan kesulitan dalam mencari riwayat layanan.
2. Kurangnya sistem pengingat otomatis membuat jadwal pengerjaan sering

terlewat.

3. Kurang transparan bagi pelanggan karena tidak ada fitur pelacakan kemajuan layanan.

3.5.2 Permasalahan

Berdasarkan hasil observasi terhadap proses layanan di Bengkel AC Adi Jaya Teknik, ditemukan beberapa titik yang berpengaruh terhadap kinerja operasional dan kepuasan pelanggan, antara lain:

1. Pencatatan manual

Proses pencatatan pesanan layanan masih dilakukan secara manual melalui aplikasi *WhatsApp* dan buku catatan. Hal ini sering menyebabkan data pesanan tercecer, sulit dilacak kembali, dan rawan kesalahan (*human error*). Selain itu, jika volume pesanan tinggi, risiko kehilangan atau lupa

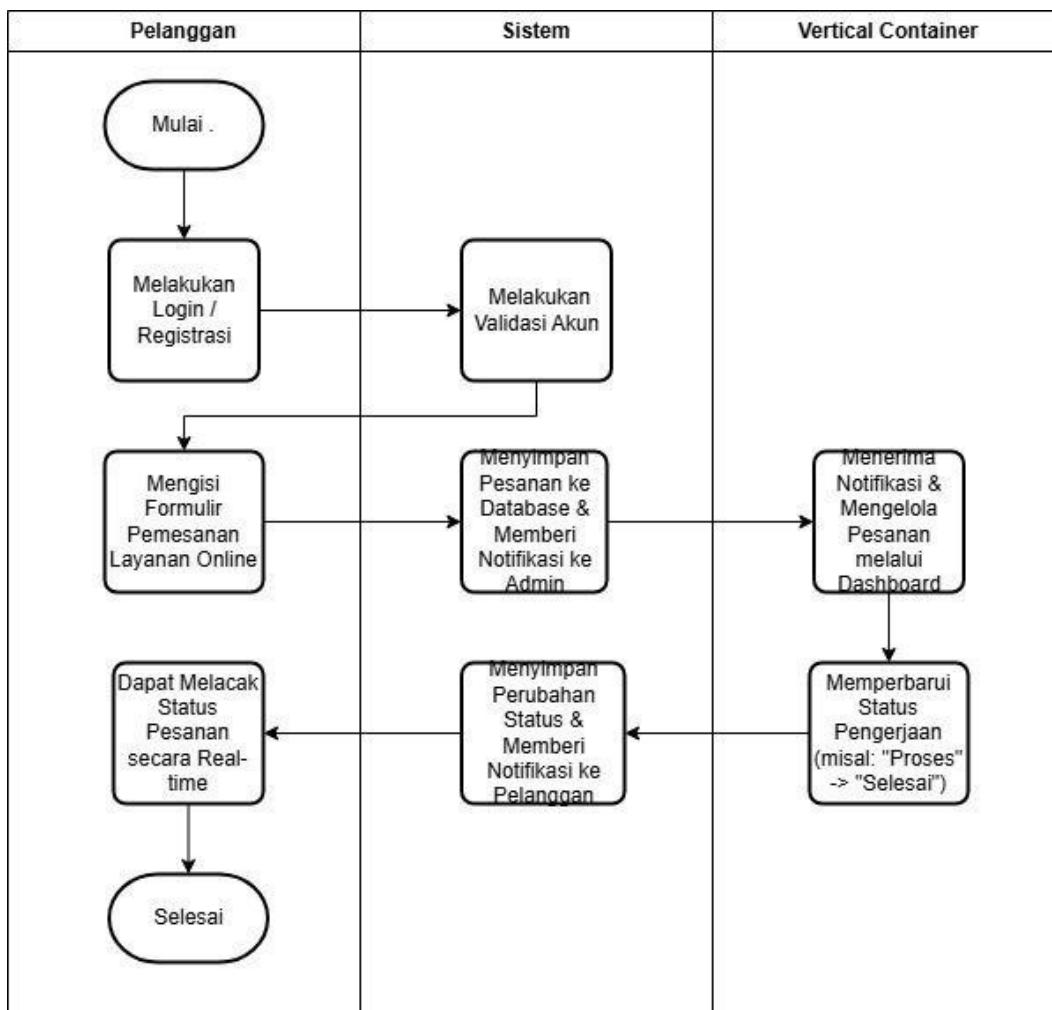
2. Kurangnya keterpaduan data

Data pelanggan dan teknisi tidak dikelola dalam sistem yang terintegrasi. Akibatnya, proses penjadwalan layanan menjadi tidak teratur dan berpotensi terjadi tumpang tindih jadwal atau keterlambatan dalam penanganan pesanan.

3. Tidak adanya pelacakan status servis secara real-time

Pelanggan tidak memiliki akses untuk mengetahui status pengerjaan servis mereka secara langsung. Informasi kemajuan pengerjaan hanya bisa diperoleh dengan cara menghubungi teknisi secara manual, yang kurang praktis dan sering membuat pelanggan tidak puas.

3.5.3 Kebutuhan Sistem



Gambar 3.5 Alur Usulan Sistem

Alur pada Gambar 3.5 menunjukkan bagaimana sistem baru mengotomatisasi proses bisnis. Proses dimulai dari pelanggan yang berinteraksi langsung dengan sistem melalui formulir digital, menghilangkan kebutuhan pencatatan manual. Setiap pesanan dan perubahan status akan tersimpan secara terpusat di *database* dan sistem secara proaktif memberikan notifikasi kepada admin dan pelanggan. Hal ini secara langsung menjawab kelemahan pada sistem berjalan dengan menciptakan alur kerja yang terdokumentasi, efisien, dan transparan.

Untuk mengatasi permasalahan di atas, sistem informasi yang dirancang diharapkan dapat memenuhi beberapa kebutuhan penting berikut:

1. Pencatatan pesanan secara digital

Sistem harus mampu mencatat dan mendokumentasikan seluruh pesanan layanan secara terstruktur, mulai dari data pelanggan, jenis layanan yang diminta, hingga jadwal pengkerjaan teknisi.

2. Integrasi data pelanggan dan teknisi

Sistem informasi yang diperlukan untuk menyimpan semua data terkait pelanggan, petugas, jadwal, dan layanan riwayat, sehingga memudahkan pengelolaan operasional dan pencarian data ketika dibutuhkan.

3. Fitur layanan pelacakan status

Sistem harus memberikan akses bagi pelanggan untuk menampilkan kemajuan pengkerjaan layanan secara real-time melalui antarmuka berbasis web yang dapat diakses menggunakan komputer maupun perangkat mobile.

4. Notifikasi otomatis

Sistem juga perlu dilengkapi dengan fitur notifikasi (misalnya melalui email atau *WhatsApp API*) untuk memberikan pengingat jadwal servis kepada pelanggan serta update status layanan.

3.6 Perancangan Sistem

Perancangan sistem dilakukan sebagai tahap lanjutan setelah analisis kebutuhan untuk menghasilkan rencana sistem informasi layanan bengkel AC yang sesuai dengan permasalahan yang telah diidentifikasi di Bengkel AC Adi Jaya Teknik. Tahap ini bertujuan untuk menggambarkan rancangan sistem secara detail, mulai dari alur kerja, struktur data, hingga rancangan antarmuka pengguna yang akan digunakan sebagai acuan dalam pembangunan sistem.

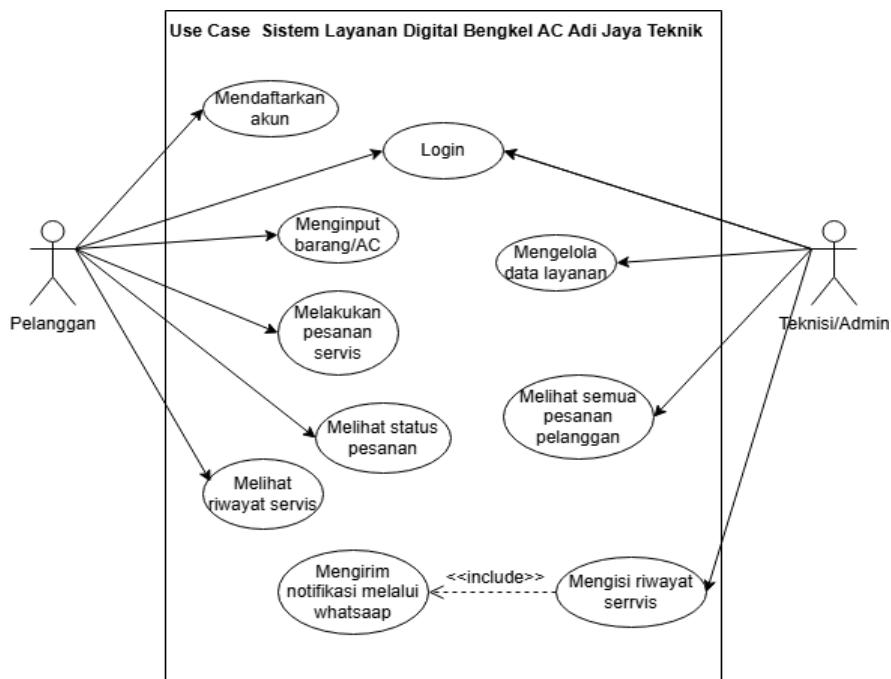
Perancangan sistem ini dirancang berbasis web agar dapat diakses melalui perangkat desktop maupun *mobile* browser, sehingga memberikan harapan bagi pelanggan maupun pihak bengkel dalam mengakses layanan. Sistem informasi ini akan memfasilitasi pencatatan pesanan secara digital, integrasi data pelanggan dan teknisi, pelacakan status servis secara real-time, hingga layanan notifikasi untuk meningkatkan kepuasan pelanggan.

3.6.1 Use Case Diagram

Pada tahap perancangan sistem, *Use Case Diagram* menjadi salah satu alat penting untuk memvisualisasikan kebutuhan sistem secara menyeluruh. Diagram ini menggambarkan hubungan antara aktor (pengguna sistem) dengan fitur-fitur yang ada dalam sistem informasi layanan bengkel AC yang akan dikembangkan.

Aktor dalam sistem ini terdiri dari Admin/Teknisi dan Pelanggan. Admin/Teknisi bertanggung jawab terhadap pengelolaan operasional, seperti pencatatan pesanan, manajemen jadwal layanan, dan pembaruan status pengerjaan. Sementara itu, Pelanggan sebagai pengguna akhir dapat mengakses fitur pemesanan layanan, melihat riwayat layanan, hingga memantau status pengerjaan servis secara real-time. Use Case Diagram ini berfungsi untuk:

1. Memetakan alur interaksi antara pengguna dan sistem.
2. Menunjukkan batasan sistem (system boundary) yang akan dikembangkan.
3. Mempermudah tim pengembang dalam memahami fungsionalitas sistem berdasarkan kebutuhan pengguna.



Gambar 3. 6 Use case Diagram

Diagram Use Case yang ditampilkan mencakup fitur-fitur utama yang relevan dengan kebutuhan layanan bengkel AC, mulai dari pendaftaran akun, pengelolaan

data layanan, hingga notifikasi otomatis melalui WhatsApp. Diagram ini juga menggambarkan batasan peran masing-masing aktor dalam menggunakan sistem berikut penjelasannya:

1. Login

Aktor: Pelanggan, Teknisi/Admin

Deskripsi: Fungsionalitas ini merupakan gerbang utama untuk mengakses sistem. Setiap aktor yang terdaftar harus melakukan autentikasi dengan memasukkan *username* dan *password* yang valid. Sistem akan memverifikasi kredensial tersebut untuk memastikan keamanan data dan memberikan akses ke fitur-fitur yang sesuai dengan peran masing-masing aktor (Pelanggan atau Admin).

Fungsionalitas Aktor Pelanggan

2. Mendaftarkan Akun

Aktor: Pelanggan

Deskripsi: Use case ini memungkinkan pengguna baru untuk membuat akun pribadi di dalam sistem. Pelanggan akan mengisi formulir pendaftaran yang berisi data diri seperti nama lengkap, alamat, nomor telepon, dan membuat *username* serta *password*. Tujuannya adalah untuk mendokumentasikan data pelanggan secara digital agar dapat digunakan untuk proses pemesanan di masa depan.

3. Menginput Barang/AC

Aktor: Pelanggan

Deskripsi: Fungsionalitas ini memungkinkan pelanggan untuk mendaftarkan aset atau unit elektronik miliknya (seperti AC, kulkas, dll.) ke dalam sistem. Pelanggan akan memasukkan detail seperti jenis barang, merek, dan tipe model. Data ini akan tersimpan di akun pelanggan dan dapat dipilih dengan mudah saat akan melakukan pemesanan servis, sehingga proses menjadi lebih cepat dan akurat.

4. Melakukan Pesanan Servis

Aktor: Pelanggan

Deskripsi: Ini adalah use case inti bagi pelanggan. Fungsionalitas ini memungkinkan pelanggan untuk membuat permintaan layanan servis secara online. Pelanggan akan memilih barang yang akan diservis, jenis layanan yang dibutuhkan, dan menentukan jadwal yang diinginkan melalui formulir pemesanan. Proses ini menggantikan metode pemesanan manual dan memastikan semua permintaan tercatat secara terstruktur.

5. Melihat Status Pesanan

Aktor: Pelanggan

Deskripsi: Fitur ini memberikan transparansi kepada pelanggan dengan memungkinkan mereka untuk melacak progres penggerjaan dari pesanan yang sedang aktif. Sistem akan menampilkan status terkini dari sebuah pesanan, misalnya "Pending", "Proses", atau "Selesai", sehingga pelanggan selalu mengetahui perkembangan layanannya.

6. Melihat Riwayat Servis

Aktor: Pelanggan

Deskripsi: Fungsionalitas ini berfungsi sebagai arsip digital bagi pelanggan. Pelanggan dapat mengakses daftar lengkap dari semua layanan servis yang pernah mereka pesan dan telah selesai dikerjakan. Riwayat ini mencakup detail seperti tanggal, jenis layanan, dan catatan teknisi, yang berguna untuk referensi perawatan di masa depan.

Fungsionalitas Aktor Teknisi/Admin

7. Mengelola Data Layanan

Aktor: Teknisi/Admin

Deskripsi: Ini adalah fungsi administratif yang memungkinkan Admin untuk mengelola data master layanan yang ditawarkan oleh bengkel. Admin dapat menambah, mengubah, atau menghapus jenis layanan beserta harganya. Fitur ini memastikan bahwa daftar layanan yang dapat dipilih oleh pelanggan selalu relevan dan sesuai dengan kebijakan bisnis.

8. Melihat Semua Pesanan Pelanggan

Aktor: Teknisi/Admin

Deskripsi: Fungsionalitas ini memberikan akses kepada Admin untuk melihat daftar keseluruhan pesanan yang masuk dari semua pelanggan. Ini berfungsi sebagai pusat pemantauan untuk mengelola antrian pekerjaan, melihat pesanan yang masih berstatus "Pending", dan mendapatkan gambaran umum mengenai aktivitas operasional harian.

9. Mengisi Riwayat Servis

Aktor: Teknisi/Admin

Deskripsi: Setelah sebuah pekerjaan servis selesai dilakukan, Admin atau Teknisi menggunakan fitur ini untuk mendokumentasikan detail penggerjaan. Mereka akan mengubah status pesanan menjadi "Selesai" dan menambahkan catatan teknis, seperti komponen yang diganti atau rekomendasi perawatan. Proses ini sangat penting untuk melengkapi data historis dari sebuah layanan.

10. Mengirim Notifikasi melalui WhatsApp

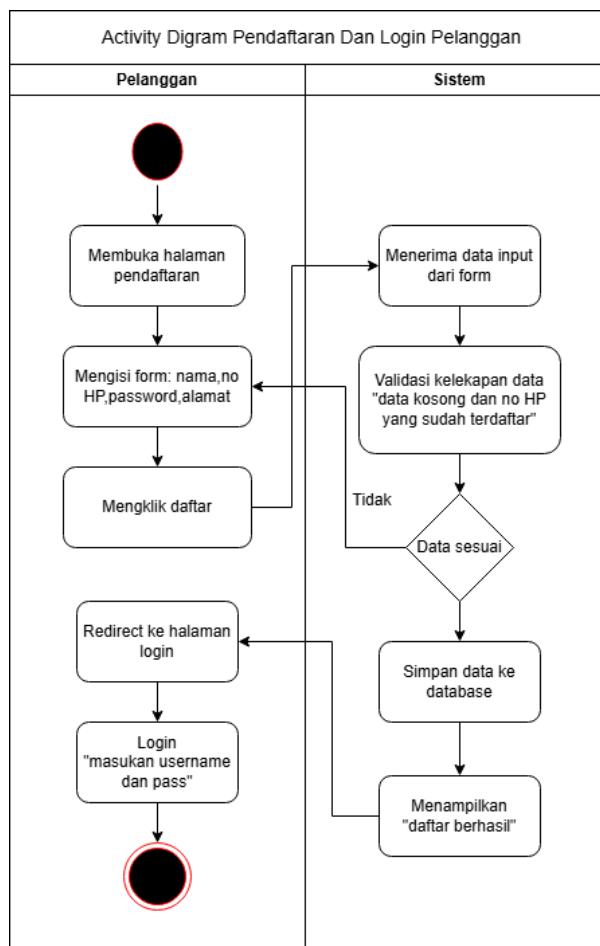
Aktor: (Sistem, dipicu oleh Admin)

Deskripsi: Use case ini merepresentasikan kemampuan sistem untuk mengirimkan pemberitahuan secara otomatis kepada pelanggan. Seperti yang ditunjukkan oleh relasi <<include>>, fungsionalitas ini wajib dijalankan setiap kali Admin menyelesaikan sebuah pekerjaan melalui use case Mengisi Riwayat Servis. Tujuannya adalah untuk memberikan konfirmasi dan informasi status terbaru kepada pelanggan secara proaktif.

3.6.2 *Activity Diagram* Pendaftaran dan *Login* Pelanggan

Activity Diagram pendaftaran dan *login* pelanggan ini menggambarkan alur aktivitas yang terjadi ketika pengguna baru ingin membuat akun pada sistem layanan bengkel AC berbasis web, hingga proses *login* setelah akun berhasil dibuat. Proses ini dibagi menjadi dua bagian utama, yaitu aktivitas pelanggan dan aktivitas sistem, yang digambarkan dengan swimlane untuk menunjukkan peran masing-masing.

Proses dimulai ketika pelanggan membuka halaman pendaftaran pada aplikasi web. Selanjutnya, pelanggan mengisi form pendaftaran dengan data yang diperlukan, seperti nama lengkap, nomor handphone, password, dan alamat. Setelah semua data diinput, pelanggan menekan tombol “Daftar” untuk mengirimkan data ke sistem.



Gambar 3. 7 Activity Diagram Pendaftaran dan Login Pelanggan

Sistem kemudian memproses data yang diterima dengan langkah berikut:

1. Menerima data input dari form.
2. Melakukan validasi data untuk memastikan semua kolom terisi dengan lengkap dan nomor handphone belum pernah terdaftar.
 - a) Jika ditemukan data tidak lengkap atau nomor HP sudah terdaftar, sistem akan menolak pendaftaran dan mengembalikan pelanggan ke halaman form untuk memperbaiki data yang salah.

- b) Jika data sudah valid, sistem akan melanjutkan ke tahap berikutnya.
- 3. Menyimpan data ke dalam *database* sebagai akun baru.
- 4. Menampilkan pesan notifikasi “Pendaftaran Berhasil” kepada pelanggan.

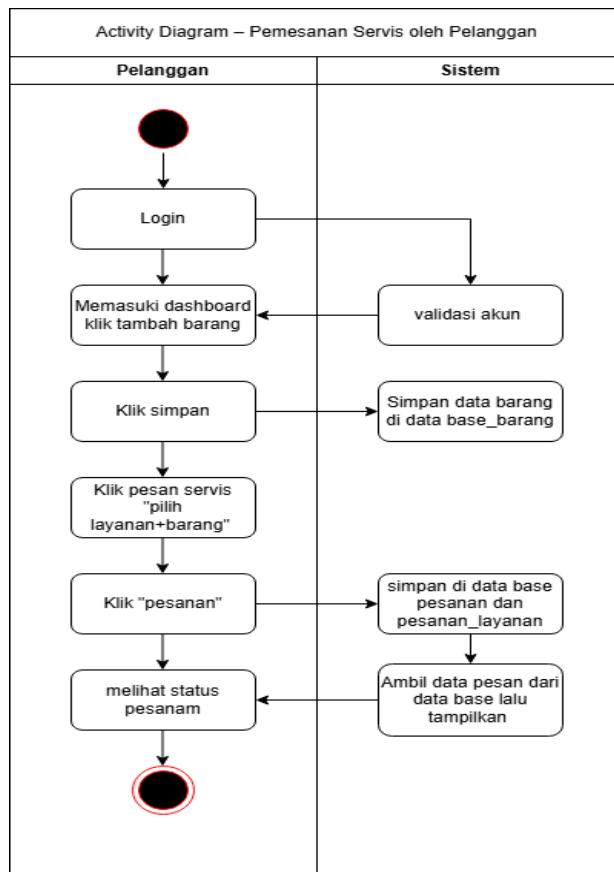
Setelah proses pendaftaran berhasil, pelanggan diarahkan otomatis ke halaman login. Di halaman ini, pelanggan memasukkan username (nomor handphone atau email) dan password yang telah didaftarkan untuk melakukan proses *login* ke sistem. Jika *login* berhasil, pelanggan dapat mengakses seluruh fitur layanan yang tersedia. Diagram ini membantu pengembang memahami alur kerja awal pengguna dengan sistem, sehingga mempermudah dalam tahap implementasi dan pengujian sistem nantinya.

3.6.3 *Activity Diagram* Pemesanan Servis oleh Pelanggan

Activity Diagram ini menggambarkan alur proses ketika pelanggan melakukan pemesanan layanan servis melalui sistem berbasis web. Diagram ini dibagi menjadi dua swimlane utama, yaitu Pelanggan dan Sistem, yang menunjukkan peran masing-masing dalam proses pemesanan. Diagram ini dirancang untuk mempermudah pelanggan dalam mendaftarkan unit (barang) yang ingin dilayani sekaligus melakukan pemesanan layanan, sehingga seluruh proses menjadi lebih cepat, praktis, dan terstruktur.

Proses dimulai setelah pelanggan berhasil *login* ke dalam sistem. *Login* diperlukan sebagai langkah autentikasi untuk memastikan hanya pengguna terdaftar yang dapat mengakses layanan. Setelah *login*, pelanggan diarahkan ke halaman dashboard. Di sini, pelanggan dapat memilih untuk menambahkan barang (misalnya, AC atau perangkat lainnya) yang akan diservis dengan mengisi detail data barang. Setelah data barang diinput, pelanggan menekan tombol “Simpan” agar data tersebut tersimpan di *database* sistem pada tabel *database_barang*. Berikutnya, pelanggan melanjutkan dengan memilih menu “Pesanan Servis”. Pada tahap ini, pelanggan diminta memilih jenis layanan yang diinginkan serta barang yang sudah didaftarkan sebelumnya. Setelah memilih layanan dan

barang, pelanggan menekan tombol “Pesanan” untuk mengirimkan permintaan servis ke sistem.



Gambar 3. 8 Activity Diagram Pemesanan Servis oleh Pelanggan

Di sisi sistem, langkah-langkah yang dilakukan meliputi:

1. Start

Proses dimulai ketika pelanggan *login* ke dalam sistem menggunakan akun yang telah terdaftar.

2. Dashboard Pelanggan

Setelah *login* berhasil, pelanggan diarahkan ke dashboard utama. Pada dashboard ini terdapat dua opsi: Tambah Barang dan Pesan Layanan .

3. Pilih Menu Pesan Layanan

Pelanggan memilih menu *Pesan Layanan* untuk membuat pesanan servis.

4. Cek Data Unit Pelanggan

Sistem akan memeriksa apakah pelanggan sudah memiliki data unit (AC, kulkas, atau perangkat lain) yang pernah didaftarkan sebelumnya.

5. Apakah Unit Sudah Terdaftar?

Jika YA: Sistem menampilkan daftar unit milik pelanggan.

Jika TIDAK: Pelanggan diarahkan ke form *Tambah Barang*.

6. Tambah Barang (Jika Diperlukan)

Pelanggan mengisi data barang seperti nama unit, tipe, tahun pembelian, dan catatan tambahan. Setelah data disimpan, sistem menambahkan unit ke *database*.

7. Pilih Unit

Pelanggan memilih unit yang akan dilayani dari daftar unit yang tersedia.

8. Isi Data Pemesanan

Pelanggan mengisi detail pemesanan seperti jenis layanan (servis, bongkar pasang, dll.), jadwal yang diinginkan, dan catatan tambahan.

9. Konfirmasi Pemesanan

Sistem menampilkan ringkasan pemesanan untuk dikonfirmasi oleh pelanggan.

10. Simpan & Kirim Pesanan

Setelah konfirmasi, sistem menyimpan data pemesanan ke *database* dan mengirimkan notifikasi ke admin/teknisi.

11. Akhir

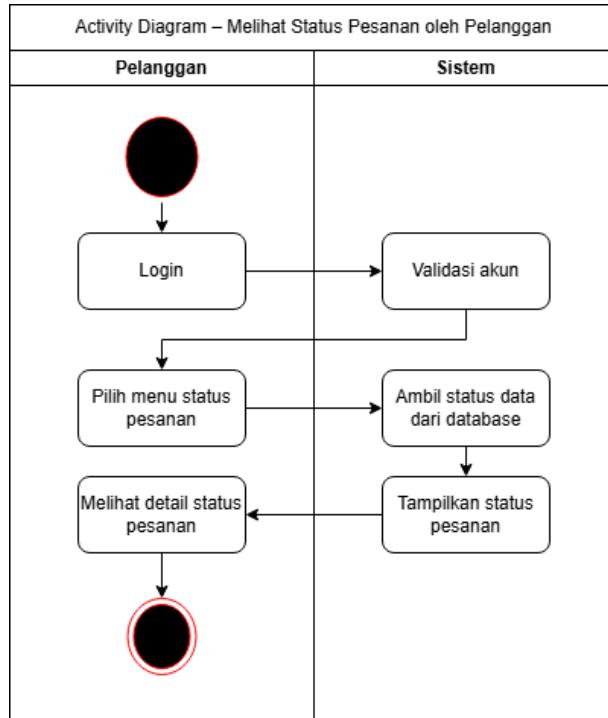
Proses pemesanan layanan selesai.

Setelah seluruh proses selesai, pelanggan dapat melihat status pesanannya secara langsung pada halaman status, yang memperlihatkan apakah pesanan sedang diproses, ditunda, atau telah selesai. Diagram ini membantu memastikan bahwa seluruh alur pemesanan dapat ditangani secara efisien dan terdokumentasi dengan baik oleh sistem, sekaligus memberikan transparansi bagi pelanggan terhadap status layanan yang mereka pesan.

3.6.4 Activity Diagram Melihat Status Pesanan oleh Pelanggan

Activity Diagram ini menjelaskan proses alur kerja ketika pelanggan ingin memeriksa status pesanan servis yang telah dibuat sebelumnya. Diagram ini

menggambarkan interaksi antara Pelanggan dan Sistem , mulai dari *login* hingga informasi status pesanan yang ditampilkan.



Gambar 3. 9 Activity Diagram Melihat Status Pesanan oleh Pelanggan

Proses ini dirancang untuk memberikan transparansi kepada pelanggan sehingga mereka dapat menampilkan kemajuan pengerjaan layanan secara real-time.berikut penjelasan alur proses :

1. *Login*

Pelanggan melakukan *login* ke sistem dengan memasukkan username dan password. Jika validasi berhasil, pelanggan diarahkan ke dashboard utama.

2. *Pilih Menu Status Pesanan*

Setelah masuk ke dashboard, pelanggan memilih opsi *Lihat Status Pesanan* untuk mengetahui progres layanan yang dipesan.

3. *Permintaan Data ke Sistem*

Sistem menerima permintaan dari pelanggan dan memprosesnya dengan melakukan validasi data akun serta mencari data status pesanan yang sesuai di *database*.

4. *Pengambilan Data Status*

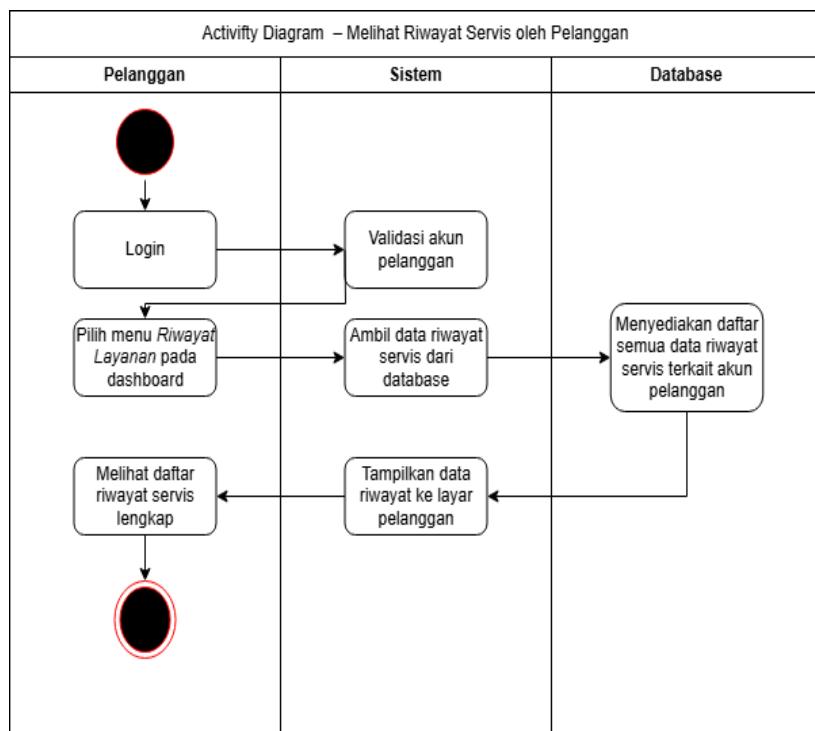
Sistem mengambil data status pesanan dari *database*, termasuk informasi seperti tahap pengerjaan (*Pending* , *Sedang Dikerjakan* , *Selesai*), tanggal pengerjaan, dan teknisi yang bertugas.

5. Tampilkan Data ke Pelanggan

Sistem menampilkan status pesanan secara lengkap kepada pelanggan melalui antarmuka web. Informasi ini memungkinkan pelanggan untuk mengetahui kondisi terkini pesanan mereka tanpa harus menghubungi teknisi secara manual.

3.6.5 *Activity Diagram Melihat Riwayat Servis*

Activity Diagram ini menggambarkan alur proses ketika pelanggan ingin melihat riwayat layanan servis yang pernah mereka lakukan di Bengkel AC Adi Jaya Teknik. Proses ini dirancang agar pelanggan dapat mengakses informasi dengan cepat dan mudah melalui sistem berbasis web. Dengan fitur ini, pelanggan tidak perlu lagi menanyakan status atau riwayat secara manual kepada teknisi karena semua data sudah tersedia secara digital.



Gambar 3. 10 *Activity Diagram Melihat Riwayat Servis oleh Pelanggan*

Alur Proses:

- 1. *Login Pelanggan***

Pelanggan melakukan *login* ke sistem dengan memasukkan username dan password. Sistem akan memverifikasi kredensial dan mengarahkan pelanggan ke dashboard jika validasi berhasil.

- 2. *Pilih Menu Riwayat Layanan***

Setelah berhasil masuk ke dashboard, pelanggan memilih menu *Riwayat Layanan* untuk mengakses daftar layanan yang pernah mereka gunakan.

- 3. *Sistem Memproses pengajuan***

Sistem secara otomatis mengambil data riwayat servis milik pelanggan dari *database* tanpa memerlukan permintaan tambahan. Proses ini dilakukan secara *backend* agar lebih cepat dan lancar bagi pengguna.

- 4. *Tampilkan Daftar Riwayat Layanan***

Sistem menampilkan daftar lengkap riwayat layanan kepada pelanggan melalui antarmuka web. Informasi yang meliputi ditampilkan:

- a) Tanggal layanan
- b) Jenis layanan
- c) Layanan Status (Selesai, Pending, Dalam Proses)
- d) Keterangan teknisi yang bertugas

- 5. *Pelanggan Melihat Detail Riwayat Riwayat (Opsional)***

Jika pelanggan ingin mengetahui detail lebih lanjut, mereka dapat memilih salah satu riwayat servis lebih lanjut untuk melihat informasi lengkapnya.

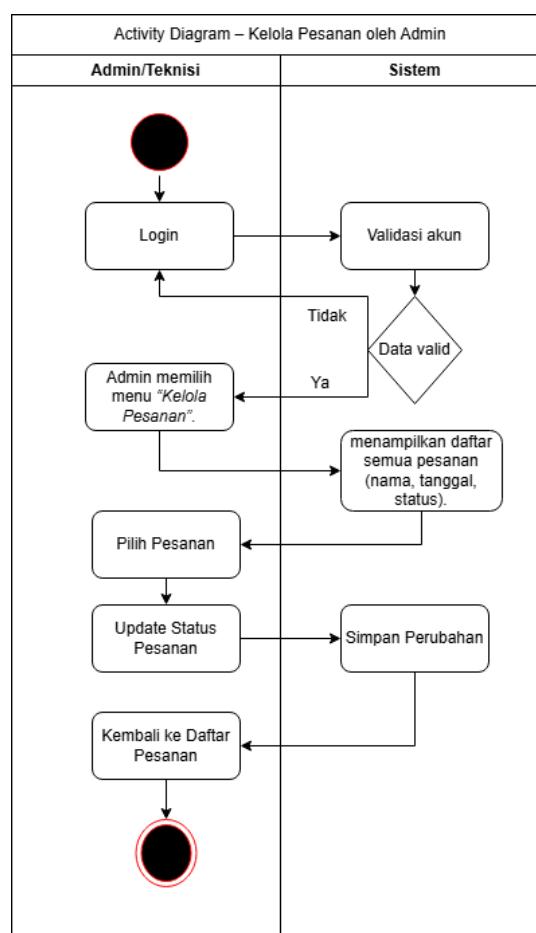
Dengan adanya fitur ini, transparansi layanan meningkat, pelanggan lebih mengambil keputusan untuk perawatan lanjutan.

3.6.6 *Activity Diagram Kelola Pesanan oleh Admin*

Dalam perancangan sistem informasi ini, alur kerja untuk aktor Admin/Teknisi sengaja dipisahkan menjadi tiga diagram aktivitas yang berbeda meskipun saling terkait. Pemisahan ini didasarkan pada prinsip pemisahan fungsi (*separation of concerns*) untuk menciptakan alur kerja yang lebih modular, efisien, dan logis. Setiap diagram merepresentasikan sebuah konteks pekerjaan yang

spesifik, yaitu manajemen operasional, dokumentasi purna-layanan, dan manajemen relasi pelanggan.

Diagram ini berfungsi sebagai pusat kendali operasional yang berfokus pada manajemen siklus hidup pesanan (*order lifecycle management*) secara *real-time*. Tujuannya adalah untuk merefleksikan progresi pekerjaan di lapangan secara taktis dan efisien. Perubahan status menjadi "Selesai" dalam alur ini merupakan penanda operasional bahwa pekerjaan fisik telah rampung, yang kemudian menjadi pemicu untuk alur kerja administratif selanjutnya.



Gambar 3. 11 Activity Diagram Kelola Pesanan oleh Admin

Alur Proses:

1. **Login & Validasi Akun**

Sistem memverifikasi kredensial yang dimasukkan. Jika data *login* benar, admin diarahkan ke dashboard. Jika salah, sistem menampilkan pesan error dan meminta input ulang.

2. Dashboard Admin

Setelah berhasil *login*, admin masuk ke dashboard utama yang menampilkan berbagai menu. Admin memilih menu “*Kelola Pesanan*” untuk melihat daftar semua pesanan yang masuk.

3. Tampilkan Daftar Pesanan

Sistem menampilkan daftar lengkap pesanan pelanggan, meliputi nama pelanggan, tanggal pemesanan, dan status terkini (Pending, Proses, Selesai).

4. Pilih Pesanan

Admin memilih salah satu pesanan yang ingin diproses lebih lanjut.

5. Update Status Pesanan

Admin memperbarui status pesanan sesuai progres penggerjaan, misalnya:

Pending → Proses

Proses → Selesai

Admin juga dapat menambahkan catatan seperti “Teknisi dalam perjalanan” untuk informasi tambahan.

6. Simpan Perubahan

Setelah melakukan perubahan, sistem menyimpan data tersebut ke dalam *database* dan secara otomatis mengirimkan notifikasi ke pelanggan (melalui web atau WhatsApp).

7. Kembali ke Daftar Pesanan

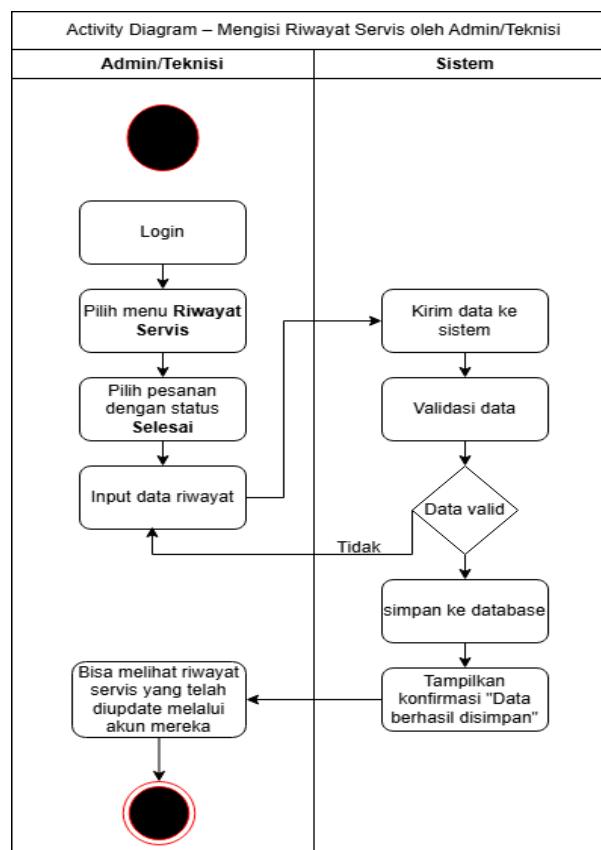
Admin kembali ke daftar pesanan untuk memproses pesanan lain jika ada.

Dengan adanya fitur ini, admin dapat memastikan semua pesanan tercatat dengan baik, progres penggerjaan terpantau, dan pelanggan selalu mendapatkan informasi terkini tentang status layanan mereka. Hal ini mendukung efisiensi operasional bengkel serta meningkatkan kepuasan pelanggan.

3.6.7 *Activity Diagram* Mengisi Riwayat Servis oleh Admin/Teknisi

Diagram ini memodelkan alur kerja administratif yang terpisah, yaitu proses dokumentasi purna-layanan. Alur ini secara spesifik menangani pencatatan data historis dan teknis setelah sebuah pesanan ditandai “Selesai”. Pemisahan ini bertujuan untuk menjamin integritas dan kelengkapan data dokumentasi, karena

prosesnya dilakukan dalam konteks yang berbeda dari manajemen operasional harian. Fungsinya adalah untuk membangun arsip teknis yang akurat untuk setiap pekerjaan yang telah diselesaikan.



Gambar 3. 12 Activity Diagram Mengisi Riwayat Servis oleh Admin

Alur Proses

1. **Login**

Admin/Teknisi memasukkan username dan password untuk mendapatkan akses ke sistem dashboard.

2. **Pilih Menu Riwayat Layanan**

Setelah berhasil *login*, mereka memilih menu *Riwayat Layanan* di dashboard untuk menginput data pekerjaan yang telah selesai.

3. **Pilih Pesanan Selesai**

Admin/Teknisi memilih salah satu pesanan yang statusnya telah selesai dari daftar pesanan yang ditampilkan sistem.

4. Input Data Riwayat Layanan

Admin/Teknisi mengisi detail riwayat servis, termasuk:

- a) Nama pelanggan
- b) Jenis layanan yang dilakukan (misalnya: servis AC, pemasangan CCTV)
- c) Tanggal penggeraan
- d) Catatan tambahan tentang kondisi teknis atau rekomendasi untuk pelanggan

5. Kirim Data ke Sistem

Setelah semua data terisi, Admin/Teknisi menekan tombol *Simpan*.

6. Validasi Data oleh Sistem

Sistem memvalidasi input data.

Jika data valid → Data disimpan ke *database*.

Jika data tidak valid → Sistem menampilkan pesan kesalahan untuk perbaikan.

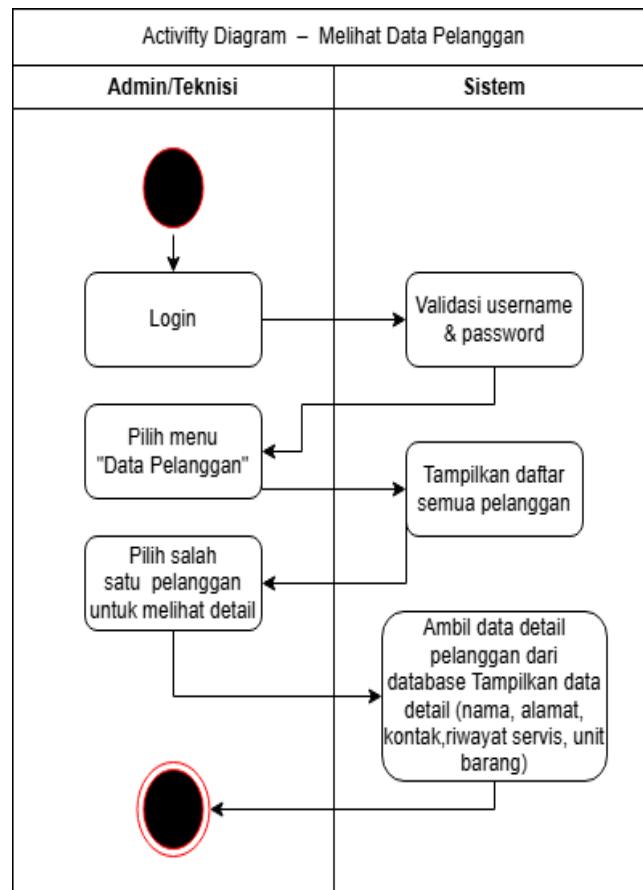
7. Konfirmasi Penyimpanan

Sistem memberikan notifikasi bahwa riwayat servis berhasil disimpan.

3.6.8 *Activity Diagram Melihat Data Pelanggan*

Alur kerja ini dirancang sebagai fungsi *Customer Relationship Management* (CRM) yang terpusat pada entitas pelanggan, bukan pada entitas pesanan. Titik masuk (*entry point*) dari proses ini adalah identifikasi pelanggan, yang memungkinkan admin untuk mengakses pandangan holistik—mencakup profil, daftar aset (barang), dan seluruh riwayat transaksinya. Tujuannya adalah untuk mendukung fungsi layanan pelanggan di luar konteks pengelolaan satu pesanan aktif.

Activity Diagram ini menjelaskan alur proses ketika Admin/Teknisi ingin melihat data pelanggan pada sistem. Fitur ini penting untuk mendukung pengelolaan informasi pelanggan secara ringkas, sehingga memudahkan teknisi dalam melayani permintaan servis, mengetahui riwayat layanan, serta mengelola komunikasi dengan pelanggan.



Gambar 3. 13 Activity Diagram Melihat Data Pelanggan

Penjelasan Proses:

1. Start

Proses dimulai ketika Admin/Teknisi login ke dalam sistem menggunakan username dan password yang valid.

2. Validasi Login

Sistem memeriksa keabsahan data *login*. Jika data valid, pengguna diarahkan ke halaman dashboard. Jika tidak, sistem menampilkan pesan kesalahan.

3. Menu Akses Data Pelanggan

Admin memilih menu “Data Pelanggan” pada dashboard untuk mengelola informasi pelanggan.

4. Tampilkan Daftar Pelanggan

Sistem yang menampilkan daftar semua pelanggan beserta informasi singkat seperti nama, kontak, dan alamat.

5. Pilih Detail Pelanggan

Admin dapat memilih salah satu pelanggan untuk melihat detail lengkap, termasuk riwayat servis, unit barang, dan catatan lainnya.

6. Tampilkan Detail Pelanggan

Sistem mengambil data dari *database* dan menampilkan informasi lengkap tersebut kepada Admin/Teknisi.

7. End

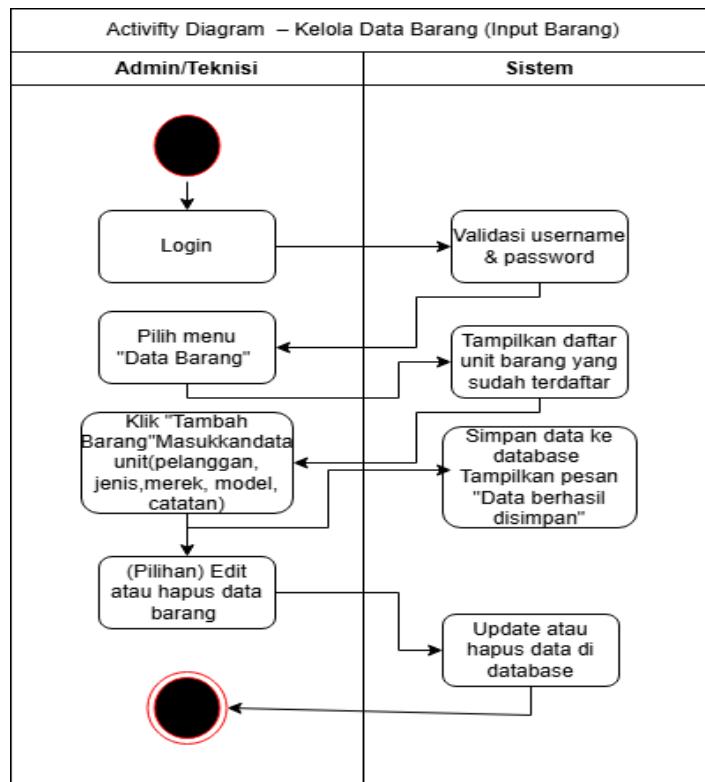
Proses berakhir setelah Admin/Teknisi selesai memeriksa atau mengelola data pelanggan.

Fitur ini membantu memastikan setiap layanan yang diberikan dapat dilacak kembali secara mudah, sekaligus meningkatkan transparansi data bagi pihak bengkel.

Dengan demikian, dekomposisi alur kerja Admin menjadi beberapa diagram aktivitas yang berbeda bukan merupakan redundansi, melainkan sebuah keputusan desain yang strategis untuk memastikan setiap fungsi dapat berjalan secara optimal sesuai dengan tujuan spesifiknya.

3.6.9 *Activity Diagram Kelola Data Barang*

Activity Diagram ini menggambarkan proses ketika Admin/Teknisi mengelola data barang milik pelanggan. Fitur ini memungkinkan admin untuk menambahkan unit barang (AC, kulkas, dll.) yang dilayani, sehingga setiap layanan dapat terhubung langsung ke unit barang tertentu milik pelanggan.



Gambar 3. 14 Activity Diagram Kelola Data Barang

Penjelasan Proses:

1. Start

Proses dimulai dengan *login* Admin/Teknisi ke dalam sistem menggunakan akun yang terdaftar.

2. Validasi *Login*

Sistem memeriksa keabsahan *login* dan mengarahkan ke dashboard jika data valid. Jika tidak valid, sistem menampilkan notifikasi error.

3. Menu Akses Data Barang

Admin/Teknisi memilih menu “Data Barang” untuk mengelola unit informasi milik pelanggan.

4. Tampilkan Daftar Barang

Sistem menampilkan daftar semua barang milik pelanggan yang telah diinput sebelumnya.

5. Tambah Data Barang

Admin memilih opsi “Tambah Barang” untuk memasukkan unit data baru.

Data yang diinput meliputi:

- a) Nama pelanggan
- b) Jenis barang (AC, kulkas, dll.)
- c) Merek, model, nomor seri
- d) Catatan tambahan jika ada

6. Simpan Data Barang

Sistem menyimpan data ke *database* dan menampilkan notifikasi bahwa data berhasil disimpan.

7. Edit/Hapus Data Barang (*Opsional*)

Admin juga dapat memperbarui atau menghapus data barang jika ada perubahan atau data tidak lagi relevan.

8. End

Proses berakhir setelah semua data barang diperbarui sesuai kebutuhan.

Fitur ini sangat penting untuk memastikan keterpaduan data antara pelanggan, unit barang yang dilayani, dan riwayat layanan.

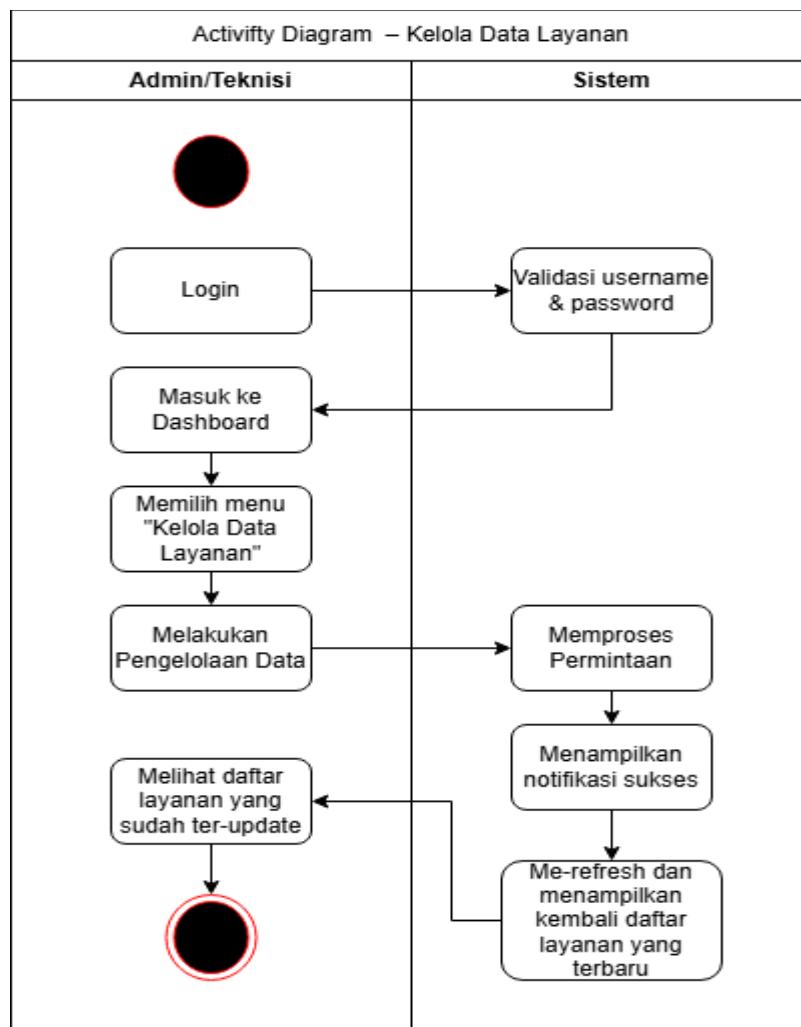
3.6.10 Activity Diagram Kelola Data Layanan

Activity Diagram ini memodelkan alur kerja administratif untuk pengelolaan data master layanan dari perspektif tingkat tinggi (*high-level*). Berbeda dengan diagram yang merinci setiap cabang logika, diagram ini sengaja menyederhanakan beberapa aksi (seperti Tambah, Edit, dan Hapus) menjadi satu aktivitas umum.

Tujuannya adalah untuk memberikan gambaran yang jelas dan ringkas mengenai alur proses utama tanpa terlalu teknis. Dengan demikian, fokus tetap pada interaksi fundamental antara Admin dan Sistem dalam mengelola data layanan yang ditawarkan oleh bengkel.

Aktor & Komponen Terlibat:

- a) Aktor: Admin/Teknisi
- b) Komponen: Sistem (Antarmuka dan Proses *Backend*)



Gambar 3. 15 Activity Diagram Kelola Data Layanan

Alur Proses Berdasarkan Diagram:

1. **Inisiasi Proses:** Alur dimulai saat Admin/Teknisi berhasil login ke sistem dan memilih menu "Kelola Data Layanan" dari halaman dashboard.
2. **Penyajian Data Awal:** Sistem merespons dengan menampilkan sebuah halaman yang berisi daftar semua data layanan yang telah terdaftar sebelumnya.
3. **Aksi Pengelolaan oleh Admin:** Pada halaman ini, Admin melakukan aksi pengelolaan data. Aktivitas umum ini mencakup tiga kemungkinan yang

diabstraksikan menjadi satu langkah, yaitu: menambah data layanan baru, mengubah data layanan yang sudah ada, atau menghapus data layanan yang tidak lagi relevan.

4. **Pemrosesan oleh Sistem:** **Sistem** menerima permintaan dari Admin dan memprosesnya di sisi backend. Proses ini secara generik meliputi validasi input dan eksekusi perintah yang sesuai ke database (baik itu INSERT untuk data baru, UPDATE untuk perubahan, maupun DELETE untuk penghapusan).
5. **Konfirmasi dan Umpam Balik:** Setelah pemrosesan di database berhasil, **Sistem** memberikan notifikasi visual (misalnya: "Data berhasil disimpan") kepada Admin dan secara otomatis memperbarui (*me-refresh*) tampilan daftar layanan untuk merefleksikan perubahan yang baru saja terjadi.
6. **Penyelesaian:** Admin melihat daftar layanan yang telah ter-update, dan alur kerja untuk sesi pengelolaan tersebut dianggap selesai.

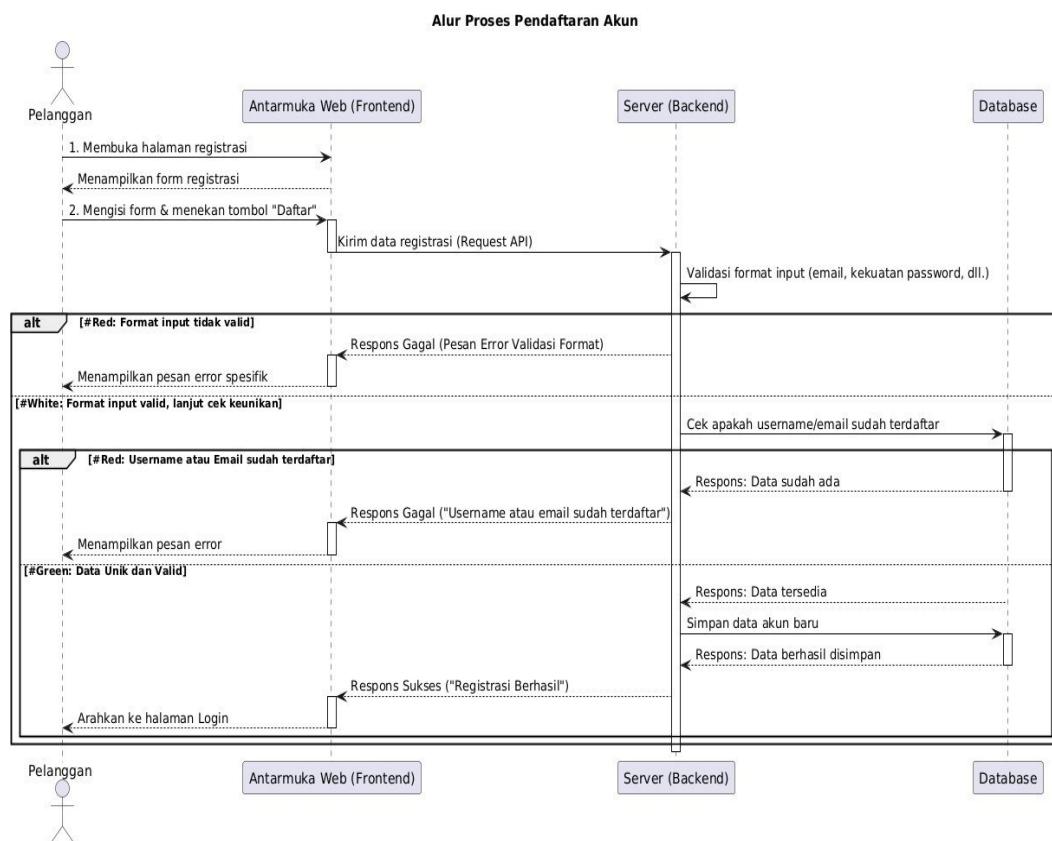
3.6.11 Sequence Diagram Pendaftaran Akun

Sequence Diagram pendaftaran akun ini menggambarkan interaksi antara aktor Pelanggan dengan sistem saat melakukan proses registrasi pada layanan Bengkel AC Adi Jaya Teknik berbasis web. Diagram ini bertujuan untuk memvisualisasikan alur komunikasi yang terjadi antara pengguna (*frontend*), server *backend*, dan *database* dalam proses pembuatan akun baru, agar seluruh proses registrasi berjalan dengan baik dan terhindar dari data yang duplikat.

Proses dimulai ketika pelanggan mengakses halaman registrasi melalui web browser. Sistem *frontend* akan menampilkan form pendaftaran yang berisi input data penting, seperti nama lengkap, alamat email, nomor telepon, username, dan password. Setelah pelanggan mengisi semua data dan menekan tombol *Daftar*, sistem *frontend* mengirimkan input data tersebut ke *backend* untuk diproses. Sistem *backend* kemudian melakukan validasi terhadap data yang dikirimkan. Validasi ini mencakup pemeriksaan apakah username atau email sudah terdaftar sebelumnya, pengecekan format email, serta kekuatan password. Jika ditemukan data yang tidak

valid, *backend* akan mengirimkan pesan kesalahan ke *frontend* untuk ditampilkan kepada pelanggan.

Apabila semua data valid, sistem *backend* menyimpan informasi akun baru ke dalam *database* dan mengatur status akun sebagai *aktif*. Setelah data berhasil disimpan, *backend* mengirimkan respons sukses ke *frontend*. Sistem *frontend* kemudian menampilkan pesan konfirmasi bahwa registrasi berhasil, sekaligus memberikan opsi bagi pelanggan untuk langsung menuju halaman *login* dan masuk ke akun mereka.



Gambar 3. 16 Sequence Diagram Pendaftaran Akun

Berikut adalah urutan interaksi berdasarkan diagram:

- Inisiasi Pendaftaran:** Proses diawali oleh Pelanggan yang mengakses halaman registrasi melalui UI. Sistem kemudian menampilkan formulir pendaftaran yang harus diisi.
- Pengisian dan Pengiriman Data:** Pelanggan mengisi semua kolom data yang diperlukan (seperti nama, email, password) dan menekan tombol

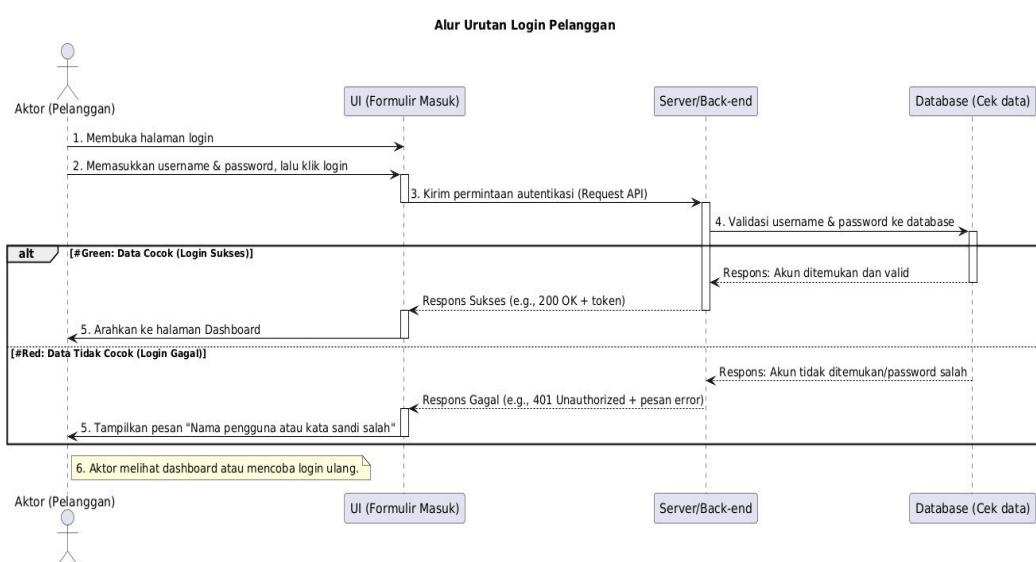
"Daftar". UI kemudian mengumpulkan data tersebut dan mengirimkannya ke Server (*Backend*) untuk diproses.

3. **Validasi Data oleh Server:** Ini adalah langkah krusial yang terdiri dari beberapa tahap validasi:
 - a) **Validasi Format:** Server pertama-tama memeriksa apakah format data yang dikirim sudah benar (misalnya, format email valid, password memenuhi kriteria keamanan). Jika tidak, proses berhenti dan server mengirim pesan error ke UI.
 - b) **Pengecekan Keunikan Data:** Jika format data benar, Server melanjutkan dengan mengirim query ke *Database* untuk memastikan data yang bersifat unik (seperti username atau email) belum pernah terdaftar sebelumnya.
4. **Penanganan Hasil Validasi:** Alur proses memiliki beberapa kemungkinan (digambarkan dalam blok alt bertingkat):
 - a) **Gagal karena Data Tidak Unik:** Jika *Database* menemukan bahwa username atau email sudah ada, ia mengirimkan respons negatif ke Server. Server kemudian meneruskan pesan error ("Username atau email sudah terdaftar") ke UI.
 - b) **Berhasil dan Data Disimpan:** Jika data unik dan valid, *Database* memberikan respons positif ke Server.
5. **Penyimpanan Data:** Setelah semua validasi berhasil, Server mengirimkan perintah ke *Database* untuk membuat entri baru dan menyimpan data pelanggan.
6. **Notifikasi dan Pengalihan:** Setelah *Database* mengonfirmasi bahwa data telah berhasil disimpan, Server mengirimkan respons sukses ke UI. UI kemudian menampilkan pesan "Registrasi Berhasil" dan secara otomatis mengarahkan pelanggan ke halaman *login* untuk dapat masuk dengan akun yang baru dibuat.

3.6.12 Sequence Diagram Login Akun

Sequence Diagram *login* akun ini menggambarkan proses autentikasi yang dilakukan oleh Pelanggan atau Admin/Teknisi saat mengakses sistem informasi layanan Bengkel AC Adi Jaya Teknik. Diagram ini memvisualisasikan langkah-langkah komunikasi antara pengguna, *frontend*, *backend*, dan *database* untuk memastikan hanya akun yang terdaftar dan valid yang dapat mengakses sistem.

Proses dimulai ketika pengguna membuka halaman *login* pada aplikasi web dan memasukkan username serta password ke dalam form *login*. Sistem *frontend* kemudian mengirimkan data autentikasi tersebut ke *backend* untuk diproses. *backend* menerima permintaan *login* dan memvalidasi data dengan memeriksa *database* apakah username yang dimasukkan ada, dan jika ada, apakah password yang dikirimkan sesuai dengan data yang tersimpan. Jika data tidak valid (contohnya username salah atau password tidak cocok), *backend* akan mengirimkan respon error ke *frontend* untuk menampilkan pesan kesalahan kepada pengguna, seperti “Username atau Password salah.” Jika data valid, *backend* memberikan respon sukses yang berisi token sesi atau identitas pengguna, kemudian mengarahkan pengguna ke halaman dashboard sesuai perannya (Pelanggan akan diarahkan ke dashboard pelanggan, sedangkan Admin/Teknisi ke dashboard admin).



Gambar 3. 17 Sequence Diagram Login

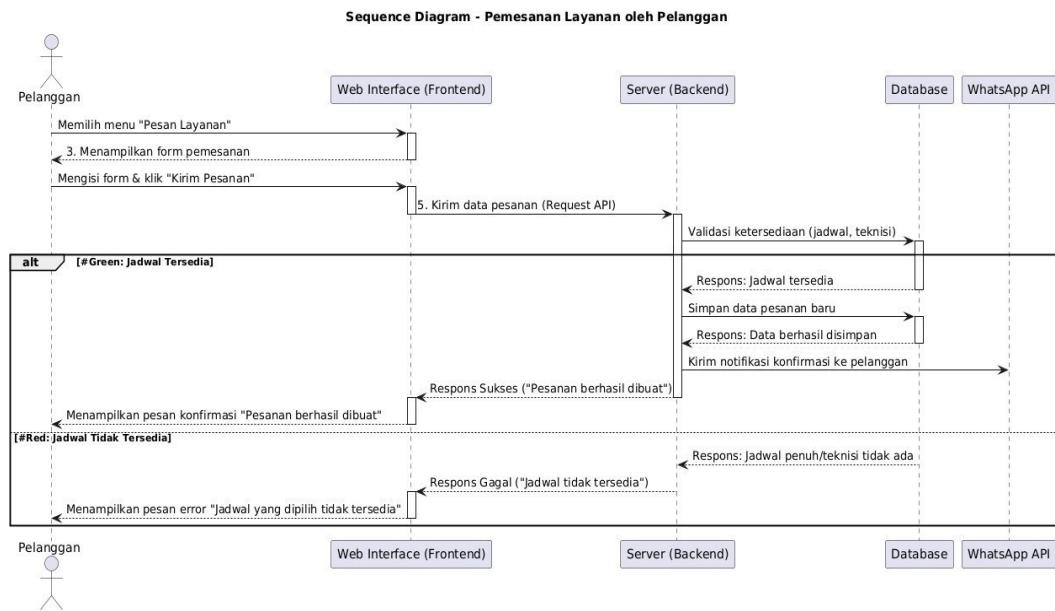
Berikut adalah urutan interaksi berdasarkan diagram:

1. **Inisiasi oleh Pelanggan:** Proses dimulai ketika Pelanggan membuka aplikasi atau web dan berinteraksi dengan UI untuk mengakses halaman *login*. Pelanggan kemudian memasukkan kredensial (username dan password) dan menekan tombol *login*.
2. **Pengiriman Permintaan:** UI menerima input dari pelanggan dan mengirimkan permintaan autentikasi ke Server (*Backend*). Permintaan ini biasanya dikirim melalui panggilan API (Application Programming Interface) dan berisi data kredensial yang telah dimasukkan.
3. **Validasi di Sisi Server:** Server menerima permintaan dan melanjutkan proses dengan melakukan validasi. Server mengirimkan query ke *Database* untuk memeriksa apakah username yang diberikan ada dan apakah password yang menyertainya cocok dengan data yang tersimpan.
4. **Penanganan Respons dari Database:** Proses ini memiliki dua kemungkinan hasil (digambarkan dalam blok alt):
 - a) **Login Berhasil (Jalur Hijau):** Jika *Database* menemukan akun yang cocok dan kredensialnya valid, ia akan mengirimkan respons sukses ke **Server**. Server kemudian membuat sesi untuk pelanggan (misalnya dengan menghasilkan token autentikasi) dan mengirimkan respons sukses kembali ke UI.
 - b) **Login Gagal (Jalur Merah):** Jika *Database* tidak menemukan akun atau jika password salah, ia akan mengirimkan respons gagal ke **Server**. Server kemudian meneruskan respons gagal ini ke UI, biasanya disertai pesan error spesifik (contoh: "Nama pengguna atau kata sandi salah").
5. **Tindakan di Sisi Antarmuka (UI):** UI menerima respons dari server dan bertindak sesuai hasilnya.
 - a) Jika berhasil, UI akan mengarahkan pelanggan ke halaman utama atau dashboard aplikasi.
 - b) Jika gagal, UI akan tetap di halaman *login* dan menampilkan notifikasi error yang diterima dari server.

6. **Hasil Akhir untuk Pelanggan:** Pelanggan akan melihat halaman dashboard jika *login* berhasil, atau melihat pesan error dan dapat mencoba *login* kembali jika gagal.

3.6.13 Sequence Diagram Pemesanan Layanan oleh Pelanggan

Sequence Diagram ini menggambarkan alur interaksi antara Pelanggan, sistem web (*Frontend*), Server *Backend*, dan *Database* saat pelanggan melakukan pemesanan layanan servis melalui aplikasi berbasis web. Proses ini juga melibatkan pengiriman notifikasi kepada pelanggan sebagai konfirmasi bahwa pesanan telah berhasil disimpan.



Gambar 3. 18 Sequence Diagram Pemesanan Layanan

Berikut adalah urutan interaksi berdasarkan diagram:

- Inisiasi Pemesanan:** Proses diasumsikan dimulai setelah pelanggan berhasil *login* dan berada di halaman utama (dashboard). Pelanggan memilih menu "Pesanan Layanan" pada Web Interface (*Frontend*). *Frontend* kemudian merespons dengan menampilkan formulir pemesanan yang relevan.
- Pengiriman Data Pesanan:** Pelanggan mengisi detail yang diperlukan pada formulir (seperti jenis layanan, unit yang akan diservis, serta tanggal dan waktu yang diinginkan), lalu menekan tombol untuk mengirim pesanan.

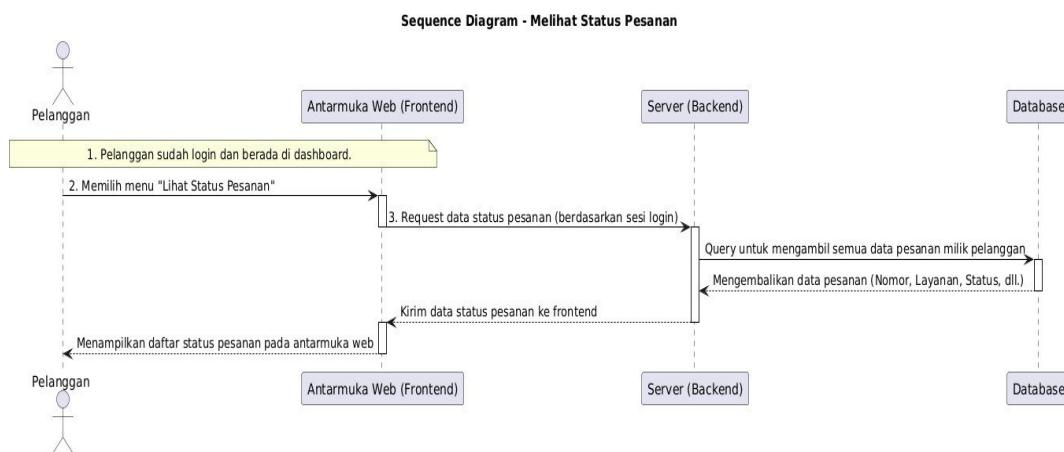
Frontend mengumpulkan semua data ini dan mengirimkannya sebagai sebuah permintaan (API request) ke Server (*Backend*).

3. **Validasi di Sisi Server:** Setelah menerima permintaan, Server memulai proses validasi. Langkah pertama adalah memeriksa ketersediaan sumber daya dengan mengirimkan query ke *Database*. Validasi ini bertujuan untuk memastikan apakah jadwal dan teknisi tersedia pada waktu yang diminta oleh pelanggan.
4. **Penanganan Skenario (Logika Percabangan):** Diagram ini menggunakan blok alt untuk menggambarkan dua kemungkinan hasil utama dari proses validasi:
 - a) **Skenario Sukses (Jalur Hijau - Jadwal Tersedia):** Jika *Database* mengonfirmasi bahwa jadwal tersedia, **Server** melanjutkan alur proses. Ia mengirimkan perintah INSERT ke *Database* untuk menyimpan semua detail pesanan baru.
 - b) **Skenario Gagal (Jalur Merah - Jadwal Tidak Tersedia):** Jika *Database* menginformasikan bahwa jadwal yang diminta sudah penuh atau teknisi tidak tersedia, Server akan menghentikan proses pemesanan. Ia akan menyiapkan respons gagal untuk dikirim kembali ke *frontend*.
5. **Notifikasi dan Respons (Jalur Sukses):** Setelah pesanan berhasil disimpan di *Database*, **Server** melakukan dua tindakan secara paralel atau berurutan:
 - a) Mengirim perintah ke layanan eksternal, dalam hal ini WhatsApp API, untuk mengirimkan notifikasi konfirmasi langsung ke nomor WhatsApp pelanggan.
 - b) Mengirim respons sukses ke *Frontend* yang berisi pesan konfirmasi, seperti "Pesanan berhasil dibuat".
6. **Tampilan Hasil Akhir:** *Frontend* menerima respons dari **Server** dan menampilkannya kepada Pelanggan. Jika berhasil, pelanggan akan melihat pesan konfirmasi di layar. Jika gagal, pelanggan akan melihat pesan error yang menjelaskan mengapa pesanan tidak dapat diproses (misalnya, "Jadwal yang dipilih tidak tersedia").

Secara keseluruhan, diagram ini secara efektif menggambarkan alur pemesanan layanan yang dinamis, termasuk validasi data real-time, penanganan kondisi sukses dan gagal, serta integrasi dengan sistem notifikasi eksternal untuk meningkatkan pengalaman pengguna.

3.6.14 Sequence Diagram Melihat Status Pesanan

Sequence Diagram ini menggambarkan alur interaksi antara Pelanggan, sistem web (*Frontend*), Server *Backend*, dan Database saat pelanggan ingin memeriksa status pesanan layanan yang telah mereka buat. Proses ini dirancang untuk memberikan kemudahan dan transparansi kepada pelanggan terkait progres pengerjaan layanan.



Gambar 3. 19 Sequence Diagram Melihat Status Pesanan

Berikut adalah urutan interaksi berdasarkan diagram:

- Asumsi Awal:** Proses ini dimulai dengan asumsi bahwa Pelanggan telah berhasil melakukan *login* ke dalam sistem dan saat ini berada di halaman utama atau dashboard. Keberhasilan *login* memastikan bahwa sistem mengenali identitas pelanggan.
- Inisiasi Permintaan oleh Pelanggan:** Pelanggan memilih menu atau opsi "Lihat Status Pesanan" yang tersedia pada Antarmuka Web (*Frontend*). Tindakan ini memicu permintaan untuk menampilkan data.
- Permintaan Data ke Server:** *Frontend* menerima perintah dari pelanggan dan meneruskannya dengan membuat sebuah permintaan (API request) ke

Server *Backend*. Permintaan ini secara implisit berisi informasi identitas pelanggan (misalnya, dari sesi *login* atau token) sehingga server tahu data milik siapa yang harus diambil.

4. **Pengambilan Data dari *Database*:** Server *Backend* menerima permintaan tersebut dan melanjutkan dengan melakukan *query* ke *Database*. *Query* ini bertujuan untuk mencari dan mengambil semua data pesanan yang terkait dengan akun pelanggan yang sedang aktif.
5. **Aliran Data Kembali:** *Database* memproses *query*, mengumpulkan data yang relevan (seperti nomor pesanan, jenis layanan, tanggal, dan status penggerjaan), lalu mengirimkan kumpulan data tersebut kembali ke Server *Backend*.
6. **Penyajian Data ke Pengguna:** Server *Backend* meneruskan data yang diterimanya ke *Frontend*. *Frontend* kemudian bertanggung jawab untuk menyajikan data tersebut dalam format yang mudah dibaca, seperti tabel atau daftar, sehingga Pelanggan dapat dengan mudah melihat status dari setiap pesanannya.

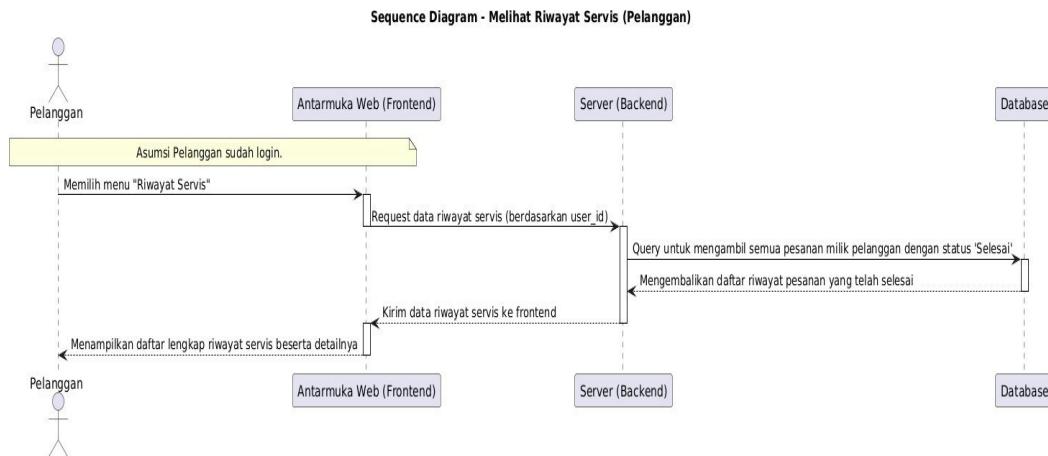
Secara ringkas, diagram ini menggambarkan sebuah alur "baca data" (*read operation*) yang efisien, di mana permintaan pengguna diterjemahkan menjadi *query* ke *database*, dan hasilnya ditampilkan kembali kepada pengguna tanpa mengubah data apa pun.

3.6.15 Sequence Diagram Melihat Riwayat Servis oleh Pelanggan

Sequence Diagram ini memodelkan alur kerja ketika seorang pelanggan mengakses riwayat dari semua layanan servis yang pernah dipesannya dan telah selesai dikerjakan. Tujuan utama dari fitur ini adalah untuk memberikan transparansi dan kemudahan akses bagi pelanggan terhadap data historis mereka. Dengan adanya fitur ini, pelanggan dapat dengan mudah melacak kembali pekerjaan apa saja yang pernah dilakukan pada unit barang miliknya tanpa harus menghubungi pihak bengkel, sehingga meningkatkan kepuasan dan kepercayaan pelanggan.

Aktor & Komponen Terlibat:

1. Aktor: Pelanggan
2. Komponen: Antarmuka Web (*Frontend*), Server (*Backend*), Database



Gambar 3. 20 Sequence Diagram Melihat Riwayat Servis

Berikut adalah urutan interaksi berdasarkan diagram:

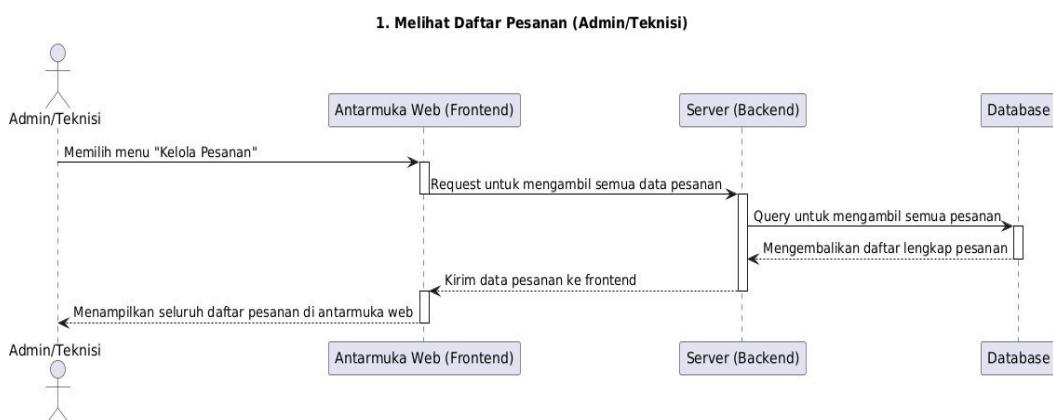
1. **Asumsi Awal:** Proses dimulai dengan asumsi bahwa Pelanggan telah berhasil melakukan login ke dalam sistem dan berada di halaman dashboard.
2. **Inisiasi oleh Pelanggan:** Pelanggan memilih menu atau opsi "Riwayat Servis" yang tersedia pada Antarmuka Web (*Frontend*) untuk melihat histori pesanannya.
3. **Permintaan Data:** *Frontend* menerima aksi tersebut dan mengirimkan sebuah permintaan (*API request*) ke Server *Backend*. Permintaan ini secara otomatis menyertakan identitas pelanggan yang didapat dari sesi login yang sedang aktif.
4. **Query Spesifik ke Database:** Server *Backend* menerima permintaan dan membuat *query* spesifik untuk *Database*. *Query* ini dirancang untuk mengambil data pesanan yang memenuhi dua kriteria utama:
 - a) Milik pelanggan yang sedang login.
 - b) Memiliki status "Selesai".
5. **Pengembalian Data dari Database:** *Database* memproses *query*, mengumpulkan semua catatan pesanan yang cocok, dan mengembalikan kumpulan data historis tersebut ke Server *Backend*.

6. Penyajian Data ke Pelanggan: Server meneruskan data yang diterimanya ke *Frontend*. *Frontend* kemudian bertanggung jawab untuk menyajikan informasi tersebut dalam format yang mudah dibaca (seperti daftar atau tabel), yang menampilkan detail seperti tanggal layanan, jenis servis, dan catatan teknisi untuk setiap riwayat pekerjaan.

Secara esensial, diagram ini menggambarkan alur "baca data" yang disaring secara spesifik untuk menyajikan informasi historis yang relevan bagi pelanggan, yang mendukung transparansi layanan dan memudahkan pelanggan dalam mengelola aset mereka.

3.6.16 Sequence Diagram Mengelola Pesanan oleh Admin/Teknisi

Sequence Diagram ini menjelaskan alur interaksi ketika Admin/Teknisi mengelola pesanan yang masuk melalui sistem. Proses ini meliputi *login*, menampilkan daftar pesanan, hingga memperbarui status pesanan sesuai progres pengerjaan. Karena alurnya memiliki dua bagian utama melihat daftar pesanan dan memperbarui satu pesanan disini akan membuatkan dua diagram terpisah agar alurnya lebih jelas dan mudah dibaca.

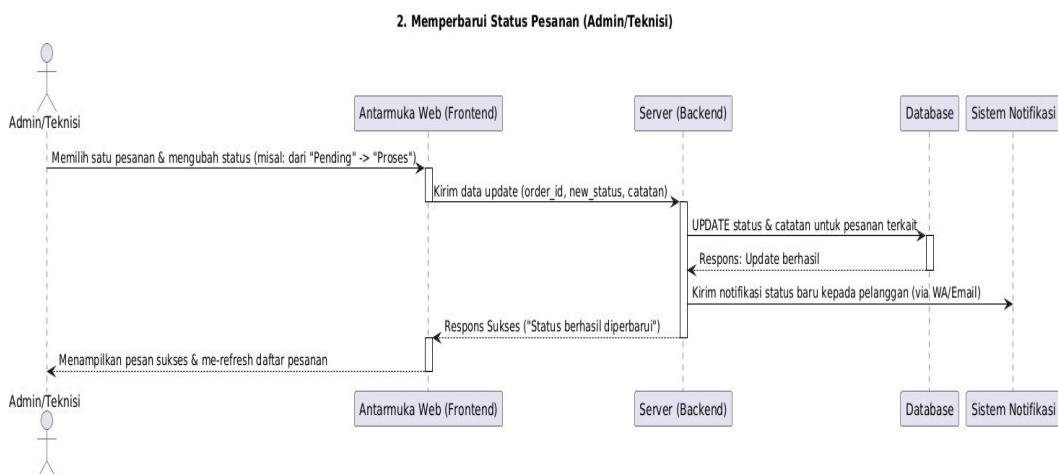


Gambar 3. 21 Sequence Diagram Melihat Daftar Pesanan

Diagram ini memvisualisasikan langkah awal dalam proses pengelolaan, yaitu ketika Admin/Teknisi mengakses sistem untuk melihat semua pesanan yang masuk. Ini adalah operasi "baca data" (read operation) yang penting untuk memberikan gambaran umum sebelum tindakan lebih lanjut.

Berikut adalah urutan interaksi berdasarkan diagram:

1. **Inisiasi oleh Admin/Teknisi:** Proses dimulai setelah Admin/Teknisi berhasil *login*. Mereka memilih menu "Kelola Pesanan" pada Antarmuka Web (*Frontend*) untuk memulai.
2. **Permintaan Daftar Pesanan:** *Frontend* menerima aksi tersebut dan mengirimkan sebuah permintaan (API request) ke Server (*Backend*) untuk mengambil data semua pesanan.
3. **Pengambilan Data dari Database:** Server memproses permintaan dengan mengirimkan *query* ke *Database*. *Query* ini bertujuan untuk mengambil seluruh data dari tabel pesanan, termasuk detail seperti nama pelanggan, tanggal, dan status saat ini.
4. **Pengiriman Data ke Frontend:** *Database* mengembalikan kumpulan data pesanan yang lengkap ke Server. Server kemudian meneruskan data ini ke *Frontend*.
5. **Penyajian Data:** *Frontend* menerima data dan menyajikannya dalam format yang terstruktur, seperti tabel atau daftar, sehingga Admin/Teknisi dapat dengan mudah melihat semua pesanan yang perlu dikelola.



Gambar 3. 22 Sequence Diagram Memperbarui Status Pesanan

Diagram ini fokus pada alur kerja spesifik yang terjadi setelah Admin/Teknisi memilih satu pesanan dari daftar untuk diubah statusnya. Ini adalah operasi "tulis data" (write operation) yang juga memicu notifikasi kepada pelanggan.

Berikut adalah urutan interaksi berdasarkan diagram:

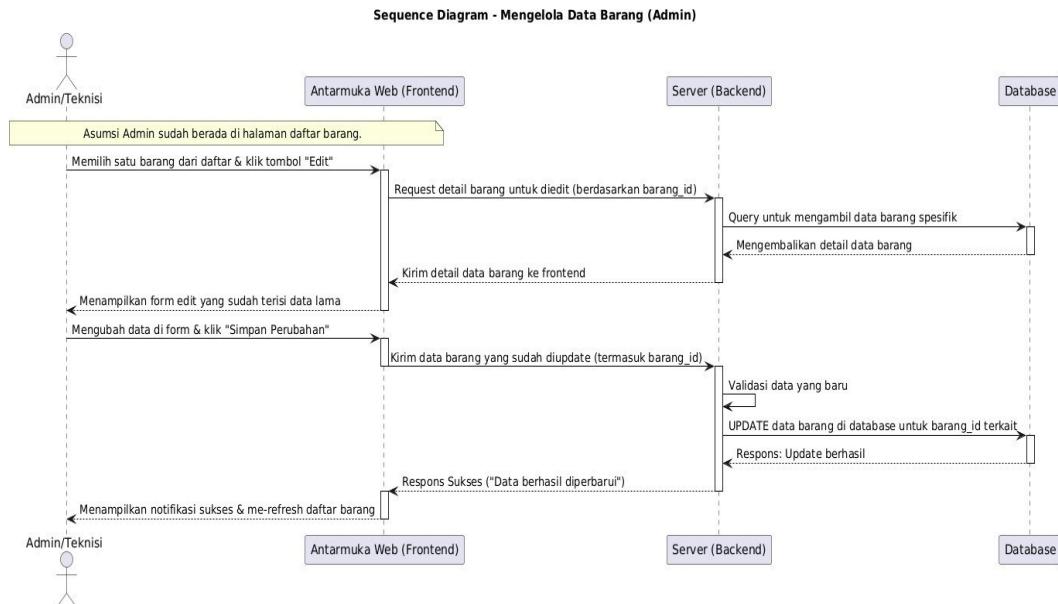
1. **Inisiasi Pembaruan:** Dari daftar yang ditampilkan, Admin/Teknisi memilih satu pesanan dan melakukan perubahan, misalnya mengubah status dari "Pending" menjadi "Proses" atau menambahkan catatan teknis pada *Frontend*.
2. **Pengiriman Data Perubahan:** *Frontend* mengirimkan permintaan pembaruan ke Server (*Backend*). Permintaan ini berisi data spesifik, yaitu ID pesanan yang akan diubah serta data barunya (status dan/atau catatan).
3. **Penyimpanan Perubahan ke Database:** Server memvalidasi permintaan dan mengirimkan perintah UPDATE ke *Database* untuk mengubah data pada catatan pesanan yang sesuai.
4. **Konfirmasi dan Notifikasi:** Setelah *Database* mengonfirmasi bahwa pembaruan berhasil, Server melakukan dua tugas penting:
 - a) **Mengirim Notifikasi:** Server memicu Sistem Notifikasi (misalnya, WhatsApp atau Email API) untuk secara otomatis mengirimkan informasi status terbaru kepada pelanggan yang bersangkutan.
 - b) **Menyiapkan Respons:** Server menyiapkan respons sukses untuk dikirim kembali ke *frontend*.
5. **Tampilan Hasil Akhir:** *Frontend* menerima respons sukses dari Server, lalu menampilkan pesan konfirmasi kepada Admin/Teknisi (contoh: "Status berhasil diperbarui") dan memperbarui tampilan daftar pesanan untuk merefleksikan perubahan yang baru saja dibuat.

3.6.17 Sequence Diagram Mengelola Data Barang oleh Admin/Teknisi

Sequence Diagram ini memodelkan alur kerja administratif yang dilakukan oleh Admin atau Teknisi untuk memperbarui atau mengoreksi data barang milik pelanggan yang sudah tersimpan di dalam sistem. Fungsi ini sangat krusial untuk menjaga akurasi dan integritas data aset, terutama untuk menangani skenario seperti kesalahan input awal oleh pelanggan atau adanya perubahan pada detail unit barang. Alur ini menggambarkan proses dua tahap: pengambilan data yang ada dan penyimpanan data yang telah diperbarui.

Aktor & Komponen Terlibat:

1. Aktor: Admin/Teknisi
2. Komponen: Antarmuka Web (*Frontend*), Server (*Backend*), Database



Gambar 3. 23 Sequence Diagram Mengelola Data Barang

Berikut adalah urutan interaksi berdasarkan diagram:

1. **Inisiasi Proses Edit:** Proses dimulai saat Admin/Teknisi, dari halaman daftar barang, memilih satu unit barang spesifik dan menekan tombol "Edit".
2. **Pengambilan Data Awal:** *Frontend* mengirimkan permintaan ke *Server Backend* untuk mengambil detail lengkap dari barang yang dipilih, berdasarkan ID unik barang tersebut.
3. **Query Data Spesifik:** Server meneruskan permintaan ini dengan melakukan *query SELECT* ke *Database* untuk mendapatkan semua atribut dari record barang yang dimaksud.
4. **Populasi Formulir:** *Database* mengembalikan data yang diminta ke Server, yang kemudian meneruskannya ke *Frontend*. *Frontend* menggunakan data ini untuk mengisi formulir edit, sehingga Admin dapat melihat informasi yang tersimpan saat ini.

5. **Aksi Pembaruan oleh Admin:** Admin/Teknisi mengubah data pada kolom yang diperlukan di dalam formulir, lalu menekan tombol "Simpan Perubahan".
6. **Pengiriman Data yang Diperbarui:** *Frontend* mengirimkan kembali seluruh data dari formulir (termasuk data yang telah diubah) ke **Server Backend**, beserta ID unik barang sebagai penanda record mana yang harus diperbarui.
7. **Eksekusi UPDATE di Database:** Server memvalidasi data yang masuk, lalu mengeksekusi perintah UPDATE ke *Database* untuk memperbarui record barang yang sesuai dengan data baru tersebut.
8. **Konfirmasi Akhir:** Setelah *Database* mengonfirmasi bahwa pembaruan berhasil, **Server** mengirimkan respons sukses ke *Frontend*, yang kemudian menampilkan notifikasi "Data berhasil diperbarui" kepada Admin dan me-refresh tampilan daftar barang.

Secara keseluruhan, diagram ini secara efektif mengilustrasikan alur kerja "edit" yang standar, yang menggabungkan operasi baca (*read*) untuk menampilkan data awal dan operasi tulis (*write*) untuk menyimpan perubahan, sehingga memastikan proses manajemen data berjalan secara aman dan akurat.

3.6.18 Sequence Diagram Pengelolaan Data Layanan (Admin)

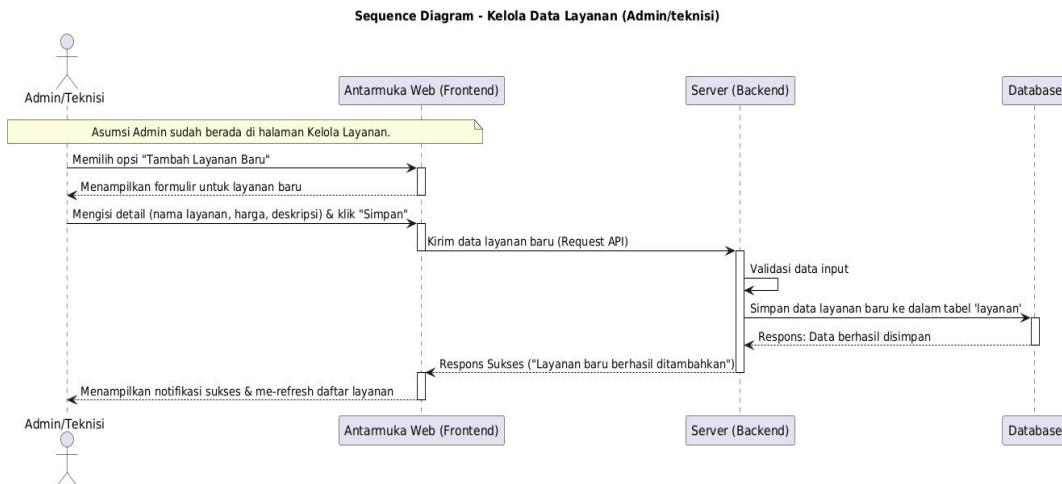
Sequence Diagram ini memodelkan alur kerja administratif saat Admin atau Teknisi melakukan pengelolaan data master layanan. Untuk menjaga kejelasan diagram, alur yang divisualisasikan secara spesifik adalah proses penambahan layanan baru.

Tujuan diagram ini adalah untuk menggambarkan secara rinci bagaimana interaksi antar komponen sistem mulai dari antarmuka pengguna (*Frontend*), server (*Backend*), hingga *Database* terjadi saat sebuah layanan baru diciptakan. Proses ini fundamental untuk memastikan sistem dapat beradaptasi dengan perubahan bisnis, seperti penambahan jenis servis baru yang ditawarkan oleh Bengkel.

Aktor & Komponen Terlibat:

- a) Aktor: Admin/Teknisi

b) Komponen: Antarmuka Web (*Frontend*), Server (*Backend*), *Database*



Gambar 3. 24 Sequence Diagram Mengelola Data Layanan

Berikut adalah urutan interaksi berdasarkan diagram:

1. **Inisiasi oleh Admin:** Proses diasumsikan dimulai pada halaman "Kelola Layanan". Admin/Teknisi memilih opsi untuk "Tambah Layanan Baru" pada Antarmuka Web (*Frontend*).
2. **Penyajian Formulir:** *Frontend* merespons dengan menampilkan sebuah formulir kosong yang perlu diisi untuk detail layanan baru.
3. **Input dan Pengiriman Data:** Admin mengisi semua detail yang diperlukan (seperti nama layanan, harga, dan deskripsi) lalu menekan tombol "Simpan". *Frontend* kemudian memaketkan data ini dan mengirimkannya ke Server *Backend* melalui sebuah panggilan API.
4. **Validasi di Sisi Server:** Server *Backend* menerima data, lalu melakukan proses validasi internal untuk memastikan data yang dikirim sesuai dengan format yang diharapkan (misalnya, harga harus berupa angka).
5. **Penyimpanan ke Database:** Setelah validasi berhasil, Server mengeksekusi perintah INSERT untuk menyimpan catatan layanan baru tersebut ke dalam tabel 'layanan' di *Database*.
6. **Konfirmasi dari Database:** *Database* memproses perintah dan mengirimkan respons kembali ke Server yang menandakan bahwa data telah berhasil disimpan.

7. **Penyampaian Respons Sukses:** Server meneruskan informasi keberhasilan ini ke *Frontend*, biasanya dalam bentuk respons API yang sukses beserta pesan konfirmasi.
8. **Tampilan Hasil Akhir:** *Frontend* menerima respons sukses, lalu menampilkan notifikasi visual kepada Admin (misalnya, "Layanan baru berhasil ditambahkan") dan memperbarui (me-refresh) daftar layanan yang ada di halaman.

Secara keseluruhan, diagram ini secara efektif menggambarkan alur 'create' dalam operasi CRUD (Create, Read, Update, Delete) pada manajemen data layanan, yang menunjukkan aliran data yang jelas dari input pengguna hingga penyimpanan permanen di database dan umpan balik konfirmasi.

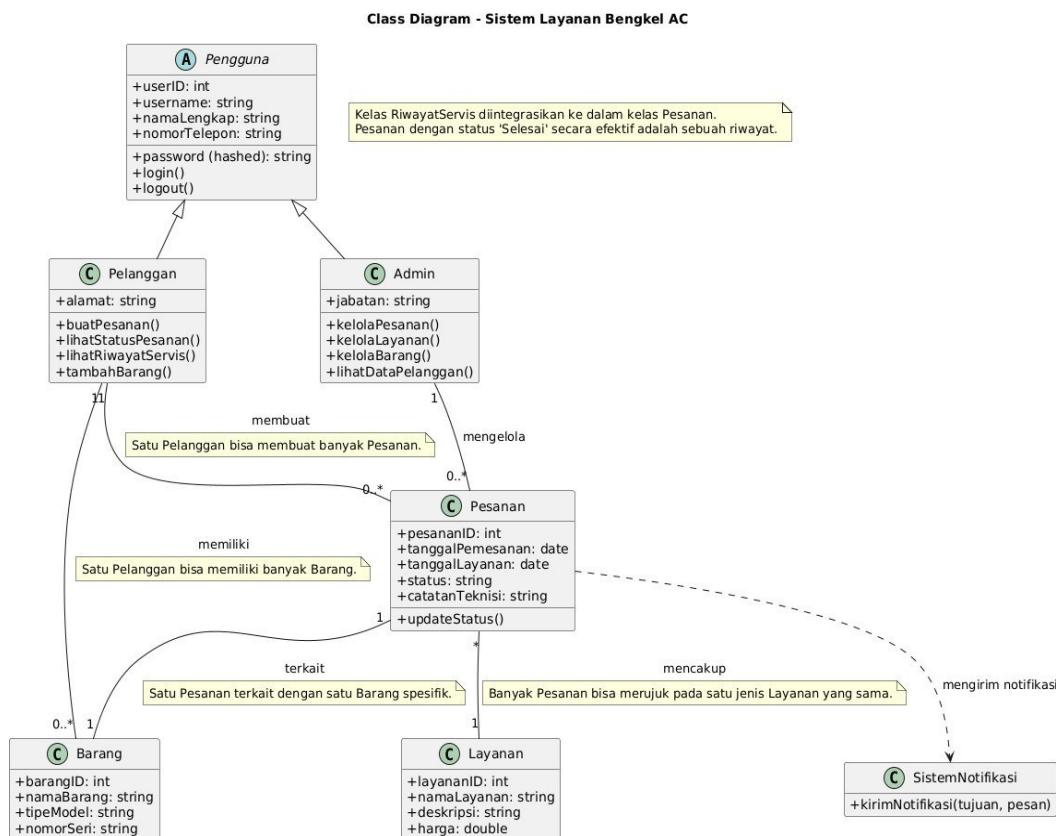
3.6.19 Class Diagram

Class Diagram ini berfungsi sebagai cetak biru (*blueprint*) arsitektur perangkat lunak untuk Sistem Informasi Layanan Bengkel AC. Diagram ini memetakan struktur statis sistem dengan mendefinisikan kelas-kelas utama, atribut (data) yang dimiliki setiap kelas, serta metode (fungsi) yang dapat dilakukannya. Lebih penting lagi, diagram ini memvisualisasikan hubungan logis dan struktural antar kelas, seperti pewarisan dan asosiasi, yang menjadi dasar fundamental bagi perancangan struktur database dan pengembangan kode program nantinya.

Penjelasan Komponen Kelas Utama:

1. **Pengguna:** Merupakan kelas abstrak (*abstract class*) yang menjadi dasar bagi semua jenis pengguna dalam sistem. Kelas ini menyimpan data umum seperti userID, username, password, dan namaLengkap.
2. **Pelanggan & Admin:** Adalah kelas turunan (*child class*) dari Pengguna yang memiliki peran dan fungsi spesifik. Pelanggan memiliki fungsi terkait pemesanan, sedangkan Admin fokus pada fungsi pengelolaan sistem secara keseluruhan.
3. **Pesanan:** Merupakan kelas sentral yang merepresentasikan inti dari transaksi layanan. Kelas ini berfungsi sebagai penghubung antara Pelanggan, Barang yang diservis, dan jenis Layanan yang diberikan.

4. **Barang:** Merepresentasikan aset atau unit elektronik milik pelanggan (misalnya: AC, kulkas) yang menjadi objek dari sebuah layanan.
5. **Layanan:** Merupakan data master yang berisi daftar jenis-jenis servis yang ditawarkan oleh bengkel beserta informasi harga dan deskripsinya.
6. **SistemNotifikasi:** Merepresentasikan komponen atau layanan eksternal yang digunakan oleh sistem untuk mengirim pemberitahuan kepada pengguna.



Gambar 3. 25 Class Diagram Sistem

Penjelasan Hubungan Antar Kelas:

Hubungan antar kelas dalam diagram ini mendefinisikan bagaimana objek-objek saling berinteraksi dan terhubung di dalam sistem.

1. **Pewarisan (Inheritance)** Struktur sistem menggunakan pola pewarisan di mana Pelanggan dan Admin merupakan spesialisasi dari kelas Pengguna. Pendekatan ini sangat efisien karena data dan fungsi umum (seperti login())

dan logout()) cukup didefinisikan sekali di kelas Pengguna dan akan diwariskan ke kelas turunannya.

2. **Asosiasi dan Multiplicitas** Hubungan inti dalam sistem ini berpusat pada kelas Pesanan, yang memiliki asosiasi sebagai berikut:
 - a) **Pelanggan dan Pesanan:** Satu Pelanggan dapat memiliki banyak (0..*) Pesanan, namun setiap Pesanan hanya dimiliki oleh satu (1) Pelanggan.
 - b) **Pesanan dan Barang:** Setiap Pesanan terkait dengan satu (1) Barang spesifik yang sedang diservis.
 - c) **Pesanan dan Layanan:** Banyak (*) Pesanan dapat merujuk pada satu (1) jenis Layanan yang sama dari daftar master.
3. **Keputusan Desain:** Integrasi Riwayat Servis Dalam perancangan ini, diputuskan untuk tidak membuat kelas RiwayatServis secara terpisah. Sebaliknya, fungsionalitas riwayat diintegrasikan langsung ke dalam kelas Pesanan. Sebuah objek Pesanan dengan atribut status bernilai 'Selesai' secara efektif telah merepresentasikan sebuah riwayat servis. Pendekatan ini dipilih untuk menyederhanakan struktur database dan menghindari redundansi data, karena semua informasi yang dibutuhkan untuk riwayat (seperti tanggal, layanan, dan catatan teknisi) sudah tersedia di dalam kelas Pesanan.

3.6.20 Perancangan Tabel Database

Berikut adalah rincian struktur tabel yang akan digunakan untuk mengimplementasikan database Sistem Informasi Layanan Bengkel. Desain ini didasarkan pada Entity-Relationship Diagram (ERD) dan mencakup definisi setiap kolom, tipe data, serta kunci (Primary Key dan Foreign Key) untuk memastikan integritas dan relasi data yang kuat.

1. Tabel Pengguna

Tabel 3. 1 Database Pengguna

Nama Kolom	Tipe Data	Kunci	Keterangan
userID	INT	PK	ID unik untuk setiap pengguna (Primary Key).
username	VARCHAR(50)	-	Nama pengguna untuk login, harus unik.
password_hash	VARCHAR(255)	-	Kata sandi yang sudah dienkripsi.
nama_lengkap	VARCHAR(100)	-	Nama lengkap dari pengguna.
nomor_telepon	VARCHAR(20)	-	Nomor telepon aktif pengguna.
alamat	TEXT	-	Alamat lengkap pengguna.
peran	ENUM('pelanggan','admin')	-	Peran pengguna dalam sistem.

Tabel ini berfungsi untuk menyimpan data semua pengguna yang dapat mengakses sistem, baik itu Pelanggan maupun Admin/Teknisi.

2. Tabel Barang

Tabel 3. 2 Database Barang

NamaKolom	Tipe Data	Kunci	Keterangan
barangID	INT	PK	ID unik untuk setiap barang (Primary Key).
userID	INT	FK	Terhubung ke userID di Tabel Users (Foreign Key).
nama_barang	VARCHAR(100)	-	Nama atau jenis barang (misal: AC, Kulkas).
tipe_model	VARCHAR(100)	-	Tipe atau model spesifik dari barang.
nomor_seri	VARCHAR(100)	-	Nomor seri unik dari unit barang.

Tabel ini digunakan untuk menyimpan informasi mengenai aset atau unit elektronik milik pelanggan.

3. Tabel Layanan

Tabel 3. 3 Database Layanan

Nama Kolom	Tipe Data	Kunci	Keterangan
layananID	INT	PK	ID unik untuk setiap jenis layanan (Primary Key).
nama_layanan	VARCHAR(100)	-	Nama dari layanan (misal: Servis Rutin AC).
deskripsi	TEXT	-	Penjelasan singkat mengenai cakupan layanan.
harga	DECIMAL(10, 2)	-	Harga standar untuk layanan tersebut.

Tabel ini merupakan tabel master yang berisi daftar semua jenis layanan yang ditawarkan oleh bengkel.

4. Tabel Pesanan

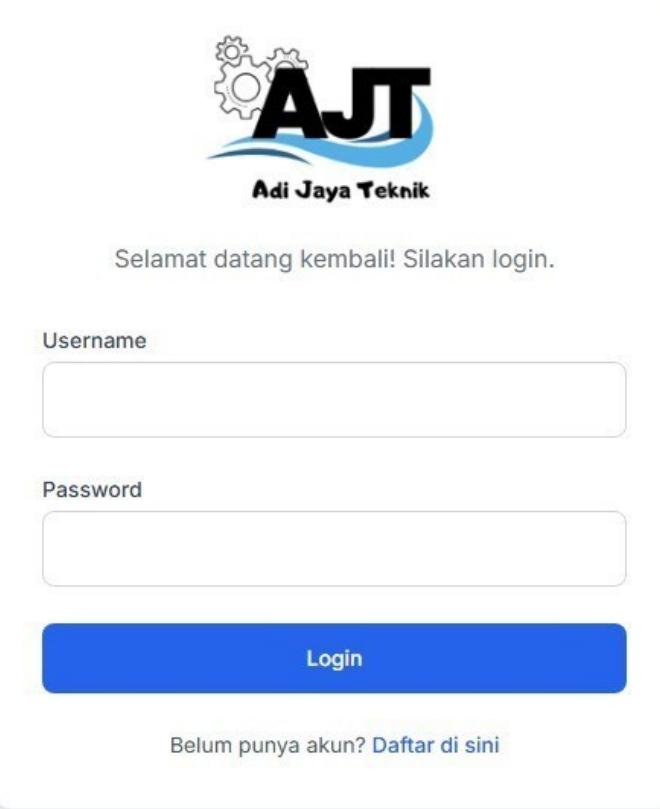
Tabel 3. 4 Database Pesanan

Nama Kolom	Tipe Data	Kunci	Keterangan
pesananiD	INT	PK	ID unik untuk setiap pesanan (Primary Key).
pelangganID	INT	FK	Terhubung ke userID di Tabel Users (Foreign Key).
barangID	INT	FK	Terhubung ke barangID di Tabel Barang (Foreign Key).
layananID	INT	FK	Terhubung ke layananID di Tabel Layanan (Foreign Key).
tanggal_pemesanan	DATETIME	-	Waktu saat pesanan dibuat.
tanggal_layanan	DATETIME	-	Jadwal waktu penggerjaan layanan.
status	ENUM('Pending', 'Proses', 'Selesai')	-	Status progres penggerjaan pesanan.
catatan_teknisi	TEXT	-	Catatan dari teknisi terkait penggerjaan.

Ini adalah tabel transaksi utama yang mencatat setiap pesanan layanan yang dibuat oleh pelanggan.

3.6.21 Tampilan Pengguna (User Interface)

1. Halaman Login



Gambar 3. 26 Perancangan Antarmuka Halaman *Login*

Gambar 3.25 merupakan rancangan antarmuka untuk halaman login. Halaman ini dirancang dengan desain yang bersih dan minimalis untuk memudahkan pengguna masuk ke dalam sistem. Terdapat kolom input untuk username dan password, tombol login sebagai aksi utama, serta tautan untuk menuju halaman registrasi bagi pengguna baru.

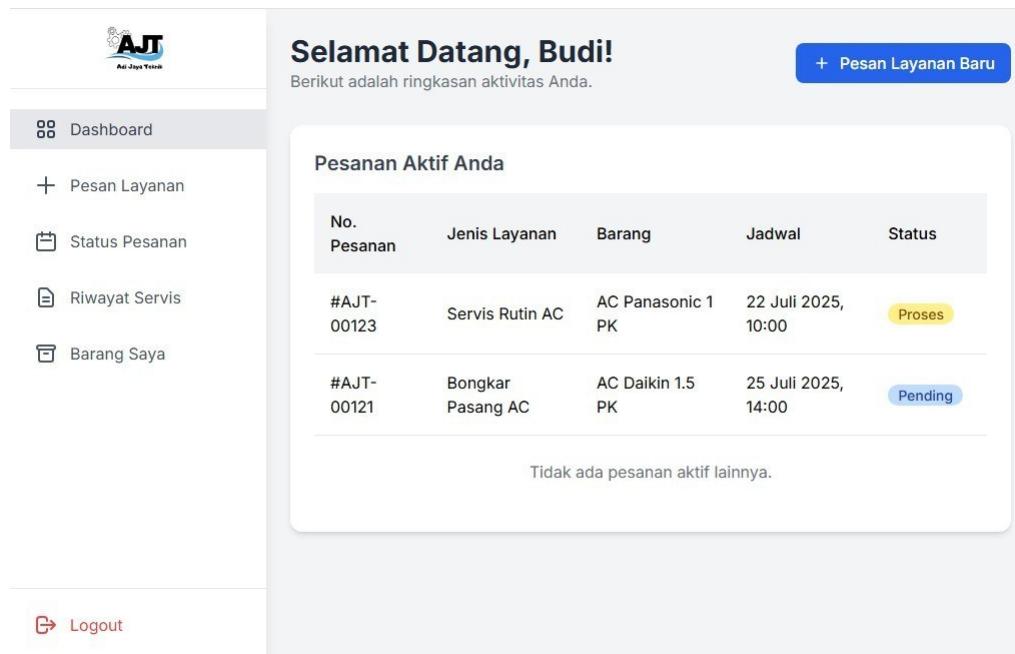
2. Halaman Registrasi Pelanggan

The screenshot shows a registration form titled "Buat Akun Baru" (Create New Account). At the top is the AJT logo with the text "Adi Jaya Teknik". Below the title is a placeholder text: "Silakan isi data diri Anda di bawah ini." The form consists of five input fields: "Nama Lengkap" (Full Name), "Username", "Nomor Telepon" (Phone Number), "Alamat" (Address), and "Password". Each field has a corresponding text input box. A large blue button at the bottom is labeled "Daftar" (Register). Below the button, a link says "Sudah punya akun? [Login di sini](#)".

Gambar 3. 27 Perancangan Antarmuka Halaman Registrasi

Gambar 3.26 menampilkan rancangan antarmuka untuk Halaman Registrasi, yang dirancang untuk memberikan pengalaman pendaftaran akun yang mudah dan *intuitif* bagi pengguna baru. Desain halaman ini menjaga konsistensi visual dengan halaman *login*, menggunakan palet warna dan tata letak yang serupa untuk menciptakan alur yang mulus.

3. Halaman Dashboard Pelanggan



Gambar 3. 28 Perancangan Antarmuka Dashboard Pelanggan

Gambar 3.27 menampilkan rancangan antarmuka untuk halaman Dashboard Pelanggan, yang merupakan halaman utama yang dilihat pengguna setelah berhasil login. Halaman ini dirancang sebagai pusat kendali (*control center*) bagi pelanggan untuk mengakses semua fitur utama sistem secara cepat dan efisien. Desainnya menggunakan tata letak dua kolom dengan *sidebar* navigasi di sisi kiri dan area konten utama di kanan, sebuah pola yang umum digunakan untuk aplikasi web modern karena kemudahan navigasinya.

Komponen Utama Halaman:

1. **Sidebar Navigasi (Sisi Kiri):**

- Logo Perusahaan:** Terletak di bagian atas sidebar untuk memperkuat identitas merek.
- Menu Navigasi:** Berisi daftar tautan vertikal dengan ikon yang jelas untuk setiap fungsi utama, seperti:
 - Dashboard
 - Pesan Layanan
 - Status Pesanan
 - Riwayat Servis

5) Barang Saya

- c) Menu yang sedang aktif diberi latar belakang yang berbeda untuk memberikan orientasi visual kepada pengguna.
- d) **Tombol Logout:** Terletak di bagian bawah sidebar, memungkinkan pengguna untuk keluar dari sistem dengan aman.

2. Area Konten Utama (Sisi Kanan):

- a) **Header Konten:** Menyambut pengguna dengan nama mereka (misalnya, "Selamat Datang, Budi!") untuk memberikan sentuhan personal. Terdapat juga tombol *call-to-action* utama, yaitu "Pesan Layanan Baru", untuk mempermudah akses ke fungsi yang paling sering digunakan.
- b) **Ringkasan Pesanan Aktif:** Bagian inti dari dashboard ini adalah sebuah tabel yang menampilkan ringkasan pesanan yang sedang dalam status "Pending" atau "Proses". Informasi yang disajikan mencakup nomor pesanan, jenis layanan, barang, jadwal, dan status terkini yang ditandai dengan label berwarna untuk identifikasi cepat.

Secara keseluruhan, halaman Dashboard Pelanggan ini dirancang untuk memberikan pengalaman pengguna yang informatif dan terorganisir, di mana pelanggan dapat dengan cepat melihat ringkasan aktivitas mereka dan mengakses semua fitur penting sistem dari satu lokasi terpusat.

4. Tampilan Halaman Pemesanan Layanan

Gambar 3. 29 Perancangan Antarmuka Formulir Pemesanan Layanan

Gambar 3.28 menampilkan rancangan antarmuka untuk halaman Formulir Pemesanan Layanan. Halaman ini merupakan inti dari fitur pemesanan, di mana pelanggan dapat secara rinci menentukan kebutuhan servis mereka. Desain formulir ini ditempatkan di dalam area konten utama dari tata letak dashboard, sehingga pelanggan tetap dapat mengakses menu navigasi utama di sidebar saat melakukan pemesanan. Pendekatan ini memastikan pengalaman pengguna yang konsisten dan terintegrasi.

Komponen Utama Halaman:

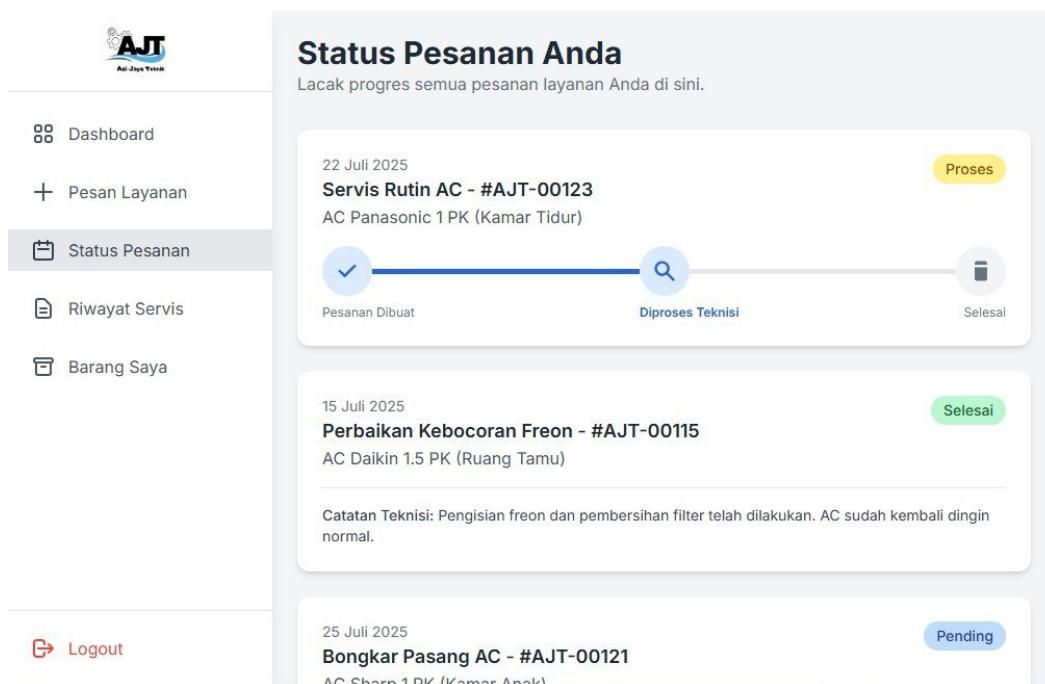
1. **Header Halaman:** Terdiri dari judul yang jelas, "Pesan Layanan Servis", dan sub-judul yang memberikan instruksi singkat, bertujuan untuk mengarahkan pengguna dalam mengisi formulir.
2. **Formulir Interaktif:** Formulir ini dirancang untuk memandu pelanggan langkah demi langkah dalam proses pemesanan:
 - a) **Pilih Barang/Unit:** Sebuah menu *dropdown* yang memungkinkan pelanggan untuk memilih unit (misalnya, AC) yang sudah pernah mereka daftarkan sebelumnya. Terdapat juga tautan praktis

"Tambah barang baru" bagi pelanggan yang ingin mendaftarkan unit baru secara langsung dari halaman ini.

- b) **Jenis Layanan:** Menu *dropdown* kedua yang berisi daftar semua layanan yang ditawarkan oleh bengkel, seperti "Servis Rutin", "Bongkar Pasang", atau "Perbaikan Kebocoran".
 - c) **Tanggal & Waktu:** Sebuah kolom input khusus (*datetime-local*) yang memudahkan pelanggan untuk memilih jadwal penggerjaan yang diinginkan.
 - d) **Catatan Tambahan:** Area teks (*textarea*) yang bersifat opsional, dimana pelanggan dapat memberikan informasi tambahan atau deskripsi keluhan yang lebih spesifik mengenai masalah pada unitnya.
3. **Tombol Aksi:** Sebuah tombol "Kirim Pesanan" yang diletakkan secara strategis di bagian bawah formulir. Tombol ini dirancang menonjol untuk menandakan aksi final dalam proses pemesanan.

Secara keseluruhan, formulir ini dirancang dengan fokus pada kemudahan penggunaan (*usability*), memastikan semua informasi yang dibutuhkan untuk sebuah pesanan layanan dapat dikumpulkan secara lengkap dan terstruktur dalam satu alur yang sederhana.

5. Tampilan Halaman Status Pesanan



Gambar 3. 30 Perancangan Antarmuka Halaman Status Pesanan

Gambar 3.29 menampilkan rancangan antarmuka untuk Halaman Status Pesanan. Halaman ini dirancang untuk memberikan transparansi penuh kepada pelanggan mengenai progres dari setiap layanan yang mereka pesan. Berbeda dengan ringkasan di dashboard, halaman ini menyajikan informasi yang lebih detail dan terperinci untuk setiap pesanan. Desainnya menggunakan format kartu (*card-based layout*), di mana setiap kartu merepresentasikan satu pesanan, sehingga informasi menjadi lebih terorganisir dan mudah dibaca.

Komponen Utama Halaman:

1. **Header Halaman:** Terletak di bagian atas area konten, header ini berisi judul "Status Pesanan Anda" dan sub-judul yang menjelaskan fungsi halaman, memberikan orientasi yang jelas bagi pengguna.
2. **Daftar Kartu Pesanan:** Setiap pesanan ditampilkan dalam kartu individual yang terpisah. Struktur ini memungkinkan penyajian informasi yang kaya tanpa terlihat berantakan. Setiap kartu berisi:

- a) **Informasi Dasar:** Nomor pesanan, jenis layanan, nama barang, dan tanggal pemesanan.
- b) **Label Status:** Sebuah label berwarna yang menonjol di sudut kanan atas kartu untuk identifikasi status secara cepat (misalnya, kuning untuk "Proses", hijau untuk "Selesai", dan biru untuk "Pending").
3. **Visualisasi Progres (untuk Pesanan Aktif):** Untuk pesanan yang sedang dalam status "Proses", ditambahkan sebuah *timeline* atau bilah progres visual. Elemen ini secara intuitif menunjukkan tahapan mana yang sudah selesai dan tahapan mana yang sedang berjalan, memberikan gambaran yang lebih dinamis kepada pelanggan.
4. **Catatan Teknisi (untuk Pesanan Selesai):** Untuk pesanan yang sudah selesai, area khusus di dalam kartu digunakan untuk menampilkan catatan akhir dari teknisi. Ini memberikan nilai tambah berupa informasi detail mengenai hasil pekerjaan atau rekomendasi perawatan.

Secara keseluruhan, halaman ini dirancang dengan fokus pada transparansi dan pengalaman pengguna yang informatif. Dengan memvisualisasikan progres dan menyajikan detail secara terstruktur, pelanggan dapat merasa lebih terhubung dan percaya dengan proses layanan yang diberikan.

6. Tampilan Dashboard Admin

The dashboard features a sidebar with navigation links: Dashboard, Kelola Pesanan, Data Pelanggan, Data Barang, Data Layanan, and Logout. The main area displays a welcome message and three summary cards: Pesanan Masuk Hari Ini (8), Total Pesanan Pending (12), and Pesanan Selesai Bulan Ini (45). Below this is a table titled 'Pesanan Terbaru' showing the latest five orders with columns for No. Pesanan, Pelanggan, Layanan, Jadwal, Status, and Aksi.

No. Pesanan	Pelanggan	Layanan	Jadwal	Status	Aksi
#AJT-00123	Budi Santoso	Servis Rutin AC	22 Juli 2025	Proses	Detail
#AJT-00122	Citra Lestari	Instalasi Listrik	24 Juli 2025	Pending	Detail
#AJT-00121	Agus Wiwaha	Bongkar Pasang AC	25 Juli 2025	Pending	Detail

Gambar 3. 31 Perancangan Antarmuka Dashboard Admin

Gambar 3.30 menampilkan rancangan antarmuka untuk halaman Dashboard Admin, yang merupakan pusat kendali utama bagi Admin atau Teknisi untuk mengelola seluruh operasional sistem. Desain halaman ini lebih padat informasi dibandingkan dashboard pelanggan, dengan fokus pada penyajian data agregat dan akses cepat ke fungsi-fungsi manajemen. Tata letaknya konsisten menggunakan sidebar navigasi di kiri dan area konten utama di kanan.

Komponen Utama Halaman:

1. **Sidebar Navigasi (Sisi Kiri):** Sidebar ini berisi menu-menu yang spesifik untuk peran administrator, yang mencakup semua fungsi pengelolaan data master dan transaksi. Menu yang tersedia antara lain:
 - a) Dashboard
 - b) Kelola Pesanan
 - c) Data Pelanggan
 - d) Data Barang
 - e) Data Layanan
2. **Area Konten Utama (Sisi Kanan):**
 - a) **Header Konten:** Menyambut admin dan memberikan ringkasan singkat tentang tujuan halaman.
 - b) **Kartu Statistik:** Di bagian atas, terdapat serangkaian kartu statistik yang menyajikan data kunci secara visual. Kartu-kartu ini menampilkan metrik penting seperti "Pesanan Masuk Hari Ini", "Total Pesanan Pending", dan "Pesanan Selesai Bulan Ini". Tujuannya adalah untuk memberikan gambaran cepat mengenai kondisi operasional saat ini.
 - c) **Tabel Pesanan Terbaru:** Bagian inti dari dashboard ini adalah sebuah tabel yang menampilkan daftar pesanan terbaru yang memerlukan perhatian atau tindakan segera. Tabel ini menyajikan informasi penting seperti nomor pesanan, nama pelanggan, jenis layanan, jadwal, dan status. Terdapat juga tombol "Detail" pada setiap baris untuk memungkinkan admin melihat informasi lebih lanjut atau mengelola pesanan tersebut.

Secara keseluruhan, Dashboard Admin dirancang untuk menjadi pusat informasi yang efisien, memungkinkan admin untuk memantau aktivitas sistem, mengidentifikasi prioritas pekerjaan, dan mengakses semua fungsi pengelolaan dengan mudah dari satu halaman terpusat.

BAB IV

KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang ditarik dari keseluruhan proses perancangan sistem informasi layanan untuk Bengkel AC Adi Jaya Teknik, serta saran-saran yang dapat menjadi acuan untuk pengembangan sistem di masa mendatang agar lebih komprehensif dan berdaya saing.

4.1 Kesimpulan

Berdasarkan hasil perancangan dan analisis yang telah dilakukan, dapat disimpulkan bahwa sebuah sistem informasi layanan bengkel AC berbasis web telah berhasil dirancang untuk menjawab kebutuhan Bengkel AC Adi Jaya Teknik. Perancangan ini menghasilkan sebuah cetak biru (*blueprint*) sistem yang mampu mencatat permintaan servis secara terstruktur dan memfasilitasi pengelolaan informasi pelanggan serta teknisi secara efisien, yang secara langsung menggantikan proses manual yang berjalan saat ini. Dengan fitur-fitur seperti pemesanan online, pelacakan status, dan manajemen data terpusat, sistem yang dirancang ini secara efektif memenuhi tujuan untuk meningkatkan efisiensi operasional dan transparansi layanan.

4.2 Saran

Meskipun perancangan saat ini sudah mencakup fungsi-fungsi inti, terdapat beberapa peluang pengembangan lebih lanjut untuk menyempurnakan sistem di masa depan. Berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Pengembangan Aplikasi Mobile

Untuk meningkatkan aksesibilitas dan kenyamanan pelanggan, disarankan untuk mengembangkan versi aplikasi mobile (Android/iOS). Aplikasi mobile akan memudahkan pelanggan dalam melakukan pemesanan, melacak status, dan menerima notifikasi secara langsung di perangkat mereka.

2. Integrasi Gerbang Pembayaran (*Payment Gateway*)

Menambahkan fitur pembayaran online melalui integrasi dengan *payment gateway* akan menyempurnakan alur transaksi. Pelanggan dapat langsung membayar biaya layanan melalui aplikasi, yang akan mengotomatisasi proses penagihan dan pencatatan keuangan.

3. Fitur Khusus untuk Teknisi

Disarankan untuk membuat antarmuka khusus bagi teknisi di lapangan. Fitur ini dapat mencakup jadwal kerja harian, detail tugas, navigasi ke lokasi pelanggan, serta kemampuan untuk memperbarui status pekerjaan dan mengunggah foto bukti pengerjaan langsung dari lokasi.

4. Modul Manajemen Inventaris

Untuk mengelola suku cadang (spare parts) dengan lebih baik, dapat dikembangkan sebuah modul manajemen inventaris. Fitur ini akan membantu admin melacak stok komponen seperti freon, pipa, atau filter, dan secara otomatis mengurangi stok saat digunakan dalam sebuah layanan.

5. Dashboard Analitik dan Pelaporan

Menambahkan fitur dasbor analistik untuk admin dapat memberikan wawasan bisnis yang berharga. Admin dapat melihat laporan mengenai layanan yang paling diminati, pendapatan bulanan, hingga tingkat kepuasan pelanggan, yang dapat digunakan untuk pengambilan keputusan strategis.

6. Pengujian Usabilitas (*Usability Testing*)

Sebelum sistem diimplementasikan secara penuh, sangat disarankan untuk melakukan pengujian usabilitas dengan melibatkan calon pengguna akhir (pelanggan dan admin). Umpulan balik dari pengujian ini akan sangat berharga untuk menyempurnakan desain antarmuka dan pengalaman pengguna agar lebih optimal.

DAFTAR PUSTAKA

- [1] D. Puryati, S. Kuntadi, C. Author, S. Tinggi, and I. Ekonomi, “Implementasi Digitalisasi Manajemen Usaha Melalui Digital Marketing dan Pembukuan Digital pada UMKM Bengkel Vespa Kota Bandung,” *Jurnal Dharma Bhakti Ekuitas*, vol. 08, no. 02, 2024.
- [2] S. Sofiani, S. Kosasi, and S. Margaretha Kuwai, “PERANCANGAN MOBILE WEB BENGKEL MOBIL MIZU PONTIANAK,” 2024. [Online]. Available: <http://ejournal.stmik-time.ac.id>
- [3] Lilis Setyowati and Untung Lasiyono, “Optimalisasi Perencanaan dan Penjadwalan Produksi: Kunci Meningkatkan Efisiensi Operasional,” *Jurnal Inovasi Manajemen, Kewirausahaan, Bisnis dan Digital*, vol. 1, no. 4, pp. 50–56, Oct. 2024, doi: 10.61132/jimakebidi.v1i4.354.
- [4] D. Vernanda, Z. Zain Aprilia Putri, S. Budi, and P. Negeri Subang, “SISTEM INFORMASI BENGKEL ONLINE BERBASIS APLIKASI,” 2024.
- [5] R. Empu and C. Yuwono, “PERANCANGAN SISTEM INFORMASI INVENTARIS SEKOLAH BERBASIS WEB DENGAN METODE WATERFALL (STUDI KASUS: YAYASAN PENDIDIKAN PUTRA SATRIA),” 64 |*Jurnal Ilmu Komputer JIK*, vol. 02, 2023.
- [6] D. R. Kaparang *et al.*, “PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA SMK,” 2022.
- [7] Y. Wahyudin and D. N. Rahayu, “Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review,” *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 15, no. 3, pp. 26–40, Oct. 2020, doi: 10.35969/interkom.v15i3.74.
- [8] D. Anggraeni and W. Arafah, “Pengaruh Kualitas Digitalisasi Layanan terhadap Customer Satisfaction dan Customer Retention Intention pada Platform Digital Banking.” [Online]. Available: <http://Jiip.stkipyapisdompu.ac.id>
- [9] A. Fajar, P. Sundari, S. SEMARANG Semarang, and J. Tengah, “Seminar Nasional dan Call for Papers 2024 Sekolah Tinggi Ilmu Ekonomi semarang

Peran Digitalisasi Layanan Pelanggan dalam Memperkuat Loyalitas di Industri E-Commerce.”

- [10] T. Wahyuningjati, S. H. Iskandar, A. N. Sabrina, and M. Z. Saleh, “Transformasi Digital dalam Manajemen Jasa: Tantangan dan Peluang di Era Industri 4.0,” *Business and Investment Review*, vol. 2, no. 6, Dec. 2024, doi: 10.61292/birev.139.
- [11] Supriyadi, “INTEGRASI SISTEM INFORMASI MANAJEMEN SDM DALAM TRANSFORMASI DIGITAL: PENGARUH TERHADAP EFISIENSI OPERASIONAL (Studi Pada Industri Manufaktur di Cilegon),” 2024.
- [12] U. Bakti Tunas Husada, “OPTIMALISASI IMPLEMENTASI SISTEM INFORMASI MANAJEMEN BERBASIS CLOUD UNTUK MENINGKATAN EFISIENSI OPERASIONAL DI SEKTOR INDUSTRI : STUDI LITERATUR,” 2025.
- [13] Y. I. Susilawati¹, Y. Rumanto¹, U. Maslahah¹, R. Al-Qodiri¹, H. Tursulistyono, and Y. Achsan¹, “Pengaruh ERP Berbasis Cloud Terhadap Efisiensi Rantai Pasok,” 2024.
- [14] F. Z. Simbolon, F. Izhari, and Z. Sitorus, “Implementasi Sistem Arsip Elektronik Dalam Meningkatkan Efisiensi Operasional Di Smk Gelora Jaya Nusantara Medan,” *Jurnal Minfo Polgan*, vol. 13, no. 2, pp. 1582–1589, Oct. 2024, doi: 10.33395/jmp.v13i2.14157.
- [15] S. Onassis, T. Utama, and Sutarno, “Pengaruh Customer Relationship Management (CRM) Terhadap Keputusan Pembelian,” *Journal of Trends Economics and Accounting Research*, vol. 4, no. 3, pp. 647–653, Mar. 2024, doi: 10.47065/jtear.v4i3.1043.
- [16] D. Ayu Patmawati and A. Lestari Andjarwati, “Volume 11 Nomor 2 Halaman 323-336 Jurnal Ilmu Manajemen Peran pengalaman pelanggan dalam memediasi pengaruh kualitas layanan, citra perusahaan, dan harga terhadap loyalitas pelanggan PT. Pos Indonesia,” 2023.

- [17] Q. Imanda, D. Gusman, and E. Azriadi, “ACADEMIC SYSTEM MANAGEMENT BERBASIS WEBSITE DI SEKOLAH DASAR NEGERI 012 LANGGINI (PROGRAMMING),” 2022.
- [18] Dinda Khoirunisa, “PENGANTAR BASIS DATA LANJUTAN,” 2023.
- [19] I. Fahzirah, “PENGENALAN SISTEM DATABASE : KONSEP DASAR DAN MANFAATNYA DALAM PERUSAHAAN Muhammad Irwan Padli Nasution,” *Jurnal Ilmiah Nusantara (JINU)*, vol. 1, no. 4, 2024, doi: 10.61722/jinu.v1i4.1884.
- [20] N. E. Lim and M. Silalahi, “RANCANG BANGUN SISTEM E-ADMINISTRASI BERBASIS CODEIGNITER FRAMEWORK DI KP2A BATAM,” *JURNAL COMASIE*, vol. 08, no. 1, 2023.
- [21] Albert Yakobus Chandra and Putry Wahyu Setyaningsih, “Benchmarking Local Development Environments: Analyzing the Performance of XAMPP, MAMP, and Laragon,” 2025.
- [22] G. B. Dirgantara, F. Budiman, J. ’ Far, S. Mulana, and A. S. Adam, “Peran Penggunaan HTML dalam Meningkatkan Keterampilan Komunikasi Siswa SMA,” 2024.
- [23] A. Mardiansyah *et al.*, “Pengembangan Dasar HTML Dan CSS: Langkah Pertama Dalam Pengembangan Web,” 2024, [Online]. Available: <https://jurnalmahasiswa.com/index.php/appa>
- [24] Setiaji, Juan Fachriza, F. Akbar, and A. Ari, “IMPLEMENTASI MODEL UNIFIED MODELLING LANGUAGE(UML) PADA PERANCANGAN SISTEM INFORMASI DATA KEPENDUDUAKN,” *Jurnal Informatika Teknologi dan Sains*, 2024.
- [25] A. Oktarino, I. T. Amri, A. Afriansyah, A. Ferdian Shobur, B. Gultom, and R. Dauli, “Implementasi Diagram Unified Modeling Language (UML) Sistem Informasi Berbasis Web SD Negeri 042/XI Seberang Kota Sungai Penuh Implementation of UML (Unified Modeling Language) Diagram of Web-Based Information System of SD Negeri 042/XI Seberang Sungai Penuh CITY,” 2024. [Online]. Available: www.jurnal.unimed.ac.id

- [26] Rospricilia Tita Ayu, M. Nizar, and P. Ma'ady, "Pemodelan Integration Use Case (IUC): Perancangan Use Case Diagram (UML) untuk Sistem-sistem yang Terintegrasi," 2024.
- [27] A. Septiansyah, S. Hasanah, V. Nita Permatasari, and A. Yuliawati, "SISTEM INFORMASI OTOMATISASI PELAPORAN DATA PENJUALAN TOKO BUKU NAZWA YANG MASUK DAN YANG KELUAR," 2024, doi: 10.37817/ikraith-informatika.v8i1.
- [28] R. Elmansouri, S. Meghzili, and A. Chaoui, "UML 2.0 ACTIVITY DIAGRAMS/CSP INTEGRATED APPROACH FOR MODELING AND VERIFICATION OF SOFTWARE SYSTEMS," *Computer Science*, vol. 22, no. 2, pp. 209–235, 2021, doi: 10.7494/csci.2021.22.2.3478.
- [29] F. Siewe and G. M. Ngounou, "On the Execution and Runtime Verification of UML Activity Diagrams," *Software*, vol. 4, no. 1, p. 4, Feb. 2025, doi: 10.3390/software4010004.
- [30] S. Al-Fedaghi, "UML Sequence Diagram: An Alternative Model," 2021. [Online]. Available: www.thesai.org
- [31] S. W. Ramdany, S. Aulia Kaidar, B. Aguchino, C. Amelia, A. Putri, and R. Anggie, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," 2024.
- [32] J. Pengabdian *et al.*, "P E N G A M A T A N : Pelatihan Website Responsif UMKM dengan Flask dan Bootstrap 4.5 di Cimanuk," 2024. [Online]. Available: <https://ojs3.unpatti.ac.id/index.php/pengamatan>
- [33] M. Y. Putra, "Cara sitasi: Putra MY. 2020. Responsive Web Design Menggunakan Boostrap Dalam Merancang Layout Web," *Information System for Educators and Professionals*, vol. 5, no. 1, pp. 61–70, 2020.



Nama : Riski keliolan

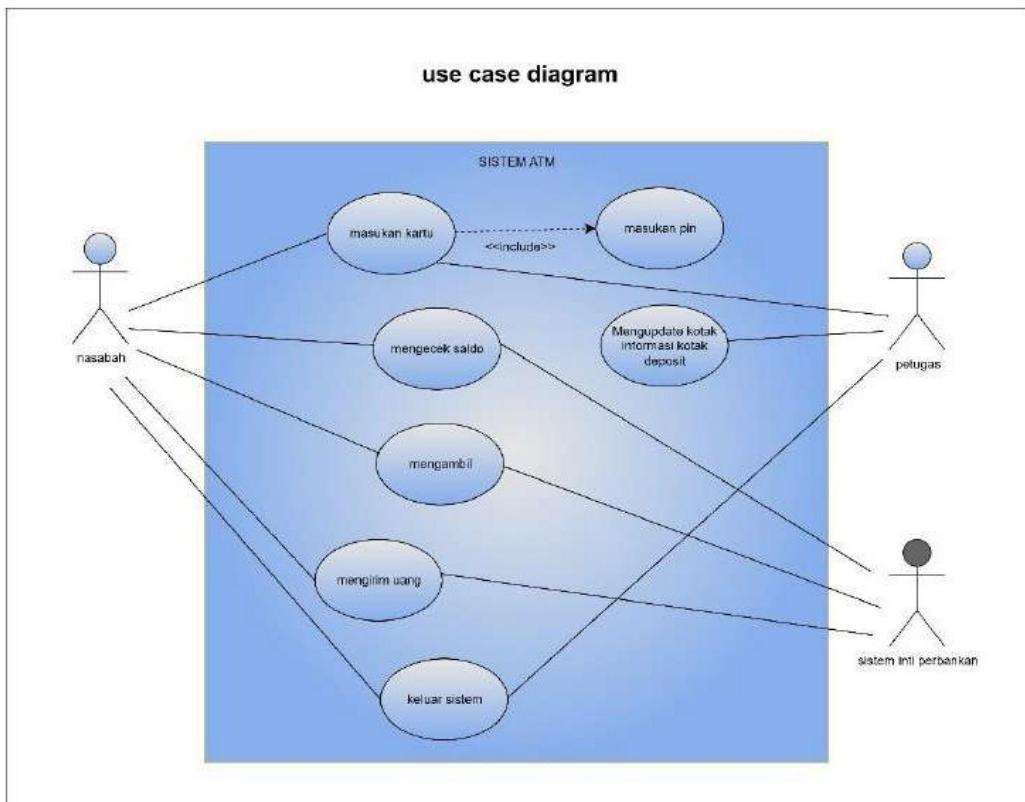
Nim 312310716

Kelas : TI .23.A 3

Penjelasan Diagram dengan bahasa Unified Modelling Language(UML) mengikuti materi SAD Project Sistem ATM

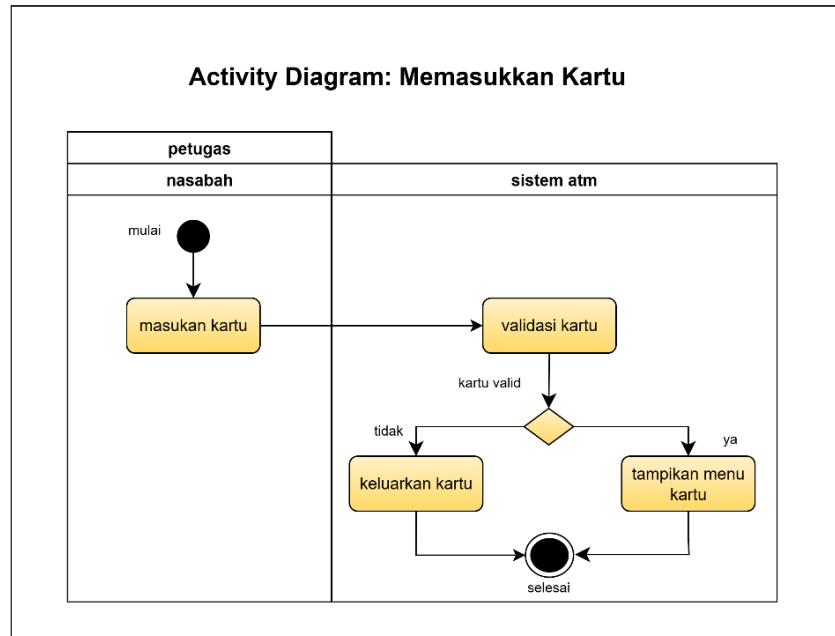
Bagian Analisis:

1. Usecase Diagram

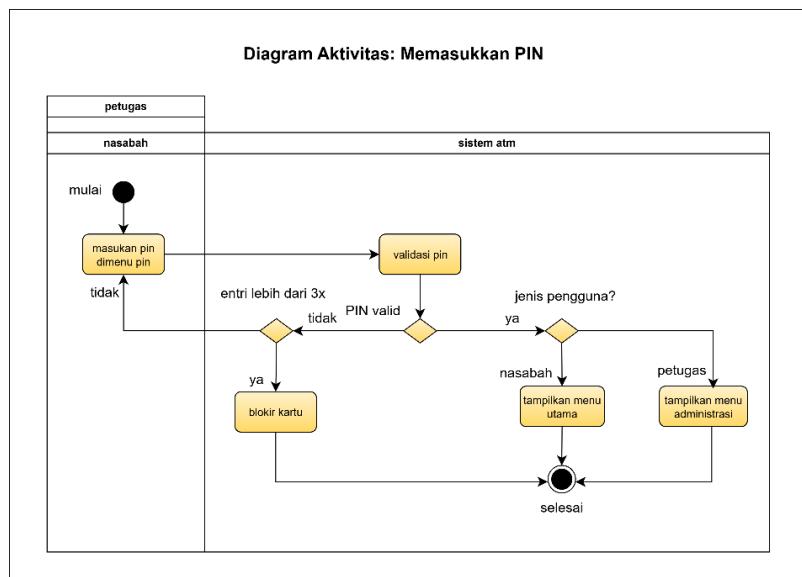


Activity Diagram:

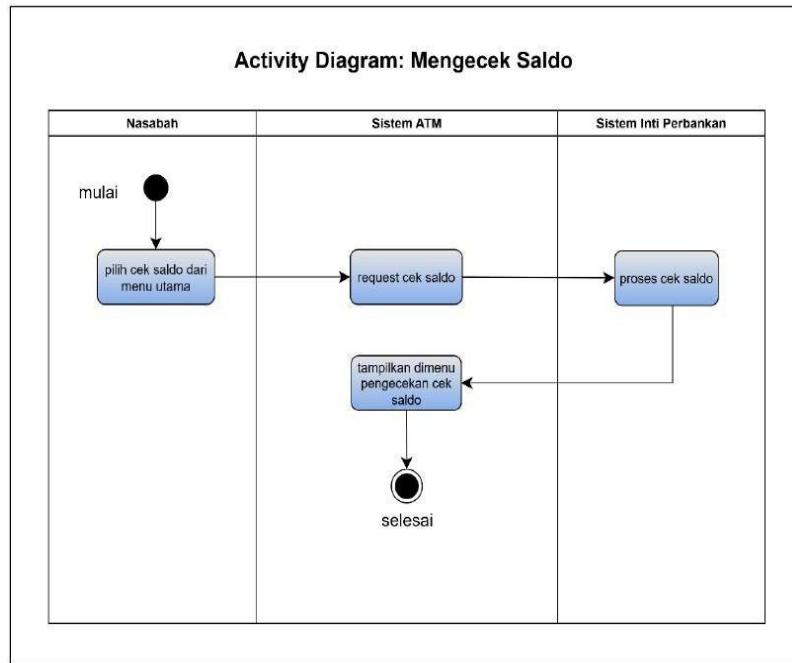
AD Memasukkan Kartu



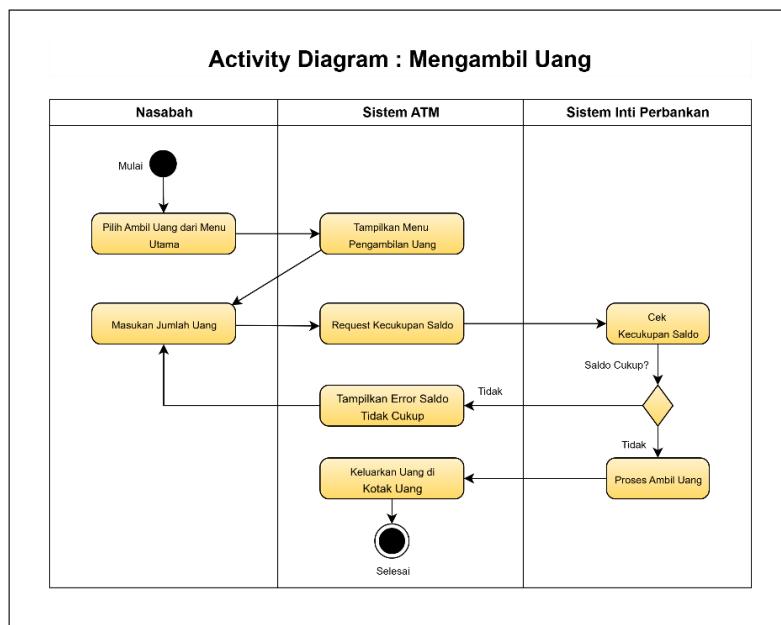
AD Memasukkan PIN



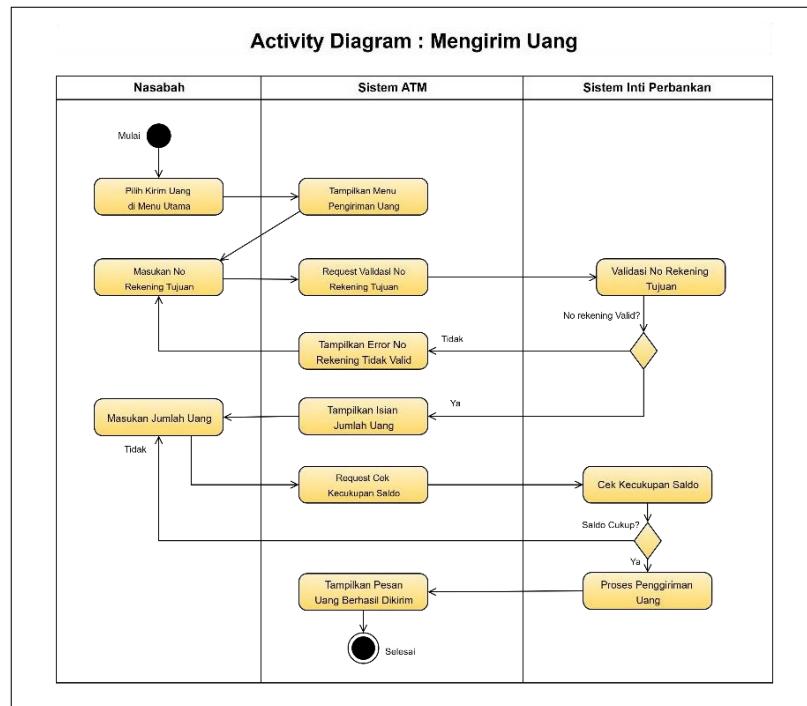
AD Melihat Saldo



AD Mengambil Uang

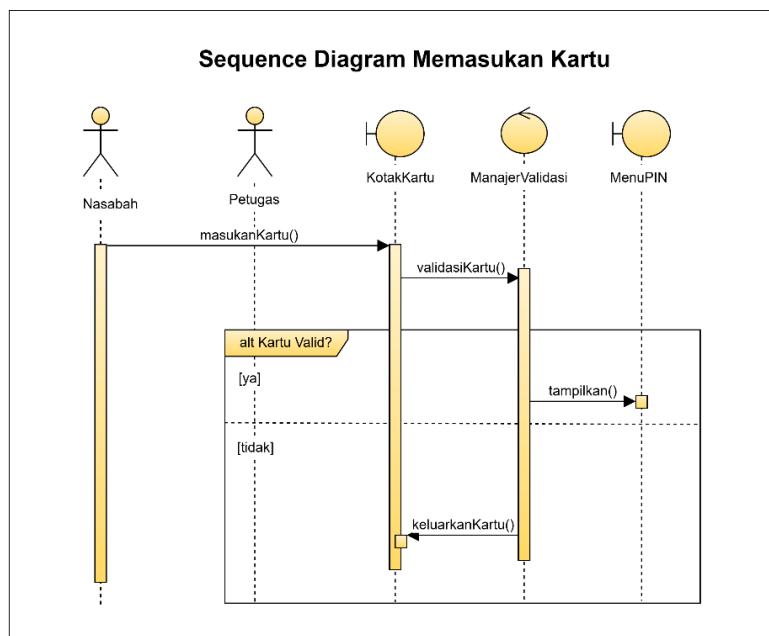


AD Mengirim/mentransfer Uang

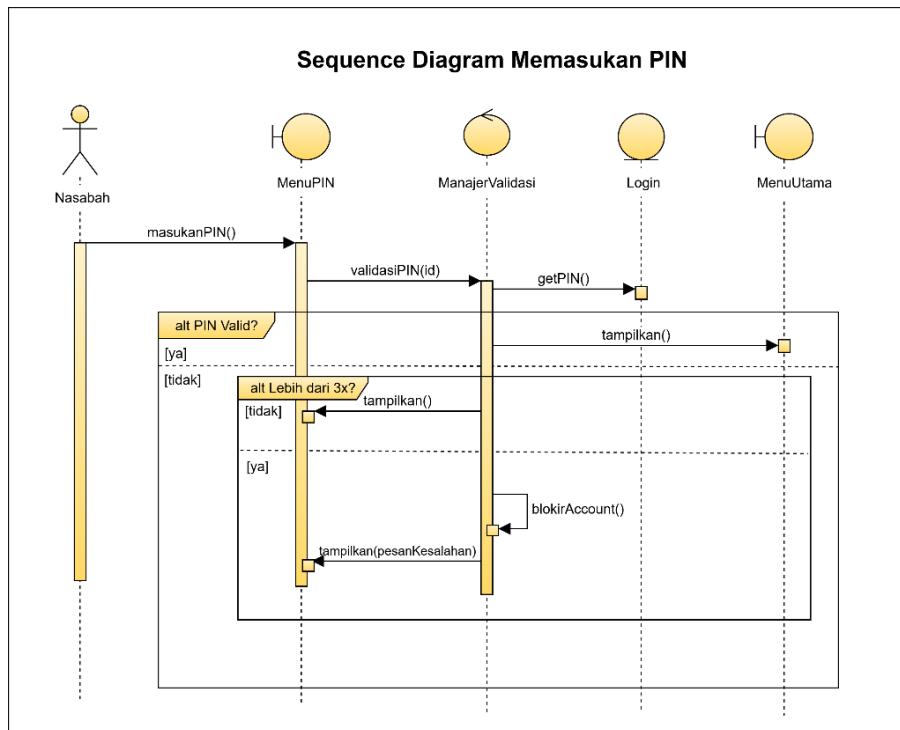


Sequence Diagram:

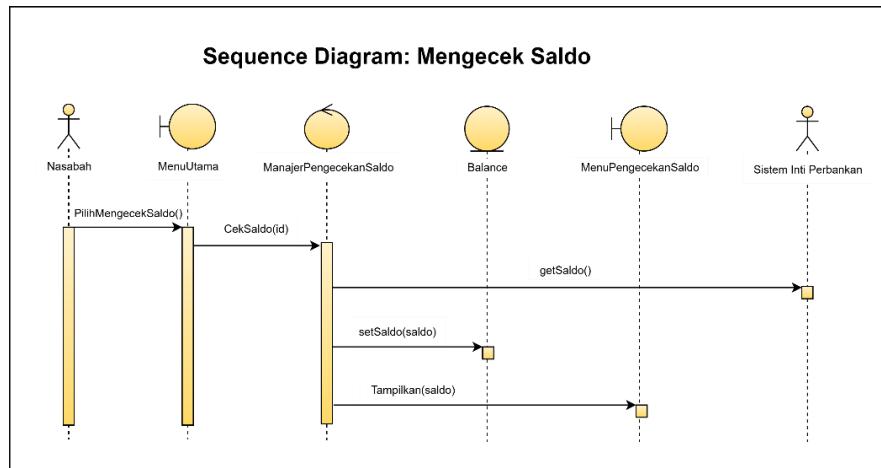
SD Memasukkan Kartu



SD Memasukkan PIN

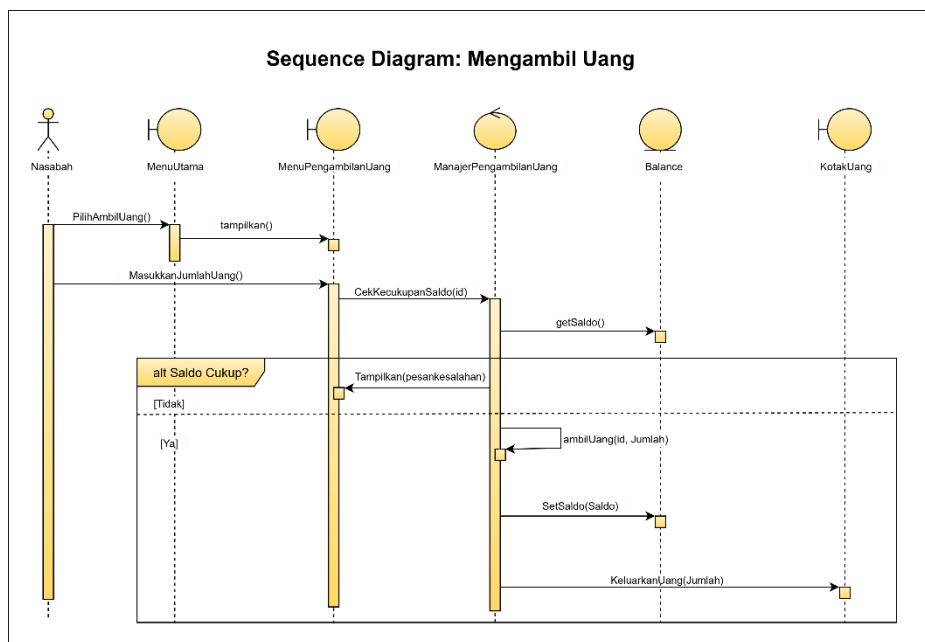


SD Melihat Saldo

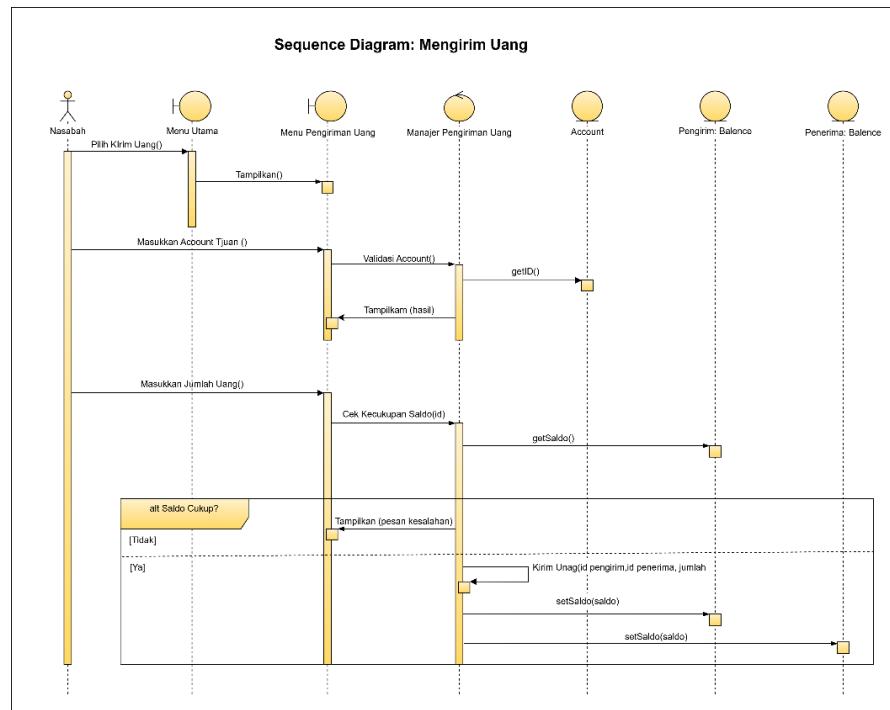


SD Mengambil Uang

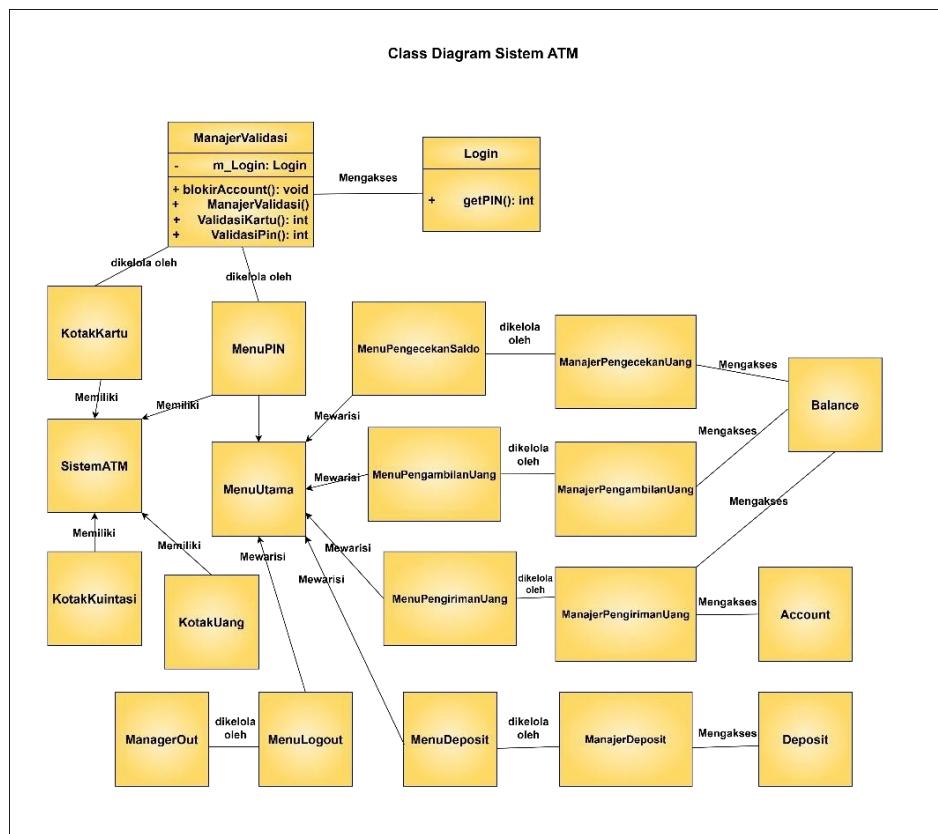
SD



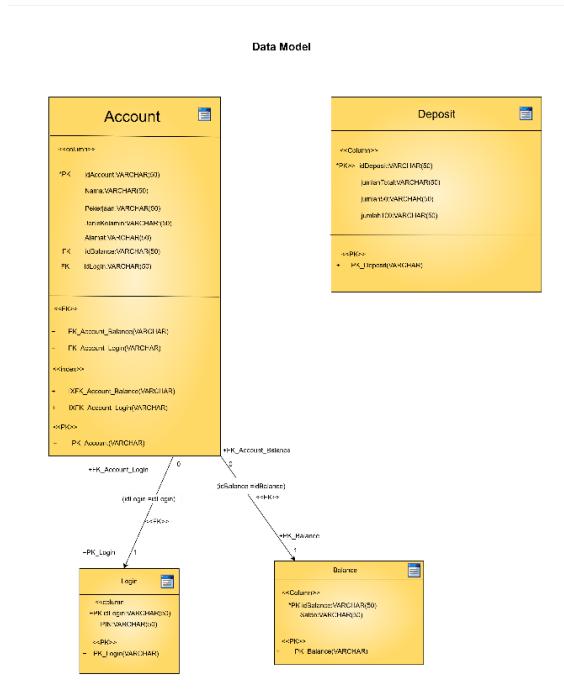
Mengirim/mentransfer Uang



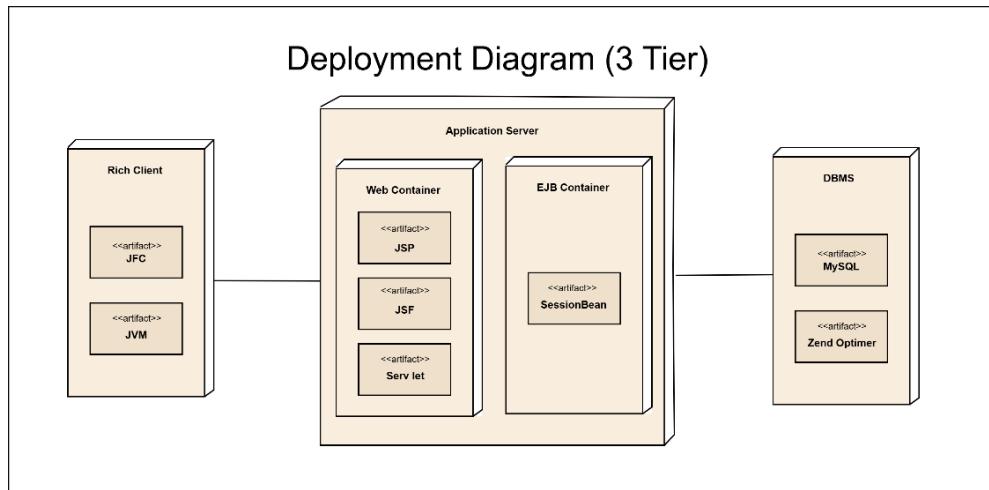
Class Diagram



Data Model

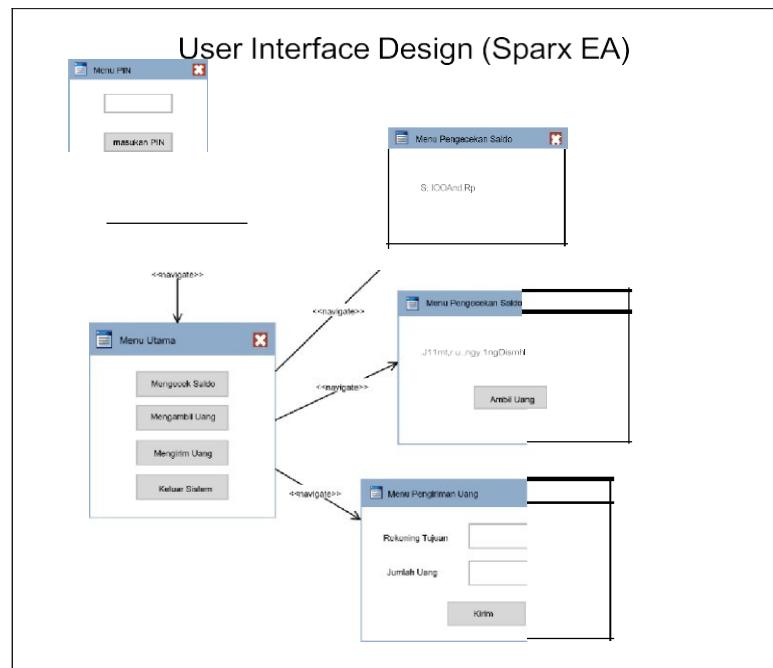


Deployment Diagram



User Interface





User Interface Design (Netbeans)

