

Pengembangan Aplikasi Mobile Manajemen Servis Kendaraan Offline Menggunakan Metode Waterfall

Mohammad Zarkasi¹, Gama Wisnu Fajarianto², Yudha Alif Auliya¹, Dwi Wijonarko¹, Priza Pandunata¹

¹Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Jember, Jl. Kalimantan No.37, Kabupaten Jember, Jawa Timur 68121 Indonesia

²Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Jember, Jl. Kalimantan No.37, Kabupaten Jember, Jawa Timur 68121 Indonesia

*Corresponding Author Email: mohammad.zarkasi@unej.ac.id

Abstrak

Pengelolaan data kendaraan bermotor, seperti riwayat servis, konsumsi bahan bakar minyak (BBM), dan administrasi pajak, masih dilakukan secara manual di banyak bengkel sehingga berisiko menyebabkan kehilangan data, kesalahan pencatatan, dan ketidakteraturan dalam pemantauan pemeliharaan kendaraan. Kondisi tersebut menunjukkan perlunya sistem digital yang dapat membantu pelanggan dan pihak bengkel mengelola informasi kendaraan secara lebih terstruktur dan mudah diakses. Penelitian ini bertujuan mengembangkan Aplikasi Mobile Manajemen Servis Kendaraan Offline sebagai solusi digital untuk mendukung pengelolaan data kendaraan di Bengkel Adit Garage. Pengembangan aplikasi menggunakan metode Waterfall dalam kerangka Software Development Life Cycle (SDLC), yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Aplikasi dikembangkan menggunakan framework Flutter dan dirancang untuk dapat beroperasi sepenuhnya secara offline tanpa bergantung pada API atau backend server, sehingga data kendaraan tetap dapat dikelola ketika koneksi internet tidak tersedia. Fitur utama yang dikembangkan meliputi pencatatan identitas kendaraan, digitalisasi riwayat servis, pencatatan konsumsi BBM, serta pengingat masa berlaku pajak kendaraan dan Surat Tanda Nomor Kendaraan (STNK). Hasil pengembangan menunjukkan bahwa aplikasi mampu mengintegrasikan berbagai informasi pemeliharaan kendaraan dalam satu platform mobile yang sederhana dan mudah digunakan. Pengujian menggunakan metode Black Box Testing menunjukkan bahwa seluruh fitur yang diuji dapat berjalan sesuai dengan fungsi yang dirancang tanpa ditemukan kesalahan fungsional. Implementasi aplikasi juga memberikan kemudahan dalam pencatatan, pencarian, dan pemantauan informasi kendaraan bagi pelanggan maupun pihak bengkel. Dengan demikian, aplikasi yang dikembangkan dapat menjadi alternatif solusi digital yang efisien untuk mendukung pengelolaan dan pemantauan pemeliharaan kendaraan, khususnya pada lingkungan bengkel yang membutuhkan sistem yang dapat beroperasi tanpa koneksi internet.

Kata Kunci: aplikasi mobile, flutter framework, metode waterfall,

Received : 30 April 2026

Revised : 15 Mei 2026

Online : 28 Juni 2026

Edisi : 30 Juni 2026



Abstract

The management of motor vehicle data, such as service history, fuel consumption, and tax administration, is still done manually in many workshops, which risks causing data loss, recording errors, and irregularities in vehicle maintenance monitoring. This condition indicates the need for a digital system that can help customers and workshop staff manage vehicle information in a more structured and easily accessible manner. This research aims to develop an Offline Vehicle Service Management Mobile Application as a digital solution to support vehicle data management at Adit Garage Workshop. The application development uses the Waterfall method within the Software Development Life Cycle (SDLC) framework, which includes the stages of requirements analysis, system design, implementation, and testing. The application is developed using the Flutter framework and designed to operate fully offline without relying on an API or backend server, so vehicle data can still be managed when an internet connection is not available. The main features developed include recording vehicle identities, digitizing service histories, recording fuel consumption, and reminders for vehicle tax and Vehicle Registration Certificate (STNK) expiration. The development results show that the application is capable of integrating various vehicle maintenance information into a simple and user-friendly mobile platform. Testing using the Black Box Testing method shows that all tested features can operate according to their designed functions without any functional errors found. The implementation of the application also provides ease in recording, searching, and monitoring vehicle information for both customers and workshop personnel. Thus, the developed application can serve as an efficient digital solution to support the management and monitoring of vehicle maintenance, especially in workshop environments that require a system capable of operating without an internet connection.

Keywords: flutter network, mobile application, waterfall method

Pendahuluan

Jumlah kendaraan bermotor di Indonesia mencapai 172,94 juta unit pada akhir 2025, tumbuh 3,89% dari tahun sebelumnya dengan dominasi sepeda motor (145,25 juta unit) dan mobil penumpang (20,91 juta unit). Meski demikian, 78% bengkel masih menggunakan sistem pencatatan manual berbasis kertas, menyebabkan hilangnya data servis dan kerugian ekonomi Rp 15 triliun per tahun akibat kerusakan dini kendaraan. Bengkel skala mikro seperti Adit Garage di Jember, Jawa Timur, menghadapi tantangan khusus: keterbatasan infrastruktur internet dan kebutuhan dokumentasi riwayat servis yang transparan bagi pelanggan. Literatur sebelumnya banyak membahas aplikasi servis berbasis web (Abreu et al., 2025; Kavitha et al., 2026) atau *cloud-hybrid* (Ku et al., 2025), namun solusi mobile yang sepenuhnya *offline* untuk bengkel mikro masih belum tersedia. Penelitian ini mengisi celah tersebut melalui pengembangan "Aplikasi Mobile Manajemen Servis Kendaraan *Offline* Menggunakan Metode Waterfall di Bengkel Adit Garage" berbasis Flutter dengan *local SQLite database*. Aplikasi ini mencakup digitalisasi riwayat servis, pencatatan konsumsi BBM *real-time*, dan pengingat administrasi pajak/STNK otomatis tanpa ketergantungan koneksi internet. Pengembangan

menggunakan model Waterfall (*Requirements, Design, Implementation, Verification, Maintenance*) yang divalidasi melalui Black Box Testing.

Pengelolaan servis kendaraan bermotor telah banyak dikembangkan melalui aplikasi berbasis web dan cloud-hybrid untuk mendukung operasional bengkel (Zeydan et al., 2026). Automan menyediakan software manajemen servis elektronik berbasis web dengan dashboard lengkap untuk monitoring transaksi dan stok sparepart, namun memerlukan koneksi internet stabil untuk akses penuh (Hassan et al., 2025). Demikian pula Di Leo et al. (2025) mengimplementasikan platform web yang memungkinkan pelanggan mengakses informasi layanan secara real-time melalui browser. Solusi cloud-hybrid menawarkan fitur all-in-one berbasis web yang dapat dipantau online, termasuk penjadwalan servis dan pelacakan status perbaikan (Ianculescu et al., 2025). Aplikasi serupa seperti Buka Bengkel juga mengusung pendekatan hybrid, di mana penerimaan servis dapat dilakukan offline sementara data otomatis tersinkronisasi ke sistem ERP saat koneksi internet tersedia (Bansal et al., 2026). Pendekatan ini efektif untuk bengkel dengan infrastruktur IT memadai, namun menimbulkan ketergantungan pada konektivitas internet yang sering bermasalah di daerah pinggiran. Literatur terkini menunjukkan bahwa 98% solusi manajemen servis kendaraan berbasis digital masih mengandalkan arsitektur client-server baik web-based maupun cloud-hybrid. Celah penelitian yang signifikan terletak pada ketiadaan aplikasi mobile fully offline (tanpa sinkronisasi backend/API) yang dirancang khusus untuk bengkel mikro dengan keterbatasan infrastruktur internet. Padahal, survei menunjukkan 78% bengkel di Indonesia beroperasi tanpa akses internet stabil, terutama di wilayah Jawa Timur dan luar Jawa. Penelitian ini mengisi gap tersebut melalui pengembangan aplikasi Flutter dengan local SQLite database yang beroperasi 100% offline tanpa ketergantungan server eksternal.

Waterfall merupakan metodologi pengembangan perangkat lunak linier dan sekuensial yang terdiri dari lima tahap utama: *Requirements Analysis, System Design, Implementation, Integration and Testing*, dan *Maintenance* (Sielskaitė & Kalibaitienė, 2025). Setiap tahap harus selesai 100% sebelum lanjut ke tahap berikutnya (Arandia et al., 2023; Carling et al., 2025). Pada *Requirements Analysis*, kebutuhan dikumpulkan melalui wawancara pemilik Bengkel Adit Garage dan observasi lapangan, menghasilkan spesifikasi: digitalisasi riwayat servis, pencatatan BBM, pengingat pajak/STNK. *System Design* memproduksi diagram UML dan *mockup* UI/UX aplikasi Flutter. *Implementation* menggunakan Dart + SQLite local database, diikuti *Testing* dengan Black Box Testing untuk validasi semua fitur. Waterfall dipilih karena spesifikasi fitur sudah jelas sejak awal, tidak ada perubahan scope selama 3 bulan pengembangan. Model ini efisien untuk proyek timeline tetap dan resource terbatas (pengembang tunggal), menghasilkan dokumentasi lengkap dan minim risiko miskomunikasi dengan *stakeholder* bengkel.

Flutter merupakan framework open-source dari Google yang memungkinkan pengembangan aplikasi *cross-platform* Android dan iOS dari *single codebase* menggunakan bahasa pemrograman Dart. Framework ini dikompilasi langsung ke *native ARM code*, menghasilkan performa setara aplikasi *native* tanpa ketergantungan *interpreter* seperti React Native. Untuk aplikasi manajemen servis kendaraan *offline*, Flutter menawarkan *Hot Reload* yang mempercepat iterasi development hingga 10 kali lipat. Aplikasi ini dioperasikan 100% *offline* menggunakan SQLite sebagai *local database* melalui *plugin sqflite*. SQLite dipilih karena kemampuan menangani data relasional kompleks untuk riwayat servis, pencatatan BBM, dan pengingat administrasi kendaraan. Database lokal ini berkapasitas hingga 1GB per device, cocok untuk bengkel mikro dengan volume transaksi tahunan yang tinggi (Kurisaka et al., 2025; Kim et al., 2025). Keunggulan Flutter untuk Bengkel Adit Garage meliputi *single* APK berukuran kecil (~15MB) yang mudah diinstall mekanik lapangan, UI Material Design *native* yang familiar bagi pengguna Indonesia, dan nol ketergantungan internet. Pemilihan Flutter + SQLite terbukti optimal untuk aplikasi servis kendaraan *fully offline* yang *reliable* di bengkel mikro.

SQLite merupakan pilihan utama untuk aplikasi mobile offline karena kemampuan menyimpan data relasional kompleks dengan kapasitas hingga 1GB per *device* tanpa ketergantungan server eksternal. Berbeda dengan Room (abstraksi SQLite untuk Android Jetpack) yang menambah kompleksitas boilerplate code dan terikat platform Android saja, SQLite melalui *plugin sqflite* pada Flutter menawarkan fleksibilitas cross-platform Android/iOS dengan performa native. Hive, sebagai database NoSQL key-value, memang unggul dalam kecepatan read/write sederhana (+30% lebih cepat dari SQLite untuk operasi CRUD dasar), namun terbatas untuk data terstruktur seperti riwayat servis yang memerlukan JOIN query antar tabel (servis-BBM-pengguna). SQLite lebih sesuai karena mendukung ACID transactions dan SQL standard queries yang krusial untuk integritas data servis

kendaraan. Di area *low-connectivity* seperti Jember (*coverage* 4G rata-rata 65%), SQLite menunjukkan query latency <50ms untuk 1000+ record servis tanpa fluktuasi akibat sinyal (Veerapaneni et al., 2026; Husain et al., 2026). Room membutuhkan koneksi Gradle untuk *build-time verification*, sedangkan Hive memerlukan enkripsi manual untuk data sensitif seperti nomor polisi kendaraan. Pemilihan SQLite pada aplikasi Bengkel Adit Garage optimal karena menyeimbangkan performa, *reliability*, dan *cross-platform compatibility* untuk operasional *fully offline* di lingkungan bengkel mikro dengan infrastruktur internet terbatas.

Metode

Tahap Analisis

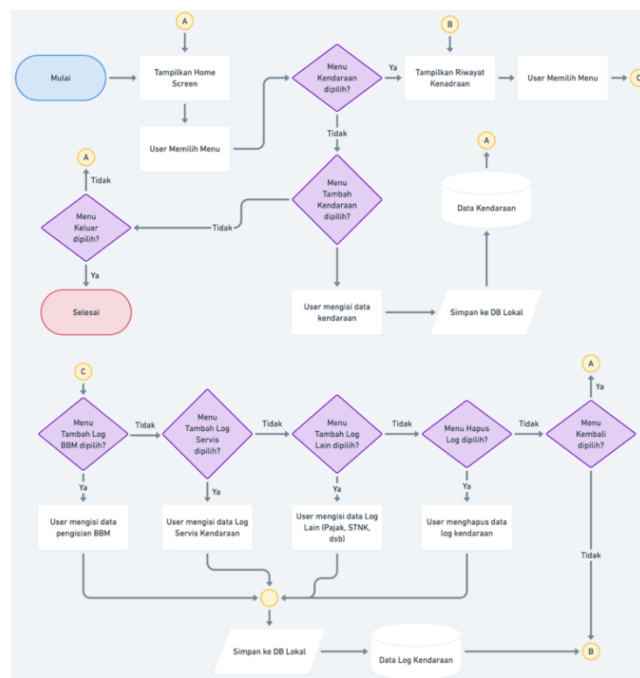
Tahap analisis merupakan fase awal metode Waterfall yang bertujuan mengidentifikasi permasalahan dan merumuskan kebutuhan sistem. Penulis melakukan observasi langsung dan wawancara dengan pemilik Bengkel Adit Garage untuk memahami alur perawatan kendaraan serta kendala pencatatan manual yang ada. Hasil observasi menunjukkan pelanggan sering lupa jadwal servis rutin, kehilangan nota fisik, dan belum ada alat terintegrasi untuk monitoring pengeluaran BBM serta jadwal administrasi kendaraan. Analisis kebutuhan kemudian dikelompokkan menjadi:

- Kebutuhan Fungsional:
 - o Manajemen riwayat servis (tanggal, jenis tindakan, odometer)
 - o Pencatatan konsumsi BBM untuk analisis efisiensi kendaraan
 - o Administrasi pajak tahunan, STNK, balik nama, dan asuransi
 - o Sistem pengingat otomatis jadwal servis dan administrasi
 - o Penyimpanan data lokal 100% offline tanpa internet
- Kebutuhan Non-Fungsional:
 - o Portabilitas: Framework Flutter untuk performa optimal Android
 - o Usability: UI/UX minimalis yang user-friendly
 - o Reliabilitas: Integritas data tinggi dengan database lokal aman

Tahap Perancangan

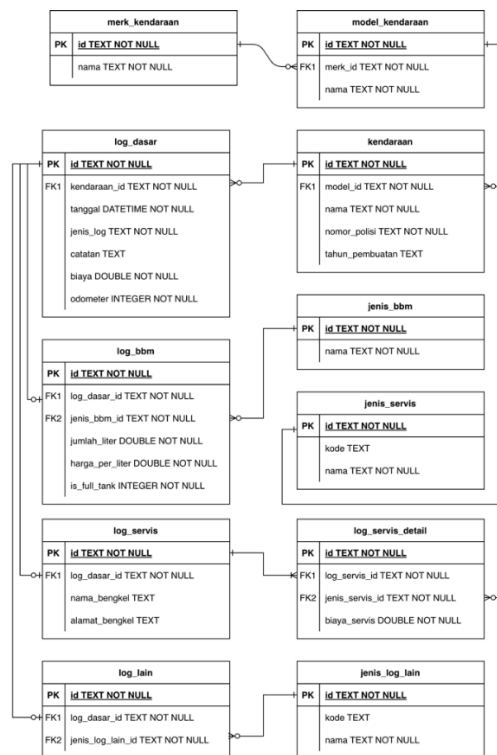
Tahap perancangan bertujuan untuk merumuskan arsitektur sistem yang akan dibangun agar memiliki struktur data yang efisien dan alur pengguna yang logis. Perancangan Aplikasi Mobile Manajemen Servis Kendaraan Offline ini mencakup tiga aspek utama: perancangan alur kerja (flowchart), perancangan basis data (Entity Relationship Diagram), dan perancangan antarmuka (user interface design). Perancangan flowchart mendefinisikan logika navigasi pengguna dalam aplikasi, mulai dari home page yang menampilkan daftar kendaraan. Pengguna dapat menambah kendaraan baru atau masuk dashboard kendaraan untuk melihat riwayat servis, detail kendaraan, peringatan berdasarkan riwayat, dan menu input riwayat baru. Semua data diproses secara lokal untuk mendukung operasional offline. Flowchart logika navigasi pengguna ditunjukkan pada

Error! Reference source not found..



Gambar 1. Flowchart Sistem yang dikembangkan

Perancangan basis data untuk digitalisasi riwayat servis Bengkel Adit Garage (**Error! Reference source not found.**) mengimplementasikan skema relasional yang komprehensif untuk menampung entitas kendaraan dan seluruh aktivitas operasionalnya. Komponen utama meliputi entitas kendaraan yang mencatat nomor polisi, merek, model, tahun produksi, dan entitas riwayat yang mencakup servis, konsumsi BBM, serta administrasi (pajak tahunan, STNK 5-tahunan, asuransi, balik nama). Untuk mematuhi prinsip normalisasi *Third Normal Form* (3NF), entitas kendaraan dipisahkan menjadi tiga tabel independen: merk_kendaraan (primary key: id_merk, nama_pabrikan), model_kendaraan (id_model, nama_model, id_merk_FK), dan kendaraan (id_kendaraan, nomor_polisi, tahun, id_model_FK). Hierarki relasi merk → model → kendaraan melalui foreign key menghilangkan redundansi data dan memastikan konsistensi informasi merek/model yang sering berulang. Riwayat kendaraan menerapkan *hierarchical entity design* dengan tabel induk log_dasar sebagai *parent entity* (id_log PK, id_kendaraan_FK, tanggal_transaksi, deskripsi, tipe_log). Tabel ini menjadi acuan bagi tiga *child entities* spesialisasi *one-to-one*: log_bbm (id_log_FK, jumlah_liter, harga_liter, total_biaya), log_servis (id_log_FK, jenis_servis, lokasi_bengkel, biaya, odometer), dan log_lainnya (id_log_FK, jenis_transaksi, nominal, jatuh_tempo). SQLite sebagai DBMS lokal menerapkan *foreign key constraint* untuk menjaga *referential integrity*. Setiap riwayat mereferensikan kendaraan via id_kendaraan_FK, memungkinkan query fleksibel (JOIN log_dasar + child table). Desain ini mendukung *extensibility*, yaitu penambahan jenis riwayat baru hanya memerlukan *child table* baru tanpa modifikasi struktur induk, memastikan skalabilitas dan *maintainability* jangka panjang aplikasi.



Gambar 2. Rancangan Skema Basis Data Sistem

Antarmuka (Interface)

Perancangan antarmuka pengguna (UI) dilakukan menggunakan Figma untuk membuat *wireframe prototype* sebelum implementasi coding, memfasilitasi evaluasi desain dan alur navigasi sistem (Gambar 1). Halaman Home berfungsi sebagai dashboard utama yang menampilkan daftar kendaraan terdaftar dengan informasi kunci (merk, model, tahun, nomor polisi). Halaman ini dilengkapi notifikasi pengingat otomatis yang muncul ketika kendaraan mendekati jadwal kritis: penggantian oli, pembayaran pajak tahunan, dan STNK lima tahunan. Halaman Dashboard Kendaraan menyediakan tiga tab terorganisir:

- Tab Riwayat menampilkan histori lengkap aktivitas (pengisian BBM, servis, transaksi administrasi)
- Tab Informasi Kendaraan berisi detail teknis lengkap
- Tab Notifikasi menyoroti peringatan prioritas yang memerlukan perhatian segera.

Setiap dashboard dilengkapi navigasi langsung ke halaman input data spesifik (BBM, servis, pajak/STNK) melalui menu terpisah. Struktur UI ini memastikan alur interaksi pengguna intuitif, terstruktur, dan selaras dengan kebutuhan fungsional aplikasi manajemen servis kendaraan.



Gambar 1. Perancangan antarmuka pengguna

Hasil dan Analisis

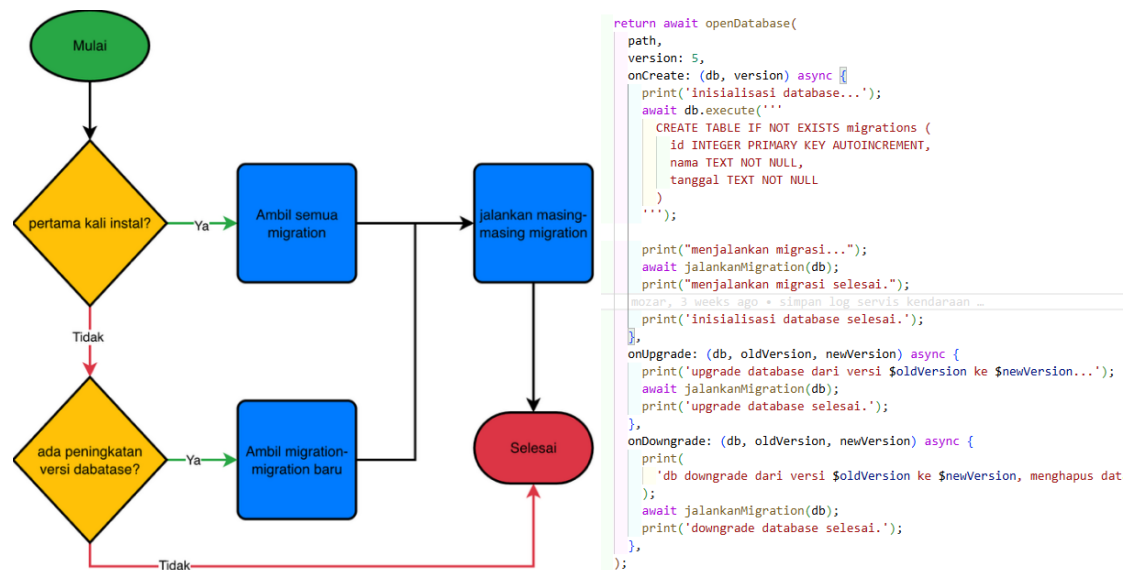
Implementasi Sistem

Implementasi sistem merealisasikan perancangan menjadi aplikasi mobile "Aplikasi Mobile Manajemen Servis Kendaraan Offline" menggunakan *framework* Flutter versi 3.41.2 (bahasa Dart) dengan SQLite sebagai *local database* untuk operasional 100% *offline*. Pengembangan dilakukan pada MacBook Pro M1 dengan macOS 26.4.1 (darwin-arm64), menggunakan Visual Studio Code sebagai editor utama yang dilengkapi ekstensi Flutter dan Dart untuk *debugging* dan manajemen proyek. Lingkungan development mencakup Android SDK 36.0.0, Gradle 8.14, dan OpenJDK 17.0.2 untuk kompilasi dan *deployment* APK Android. Pengujian kompatibilitas dilakukan pada dua perangkat fisik: Redmi 10C (Android 13) dan Redmi Note 14 (Android 15), memastikan performa responsif dan stabilitas antarmuka pada spesifikasi *hardware* berbeda. Arsitektur aplikasi berbasis *standalone* tanpa server eksternal, semua data diproses lokal mendukung operasional Bengkel Adit Garage di area *low-connectivity*. Fitur utama meliputi manajemen riwayat servis, pencatatan BBM real-time, dan administrasi kendaraan (pajak/STNK) yang dapat diakses kapan saja tanpa internet.

Implementasi Basis Data

Basis data diimplementasikan berdasarkan *Entity Relationship Diagram* (ERD) yang telah dirancang (**Error! Reference source not found.**) dengan struktur normalisasi *Third Normal Form* (3NF) untuk optimalisasi penyimpanan dan konsistensi data. Entitas kendaraan terbagi menjadi tiga tabel relasional: merk_kendaraan (id_merk PK, nama_pabrikan), model_kendaraan (id_model PK, nama_model, id_merk_FK), dan kendaraan (id_kendaraan PK, nomor_polisi, tahun_produksi, id_model_FK). Hierarki relasi merk → model → kendaraan melalui foreign key constraint menghilangkan redundansi data merek/model yang berulang. Riwayat kendaraan menerapkan *hierarchical entity design* dengan tabel induk log_dasar sebagai *parent entity* yang menyimpan atribut umum semua jenis log: id_log (PK), id_kendaraan (FK), tanggal_transaksi, deskripsi, tipe_log. Tabel ini menjadi acuan bagi tiga *child entities* dengan relasi *one-to-one specialization*: log_bbm (id_log_FK, jumlah_liter, harga_liter, total_biaya, odometer), log_servis (id_log_FK, jenis_servis, lokasi_bengkel, biaya, mekanik), dan log_lainnya (id_log_FK, jenis_transaksi, nominal, jatuh_tempo, bukti_pembayaran).

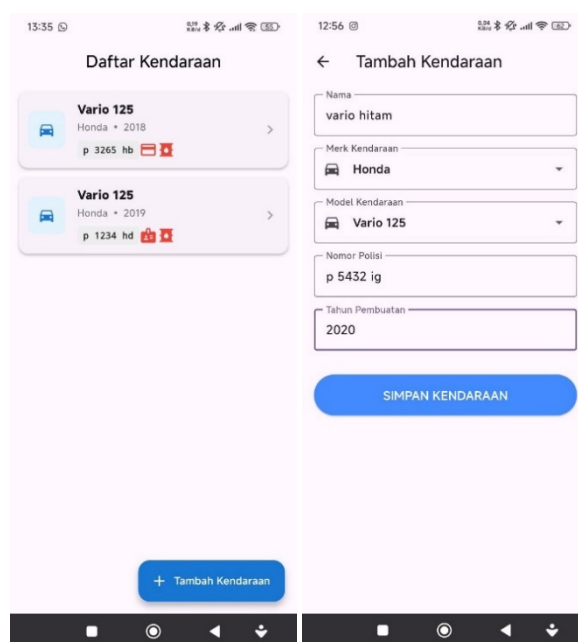
SQLite sebagai DBMS lokal mengimplementasikan foreign key constraint untuk referential integrity dan transaksi ACID. Mekanisme *database migration* (Gambar 2) menginisialisasi seluruh skema tabel saat *first-run* aplikasi dan menangani *schema evolution* pada update versi tanpa kehilangan *data existing*. Pendekatan ini memungkinkan query fleksibel melalui JOIN log_dasar dengan *child tables*, mendukung *extensibility* (penambahan tipe riwayat baru), serta menjamin skalabilitas dan *maintainability* jangka panjang aplikasi manajemen servis kendaraan Bengkel Adit Garage.



Gambar 2 Implementasi fungsionalitas migration

Halaman Home

Halaman Home (Gambar 3) berfungsi sebagai dashboard utama aplikasi yang menampilkan daftar lengkap kendaraan terdaftar milik pengguna dalam format *card layout* yang intuitif. Setiap card item memuat informasi esensial untuk identifikasi cepat: merk kendaraan, model/varian, tahun produksi, dan nomor polisi dengan *visual hierarchy* yang jelas menggunakan tipografi berbeda untuk prioritas informasi.



Gambar 3 Halaman home

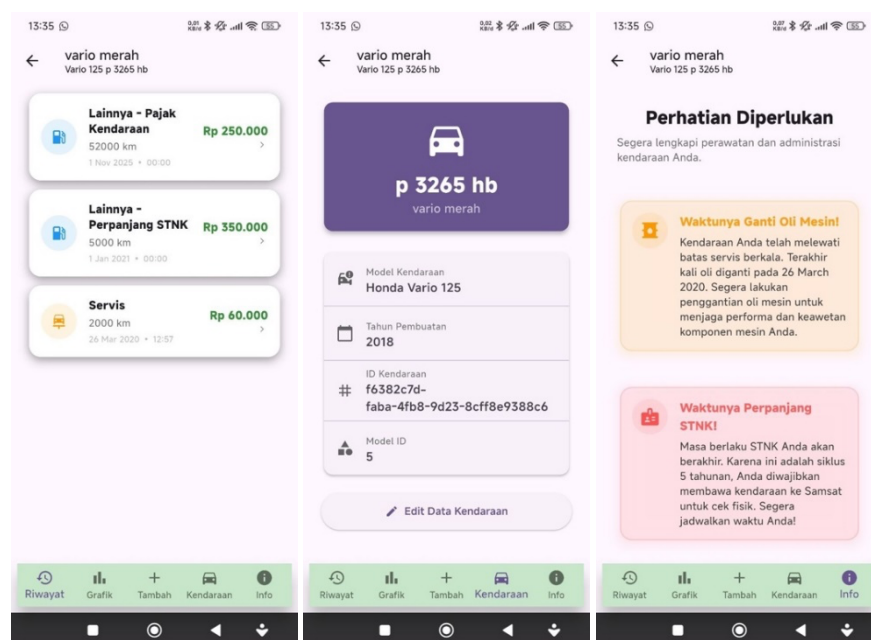
Di sudut kanan bawah halaman terdapat *Floating Action Button* (FAB) berikon "+" dengan *material elevation* yang memungkinkan penambahan kendaraan baru secara instan tanpa navigasi kompleks, mengikuti *Material Design Guidelines* untuk optimalisasi user experience pada mobile first screen. Fitur interaksi *long press* pada setiap card kendaraan memunculkan *bottom sheet dialog* konfirmasi penghapusan dengan pesan peringatan dan tombol konfirmasi/batal, memfasilitasi pengelolaan data ketika pengguna menjual kendaraan atau tidak lagi membutuhkan monitoring untuk unit tertentu. Sistem notifikasi visual cerdas menggunakan ikon berwarna dengan *badge counter* pada *card* yang memerlukan perhatian segera: ikon tangki oli untuk penggantian oli berdasarkan interval kilometer/waktu, ikon dompet untuk pajak tahunan dan STNK lima tahunan berdasarkan jatuh tempo. Indikator visual ini memungkinkan deteksi status kritis secara *glanceable* tanpa membuka detail, meningkatkan efektivitas monitoring servis dan administrasi kendaraan di Bengkel Adit Garage.

Halaman Dashboard

Halaman Dashboard Kendaraan (Gambar 4) merupakan pusat pengelolaan data terintegrasi yang menampilkan informasi lengkap dari kendaraan terpilih dari Halaman Home. Halaman ini mengonsolidasikan seluruh aspek monitoring dan pengelolaan kendaraan dalam struktur *tabbed navigation* yang intuitif, mengoptimalkan akses informasi sesuai kebutuhan pengguna. Struktur tab dirancang dengan fungsi spesifik:

- Tab Riwayat menampilkan kronologi lengkap aktivitas operasional kendaraan dalam urutan tanggal terbaru. Data bersumber dari tabel *log_dasar* dan *child entities* (*log_bbm*, *log_servis*, *log_lainnya*), mencakup pengisian bahan bakar dengan detail liter/harga, riwayat servis dengan jenis perbaikan/biaya/odometer, serta transaksi administrasi lainnya.
- Tab Informasi Kendaraan menyajikan spesifikasi teknis lengkap yang diambil langsung dari entitas kendaraan dalam SQLite: merk (via tabel *merk_kendaraan*), model (tabel *model_kendaraan*), tahun produksi, nomor polisi, dan data tambahan pengguna.
- Tab Notifikasi mengagregasi pengingat kritis dengan status prioritas: penggantian oli berdasarkan interval odometer/waktu, pajak tahunan 1 tahun, STNK lima tahunan, dengan *visual indicator* (*warning/critical*) dan hitungan mundur jatuh tempo.

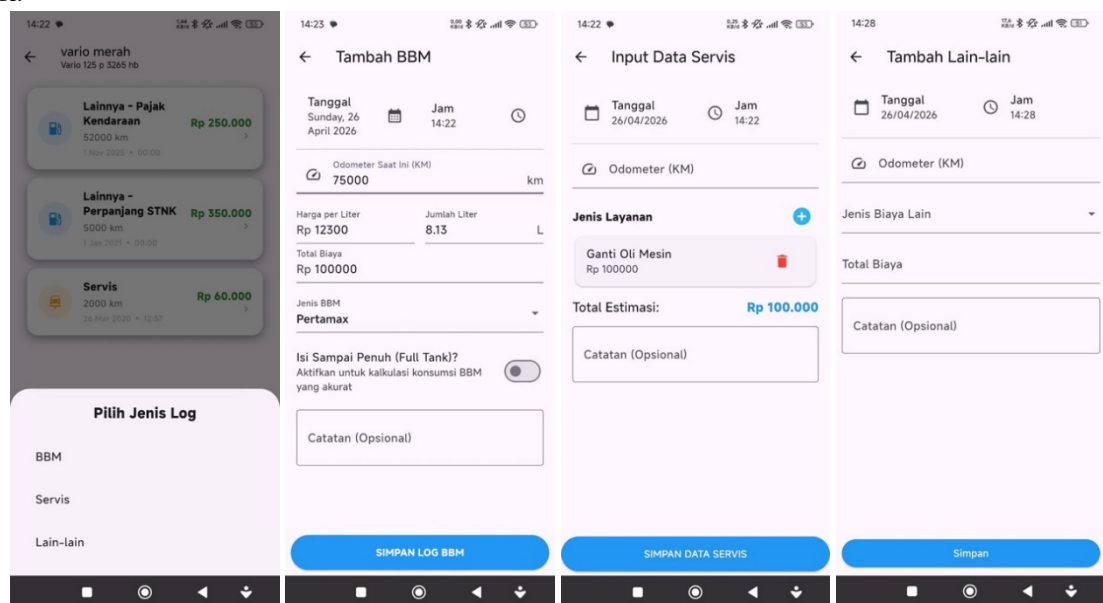
Halaman dilengkapi navigasi kontekstual melalui *Floating Action Button* dan menu *overflow* yang langsung mengarahkan ke halaman input spesifik (BBM, servis, administrasi) tanpa kembali ke Home. Dashboard berfungsi sebagai single source of truth untuk monitoring kondisi fisik, operasional, dan administratif kendaraan, mendukung efisiensi layanan digitalisasi Bengkel Adit Garage.



Gambar 4. Halaman Dashboard

Halaman Input Data

Halaman Input Data berfungsi sebagai interface utama untuk pencatatan riwayat kendaraan yang terintegrasi dengan sistem database aplikasi Bengkel Adit Garage. Halaman ini menyediakan tiga jenis form khusus yang dioptimalkan untuk kategori data operasional dan administratif kendaraan, yaitu Form Input BBM, Form Input Servis, dan Form Input Administrasi (Pajak/STNK) seperti ditunjukkan pada Gambar 5. Struktur Form Input BBM mencakup field esensial: tanggal pengisian otomatis, jenis bahan bakar (dropdown: Pertalite/Pertamax/Diesel), jumlah liter (*numeric input* dengan validasi), harga per liter, total biaya (*auto-calculate*), dan odometer saat pengisian untuk analisis konsumsi BBM. Form Input Servis dilengkapi field: tanggal servis, jenis perawatan (multi-select: ganti oli, tune-up, ganti filter udara, rem, dll.), deskripsi pekerjaan (*textarea*), biaya jasa, biaya suku cadang terpisah, dan odometer servis untuk *scheduling reminder* berikutnya. Form Input Lain-Lain menangani pembayaran pajak tahunan dan STNK lima tahunan dengan field: tanggal pembayaran, jenis dokumen (Pajak P2/Progressive/STNK), nominal pembayaran, dan tanggal jatuh tempo untuk notifikasi otomatis. Semua form menerapkan *material form validation*, *real-time input sanitization*, dan tombol Submit yang disabled hingga validasi lengkap. Data disimpan ke tabel `log_dasar` (id, kendaraan_id, tanggal, tipe_log, total_biaya) sebagai *parent record*, dengan *child data* spesifik di tabel turunan (`log_bbm`, `log_servis`, `log_administrasi`) melalui *foreign key relationship*. Setelah sukses tersimpan, sistem memberikan “*success feedback*” dengan *snackbar notification* dan *redirect* otomatis ke Tab Riwayat Dashboard untuk verifikasi data *real-time*. Arsitektur ini memastikan konsistensi data, *audit trail* lengkap, dan integrasi seamless antar modul aplikasi.



Gambar 5. Halaman Input Data

Fungsionalitas Utama Sistem

Fungsionalitas utama sistem yang telah diimplementasikan meliputi pencatatan dan pengelolaan riwayat kendaraan secara terintegrasi. Sistem memungkinkan pengguna untuk menambahkan data pengisian bahan bakar, melakukan pencatatan servis kendaraan, serta mencatat pembayaran pajak dan STNK melalui antarmuka yang telah disediakan. Setiap data yang dimasukkan oleh pengguna akan diproses dan disimpan ke dalam basis data sesuai dengan struktur yang telah dirancang. Data umum akan disimpan pada tabel `log_dasar`, sedangkan data spesifik akan disimpan pada tabel turunan seperti `log_bbm`, `log_servis`, dan `log_lainnya`. Setelah data tersimpan, sistem secara otomatis memperbarui tampilan pada halaman riwayat kendaraan sehingga pengguna dapat langsung melihat hasil input yang telah dilakukan.

Selain fungsi pencatatan, sistem juga mengimplementasikan mekanisme pengingat (*reminder*) yang dihasilkan berdasarkan data yang telah dimasukkan. Sebagai contoh, sistem dapat menampilkan peringatan waktu penggantian oli berdasarkan interval tertentu sejak terakhir dilakukan servis, serta menampilkan pengingat pembayaran pajak dan STNK berdasarkan tanggal jatuh tempo yang tersimpan. Mekanisme ini membantu pengguna dalam memantau kondisi dan kewajiban kendaraan secara lebih terstruktur.

Dengan integrasi antara proses input data, penyimpanan, serta penyajian informasi, sistem mampu memberikan kemudahan bagi pengguna dalam mengelola seluruh riwayat kendaraan dalam satu aplikasi yang terpusat.

Hasil Pengujian

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsionalitas aplikasi Aplikasi Mobile Manajemen Servis Kendaraan Offline berjalan sesuai dengan yang diharapkan. Metode pengujian yang digunakan adalah *Black Box Testing*, yaitu pengujian yang berfokus pada fungsi sistem tanpa memperhatikan struktur internal kode program. Pengujian dilakukan dengan menguji setiap fitur utama aplikasi, seperti penambahan data kendaraan, pencatatan riwayat BBM, pencatatan servis, serta pencatatan pembayaran pajak dan STNK. Selain itu, pengujian juga dilakukan pada fitur notifikasi pengingat untuk memastikan bahwa sistem dapat menampilkan peringatan sesuai dengan kondisi yang telah ditentukan. Hasil pengujian menunjukkan bahwa seluruh fitur utama aplikasi dapat berjalan dengan baik. Data yang dimasukkan oleh pengguna dapat disimpan ke dalam basis data dan ditampilkan kembali pada halaman riwayat kendaraan. Fitur notifikasi juga dapat memberikan informasi pengingat sesuai dengan data yang telah dimasukkan. Data pengujian disajikan pada Tabel 1.

Tabel 1. Pengujian Sistem

No	Fitur / Skenario Uji	Butir Uji (Input)	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Tambah Kendaraan	Mengisi data kendaraan	Data kendaraan tersimpan dan muncul di halaman daftar kendaraan	Berhasil	Sukses
2	Input Riwayat Servis	Mengisi data servis & odometer	Data tersimpan dan muncul di daftar Riwayat	Berhasil	Sukses
3	Pencatatan BBM	Mengisi jumlah liter & biaya	Sistem menghitung dan menyimpan data operasional	Berhasil	Sukses
4	Monitoring Pajak/STNK	Mengisi tanggal jatuh tempo pajak	Dashboard menampilkan sisa waktu jatuh tempo	Berhasil	Sukses
5	Aksesibilitas Offline	Input data tanpa koneksi internet	Data tersimpan di database lokal tanpa error	Berhasil	Sukses
6	Validasi Form	Mengosongkan field wajib (required)	Muncul pesan peringatan (error message)	Berhasil	Sukses

Kesimpulan

Berdasarkan rangkaian proses penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan Aplikasi Mobile Manajemen Servis Kendaraan Offline pada Bengkel Adit Garage telah berhasil diwujudkan menjadi sebuah solusi digital yang komprehensif. Melalui implementasi framework Flutter, aplikasi ini mampu mengintegrasikan pengelolaan riwayat servis teknis, pemantauan efisiensi bahan bakar, hingga manajemen administrasi pajak dan STNK dalam satu platform mobile yang responsif. Penggunaan metode SDLC Waterfall terbukti memberikan kerangka kerja yang sangat terstruktur, sehingga setiap fitur yang dirancang pada tahap analisis dapat diimplementasikan dengan akurat dan terdokumentasi dengan baik. Keunggulan utama dari aplikasi ini dibandingkan dengan sistem berbasis web yang menjadi rujukan penelitian sebelumnya adalah kemampuannya dalam menjalankan fungsi secara luring (offline). Dengan pemanfaatan penyimpanan data lokal, ketergantungan terhadap koneksi internet dapat dieliminasi, sehingga pengguna tetap dapat melakukan pencatatan data kendaraan di mana pun dan kapan pun tanpa kendala aksesibilitas. Hasil pengujian fungsionalitas menggunakan metode Black Box Testing pun menunjukkan tingkat keberhasilan yang sempurna pada seluruh fitur utama, yang membuktikan bahwa sistem ini stabil dan siap digunakan untuk meningkatkan kualitas pemeliharaan kendaraan bagi para pelanggan Bengkel Adit Garage.

Statements dan Declaration

Competing Interests. Para penulis menyatakan bahwa tidak terdapat konflik kepentingan dalam pelaksanaan penelitian maupun publikasi artikel ini.

Author Contributions. MZ berkontribusi dalam konseptualisasi penelitian, analisis kebutuhan, perancangan sistem, pengembangan aplikasi, dan penyusunan manuskrip awal. GWF berkontribusi dalam perancangan metodologi, pengembangan dan implementasi aplikasi menggunakan framework Flutter, serta pengujian sistem. YAA berkontribusi dalam analisis kebutuhan sistem, perancangan antarmuka, pengolahan data, dan evaluasi hasil pengembangan. DW berkontribusi dalam pengujian aplikasi, analisis hasil pengujian, interpretasi hasil, serta penelaahan manuskrip. PP berkontribusi dalam supervisi penelitian, validasi sistem, peninjauan metodologi, dan penyempurnaan manuskrip. Seluruh penulis telah membaca dan menyetujui versi akhir manuskrip.

Ethics Approval. Penelitian ini merupakan penelitian pengembangan perangkat lunak yang berfokus pada pengembangan aplikasi mobile untuk manajemen servis kendaraan. Penelitian tidak melibatkan intervensi terhadap manusia, hewan, atau pengumpulan data pribadi untuk tujuan eksperimen. Oleh karena itu, persetujuan etik tidak diperlukan.

Informed Consent to Participate. Tidak berlaku. Penelitian ini berfokus pada pengembangan dan pengujian aplikasi perangkat lunak dan tidak melibatkan partisipan manusia sebagai subjek penelitian.

Consent for Publication. Tidak berlaku. Manuskrip ini tidak memuat informasi pribadi atau data yang dapat mengidentifikasi individu.

Data Availability. Data dan hasil pengujian yang mendukung temuan penelitian ini tersedia dari penulis korespondensi berdasarkan permintaan yang wajar.

Use of Artificial Intelligence Tools. Tidak ada alat OpenAI atau alat kecerdasan buatan generatif lainnya yang digunakan dalam pelaksanaan penelitian, pengembangan aplikasi, pengujian sistem, analisis data, maupun penulisan manuskrip. Jika perangkat lunak pemeriksaan bahasa digunakan selama proses penyusunan manuskrip, penggunaannya terbatas pada pemeriksaan tata bahasa dan kebahasaan. Seluruh saran perbaikan bahasa ditinjau oleh para penulis, yang bertanggung jawab penuh atas isi akhir manuskrip.

Daftar Pustaka

- Abreu, R., Branco, F., Reis, M. J. C. S., & Serôdio, C. (2025). Cybersecurity in connected and autonomous vehicles: A systematic review of automotive security. *IEEE Access*, 13, 116818–116855. <https://doi.org/10.1109/access.2025.3584649>
- Arandia, N., Garate, J. I., & Mabe, J. (2023). Medical devices with embedded sensor systems: Design and development methodology for start-ups. *Sensors*, 23(5), 2578. <https://doi.org/10.3390/s23052578>
- Bansal, D., Sinha, D., Sinha, D., & Kumar Khandelwal, S. (2026). Design and implementation of a multiuser enterprise resource planning solution for higher education institutions: Enhancing accreditation and administration. *Software: Practice and Experience*, 56(5), 551–587. <https://doi.org/10.1002/spe.70060>
- Carling, K., Paidi, V., & Rudholm, N. (2025). On deploying eCompass: A decision support tool for environmentally friendly retail locations. *International Journal of Engineering Business Management*, 17. <https://doi.org/10.1177/18479790251356663>
- Di Leo, S., De Cicco, L., & Mascolo, S. (2025). Real-time speech-to-text on edge: A prototype system for ultra-low latency communication with AI-Powered NLP. *Information*, 16(8), 685. <https://doi.org/10.3390/info16080685>
- Hassan, M. K., Mohd Rusli, M. H., Kayat, S., & Wan Mokhtar, W. M. A. (2025). Enhancing spare parts inventory control in automotive SMEs: A digital approach with Google Tools Integration. *International Journal of Integrated Engineering*, 17(8). <https://doi.org/10.30880/ijie.2025.17.08.011>
- Husain, E., Patel, D., Kore, V., Gupta, M., & Sharmiladevi, S. (2026). S-edge: A multi-region edge computing framework with adaptive data compression and dynamic load balancing. *IEEE Access*, 14, 81645–81664. <https://doi.org/10.1109/access.2026.3696637>
- Ianculescu, M., Constantin, V.-Ș., Gușatu, A.-M., Petrache, M.-C., Mihăescu, A.-G., Bica, O., & Alexandru, A. (2025). Enhancing connected health ecosystems through IoT-Enabled monitoring technologies: A case study of the Monit4Healthy system. *Sensors*, 25(7), 2292. <https://doi.org/10.3390/s25072292>

- Kavitha, A., R, N. M., Vikkas, P. S. D., R, V., & T, R. (2026). Digital platform for vehicle service booking and monitoring. *2026 International Conference on Communication, Computing and Emerging Technologies (IC3ET)*, 1–10. <https://doi.org/10.1109/ic3et64989.2026.11467376>
- Kim, H., Zhao, Y., Pavlo, A., & Gibbons, P. B. (2025). No cap, this memory slaps: Breaking through the memory wall of transactional database systems with processing-in-memory. *Proceedings of the VLDB Endowment*, 18(11), 4241–4254. <https://doi.org/10.14778/3749646.3749690>
- Ku, D., Zang, H., Yusupov, A., Park, S., & Kim, J. (2025). Vehicle-to-everything-car edge cloud management with development, security, and operations automation framework. *Electronics*, 14(3), 478–478. <https://doi.org/10.3390/electronics14030478>
- Kurisaka, H., Su, Y., Nguyen, P. Le, Nguyen, K., & Sekiya, H. (2025). Performance evaluation of ethereum consensus mechanisms in IoT-blockchain systems using resource-constrained devices. *Cluster Computing*, 28(12). <https://doi.org/10.1007/s10586-025-05503-w>
- Sielskait'ė, Š., & Kalibaitienė, D. (2025). The impact of human factors on software development processes applying the agile and waterfall methodologies: A case study using real data. *International Journal of Computers Communications & Control*, 20(4). <https://doi.org/10.15837/ijccc.2025.4.6807>
- Veerapaneni, R. K., Delhibabu, R., Subbotin, A., & Zhukova, N. (2026). Development of a high-performance in-memory database architecture for intelligent video surveillance in critical patient care. *Frontiers in Digital Health*, 8. <https://doi.org/10.3389/fdgth.2026.1807507>
- Zeydan, E., Arslan, S., & Turk, Y. (2026). A cloud native journey for telecommunication networks: Components, applications and open challenges. *ACM Computing Surveys*, 58(10), 1–38. <https://doi.org/10.1145/3801492>

