

BENGKELGO - SERVIS KENDARAAN ON DEMAND

disusun untuk memenuhi tugas
mata kuliah pemrograman berbasis mobile

oleh:

M Agradika Ridhal Eljatin (2208107010020)

Muhammad Khalid Al Ghifari (2208107010044)



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2025**

1. LATAR BELAKANG

Kendaraan baik roda dua maupun roda empat, telah menjadi bagian penting dalam kehidupan sehari-hari masyarakat. Aktivitas seperti bekerja, bersekolah, maupun bepergian ke tempat umum sangat bergantung pada ketersediaan dan kondisi kendaraan yang prima. Namun, kenyataannya banyak pemilik kendaraan yang kesulitan meluangkan waktu untuk melakukan servis rutin ke bengkel, apalagi ketika mengalami gangguan mendadak di jalan seperti mogok atau ban bocor.

Di sisi lain, banyak teknisi dan tukang bengkel yang memiliki keterampilan namun belum memiliki sarana untuk menjangkau pelanggan secara luas dan fleksibel. Hal ini menciptakan kesenjangan antara kebutuhan servis yang tinggi dan ketersediaan tenaga teknisi yang belum tersalurkan secara optimal.

Dengan semakin berkembangnya teknologi digital dan meningkatnya penggunaan perangkat mobile, hadir peluang untuk menghadirkan solusi berbasis aplikasi yang mampu menjembatani kebutuhan tersebut. Aplikasi ini dirancang untuk menghubungkan pelanggan dengan teknisi kendaraan secara langsung, di mana teknisi dapat dipanggil ke lokasi pelanggan kapan saja sesuai kebutuhan.

Melalui konsep layanan servis kendaraan on-demand, aplikasi ini diharapkan dapat memberikan kemudahan bagi pengguna dalam merawat kendaraannya, sekaligus membuka peluang kerja yang lebih luas dan fleksibel bagi para teknisi.

2. TUJUAN

- Memberikan kemudahan bagi pengguna dalam mendapatkan layanan servis kendaraan.
- Menyediakan platform bagi teknisi untuk menawarkan jasa secara fleksibel.
- Meningkatkan efisiensi waktu dan akses terhadap layanan perbaikan ringan.
- Mendukung pengembangan ekonomi digital di sektor otomotif.

3. FITUR UTAMA APLIKASI

- Servis Kendaraan di Lokasi: Teknisi akan datang langsung ke lokasi pengguna untuk melakukan berbagai jenis perawatan ringan seperti ganti oli, pengecekan aki, tune-up, atau tambal ban, sehingga pengguna tidak perlu repot ke bengkel.
- Emergency Call 24/7: Memberikan layanan darurat kapan saja saat kendaraan mengalami kendala mendadak seperti mogok atau ban bocor di jalan.
- Spare Part Shop: Menyediakan berbagai suku cadang resmi yang dapat dibeli langsung dari aplikasi dan digunakan oleh teknisi saat melakukan servis.
- View Nearest: Menampilkan teknisi terdekat dari lokasi pengguna secara otomatis, sehingga layanan bisa diberikan dengan lebih cepat dan efisien.

4. RANCANGAN APLIKASI

4.1. Arsitektur Sistem

Aplikasi BengkelGo dibangun sebagai aplikasi mobile native Android menggunakan bahasa pemrograman Kotlin. Untuk backend, aplikasi ini memanfaatkan layanan Firebase, termasuk:

- Firebase Authentication: Untuk menangani registrasi, login, dan manajemen sesi pengguna.
- Cloud Firestore: Sebagai database NoSQL untuk menyimpan data transaksi servis, data produk suku cadang, keranjang belanja pengguna, dan pesanan suku cadang.

Secara umum, alur kerja aplikasi melibatkan interaksi pengguna melalui antarmuka di perangkat Android, yang kemudian berkomunikasi dengan Firebase untuk operasi data dan autentikasi.

4.2. Desain Antarmuka Pengguna (UI/UX)

Desain antarmuka pengguna BengkelGo dirancang agar bersih, modern, dan mudah dinavigasi. Perancangan awal desain dibuat melalui perangkat lunak Figma, penggunaan Material Design Components memastikan konsistensi visual dan interaksi yang familiar bagi pengguna Android. Alur pemesanan servis dan pembelian suku cadang dibuat sesederhana mungkin untuk mengurangi friksi pengguna.

4.3. Struktur Database (Cloud Firestore)

Berikut adalah rancangan struktur koleksi utama dalam Cloud Firestore:

- serviceOrders: Menyimpan detail setiap pesanan servis kendaraan.
 - Dokumen: {orderId}
 - Fields: `userId`, `userName`, `vehicleType`, `serviceType`, `selectedLocation`, `fullAddress`, `serviceSchedule`, `paymentMethod`, `orderTimestamp`, `status` (sesuai `VehicleServiceOrder.kt`).
- spareParts: Menyimpan katalog produk suku cadang.
 - Dokumen: {partId} (bisa Auto-ID)
 - Fields: `name`, `brand`, `category`, `description`, `imageName`, `imageUrl`, `price`, `stock`, `features` (array), `specifications` (map) (sesuai `SparePart.kt`).
- userCarts: Menyimpan item keranjang belanja untuk setiap pengguna.
 - Dokumen: {userId}
 - Sub-koleksi: `items`
 - Dokumen: {partId}
 - Fields: `partId`, `name`, `price`, `quantity`, `imageName` (sesuai `CartItem.kt`).
- sparePartOrders: Menyimpan detail setiap pesanan suku cadang.
 - Dokumen: {orderId} (bisa Auto-ID)
 - Fields: `userId`, `userName`, `items` (list of `CartItem`), `totalAmount`, `shippingAddress`, `paymentMethod`, `orderTimestamp`, `status` (sesuai `SparePartOrder.kt`).

5. TEKNOLOGI YANG DIGUNAKAN

- Bahasa Pemrograman: Kotlin
- Platform Pengembangan: Android Studio
- Minimum SDK: API 24 (Android 7.0 Nougat)
- Target SDK: API 35 (Android 15 Upside Down Cake)
- Backend as a Service (BaaS):
 - Firebase Authentication (Email/Password)

- Cloud Firestore (Database NoSQL)
- Library UI: Material Design Components, AndroidX libraries.
- Build System: Gradle

6. IMPLEMENTASI FITUR

Bagian ini menjelaskan implementasi detail dari fitur-fitur utama aplikasi BengkelGo, merujuk pada komponen-komponen kode yang relevan dalam proyek.

6.1. Manajemen Pengguna (Autentikasi)

Fitur manajemen pengguna ditangani menggunakan Firebase Authentication dan mencakup beberapa alur kerja yang terdistribusi di beberapa Activity:

- **Registrasi Pengguna Baru:**
 - Dimulai pada `RegisterActivity.kt` (). Pengguna memasukkan nama, email, nomor telepon, alamat, dan password melalui form yang didefinisikan dalam `activity_register.xml`.
 - Proses pendaftaran menggunakan metode `createUserWithEmailAndPassword` dari Firebase Authentication.
 - Setelah berhasil mendaftar, nama pengguna disimpan dalam profil Firebase menggunakan `UserProfileChangeRequest.Builder`.
 - Pengguna kemudian diarahkan ke `CongratulationsActivity.kt` () yang menampilkan pesan sukses registrasi (berdasarkan intent extra "TYPE" bernilai "REGISTER").
- **Login Pengguna:**
 - Dikelola oleh `LoginActivity.kt` (), dengan layout dari `activity_login.xml`.
 - Pengguna memasukkan email dan password. Kredensial divalidasi menggunakan `signInWithEmailAndPassword` dari Firebase Authentication.
 - Jika berhasil, pengguna diarahkan ke `MainActivity.kt`.
 - Terdapat pengecekan status login pada `onStart()` di `LoginActivity.kt` dan `SplashActivity.kt` untuk otomatis mengarahkan pengguna yang sudah login ke `MainActivity.kt`.
- **Reset Password:**

- Pengguna dapat mereset password melalui `ResetPasswordActivity.kt` (), diakses dari `LoginActivity.kt`.
- Pengguna memasukkan alamat email terdaftar. Firebase Authentication mengirimkan email reset password menggunakan `sendPasswordResetEmail`.
- Setelah permintaan berhasil, pengguna diarahkan ke `CongratulationsActivity.kt` dengan pesan sukses reset password (intent extra "TYPE" bernilai "RESET_PASSWORD").
- **Profil Pengguna dan Logout:**
 - `ProfileActivity.kt` () menampilkan nama dan email pengguna yang sedang login, diambil dari `FirebaseAuth.currentUser`.
 - Fitur logout memanggil `auth.signOut()` dan mengarahkan pengguna kembali ke `LoginActivity.kt` dengan membersihkan back stack. Proses logout didahului dengan dialog konfirmasi kustom (`dialog_logout_confirmation.xml`).
- **Splash Screen:**
 - `SplashActivity.kt` () ditampilkan saat aplikasi pertama kali dibuka. Setelah delay singkat, ia memeriksa status login pengguna saat ini. Jika sudah login, pengguna diarahkan ke `MainActivity.kt`, jika tidak maka ke `LoginActivity.kt`.

6.2. Pemesanan Servis Kendaraan

Alur pemesanan servis kendaraan melibatkan beberapa tahapan:

- **Inisiasi Pemesanan:**
 - Dari `MainActivity.kt` (), pengguna memilih layanan "Vehicle Repair" yang mengarahkan ke `VehicleRepairFormActivity.kt`.
- **Pengisian Form Servis:**
 - Di `VehicleRepairFormActivity.kt` (), pengguna mengisi detail servis melalui `AutoCompleteTextView` untuk jenis kendaraan, jenis servis, area lokasi, dan metode pembayaran. Data untuk dropdown diambil dari `arrays.xml` di `res/values/strings.xml`.
 - Jenis servis diperbarui secara dinamis berdasarkan pilihan jenis kendaraan.
 - Jadwal servis dipilih menggunakan `DatePickerDialog` dan `TimePickerDialog` yang diaktifkan melalui `TextInputEditText` (`edtServiceSchedule`).
- **Ringkasan dan Konfirmasi Pesanan:**

- Setelah mengisi form, pengguna menekan tombol "Next" dan data dikirim via Intent ke `OrderSummaryActivity.kt`.
- Di `OrderSummaryActivity.kt` (), ringkasan pesanan ditampilkan, dan pengguna diminta memasukkan alamat lengkap (`edtCompleteAddress`).
- **Penyimpanan ke Firestore:**
 - Dengan menekan "Order Now", objek `VehicleServiceOrder` () yang berisi semua detail pesanan (termasuk `userId`, `userName`, `fullAddress`, dan `orderTimestamp` yang di-generate otomatis oleh server) disimpan ke koleksi `serviceOrders` di Cloud Firestore.
- **Konfirmasi Akhir:**
 - Setelah berhasil menyimpan, pengguna diarahkan ke `OrderConfirmationActivity.kt` () yang menampilkan pesan bahwa pesanan telah diterima. Tombol "Back to Home" akan mengarahkan kembali ke `MainActivity.kt` dan membersihkan tumpukan Activity sebelumnya.

6.3. Panggilan Darurat

- Di `MainActivity.kt` (), terdapat CardView "Emergency Call".
- Ketika dipilih, method `showEmergencyCallDialog()` akan dipanggil. Method ini meng-inflate layout `dialog_emergency_call.xml` () yang menampilkan deskripsi layanan darurat dan nomor telepon dummy BengkelGo (dari `strings.xml`).
- Tombol "Hubungi Sekarang" pada dialog akan memicu `Intent.ACTION_DIAL` untuk membuka aplikasi telepon dengan nomor yang sudah terisi. Izin untuk `ACTION_DIAL` dideklarasikan dalam `AndroidManifest.xml`.

6.4. Toko Suku Cadang (Spare Part Shop)

Fitur ini memungkinkan pengguna untuk membeli suku cadang melalui aplikasi:

- **Akses Toko dan Daftar Produk:**
 - Pengguna mengakses toko dari `MainActivity.kt` (CardView "Spare Part Shop") yang mengarah ke `SparePartListActivity.kt`.
 - `SparePartListActivity.kt` mengambil data produk dari koleksi `spareParts` di Firestore. Data produk sesuai dengan model `SparePart.kt`.
 - Produk ditampilkan menggunakan `RecyclerView` dengan `SparePartAdapter.kt`. Layout untuk setiap item adalah `item_spare_part.xml`.

- Terdapat fitur pencarian (`etSearchRims`) dan filter berdasarkan kategori (misalnya, "Rims", "Tires") melalui tombol-tombol kategori.
- Ikon keranjang di toolbar mengarahkan ke `ShoppingCartActivity.kt`.
- **Detail Produk:**
 - Memilih "View Details" pada item di `SparePartListActivity.kt` akan membuka `SparePartDetailActivity.kt()`, dengan melewati `PART_ID` dan `PART_NAME` melalui Intent.
 - Activity ini mengambil detail lengkap produk dari Firestore berdasarkan `partId`.
 - Informasi seperti nama, harga, deskripsi, stok, gambar (dimuat dari drawable berdasarkan `imageName` di `SparePart.kt`), fitur, dan spesifikasi ditampilkan. Layoutnya adalah `activity_spare_part_detail.xml`.
 - `TabLayout` digunakan untuk memisahkan bagian "Overview", "Features", dan "Specifications".
- **Manajemen Keranjang Belanja:**
 - Di `SparePartDetailActivity.kt`, tombol "Add To Cart" memungkinkan pengguna menambahkan produk ke keranjang.
 - Item disimpan sebagai objek `CartItem ()` dalam sub-koleksi `items` di bawah dokumen `userCarts/{userId}` di Firestore. Proses ini menggunakan Firestore Transaction untuk menangani penambahan kuantitas jika item sudah ada atau membuat item baru, serta memeriksa ketersediaan stok.
 - `ShoppingCartActivity.kt ()` menampilkan item dalam keranjang pengguna secara real-time (menggunakan `ListenerRegistration`).
 - `CartAdapter.kt ()` digunakan dengan `RecyclerView` untuk menampilkan item keranjang (layout `item_cart.xml`). Pengguna dapat mengubah kuantitas item atau menghapusnya. Perubahan ini akan diperbarui di Firestore.
 - Total item dan total harga ditampilkan dan diperbarui secara dinamis.
- **Checkout Suku Cadang:**
 - Dari `ShoppingCartActivity.kt`, tombol "Proceed to Checkout" mengarahkan ke `SparePartCheckoutActivity.kt`.
 - `SparePartCheckoutActivity.kt ()` memuat item keranjang saat ini untuk ditampilkan sebagai ringkasan pesanan. Pengguna memasukkan alamat pengiriman (`edtCheckoutAddress`).

- Ketika "Place Order" ditekan, objek `SparePartOrder ()` dibuat.
- Firestore Batch Write digunakan untuk:
 1. Menyimpan pesanan baru ke koleksi `sparePartOrders`.
 2. Mengurangi stok setiap item yang dipesan dari koleksi `spareParts` menggunakan `FieldValue.increment(-quantity)`.
 3. Menghapus semua item dari keranjang pengguna (`userCarts/{userId}/items`).
- Setelah berhasil, pengguna diarahkan ke [OrderConfirmationActivity.kt](#).

6.5. Lokasi Servis Terdekat (Nearest Service)

Fitur ini memungkinkan pengguna untuk melihat lokasi servis terdekat melalui map.

7. PENGUJIAN APLIKASI

Pengujian aplikasi BengkelGo dilakukan melalui beberapa metode untuk memastikan kualitas dan fungsionalitas:

- **Unit Testing:**
 - Pengujian dilakukan pada logika bisnis individual di dalam class Kotlin, seperti fungsi validasi input, kalkulasi, atau manipulasi data sederhana.
 - File `ExampleUnitTest.kt ()` menyediakan kerangka awal untuk unit test.
- **Instrumented Testing:**
 - Pengujian yang berjalan di perangkat Android atau emulator untuk memverifikasi interaksi UI dan integrasi antar komponen Android (Activity, Service, Content Provider).
 - File `ExampleInstrumentedTest.kt ()` berisi contoh awal pengujian terinstrumentasi, seperti memverifikasi `packageName` aplikasi.
- **Pengujian Manual Fungsional:**
 - Pengujian fungsionalitas secara manual dilakukan pada berbagai skenario penggunaan di emulator Android Studio dan perangkat fisik.
 - Alur kerja utama yang diuji meliputi:

- Registrasi pengguna baru dan validasi input.
- Login dengan kredensial valid dan invalid.
- Proses reset password.
- Logout dan dampaknya pada akses fitur.
- Alur pemesanan servis kendaraan dari awal hingga konfirmasi, termasuk validasi input form dan pemilihan jadwal.
- Penggunaan fitur panggilan darurat dan verifikasi intent dial.
- Penjelajahan toko suku cadang, penggunaan filter kategori, dan pencarian produk.
- Melihat detail produk suku cadang.
- Penambahan item ke keranjang belanja, perubahan kuantitas, dan penghapusan item.
- Proses checkout suku cadang, termasuk pengisian alamat dan konfirmasi pesanan.
- Verifikasi penyimpanan data ke Firebase Firestore (serviceOrders, sparePartOrders, userCarts) melalui Firebase Console.
- Responsivitas UI pada berbagai ukuran layar (simulasi).
- Penanganan error dan pesan feedback kepada pengguna (misalnya, saat input tidak valid, stok habis, atau terjadi kesalahan jaringan).

8. KESIMPULAN DAN SARAN

8.1 Kesimpulan

Aplikasi BengkelGo berhasil dikembangkan sebagai platform mobile untuk layanan servis kendaraan on-demand dan pembelian suku cadang. Fitur-fitur utama seperti manajemen pengguna (registrasi, login, reset password, profil), pemesanan servis kendaraan terjadwal, panggilan darurat, dan toko suku cadang lengkap dengan keranjang belanja serta proses checkout telah berhasil diimplementasikan menggunakan Kotlin sebagai bahasa pemrograman utama dan Firebase (Authentication dan Cloud Firestore) sebagai backend. Aplikasi ini menunjukkan potensi besar dalam memberikan kemudahan akses layanan otomotif bagi pengguna kendaraan dan membuka peluang baru bagi teknisi.

8.2 Saran Pengembangan Selanjutnya:

Untuk pengembangan aplikasi BengkelGo di masa mendatang, beberapa saran berikut dapat dipertimbangkan:

- Implementasi fitur "View Nearest Technician" menggunakan layanan lokasi perangkat dan integrasi Google Maps API untuk menampilkan teknisi terdekat secara real-time.
- Pengembangan aplikasi sisi teknisi yang terpisah, mencakup fitur pendaftaran teknisi, pengaturan profil dan keahlian, manajemen jadwal ketersediaan, serta penerimaan dan pengelolaan order servis atau panggilan darurat.
- Integrasi sistem pembayaran online melalui payment gateway untuk memudahkan transaksi non-tunai baik untuk servis maupun pembelian suku cadang.
- Implementasi fitur rating dan ulasan (review) bagi teknisi setelah menyelesaikan layanan servis, dan juga untuk produk suku cadang yang dibeli.
- Penambahan fitur estimasi biaya servis sebelum pengguna melakukan pemesanan akhir.
- Pengembangan fitur pelacakan (tracking) posisi teknisi secara live saat sedang dalam perjalanan menuju lokasi pelanggan.
- Implementasi notifikasi push untuk memberikan update status pesanan (servis dan suku cadang), promosi, atau pengingat servis berkala kepada pengguna.
- Peningkatan desain UI/UX secara berkelanjutan berdasarkan feedback dari pengguna untuk meningkatkan kenyamanan dan kemudahan penggunaan.
- Optimalisasi performa aplikasi, terutama pada bagian pemuatan data dari Firestore dan manajemen memori untuk pengalaman pengguna yang lebih lancar.
- Peningkatan keamanan aplikasi, termasuk validasi input yang lebih ketat dan pengamanan akses data pengguna.