



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN SISTEM PARKIR PADA GEDUNG
PERKANTORAN DENGAN METODE PENGENALAN PLAT NOMOR
BERBASIS ANDROID**

**“Perancangan Aplikasi Android Sistem Parkir Gedung Perkantoran dengan
Pengenalan Plat Nomor Kendaraan”**

TUGAS AKHIR

**POLITEKNIK
NEGERI
JAKARTA**

Lukman Alghifarri

2203332075

PROGRAM STUDI TEKNIK TELEKOMUNIKASI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2025



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Lukman Alghifarri

NIM : 2203332075

Tanda Tangan :

Tanggal : 24 Juli 2025

**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama : Lukman Alghifarri

NIM : 2203332075

Program Studi : Teknik Telekomunikasi

Judul Tugas Akhir : Rancang Bangun Sistem Parkir Pada Gedung Perkantoran
Dengan Metode Pengenalan Plat Nomor Berbasis Android

Sub Judul : Perancangan Aplikasi Android Sistem Parkir Gedung
Perkantoran dengan Pengenalan Plat Nomor Kendaraan

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada dan dinyatakan LULUS/TIDAK LULUS.

Pembimbing

: Rifqi Fuadi Hasani, S.T., M.T.
NIP. 199208182019031015

()

Depok, 23 Juli 2025

**POLITEKNIK
NEGERI
JAKARTA**

Disahkan oleh
Ketua Jurusan Teknik Elektro




Dr. Murie Dwiyaniti, S.T., M.T.
NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Rifqi Fuadi Hasani, S.T., M.T. selaku dosen pembimbing yang telah membimbing, mengarahkan, dan memberikan masukan berharga selama proses penyusunan tugas akhir ini.
2. Seluruh keluarga tercinta atas doa, kasih sayang, dan dukungan yang tiada henti, baik secara moral maupun material.
3. Rahma, selaku donatur dan pendukung selama masa perkuliahan hingga penyusunan tugas akhir ini.

Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca serta menjadi referensi yang berguna bagi pengembangan sistem serupa di kemudian hari.

Depok, 24 Juli 2025

Penulis

Lukman Alghifarri



RANCANG BANGUN SISTEM PARKIR PADA GEDUNG PERKANTORAN DENGAN METODE PENGENALAN PLAT NOMOR BERBASIS ANDROID

Perancangan Aplikasi Android Sistem Parkir Gedung Perkantoran dengan
Pengenalan Plat Nomor Kendaraan

Abstrak

Sistem parkir konvensional pada gedung perkantoran seringkali menimbulkan antrean panjang, kesalahan pencatatan pelat nomor, serta minimnya efisiensi dan keamanan. Untuk mengatasi permasalahan tersebut, dikembangkan sebuah aplikasi Android sistem parkir otomatis berbasis pengenalan plat nomor kendaraan (ALPR) yang terintegrasi dengan perangkat IoT dan layanan Firebase. Penelitian ini memfokuskan pada pengembangan fitur utama aplikasi seperti autentikasi pengguna, pengelolaan data kendaraan, manajemen langganan parkir, riwayat aktivitas parkir, serta tampilan status kendaraan secara real-time. Metode yang digunakan melibatkan integrasi Firebase Authentication, Cloud Firestore, dan Realtime Database untuk komunikasi data, serta pemrograman menggunakan Dart dan Flutter. Sistem juga mendukung perangkat keras Raspberry Pi, ESP32, dan komunikasi LoRa untuk otomatisasi kontrol palang. Pengujian dilakukan terhadap fungsionalitas aplikasi, sistem keamanan, Quality of Service (QoS), dan kualitas jaringan seluler menggunakan G-NetTrack. Hasil pengujian menunjukkan seluruh fitur aplikasi berjalan sesuai rancangan, dengan tingkat keberhasilan 100% dalam lima skenario utama, serta sistem autentikasi berhasil membatasi akses hanya untuk pengguna terverifikasi. QoS menunjukkan nilai throughput maksimal 368,4 Kbps, delay rata-rata 22,88 ms, dan packet loss 0%. Sementara itu, kualitas jaringan dinyatakan stabil dengan rata-rata RSRP -93 dBm, RSRQ -7,7 dB, SNR 13 dB, dan RSSI -64,3 dBm. Dengan capaian tersebut, sistem ini terbukti mampu menghadirkan solusi parkir otomatis yang efisien, aman, dan berbasis digital secara real-time.

Kata kunci: ALPR, Firebase, Flutter, G-NetTrack, IoT, QoS, Sistem parkir otomatis.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DESIGN OF A PARKING SYSTEM FOR OFFICE BUILDINGS USING AN ANDROID-BASED LICENSE PLATE RECOGNITION METHOD

*Design of an Android Application for Office Building Parking System with Vehicle
License Plate Recognition*

Abstract

Conventional parking systems in office buildings often cause long queues, license plate recording errors, and lack efficiency and security. To address these issues, an Android-based automated parking system application was developed using Automatic License Plate Recognition (ALPR), integrated with IoT devices and Firebase services. This study focuses on the development of key application features such as user authentication, vehicle data management, parking subscription management, parking activity history, and real-time vehicle status display. The methodology involves the integration of Firebase Authentication, Cloud Firestore, and Realtime Database for data communication, along with programming using Dart and Flutter. The system also supports hardware integration with Raspberry Pi, ESP32, and LoRa communication for automated gate control. Testing was conducted on application functionality, system security, Quality of Service (QoS), and mobile network quality using G-NetTrack. Results show that all features operated as designed, with a 100% success rate across five main scenarios, and the authentication system effectively restricted access to verified users only. QoS testing indicated a maximum throughput of 368.4 Kbps, an average delay of 22.88 ms, and 0% packet loss. Meanwhile, network quality remained stable with an average RSRP of -93 dBm, RSRQ of -7.7 dB, SNR of 13 dB, and RSSI of -64.3 dBm. These results demonstrate that the system effectively delivers a secure, efficient, and real-time digital parking solution.

Keywords: Automatic parking system, ALPR, Firebase, Flutter, IoT, QoS, G-NetTrack.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN TUGAS AKHIR.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Luaran.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Sistem Parkir	3
2.2 ALPR.....	3
2.3 <i>Internet of Things</i>	5
2.4 <i>Android Studio</i>	6
2.5 <i>Dart</i>	7
2.6 <i>Flutter</i>	9
2.7 <i>Riverpod</i>	10
2.8 <i>Firebase</i>	11
2.9 HTTP	12
2.10 QOS	13
2.11 <i>GnetTrack</i>	15
BAB III PERENCANAAN DAN REALISASI.....	17
3.1 Rancangan Aplikasi	17
3.1.1 Deskripsi Aplikasi	17
3.1.2 Cara Kerja Aplikasi.....	18
3.1.3 Spesifikasi Aplikasi.....	19
3.1.4 Diagram Blok.....	20
3.1.5 Perancangan <i>Authentication</i>	21
3.1.6 Perancangan <i>Firestore</i>	21
3.1.7 Perancangan <i>Realtime Database</i>	22
3.1.8 Perancangan Aplikasi <i>Android</i>	23
3.2 Realisasi Aplikasi.....	24
3.2.1 Realisasi <i>Database</i>	24
3.2.2 Realisasi Aplikasi <i>Android</i>	27
BAB IV PEMBAHASAN.....	57
4.1 Pengujian Aplikasi <i>Android</i>	57
4.1.1 Deskripsi Pengujian Aplikasi <i>Android</i>	57
4.1.2 Prosedur Pengujian Aplikasi <i>Android</i>	57
4.1.3 Data Hasil Pengujian Aplikasi <i>Android</i>	58
4.1.4 Analisa Hasil Data Pengujian Aplikasi <i>Android</i>	64
4.2 Pengujian <i>Quality of Service</i>	65
4.2.1 Deskripsi Pengujian <i>Quality of Service</i>	65

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.2.2 Prosedur Pengujian <i>Quality of Service</i>	65
4.2.3 Data Hasil Pengujian <i>Quality of Service</i>	66
4.2.4 Analisa Hasil Data <i>Quality of Service</i>	69
BAB V PENUTUP	76
5.1 Kesimpulan	76
5.2 Saran	77
DAFTAR PUSTAKA	78
DAFTAR RIWAYAT HIDUP	80
LAMPIRAN	81





DAFTAR GAMBAR

Gambar 2.1	Logo <i>Android Studio</i>	7
Gambar 3.1	Ilustrasi Sistem Parkir Otomatis	18
Gambar 3.2	<i>Flowchart</i> Aplikasi <i>Park Here</i>	19
Gambar 3.3	Diagram Blok Sistem Parkir Berbasis ALPR	21
Gambar 3.4	<i>Flowchart</i> Perancangan Aplikasi <i>Android</i>	23
Gambar 3.5	Halaman <i>Dashboard Firebase</i>	24
Gambar 3.6	Halaman <i>Authentication Firebase</i>	25
Gambar 3.7	Struktur Koleksi <i>Firestore</i>	26
Gambar 3.8	Struktur <i>Realtime Database</i>	27
Gambar 3.9	Tampilan <i>Splash Screen</i>	29
Gambar 3.10	Tampilan <i>Onboarding Screen</i>	30
Gambar 3.11	Tampilan <i>Login Screen</i>	32
Gambar 3.12	Tampilan <i>Registration Screen</i>	34
Gambar 3.13	Tampilan <i>Dashboard Screen</i>	36
Gambar 3.14	Tampilan <i>History Screen</i>	38
Gambar 3.15	Tampilan <i>Subscription Screen</i>	40
Gambar 3.16	Tampilan <i>Payment Confirmation Screen</i>	44
Gambar 3.17	Tampilan <i>Profile Screen</i>	45
Gambar 3.18	Tampilan <i>Edit Vehicles Screen</i> , <i>Help Screen</i> , dan <i>About Screen</i>	47
Gambar 3.19	Tampilan <i>Notification Screen</i>	50
Gambar 3.20	Tampilan <i>Parking Spot Screen</i>	52
Gambar 3.21	Tampilan <i>Admin Dashboard Screen</i>	54
Gambar 4.1	Tampilan Pengujian <i>Login Aplikasi</i>	59
Gambar 4.2	Tampilan Pengujian Penambahan Kendaraan	60
Gambar 4.3	Tampilan Pengujian Status Langganan	62
Gambar 4.4	Tampilan Pengujian Riwayat Aktivitas Parkir	63
Gambar 4.5	Tangkapan Layar Hasil Pengujian Jaringan Menggunakan <i>G-NetTrack Lite</i>	75

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



DAFTAR TABEL

Tabel 2.1 Kategori Kualitas <i>Throughput</i>	14
Tabel 2.2 Kategori Kualitas <i>Packet Loss</i>	14
Tabel 2.3 Kategori Kualitas <i>Delay</i>	15
Tabel 3.1 Spesifikasi Aplikasi	20
Tabel 3.2 Spesifikasi <i>Smartphone</i>	20
Tabel 4.1 Data Hasil Pengujian Halaman <i>Login</i> Aplikasi.....	59
Tabel 4.2 Data Hasil Pengujian Penambahan Kendaraan	61
Tabel 4.3 Data Hasil Pengujian Penambahan Kendaraan	62
Tabel 4.4 Data Hasil Pengujian Riwayat Aktivitas Parkir	63
Tabel 4.5 Parameter Hasil Perekaman <i>Wireshark</i> (Hari Pertama)	68
Tabel 4.6 Parameter Hasil Perekaman <i>Wireshark</i> (Hari Kedua)	69
Tabel 4.7 Parameter Hasil Perekaman <i>Wireshark</i> (Hari Ketiga).....	70
Tabel 4.8 Parameter Hasil Perekaman <i>GnetTrack Lite</i>	74

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

L-1 Tampilan Aplikasi.....	82
L-2 <i>Source Code</i>	85





BAB I PENDAHULUAN

1.1 Latar Belakang

Sistem parkir konvensional di banyak gedung perkantoran masih bergantung pada pencatatan manual oleh petugas. Pelat nomor kendaraan dicatat secara langsung untuk mengatur akses keluar masuk, yang menimbulkan berbagai kendala seperti potensi kesalahan pencatatan, antrean panjang, dan celah keamanan. Selain itu, sistem manual memerlukan kehadiran petugas secara terus-menerus yang kurang efisien serta rentan terhadap kelelahan dan *human error*.

Di era digital saat ini, perkembangan teknologi informasi membuka peluang untuk merancang sistem parkir yang lebih efisien dan aman melalui otomasi. Salah satu solusinya adalah penerapan aplikasi berbasis *Android* yang terintegrasi dengan teknologi pengenalan pelat nomor kendaraan secara otomatis menggunakan kamera dan metode *Optical Character Recognition* (OCR). Aplikasi ini dapat membantu membaca dan mengidentifikasi pelat nomor kendaraan secara *real-time*, mencocokkannya dengan basis data, serta mengatur akses kendaraan secara otomatis melalui antarmuka yang dapat dipantau dan dikendalikan dari perangkat *Android*.

Untuk mendukung proses komunikasi data dan kontrol sistem parkir secara nirkabel, aplikasi ini diintegrasikan dengan perangkat IoT seperti Raspberry Pi dan ESP32, yang saling terhubung menggunakan komunikasi *Long Range* (LoRa) dan menggunakan *firebase* untuk penyimpanan data. Aplikasi *Android* akan berfungsi sebagai pusat pemantauan dan pengendalian sistem parkir, memberikan tampilan *real-time* status kendaraan serta log aktivitas keluar masuk.

Berdasarkan penjelasan di atas, maka dalam tugas akhir ini dirancang dan dibangun aplikasi *Android* untuk Sistem Parkir Otomatis pada Gedung Perkantoran dengan Pengenalan Pelat Nomor, yang bertujuan untuk meningkatkan efisiensi dan keamanan sistem parkir melalui teknologi perangkat lunak dan integrasi dengan perangkat IoT.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, adapun beberapa masalah yang perlu dirumuskan adalah:

1. Bagaimana merancang aplikasi *Android* untuk mendukung sistem parkir otomatis berbasis pengenalan pelat nomor kendaraan?
2. Bagaimana merealisasikan komunikasi data antara aplikasi *Android* dan perangkat IoT (Raspberry Pi & ESP32) melalui internet dengan *Firebase*?
3. Bagaimana aplikasi dapat menampilkan informasi status kendaraan, hasil pengenalan pelat nomor, serta pengendalian akses kendaraan secara *real-time*?
4. Bagaimana menguji fungsionalitas, sistem keamanan, *Quality of Service* (QoS) dan kualitas layanan dari aplikasi *Android* untuk sistem parkir otomatis berbasis pengenalan pelat nomor kendaraan?

1.3 Tujuan

Tujuan dari penulisan laporan tugas akhir ini adalah:

1. Merancang dan mengembangkan aplikasi *Android* sistem parkir otomatis berbasis pengenalan pelat nomor kendaraan.
2. Mengimplementasikan komunikasi data antara aplikasi *Android* dan perangkat IoT melalui internet dengan menggunakan layanan *Firebase*.
3. Membangun antarmuka pengguna (*user interface*) pada aplikasi *Android* untuk menampilkan informasi kendaraan serta mendukung pengendalian akses parkir secara otomatis.
4. Menguji fungsionalitas, sistem keamanan, *Quality of Service* (QoS) dan kualitas layanan dari aplikasi *Android* untuk sistem parkir otomatis berbasis pengenalan pelat nomor kendaraan.

1.4 Luaran

Adapun luaran dari tugas akhir ini adalah:

1. Sebuah aplikasi *Android* untuk sistem parkir otomatis berbasis pengenalan pelat nomor kendaraan.
2. Laporan Tugas Akhir.
3. Artikel ilmiah.
4. Poster.

BAB V PENUTUP

5.1 Kesimpulan

Adapun kesimpulan dari perancangan dan hasil pengujian dari alat tugasakhir yang telah dibuat, dapat disimpulkan bahwa :

1. Aplikasi Android sistem parkir otomatis dengan pengenalan pelat nomor kendaraan berhasil dikembangkan, mencakup fitur login pengguna, pendaftaran dan pengelolaan kendaraan, pemilihan langganan parkir, serta pemantauan status dan riwayat kendaraan. Pengujian pada lima skenario interaksi, termasuk login menggunakan akun Google, input data kendaraan, dan akses menu riwayat, menunjukkan semua fungsi dapat dijalankan dengan benar, data kendaraan berhasil disimpan ke Firestore dan ditampilkan ulang dalam waktu kurang dari 2 detik, serta tampilan status langganan muncul sinkron dengan pembaruan data di Firebase, tanpa ditemukan hambatan atau gagal muat antarmuka.
2. Sistem komunikasi data antara aplikasi Android dan perangkat IoT (Raspberry Pi dan ESP32) menggunakan Firebase Realtime Database telah diuji dalam lima sesi terpisah. Pengiriman data plat nomor dari hasil deteksi kamera ke Firebase berlangsung dalam rentang waktu rata-rata 120–160 milidetik. Setelah data masuk ke Firebase, status kendaraan berhasil ditampilkan pada aplikasi, dan update status langganan dari aplikasi juga berhasil dikirim ke ESP32, yang memicu servo untuk membuka palang parkir. Tidak ditemukan kesalahan parsing data, dan sinyal kontrol palang berhasil dikirim ulang dengan delay < 200 ms saat jaringan berada pada RSRP sekitar -93 dBm.
3. Antarmuka pengguna diuji dari sisi waktu respon dan konsistensi tampilan antarfitur. Waktu perpindahan dari halaman utama ke halaman riwayat atau status kendaraan rata-rata berada pada rentang 1,1–1,3 detik. Penggunaan komponen UI seperti tombol “Tambah Kendaraan”, “Pilih Paket”, dan “Lihat Riwayat” berhasil dilakukan tanpa error atau bug tampilan dalam seluruh skenario. Data Firebase yang diakses melalui aplikasi tampil sinkron dengan update aktual di database, setelah status langganan diubah dari aplikasi, halaman status memperbarui tampilan dalam waktu kurang dari 1 detik.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4. Hasil pengujian *Quality of Service* (QoS) menunjukkan throughput tertinggi 368,50 Kbps, delay minimum 22,89 ms, dan packet loss 0%. Sementara itu, pengujian kualitas jaringan menggunakan *G-NetTrack* mencatat RSRP rata-rata -93 dBm, RSRQ -7,7 dB, SNR 13 dB, dan RSSI -65 dBm. Berdasarkan hasil tersebut, sistem dapat berfungsi andal dalam kondisi jaringan seluler yang stabil untuk mendukung komunikasi data *real-time* dan implementasi parkir otomatis berbasis IoT.

5.2 Saran

Berdasarkan hasil perancangan dan pengujian, disarankan pengembangan aplikasi meliputi integrasi fitur pembayaran digital seperti QRIS atau payment gateway untuk mempermudah transaksi, penambahan push notification guna mengingatkan jatuh tempo langganan dan status kendaraan, serta penyajian laporan riwayat parkir dalam bentuk grafik atau statistik untuk memudahkan monitoring. Selain itu, peningkatan keamanan autentikasi seperti verifikasi dua langkah (2FA) atau biometrik juga perlu diterapkan guna melindungi data pengguna.

**POLITEKNIK
NEGERI
JAKARTA**

DAFTAR PUSTAKA

- Budianto, A. (2018). Pengenalan Plat Nomor Otomatis: Tinjauan dan Studi Kasus di Indonesia. *Jurnal Sains dan Informatika*, 5(2), 258–270.
- ETSI. (2007). Harmonisasi Jaringan Telekomunikasi dan Protokol Internet (TIPHON); Metrik Kualitas Layanan (QoS). ETSI TR 101 329-2 V1.1.1. Institut Standar Telekomunikasi Eropa (ETSI).
- Google Developers. (2023). *GnetTrack* Vehicle Tracking Documentation. Diakses dari <https://developers.google.com/gnetrack>
- Hidayat, A. F. (2023). Manajemen state menggunakan Riverpod dalam pengembangan aplikasi Flutter. *Jurnal Teknologi dan Rekayasa Perangkat Lunak*, 5(1), 33–41.
- Hidayat, L., Kurniawan, E., & Ramdhani, M. (2022). Perancangan sistem palang parkir otomatis dan pendeteksi slot parkir berbasis IoT. *E-Proceeding of Engineering*, c9(2), 174–180. ISSN: 2355-9365.
- Maulana, R. (2022). Analisis performa bahasa pemrograman Dart dalam pengembangan aplikasi mobile berbasis Flutter. *Jurnal Teknologi dan Sistem Komputer*, 10(1), 44–51.
- Nutalapati, V. (2019). Performance comparison between Kotlin and Java in *Android* development. *International Research Journal of Engineering and Applied Sciences*, 7(1).
- Purwanto, W., & Septiani, M. (2023). Implementasi automatic license plate recognition untuk mengurangi pelanggaran lalu lintas berbasis artificial intelligence. *Informatics for Educators and Professionals: Journal of Informatics*, 8(2), 148–157.
- Ramdani, A. (2022). Implementasi *Firebase* dalam pengembangan aplikasi mobile menggunakan Flutter. *Jurnal Teknologi Informasi dan Pendidikan*, 15(2), 85–92.
- Riyanto, Y. (2021). Pengenalan Dart dan implementasinya dalam pengembangan aplikasi Flutter. *Jurnal Teknologi Informasi dan Komputer*, 7(2), 134–140.
- Saputra, R. A. (2022). Pengembangan aplikasi mobile multiplatform menggunakan Flutter dan Dart. *Jurnal Teknologi dan Sistem Informasi*, 8(1), 55–62.

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Sukmandhani, A. (2020). Analisis kinerja jaringan dengan parameter QoS pada jaringan komputer. Politeknik Negeri Jakarta.

Senstar. (n.d.). Automated license plate readers (ALPRs). Diakses dari <https://senstar.com/senstarpedia/automated-license-plate-readers-alprs/>

TIPHON. (2007). Kualitas Layanan (QoS); Bagian 2: Definisi parameter Kualitas Layanan dan cara perhitungannya (ETSI TR 101 329-2 V1.1.1). Institut Standar Telekomunikasi Eropa (ETSI).



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP

Lukman Alghifarri

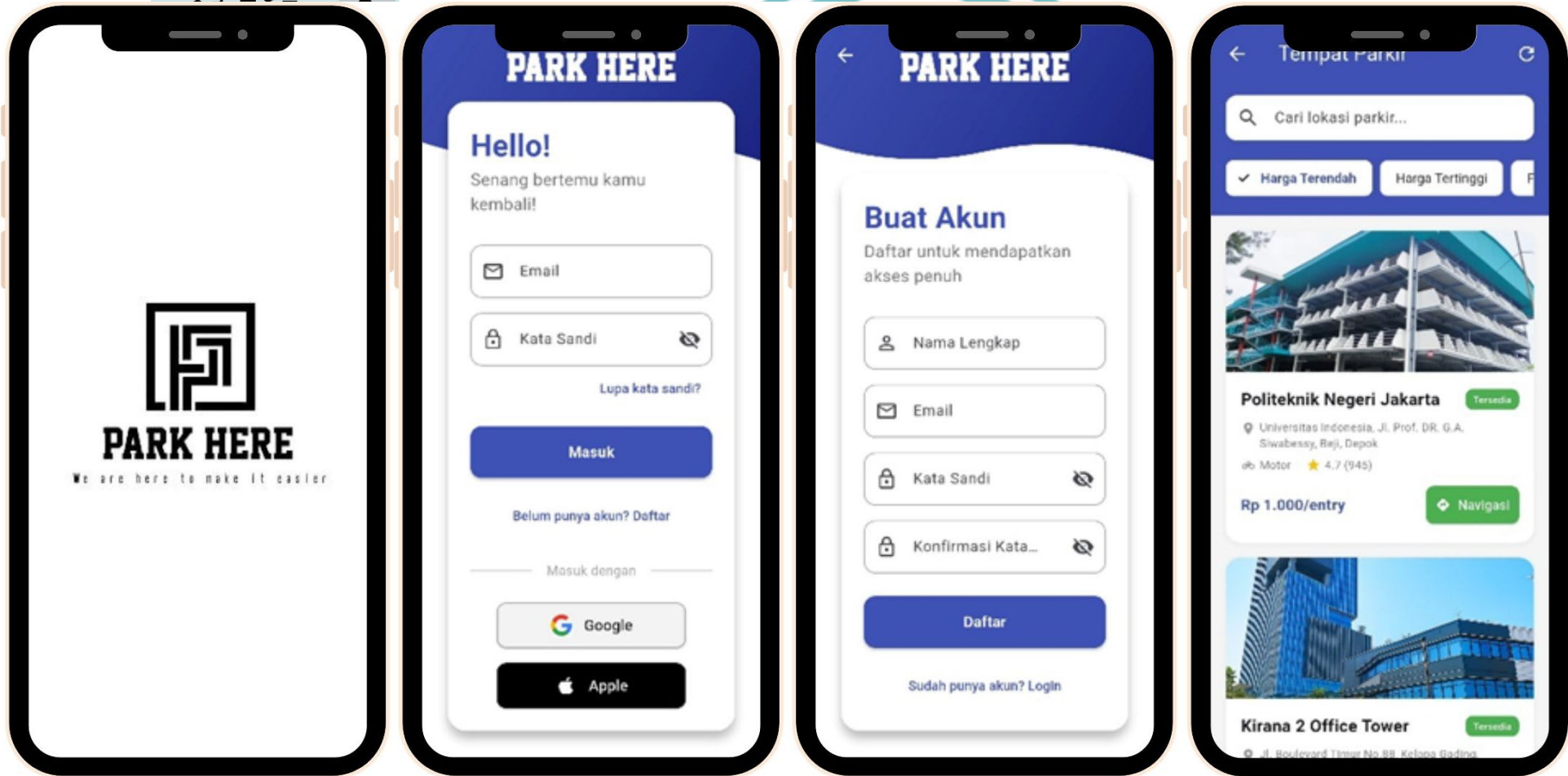


Lahir di Jakarta, 07 Agustus 2002. Lulus dari SDN 06 Sungai Bambu tahun 2014, SMPN 65 Jakarta Utara tahun 2017, dan SMKN 1 Jakarta tahun 2020. Gelar Diploma Tiga (D3) diperoleh pada tahun 2025 dari Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi, Politeknik Negeri Jakarta



LAMPIRAN

L-1 Tampilan Aplikasi



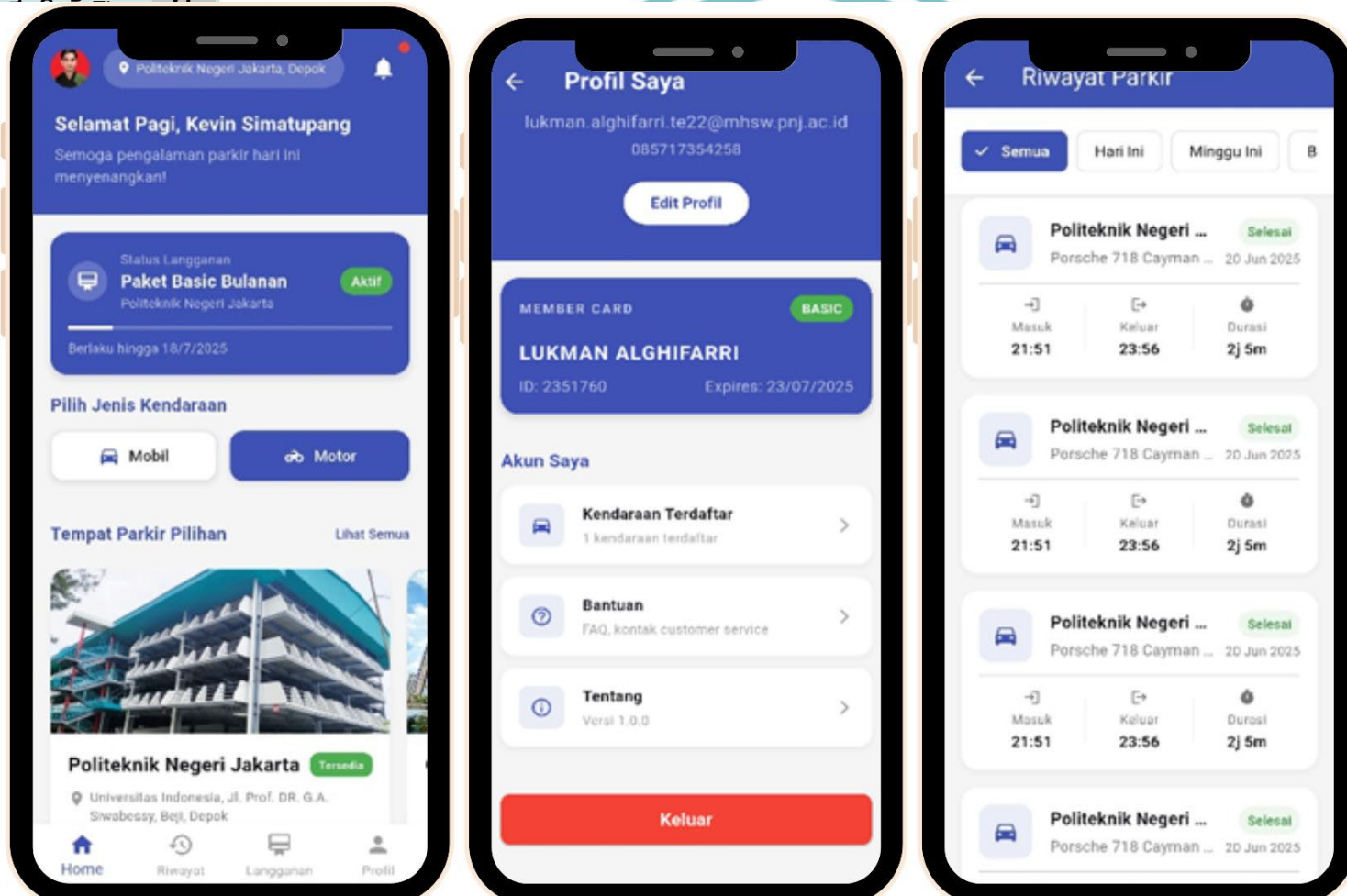
01

TAMPILAN APLIKASI ANDROID 1



PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	: Lukman Alghifarri
Diperiksa	: Rifqi Fuadi Hasani, S.T., M.T.
Tanggal	: 18 Juni 2025



02

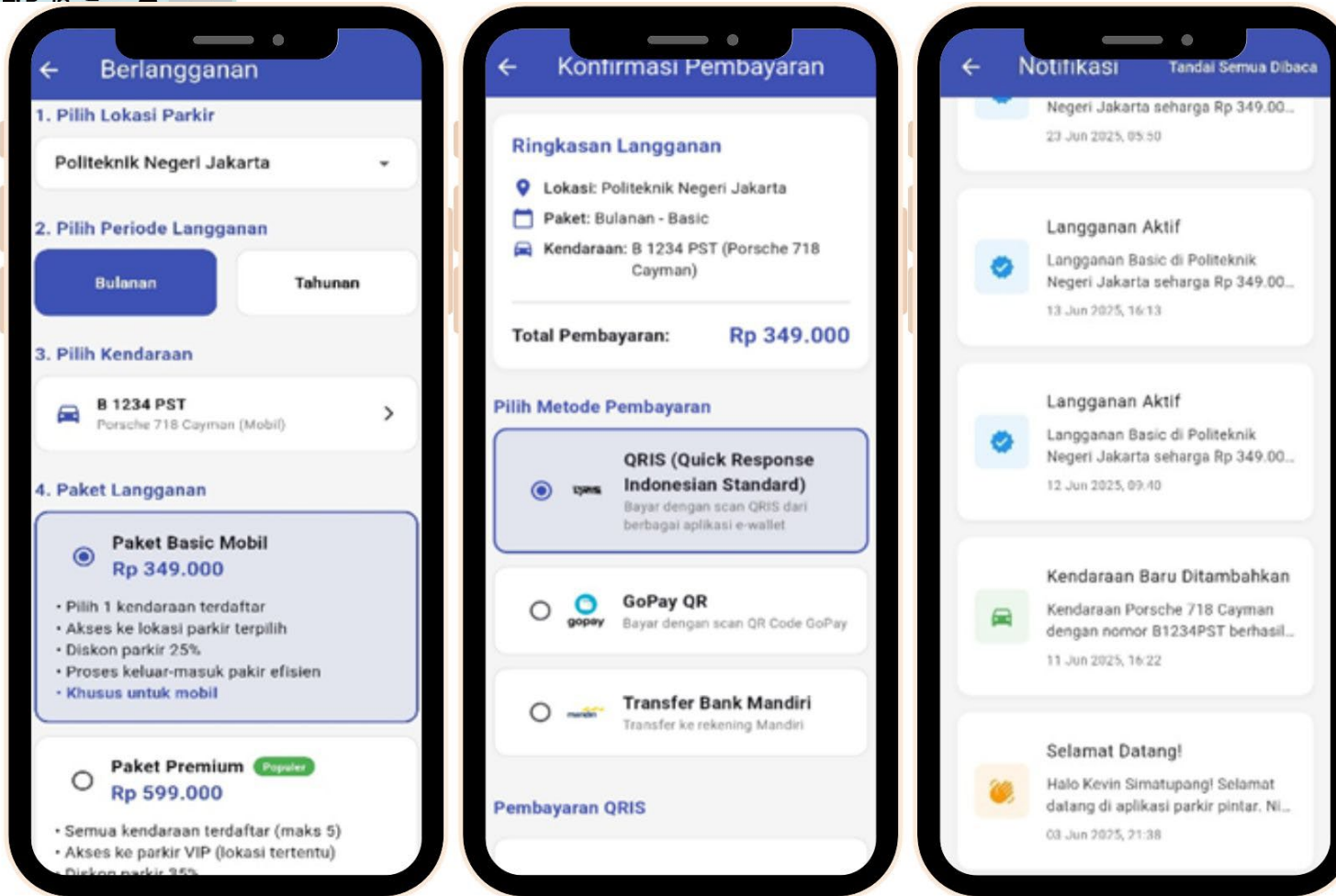
TAMPILAN APLIKASI ANDROID 2



**PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar	: Lukman Alghifarri
Diperiksa	: Rifqi Fuadi Hasani, S.T., M.T.
Tanggal	: 18 Juni 2025

Hak Cipta mil
1. Dilaran
a. Peng
b. Peng
2. Dilaran
tanpa i



03

TAMPILAN APLIKASI ANDROID 3



PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA

Digambar	: Lukman Alghifarri
Diperiksa	: Rifqi Fuadi Hasani, S.T., M.T.
Tanggal	: 18 Juni 2025

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Source Code main.dart

```
import 'package:flutter/material.dart';
import
'package:flutter_native_splash/flutter_native_splash.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:provider/provider.dart';
import 'firebase_options.dart';
import '/screens/early_screen/onboarding_screen.dart';
import 'screens/early_screen/login_screen.dart';
import 'screens/main_screen/dashboard/dashboard_screen.dart';
import 'screens/admin/providers/admin_provider.dart';

void main() async {
  WidgetsBinding widgetsBinding =
  WidgetsFlutterBinding.ensureInitialized();

  FlutterNativeSplash.preserve(widgetsBinding: widgetsBinding);

  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );

  final prefs = await SharedPreferences.getInstance();
  final showOnboarding = prefs.getBool('showOnboarding') ??
true;

  final user = FirebaseAuth.instance.currentUser;

  runApp(MyApp(
    showOnboarding: showOnboarding,
    isLoggedIn: user != null,
  ));
}

class MyApp extends StatefulWidget {
  final bool showOnboarding;
  final bool isLoggedIn;

  const MyApp({
    super.key,
    required this.showOnboarding,
    required this.isLoggedIn,
  });

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  void initState() {
    super.initState();
  }
}
```

Source Code onboarding_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import 'package:flutter/material.dart';
import 'package:introduction_screen/introduction_screen.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:animate_do/animate_do.dart';
import 'login_screen.dart';

class OnboardingScreen extends StatelessWidget {
  OnboardingScreen({super.key});
  final introKey = GlobalKey<IntroductionScreenState>();

  void _onIntroEnd(context) {
    Navigator.of(context).pushReplacement(
      MaterialPageRoute(builder: (_) => const LoginScreen()),
    );
  }

  @override
  Widget build(BuildContext context) {
    final screenSize = MediaQuery.of(context).size;
    final titleStyle = GoogleFonts.poppins(
      fontSize: 28.0,
      fontWeight: FontWeight.bold,
      color: const Color(0xFF3F51B5),
    );
    final bodyStyle = GoogleFonts.poppins(
      fontSize: 16.0,
      color: const Color(0xFF424242),
      height: 1.5,
    );
    final buttonStyle = ElevatedButton.styleFrom(
      backgroundColor: const Color(0xFF3F51B5),
      foregroundColor: Colors.white,
      padding: const EdgeInsets.symmetric(horizontal: 30,
        vertical: 12),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
      ),
      elevation: 3,
    );

    Widget buildImageContainer(String imagePath) {
      return Container(
        width: screenSize.width * 0.8,
        height: screenSize.height * 0.4,
        decoration: BoxDecoration(
          color: const Color(0xFFE8EAF6),
          borderRadius: BorderRadius.circular(30),
        ),
        child: ClipRRect(
          borderRadius: BorderRadius.circular(30),
          child: Image.asset(
            imagePath,
            fit: BoxFit.contain,
          ),
        ),
      );
    }
  }
}
```

Source Code login_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  bool _isPasswordVisible = false;

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  Future<bool> _isAdmin(String uid) async {
    try {
      DocumentSnapshot adminDoc = await
        FirebaseFirestore.instance
          .collection('admins')
          .doc(uid)
          .get();
      return adminDoc.exists;
    } catch (e) {
      debugPrint('Error checking admin status: $e');
      return false;
    }
  }

  void _navigateBasedOnRole(bool isAdmin) {
    if (isAdmin) {
      Navigator.of(context).pushAndRemoveUntil(
        MaterialPageRoute(builder: (context) => const
          AdminDashboardScreen()),
        (route) => false,
      );
    }
  }

  // Firebase Email/Password Login with Role Detection
  Future<void> _signInWithEmailAndPassword() async {
    if (!_formKey.currentState!.validate()) return;

    setState(() {
      _isLoading = true;
    });
  }
}
```

Source Code register_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

class RegisterScreen extends StatefulWidget {
  const RegisterScreen({super.key});

  @override
  State<RegisterScreen> createState() =>
    _RegisterScreenState();
}

class _RegisterScreenState extends State<RegisterScreen> {
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  final _confirmPasswordController = TextEditingController();
  final _nameController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  bool _isPasswordVisible = false;
  bool _isConfirmPasswordVisible = false;

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    _confirmPasswordController.dispose();
    _nameController.dispose();
    super.dispose();
  }

  Future<void> _registerWithEmailAndPassword() async {
    if (!_formKey.currentState!.validate()) return;

    setState(() {
      _isLoading = true;
    });

    try {
      // Create the user account in Firebase
      final userCredential = await
        FirebaseAuth.instance.createUserWithEmailAndPassword(
          email: _emailController.text.trim(),
          password: _passwordController.text.trim(),
        );

      await
        userCredential.user?.updateDisplayName(_nameController.text.trim());

      if (mounted) {
        setState(() {
          _isLoading = false;
        });
      }
    }
  }
}
```

Source Code reset_password_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'dart:math' as math;

class ResetPasswordScreen extends StatefulWidget {
  const ResetPasswordScreen({super.key});

  @override
  State<ResetPasswordScreen> createState() =>
    _ResetPasswordScreenState();
}

class _ResetPasswordScreenState extends
  State<ResetPasswordScreen> with SingleTickerProviderStateMixin
{
  final _emailController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  bool _isLoading = false;
  bool _emailSent = false;

  // Animation controller for floating icons
  late AnimationController _animationController;
  final List<FloatingIcon> _floatingIcons = [];

  @override
  void initState() {
    super.initState();

    // Initialize animation controller
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 10),
    )..repeat();

    // Create floating icons
    for (int i = 0; i < 5; i++) {
      _floatingIcons.add(FloatingIcon(
        icon: _getRandomIcon(),
        position: Offset(
          math.Random().nextDouble() * 300,
          math.Random().nextDouble() * 600,
        ),
        size: math.Random().nextDouble() * 20 + 10,
        speed: math.Random().nextDouble() * 2 + 0.5,
      ));
    }
  }

  IconData _getRandomIcon() {
    final icons = [
      Icons.lock_reset,
      Icons.password,
      Icons.security,
      Icons.key,
      Icons.lock_open,
    ];
  }
}
```

Source Code dashboard_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import '/screens/main_screen/history/history_screen.dart';
import '/screens/notif_screen/notification_screen.dart';

class DashboardScreen extends StatefulWidget {
  const DashboardScreen({super.key});

  @override
  State<DashboardScreen> createState() =>
    _DashboardScreenState();
}

class _DashboardScreenState extends State<DashboardScreen> {
  int _selectedIndex = 0;
  String _selectedVehicleType = 'Mobil';

  final DashboardService _dashboardService =
    DashboardService();
  SubscriptionStatus? _subscriptionStatus;
  bool _isLoadingSubscription = true;
  String? _subscriptionError;

  @override
  void initState() {
    super.initState();
    _loadDashboardData();
  }

  Future<void> _loadDashboardData() async {
    await _loadSubscriptionData();
  }

  Future<void> _loadSubscriptionData() async {
    try {
      setState(() {
        _isLoadingSubscription = true;
        _subscriptionError = null;
      });

      await
        _dashboardService.checkAndUpdateExpiredSubscriptions();

      final subscription = await
        _dashboardService.getUserSubscription();

      if (mounted) {
        setState(() {
          _subscriptionStatus = subscription;
          _isLoadingSubscription = false;
        });
      }
    }
  }
}
```

Source Code notif_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// notification_screen.dart
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:intl/intl.dart';
import 'package:intl/date_symbol_data_local.dart'; // Tambahkan
import ini

class NotificationScreen extends StatefulWidget {
  const NotificationScreen({super.key});

  @override
  State<NotificationScreen> createState() =>
    _NotificationScreenState();
}

class _NotificationScreenState extends
  State<NotificationScreen> {
  final FirebaseFirestore _firestore =
    FirebaseFirestore.instance;
  final FirebaseAuth _auth = FirebaseAuth.instance;
  bool _isLocaleInitialized = false; // Flag untuk cek apakah
  locale sudah diinisialisasi

  @override
  void initState() {
    super.initState();
    _initializeLocaleAndNotifications();
  }

  // Inisialisasi locale dan notifikasi
  Future<void> _initializeLocaleAndNotifications() async {
    try {
      // Inisialisasi locale untuk formatting tanggal
      await _initializeDateFormatting();
      await _initializeNotifications();
    } catch (e) {
      print('Error initializing locale and notifications: $e');
    }
  }

  // Inisialisasi date formatting locale
  Future<void> _initializeDateFormatting() async {
    try {
      await initializeDateFormatting('id_ID', null);
      setState(() {
        _isLocaleInitialized = true;
      });
    } catch (e) {
      print('Error initializing date formatting: $e');
      // Fallback ke locale default jika gagal
      setState(() {
        _isLocaleInitialized = true;
      });
    }
  }
}
```

Source Code subscription_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
class SubscriptionScreen extends StatefulWidget {
  const SubscriptionScreen({super.key});

  @override
  State<SubscriptionScreen> createState() =>
    _SubscriptionScreenState();
}

class _SubscriptionScreenState extends
  State<SubscriptionScreen> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
    FirebaseFirestore.instance;

  // Form state
  String? _selectedParkingLocation;
  String _selectedPeriod = 'Bulanan';
  String _selectedTier = 'Basic';
  List<Map<String, dynamic>> _selectedVehicles = [];
  List<Map<String, dynamic>> _userVehicles = [];
  bool _isLoading = false;

  // Subscription status
  Map<String, dynamic>? _activeSubscription;
  bool _isLoadingSubscription = true;

  // Static parking locations (from your
  parking_spot_screen.dart)
  final List<String> _parkingLocations = [
    'Politeknik Negeri Jakarta',
    'Menara Satu Sentra',
    'Citra Tower',
    'Kirana Two Office Tower',
    'The Prime Office Suite'
  ];

  // Subscription plans with pricing (updated with vehicle-
  based Basic pricing)
  final Map<String, Map<String, int>> _pricingPlans = {
    'Bulanan': {
      'BasicMotor': 149000,      // Basic untuk Motor
      'BasicMobil': 349000,     // Basic untuk Mobil
      'Premium': 599000,
    },
    'Tahunan': {
      'BasicMotor': 1349000,    // Basic untuk Motor (hemat
      ~25%)
      'BasicMobil': 3099000,    // Basic untuk Mobil (hemat
      ~15%)
      'Premium': 5399000,
    },
  };
};
```

Source Code profile_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
class ProfileScreen extends StatefulWidget {
  const ProfileScreen({super.key});

  @override
  State<ProfileScreen> createState() => _ProfileScreenState();
}

class _ProfileScreenState extends State<ProfileScreen> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
    FirebaseFirestore.instance;
  String _userName = "User";
  String _userEmail = "email@example.com";
  String _phoneNumber = "";
  String? _photoUrl;
  String? _localPhotoPath;
  bool _isLoading = true;
  int _vehicleCount = 0;

  // Subscription status
  Map<String, dynamic>? _activeSubscription;
  bool _isLoadingSubscription = true;

  @override
  void initState() {
    super.initState();
    _loadUserData();
  }

  @override
  void didChangeDependencies() {
    super.didChangeDependencies();
    // This ensures the profile is refreshed when navigating
    back to this screen
    _loadLocalPhoto();
    _loadVehicleCount();
    _loadActiveSubscription(); // Add this to refresh
    subscription status
  }

  Future<void> _loadUserData() async {
    setState(() {
      _isLoading = true;
    });

    } catch (e) {
      debugPrint('Error loading subscription: $e');
      setState(() {
        _activeSubscription = null;
        _isLoadingSubscription = false;
      });
    }
  }
}
```

Source Code registered_vehicles_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class RegisteredVehiclesScreen extends StatefulWidget {
  const RegisteredVehiclesScreen({super.key});

  @override
  State<RegisteredVehiclesScreen> createState() =>
    _RegisteredVehiclesScreenState();
}

class _RegisteredVehiclesScreenState extends
  State<RegisteredVehiclesScreen> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
    FirebaseFirestore.instance;

  List<Map<String, dynamic>> _vehicles = [];
  bool _isLoading = true;

  @override
  void initState() {
    super.initState();
    _loadVehicles();
  }

  // Format license plate with spaces for display
  String _formatLicensePlateForDisplay(String plate) {
    // Ensure the plate is uppercase
    plate = plate.toUpperCase();

    // Apply specific Indonesian license plate format (e.g., B
    1234 XYZ)
    // This is a simplified approach - may need adjustment for
    all plate formats
    if (plate.length >= 3) {
      // Find the first digit position
      int firstDigitPos = -1;
      for (int i = 0; i < plate.length; i++) {
        if (RegExp(r'[0-9]').hasMatch(plate[i])) {
          firstDigitPos = i;
          break;
        }
      }

      if (firstDigitPos > 0) {
        // Find the first letter after digits
        int firstLetterAfterDigitsPos = -1;
        for (int i = firstDigitPos; i < plate.length; i++) {
          if (RegExp(r'[A-Z]').hasMatch(plate[i])) {
            firstLetterAfterDigitsPos = i;
            break;
          }
        }
      }
    }
  }
}
```

Source Code notification_screen.dart

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';

class NotificationScreen extends StatefulWidget {
  const NotificationScreen({super.key});

  @override
  State<NotificationScreen> createState() =>
    _NotificationScreenState();
}

class _NotificationScreenState extends
  State<NotificationScreen> {
  final FirebaseFirestore _firestore =
    FirebaseFirestore.instance;
  final FirebaseAuth _auth = FirebaseAuth.instance;
  bool _isLocaleInitialized = false; // Flag untuk cek apakah
  locale sudah diinisialisasi

  @override
  void initState() {
    super.initState();
    _initializeLocaleAndNotifications();
  }

  // Inisialisasi locale dan notifikasi
  Future<void> _initializeLocaleAndNotifications() async {
    try {
      // Inisialisasi locale untuk formatting tanggal
      await _initializeDateFormatting();
      await _initializeNotifications();
    } catch (e) {
      print('Error initializing locale and notifications: $e');
    }
  }

  // Inisialisasi date formatting locale
  Future<void> _initializeDateFormatting() async {
    try {
      await initializeDateFormatting('id_ID', null);
      setState(() {
        _isLocaleInitialized = true;
      });
    } catch (e) {
      print('Error initializing date formatting: $e');
      // Fallback ke locale default jika gagal
      setState(() {
        _isLocaleInitialized = true;
      });
    }
  }

  // Inisialisasi notifikasi saat pertama kali masuk
  Future<void> _initializeNotifications() async {
    final user = _auth.currentUser;
```

Source Code admin_dashboard_screen.dart



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
// screens/admin/admin_dashboard_screen.dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:provider/provider.dart';
import '../early_screen/login_screen.dart';
import 'providers/admin_provider.dart';
import 'widgets/statistics_card_widget.dart';
import 'widgets/subscription_list_widget.dart';

class AdminDashboardScreen extends StatefulWidget {
  const AdminDashboardScreen({super.key});

  @override
  State<AdminDashboardScreen> createState() =>
    _AdminDashboardScreenState();
}

class _AdminDashboardScreenState extends
  State<AdminDashboardScreen>
  with SingleTickerProviderStateMixin {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  late TabController _tabController;

  @override
  void initState() {
    super.initState();
    _tabController = TabController(length: 4, vsync: this);

    // Initialize provider after widget is built
    WidgetsBinding.instance.addPostFrameCallback((_) {
      final adminProvider = Provider.of<AdminProvider>(context,
        listen: false);
      adminProvider.initialize();
    });
  }

  @override
  void dispose() {
    _tabController.dispose();
    super.dispose();
  }

  Future<void> _handleLogout() async {
    try {
      await _auth.signOut();
      if (mounted) {
        Navigator.pushAndRemoveUntil(
          context,
          MaterialPageRoute(builder: (context) => const
            LoginScreen()),
            (route) => false,
        );
      }
    } catch (e) {
      // Handle logout error if needed
    }
  }
}
```