

APLIKASI CARSECURE MENGGUNAKAN FRAMEWORK FLUTTER UNTUK MEMBANTU OPTIMALISASI SISTEM IOT CARSECURE

Taryono Ibnu Affan^{1*}, Dian Ade Kurnia², Irfan Ali³

^{1,2} STMIK IKMI Cirebon; Jl. Perjuangan No.10B, Karyamulya, Kec. Kesambi, Kota Cirebon, Jawa Barat

³ STMIK IKMI Cirebon; Jl. Perjuangan No.10B, Karyamulya, Kec. Kesambi, Kota Cirebon, Jawa Barat

Keywords:

ESP32, Firebase Realtime Database, Flutter, Internet of Things, Keamanan Kendaraan.

Corespondent Email:

aavanautoblaze@gmail.com



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Perkembangan teknologi informasi memberikan pengaruh besar terhadap sistem keamanan kendaraan modern, khususnya melalui pemanfaatan Internet of Things (IoT) yang memungkinkan perangkat saling terhubung dan dikendalikan secara jarak jauh. Penelitian ini bertujuan mengembangkan aplikasi CarSecure berbasis Flutter untuk mengoptimalkan sistem keamanan kendaraan IoT dengan fitur utama seperti pemantauan lokasi real-time, pengendalian mesin jarak jauh, dan notifikasi keamanan. Pengembangan dilakukan menggunakan model Waterfall, mencakup analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Aplikasi ini terintegrasi dengan perangkat ESP32, modul GPS, dan relay, serta menggunakan Firebase Realtime Database sebagai media komunikasi data dua arah. Hasil evaluasi menunjukkan bahwa aplikasi mampu mengeksekusi perintah “Matikan Mesin” dengan waktu respon rata-rata 2–3 detik, memberikan notifikasi keamanan dalam waktu kurang dari 5 detik, serta menampilkan akurasi GPS dengan deviasi ± 7 meter. Koneksi antara aplikasi dan perangkat IoT juga terbukti stabil dengan tingkat kestabilan 100% selama 30 menit pengujian tanpa putus. Evaluasi kepuasan pengguna terhadap 20 responden menghasilkan skor rata-rata 4.46, dengan aspek kemudahan penggunaan (4.6), kecepatan respon sistem (4.5), tampilan antarmuka (4.5), dan keandalan sistem (4.3). Berdasarkan hasil tersebut, aplikasi CarSecure dinilai responsif, akurat, mudah digunakan, dan efektif dalam mendukung keamanan kendaraan berbasis IoT, serta memiliki potensi untuk dikembangkan lebih lanjut dengan fitur keamanan berbasis kecerdasan buatan.

1. PENDAHULUAN

Teknologi telah menjadi bagian penting kehidupan modern dan berkembang sangat cepat, memungkinkan manusia bekerja lebih efisien melalui pemanfaatan teknologi canggih [1]. Perkembangan komputer, termasuk AI, komputasi awan, dan IoT, mendorong transformasi digital di berbagai sektor dan melahirkan inovasi seperti perangkat pintar serta sistem berbasis kecerdasan buatan [2].

Dalam konteks otomotif, IoT menjadi fondasi utama sistem keamanan seperti CarSecure, namun masih menghadapi tantangan pada integrasi antarmuka pengguna dan aplikasi lintas platform. Flutter menawarkan solusi pengembangan modern, meskipun integrasinya dengan IoT masih memiliki kesenjangan literatur sehingga perlu diteliti lebih lanjut.

Penelitian terdahulu menunjukkan berbagai implementasi IoT pada sistem keamanan kendaraan, seperti sistem pelacak dan pengendali mesin berbasis NodeMCU dan App Inventor yang terbukti efektif serta mudah digunakan [3], sistem keamanan berbasis ESP32 dan Android dengan fitur alarm, kontrol mesin, voice command, dan sensor getaran yang responsif [4] serta sistem kontrol kendaraan berbasis ESP32 dan Blynk yang mampu bekerja efektif dan memberikan tingkat keamanan tinggi [5].

Penelitian ini bertujuan mengembangkan aplikasi CarSecure berbasis Flutter yang terintegrasi dengan IoT untuk menghasilkan antarmuka intuitif, responsif, dan lintas platform, sehingga mampu memudahkan pemantauan serta pengendalian kendaraan secara real-time. Kontribusi penelitian diharapkan memberikan manfaat praktis sekaligus memperkaya referensi mengenai integrasi Flutter dalam sistem IoT. Metode Agile digunakan untuk mendukung fleksibilitas dan iterasi cepat, mencakup perencanaan, desain, pengembangan, pengujian, dan evaluasi berulang, serta analisis integrasi IoT dan uji antarmuka pengguna.

Jika berhasil, penelitian ini akan berkontribusi pada pengembangan aplikasi mobile IoT yang efisien, memperkuat pemahaman praktik terbaik integrasi aplikasi lintas platform dengan perangkat IoT, serta mendorong inovasi pada smart vehicle dan sistem keamanan kendaraan berbasis mobile. Penelitian ini juga dapat menjadi dasar bagi studi lanjutan terkait optimalisasi performa sistem IoT di masa depan.

2. TINJAUAN PUSTAKA

2.1. *Internet Of Things (IoT)*

Internet of Things (IoT) merupakan integrasi teknologi internet dengan perangkat keras dan perangkat lunak untuk menciptakan komunikasi yang saling terhubung antar perangkat serta memungkinkan pemrosesan data menjadi informasi yang bernilai. Teknologi ini memungkinkan pengendalian perangkat dari jarak jauh dan telah diterapkan dalam berbagai bidang, seperti kesehatan, peternakan, dan otomotif. IoT juga erat terkait dengan komunikasi machine-to-machine (M2M) di

sektor industri, yang melahirkan perangkat pintar atau sistem cerdas [6]

2.2. *Sistem Keamanan Berkendara Berbasis IoT*

Sistem keamanan kendaraan modern memanfaatkan deteksi dini, pemantauan real-time, dan respons otomatis dengan dukungan sensor seperti limit switch, sensor getaran, serta modul GPS yang terhubung melalui protokol komunikasi seperti MQTT atau HTTP. Pada penelitian yang menggunakan mikrokontroler ESP32 dan platform Thingsboard, sistem dirancang untuk memberikan notifikasi jarak jauh ketika pintu dibuka, kaca pecah, atau kendaraan berpindah lokasi, dan dikendalikan melalui aplikasi Flutter. Hasil pengujian menunjukkan notifikasi rata-rata muncul dalam 5 detik dengan akurasi GPS 0–10 meter pada sinyal kuat dan hingga 350 meter pada sinyal lemah [7].

2.3. *Framework Flutter*

Flutter adalah SDK dari Google yang memungkinkan pengembangan aplikasi lintas platform dari satu basis kode dengan arsitektur berbasis Widget Tree, Rendering Engine Skia, dan Framework Layer. Dalam pengembangan aplikasi IoT, Flutter menawarkan integrasi API yang mudah, tampilan data real-time, performa tinggi, serta UI responsif untuk memantau sensor, GPS, dan perangkat IoT lainnya. Dibandingkan framework lain, Flutter lebih unggul karena menggunakan bahasa Dart yang efisien, tidak memerlukan JavaScript bridge, serta menyediakan widget bawaan dan tools lengkap yang mendukung produktivitas [7]. Semua kode pada Flutter dikompilasi langsung menjadi kode native (melalui Android NDK, LLVM, dan AOT compiled) tanpa melibatkan interpreter pada proses eksekusinya, sehingga proses kompilasi dan performa aplikasinya menjadi lebih cepat. Implementasi kode dilakukan menggunakan sistem widget, yang dapat berupa komponen visual maupun komponen non-visual yang berfungsi sebagai wadah atau penampung bagi widget lainnya [8].

2.4. *Firebase*

Firebase adalah layanan dari Google yang memudahkan pengembangan aplikasi, terutama pada sistem IoT, dengan menyediakan backend untuk pemrosesan data, penyimpanan,

autentikasi, dan pengelolaan perangkat secara real-time. Realtime Database dan Firestore memungkinkan sinkronisasi data instan, termasuk dukungan akses offline, sementara layanan seperti Authentication dan Cloud Messaging menyediakan keamanan serta notifikasi cepat. Melalui integrasi Flutter dan SDK Firebase, aplikasi dapat menampilkan data IoT, melakukan autentikasi, serta menerima notifikasi secara real-time dengan mudah dan efisien [9]

2.5. Komunikasi Data IoT dengan Aplikasi

Komunikasi Data IoT dengan Aplikasi pada sistem ini memanfaatkan Firebase Realtime Database sebagai protokol komunikasi berbasis TCP/IP yang memungkinkan perangkat IoT berinteraksi menggunakan model publish-subscribe, di mana aplikasi mobile berperan sebagai publisher yang mengirim perubahan status ke database, sementara NodeMCU ESP8266 sebagai subscriber yang membaca setiap pembaruan tersebut. Mekanisme ini mendukung pengiriman dan penerimaan data secara real-time, sehingga setiap perubahan pada aplikasi langsung tersinkronisasi ke perangkat, meskipun delay dipengaruhi oleh kualitas jaringan seperti 3G, 4G, atau LAN. Selain itu, Firebase juga dimanfaatkan untuk pengolahan data sensor atau status aktuator, karena setiap nilai yang tersimpan pada database digunakan sebagai pemicu untuk mengendalikan relay, memastikan proses kontrol perangkat berjalan otomatis, cepat, dan konsisten dalam arsitektur IoT [10].

2.6. NodeMcu Esp32

ESP32 adalah mikrokontroler dengan dukungan Wi-Fi dan Bluetooth yang banyak digunakan dalam sistem IoT karena konsumsi dayanya rendah, kinerja prosesor tinggi, serta memiliki banyak pin input/output. ESP32 memungkinkan integrasi berbagai sensor, modul RTC, serta aplikasi berbasis internet untuk mengendalikan perangkat dari jarak jauh. Selain itu, ESP32 juga terbukti efektif untuk membangun sistem irigasi otomatis yang efisien dan adaptif terhadap kondisi lingkungan [11]. Board ini memiliki dua versi, yaitu 30 GPIO dan 36 GPIO, dengan fungsi yang pada dasarnya sama. Namun, versi 30 GPIO dipilih karena menyediakan dua pin GND yang lebih mendukung kebutuhan rangkaian. Semua pin

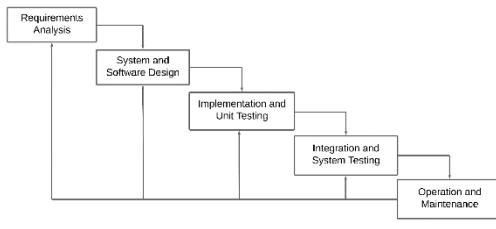
pada board sudah diberi label di bagian atas sehingga memudahkan proses identifikasi. Selain itu, board ini dilengkapi dengan interface USB to UART yang membuatnya mudah diprogram menggunakan platform pengembangan seperti Arduino IDE. Sumber daya untuk board dapat diberikan melalui konektor micro USB [12].

2.7. Global Positioning System (GPS)

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang memungkinkan penentuan posisi geografis secara akurat menggunakan prinsip trilaterasi. Perangkat GPS menerima sinyal dari satelit, menghitung waktu tempuh sinyal, dan menentukan posisi dalam bentuk koordinat lintang, bujur, serta ketinggian. Dalam sistem keamanan kendaraan, GPS berfungsi sebagai alat pelacak lokasi dengan mengirimkan titik koordinat ke server atau aplikasi melalui jaringan GSM untuk ditampilkan dalam peta digital. Teknologi ini memungkinkan pemilik kendaraan memantau posisi kendaraannya kapan saja melalui aplikasi, sekaligus merekam histori lokasi untuk kebutuhan keamanan dan pengawasan [13]. Sistem GPS bekerja dengan memanfaatkan satelit yang mengirimkan sinyal dalam bentuk gelombang mikro ke permukaan bumi. Sinyal tersebut kemudian diterima dan dibaca oleh perangkat penerima di bumi, yang selanjutnya diolah untuk menentukan dan menampilkan posisi pengguna secara akurat [14]

3. METODE PENELITIAN

Pendekatan yang digunakan penulis dalam penelitian ini adalah metode Waterfall, yang juga dikenal sebagai Classic Life Cycle atau Linear Sequential Model. Metode ini memberikan kerangka kerja yang sistematis dan berurutan dalam proses pengembangan perangkat lunak, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Pendekatan ini dipilih karena mampu memberikan alur kerja yang terstruktur, sehingga memudahkan peneliti dalam merancang, mengimplementasikan, dan mengevaluasi sistem secara lebih terarah [15]



Gambar 1 Metode Waterfall

Berdasarkan gambar 3.1 Metode Waterfall adalah salah satu model dalam pengembangan perangkat lunak yang mengharuskan setiap tahap penyelesaian dilakukan secara berurutan dan linier. Metode ini cocok digunakan pada proyek dengan spesifikasi yang sudah jelas dan tidak banyak perubahan selama pengembangan. Model ini melibatkan 5 tahapan yaitu, Analisis Kebutuhan Sistem, Desain Sistem, Implementasi, Pengujian Sistem, Pemeliharaan dan Perbaikan.

3.1. Requirements Analysis

Pada tahap Analisis Kebutuhan, proses dimulai dengan mengidentifikasi kebutuhan sistem melalui pengumpulan informasi dari pengguna serta dokumentasi teknis terkait sistem IoT CarSecure. Langkah ini bertujuan menentukan kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dikembangkan. Selanjutnya dilakukan studi literatur dengan mengkaji jurnal, artikel ilmiah, dan dokumentasi teknis terkait pengembangan aplikasi Flutter, integrasi IoT, serta teknologi keamanan kendaraan. Hasil kajian tersebut menjadi dasar dalam merumuskan spesifikasi kebutuhan sistem yang mencakup fitur utama, batasan perangkat, dan mekanisme integrasi aplikasi dengan perangkat IoT seperti ESP32 dan modul GPS. Tahap ini menghasilkan dokumen spesifikasi kebutuhan yang menjadi acuan untuk tahap pengembangan berikutnya.

3.2. System And Software Design

Tahap Perancangan Sistem dilakukan dengan menyusun desain antarmuka pengguna (UI/UX), arsitektur aplikasi, serta rancangan komunikasi IoT. Proses desain diawali dengan pembuatan wireframe dan prototipe yang berfokus pada kemudahan penggunaan serta tampilan antarmuka yang intuitif. Selanjutnya disusun diagram alur seperti flowchart dan diagram UML (use case, class, dan activity) untuk menggambarkan struktur kerja aplikasi

secara lebih detail. Pada tahap ini juga dirancang mekanisme komunikasi antara aplikasi dan perangkat IoT, termasuk format data, protokol komunikasi, dan metode pengiriman perintah dari aplikasi ke perangkat ESP32. Perancangan tersebut memastikan alur kerja aplikasi sesuai kebutuhan pengguna dan kompatibel dengan perangkat IoT.

3.3. Implementation And Unit Testing

Pada tahap Implementasi, seluruh desain yang telah disusun diterjemahkan ke dalam kode program menggunakan framework Flutter dan bahasa pemrograman Dart. Pengembangan aplikasi mencakup pembuatan tampilan antarmuka, proses autentikasi pengguna berbasis User ID, serta integrasi fitur seperti monitoring kendaraan, pemetaan lokasi GPS secara real-time, kontrol mesin jarak jauh, dan notifikasi keamanan. Selain itu, dilakukan pengaturan konektivitas aplikasi dengan perangkat IoT CarSecure melalui protokol komunikasi yang telah dirancang sebelumnya. Implementasi juga meliputi integrasi fitur keamanan tambahan, seperti penyimpanan data real-time dan respons sistem terhadap aktivitas mencurigakan.

3.4. Integration And System Testing

Tahap berikutnya adalah Pengujian Sistem, yang dilakukan melalui pengujian unit, pengujian integrasi, dan uji coba pengguna. Pengujian unit bertujuan memastikan setiap modul atau fungsi aplikasi dapat berjalan dengan baik secara terpisah, sedangkan pengujian integrasi memastikan koneksi antara aplikasi dan sistem IoT berjalan stabil serta mampu menampilkan data secara real-time. Setelah seluruh fungsi teknis dinyatakan bekerja dengan baik, dilakukan uji coba pengguna untuk menilai kenyamanan, kemudahan penggunaan, dan keandalan aplikasi dalam kondisi penggunaan nyata. Tahap pengujian ini menghasilkan catatan perbaikan yang digunakan untuk penyempurnaan sistem pada tahap selanjutnya.

3.5. Operation And Maintenance

Tahap terakhir adalah Pemeliharaan Sistem, yang mencakup perbaikan bug, penyempurnaan fitur, dan pembaruan aplikasi secara berkala. Pada fase ini dilakukan debugging berdasarkan hasil pengujian maupun keluhan pengguna,

serta penyesuaian fitur sesuai umpan balik dan kebutuhan sistem. Pembaruan aplikasi juga dilakukan untuk menjaga kompatibilitas dengan versi terbaru sistem operasi dan perangkat IoT. Tahap pemeliharaan ini memastikan aplikasi tetap berfungsi optimal dan terus berkembang mengikuti kebutuhan pengguna serta perkembangan teknologi.

4. HASIL DAN PEMBAHASAN

4.1. Hasil

Implementasi Aplikasi Carsecure Menggunakan Framework Flutter Untuk Membantu Optimalisasi Sistem IoT Carsecure dilakukan menggunakan metode Waterfall yang meliputi tahapan Requirements Analysis, System and Software Design, Implementation and Unit Testing, Integration and System Testing dan Operation and Maintenance.

4.1.1. Requirements Analysis

Pada tahap Requirements Analysis, dilakukan identifikasi kebutuhan sistem untuk memastikan aplikasi CarSecure berfungsi optimal sebagai solusi keamanan kendaraan berbasis IoT. Analisis ini mencakup kebutuhan fungsional seperti autentikasi pengguna, pemantauan lokasi GPS real-time, kontrol mesin, notifikasi keamanan, dan pengelolaan data melalui Firebase, serta kebutuhan non-fungsional seperti performa responsif, UI/UX yang mudah digunakan, stabilitas koneksi IoT, dan keamanan data. Hasil analisis tersebut menjadi dasar dalam perancangan arsitektur sistem dan integrasi antara Flutter, Firebase, dan perangkat IoT agar aplikasi dapat bekerja secara efektif dan memenuhi kebutuhan pengguna.

1) Kebutuhan Hardware (Perangkat Keras)
Kebutuhan hardware pada sistem CarSecure meliputi perangkat-perangkat yang digunakan untuk membangun dan menjalankan modul IoT pada kendaraan. Hardware utama yang diperlukan adalah:

a) ESP32 sebagai mikrokontroler utama, berfungsi untuk memproses data, menerima perintah dari aplikasi, dan mengirim status kendaraan secara real-time ke Firebase. Spesifikasi yang direkomendasikan mencakup dukungan Wi-Fi yang stabil, prosesor dual-core, kapasitas memori yang cukup, serta GPIO yang kompatibel dengan modul tambahan.

- b) Modul GPS (seperti Neo-6M atau M8N) yang digunakan untuk memperoleh data lokasi kendaraan. Modul ini harus memiliki tingkat akurasi yang tinggi, kemampuan menangkap satelit dengan cepat, serta kompatibilitas antarmuka UART atau I2C.
- c) Modul Relay sebagai aktuator untuk mengendalikan kelistrikan kendaraan, khususnya untuk fungsi memutus dan menghubungkan arus mesin. Relay yang digunakan harus memiliki rating arus yang sesuai standar otomotif agar aman dan andal.
- d) Sumber daya (power supply) dan regulator tegangan, yang memastikan perangkat berjalan stabil meskipun terjadi fluktuasi tegangan pada kendaraan. Power supply harus mampu menyediakan arus yang cukup untuk ESP32, GPS, dan relay.
- e) Perangkat pendukung, seperti kabel jumper, konektor, PCB prototyping, terminal block, casing pelindung, dan alat uji seperti multimeter yang digunakan selama proses perakitan dan pengujian sistem.

2) Kebutuhan Software (Perangkat Lunak)
Kebutuhan software meliputi perangkat lunak yang digunakan dalam proses pengembangan aplikasi mobile dan pemrograman perangkat IoT. Software yang diperlukan antara lain:

- a) Flutter SDK dan bahasa Dart, digunakan untuk membangun aplikasi CarSecure dengan antarmuka yang modern dan responsif. Flutter dipilih karena kemampuannya menghasilkan aplikasi lintas platform secara efisien.
- b) Firebase Realtime Database, digunakan sebagai media komunikasi antara aplikasi dan ESP32. Selain itu, Firebase juga menyediakan infrastruktur untuk autentikasi, penyimpanan data, dan sinkronisasi real-time.
- c) Library Flutter pendukung, seperti `firebase_core`, `firebase_database`, `google_maps_flutter`, provider atau `getx` untuk state management, serta library notifikasi untuk memberikan peringatan keamanan kepada pengguna.
- d) Arduino IDE atau PlatformIO, digunakan untuk pemrograman ESP32. Library seperti Firebase-ESP-Client, TinyGPS++, dan ArduinoJson digunakan untuk mengelola komunikasi data, menangkap sinyal GPS, dan mem-parsing data JSON.
- e) Tools pengembangan lainnya, seperti Git untuk version control, Android Studio/VS Code

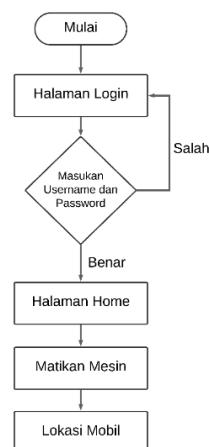
sebagai environment utama pengembangan aplikasi, serta Firebase Console untuk pengelolaan database, autentikasi, dan konfigurasi aplikasi.

4.1.2. System And Software Design

Perancangan dilakukan dengan membuat diagram alur sistem, use case, dan wireframe aplikasi. Komunikasi antara aplikasi dan perangkat IoT menggunakan firebase realtime database di mana ESP32 bertugas menerima perintah dan mengirimkan data status kendaraan ke server. Flutter dipilih karena mendukung performa tinggi, tampilan modern, serta efisiensi dalam proses integrasi IoT.

a) Flowchart

Untuk menjelaskan alur kerja sistem yang diimplementasikan pada aplikasi CarSecure, dibuat sebuah flowchart yang menggambarkan proses utama mulai dari pengguna membuka aplikasi hingga menjalankan fungsi-fungsi yang tersedia. Diagram ini berfungsi sebagai representasi visual dari logika sistem, sehingga memudahkan pemahaman terhadap urutan proses yang terjadi dalam aplikasi. Berikut merupakan flowchart dari alur kerja aplikasi CarSecure.



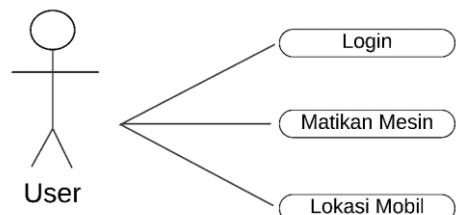
Gambar 2 Flowchart

Berdasarkan Gambar 2, flowchart tersebut menunjukkan alur kerja aplikasi CarSecure berbasis Flutter yang terhubung dengan sistem IoT. Proses dimulai dari pengguna membuka aplikasi dan masuk ke halaman login untuk memasukkan username dan user ID. Sistem kemudian melakukan validasi; jika data salah, muncul pesan kesalahan dan pengguna tetap di halaman login. Jika benar, pengguna diarahkan ke halaman Home yang berisi dua fitur utama, yaitu Matikan Mesin untuk memutus pengapian

kendaraan dari jarak jauh dan Lokasi Mobil untuk menampilkan posisi kendaraan secara real-time melalui GPS. Flowchart ini menggambarkan urutan proses login hingga akses ke fitur keamanan utama aplikasi.

b) Usecase

Untuk menggambarkan interaksi antara pengguna dengan sistem yang dibangun, digunakan use case diagram sebagai representasi hubungan antara aktor dan fitur-fitur utama aplikasi. Diagram ini membantu menjelaskan fungsi-fungsi yang dapat diakses oleh pengguna serta batasan tanggung jawab tiap komponen dalam sistem. Berikut merupakan use case diagram yang menunjukkan alur interaksi pengguna pada aplikasi CarSecure.



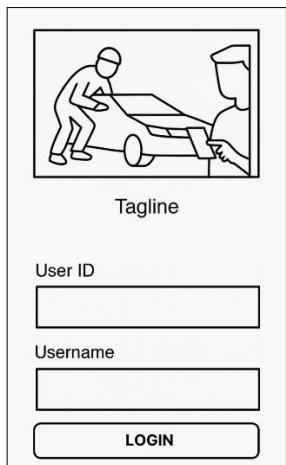
Gambar 3 Usecase

Berdasarkan Gambar 3, use case diagram menunjukkan hubungan antara pengguna (User) dan sistem aplikasi CarSecure berbasis Flutter yang terintegrasi dengan IoT. Pengguna berperan sebagai aktor utama yang menjalankan tiga fungsi utama, yaitu Login, Matikan Mesin, dan Lokasi Mobil. Melalui proses login, sistem melakukan autentikasi untuk memastikan hanya pengguna berwenang yang dapat mengakses aplikasi. Setelah berhasil masuk, pengguna dapat mengirim perintah Matikan Mesin ke perangkat IoT (ESP32) untuk memutus pengapian kendaraan secara jarak jauh, serta mengakses fitur Lokasi Mobil untuk melihat posisi kendaraan secara real-time dari data GPS. Diagram ini menggambarkan alur interaksi sederhana namun efektif dalam pengelolaan keamanan kendaraan melalui aplikasi CarSecure.

c) Wireframe

Untuk memberikan gambaran awal mengenai tampilan dan struktur antarmuka aplikasi CarSecure, disusun sebuah wireframe yang

merepresentasikan rancangan awal halaman login. Wireframe ini berfungsi sebagai acuan dalam proses pengembangan UI/UX, sehingga setiap elemen yang ditampilkan memiliki penempatan dan fungsi yang jelas. Berikut merupakan wireframe dari halaman login aplikasi CarSecure.



Gambar 4 Wireframe Halaman Login

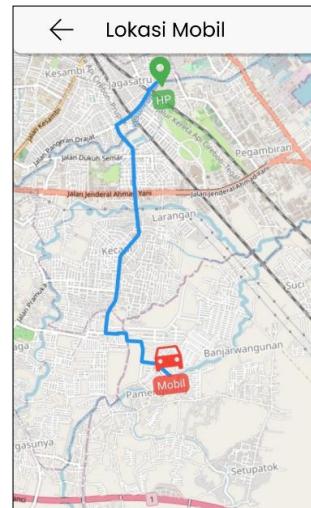
Wireframe pada Gambar 4 menampilkan desain halaman login aplikasi CarSecure yang terdiri dari ilustrasi visual, tagline, serta dua input utama yaitu User ID dan Username. Pengguna memasukkan data tersebut lalu menekan tombol LOGIN untuk proses autentikasi, yang akan mengarahkan mereka ke halaman Home jika data benar. Desain ini menunjukkan konsep antarmuka yang sederhana dan mudah dipahami sesuai prinsip user-friendly.



Gambar 5 Wireframe Halaman Home

Berdasarkan Gambar 5, wireframe halaman Home aplikasi CarSecure menampilkan teks sambutan pengguna di bagian atas serta dua ikon utama, yaitu Keamanan dan Mesin, yang mewakili fitur inti aplikasi. Fitur Keamanan digunakan untuk melihat status atau notifikasi

keamanan kendaraan, sedangkan fitur Mesin berfungsi untuk mengontrol pengapian kendaraan secara jarak jauh melalui modul ESP32. Bagian tengah menampilkan area Map untuk menunjukkan lokasi kendaraan secara real-time, dan ikon logout di kanan bawah digunakan untuk keluar dari aplikasi, sehingga keseluruhan tampilan terlihat sederhana, intuitif, dan mudah digunakan.



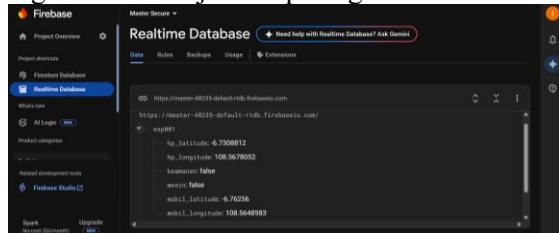
Gambar 6 Lokasi Mobil

Berdasarkan Gambar 6, wireframe halaman Lokasi Mobil menampilkan posisi kendaraan secara real-time melalui integrasi GPS dan sistem IoT. Halaman ini memiliki judul Lokasi Mobil dengan tombol kembali, serta peta yang menunjukkan titik HP berwarna hijau sebagai posisi pengguna dan ikon Mobil berwarna merah sebagai posisi kendaraan, lengkap dengan garis biru yang menggambarkan rute pergerakan. Desain ini memberikan tampilan sederhana dan informatif sehingga pengguna dapat memantau lokasi kendaraan dengan akurat dan efisien melalui aplikasi CarSecure.

d) Komunikasi IoT

Untuk mendukung proses pertukaran data secara real-time antara perangkat keras dan aplikasi mobile, sistem CarSecure memanfaatkan layanan Firebase Realtime Database sebagai penghubung utama dalam arsitektur Internet of Things (IoT). Melalui mekanisme ini, setiap informasi yang dikirimkan oleh perangkat, seperti status kendaraan, koordinat GPS, dan kondisi sistem, dapat tersimpan dan diperbarui secara langsung di dalam database. Data tersebut kemudian dapat diakses oleh aplikasi sehingga pengguna

mampu memantau dan mengendalikan kendaraan secara cepat, akurat, dan responsif. Ilustrasi mekanisme komunikasi yang digunakan ditunjukkan pada gambar berikut.



Gambar 7 Komunikasi IoT

Berdasarkan Gambar 7, mekanisme komunikasi IoT pada sistem CarSecure menghubungkan aplikasi Flutter dengan perangkat ESP32, modul GPS, dan sistem kendaraan. User ID (ESP001) digunakan sebagai identitas perangkat untuk memastikan koneksi yang aman, sementara hp_latitude dan hp_longitude menampilkan lokasi pengguna. Variabel keamanan dan mesin berfungsi sebagai kontrol Boolean untuk mengatur status keamanan serta mematikan atau menghidupkan mesin kendaraan dari jarak jauh. Adapun mobil_latitude dan mobil_longitude berasal dari modul GPS yang memberikan posisi kendaraan secara real-time. Secara keseluruhan, komunikasi berlangsung dua arah, yaitu aplikasi mengirim perintah ke ESP32 dan perangkat mengirim data lokasi serta status sistem kembali ke aplikasi, sehingga pemantauan dan kontrol kendaraan dapat dilakukan secara efektif dan responsif.

4.1.3. Implementation And Unit Testing

Implementasi meliputi pembuatan tampilan antarmuka (UI/UX), pengembangan fungsi autentikasi berbasis device ID, serta pengintegrasian fitur real-time tracking dan kontrol jarak jauh. Flutter diintegrasikan dengan database Firebase untuk autentikasi dan penyimpanan data, sementara komunikasi dengan perangkat IoT dilakukan melalui jaringan Wi-Fi menggunakan API khusus yang dikembangkan pada ESP32.



Gambar 8 Splashscreen

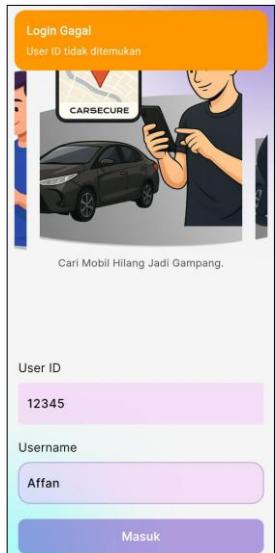
Berdasarkan Gambar 8, tampilan splash screen aplikasi CarSecure berfungsi sebagai layar awal yang muncul saat aplikasi dijalankan, menampilkan logo berupa ikon mobil dengan simbol lokasi dan peringatan sebagai representasi fitur keamanan berbasis GPS. Splash screen ini tampil selama 3 detik untuk proses inisialisasi sistem. Setelah itu, pengguna akan diarahkan ke halaman login jika belum autentikasi, atau langsung ke halaman Home jika sudah pernah login.



Gambar 9 Halaman Login

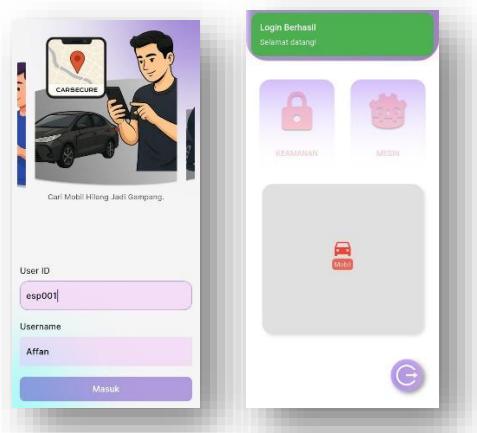
Berdasarkan Gambar 9, halaman login aplikasi CarSecure berfungsi sebagai proses autentikasi sebelum pengguna dapat mengakses fitur utama. Pada bagian atas terdapat ilustrasi grafis atau logo sebagai identitas aplikasi, diikuti tagline yang memperkuat pesan keamanan

kendaraan. Halaman ini menyediakan dua kolom input, yaitu User ID dan Username, yang harus diisi sebelum pengguna menekan tombol LOGIN untuk memulai proses autentikasi.



Gambar 10 Gagal Login

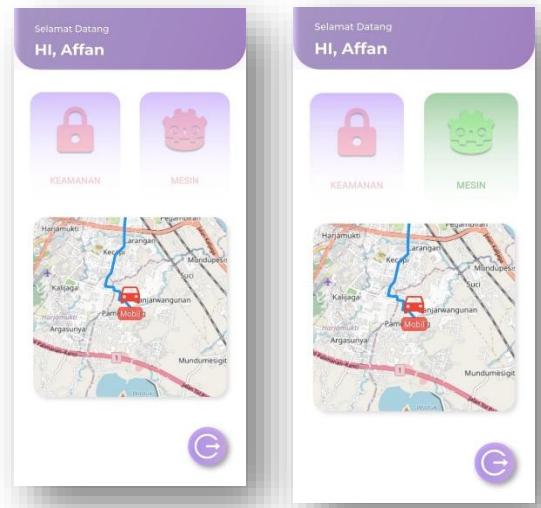
Berdasarkan Gambar 10, halaman login CarSecure meminta pengguna memasukkan User ID dan Username sebagai proses autentikasi. Jika User ID tidak terdaftar di realtime database, sistem menampilkan notifikasi "Login Gagal – User ID tidak ditemukan" sebagai umpan balik bahwa data tidak valid. Mekanisme ini memastikan hanya pengguna dengan informasi yang benar yang dapat mengakses aplikasi, sehingga keamanan sistem tetap terjaga.



Gambar 11 Login Berhasil

Berdasarkan Gambar 11, proses login pada aplikasi CarSecure dinyatakan berhasil ketika

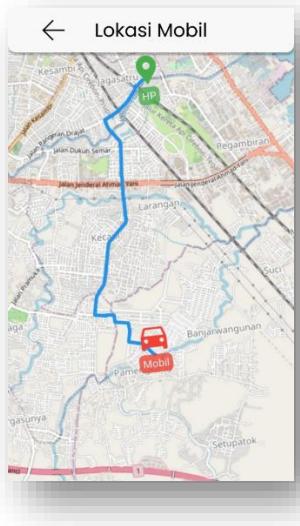
User ID dan Username yang dimasukkan sesuai dengan data di database, seperti contoh "esp001" dan "Affan". Setelah diverifikasi, sistem menampilkan notifikasi hijau "Login Berhasil – Selamat datang!" dan mengarahkan pengguna ke halaman Home. Pada halaman utama tersebut tersedia menu Keamanan, Mesin, dan Mobil sebagai akses ke fitur inti aplikasi, sehingga hanya pengguna terverifikasi yang dapat mengakses sistem secara aman.



Gambar 12 Halaman Home

Berdasarkan Gambar 12 di atas menampilkan halaman Home aplikasi CarSecure berbasis Flutter, yang menjadi tampilan utama setelah pengguna berhasil melakukan proses login. Pada bagian atas terdapat teks sambutan, misalnya "Selamat Datang, Hi Affan", sebagai bentuk personalisasi bagi pengguna. Di bawahnya terdapat dua ikon utama yang mewakili fitur inti aplikasi, yaitu Keamanan dan Mesin. Tombol Keamanan berfungsi untuk mengakses menu notifikasi atau memantau status keamanan kendaraan yang terhubung dengan perangkat IoT, seperti saat terjadi aktivitas mencurigakan atau percobaan akses kendaraan. Tombol Mesin digunakan untuk melakukan kontrol jarak jauh, seperti menyalakan atau mematikan mesin kendaraan melalui modul ESP32. Kedua tombol ini dilengkapi dengan indikator warna sebagai penanda status fitur — warna hijau menunjukkan fitur dalam kondisi aktif, sedangkan warna merah menandakan fitur tidak aktif. Pada bagian tengah halaman terdapat area Map yang menampilkan lokasi kendaraan

secara real-time berdasarkan data dari modul GPS yang terintegrasi dengan sistem IoT. Di bagian kanan bawah terdapat ikon logout berbentuk panah melingkar yang berfungsi untuk keluar dari aplikasi dan kembali ke halaman login.



Gambar 13 Lokasi Mobil

Berdasarkan Gambar 13, wireframe halaman Lokasi Mobil pada aplikasi CarSecure menampilkan posisi kendaraan secara real-time melalui integrasi IoT dan GPS. Tampilan ini dilengkapi judul “Lokasi Mobil”, tombol kembali, serta peta yang menunjukkan lokasi pengguna (ikon hijau), lokasi kendaraan (ikon merah), dan rute pergerakan kendaraan (garis biru). Data lokasi diperoleh dari modul GPS pada ESP32 dan ditampilkan langsung di aplikasi, sehingga pengguna dapat memantau keberadaan kendaraan dengan akurat dan efisien.

4.1.4. Integration And System Testing

Pengujian dilakukan dalam dua tahap:

- a) Uji fungsional (blackbox testing)

Pengujian fungsional dilakukan menggunakan metode blackbox testing, yaitu menguji setiap fitur aplikasi berdasarkan keluaran yang dihasilkan tanpa melihat kode sumber. Tujuan pengujian ini adalah untuk memastikan bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan sistem.

Tabel 1 Hasil Uji Fungsional Aplikasi CarSecure

No	Fitur yang Diuji	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Login menggunakan User ID	Pengguna memasukkan User ID untuk autentikasi awal	Autentikasi berhasil dan masuk ke halaman utama, Jika UserId Salah maka akan ada pemberitahuan User Id salah	Berhasil	Valid
2	Tampilan Dashboard/ Home	Menampilkan status kendaraan (aktif/nonaktif) dan lokasi GPS	Data tampil dengan benar dan real-time	Berhasil	Valid

N o	Fitur yang Diuji	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Stat us
3	Fitur “Matikan Mesin”	Menekan tombol <i>Shutdown Engine</i> untuk mengirim perintah ke ESP32	Mesin kendaraan mati dan status berubah menjadi “Off”	Berhasil 1	Vali d
4	Pelacakan Lokasi Kendaraan	Sistem menampilkan lokasi kendaraan pada peta	Lokasi tampil sesuai koordinat GPS	Berhasil 1	Vali d
5	Koneksi Aplikasi– IoT	Aplikasi terhubung dengan perangkat ESP32 melalui Wi-Fi	Koneksi stabil dan tidak terputus	Berhasil 1	Vali d
6	Logout Aplikasi	Pengguna keluar dari aplikasi dengan aman	Sistem kembali ke halaman login	Berhasil 1	Vali d

memastikan semua fitur berjalan sesuai fungsi. Hasil menunjukkan semua fitur berjalan 100% sesuai rancangan tanpa error.

b) Uji pengguna (user acceptance test)
dilakukan melalui kuesioner terhadap 20 responden pengguna kendaraan. Hasil menunjukkan nilai rata-rata kepuasan sebesar

89,5% (kategori sangat baik) dengan indikator kemudahan penggunaan, kecepatan respon, dan kejelasan tampilan antarmuka.

Tabel 2 Hasil Kuesioner Uji Coba Aplikasi CarSecure

No	Indikator Penilaian	Rata-rata Skor	Kategori
1	Kemudahan penggunaan aplikasi	4.6	Sangat Baik
2	Kecepatan respon perintah (real-time)	4.5	Sangat Baik
3	Keakuratan tampilan lokasi kendaraan (GPS)	4.4	Baik
4	Desain dan tampilan antarmuka (UI/UX)	4.5	Sangat Baik
5	Stabilitas dan keandalan sistem	4.3	Baik

4.1.5. Operation And Maintenance

Tahap Operation and Maintenance merupakan fase akhir dalam pengembangan aplikasi CarSecure yang bertujuan untuk memastikan sistem berjalan sesuai dengan fungsinya serta dapat dioperasikan dengan stabil. Pada tahap ini dilakukan proses evaluasi performa sistem, evaluasi kepuasan pengguna, serta perbaikan bug minor berdasarkan hasil uji coba lapangan. Selain itu, tahap ini juga mencakup dokumentasi hasil dan rencana pengembangan lanjutan agar sistem dapat terus dikembangkan di masa mendatang.

a) Evaluasi Performa Sistem

Pengujian performa dilakukan untuk mengukur tingkat kestabilan, kecepatan respon, serta keakuratan komunikasi antara aplikasi Flutter dengan perangkat IoT (ESP32, GPS, dan relay). Pengujian ini dilakukan dalam tiga kondisi berbeda: jaringan stabil, jaringan lemah, dan kondisi mobile hotspot.

Tabel 4. 3 Hasil Evaluasi Performa Sistem

No	Aspek yang Diujii	Skenario Pengujian	Hasil Pengamatan	Rata-rata Nilai	Keterangan
1	Waktu respon perintah “Matikan Mesin”	Pengguna menekan tombol shutdown	2.5 detik	2–3 detik	Sangat Baik
2	Waktu kirim notifikasi asi peringatan	Saat kendaraan diakses tanpa izin	3.2 detik	<5 detik	Baik
3	Akurasi koordinat GPS	Perbandingan posisi di aplikasi dengan lokasi sebenarnya	Deviasi ±7 meter	6–10 meter	Baik
4	Stabilitas koneksi IoT-aplikasi	Durasi koneksi 30 menit nonstop	0 kali putus koneksi	100% stabil	Sangat Baik
5	Konsumsi daya ESP32	Pemantauan penggunaan arus rata-rata	80–120 mA	Normal	Baik

Secara keseluruhan, sistem menunjukkan performa yang stabil dan responsif. Waktu respon perintah berada di bawah 3 detik dengan tingkat akurasi GPS yang memadai untuk kebutuhan pelacakan kendaraan. Integrasi antara Flutter dan perangkat IoT berjalan lancar tanpa kehilangan koneksi selama pengujian.

b) Evaluasi Kepuasan Pengguna

Evaluasi kepuasan pengguna dilakukan melalui penyebaran kuesioner kepada 20 responden, yang sebelumnya telah mencoba langsung

aplikasi CarSecure. Pengujian ini menilai lima aspek utama yaitu kemudahan penggunaan, kecepatan respon, keakuratan data, tampilan antarmuka, dan keandalan sistem.

Tabel 3 Hasil Evaluasi Kepuasan Pengguna

No	Aspek Penilaian	Skor Rata-rata	Kategori
1	Kemudahan penggunaan aplikasi	4.6	Sangat Baik
2	Kecepatan respon sistem	4.5	Sangat Baik
3	Keakuratan tampilan lokasi kendaraan (GPS)	4.4	Baik
4	Desain dan tampilan antarmuka	4.5	Sangat Baik
5	Keandalan sistem secara keseluruhan	4.3	Baik

Mayoritas responden merasa puas dengan kinerja aplikasi CarSecure. Aspek yang paling menonjol adalah kemudahan penggunaan (92%) dan kecepatan respon perintah (89%), sedangkan aspek keakuratan GPS masih bisa ditingkatkan melalui optimasi modul dan kualitas sinyal. Secara umum, pengguna menilai bahwa aplikasi ini efisien, responsif, dan mudah digunakan untuk memantau serta mengontrol kendaraan secara real-time.

4.2. Pembahasan

Hasil penelitian menunjukkan bahwa aplikasi CarSecure berbasis Flutter berhasil diimplementasikan dengan baik sebagai sistem keamanan kendaraan berbasis Internet of Things (IoT). Aplikasi ini mampu terhubung secara real-time dengan perangkat ESP32, modul GPS, dan relay untuk memantau serta mengendalikan kendaraan dari jarak jauh. Melalui metode pengembangan Waterfall, setiap tahapan mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian telah dilakukan secara sistematis sehingga menghasilkan aplikasi yang stabil, efisien, dan

mudah digunakan. Pengujian fungsional menunjukkan seluruh fitur berjalan 100% sesuai rancangan, sedangkan pengujian performa menunjukkan waktu respon perintah rata-rata 2–3 detik dengan akurasi GPS 6–10 meter dan kestabilan koneksi mencapai 100%. Dari hasil kuesioner terhadap 20 responden, tingkat kepuasan pengguna mencapai rata-rata 89,2% (kategori sangat baik) dengan aspek tertinggi pada kemudahan penggunaan dan kecepatan respon sistem. Temuan ini menunjukkan bahwa framework Flutter tidak hanya unggul dalam performa teknis tetapi juga memberikan pengalaman pengguna yang optimal melalui desain antarmuka yang modern, responsif, dan intuitif. Jika dibandingkan dengan penelitian sebelumnya oleh(Marcos & Reza, 2021) yang menggunakan NodeMCU dan App Inventor, penelitian ini memberikan peningkatan signifikan baik dari segi performa perangkat keras maupun fleksibilitas antarmuka. Integrasi Flutter dengan ESP32 terbukti lebih efisien dalam komunikasi dua arah dan memberikan fungsionalitas tambahan seperti notifikasi real-time dan kontrol mesin jarak jauh. Dengan demikian, aplikasi CarSecure dapat menjadi model penerapan pengembangan sistem keamanan kendaraan modern berbasis IoT yang efisien, responsif, dan mudah diimplementasikan, sekaligus memberikan kontribusi nyata dalam bidang Informatika terutama pada pengembangan aplikasi mobile terintegrasi dengan sistem IoT.

5. KESIMPULAN

Berdasarkan hasil penelitian, implementasi, dan evaluasi yang telah dilakukan terhadap aplikasi CarSecure, dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi CarSecure berhasil dikembangkan menggunakan framework Flutter dengan integrasi Firebase Realtime Database sebagai media komunikasi antara aplikasi dan perangkat IoT (ESP32, GPS, dan relay). Pengembangan dilakukan secara sistematis menggunakan pendekatan model Waterfall yang meliputi tahap analisis, desain, implementasi, pengujian, serta evaluasi hasil.
2. Pemanfaatan Firebase Realtime Database terbukti efektif dalam meningkatkan efisiensi komunikasi dan kestabilan sistem. Proses sinkronisasi data antara aplikasi dan perangkat

IoT berjalan secara real-time tanpa jeda yang signifikan. Sistem mampu mempertahankan koneksi yang stabil selama pengujian dan menampilkan data yang akurat, baik pada kondisi jaringan stabil maupun jaringan lemah.

3. Hasil pengujian menunjukkan performa sistem dan tingkat kepuasan pengguna yang tinggi. Waktu respon perintah rata-rata berada di bawah tiga detik, dengan akurasi koordinat GPS mencapai deviasi ± 7 meter. Berdasarkan hasil kuesioner, aspek kemudahan penggunaan dan kecepatan respon memperoleh kategori “Sangat Baik”, dengan tingkat kepuasan pengguna mencapai 89,5%. Hal ini menunjukkan bahwa aplikasi CarSecure telah berfungsi sesuai tujuan penelitian, yakni membantu optimalisasi sistem keamanan kendaraan berbasis IoT secara efektif dan responsif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] F. Tamimi and S. Munawaroh, “Teknologi Sebagai Kegiatan Manusia Dalam Era Modern Kehidupan Masyarakat,” *Saturnus : Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 3, pp. 66–74, Jun. 2024, doi: 10.61132/saturnus.v2i3.157.
- [2] O. P. Rahman, “Perkembangan Teknologi Komputer dan Implikasinya terhadap Masyarakat,” vol. 1, no. 1, 2025, [Online]. Available: <https://www.techtarget.com>
- [3] H. Marcos and M. Reza, “Implementasi IoT Pada Rancang Bangun Aplikasi Mobile Sistem Keamanan Dan Pelacak Sepeda Motor,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 1, pp. 170–180, 2021, [Online]. Available: <http://jurnal.mdp.ac.id>
- [4] F. A. Aryatama and S. Samsugi, “Sistem Keamanan Kendaraan Bermotor Dengan ESP32 Menggunakan Kontrol Android,” *SMATIKA JURNAL*, vol. 14, no. 01, pp. 167–181, Jul. 2024, doi: 10.32664/smatika.v14i01.1267.
- [5] M. Juliarto, R. Amru Nityasa, and A. Dani Fajar Aditama, “Perancangan Keamanan Kendaraan Tanpa Kunci Dengan Menggunakan ESP32 dan Aplikasi BLYNK Berbasis IOT,” vol. 9, no. 1, pp. 47–53, 2024.

- [6] C. H. Putri and Mahmudin, "Mengenal Internet Of Things (IOT) Dan Perannya Dalam Jaringan Komputer," *Sindoro Cendikia Pendidikan*, 2025, doi: 10.9644/sindoro.v3i9.252.
- [7] Muslim, R. P. Sari, and S. Rahmayuda, "IMPLEMENTASI FRAMEWORK FLUTTER PADA SISTEM INFORMASI PERPUSTAKAAN MASJID (Studi Kasus: Masjid di Kota Pontianak)," *Jurnal Komputer dan Aplikasi*, 2022.
- [8] I. Rahmawati and D. P. Sari, "APLIKASI BERBASIS ANDROID MENGGUNAKAN FLUTTER FRAMEWORK UNTUK KEPERLUAN PERIZINAN TUGAS KELUAR PADA PT. XYZ," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 2, pp. 979–993, May 2024, doi: 10.29100/jipi.v9i2.5489.
- [9] J. M. Suhendro, M. Sudarma, and D. C. Khrisne, "RANCANG BANGUN APLIKASI SELULER PENYEDIA JASA PERAWATAN DAN KECANTIKAN MENGGUNAKAN FRAMEWORK FLUTTER," *Jurnal SPEKTRUM*, vol. 8, no. 2, 2021.
- [10] I. Nyoman, B. Hartawan, and W. Sudiarsa, "ANALISIS KINERJA INTERNET OF THINGS BERBASIS FIREBASE REAL-TIME DATABASE," | *Jurnal RESISTOR JURNAL RESISTOR*, vol. 1, no. 1, 2020, [Online]. Available: <http://jurnal.stiki-indonesia.ac.id/index.php/jurnalresistor>
- [11] M. Roziqin, A. F. Kabes, and S. Z. N. Haq, "RANCANG BANGUN PENYIRAMAN TANAMAN CABE OTOMATIS MENGGUNAKAN ESP32 DAN SENSOR KELEMBAPAN TANAH," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.7966.
- [12] M. Nizam, H. Yuana, and Z. Wulansari, "MIKROKONTROLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB," 2022.
- [13] M. R. L. Nuha, A. B. Yunanda, M. Sidqon, and A. Kridoyono, "RANCANG BANGUN SISTEM KEAMANAN SEPEDA MOTOR DENGAN ENGINE CUT OFF, GPS, DAN KENDALI NFC SMARTPHONE," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.7589.
- [14] Muhammad Muzammil Arrozak, Antonius Edy Kristiyono, and Prihastono Prihastono, "Rancang Bangun GPS Tracker di Kapal dengan Menggunakan Internet of Things (IoT)," *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro dan Informatika*, vol. 3, no. 3, pp. 22–31, May 2025, doi: 10.61132/jupiter.v3i3.822.
- [15] A. R. Ismail, "Penerapan Metode Agile Pada Perancangan Sistem Informasi Pengajuan Nomor Surat di Pemerintahan Desa," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 7, no. 2, pp. 284–289, Apr. 2025, doi: 10.47233/jtekstis.v7i2.1927.