

BAB III

METODE PENELITIAN

3.1 Model Penelitian

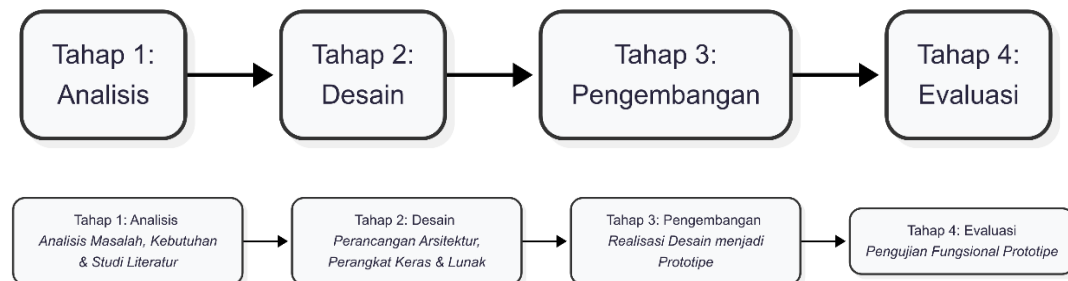
Penelitian ini menggunakan metode Penelitian Desain dan Pengembangan (*Design and Development*) untuk merancang dan mengembangkan prototipe sistem keamanan sepeda motor berbasis *Internet of Things* (IoT). Menurut Richey dan Klein (2007), metode ini didefinisikan sebagai "studi sistematis tentang proses desain, pengembangan, dan evaluasi dengan tujuan untuk membangun landasan empiris bagi penciptaan produk dan perangkat instruksional maupun non-instruksional". Pendekatan ini relevan karena berfokus pada penciptaan solusi praktis melalui pengembangan produk yang fungsional dan teruji.

Secara spesifik, penelitian ini termasuk dalam kategori penelitian produk dan perangkat (*product and tool research*), yang berpusat pada proses penciptaan produk dari awal hingga akhir. Proses ini mencakup empat tahapan utama yaitu:

- 1) Analisis, yang meliputi studi permasalahan, analisis kebutuhan, dan studi literatur.
- 2) Desain, yang mencakup perancangan arsitektur, perangkat keras, dan perangkat lunak.
- 3) Pengembangan, yaitu tahap produksi di mana spesifikasi desain diwujudkan dalam bentuk prototipe fisik.
- 4) Evaluasi, di mana prototipe diuji secara fungsional untuk mengukur kinerjanya.

Alur penelitian ini dirancang secara sistematis melalui empat tahapan utama. Proses dimulai dengan tahap analisis untuk mengidentifikasi permasalahan, menganalisis kebutuhan, dan melakukan studi literatur. Hasil dari analisis tersebut kemudian menjadi landasan pada tahap desain untuk merancang arsitektur sistem, perangkat keras, dan perangkat lunak. Selanjutnya, rancangan tersebut diimplementasikan menjadi prototipe fungsional pada tahap pengembangan. Penelitian ini diakhiri dengan tahap evaluasi, di mana dilakukan pengujian fungsional terhadap prototipe yang telah dibuat untuk memastikan kinerjanya

sesuai dengan tujuan penelitian (Gambar 3.1). Pendekatan sistematis ini sejalan dengan praktik penelitian serupa yang bertujuan menghasilkan dan memvalidasi prototipe teknologi baru, seperti yang dilakukan oleh Wahid & Budiarmo (2021). (Richey & Klein, 2007)



Gambar 3.1 Diagram Alur Penelitian

3.2 Prosedur Penelitian

Prosedur penelitian ini dijalankan secara berurutan mengikuti empat tahapan model D&D yang telah dirumuskan.

3.2.1 Tahap 1: Analisis Kebutuhan

Pada tahap ini, analisis dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem keamanan sepeda motor. Analisis ini mencakup pemahaman tentang kebutuhan pengguna untuk keamanan kendaraan yang dapat melacak posisi kendaraan secara *real-time* dan mengautentikasi pengguna melalui NFC dan BLE. Data dari perangkat GPS, NFC, dan status *relay* akan dikumpulkan untuk mendukung pengambilan keputusan sistem. Dari analisis ini, diperoleh bahwa sistem harus mampu menangani autentikasi berbasis jarak (BLE) dan autentikasi aktif (NFC), serta melacak lokasi kendaraan secara *real-time* melalui GPS dan Firebase.

Untuk antarmuka pengguna, dasbor web yang dirancang harus mampu menyajikan data pemantauan secara intuitif. Ini mencakup visualisasi lokasi *real-time* pada peta interaktif serta panel status yang menampilkan informasi vital perangkat seperti persentase baterai, kekuatan sinyal, dan status *relay*. Lebih dari sekadar pemantauan langsung, dasbor juga harus menyediakan fitur manajemen data historis yang memungkinkan pengguna untuk melihat seluruh riwayat

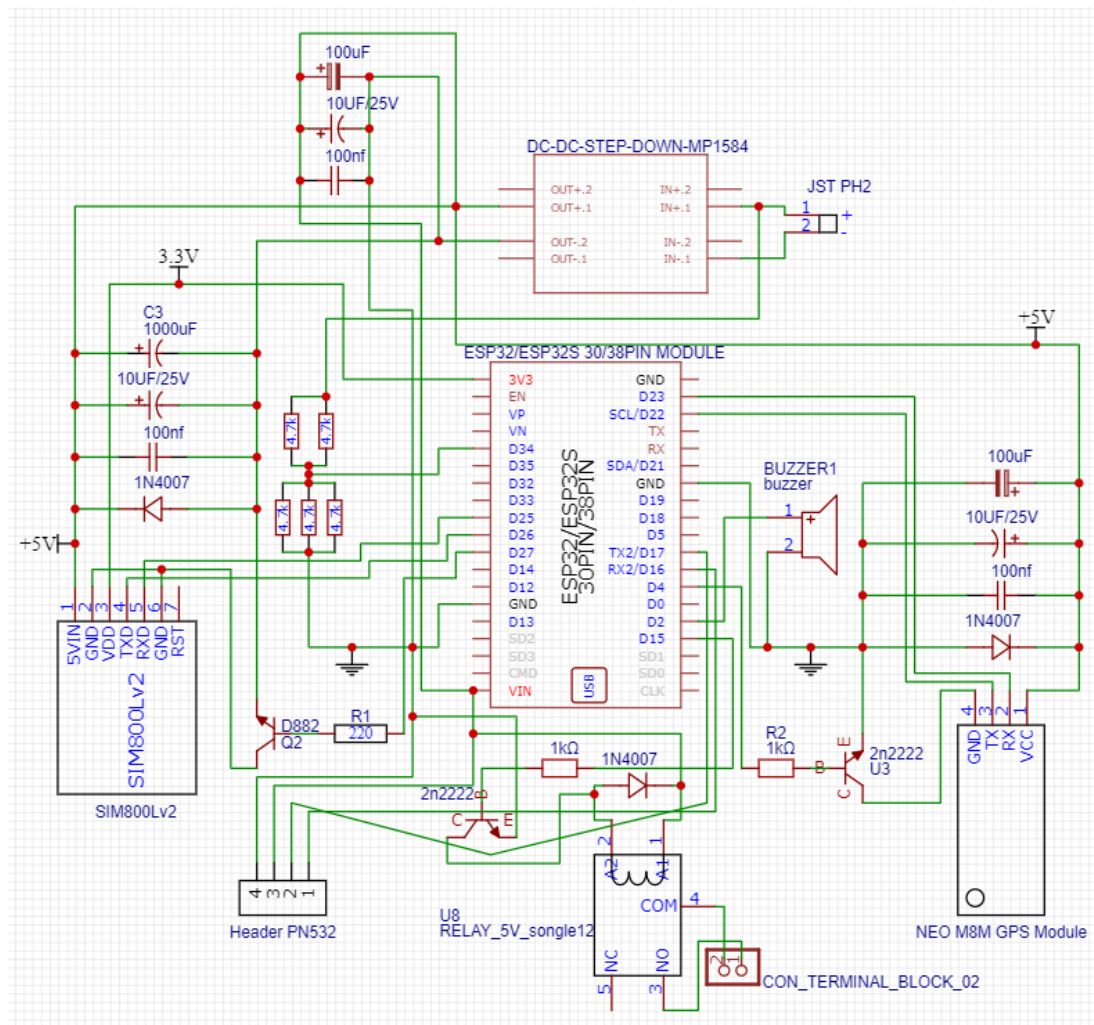
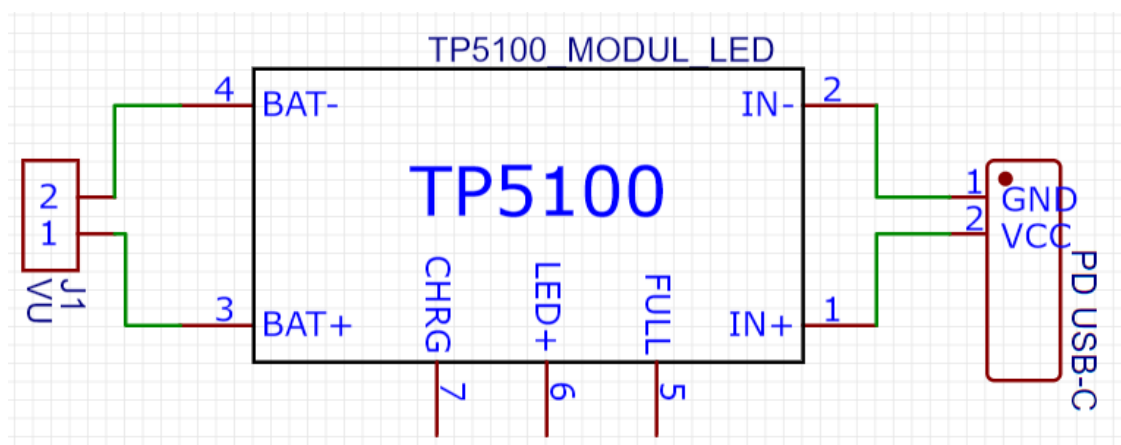
perjalanan, memfilternya berdasarkan rentang tanggal, melakukan pencarian, hingga mengekspor data tersebut ke format CSV, JSON, atau Excel. Untuk menjamin keamanan data, seluruh akses ke dasbor ini wajib dilindungi oleh sebuah sistem *login* pengguna yang aman.

3.2.2 Tahap 2: Perancangan Sistem (*Design*)

Setelah kebutuhan teridentifikasi, tahap berikutnya adalah merancang sistem, termasuk perangkat keras dan perangkat lunak. Prototipe sistem menggunakan ESP32 sebagai mikrokontroler utama dengan kemampuan *Wi-Fi* dan *BLE*. Modul GPS NEO-M8L digunakan untuk melacak posisi kendaraan, PN532 digunakan untuk membaca NFC pada e-KTP sebagai autentikasi, dan Modul GSM SIM800Lv2 untuk komunikasi dengan server *endpoint* melalui protokol HTTP. Firebase digunakan sebagai platform *backend* untuk menyimpan data kendaraan secara *real-time*. Diagram blok, skematik, dan struktur *database* Firebase dikembangkan pada tahap ini untuk memastikan integrasi antara komponen-komponen tersebut. Detail perancangan alat sebagai berikut:

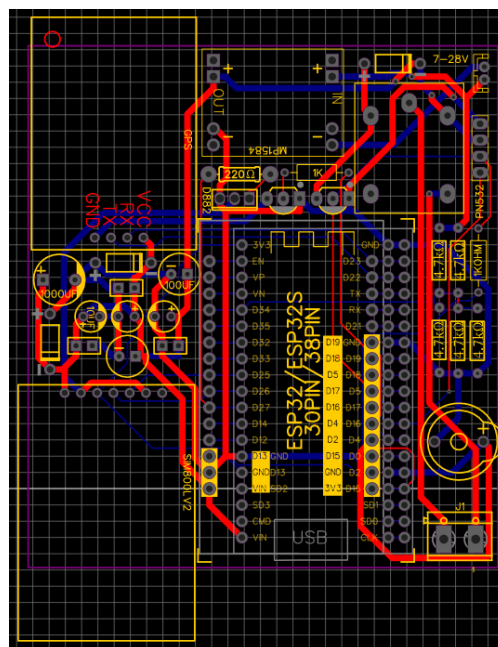
3.2.2.1 Rancangan Skematik Elektronik

Rancangan logis dari sirkuit elektronik dibuat menggunakan perangkat lunak *Electronic Design Automation* (EDA). Hasilnya adalah sebuah diagram skematik yang mendefinisikan setiap komponen dan interkoneksinya secara detail. Gambar 3.2 menampilkan rancangan skematik untuk sirkuit utama (*main board*) dari prototipe. Skematik ini berfungsi sebagai cetak biru logis yang mendefinisikan hubungan elektrikal antar semua komponen inti, termasuk mikrokontroler ESP32, modul SIM800L, modul GPS NEO M8L, modul NFC PN532, serta sirkuit aktuator seperti *relay* dan *buzzer*. Sementara itu, Gambar 3.3 menunjukkan skematik untuk sirkuit pengisi daya (*charger*) terpisah yang menggunakan modul TP5100 untuk manajemen pengisian baterai Li-ion secara aman.

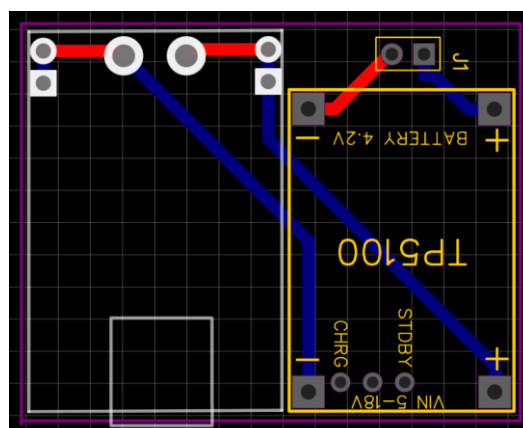
Gambar 3.2 Rancangan Skematik *MainBoard*Gambar 3.3 Rancangan Skematik *Charger*

3.2.2.2 Rancangan Tata Letak PCB (*Printed Circuit Board*)

Berdasarkan rancangan skematik, dibuatlah desain tata letak Papan Sirkuit Cetak (*Printed Circuit Board* - PCB) untuk merealisasikan perangkat keras secara fisik. Gambar 3.4 menunjukkan desain tata letak untuk *main board*, di mana posisi setiap komponen dan jalur tembaga diatur sedemikian rupa untuk mencapai ukuran yang ringkas dan koneksi yang andal. Gambar 3.5 menampilkan desain PCB untuk modul pengisi daya.



Gambar 3.4 Rancangan Tata Letak PCB *MainBoard*

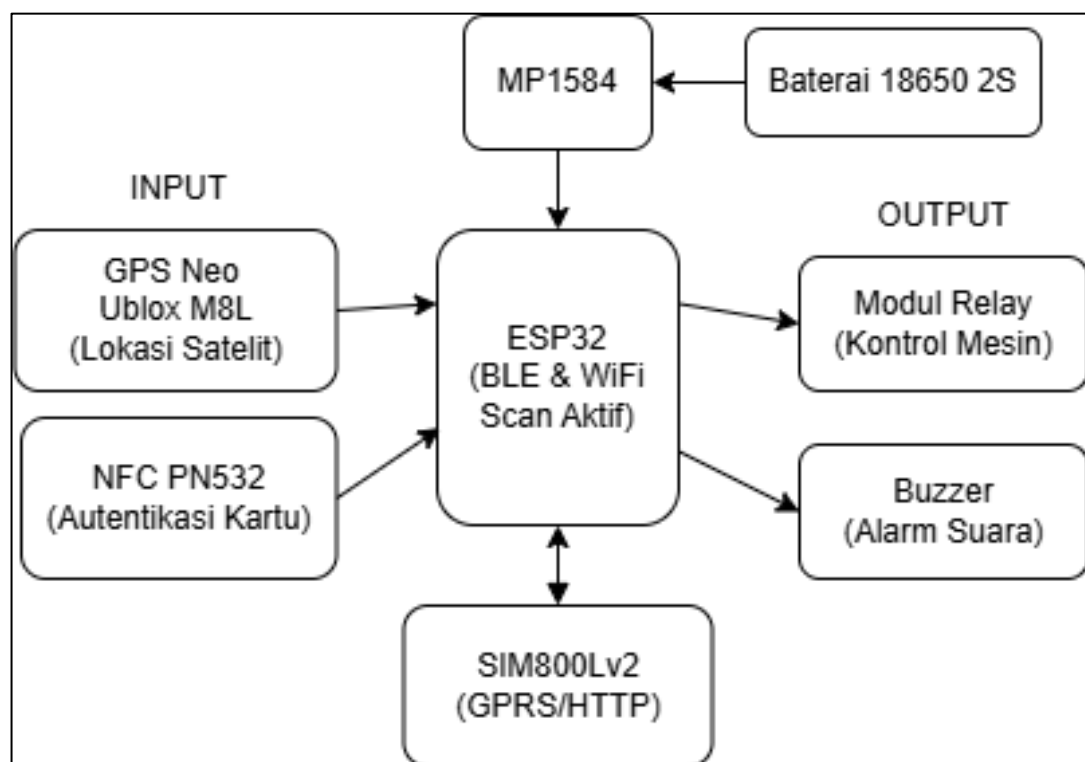


Gambar 3.5 Rancangan Tata Letak PCB *Charger*

3.2.2.3 Blok Diagram Sistem

Blok diagram sistem adalah representasi grafis yang menunjukkan hubungan antar komponen utama dalam sistem *GPS Tracker*. Diagram ini memberikan gambaran umum tentang bagaimana setiap komponen, seperti modul GPS, modul GSM/GPRS, mikrokontroler, dan sumber daya, berinteraksi satu sama lain.

Untuk memberikan gambaran fungsional secara keseluruhan, arsitektur perangkat keras disajikan dalam bentuk diagram blok pada Gambar 3.6. Diagram ini mengelompokkan komponen berdasarkan fungsinya seperti *Input*, *Output*, Unit Pemroses, dan Komunikasi untuk mempermudah pemahaman tentang bagaimana setiap bagian sistem berinteraksi satu sama lain.

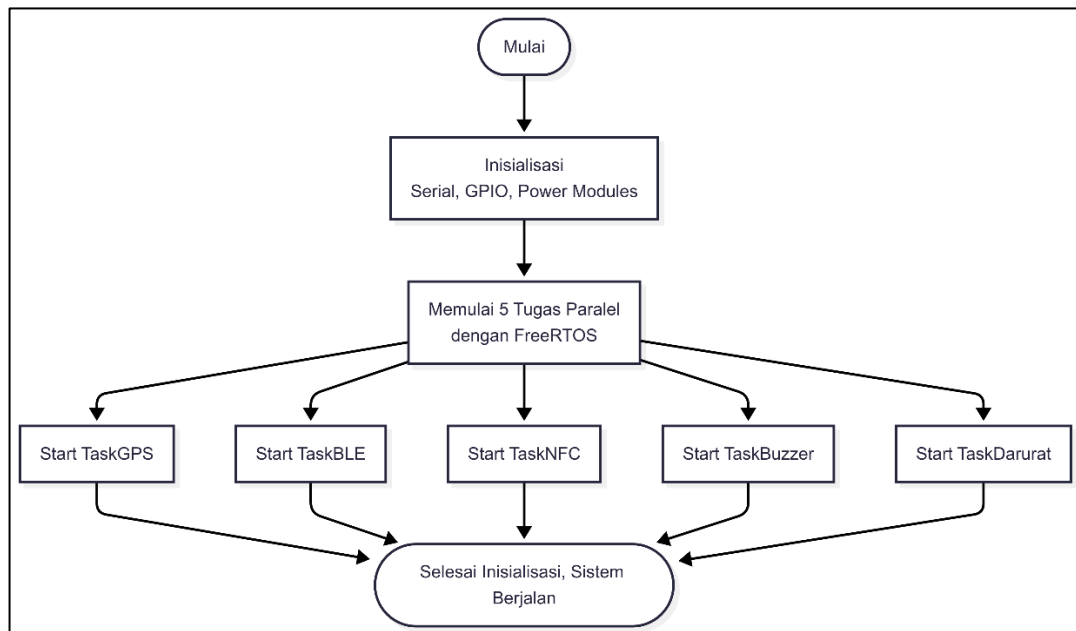


Gambar 3.6 Blok Diagram Sistem

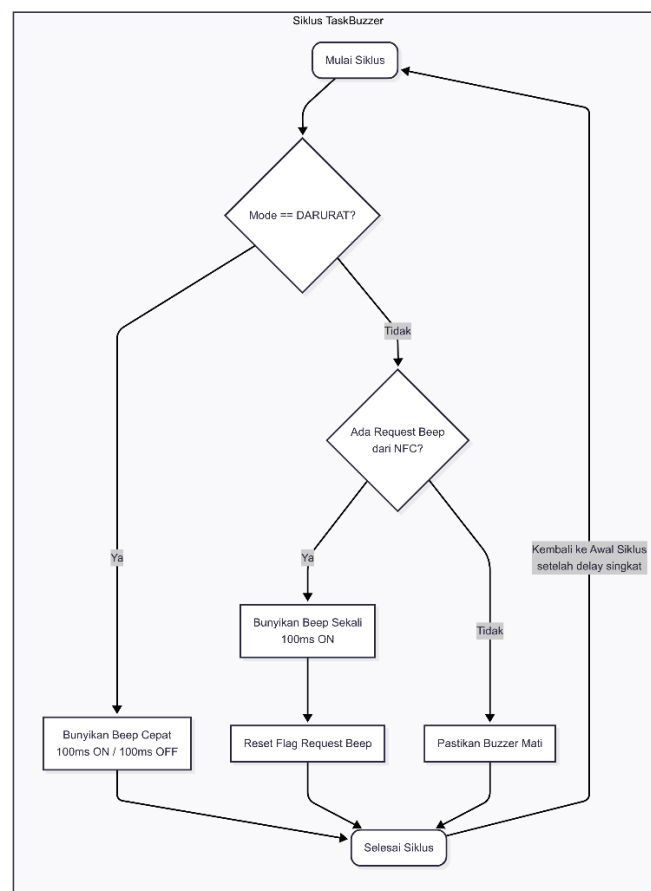
3.2.2.4 Diagram Alur Sistem/*Flowchart*

Diagram alur yang memvisualisasikan logika kerja utama pada *firmware* perangkat, mulai dari proses inisialisasi, manajemen mode, hingga pengiriman data periodik. Logika kerja pada *firmware* perangkat dirancang secara *multi-tasking* menggunakan FreeRTOS. Untuk memvisualisasikan alur kerja dari setiap tugas yang berjalan secara paralel, dirancang serangkaian diagram alur (*flowchart*) berikut:

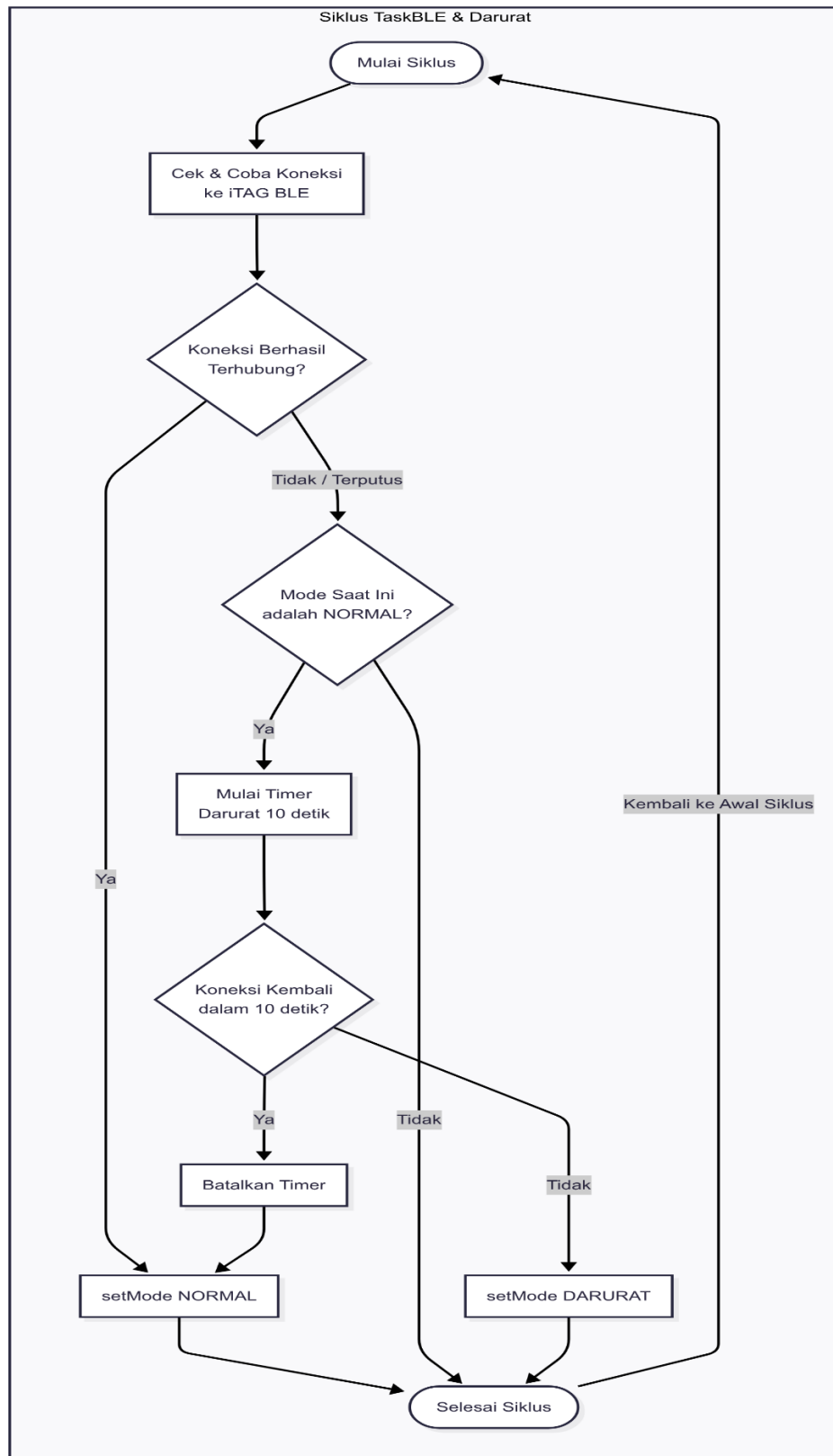
- a) Gambar 3.7 menunjukkan proses Inisialisasi Sistem yang berjalan satu kali saat perangkat dinyalakan untuk memulai semua tugas.
- b) Gambar 3.8 merinci alur kerja TaskGPS, yang merupakan *loop* utama untuk pengumpulan data sensor, pengiriman data ke *cloud*, dan logika pemeriksaan *geofencing*.
- c) Gambar 3.9 menggambarkan logika TaskNFC, yang secara terus-menerus memindai kartu dan memicu perubahan mode antara NORMAL dan PARKIR.
- d) Gambar 3.10 menjelaskan alur kerja gabungan dari TaskBLE dan TaskDarurat, yang bertanggung jawab memonitor koneksi iTAG dan memicu mode DARURAT jika terjadi pemutusan koneksi.
- e) Gambar 3.11 menunjukkan logika reaktif dari TaskBuzzer yang mengontrol alarm suara sesuai dengan mode sistem saat ini.



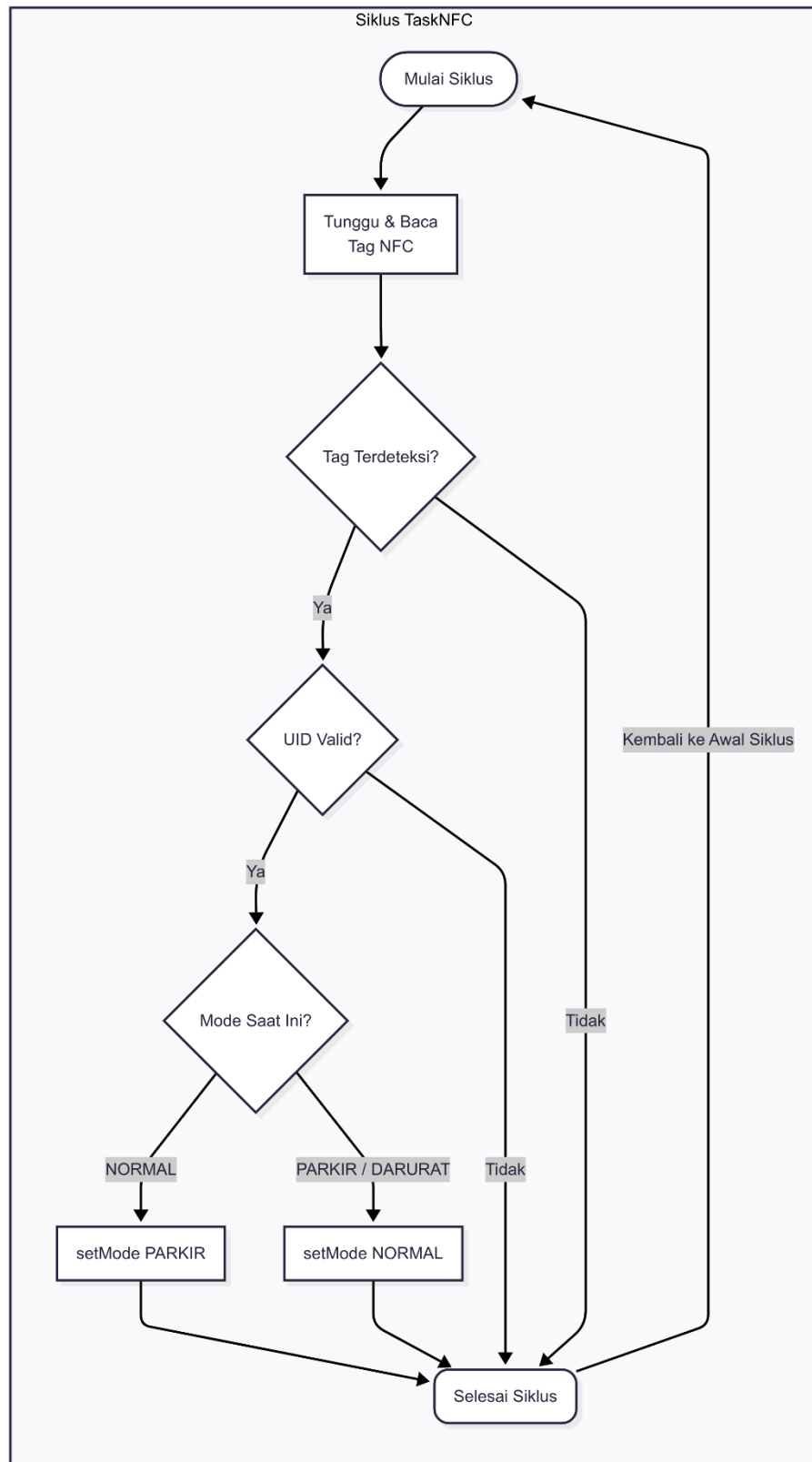
Gambar 3.7 *Flowchart* Inisialisasi FreeRTOS



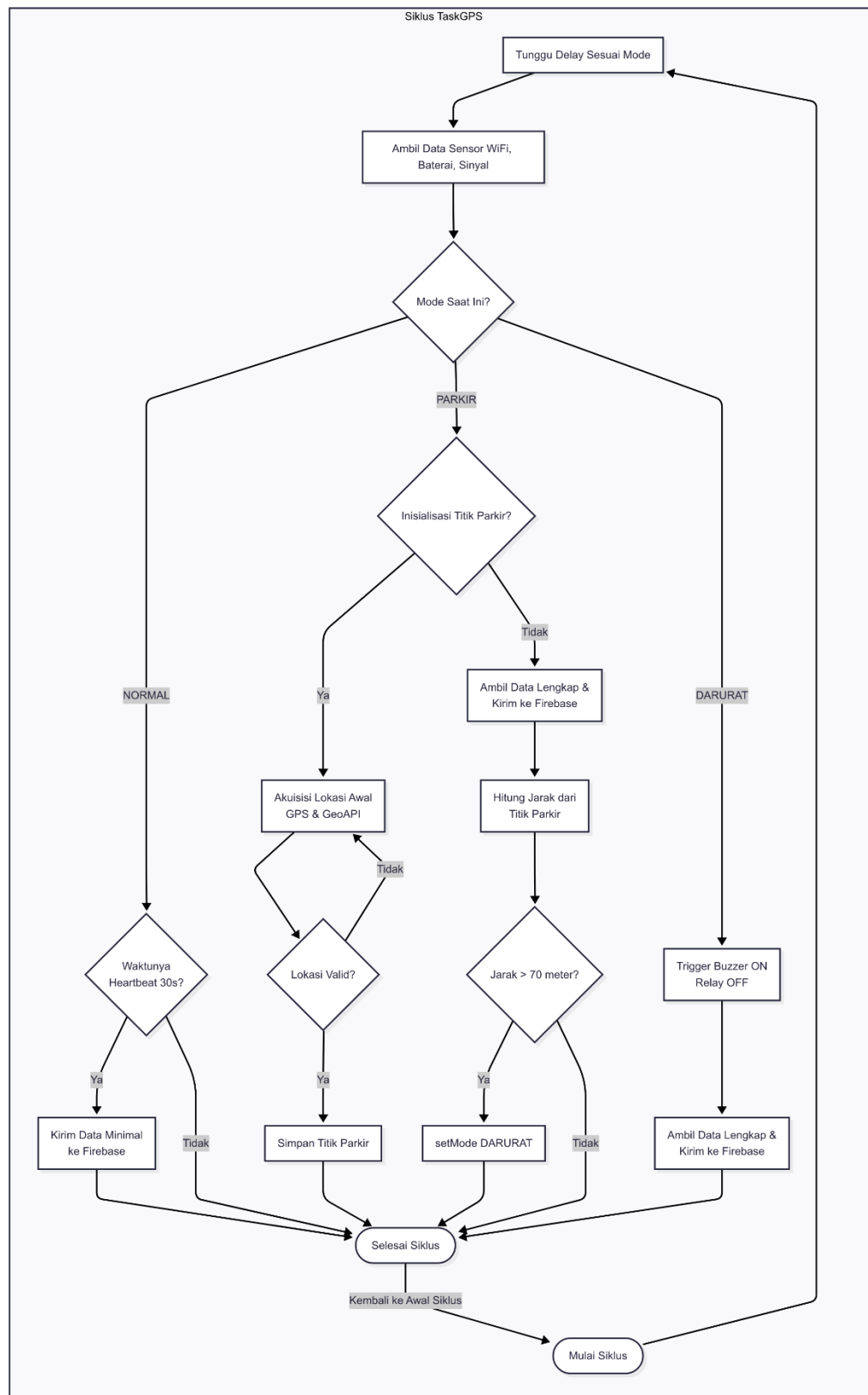
Gambar 3.8 *Flowchart* TaskBuzzer



Gambar 3.9 Flowchart TaskBLE dan Darurat



Gambar 3.10 Flowchart TaskNFC



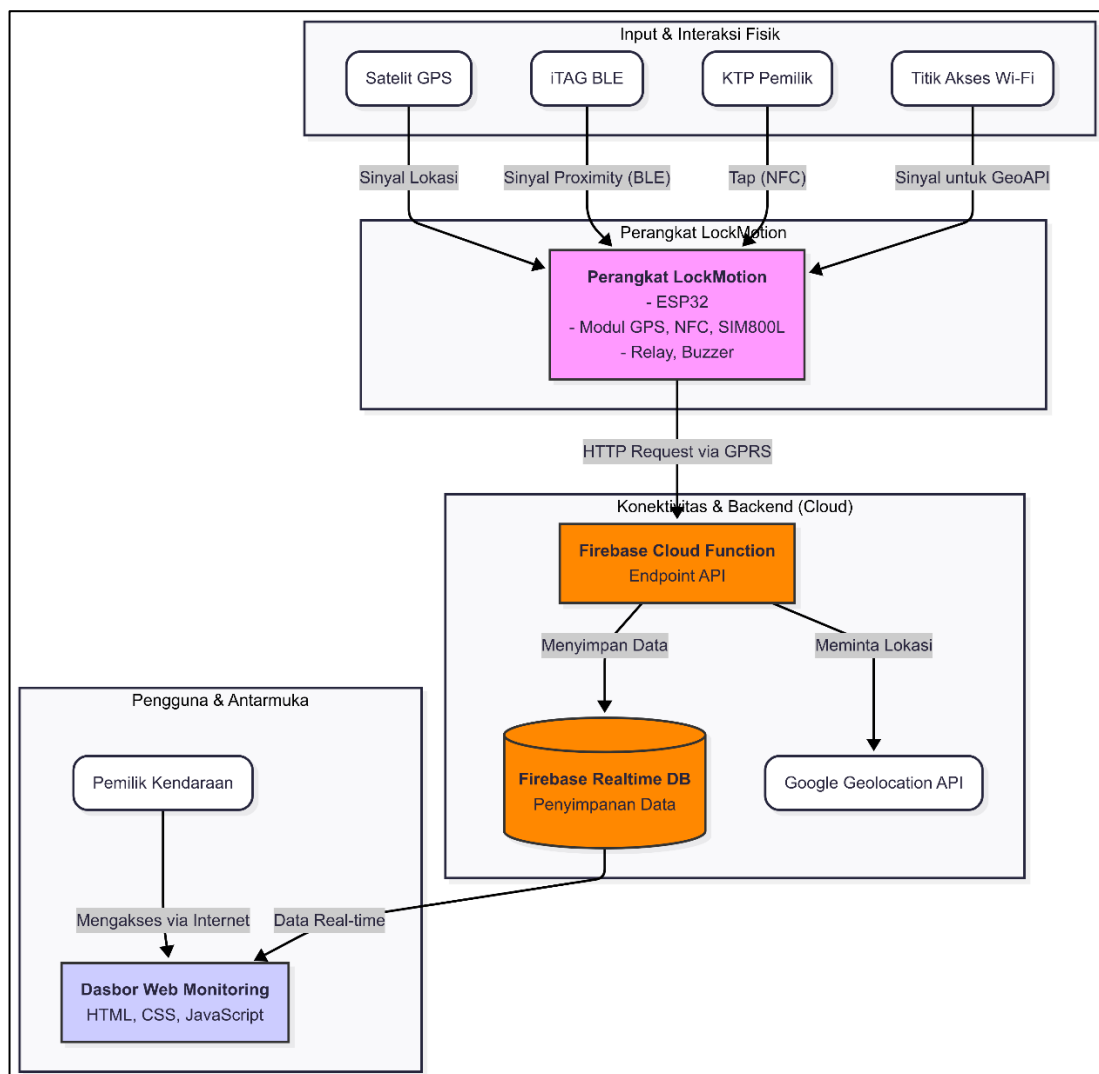
Gambar 3.11 Flowchart TaskGPS (Main Loop)

3.2.2.5 Arsitektur Sistem

Arsitektur sistem secara keseluruhan, yang diilustrasikan pada Gambar 3.12, menggambarkan ekosistem *end-to-end* dari proyek LockMotion, mulai dari interaksi fisik hingga penyajian data kepada pengguna. Arsitektur ini terdiri dari empat lapisan utama yang saling berkomunikasi:

- a) Lapisan Input & Interaksi Fisik: Lapisan ini merupakan sumber data dan pemicu awal dari sistem. Satelit GPS menyediakan sinyal lokasi mentah, iTAG BLE memberikan sinyal proksimitas untuk autentikasi implisit, KTP Pemilik berfungsi sebagai pemicu autentikasi eksplisit melalui tap NFC, dan Titik Akses Wi-Fi di sekitar menyediakan data lingkungan untuk sistem pelacakan sekunder.
- b) Lapisan Perangkat (*On-Vehicle*): Perangkat LockMotion yang terpasang di sepeda motor bertindak sebagai *edge node* yang mengagregasi semua *input*. Ditenagai oleh mikrokontroler ESP32, perangkat ini membaca data dari modul GPS dan NFC, memonitor status koneksi BLE, dan mengolahnya menjadi sebuah paket data berformat JSON.
- c) Lapisan Konektivitas & *Backend (Cloud)*: Perangkat mengirimkan paket data JSON melalui jaringan GPRS menggunakan protokol HTTP POST *Request*. Permintaan ini diterima oleh *endpoint API* yang berjalan pada Firebase Cloud Functions. *Cloud Function* ini kemudian melakukan dua tugas utama: (1) menyimpan data yang diterima ke Firebase Realtime Database, dan (2) jika diperlukan, mengirim data pemindaian Wi-Fi ke Google Geolocation API untuk mendapatkan data lokasi sekunder. Firebase Realtime Database berfungsi sebagai *single source of truth* untuk seluruh sistem.
- d) Lapisan Pengguna & Antarmuka: Pengguna (Pemilik Kendaraan) berinteraksi dengan sistem melalui Dasbor Web Monitoring. Aplikasi web ini terhubung langsung ke Firebase Realtime Database dan mendengarkan perubahan data secara *real-time*. Ketika data baru masuk ke *database*, dasbor akan secara otomatis memperbarui tampilan peta dan status tanpa perlu *refresh*, memberikan pengalaman pemantauan yang *live* dan interaktif.

Secara keseluruhan, arsitektur ini mendemonstrasikan sebuah sistem IoT yang terdistribusi, di mana pemrosesan data terjadi baik di tingkat perangkat (*edge*) maupun di *cloud*, untuk menghasilkan sebuah solusi keamanan yang responsif dan andal.



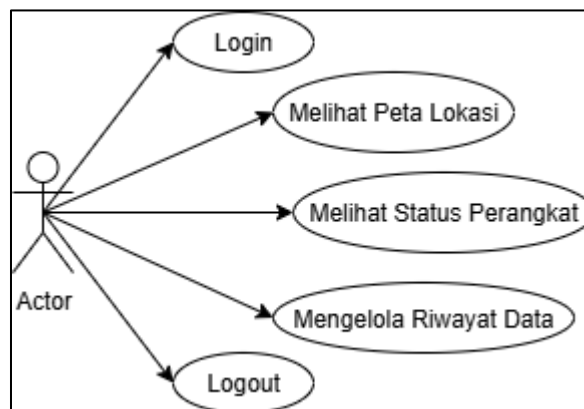
Gambar 3.12 Arsitektur Sistem

3.2.2.6 Perancangan Aplikasi Web

Perancangan aplikasi web sebagai antarmuka pengguna difokuskan pada fungsionalitas, interaksi, dan alur data. Untuk memodelkan arsitektur perangkat lunak pada sisi klien, digunakan beberapa diagram UML (Unified Modeling Language) yang masing-masing menjelaskan aspek perancangan yang berbeda.

1) Use Case Diagram

Untuk memodelkan interaksi fungsional antara pengguna dan sistem, digunakan *Use Case Diagram* seperti yang disajikan pada Gambar 3.13. Diagram ini mengidentifikasi satu aktor utama, yaitu Pemilik Kendaraan (*Actor*), yang berinteraksi dengan sistem melalui serangkaian kasus penggunaan esensial. Fungsionalitas inti mencakup kemampuan untuk *Login* dan *Logout* guna mengelola sesi akses yang aman. Selama sesi aktif, pengguna dapat Melihat Peta Lokasi untuk memvisualisasikan data geografis, Melihat Status Perangkat untuk memantau secara *real-time*, serta Mengelola Riwayat Data, yang memberinya kemampuan untuk melihat, memfilter, mencari, dan mengeksport data historis perjalanan.



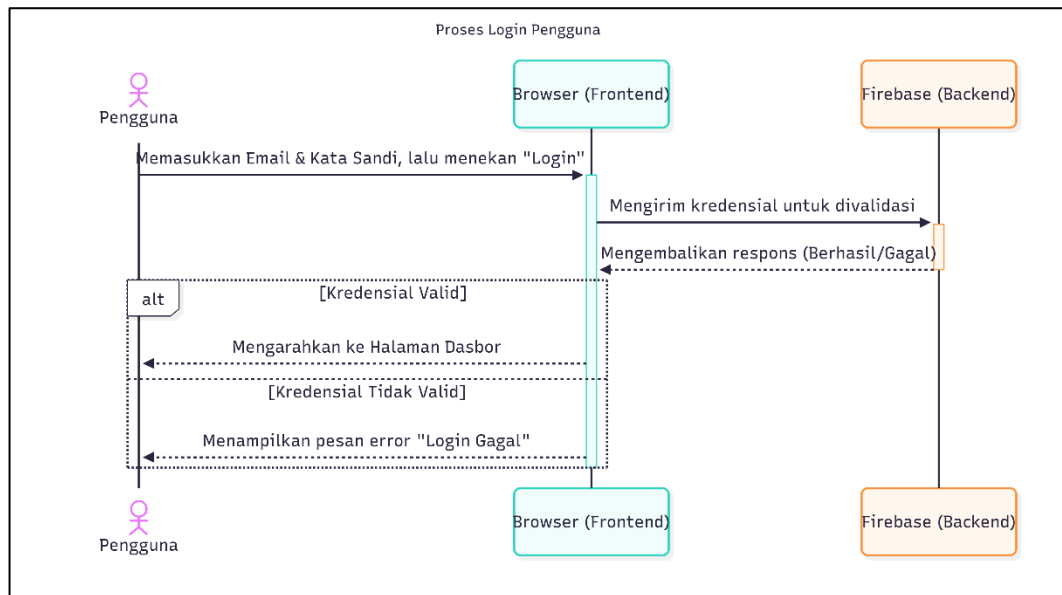
Gambar 3.13 Use Case Diagram Dasbor Web

2) Sequence Diagram

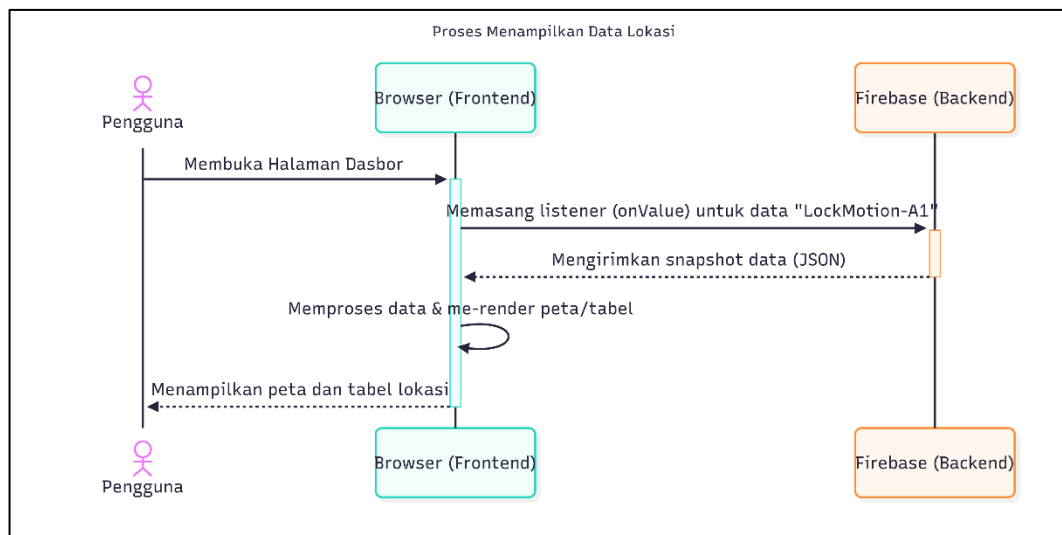
Untuk menggambarkan alur interaksi dinamis dan pertukaran pesan antar komponen sistem secara kronologis, digunakan *Sequence Diagram*. Diagram ini secara efektif memvisualisasikan komunikasi antara pengguna, aplikasi *frontend*, dan layanan *backend* Firebase.

Gambar 3.14 mengilustrasikan urutan proses autentikasi pengguna. Alur dimulai saat pengguna memasukkan kredensial, yang kemudian dikirim oleh *frontend* ke Firebase untuk validasi. Berdasarkan respons dari *backend*, sistem akan memberikan akses ke dasbor atau menampilkan pesan kesalahan. Sementara itu, Gambar 3.15 merinci proses menampilkan data lokasi. Setelah pengguna membuka

dasbor, aplikasi *frontend* secara otomatis memasang listener ke Firebase Realtime Database. Setiap kali ada data baru, Firebase akan mengirimkannya ke *frontend*, yang kemudian memproses dan *me-render* informasi tersebut pada peta dan tabel untuk ditampilkan kepada pengguna.



Gambar 3.14 Sequence Diagram Login Web



Gambar 3.15 Sequence Diagram Dashboard Web

3.2.3 Tahap 3: Pengembangan dan Implementasi (*Development*)

Berdasarkan hasil perancangan, prototipe sistem direalisasikan menjadi

produk yang fungsional. Kegiatan pada tahap ini mencakup:

3.2.3.1 Perakitan Perangkat Keras/*Hardware*

Tahap pengembangan perangkat keras adalah proses merealisasikan rancangan skematik dan tata letak PCB yang telah final pada tahap desain (Sub-bab 3.2.2) menjadi sebuah prototipe fisik. Proses ini direncanakan secara sistematis untuk memastikan semua komponen terpasang dengan benar dan berfungsi sesuai rancangan.

Kegiatan utama dalam tahap ini adalah proses *soldering* komponen elektronik ke Papan Sirkuit Cetak (PCB) yang telah dirancang. Langkah-langkah perakitan yang direncanakan adalah sebagai berikut:

- a) Pemasangan Komponen Pasif: Proses dimulai dengan menyolder komponen-komponen dasar seperti resistor, kapasitor, dan dioda ke papan sirkuit utama (*main board*) dan papan sirkuit pengisi daya (*charger*).
- b) Pemasangan Modul Utama: Setelah komponen pasif terpasang, dilanjutkan dengan pemasangan modul-modul inti. Ini mencakup pemasangan soket untuk mikrokontroler ESP32, serta pemasangan modul SIM800L, modul GPS NEO-M8L, dan modul NFC PN532 pada pin yang telah ditentukan dalam rancangan skematik.
- c) Pemasangan Aktuator dan Sirkuit Daya: Langkah terakhir adalah mengintegrasikan komponen aktuator seperti relay dan buzzer, serta komponen sirkuit daya termasuk modul *step-down* dan terminal untuk koneksi baterai.

Seluruh proses perakitan ini dilakukan dengan mengacu pada Gambar 3.2 (Rancangan Skematik *MainBoard*) dan Gambar 3.4 (Rancangan Tata Letak PCB *MainBoard*) untuk memastikan setiap koneksi elektrik akurat dan posisi komponen sesuai dengan desain. Hasil akhir dari tahap perakitan ini adalah sebuah unit prototipe perangkat keras yang utuh dan siap untuk tahap selanjutnya, yaitu pemrograman *firmware*. Wujud fisik dari prototipe yang telah selesai dirakit akan disajikan pada Bab IV sebagai bagian dari hasil penelitian.

3.2.3.2 Pemrograman *Firmware*

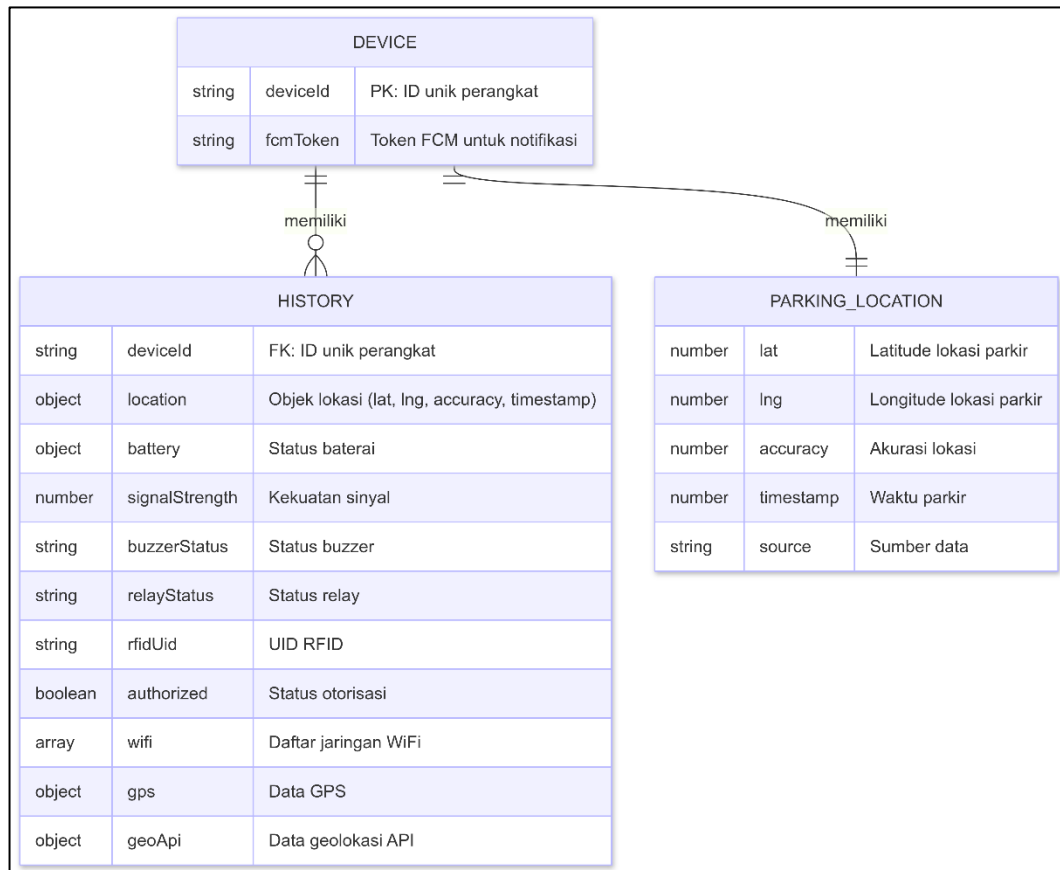
Firmware untuk prototipe dikembangkan menggunakan bahasa C++ pada platform Arduino IDE. Logika program dirancang secara *multi-tasking* dengan memanfaatkan sistem operasi *real-time* (RTOS) untuk menangani lima tugas secara paralel: TaskGPS, TaskBLE, TaskDarurat, TaskNFC, dan TaskBuzzer. Logika utama dari *firmware* ini adalah sebuah mesin keadaan (*state machine*) yang mengelola tiga mode operasi: NORMAL, PARKIR, dan DARURAT. Seluruh logika pemrograman bisa dilihat pada Gambar 3.7 sampai 3.11.

3.2.3.3 Pengembangan *Backend*

Backend untuk proyek LockMotion dikembangkan menggunakan pendekatan *serverless* melalui platform Google Firebase. Sebuah Cloud Function ditulis dalam bahasa JavaScript (Node.js) untuk menciptakan *endpoint* API yang menerima permintaan HTTP POST dari perangkat keras. Metode ini dipilih karena skalabilitasnya yang tinggi dan tidak memerlukan manajemen server manual. Inti dari *backend* ini adalah struktur data yang terorganisir dengan baik di dalam Firebase Realtime Database, yang dirancang untuk menyimpan semua informasi relevan secara efisien dan *real-time*.

Struktur data ini, seperti yang diilustrasikan pada Gambar 3.16, terdiri dari tiga entitas utama: DEVICE, HISTORY, dan PARKING_LOCATION. Entitas DEVICE berfungsi untuk mendaftarkan setiap perangkat keras LockMotion yang ada, dengan *deviceId* sebagai *Primary Key* (PK) unik untuk setiap perangkat dan *fcmToken* untuk mengelola notifikasi *push*. Entitas ini memiliki hubungan "satu-ke-banyak" dengan entitas HISTORY dan PARKING_LOCATION. Ini berarti setiap satu perangkat (DEVICE) dapat memiliki banyak catatan riwayat (HISTORY) dan satu catatan lokasi parkir (PARKING_LOCATION). Entitas HISTORY adalah pusat pencatatan data operasional dari perangkat. Setiap catatan dihubungkan ke perangkat spesifik melalui *deviceId* yang berfungsi sebagai *Foreign Key* (FK). Entitas ini menyimpan data yang sangat kaya, termasuk objek *location* (berisi *latitude*, *longitude*, akurasi, dan *timestamp*), status *battery* dan *signalStrength*, status *buzzer* dan *relay*, *rfidUid* dan status *authorized* untuk autentikasi, serta data mentah dari *gps*, *geoApi*, dan pemindaian *wifi* di sekitar. Sementara itu, entitas PARKING_LOCATION secara khusus menyimpan data titik

parkir terakhir, yang mencakup *lat* (*Latitude*), *lng* (*Longitude*), *accuracy* (akurasi lokasi), *timestamp* (waktu parkir), dan *source* (sumber data lokasi, misalnya dari GPS atau GeoAPI). Struktur yang terperinci ini memungkinkan sistem untuk tidak hanya melacak lokasi secara *real-time* tetapi juga menganalisis riwayat perjalanan dan status perangkat secara komprehensif.



Gambar 3.16 Struktur Data RTDB Firebase

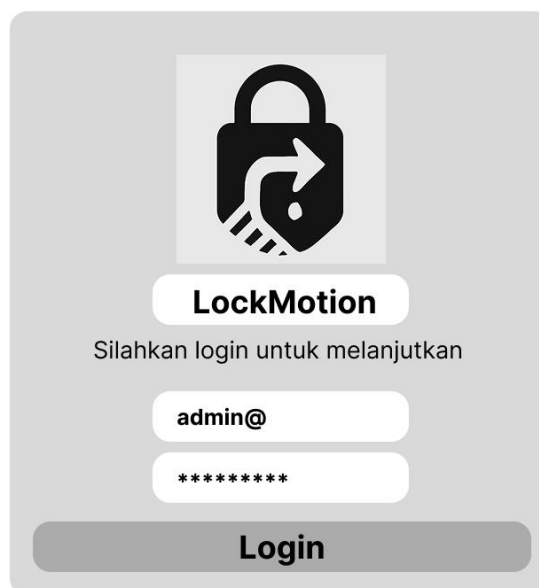
3.2.3.4 Pengembangan *Frontend*

Antarmuka pengguna (dasbor) dikembangkan sebagai aplikasi web dinamis menggunakan teknologi standar web: HTML untuk struktur konten, CSS untuk penataan visual dan pengalaman pengguna yang responsif, serta JavaScript sebagai mesin utama untuk logika aplikasi. Pendekatan ini dipilih karena universalitasnya, di mana dasbor dapat diakses dari berbagai perangkat, baik desktop maupun *mobile*, melalui peramban web tanpa memerlukan instalasi aplikasi khusus. *Website* yang dibuat akan memanfaatkan koneksi *real-time* ke Firebase, memastikan bahwa

setiap data baru yang dikirim oleh perangkat LockMotion akan langsung ditampilkan di dasbor tanpa perlu pengguna memuat ulang halaman secara manual.

Untuk memastikan keamanan akses, *frontend* akan dilengkapi dengan halaman *login* terlebih dahulu, seperti yang divisualisasikan pada *mockup* di Gambar 3.17. Halaman ini akan terhubung dengan layanan Firebase Authentication, di mana hanya pengguna dengan kredensial (email dan sandi) yang terdaftar yang dapat melanjutkan ke dasbor utama. Desainnya dibuat minimalis untuk fokus pada fungsionalitas inti, yaitu keamanan akses.

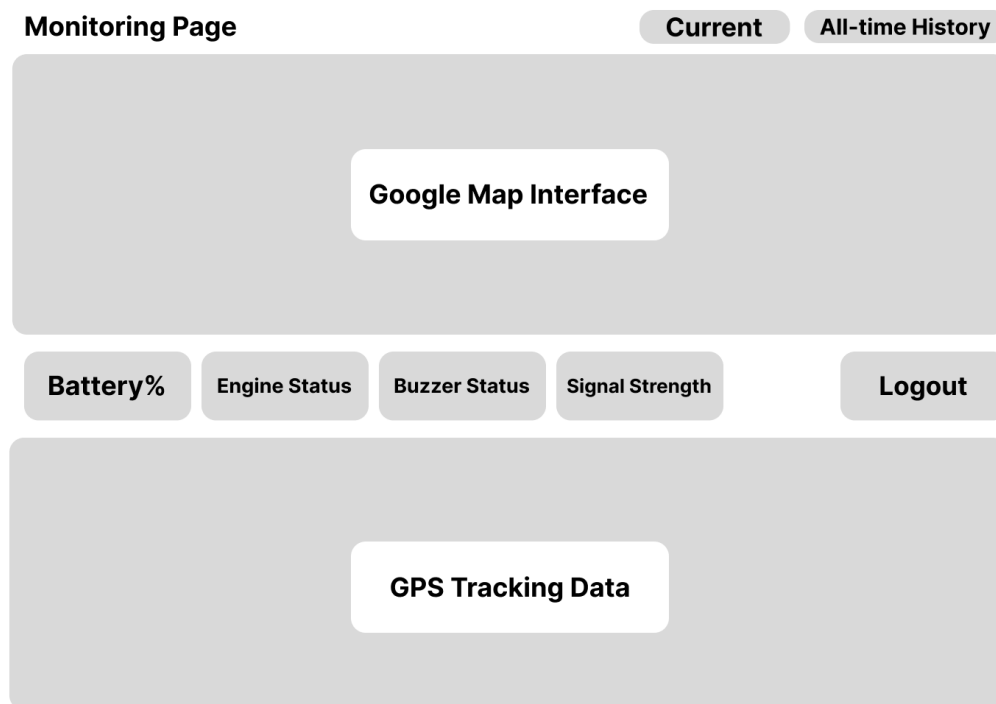
Login Page



Gambar 3.17 *Mockup Login Page LockMotion*

Halaman utama dasbor, seperti terlihat pada *mockup* di Gambar 3.18, dirancang dengan fokus pada penyajian informasi yang cepat dan mudah dipahami. Antarmuka akan dibagi menjadi tiga area utama. Area atas akan menampilkan antarmuka peta interaktif dari Google Maps, yang secara dinamis menunjukkan penanda lokasi terakhir dari perangkat. Area tengah akan diisi dengan serangkaian kartu status yang memberikan informasi vital secara sekilas, seperti persentase baterai, status *relay* (mesin ON/OFF), status *buzzer*, dan kekuatan sinyal GPRS. Area bawah didedikasikan untuk menampilkan data historis pelacakan GPS dalam format tabel yang informatif, dilengkapi dengan fitur untuk beralih antara data

terbaru (*Current*) dan seluruh riwayat (*All-time History*), serta tombol *logout* untuk keluar dari sesi dengan aman.



Gambar 3.18 Mockup Monitoring Page LockMotion

3.2.4 Tahap 4: Pengujian dan Evaluasi

Tahap pengujian merupakan fase krusial untuk memvalidasi fungsionalitas dan mengukur performa prototipe yang telah dikembangkan. Pengujian dilakukan secara sistematis, mencakup verifikasi fungsi setiap komponen utama secara terisolasi hingga pengujian sistem yang telah terintegrasi secara keseluruhan.

3.2.4.1 Pengujian Modul GPS NEO-M8L

Pengujian ini bertujuan untuk memastikan modul GPS dapat mengakuisisi sinyal satelit dan menghasilkan data koordinat yang valid.

Tabel 3.1 Rencana Pengujian Modul GPS NEO-M8L

No.	Aspek yang Diuji	Prosedur Pengujian	Hasil yang Diharapkan
1	Akuisisi Sinyal	Menghubungkan modul ke ESP32 dan menempatkannya di area dengan pandangan ke langit	LED pada modul GPS berkedip, menandakan sedang mencari sinyal
2	Pembacaan Data	Membuka Serial Monitor pada Arduino IDE untuk melihat <i>output</i> data dari modul	Serial Monitor menampilkan data mentah NMEA, termasuk <i>string</i> \$GPRMC yang berisi data <i>latitude</i> dan <i>longitude</i> yang valid

3.2.4.2 Pengujian Modul NFC PN532

Pengujian ini bertujuan untuk memvalidasi kemampuan modul dalam mendeteksi dan membaca *Unique Identifier* (UID) dari kartu NFC.

Tabel 3.2 Rencana Pengujian Modul NFC PN532

No.	Aspek yang Diuji	Prosedur Pengujian	Hasil yang Diharapkan
1	Deteksi Kartu	Menempelkan kartu e-KTP atau kartu NFC lain pada area pembaca modul PN532	Modul berhasil mendeteksi keberadaan kartu
2	Pembacaan UID	Menjalankan <i>script</i> uji pembacaan NFC dan melihat <i>output</i> pada Serial Monitor	Serial Monitor menampilkan rangkaian UID heksadesimal yang unik dari kartu yang ditempelkan

3.2.4.3 Pengujian Modul GSM SIM800L

Pengujian ini bertujuan untuk memastikan modul dapat terhubung ke jaringan seluler dan mampu mengirimkan data melalui GPRS.

Tabel 3.3 Rencana Pengujian Modul GSM SIM800L

No.	Aspek yang Diuji	Prosedur Pengujian	Hasil yang Diharapkan
1	Registrasi Jaringan	Memasukkan kartu SIM aktif dan menyalakan modul. Mengirim perintah AT (AT+CREG?) melalui Serial Monitor.	Modul memberikan respons +CREG: 0,1 atau +CREG: 0,5 yang menandakan berhasil terdaftar di jaringan.
2	Koneksi GPRS (HTTP)	Menjalankan <i>script</i> uji untuk melakukan permintaan HTTP POST ke <i>endpoint</i> API Firebase.	Modul berhasil mengirim data dan menerima kode status 200 OK dari server.

3.2.4.4 Pengujian Modul Relay dan Buzzer

Pengujian ini bertujuan untuk memverifikasi fungsi aktuator, yaitu *relay* sebagai pemutus sirkuit dan *buzzer* sebagai alarm suara.

Tabel 3.4 Rencana Modul Relay dan Buzzer

No.	Aspek yang Diuji	Prosedur Pengujian	Hasil yang Diharapkan
1	Kontrol Relay	Menjalankan <i>script</i> untuk mengubah status pin GPIO yang terhubung ke <i>relay</i> (HIGH/LOW).	Terdengar suara "klik" dari relay yang menandakan perpindahan saklar. Posisi <i>Normally Close</i> (NC) dan <i>Common</i> (COM) terhubung/terputus.
2	Kontrol Buzzer	Menjalankan <i>script</i> untuk memberikan sinyal PWM atau digital ke pin yang terhubung ke <i>buzzer</i> .	<i>Buzzer</i> mengeluarkan suara sesuai dengan sinyal yang diberikan.

3.2.4.5 Uji Validitas Perangkat Keras (*Hardware Black-Box Test*)

Pengujian ini bertujuan untuk memvalidasi bahwa setiap fungsi sistem bekerja sesuai dengan yang dirancang seperti yang bisa dilihat pada Tabel 3.5. Pengujian ini berfokus pada verifikasi alur kerja utama sistem tanpa perlu mengetahui detail implementasi internal. Keberhasilan dalam semua skenario uji ini menjadi syarat mutlak sebelum melangkah ke tahap pengukuran kinerja kuantitatif.

Tabel 3.5 Rencana Pengujian Sistem Perangkat Keras

No.	Skenario Uji (<i>Input</i>)	Aksi yang Dilakukan	Hasil yang Diharapkan (<i>Output</i>)
1.	Autentikasi iTAG BLE	Mendekatkan iTAG yang terdaftar	Sistem masuk Mode NORMAL, <i>relay</i> menyala.
2.	Autentikasi NFC	Menempelkan e-KTP yang UID-nya terdaftar	Sistem masuk Mode PARKIR, <i>relay</i> mati.
3.	Pemicu <i>Geofencing</i>	Menggerakkan perangkat > 70 meter dari titik parkir	Sistem masuk Mode DARURAT, <i>buzzer</i> berbunyi.
4.	Pemicu Putus Koneksi	Memutuskan koneksi iTAG BLE > 10 detik	Sistem masuk Mode DARURAT, <i>buzzer</i> berbunyi.
5.	Pengiriman Data	Menyalakan perangkat dengan koneksi internet	Data lokasi dan status muncul di dasbor web.

3.2.4.6 Uji Validitas Halaman Web (*Front End Black-Box Test*)

Selain pengujian pada perangkat keras, dilakukan pula pengujian validitas fungsional secara terpisah pada antarmuka pengguna, yaitu dasbor web pemantauan. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur interaktif pada halaman web dapat berjalan sesuai dengan fungsionalitas yang dirancang,

mulai dari autentikasi pengguna hingga fitur-fitur penyajian data. Rencana pengujian fungsional untuk halaman web dapat dilihat pada Tabel 3.6.

Tabel 3.6 Rencana Pengujian Halaman Web

No.	Skenario Uji (<i>Input</i>)	Aksi yang Dilakukan	Hasil yang Diharapkan (<i>Output</i>)
1.	<i>Login</i> Pengguna (Berhasil)	Memasukkan email dan kata sandi yang valid pada halaman <i>login</i> .	<i>Login</i> berhasil, pengguna diarahkan ke dasbor.
2.	<i>Login</i> Pengguna (Gagal)	Memasukkan email dan kata sandi yang salah.	Gagal <i>login</i> , pesan <i>error</i> muncul.
3.	Filter Data Riwayat	Memilih rentang tanggal pada kolom filter tanggal yang tersedia.	Tabel menampilkan data sesuai filter tanggal.
4.	Paginasi Data Riwayat	Menekan tombol nomor halaman pada bagian bawah tabel riwayat.	Tabel beralih ke halaman yang dipilih.
5.	Fungsionalitas Ekspor Data	Menekan tombol "CSV" pada dasbor.	<i>File</i> data .csv berhasil diunduh.
6.	<i>Logout</i> Pengguna	Menekan tombol " <i>Logout</i> " pada dasbor utama.	<i>Logout</i> berhasil, kembali ke halaman <i>login</i> .

3.2.4.7 Uji Kinerja dan Efisiensi Sistem

Setelah sistem terbukti valid secara fungsional, pengujian dilanjutkan untuk mengukur metrik kinerja sistem secara kuantitatif. Tahap ini penting untuk mengetahui sejauh mana efektivitas, kecepatan, dan efisiensi prototipe dalam kondisi operasional yang disimulasikan.

a) Analisis Kinerja Pelacakan Lokasi (GPS dan GeoAPI)

Pengujian ini membandingkan kinerja antara sumber lokasi utama (modul GPS NEO-M8L) dan sumber lokasi sekunder (Google Geolocation API) dari segi kecepatan dan akurasi. Pengujian dilakukan di lokasi dengan kondisi penerimaan sinyal yang berbeda (misalnya, *outdoor*, *semi-outdoor*, dan *indoor*). Perbandingan ini krusial untuk memahami keandalan sistem di berbagai skenario nyata, seperti saat kendaraan berada di tempat parkir terbuka versus di dalam garasi bawah tanah. Waktu akuisisi data menjadi tolok ukur penting bagi pengalaman pengguna, karena menentukan seberapa cepat lokasi kendaraan dapat ditampilkan setelah sistem diaktifkan.

Tabel 3.7 Rencana Perbandingan Kinerja Sumber Lokasi

No	Kondisi	Sumber Lokasi (Lat, Lon)		Waktu Akuisisi Data Valid (detik)	Akurasi (meter)
		GPS M8L	GeoAPI		
1	<i>Outdoor</i>	-	-	-	-
2	<i>Indoor</i>	-	-	-	-
...	<i>Semi-outdoor</i>	-	-	-	-
n	...	...	...	...	...

b) Analisis Keandalan Unggah Data ke Firebase RTDB

Pengujian ini mengukur tingkat keberhasilan perangkat dalam mengirimkan data ke *backend* pada kondisi kekuatan sinyal GSM yang berbeda. Tingkat keberhasilan dihitung dari jumlah transmisi sukses dibagi jumlah total percobaan. Keandalan unggah data merupakan tulang punggung dari sistem *monitoring real-time*. Pengujian ini secara langsung memvalidasi kemampuan perangkat untuk tetap terhubung dan melaporkan data bahkan di area dengan jangkauan sinyal yang kurang optimal, sehingga memastikan integritas data historis yang diterima pengguna.

Tabel 3.8 Rencana Uji Keandalan Unggah Data

Kondisi Sinyal (RSSI)	Jumlah Percobaan Unggah	Jumlah Sukses	Tingkat Keberhasilan (%)
Baik (>-75dBm)	20	-	-
Sedang (-76 s.d. -90 dBm)	20	-	-
Lemah (<-90 dBm)	20	-	-

c) Analisis Konsumsi Kuota Data

Pengujian ini bertujuan untuk mengukur efisiensi penggunaan data internet oleh perangkat untuk setiap transmisi data ke *backend*. Konsumsi data diukur dengan menghubungkan perangkat ke *hotspot* dan memantau total data yang terpakai selama periode waktu tertentu untuk menghitung rata-rata kuota per pengiriman. Efisiensi data merupakan faktor ekonomis yang sangat penting. Hasil pengujian ini akan menunjukkan kelayakan sistem untuk digunakan dalam jangka panjang dengan biaya operasional yang rendah bagi pengguna akhir..

Tabel 3.9 Rencana Uji Konsumsi Kuota Data

Waktu (Menit)	Jumlah Pengiriman Data (kumulatif)	Total Kuota Terpakai (KB)
1	2	-
2	4	-
...	...	...
n	-	-

d) Analisis Kecepatan Respons Alarm

Waktu respons alarm diukur menggunakan *stopwatch* untuk mencatat jeda waktu (dalam detik) sejak kondisi darurat terpicu hingga *buzzer* berbunyi. Dalam skenario pencurian, kecepatan respons adalah segalanya. Respons alarm yang cepat (sub-detik) dapat memberikan efek kejutan yang signifikan pada pelaku, berpotensi menggagalkan tindakan kejahatan sejak awal.

Tabel 3.10 Rencana Uji Waktu Respons Alarm

Percobaan Ke-	Skenario Pemicu	Waktu Respons (detik)
1	Putus Koneksi BLE	-
2	Pelanggaran <i>Geo-fence</i>	-
3	Campuran	-
...	-	-
n	-	-
Rata-rata		-

d) Analisis Kinerja Catu Daya

Pengujian ini mengevaluasi dua aspek utama dari sistem catu daya: daya tahan penggunaan dan durasi pengisian.

Aspek pertama yang dievaluasi adalah daya tahan baterai. Pengujian ini bertujuan untuk mengukur total waktu operasional perangkat, mulai dari kondisi baterai terisi penuh hingga habis. Untuk mendapatkan data ini, tegangan baterai dipantau secara berkala selama pengujian. Proses ini dirancang untuk membuat simulasi skenario penggunaan di dunia nyata, misalnya ketika kendaraan diparkir untuk waktu yang lama, sehingga tingkat otonomi sistem dapat ditentukan secara akurat.

Tabel 3.11 Rencana Uji Daya Tahan Baterai

Waktu Operasional (jam)	Tegangan Baterai (Volt)	Persentase Baterai (%)	Status Perangkat
0	-	-	Aktif Normal
1	-	-	Aktif Normal
2	-	-	Aktif Normal
...	-	-	...
n (akhir)	-	-	Perangkat Mati

Aspek kedua adalah durasi pengisian daya. Pengujian ini berfokus pada pengukuran waktu yang dibutuhkan untuk mengisi baterai dari kondisi habis hingga kembali penuh dengan menggunakan sirkuit *charger* yang telah dirancang. Durasi pengisian yang efisien merupakan faktor krusial karena sangat memengaruhi

kenyamanan pengguna dalam memastikan perangkat selalu dalam kondisi siap pakai.

Tabel 3.12 Rencana Uji Durasi Pengisian Daya

Percobaan Ke-	Tegangan Awal (Volt)	Tegangan Akhir (Volt)	Durasi Pengisian (Menit)
1	-	-	-
2	-	-	-
3	-	-	-
...	-	-	...
n	-	-	-

Secara keseluruhan, rangkaian pengujian kuantitatif ini akan memberikan data yang kaya dan terukur untuk melakukan analisis mendalam terhadap kekuatan dan kelemahan prototipe yang dikembangkan.

3.3 Instrumen Penelitian

Instrumen yang digunakan untuk mengumpulkan data selama tahap pengujian adalah:

- Lembar Kerja Uji Fungsional: Sebuah *checklist* untuk mencatat keberhasilan atau kegagalan setiap skenario uji.
- Stopwatch* Digital: Digunakan untuk mengukur waktu respons alarm.
- Log Pencatatan Data Kinerja: Lembar kerja digital untuk mencatat data kuantitatif hasil pengujian.

3.4 Teknik Analisis Data

Data yang telah dikumpulkan akan dianalisis menggunakan teknik berikut:

- Analisis Deskriptif Kualitatif: Data dari uji fungsional akan dianalisis secara deskriptif (berhasil/gagal) untuk menyimpulkan validitas fungsional sistem.
- Analisis Deskriptif Kuantitatif: Data dari uji kinerja (akurasi, waktu, konsumsi data) akan diolah untuk mencari nilai statistik seperti rata-rata (*mean*) dan rentang nilai untuk memberikan gambaran mengenai efektivitas kinerja prototipe.