

**PENGEMBANGAN PERANGKAT GPS TRACKER
PORTABEL DAN NON-INVASIVE DENGAN
IMPLEMENTASI GEOFENCING BERBASIS WILAYAH
ADMINISTRATIF UNTUK SISTEM KEAMANAN
KENDARAAN BERMOTOR**

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan
Program Strata 1, Program Studi Teknik Informatika,
Universitas Pasundan Bandung

Disusun oleh:

Aldi Pradana Hakim

NRP. 22.304.0035



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN BANDUNG
SEPTEMBER 2026**

**LEMBAR PENGESAHAN
LAPORAN TUGAS AKHIR**

Telah disetujui dan disahkan Proposal Tugas Akhir, dari:

Nama : Aldi Pradana Hakim

Nrp. : 22.304.0035

Dengan judul:

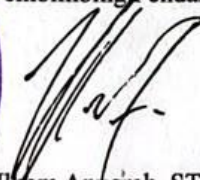
**“PENGEMBANGAN PERANGKAT GPS TRACKER PORTABEL DAN NON-INVASIVE
DENGAN IMPLEMENTASI GEOFENCING BERBASIS WILAYAH ADMINISTRATIF UNTUK
SISTEM KEAMANAN KENDARAAN BERMOTOR”**

Bandung, 30 September 2026

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

(Muhammad Tirta Mulia, ST., MT.) (Moch. Ilham Anugrah, ST., M.Sc.Eng.)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa:

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung.

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

Bandung, 30 September 2026
Yang membuat pernyataan



(Aldi Pradana Hakim)
NRP.22.304.0035

ABSTRAK

Tingginya angka pencurian kendaraan bermotor roda dua di wilayah perkotaan, seperti terungkapnya 1.494 unit kendaraan hasil kejahatan di sebuah gudang di Jakarta Selatan pada tahun 2026, menunjukkan bahwa sistem keamanan kendaraan yang tersedia saat ini masih memiliki keterbatasan. Perangkat GPS konvensional umumnya bersifat invasif, mudah dideteksi dan dinonaktifkan oleh pelaku pencurian, serta belum dilengkapi mekanisme pengawasan wilayah yang bersifat proaktif seperti geofencing berbasis wilayah administratif.

Penelitian ini bertujuan mengembangkan sistem pemantauan lokasi kendaraan bermotor secara real-time berbasis website yang dipasangkan dengan perangkat GPS tracker portabel dan non-invasive berbasis ESP32, modul GPS NEO-6M, dan modul SIM800L yang dikembangkan oleh rekan satu tim. Metodologi pengembangan mengadaptasi model Pressman yang terdiri atas tahapan Communication, Planning, Modeling, Construction, dan Deployment. Website dikembangkan menggunakan framework Laravel dengan fitur pemantauan lokasi real-time melalui protokol MQTT dan WebSocket/Pusher pada peta Leaflet.js, geofencing dinamis berbasis wilayah administratif, radius, dan poligon yang dilengkapi notifikasi pelanggaran zona melalui email, WhatsApp (API Fonnte), dan notifikasi dalam aplikasi, deteksi durasi berhenti dan deteksi kecepatan berlebih (overspeed) beserta notifikasinya, pengelolaan lokasi rumah, riwayat perjalanan dan riwayat berhenti yang dapat diekspor dalam format PDF, serta panel admin untuk pengelolaan pengguna dan perangkat.

Pengujian black-box yang dilakukan terhadap 86 kasus uji perangkat lunak dan 27 kasus uji perangkat keras, atau total 113 kasus uji, menunjukkan tingkat keberhasilan 100% pada seluruh fungsi yang diuji. Hasil tersebut menunjukkan bahwa sistem yang dikembangkan mampu membantu pengguna memantau lokasi kendaraan secara real-time, memberikan peringatan dini ketika kendaraan keluar dari zona aman yang ditentukan, serta meningkatkan efektivitas pencegahan dan penanganan tindak pencurian kendaraan bermotor.

Kata Kunci: GPS Tracker, Internet of Things, Geofencing, Pemantauan Real-Time, Notifikasi, Website

ABSTRACT

The high incidence of motorcycle theft in urban areas in Indonesia, exemplified by the 2026 discovery of 1,494 stolen vehicles at a warehouse in South Jakarta, indicates that existing vehicle security systems remain limited. Conventional GPS trackers tend to be invasive, are easily located and disabled by thieves, and lack proactive area-monitoring mechanisms such as administrative-boundary-based geofencing. This study aims to develop a real-time, website-based vehicle location monitoring system paired with a portable, non-invasive GPS tracker built on an ESP32 microcontroller, a NEO-6M GPS module, and a SIM800L GSM module developed by a teammate.

The development methodology adapts Pressman's software process model, comprising the Communication, Planning, Modeling, Construction, and Deployment phases. The website was built with the Laravel framework and provides real-time location tracking over MQTT and WebSocket/Pusher on a Leaflet.js map, dynamic geofencing based on administrative boundaries, radius, and polygon zones with violation notifications delivered via email, WhatsApp (Fonnte API), and in-app alerts, stop-duration and overspeed detection with accompanying notifications, home-location management, and trip/stop history exportable to PDF, as well as an admin panel for managing users and devices.

Black-box testing covering 86 software test cases and 27 hardware test cases, 113 test cases in total, achieved a 100% success rate across all tested functions. These results demonstrate that the developed system enables users to monitor vehicle locations in real time, issue early warnings when a vehicle leaves its designated safe zone, and improve the effectiveness of motorcycle theft prevention and recovery.

Keywords: GPS Tracker, Internet of Things, Geofencing, Real-Time Monitoring, Notification, Website

KATA PENGANTAR

Dengan mengucap syukur kehadiran Allah SWT, berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "PENGEMBANGAN PERANGKAT GPS TRACKER PORTABEL DAN NON-INVASIVE DENGAN IMPLEMENTASI GEOFENCING BERBASIS WILAYAH ADMINISTRATIF UNTUK SISTEM KEAMANAN KENDARAAN BERMOTOR". Laporan ini merupakan salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Universitas Pasundan Bandung.

Penulis menyadari bahwa keberhasilan dalam menyusun laporan ini tidak lepas dari dukungan dan bimbingan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan penghargaan dan terima kasih yang setulus-tulusnya atas bantuan moril maupun material yang telah diberikan, kepada:

1. Allah SWT, atas segala nikmat, kemudahan, dan kekuatan yang selalu diberikan kepada penulis dalam setiap langkah menyelesaikan Tugas Akhir ini.
2. Kedua orang tua dan keluarga tercinta, yang tidak pernah berhenti memanjatkan doa serta memberikan motivasi kepada penulis, baik dalam suka maupun duka, sehingga penulis mampu bertahan dan menyelesaikan studi ini hingga akhir.
3. Bapak Muhammad Tirta Mulia, ST., MT., selaku Pembimbing Utama, yang telah dengan sabar memberikan arahan, bimbingan, serta motivasi kepada penulis selama proses penyusunan Tugas Akhir ini berlangsung.
4. Bapak Moch. Ilham Anugrah, ST., M.Sc.Eng., selaku Pembimbing Pendamping, yang senantiasa memberikan arahan, saran, dan bimbingan yang sangat berarti bagi penulis selama proses penyusunan Tugas Akhir ini.
5. Seluruh dosen di Program Studi Teknik Informatika Universitas Pasundan Bandung, yang telah membekali penulis dengan ilmu pengetahuan, wawasan, dan pengalaman berharga selama masa perkuliahan.
6. Seluruh civitas akademika Program Studi Teknik Informatika Universitas Pasundan Bandung, yang telah memberikan dukungan fasilitas serta pelayanan akademik selama penulis menempuh pendidikan.
7. Teman-teman kontrakan, yang telah menemani penulis selama kurang lebih empat tahun masa perkuliahan, berbagi suka dan duka layaknya keluarga kedua bagi penulis.
8. Teman-teman seperjuangan di Universitas Pasundan Bandung yang tidak dapat disebutkan satu per satu, atas kebersamaan, dukungan, dan semangat yang selalu diberikan kepada penulis selama menempuh pendidikan.
9. Teman-teman Wargi 124, atas kebersamaan, keceriaan, dan solidaritas yang telah terjalin selama ini, yang turut menjadi warna tersendiri dalam perjalanan penulis menyelesaikan studi.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif guna perbaikan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat, tidak hanya bagi penulis, tetapi juga bagi pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi.

Bandung, 30 September 2026



Aldi Pradana Hakim

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa:

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung.

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

Bandung, 30 September 2026
Yang membuat pernyataan



(Aldi Pradana Hakim)
NRP.22.304.0035

ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1-1
1.1 Latar Belakang	1-1
1.2 Identifikasi Masalah	1-3
1.2.1 Identifikasi Masalah	1-3
1.2.2 Rumusan Masalah	1-3
1.3 Tujuan Tugas Akhir	1-4
1.4 Lingkup Tugas Akhir	1-4
1.4.1 Ruang Lingkup	1-4
1.4.2 Batasan	1-5
1.5 Metodologi Tugas Akhir	1-5
1.6 Sistematika Penulisan Laporan Tugas Akhir	1-8
BAB 2 LANDASAN TEORI	2-1
2.1 Teori Pendukung	2-1
2.1.1 Konsep Dasar Sistem	2-1
2.1.2 Aplikasi Berbasis Web	2-1
2.1.3 Internet of Things (IoT)	2-3
2.1.4 Global Positioning System (GPS)	2-4
2.1.5 MQTT (Message Queuing Telemetry Transport)	2-5
2.1.6 Laravel	2-6
2.1.7 PostgreSQL	2-7
2.1.8 WebSocket dan Pusher	2-7
2.1.9 Leaflet.js	2-8
2.1.10 GeoJSON	2-9
2.1.11 Geofencing	2-10
2.1.12 Point In Polygon	2-11
2.1.13 Reverse Geocoding	2-12
2.1.14 Notifikasi Email dan WhatsApp	2-13
2.1.15 Real-Time Monitoring	2-14
2.1.16 Black Box Testing	2-15
2.1.17 User Acceptance Testing (UAT)	2-16
2.2 Penelitian Terdahulu	2-16
BAB 3 SKEMA PENELITIAN	3-1
3.1 Alur Penyelesaian Tugas Akhir	3-1
3.2 Perumusan Masalah	3-3
3.2.1 Analisis Sebab Akibat	3-3

3.2.2 Analisis Solusi Masalah	3-4
3.3 Kerangka Pemikiran Teoritis	3-5
3.3.1 Gambaran Produk Tugas Akhir	3-6
3.4 Objek Penelitian	3-7
3.5 Analisis Kebutuhan Pengguna	3-8
3.5.1 Analisis Aktor	3-9
3.5.2 User Requirement	3-9
3.6 Analisis <i>Hardware</i> Requirement	3-10
3.7 Analisis Software Requirement	3-12
3.8 Analisis Kebutuhan Fungsional	3-14
3.9 Analisis Kebutuhan Non-Fungsional	3-16
3.10 Analisis Kebutuhan Data	3-17
3.11 Analisis Kebutuhan Pengembangan Perangkat Lunak	3-19
3.12 Analisis Kebutuhan <i>Deployment</i>	3-20
BAB 4 PERANCANGAN SISTEM	4-1
4.1 Pemodelan Berbasis Skenario	4-1
4.1.1 Business Use Case	4-1
4.1.2 Use Case	4-2
4.1.3 Diagram Sequence	4-6
4.1.4 Skenario Use Case	4-23
4.2 Pemodelan Data	4-37
4.2.1 Class Diagram	4-37
4.2.2 Entity Relationship Diagram (ERD)	4-37
4.3 Perancangan Perangkat Keras	4-38
4.3.1 Perancangan Rangkaian Catu Daya	4-38
4.3.2 Perancangan Rangkaian Akuisisi GPS	4-40
4.3.3 Perancangan Rangkaian Komunikasi GSM	4-40
4.3.4 Perancangan Mekanik	4-41
4.4 Arsitektur Sistem	4-42
4.4.1 Blok Diagram Perangkat GPS Tracker	4-42
4.4.2 Blok Diagram Alur Komunikasi Sistem	4-43
4.5 Prototype	4-44
4.5.1 Rancangan Antarmuka Halaman Login	4-44
4.5.2 Rancangan Antarmuka Halaman Aktivasi Perangkat	4-45
4.5.3 Rancangan Antarmuka Halaman Lengkapi Profil	4-46
4.5.4 Rancangan Antarmuka Halaman Lupa Password	4-47
4.5.5 Rancangan Antarmuka Halaman Informasi Kendaraan	4-47
4.5.6 Rancangan Antarmuka Halaman Utama	4-48
4.5.7 Rancangan Antarmuka Halaman Detail Kendaraan	4-49
BAB 5 IMPLEMENTASI DAN PENGUJIAN	5-1

5.1 Implementasi.....	5-1
5.1.1 Lingkup Implementasi	5-1
5.1.2 Implementasi Struktur Proyek.....	5-2
5.1.3 Implementasi Antarmuka Perangkat Lunak (Software).....	5-3
5.1.4 Implementasi Antarmuka Perangkat Keras (Hardware)	5-12
5.1.5 Implementasi Kode Program Perangkat Lunak (Software)	5-12
5.1.6 Implementasi Kode Program Perangkat Keras (Firmware ESP32).....	5-19
5.1.7 Implementasi Database	5-24
5.2 Pengujian	5-24
5.2.1 Rencana Pengujian	5-24
5.2.2 Hasil Pengujian Perangkat Lunak (Software)	5-26
5.2.3 Hasil Pengujian Perangkat Keras (Hardware).....	5-36
5.2.4 Rangkuman Hasil Pengujian	5-40
5.2.5 Pengujian User Acceptance Testing (UAT).....	5-42
BAB 6 KESIMPULAN DAN SARAN.....	6-1
6.1 Kesimpulan.....	6-1
6.2 Saran.....	6-3
DAFTAR PUSTAKA.....	xiii
LAMPIRAN	xiii
A. UAT Admin.....	1
B. UAT Pengguna	1

DAFTAR GAMBAR

Gambar 1.1. Metodologi Penyelesaian Tugas Akhir.....	1-6
Gambar 3.1. Fishbone Sebab Akibat.....	3-3
Gambar 3.2. Fishbone Solusi Masalah.....	3-4
Gambar 4.1. Diagram Business Use Case.....	4-2
Gambar 4.2. Diagram Use Case	4-6
Gambar 4.3. Diagram Sequence UC-01.....	4-6
Gambar 4.4. Diagram Sequence UC-02.....	4-7
Gambar 4.5. Diagram Sequence UC-03.....	4-8
Gambar 4.6. Diagram Sequence UC-04.....	4-8
Gambar 4.7. Diagram Sequence UC-05.....	4-9
Gambar 4.8. Diagram Sequence UC-06.....	4-9
Gambar 4.9. Diagram Sequence UC-07.....	4-10
Gambar 4.10. Diagram Sequence UC-08.....	4-11
Gambar 4.11. Diagram Sequence UC-09.....	4-12
Gambar 4.12. Diagram Sequence UC-10.....	4-13
Gambar 4.13. Diagram Sequence UC-11.....	4-14
Gambar 4.14. Diagram Sequence UC-12.....	4-14
Gambar 4.15. Diagram Sequence UC-13.....	4-15
Gambar 4.16. Diagram Sequence UC-14.....	4-15
Gambar 4.17. Diagram Sequence UC-15.....	4-16
Gambar 4.18. Diagram Sequence UC-16.....	4-16
Gambar 4.19. Diagram Sequence UC-17.....	4-17
Gambar 4.20. Diagram Sequence UC-18.....	4-18
Gambar 4.21. Diagram Sequence UC-19.....	4-18
Gambar 4.22. Diagram Sequence UC-20.....	4-19
Gambar 4.23. Diagram Sequence UC-21.....	4-19
Gambar 4.24. Diagram Sequence UC-22.....	4-20
Gambar 4.25. Diagram Sequence UC-23.....	4-20
Gambar 4.26. Diagram Sequence UC-24.....	4-21
Gambar 4.27. Diagram Sequence UC-25.....	4-22
Gambar 4.28. Diagram Sequence UC-26.....	4-22
Gambar 4.29. Class Diagram.....	4-37
Gambar 4.30. ERD	4-38
Gambar 4.31. Rangkaian Catu Daya.....	4-39
Gambar 4.32. Rangkaian Akuisisi GPS	4-40
Gambar 4.33. Rangkaian Komunikasi GSM.....	4-41
Gambar 4.34. Perancangan Mekanik	4-42
Gambar 4.35. Blok Diagram Perangkat GPS Tracker.....	4-43
Gambar 4.36. Blok Diagram Alur Komunikasi Sistem.....	4-44
Gambar 4.37. Rancangan Antarmuka Halaman Login	4-45
Gambar 4.38. Rancangan Antarmuka Halaman Aktivasi Perangkat.....	4-46
Gambar 4.39. Rancangan Antarmuka Halaman Lengkapi Profil	4-46
Gambar 4.40. Rancangan Antarmuka Halaman Lupa Password	4-47
Gambar 4.41. Rancangan Antarmuka Halaman Informasi Kendaraan	4-48
Gambar 4.42. Rancangan Antarmuka Halaman Utama	4-49
Gambar 4.43. Rancangan Antarmuka Bagian Informasi Kendaraan	4-50
Gambar 4.44. Rancangan Antarmuka Bagian Home Location.....	4-51
Gambar 4.45. Rancangan Antarmuka Bagian Geofence.....	4-52
Gambar 4.46. Rancangan Antarmuka Bagian Riwayat Perjalanan.....	4-53
Gambar 4.47. Rancangan Antarmuka Bagian Pesan.....	4-54
Gambar 4.48. Rancangan Antarmuka Bagian Kendaraan Berhenti.....	4-55
Gambar 4.49. Rancangan Antarmuka Bagian Batas Kecepatan	4-56
Gambar 5.1. Struktur Proyek Software.....	5-3
Gambar 5.2. Struktur Proyek Hardware.....	5-3

Gambar 5.3. Implementasi Antarmuka Halaman Login	5-4
Gambar 5.4. Implementasi Antarmuka Halaman Aktivasi Perangkat.....	5-5
Gambar 5.5. Implementasi Antarmuka Halaman Lengkapi Profil.....	5-5
Gambar 5.6. Implementasi Antarmuka Halaman Lupa Password	5-6
Gambar 5.7. Implementasi Antarmuka Halaman Informasi Kendaraan	5-7
Gambar 5.8. Implementasi Antarmuka Halaman Utama	5-7
Gambar 5.9. Implementasi Antarmuka Halaman Detail Informasi Kendaraan	5-8
Gambar 5.10. Implementasi Antarmuka Halaman Detail Home Location.....	5-9
Gambar 5.11. Implementasi Antarmuka Halaman Detail Geofence	5-9
Gambar 5.12. Implementasi Antarmuka Halaman Detail Riwayat Perjalanan.....	5-10
Gambar 5.13. Implementasi Antarmuka Halaman Detail Pesan.....	5-10
Gambar 5.14. Implementasi Antarmuka Halaman Detail Kendaraan Berhenti.....	5-11
Gambar 5.15. Implementasi Antarmuka Halaman Detail Batas Kecepatan	5-11
Gambar 5.16. Implementasi Antarmuka Perangkat Keras	5-12
Gambar 5.17. Kode Program Validasi Data GPS Melalui MQTT.....	5-13
Gambar 5.18. Kode Program Geofencing Administratif dan PIP.....	5-14
Gambar 5.19. Kode Program Geofencing Radius	5-15
Gambar 5.20. Kode Program Kecepatan Berlebih.....	5-16
Gambar 5.21. Kode Program Real-time.....	5-17
Gambar 5.22. Kode Program Notifikasi WhatsApp	5-18
Gambar 5.23. Kode Program Riwayat Perjalanan	5-18
Gambar 5.24. Kode Program Koneksi ke GPRS	5-19
Gambar 5.25. Kode Program Akuisisi Data GPS	5-21
Gambar 5.26. Kode Program Pengiriman Data Lokasi.....	5-22
Gambar 5.27. Kode Program Siklus Utama dan Reconnect Otomatis.....	5-23

DAFTAR TABEL

Tabel 2.1. Penelitian Terdahulu	2-16
Tabel 3.1. Alur Penyelesaian Tugas Akhir	3-1
Tabel 3.2. Deskripsi Fishbone Sebab Akibat.....	3-3
Tabel 3.3. Deskripsi Fishbone Solusi Masalah.....	3-4
Tabel 3.4. Analisis Aktor.....	3-9
Tabel 3.5. User Requirement	3-9
Tabel 3.6. Analisis Hardware Requirement.....	3-10
Tabel 3.7. Analisis Perbandingan Pemilihan Modul Perangkat Keras	3-11
Tabel 3.8. Analisis Software Requirement	3-12
Tabel 3.9. Analisis Kebutuhan Fungsional.....	3-14
Tabel 3.10. Analisis Kebutuhan Non-Fungsional	3-16
Tabel 3.11. Analisis Kebutuhan Data.....	3-17
Tabel 3.12. Analisis Kebutuhan Pengembangan Perangkat Lunak.....	3-19
Tabel 3.13. Analisis Kebutuhan Deployment	3-20
Tabel 4.1. Business Use Case	4-1
Tabel 4.2. Use Case.....	4-3
Tabel 4.3. Skenario UC-01	4-23
Tabel 4.4. Skenario UC-02.....	4-24
Tabel 4.5. Skenario UC-03	4-24
Tabel 4.6. Skenario UC-04.....	4-25
Tabel 4.7. Skenario UC-05.....	4-25
Tabel 4.8. Skenario UC-06.....	4-26
Tabel 4.9. Skenario UC-07.....	4-26
Tabel 4.10. Skenario UC-08.....	4-27
Tabel 4.11. Skenario UC-09.....	4-27
Tabel 4.12. Skenario UC-10.....	4-28
Tabel 4.13. Skenario UC-11	4-28
Tabel 4.14. Skenario UC-12.....	4-29
Tabel 4.15. Skenario UC-13	4-29
Tabel 4.16. Skenario UC-14.....	4-30
Tabel 4.17. Skenario UC-15.....	4-30
Tabel 4.18. Skenario UC-16.....	4-31
Tabel 4.19. Skenario UC-17.....	4-32
Tabel 4.20. Skenario UC-18.....	4-32
Tabel 4.21. Skenario UC-19.....	4-33
Tabel 4.22. Skenario UC-20.....	4-33
Tabel 4.23. Skenario UC-21	4-34
Tabel 4.24. Skenario UC-22.....	4-34
Tabel 4.25. Skenario UC-23	4-35
Tabel 4.26. Skenario UC-24.....	4-35
Tabel 4.27. Skenario UC-25.....	4-35
Tabel 4.28. Skenario UC-26.....	4-36
Tabel 5.1. Implementasi Komponen	5-1
Tabel 5.2. Implementasi Perangkat Lunak/Teknologi.....	5-1
Tabel 5.3. Implementasi Database	5-24
Tabel 5.4. Rancangan Pengujian Perangkat Lunak.....	5-25
Tabel 5.5. Rencana Pengujian Perangkat Keras	5-25
Tabel 5.6. Test case UC-01.....	5-26
Tabel 5.7. Test case UC-02.....	5-26
Tabel 5.8. Test case UC-03.....	5-27
Tabel 5.9. Test case UC-04.....	5-27
Tabel 5.10. Test case UC-05.....	5-27
Tabel 5.11. Test case UC-06.....	5-28
Tabel 5.12. Test case UC-07.....	5-28

Tabel 5.13. Test case UC-08	5-29
Tabel 5.14. Test case UC-09	5-29
Tabel 5.15. Test case UC-10	5-30
Tabel 5.16. Test case UC-11	5-30
Tabel 5.17. Test case UC-12	5-30
Tabel 5.18. Test case UC-13	5-31
Tabel 5.19. Test case UC-14	5-31
Tabel 5.20. Test case UC-15	5-31
Tabel 5.21. Test case UC-16	5-32
Tabel 5.22. Test case UC-17	5-32
Tabel 5.23. Test case UC-18	5-33
Tabel 5.24. Test case UC-19	5-33
Tabel 5.25. Test case UC-20	5-34
Tabel 5.26. Test case UC-21	5-34
Tabel 5.27. Test case UC-22	5-34
Tabel 5.28. Test case UC-23	5-34
Tabel 5.29. Test case UC-24	5-35
Tabel 5.30. Test case UC-25	5-35
Tabel 5.31. Test case UC-26	5-36
Tabel 5.32. Test case HW-01	5-36
Tabel 5.33. Test case HW-02	5-37
Tabel 5.34. Test case HW-03	5-37
Tabel 5.35. Test case HW-04	5-37
Tabel 5.36. Test case HW-05	5-38
Tabel 5.37. Test case HW-06	5-38
Tabel 5.38. Test case HW-07	5-39
Tabel 5.39. Test case HW-08	5-39
Tabel 5.40. Test case HW-09	5-39
Tabel 5.41. Test case HW-10	5-40
Tabel 5.42. Test case HW-11	5-40
Tabel 5.43. Rangkuman Hasil Pengujian Perangkat Lunak	5-41
Tabel 5.44. Rangkuman Pengujian Perangkat Keras	5-41
Tabel 5.45. Pengujian UAT Pengguna	5-42
Tabel 5.46. Pengujian UAT Admin	5-44
Tabel 5.47. Rangkuman Hasil UAT	5-44

BAB 1

PENDAHULUAN

Bab ini menguraikan latar belakang penelitian, identifikasi dan rumusan masalah, tujuan, ruang lingkup dan batasan, metodologi, serta sistematika penulisan yang menjadi dasar pelaksanaan Tugas Akhir mengenai sistem AF GPS Tracker.

1.1 Latar Belakang

Kendaraan bermotor roda dua merupakan salah satu sarana transportasi yang paling banyak digunakan oleh masyarakat Indonesia. Selain berfungsi sebagai alat mobilitas sehari-hari, kendaraan bermotor juga memiliki nilai ekonomi yang cukup tinggi sehingga menjadi salah satu aset yang perlu dijaga keamanannya. Tingginya tingkat penggunaan kendaraan bermotor menyebabkan kebutuhan terhadap sistem keamanan kendaraan menjadi semakin penting, terutama untuk mengurangi risiko kehilangan akibat tindak pencurian.

Meskipun berbagai teknologi keamanan kendaraan telah berkembang, kasus pencurian kendaraan bermotor masih menjadi salah satu tindak kriminal yang sering terjadi di Indonesia. Fenomena tersebut terlihat dari keberhasilan Polda Metro Jaya mengungkapkan sebuah gudang penyimpanan kendaraan bermotor ilegal di wilayah Jakarta Selatan pada tahun 2026. Dalam pengungkapan tersebut, polisi berhasil mengamankan sebanyak 1.494 unit kendaraan bermotor yang diduga berasal dari berbagai tindak kejahatan. Dari jumlah tersebut, sebanyak 957 unit kendaraan ditemukan dalam kondisi utuh, sedangkan 537 unit lainnya telah dibongkar menjadi komponen-komponen terpisah. Selain itu, Direktur sebuah perusahaan ekspor turut ditetapkan sebagai tersangka dalam kasus tersebut [DET26].

Besarnya jumlah kendaraan yang berhasil diamankan menunjukkan bahwa tindak pencurian dan peredaran kendaraan bermotor ilegal masih terjadi dalam skala besar dan dilakukan secara terorganisir. Kendaraan hasil kejahatan dapat berpindah dari satu wilayah ke wilayah lainnya dalam waktu yang relatif singkat tanpa diketahui oleh pemiliknya. Kondisi tersebut menyebabkan proses pencarian kendaraan menjadi semakin sulit karena kendaraan berpotensi dipindahkan, dijual kembali, atau bahkan dibongkar menjadi suku cadang sebelum berhasil ditemukan [DET26].

Permasalahan tersebut menunjukkan bahwa pemilik kendaraan membutuhkan suatu mekanisme yang mampu memberikan informasi keberadaan kendaraan secara cepat dan akurat. Semakin lama kendaraan tidak diketahui lokasinya, semakin kecil peluang kendaraan ditemukan dalam kondisi utuh. Oleh karena itu, diperlukan suatu sistem yang mampu membantu pengguna melakukan pemantauan kendaraan secara berkelanjutan sehingga tindakan pencegahan maupun penanganan dapat dilakukan lebih awal ketika terjadi indikasi kehilangan atau perpindahan kendaraan yang tidak wajar.

Salah satu teknologi yang saat ini banyak digunakan untuk membantu proses pelacakan kendaraan adalah Global Positioning System (GPS). Teknologi GPS memungkinkan pengguna mengetahui posisi kendaraan berdasarkan koordinat geografis yang diperoleh dari satelit. Pemanfaatan GPS pada kendaraan memberikan kemudahan bagi pengguna untuk memantau lokasi kendaraan secara *real-time* dan membantu proses pencarian kendaraan ketika terjadi kehilangan. Oleh karena itu, GPS telah menjadi salah satu solusi yang umum digunakan dalam sistem keamanan kendaraan modern.

Namun demikian, sebagian besar sistem pelacakan berbasis GPS yang tersedia saat ini masih memiliki beberapa keterbatasan. Sistem umumnya hanya menampilkan informasi koordinat atau posisi kendaraan tanpa menyediakan mekanisme pengawasan wilayah yang bersifat proaktif. Pengguna harus melakukan pemantauan secara berkala untuk mengetahui apakah kendaraan masih berada pada lokasi yang aman atau telah berpindah ke lokasi yang mencurigakan. Kondisi tersebut menyebabkan sistem pelacakan lebih berperan sebagai alat pencarian setelah kendaraan berpindah lokasi dibandingkan sebagai sarana pencegahan dini terhadap potensi pencurian kendaraan.

Salah satu pendekatan yang dapat digunakan untuk mengatasi keterbatasan tersebut adalah penerapan *geofencing*. *Geofencing* merupakan teknologi yang memungkinkan sistem membentuk batas wilayah virtual tertentu dan melakukan pemantauan terhadap pergerakan objek berdasarkan wilayah yang telah ditentukan. Dengan memanfaatkan *geofencing*, pengguna dapat menentukan zona aman bagi kendaraan yang dimiliki. Ketika kendaraan keluar dari wilayah yang telah ditetapkan, sistem dapat secara otomatis memberikan notifikasi kepada pengguna sehingga tindakan pencegahan atau penanganan dapat dilakukan lebih cepat.

Pada umumnya implementasi *geofencing* dilakukan menggunakan radius tertentu berdasarkan titik koordinat yang ditentukan pengguna. Pendekatan tersebut memiliki keterbatasan karena batas wilayah yang terbentuk belum tentu sesuai dengan batas wilayah yang dipahami oleh masyarakat. Dalam praktiknya, pengguna lebih mudah memahami batas wilayah berdasarkan wilayah administratif seperti kota

Kota/kabupaten, dibandingkan menggunakan radius koordinat tertentu. Oleh karena itu, implementasi *geofencing* berbasis wilayah administratif menjadi pendekatan yang lebih intuitif dan relevan untuk diterapkan pada sistem keamanan kendaraan bermotor.

Untuk mendukung penerapan *geofencing* berbasis wilayah administratif, diperlukan media pemantauan yang mampu menyajikan informasi lokasi kendaraan secara *real-time* dan mudah diakses oleh pengguna. Website dipilih sebagai media pemantauan karena dapat diakses melalui berbagai perangkat tanpa memerlukan proses instalasi tambahan. Melalui website, pengguna dapat memantau posisi kendaraan pada peta digital, melihat status perangkat GPS,

mengakses riwayat perjalanan kendaraan, serta menerima notifikasi melalui email dan WhatsApp ketika kendaraan keluar dari wilayah administratif yang telah ditentukan sebagai zona aman.

Berdasarkan uraian tersebut, penelitian ini berfokus pada pengembangan website pemantauan lokasi kendaraan bermotor secara real-time dengan implementasi geofencing berbasis wilayah administratif. Sistem yang dikembangkan diharapkan mampu membantu pengguna dalam melakukan pemantauan kendaraan secara lebih efektif, memberikan peringatan dini ketika kendaraan keluar dari zona yang telah ditentukan, serta meningkatkan keamanan kendaraan melalui pemanfaatan teknologi pemantauan berbasis web. Oleh karena itu, penelitian ini diangkat dengan judul “PENGEMBANGAN PERANGKAT GPS TRACKER PORTABEL DAN NON-INVASIVE DENGAN IMPLEMENTASI GEOFENCING BERBASIS WILAYAH ADMINISTRATIF UNTUK SISTEM KEAMANAN KENDARAAN BERMOTOR.”

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka permasalahan yang dimunculkan pada tugas akhir ini beserta rumusan pertanyaan penelitiannya adalah sebagai berikut:

1.2.1 Identifikasi Masalah

1. Tingginya angka kasus pencurian kendaraan bermotor roda dua di wilayah perkotaan yang terbukti dari ditemukannya 1.494 unit kendaraan bermotor di sebuah gudang di Jakarta Selatan.
2. Perangkat GPS yang tersedia saat ini belum dilengkapi fitur pembatasan wilayah administratif (*geofencing*) sehingga pemilik kendaraan tidak dapat menetapkan zona aman bagi kendaraannya.
3. Pelaku pencurian telah memahami cara mengelabui atau menonaktifkan perangkat GPS konvensional yang beredar saat ini, sehingga sistem pelacakan yang ada belum cukup efektif.
4. Kurangnya pemanfaatan teknologi keamanan kendaraan yang mudah dipasang dan tidak mengubah sistem kelistrikan kendaraan (*non-invasive*) menyebabkan tingkat adopsi perangkat pelacak masih rendah, sehingga peluang menemukan kendaraan setelah dicuri menjadi lebih kecil.

1.2.2 Rumusan Masalah

1. Bagaimana cara mencegah terjadinya pencurian kendaraan bermotor melalui teknologi pemantauan lokasi secara *real-time*?

2. Bagaimana cara menemukan kendaraan bermotor yang telah dicuri dengan memanfaatkan sistem pelacakan berbasis GPS?
3. Bagaimana cara menerapkan fitur pembatasan wilayah administratif (*geofencing*) pada perangkat pelacak kendaraan bermotor?
4. Bagaimana merancang perangkat pelacak kendaraan yang bersifat portabel dan tidak merusak (*non-invasive*) sehingga sulit dideteksi oleh pelaku pencurian?
5. Bagaimana merancang perangkat pelacak kendaraan yang mudah digunakan, terjangkau, dan dapat mendorong adopsi yang lebih luas di masyarakat?

1.3 Tujuan Tugas Akhir

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka tujuan dari pelaksanaan Tugas Akhir ini adalah sebagai berikut:

1. Mengembangkan sistem pemantauan lokasi kendaraan bermotor secara *real-time* sebagai upaya pencegahan pencurian.
2. Mengembangkan sistem pelacakan kendaraan bermotor berbasis GPS untuk memudahkan proses pencarian kendaraan yang dicuri.
3. Mengembangkan fitur pembatasan wilayah administratif (*geofencing*) sebagai mekanisme peringatan dini ketika kendaraan melampaui batas wilayah yang ditentukan.
4. Merancang perangkat pelacak kendaraan yang bersifat portabel dan tidak merusak (*non-invasive*) agar tidak mudah dikenali dan dinonaktifkan oleh pelaku pencurian.
5. Menghasilkan perangkat pelacak kendaraan yang mudah digunakan (*user-friendly*), terjangkau, dan dapat diadopsi secara luas oleh masyarakat.

1.4 Lingkup Tugas Akhir

Ruang lingkup dan batasan penelitian ditetapkan agar pelaksanaan Tugas Akhir ini tetap fokus dan terarah, sebagaimana dijelaskan pada sub-bab berikut.

1.4.1 Ruang Lingkup

Untuk menjaga fokus penelitian dan kejelasan ruang lingkup pengembangan, tugas akhir ini mencakup hal-hal sebagai berikut:

1. Perancangan dan pengembangan aplikasi website pemantauan lokasi kendaraan bermotor secara real-time berbasis web.
2. Implementasi fitur geofencing berdasarkan wilayah administratif (provinsi, kabupaten/kota, kecamatan, dan kelurahan) pada website.
3. Implementasi pengiriman notifikasi otomatis melalui email dan WhatsApp menggunakan API Fonnte ketika kendaraan melanggar zona geofencing.

4. Implementasi fitur riwayat perjalanan kendaraan yang dapat dipantau melalui website.
5. Implementasi sistem autentikasi pengguna dan pengelolaan data kendaraan pada website.
6. Implementasi tampilan status perangkat GPS secara real-time pada website

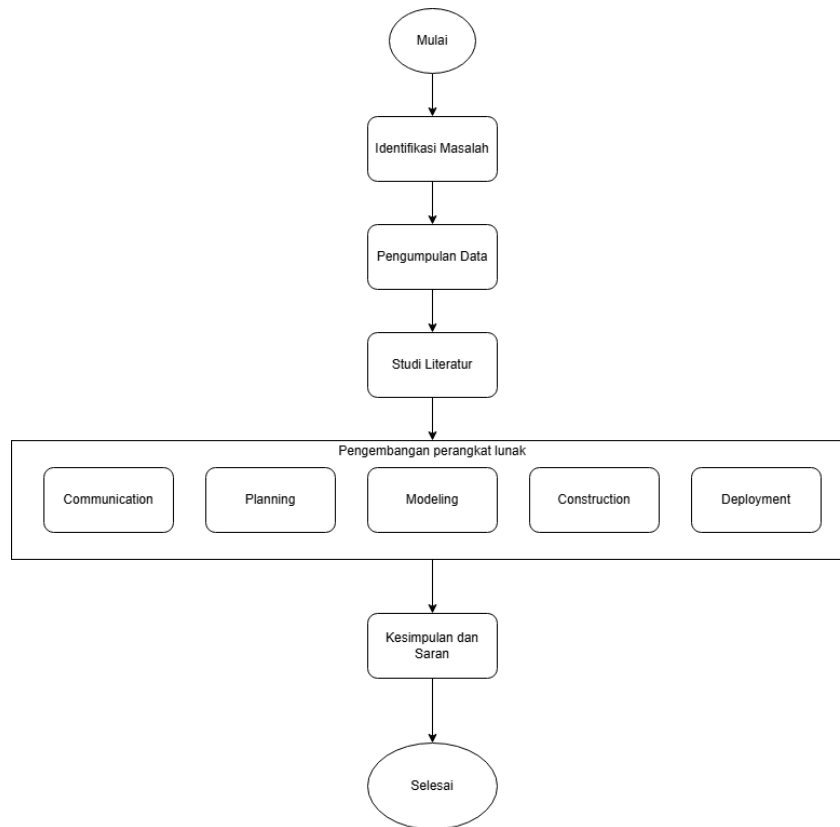
1.4.2 Batasan

Adapun batasan dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada pengembangan sisi website, tidak mencakup perancangan dan pembuatan perangkat keras GPS tracker.
2. Perancangan dan pembuatan perangkat keras GPS tracker merupakan bagian dari penelitian rekan satu tim.
3. Sistem hanya dapat diakses melalui website, tidak tersedia dalam bentuk aplikasi mobile.
4. Notifikasi WhatsApp hanya dapat dikirimkan melalui layanan API Fonnte.
5. Sistem tidak mencakup fitur pemblokiran atau immobilisasi kendaraan secara remote.
6. Sistem tidak terintegrasi dengan instansi kepolisian atau pihak berwenang secara langsung.
7. Data wilayah administratif yang digunakan terbatas pada data GeoJSON yang tersedia.
8. Akurasi dan performa pengiriman data koordinat bergantung pada perangkat keras yang dikembangkan oleh rekan satu tim

1.5 Metodologi Tugas Akhir

Penyelesaian tugas akhir ini dilakukan melalui beberapa tahapan yang saling berkaitan, mulai dari identifikasi masalah hingga pengujian sistem, sebagaimana digambarkan pada alur metodologi berikut.



Gambar 1.1. Metodologi Penyelesaian Tugas Akhir

Metodologi pengembangan yang digunakan dalam tugas akhir ini mengadaptasi model pengembangan perangkat lunak dari Pressman [PRS20] yang mencakup tahapan Communication, Planning, Modeling, Construction, dan Deployment. Metode ini dipilih karena kesesuaiannya dengan proyek pengembangan sistem yang terdiri dari dua bagian yang saling terintegrasi, yaitu pengembangan perangkat keras GPS tracker dan pengembangan website pemantauan lokasi kendaraan bermotor. Pengumpulan data dilakukan melalui studi literatur berupa pencarian dan perbandingan referensi dari jurnal ilmiah, prosiding, dan buku yang relevan dengan permasalahan yang diidentifikasi. Berikut merupakan rincian setiap tahapan metodologi yang diterapkan dalam penelitian ini:

1. Identifikasi Masalah

Pada tahap ini dilakukan pengidentifikasian permasalahan yang melatarbelakangi penelitian ini. Permasalahan diidentifikasi berdasarkan fakta lapangan mengenai tingginya angka pencurian kendaraan bermotor di wilayah perkotaan, keterbatasan perangkat GPS konvensional yang bersifat invasif dan mudah dinonaktifkan oleh pelaku pencurian, serta belum tersedianya sistem pemantauan kendaraan yang dilengkapi fitur geofencing berbasis wilayah administratif.

2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang relevan melalui studi literatur. Studi literatur dilakukan dengan mencari dan membandingkan referensi dari jurnal

ilmiah, prosiding, dan buku yang berkaitan dengan mikrokontroler ESP32, modul GPS NEO-6M, modul SIM800L, protokol MQTT, sistem manajemen daya berbasis baterai lithium, teknologi GPS, geofencing berbasis wilayah administratif, WebSocket, Leaflet.js, dan reverse geocoding.

3. Pembuatan Perangkat Lunak

Tahap pengembangan perangkat lunak mengikuti fase-fase dari model Pressman [PRS20] sebagai berikut:

- a. *Communication*: Fase ini diawali dengan inisiasi serta pendefinisian seluruh kebutuhan sistem. Kebutuhan yang teridentifikasi meliputi dua bagian pokok, yakni kebutuhan perangkat keras GPS tracker yang portabel dan non-invasive, dan kebutuhan website pemantauan lokasi kendaraan bermotor yang dilengkapi geofencing berbasis wilayah administratif beserta notifikasi otomatis melalui email dan WhatsApp.
- b. *Planning*: Fase ini menyusun estimasi waktu pengembangan sekaligus menetapkan prioritas komponen dan fitur sistem. Kegiatan perencanaan meliputi penentuan komponen elektronik untuk perangkat keras serta pemilihan teknologi dan framework yang akan dipakai membangun website.
- c. *Modeling*: Fase ini menyusun rancangan arsitektur sistem secara menyeluruh. Cakupannya meliputi skema rangkaian elektronik perangkat keras, alur pengiriman data dari perangkat menuju server, use case diagram, sequence diagram, entity relationship diagram (ERD), dan rancangan antarmuka pengguna website.
- d. *Construction*: Fase ini mencakup perakitan komponen perangkat keras sekaligus penulisan seluruh kode program sistem. Pada sisi perangkat keras, kegiatan yang dilakukan meliputi perakitan rangkaian elektronik, penulisan firmware ESP32 melalui Arduino IDE, serta pemanfaatan library TinyGPS++ dan PubSubClient. Adapun pada sisi website, kegiatan mencakup pengembangan backend dengan Laravel, penyambungan WebSocket dan Pusher untuk pemantauan real-time, penerapan geofencing berbasis GeoJSON bersama algoritma Point in Polygon, dan penyiapan notifikasi melalui SMTP Gmail serta API Fonnte.
- e. *Deployment*: Fase ini memindahkan seluruh sistem yang telah rampung dikembangkan dan diuji ke lingkungan produksi. Rangkaian kegiatannya meliputi konfigurasi web server, penyiapan basis data PostgreSQL, konfigurasi MQTT Broker EMQX, dan pengujian integrasi antara perangkat keras dengan website pemantauan pada kondisi nyata di kendaraan bermotor.

4. Kesimpulan dan Saran

Pada tahap ini dilakukan penyimpulan dari seluruh hasil penelitian yang telah dilakukan. Kesimpulan disusun sebagai jawaban langsung terhadap setiap tujuan penelitian yang telah ditetapkan pada BAB I, serta saran sebagai rekomendasi pengembangan sistem lebih lanjut yang dapat dilakukan oleh peneliti berikutnya.

1.6 Sistematika Penulisan Laporan Tugas Akhir

Laporan tugas akhir dibuat untuk mendokumentasikan pengerjaan tugas akhir, maka dari itu diusulkan sistematika penulisan yang menjelaskan mengenai bab-bab pada laporan tugas akhir beserta isinya secara rinci, serta keterkaitan antara bab sebelum dan sesudahnya. Adapun sistematik penulisannya adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memaparkan gambaran umum penelitian yang dikerjakan, mulai dari latar belakang yang mendasari pemilihan topik, identifikasi dan rumusan masalah, tujuan yang hendak dicapai, ruang lingkup beserta batasannya, metodologi pengerjaan tugas akhir, hingga sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini memuat landasan konseptual yang menopang perancangan sistem, mencakup konsep sistem dan aplikasi berbasis web, Internet of Things, Global Positioning System, protokol MQTT, framework Laravel, PostgreSQL, WebSocket dan Pusher, Leaflet.js, GeoJSON, geofencing, algoritma Point in Polygon, reverse geocoding, notifikasi email dan WhatsApp, pemantauan real-time, black box testing, serta User Acceptance Testing. Pada bagian akhir bab ini dibahas pula penelitian terdahulu yang memiliki keterkaitan dengan tugas akhir ini.

BAB III SKEMA PENELITIAN

Bab ini menguraikan tahapan analisis yang menjadi dasar pengembangan sistem, meliputi alur penyelesaian tugas akhir, perumusan masalah melalui analisis sebab akibat dan analisis solusi masalah, kerangka pemikiran teoritis, objek penelitian, kebutuhan pengguna beserta aktornya, kebutuhan perangkat keras dan perangkat lunak, kebutuhan fungsional dan non-fungsional, kebutuhan data, serta kebutuhan pengembangan dan deployment sistem.

BAB IV PERANCANGAN SISTEM

Bab ini menyajikan rancangan sistem yang disusun berdasarkan hasil analisis pada bab sebelumnya. Pembahasan mencakup pemodelan berbasis skenario berupa business use case, use case, diagram sequence, dan skenario use case, pemodelan data melalui class diagram dan entity

relationship diagram, perancangan perangkat keras, arsitektur sistem, serta rancangan antarmuka website.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas penerapan hasil perancangan menjadi sistem yang siap digunakan, mulai dari struktur proyek, implementasi antarmuka perangkat lunak dan perangkat keras, kode program, hingga basis data. Pada bab ini pula dipaparkan rencana dan hasil pengujian perangkat lunak maupun perangkat keras, rangkuman hasil pengujian, serta pengujian User Acceptance Testing terhadap pengguna dan admin.

BAB VI KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan yang menjawab setiap tujuan penelitian yang telah ditetapkan pada BAB I, disertai saran sebagai bahan pertimbangan bagi pengembangan sistem pada penelitian selanjutnya.

BAB 2

LANDASAN TEORI

Dalam bab ini dibahas teori dan konsep dasar yang diperlukan untuk menganalisis, merancang, dan mengimplementasikan sistem yang menjadi objek penelitian. Teori-teori yang dipaparkan bersumber dari buku teks, jurnal ilmiah, serta dokumentasi resmi yang berkaitan langsung dengan teknologi dan metode yang digunakan dalam pengembangan Perangkat GPS Tracker Portabel dan Non-Invasive dengan Implementasi Geofencing Berbasis Wilayah Administratif untuk Sistem Keamanan Kendaraan Bermotor.

2.1 Teori Pendukung

Sub-bab berikut menguraikan sejumlah konsep dan teknologi pendukung yang menjadi dasar dalam menganalisis, merancang, dan membangun sistem AF GPS Tracker.

2.1.1 Konsep Dasar Sistem

Sistem dapat diartikan sebagai sekumpulan elemen maupun komponen yang berinteraksi serta saling bekerja sama guna mewujudkan tujuan tertentu [SOM16]. Menurut Sommerville [SOM16], sistem perangkat lunak ialah program komputer bersama dokumentasi yang menyertainya, baik berupa satu program tunggal ataupun himpunan program yang menyediakan fungsionalitas bagi pengguna yang beragam. Sementara itu, Pressman dan Maxim [PRS20] mengemukakan bahwa suatu sistem tersusun atas masukan (input), proses (process), keluaran (output), serta umpan balik (feedback) yang bersama-sama membentuk siklus kerja berkelanjutan.

Pada sistem yang menggabungkan perangkat lunak dan perangkat keras, terdapat sejumlah komponen yang saling menopang, mencakup perangkat keras (hardware), perangkat lunak (software), data, pengguna (user), serta prosedur [PRS20]. Masing-masing komponen menjalankan peran dan fungsi tersendiri demi tercapainya tujuan sistem, sehingga kinerja sistem secara keseluruhan turut terpengaruh apabila ada satu komponen yang gagal berfungsi. Karena itu, keterkaitan antarkomponen perlu dipertimbangkan sejak tahap analisis kebutuhan ketika sistem dirancang.

2.1.2 Aplikasi Berbasis Web

Perangkat lunak yang beroperasi melalui jaringan internet maupun intranet serta memanfaatkan browser web sebagai antarmuka bagi penggunaanya disebut aplikasi berbasis web (web-based application). Aplikasi desktop dan aplikasi mobile menuntut pengguna memasangnya terlebih dahulu pada masing-masing perangkat, sedangkan aplikasi berbasis web bisa dibuka langsung lewat browser tanpa tahap pemasangan

tambahan sehingga pengelolaannya lebih luwes dan pembaruan dapat dilakukan secara terpusat oleh pengembang.

Arsitektur aplikasi berbasis web pada umumnya mengikuti model client-server, di mana sisi client (browser) bertugas menampilkan antarmuka pengguna dan mengirimkan permintaan, sementara sisi server bertugas memproses permintaan tersebut dan mengembalikan respons berupa data atau halaman web [SOM16]. Model ini memungkinkan pemrosesan data yang berat dilakukan di sisi server sehingga perangkat pengguna tidak perlu memiliki spesifikasi tinggi untuk dapat menjalankan aplikasi tersebut dengan lancar.

2.1.2.1. Keunggulan Aplikasi Berbasis Web

Aksesibilitas dan kemudahan pemeliharaan termasuk keunggulan pokok aplikasi berbasis web yang dipertimbangkan pada penelitian ini. Dua aspek tersebut dinilai relevan karena calon pengguna sistem ini adalah masyarakat umum dengan latar belakang teknis yang beragam, sehingga kemudahan mengakses sistem menentukan sejauh mana sistem dapat diterima.

- **Aksesibilitas Universal:** Sistem dapat dibuka dari beragam perangkat, mulai smartphone, tablet, sampai laptop, cukup bermodal browser dan tanpa pemasangan aplikasi, sehingga sejalan dengan tujuan penelitian untuk memperluas penerimaan GPS tracker di masyarakat.
- **Pembaruan Terpusat:** Perbaikan maupun penambahan fitur dilakukan sekali di sisi server dan seketika itu juga berlaku bagi seluruh pengguna, tanpa menuntut pembaruan manual pada masing-masing perangkat.
- **Kompatibilitas Lintas Platform:** Sistem dapat dijalankan pada aneka sistem operasi, antara lain Android, iOS, Windows, macOS, dan Linux, asalkan perangkatnya tersedia browser versi terbaru.
- **Integrasi Real-Time:** Sistem mudah disambungkan dengan layanan WebSocket semacam Pusher agar pemantauan lokasi kendaraan berlangsung real-time, tanpa konfigurasi rumit yang menyertainya.
- **Biaya Pengembangan yang Efisien:** Satu basis kode sanggup melayani bermacam jenis perangkat sehingga tidak diperlukan pembuatan aplikasi native terpisah untuk Android maupun iOS, yang berarti waktu dan biaya pengembangan dapat ditekan secara berarti.

Pada penelitian ini, sistem pemantauan lokasi kendaraan dikembangkan sepenuhnya sebagai aplikasi berbasis web menggunakan framework Laravel pada sisi server dan Leaflet.js pada sisi client. Pemilihan pendekatan berbasis web

dilandasi oleh pertimbangan aksesibilitas lintas perangkat, kemudahan pemeliharaan, serta kemampuan integrasi real-time dengan layanan WebSocket Pusher yang dapat menyinkronkan posisi kendaraan secara instan ke browser pengguna tanpa perlu melakukan reload halaman secara manual.

2.1.3 Internet of Things (IoT)

Paradigma teknologi yang membuat perangkat fisik dapat terhubung satu sama lain serta bertukar informasi melalui jaringan internet dikenal sebagai Internet of Things (IoT), sehingga data mampu dikumpulkan, dibagikan, dan diolah secara otomatis tanpa campur tangan manusia secara langsung [BAH14]. Pada umumnya perangkat IoT dibekali sensor, aktuator, dan kemampuan berkomunikasi melalui jaringan sehingga dapat berinteraksi dengan lingkungan di sekitarnya secara real-time; dengan begitu data yang dihasilkan senantiasa menggambarkan keadaan terkini objek yang dipantau.

Penerapan konsep IoT semakin meluas dalam berbagai bidang termasuk transportasi, kesehatan, dan keamanan, salah satunya melalui perangkat pelacak kendaraan yang menjadi fokus pada penelitian ini. Keberhasilan implementasi IoT sangat bergantung pada keselarasan antara karakteristik perangkat keras yang digunakan dengan protokol komunikasi yang dipilih, terutama pada perangkat yang beroperasi dengan keterbatasan daya dan bandwidth jaringan.

2.1.3.1. Arsitektur Sistem IoT

Arsitektur sistem IoT secara umum terdiri dari tiga lapisan utama yang bekerja secara berurutan dan saling melengkapi satu sama lain [BAH14]. Ketiga lapisan ini memastikan bahwa data yang dikumpulkan dari dunia fisik dapat diteruskan dan diolah hingga menjadi informasi yang bermanfaat bagi pengguna akhir.

- Lapisan Persepsi (Perception Layer): Terdiri dari sensor dan aktuator yang bertugas mengumpulkan data langsung dari lingkungan fisik. Pada penelitian ini, lapisan ini direpresentasikan oleh modul GPS NEO-6M yang membaca koordinat lokasi kendaraan secara berkala.
- Lapisan Jaringan (Network Layer): Bertugas mentransmisikan data yang dikumpulkan dari lapisan persepsi ke lapisan aplikasi melalui protokol komunikasi yang sesuai dengan karakteristik jaringan yang tersedia. Pada penelitian ini, lapisan ini direpresentasikan oleh modul SIM800L yang mengirim data melalui jaringan GSM menggunakan protokol MQTT.
- Lapisan Aplikasi (Application Layer): Bertugas memproses, menyimpan, dan menyajikan data kepada pengguna akhir dalam format yang berguna dan

mudah dipahami. Pada penelitian ini, lapisan ini direpresentasikan oleh backend Laravel dan antarmuka website yang menampilkan posisi kendaraan serta menjalankan logika geofencing.

Dalam penelitian ini, perangkat GPS tracker yang dikembangkan oleh Fakih merupakan implementasi nyata dari konsep IoT, di mana komponen fisik bekerja secara sinergis untuk mengumpulkan data koordinat lokasi kendaraan dan mengirimkannya ke server secara berkala. Data tersebut kemudian divisualisasikan dan diolah lebih lanjut pada website pemantauan yang dikembangkan oleh Aldi, sehingga membentuk satu rangkaian sistem IoT yang utuh mulai dari pengumpulan data hingga penyajian informasi kepada pengguna.

2.1.4 Global Positioning System (GPS)

Global Positioning System (GPS) adalah sistem navigasi yang memanfaatkan satelit untuk menetapkan posisi objek di permukaan bumi dengan tingkat akurasi tinggi [KAP96]. Departemen Pertahanan Amerika Serikat mengembangkan sistem ini, dan kini pemanfaatannya telah meluas ke banyak bidang seperti navigasi, transportasi, logistik, pemetaan, juga sistem pelacakan kendaraan.

Kaplan mengemukakan bahwa GPS tersusun dari tiga segmen pokok, yakni segmen angkasa (space segment), lalu segmen kontrol (control segment), serta segmen pengguna (user segment) [KAP96]. Segmen angkasa mencakup sejumlah satelit yang beredar mengelilingi bumi sambil memancarkan sinyal navigasi. Adapun segmen kontrol bertugas memantau sekaligus mengendalikan operasional satelit. Sementara segmen pengguna berupa perangkat penerima GPS yang berfungsi menangkap sinyal dari satelit dan menghitung posisi objek.

Dalam beroperasi GPS memanfaatkan prinsip trilaterasi, yakni penentuan posisi yang bertumpu pada pengukuran jarak ke sejumlah satelit. Untuk memperoleh posisi dua dimensi, perangkat GPS perlu menangkap sinyal dari minimal tiga satelit, sedangkan posisi tiga dimensi yang lebih akurat membutuhkan minimal empat satelit [KAP96]. Perhitungan posisi itu kemudian dituangkan dalam bentuk koordinat geografis, yaitu latitude dan longitude.

Latitude adalah koordinat yang menandai letak di sebelah utara atau selatan garis khatulistiwa, sementara longitude menandai letak di sebelah timur atau barat garis meridian utama. Penggabungan kedua koordinat tersebut dipakai untuk merepresentasikan lokasi sebuah objek di permukaan bumi.

Dalam penelitian ini, data koordinat GPS digunakan sebagai sumber utama informasi lokasi kendaraan. Data yang diterima server digunakan untuk menampilkan

posisi kendaraan pada peta digital, menentukan status *geofencing*, menyimpan riwayat perjalanan kendaraan, serta menghasilkan notifikasi ketika kendaraan keluar dari wilayah administratif yang telah ditentukan oleh pengguna.

2.1.5 MQTT (Message Queuing Telemetry Transport)

MQTT merupakan protokol komunikasi dengan basis publish-subscribe yang memang dirancang bagi lingkungan berbandwidth dan berdaya terbatas, misalnya perangkat IoT [OAS14]. IBM yang mengembangkan protokol ini, dan kini statusnya telah ditetapkan sebagai standar OASIS; protokol berjalan di atas TCP/IP dengan overhead header hanya 2 byte sehingga pengiriman data berukuran kecil secara berkala menjadi sangat efisien.

Karakteristik efisiensi MQTT menjadi alasan utama pemilihannya sebagai protokol komunikasi antara perangkat GPS tracker dan server pada penelitian ini. Mengingat perangkat menggunakan modul SIM800L yang mengakses internet melalui jaringan GSM dengan bandwidth GPRS yang relatif terbatas, penggunaan protokol dengan overhead besar seperti HTTP REST API akan menghabiskan kuota data dan daya baterai secara tidak efisien.

2.1.5.1. Arsitektur Publish-Subscribe

Arsitektur MQTT terdiri dari tiga komponen utama yang masing-masing memiliki peran berbeda namun saling terhubung dalam satu alur komunikasi data yang utuh [OAS14].

- **Publisher:** Komponen yang mengirimkan pesan ke Broker dengan menyertakan topik tertentu. Pada penelitian ini, ESP32 berperan sebagai Publisher yang mengirim data koordinat secara berkala.
- **Broker:** Komponen perantara yang menerima pesan dari Publisher dan mendistribusikannya ke seluruh Subscriber yang berlangganan topik yang sama. Pada penelitian ini digunakan MQTT Broker publik EMQX (broker.emqx.io).
- **Subscriber:** Komponen yang menerima pesan dari Broker berdasarkan topik yang dilanggan. Pada penelitian ini, backend Laravel berperan sebagai Subscriber yang menerima data koordinat untuk diproses lebih lanjut.

2.1.5.2. Penerapan MQTT pada Sistem

Dalam penelitian ini, ESP32 pada perangkat GPS tracker yang dikembangkan oleh Fakih mengirimkan data koordinat berformat JSON ke MQTT Broker secara berkala dengan interval yang dapat dikonfigurasi sesuai kebutuhan.

Setiap pesan yang dikirimkan berisi informasi identitas perangkat, koordinat latitude dan longitude, serta timestamp pengiriman data.

Backend Laravel yang dikembangkan oleh Aldi menerima data tersebut sebagai Subscriber untuk kemudian disimpan ke database PostgreSQL, diteruskan ke browser melalui Pusher untuk ditampilkan secara real-time, serta digunakan sebagai masukan utama bagi algoritma geofencing. Dengan demikian, MQTT menjadi jembatan komunikasi yang menghubungkan secara langsung antara hasil kerja perangkat keras yang dikembangkan oleh Fakih dengan logika pemrosesan yang dikembangkan oleh Aldi.

2.1.6 Laravel

Laravel ialah framework untuk membangun aplikasi web dengan bahasa pemrograman PHP yang menerapkan pola arsitektur Model-View-Controller (MVC), dan dikembangkan oleh Taylor Otwell mulai tahun 2011 [OTW23]. Filosofi pengembangan framework ini mengutamakan kemudahan, keeleganan kode, dan produktivitas pengembang, yang didukung beragam fitur bawaan bersifat komprehensif.

Pemilihan Laravel sebagai framework backend pada penelitian ini didasarkan pada kebutuhan untuk mengembangkan sistem yang kompleks dalam waktu yang relatif singkat. Berbagai fitur bawaan yang disediakan Laravel memungkinkan pengembang untuk fokus pada logika bisnis spesifik penelitian ini, seperti algoritma geofencing, tanpa perlu membangun fungsi dasar seperti autentikasi pengguna dari awal.

2.1.6.1. Fitur Bawaan Laravel yang Digunakan

Beberapa fitur Laravel yang dimanfaatkan secara langsung dalam pengembangan website pemantauan pada penelitian ini saling terintegrasi membentuk satu kesatuan sistem yang utuh.

- Eloquent ORM: Antarmuka berbasis objek untuk berinteraksi dengan database PostgreSQL tanpa menulis query SQL secara manual, digunakan untuk mengelola data kendaraan, pengguna, dan riwayat koordinat.
- Blade Templating Engine: Sistem templat untuk membangun tampilan antarmuka website pemantauan secara dinamis dan terstruktur dengan baik.
- Laravel Auth: Sistem autentikasi bawaan yang mengelola login dan keamanan akun pengguna secara lengkap dan teruji.
- Laravel Broadcasting: Fitur yang memungkinkan pengiriman data secara real-time ke browser melalui integrasi langsung dengan Pusher.

- **Laravel HTTP Client:** Digunakan untuk melakukan request ke layanan eksternal seperti Nominatim untuk reverse geocoding dan API Fonnte untuk notifikasi WhatsApp.

Dalam penelitian ini, Laravel digunakan oleh Aldi sebagai framework utama backend website pemantauan, yang menangani penerimaan data MQTT, penyimpanan ke database, eksekusi algoritma geofencing, serta pengiriman notifikasi. Seluruh fitur bawaan tersebut dimanfaatkan secara optimal untuk mempercepat proses pengembangan tanpa mengorbankan kualitas dan keamanan sistem yang dihasilkan.

2.1.7 PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional (RDBMS) bersifat open-source yang dipakai untuk mengelola data secara terstruktur. Bahasa SQL menjadi antarmuka standar pada PostgreSQL untuk menjalankan operasi pengambilan, penyimpanan, pembaruan, serta penghapusan data di dalam basis data.

Pemilihan PostgreSQL dalam penelitian ini didasarkan pada karakteristik data yang dikelola, di mana terdapat hubungan relasional yang jelas antara satu entitas dengan entitas lainnya. Pada sistem AF GPS Tracker, satu pengguna dapat memiliki beberapa perangkat GPS dan kendaraan, sedangkan setiap perangkat dapat menghasilkan banyak data log dan riwayat perjalanan. Selain itu, data kendaraan juga memiliki keterkaitan dengan data geofencing, riwayat berhenti, dan notifikasi. Oleh karena itu, PostgreSQL digunakan untuk merepresentasikan dan mengelola hubungan antarentitas tersebut secara terstruktur.

2.1.8 WebSocket dan Pusher

WebSocket merupakan protokol komunikasi yang membuka saluran dua arah bersifat persisten antara browser dan server lewat satu koneksi TCP, dan telah distandarisasi melalui RFC 6455 [FET11]. Berbeda dari model request-response yang dipakai protokol HTTP konvensional, protokol ini membuat server dapat mengirim data ke browser secara langsung tanpa menunggu permintaan lebih dahulu dari browser.

Penerapan WebSocket menjadi kebutuhan mutlak dalam penelitian ini mengingat tujuan utama sistem adalah menyajikan informasi pemantauan secara real-time. Tanpa WebSocket, satu-satunya cara untuk menampilkan data terbaru adalah melalui teknik polling yang membebani server dengan permintaan berulang dan menimbulkan jeda waktu yang tidak diinginkan antara data tersedia dan ditampilkan kepada pengguna.

2.1.8.1. Mekanisme Komunikasi Real-Time

Pusher menyediakan infrastruktur WebSocket sebagai layanan (WebSocket-as-a-Service) yang terintegrasi langsung dengan Laravel Broadcasting, sehingga pengembang tidak perlu mengelola infrastruktur WebSocket server secara mandiri [PUS23]. Layanan ini menangani seluruh aspek teknis terkait manajemen koneksi, scaling, dan distribusi pesan ke seluruh client yang terhubung pada saat yang bersamaan.

Dalam sistem yang dikembangkan, mekanisme ini dimanfaatkan untuk dua fungsi utama yang saling berkaitan satu sama lain dalam mendukung pengalaman pemantauan yang responsif bagi pengguna.

- **Update Posisi Real-Time:** Ketika data koordinat baru diterima dari perangkat GPS, posisi marker kendaraan pada peta langsung diperbarui di browser tanpa refresh halaman.
- **Notifikasi Status Geofencing:** Perubahan status kendaraan, baik berada di dalam maupun di luar zona, ditampilkan secara langsung pada dashboard pengguna.

Dalam penelitian ini, WebSocket dan Pusher digunakan oleh Aldi untuk mengimplementasikan fitur pemantauan lokasi kendaraan secara real-time pada website. Implementasi ini memungkinkan setiap data yang dikirimkan oleh perangkat GPS tracker milik Fakih dapat langsung terlihat oleh pengguna dalam hitungan detik tanpa adanya jeda yang berarti.

2.1.9 Leaflet.js

Leaflet.js merupakan library JavaScript bersumber terbuka yang dipakai untuk membangun peta interaktif pada halaman web, dikembangkan oleh Vladimir Agafonkin dengan mengedepankan kesederhanaan dan kinerja tinggi [AGA11]. Ukurannya pun tergolong sangat ringan, sekitar 42 KB pada versi terkompresi, sehingga tidak membuat waktu muat halaman website bertambah secara berarti.

Pemilihan Leaflet.js dalam penelitian ini didasarkan pada kemampuannya untuk merender data GeoJSON secara langsung tanpa memerlukan konversi format tambahan. Karakteristik ini sangat relevan mengingat zona geofencing berbasis wilayah administratif dalam penelitian ini direpresentasikan dalam format GeoJSON, sehingga integrasi antara data dan visualisasi dapat dilakukan secara efisien.

2.1.9.1. Fitur Leaflet.js yang Digunakan

Beberapa fitur Leaflet.js yang dimanfaatkan dalam penelitian ini bekerja secara bersamaan untuk menghasilkan tampilan peta yang informatif dan interaktif bagi pengguna.

- Marker: Digunakan untuk menandai posisi kendaraan secara real-time pada peta berdasarkan data koordinat terbaru yang diterima dari server.
- Polyline: Digunakan untuk menggambar jalur riwayat perjalanan kendaraan berdasarkan kumpulan koordinat historis yang tersimpan di database.
- L.geoJSON(): Digunakan untuk merender langsung data GeoJSON zona geofencing wilayah administratif ke dalam layer peta interaktif tanpa konversi tambahan.
- Event Handler: Digunakan untuk menangkap interaksi pengguna saat memilih wilayah zona geofencing melalui klik langsung pada peta.

Dalam penelitian ini, Leaflet.js digunakan oleh Aldi sebagai library utama visualisasi peta pada website pemantauan lokasi kendaraan bermotor. Seluruh fitur tersebut bekerja secara terintegrasi untuk menghasilkan satu tampilan peta yang dapat menampilkan posisi kendaraan, riwayat perjalanan, dan zona geofencing secara bersamaan dalam satu antarmuka yang mudah dipahami pengguna.

2.1.10 GeoJSON

GeoJSON ialah format standar berbasis JSON yang dipakai untuk merepresentasikan data geografis, meliputi titik, garis, dan poligon beserta atribut-atributnya; penetapannya dilakukan oleh IETF lewat RFC 7946 sehingga format ini menjadi standar resmi bagi pertukaran data geografis di web [RFC16]. Alasan pemilihan format ini adalah kompatibilitasnya yang luas terhadap hampir semua library pemetaan modern, di antaranya Leaflet.js yang dipakai dalam penelitian ini.

Pemanfaatan GeoJSON menjadi fondasi utama dalam implementasi fitur geofencing berbasis wilayah administratif pada penelitian ini. Tanpa format standar seperti GeoJSON, proses pertukaran data geografis antara sumber data wilayah administratif, basis data, dan tampilan peta akan memerlukan konversi format yang rumit dan rentan terhadap kesalahan.

2.1.10.1. Struktur GeoJSON

GeoJSON menyediakan sejumlah tipe geometri, dan setiap tipe dipakai untuk merepresentasikan jenis data geografis yang berbeda pada penelitian ini [RFC16].

- Point: Sebuah titik koordinat tunggal yang merepresentasikan posisi kendaraan pada satu waktu tertentu.
- LineString: Rangkaian titik yang tersambung berurutan menjadi sebuah garis, dipakai untuk merepresentasikan riwayat perjalanan kendaraan.
- Polygon: Area tertutup hasil penyambungan sejumlah titik koordinat, dipakai untuk merepresentasikan batas wilayah administratif.
- MultiPolygon: Himpunan beberapa polygon yang saling terpisah, dipakai untuk kawasan yang letaknya berjauhan secara geografis seperti gugusan kepulauan.

2.1.11 Geofencing

Geofencing merupakan teknologi yang memungkinkan penentuan batas wilayah virtual pada suatu area geografis tertentu, sehingga saat sebuah objek masuk atau keluar dari batas yang telah ditetapkan, sistem mampu mendeteksi dan meresponsnya secara otomatis [KUP11]. Penerapan teknologi ini cukup luas, mencakup sistem keamanan kendaraan, manajemen armada, sampai pemasaran berbasis lokasi.

Penerapan geofencing dalam konteks keamanan kendaraan bermotor memungkinkan pemilik kendaraan untuk menerima peringatan secara proaktif tanpa perlu memantau posisi kendaraannya secara manual setiap saat. Karakteristik proaktif inilah yang membedakan sistem yang dikembangkan dalam penelitian ini dari sistem pelacakan konvensional yang hanya bersifat pasif.

2.1.11.1. Geofencing Radius

Geofencing radius adalah implementasi geofencing yang menetapkan satu titik koordinat yang berperan sebagai pusat serta nilai jarak bersatuan meter sebagai radius, sehingga terbentuk zona virtual berupa lingkaran [KUP11]. Pendeteksian posisi objek pada zona radius memanfaatkan rumus Haversine, yang mengukur jarak sesungguhnya di antara dua titik koordinat sambil memperhitungkan kelengkungan permukaan bumi.

Kelemahan pokok geofencing radius terletak pada bentuk zona yang dihasilkan, yakni selalu berupa lingkaran sempurna sehingga tidak mampu menggambarkan batas wilayah nyata sebagaimana yang dipahami masyarakat umum. Selain itu, pengguna dituntut menetapkan koordinat titik pusat sekaligus nilai radius dalam meter, sesuatu yang terasa kurang intuitif bagi masyarakat awam yang belum memahami sistem koordinat geografis.

2.1.11.2. Geofencing Polygon

Implementasi geofencing yang memakai rangkaian titik koordinat untuk membentuk area tertutup tidak beraturan sebagai penanda batas zona virtual dinamakan geofencing polygon [KUP11]. Melalui pendekatan ini, zona yang terbentuk dapat mengikuti bentuk wilayah tidak beraturan apa pun, sehingga selaras dengan batas wilayah administratif yang sesungguhnya di lapangan.

Pendeteksian posisi objek terhadap zona polygon memerlukan algoritma Point in Polygon yang akan dibahas lebih lanjut pada subbab berikutnya. Meskipun secara komputasional lebih kompleks dibanding geofencing radius, pendekatan ini menghasilkan deteksi posisi yang jauh lebih akurat dan relevan untuk kebutuhan geofencing berbasis wilayah administratif.

Dalam penelitian ini, Aldi mengimplementasikan geofencing polygon menggunakan data GeoJSON batas wilayah administratif Indonesia, sehingga pengguna dapat menetapkan zona aman berdasarkan nama wilayah yang familiar seperti kota. Pendekatan ini dipilih karena jauh lebih intuitif bagi pemilik kendaraan dibanding harus menentukan titik koordinat dan radius dalam meter yang tidak familiar bagi masyarakat umum.

2.1.12 Point In Polygon

Point in Polygon (PIP) merupakan algoritma pada bidang geometri komputasional yang dipakai untuk menetapkan apakah suatu titik koordinat terletak di dalam maupun di luar suatu polygon tertutup [ORO98]. Dalam implementasi geofencing polygon, algoritma ini berperan sebagai komponen fundamental sekaligus mekanisme deteksi utama atas posisi kendaraan yang sedang dipantau.

Pemilihan algoritma yang tepat untuk menyelesaikan permasalahan Point in Polygon sangat penting mengingat algoritma ini akan dijalankan secara berulang setiap kali data koordinat baru diterima dari perangkat GPS tracker. Efisiensi komputasi menjadi pertimbangan utama, terutama jika sistem diharapkan dapat melayani banyak kendaraan secara bersamaan tanpa mengalami penurunan performa yang signifikan.

2.1.12.1. Mekanisme Ray Casting Algorithm

Ray Casting Algorithm dipilih sebagai implementasi PIP dalam penelitian ini karena bekerja atas dasar prinsip topologi geometri yang sederhana tetapi efektif [ORO98]. Penentuan posisi sebuah titik oleh algoritma ini ditempuh melalui sejumlah langkah yang dijalankan secara berurutan.

1. Menarik garis horizontal imajiner dari titik koordinat kendaraan ke arah kanan menuju tak terhingga.
2. Memeriksa setiap sisi polygon satu per satu untuk menentukan apakah garis horizontal tersebut memotong sisi tersebut.
3. Menghitung total jumlah sisi polygon yang dipotong oleh garis horizontal tersebut.
4. Menyimpulkan bahwa titik berada di dalam polygon jika jumlah perpotongan ganjil, dan berada di luar polygon jika jumlahnya genap termasuk nol.

Algoritma ini memiliki kompleksitas waktu $O(n)$ terhadap jumlah titik koordinat polygon, sehingga tetap efisien meskipun dijalankan secara berulang untuk polygon dengan ratusan titik koordinat seperti yang umum ditemui pada data GeoJSON batas wilayah administratif. Efisiensi ini memastikan bahwa proses deteksi geofencing dapat berjalan dalam waktu kurang dari satu milidetik pada server modern.

Dalam penelitian ini, algoritma Point in Polygon diimplementasikan oleh Aldi pada backend Laravel dan dijalankan setiap kali data koordinat baru diterima dari perangkat GPS tracker yang dikembangkan oleh Fakih. Hasil dari algoritma ini menjadi dasar penentuan apakah sistem perlu memicu pengiriman notifikasi kepada pemilik kendaraan atau tidak.

2.1.13 Reverse Geocoding

Reverse geocoding merupakan proses pengubahan koordinat geografis, yakni latitude dan longitude, menjadi alamat lokasi yang bisa dibaca manusia, misalnya nama jalan dan kota [GOO23]. Kebalikan dari geocoding, yang mengubah alamat berbentuk teks menjadi koordinat, proses ini berperan penting dalam menyajikan informasi lokasi agar mudah dipahami pengguna.

Tanpa reverse geocoding, notifikasi yang dikirimkan kepada pemilik kendaraan hanya akan berisi angka koordinat yang tidak bermakna bagi kebanyakan orang. Pemilik kendaraan akan kesulitan memahami secara cepat di mana sebenarnya kendaraannya berada jika hanya diberikan koordinat numerik tanpa konteks alamat yang familiar.

2.1.13.1. Layanan Nominatim

Nominatim merupakan layanan geocoding dan reverse geocoding bersifat open-source yang dikembangkan OpenStreetMap Foundation dan beroperasi dengan memanfaatkan data peta dari OpenStreetMap [OSM23]. Layanan ini bisa diakses

tanpa biaya melalui HTTP API dengan cara mengirimkan parameter koordinat, lalu menerima respons berupa alamat lengkap dalam format JSON.

Dalam sistem yang dikembangkan, reverse geocoding Nominatim dimanfaatkan untuk dua skenario utama yang saling berkaitan dengan fitur notifikasi pada penelitian ini.

- Notifikasi Pelanggaran Zona: Mengubah koordinat posisi kendaraan menjadi alamat lengkap saat terdeteksi keluar dari zona geofencing yang telah ditetapkan.
- Notifikasi Kendaraan Berhenti: Mengubah koordinat posisi berhenti kendaraan menjadi alamat lengkap saat melebihi durasi yang ditentukan pengguna.

Dalam penelitian ini, layanan ini digunakan oleh Aldi pada backend Laravel untuk melengkapi notifikasi dengan informasi alamat yang mudah dipahami pemilik kendaraan. Pemilihan Nominatim juga konsisten dengan sumber data GeoJSON wilayah administratif yang juga berasal dari ekosistem OpenStreetMap, sehingga tidak diperlukan integrasi dengan layanan pihak ketiga tambahan yang berbeda sumber data.

2.1.14 Notifikasi Email dan WhatsApp

Notifikasi adalah mekanisme yang menyampaikan informasi kepada pengguna secara otomatis, yang dipicu oleh kejadian atau kondisi tertentu di dalam sistem, dan tidak menuntut adanya tindakan aktif pengguna untuk meminta informasi tersebut [PRS20]. Pada konteks sistem keamanan kendaraan, notifikasi memegang peran sebagai mekanisme peringatan dini yang amat kritis bagi pemilik kendaraan.

Penelitian ini mengimplementasikan sistem notifikasi melalui dua saluran komunikasi secara bersamaan, yaitu email dan WhatsApp, untuk memaksimalkan kemungkinan pengguna menerima peringatan secara tepat waktu. Pendekatan ganda ini dipilih karena masing-masing saluran memiliki kelebihan dan potensi kelemahan yang berbeda, sehingga penggunaan keduanya secara simultan dapat saling menutupi kekurangan satu sama lain.

2.1.14.1. SMTP (Simple Mail Transfer Protocol)

SMTP merupakan protokol standar internet untuk mengirim email dari pengirim kepada penerima melalui jaringan internet, sebagaimana yang didefinisikan dalam RFC 2821 [KLE01]. Protokol ini beroperasi dengan model store-and-forward,

yaitu email diteruskan melewati sejumlah server mail transfer agent sampai tiba di tujuan akhirnya.

Dalam penelitian ini, layanan SMTP Gmail digunakan sebagai media pengiriman notifikasi email secara otomatis melalui konfigurasi Laravel Mail pada file environment aplikasi. Konten email yang dikirimkan mencakup informasi kendaraan, koordinat posisi, alamat lengkap hasil reverse geocoding, dan waktu kejadian secara lengkap.

2.1.14.2. Fonnte API

Fonnte merupakan layanan API WhatsApp pihak ketiga yang berpusat di Indonesia dan menyediakan antarmuka HTTP untuk mengirim pesan WhatsApp secara programatik, tanpa perlu mengakses WhatsApp Business API resmi yang persyaratannya lebih rumit [FON23]. Layanan ini menghubungkan nomor WhatsApp milik pengguna yang berperan sebagai perangkat pengirim, dan perangkat tersebut dapat dipanggil melalui API.

Dalam sistem yang dikembangkan, kedua kanal notifikasi ini dikirimkan secara bersamaan dalam dua kondisi utama yang telah ditetapkan sebagai bagian dari kebutuhan fungsional sistem.

- Kendaraan terdeteksi keluar dari zona geofencing yang telah ditetapkan pengguna.
- Kendaraan berhenti melebihi durasi waktu yang telah ditentukan oleh pengguna sebelumnya.

Dalam penelitian ini, kedua notifikasi tersebut dikirimkan oleh Aldi secara bersamaan untuk memaksimalkan kemungkinan pengguna menerima peringatan secara tepat waktu meskipun salah satu kanal mengalami gangguan teknis. Pendekatan ini sejalan dengan kebutuhan non-fungsional reliability yang telah ditetapkan dalam analisis kebutuhan sistem.

2.1.15 Real-Time Monitoring

Istilah Real-time monitoring merujuk pada proses pemantauan kondisi ataupun status sebuah objek secara langsung, di mana rentang waktu antara kejadian dan penyajian informasinya kepada pengguna sangat kecil [SOM16]. Sifat tersebut termasuk salah satu tujuan pokok penelitian ini, sebab keterlambatan informasi berpotensi berakibat fatal bagi upaya pencegahan serta penanganan kasus pencurian kendaraan bermotor.

Implementasi real-time monitoring yang efektif memerlukan integrasi yang erat antara berbagai teknologi yang telah dibahas pada subbab-subbab sebelumnya. Setiap

teknologi memiliki peran spesifik dalam mata rantai komunikasi data yang dimulai dari perangkat fisik hingga sampai ke tampilan akhir yang dilihat oleh pengguna.

2.1.15.1. Alur Komunikasi Data Pada Sistem

Implementasi real-time monitoring dalam penelitian ini mengikuti alur yang melibatkan integrasi seluruh teknologi yang telah dibahas sebelumnya secara berurutan dan saling terhubung.

1. NEO-6M membaca koordinat lokasi kendaraan secara berkala dari sinyal satelit yang diterima.
2. ESP32 memproses data NMEA menggunakan library TinyGPS++ untuk mengekstrak koordinat yang valid.
3. SIM800L mengirim data ke MQTT Broker melalui jaringan GSM menggunakan protokol MQTT.
4. Laravel menerima data sebagai Subscriber MQTT dan menyimpannya ke database PostgreSQL.
5. Laravel menjalankan algoritma Point in Polygon untuk memeriksa status geofencing kendaraan.
6. Laravel mem-broadcast data terbaru melalui Pusher ke seluruh client yang terhubung.
7. Leaflet.js pada browser memperbarui posisi marker kendaraan secara otomatis tanpa refresh halaman.

Dalam penelitian ini, keseluruhan alur tersebut berjalan dalam hitungan detik, memungkinkan pemilik kendaraan memantau posisi kendaraannya secara langsung dan akurat. Kecepatan alur komunikasi data ini menjadi salah satu indikator keberhasilan implementasi yang akan diuji lebih lanjut pada kebutuhan non-fungsional performance di BAB III.

2.1.16 Black Box Testing

Black box testing merupakan metode pengujian perangkat lunak yang menitikberatkan pengujian fungsionalitas sistem tanpa meninjau struktur kode program di dalamnya, yakni dengan memberi data masukan tertentu lalu membandingkan keluaran yang dihasilkan sistem terhadap hasil yang diharapkan [UMI22]. Metode ini banyak dipakai sebab dapat dijalankan oleh penguji yang tidak menguasai detail implementasi kode program, sehingga fokus pengujian dapat diarahkan pada kesesuaian perilaku sistem dengan kebutuhan fungsional yang sudah ditetapkan [UMI22].

Dalam penelitian ini, black box testing digunakan untuk menguji seluruh use case pada sistem AF GPS Tracker, dengan setiap skenario pengujian dirancang dalam bentuk kasus uji (test case) yang mencakup data masukan, langkah pengujian, hasil yang diharapkan, dan status keberhasilan pengujian.

2.1.17 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian perangkat lunak yang dikerjakan oleh pengguna akhir guna memvalidasi apakah sistem yang dibangun sudah memenuhi kebutuhan serta layak diterima untuk dipakai pada kondisi nyata [ALI24]. Pengujian ini biasanya dilakukan sesudah pengujian fungsional semisal black box testing rampung, dengan maksud memperoleh penilaian langsung dari kacamata pengguna atas kemudahan penggunaan dan kesesuaian fitur sistem [ALI24].

Dalam penelitian ini, UAT dilakukan menggunakan Skala Guttman dengan dua pilihan jawaban, yaitu "Ya" dan "Tidak", yang diberikan kepada dua kelompok responden, yaitu Pengguna/Pemilik Kendaraan dan Admin, guna menilai kelayakan sistem AF GPS Tracker secara langsung.

2.2 Penelitian Terdahulu

Penelitian terdahulu berfungsi sebagai landasan dan pembandingan untuk memposisikan kontribusi penelitian ini secara ilmiah. Berikut adalah beberapa penelitian yang relevan dengan topik yang diangkat dalam tugas akhir ini:

Tabel 2.1. Penelitian Terdahulu

No	Judul Peneliti	Peneliti	Hasil	Metode Yang Digunakan
1	Implementasi Internet of Things pada Sistem Informasi Pelacakan Kendaraan Bermotor Menggunakan GPS Berbasis Web	I Wayan Agus Arimbawa, Abdul Chalel Rahman, dan Andy Hidayat Jatmika [RAH19]	Sistem pelacakan kendaraan berbasis IoT dirancang dengan memanfaatkan GPS, Arduino Uno R3, SIM900A, dan framework Laravel sehingga posisi kendaraan dapat ditampilkan secara real-time melalui website.	Waterfall
2	Rancang Bangun Sistem Monitoring dan Kontrol Jarak Jauh pada Rental Motor Berbasis IoT dengan Geofencing	Achmad Doli Harahap, Indarto, dan Faris Humam [HAR25]	Sistem monitoring rental motor dibangun menggunakan Laravel dan GPS NEO-6M untuk memantau lokasi kendaraan, mengendalikannya dari jarak jauh, serta menerapkan geofencing berbasis polygon.	Prototyping

No	Judul Peneliti	Peneliti	Hasil	Metode Yang Digunakan
3	Desain Sistem Keamanan Sepeda Motor Dengan Memanfaatkan GPS Tracker Berbasis IoT	M. Yosi Ashadi, Sofia Ariyani, Bagus Setya Rintyarna, dan Nanda Kurnia Wardati [ASH22]	Alat keamanan sepeda motor berbasis IoT memanfaatkan GPS Tracker dan mikrokontroler ESP8266, dengan penampilan posisi kendaraan melalui aplikasi Blynk.	Experimental Research (Metode Eksperimental)
4	Sistem Monitoring Kendaraan Logistik Secara Real-Time Berbasis Internet of Things (IoT)	Ghifari Munawar [MUN24]	Aplikasi web monitoring kendaraan logistik dibangun dengan GPS, GPRS, ASP.NET MVC, SQL Server, dan Google Maps API untuk menyajikan posisi kendaraan secara real-time.	Research and Development (R&D)
5	Implementasi Metode Geofencing Untuk Sistem Pelacak Lokasi: Studi Literatur Sistematis	Dina Fadiyah Putri dan Dandun Widhiantoro [PUT25]	Kajian literatur terhadap penerapan geofencing pada sejumlah sistem pelacak lokasi berbasis IoT menyimpulkan bahwa metode tersebut efektif untuk pemantauan lokasi, dengan dukungan teknologi GPS, LoRa, GSM, dan Google Maps API.	Systematic Literature Review (SLR)

BAB 3

SKEMA PENELITIAN

Bab ini menjelaskan tentang analisis yang dilakukan dalam pengembangan Sistem GPS IoT dan Website dengan Fitur *Geofencing* Dinamis. Pembahasan meliputi alur penyelesaian tugas akhir, perumusan masalah, kerangka pemikiran teoritis, objek penelitian, serta analisis kebutuhan pengguna, kebutuhan fungsional, kebutuhan non-fungsional, kebutuhan data, kebutuhan pengembangan perangkat lunak, dan kebutuhan *deployment*.

3.1 Alur Penyelesaian Tugas Akhir

Alur penyelesaian tugas akhir menjadi landasan untuk menjelaskan tahapan-tahapan yang dilakukan dalam menyelesaikan tugas akhir dan skema analisis terkait studi kasus yang dikerjakan. Berikut ini merupakan tahapan-tahapan pada alur penelitian yang menjadi landasan dalam penyelesaian tugas akhir.

Tabel 3.1. Alur Penyelesaian Tugas Akhir

Tahapan dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 1: Identifikasi Masalah Tahap awal penelitian dilakukan dengan mengidentifikasi permasalahan yang berkaitan dengan keamanan kendaraan bermotor, keterbatasan GPS tracker konvensional, dan kebutuhan implementasi geofencing berbasis wilayah administratif. Hasil: Diperoleh permasalahan utama berupa belum tersedianya sistem keamanan kendaraan bermotor yang mampu memberikan peringatan otomatis berdasarkan batas wilayah administratif dengan perangkat yang portabel dan non-invasive.	1. Mengidentifikasi fenomena pencurian dan perpindahan kendaraan bermotor. 2. Menganalisis kebutuhan pemantauan kendaraan secara <i>real-time</i>. 3. Mengidentifikasi keterbatasan sistem pelacakan kendaraan yang telah ada.	1. [DET26] 2. [SOM16]
Tahap 2: Pengumpulan Data Tahap ini dilakukan untuk memperoleh informasi dan landasan teori yang diperlukan melalui studi literatur. Hasil: Diperoleh kebutuhan pengguna, kebutuhan sistem, kebutuhan perangkat keras, dan teori-teori yang mendukung pengembangan sistem.	1. Studi literatur terkait mikrokontroler ESP32, modul GPS NEO-6M, modul SIM800L, dan sistem manajemen daya berbasis baterai lithium. 2. Studi literatur terkait protokol MQTT untuk pengiriman data IoT. 3. Studi literatur terkait Laravel, WebSocket, dan Leaflet.js. 4. Studi literatur terkait GeoJSON, Geofencing, dan algoritma Point In Polygon. 5. Studi literatur terkait reverse geocoding menggunakan Nominatim OpenStreetMap.	1. [SOM16] 2. [PRS20] 3. [KAP96] 4. [ESP21] 5. [UBL10] 6. [SIM13] 7. [OAS14] 8. [OTW23] 9. [RFC16] 10. [ORO98]
Tahap 3: Communication Tahap komunikasi dilakukan untuk memperoleh kebutuhan sistem secara menyeluruh baik dari sisi perangkat keras maupun perangkat lunak. Hasil: Diperoleh kebutuhan pengguna,	1. Analisis kebutuhan pengguna (User Requirement). 2. Analisis kebutuhan perangkat keras (Hardware Requirement). 3. Analisis kebutuhan perangkat lunak (Software Requirement).	1. [PRS20]

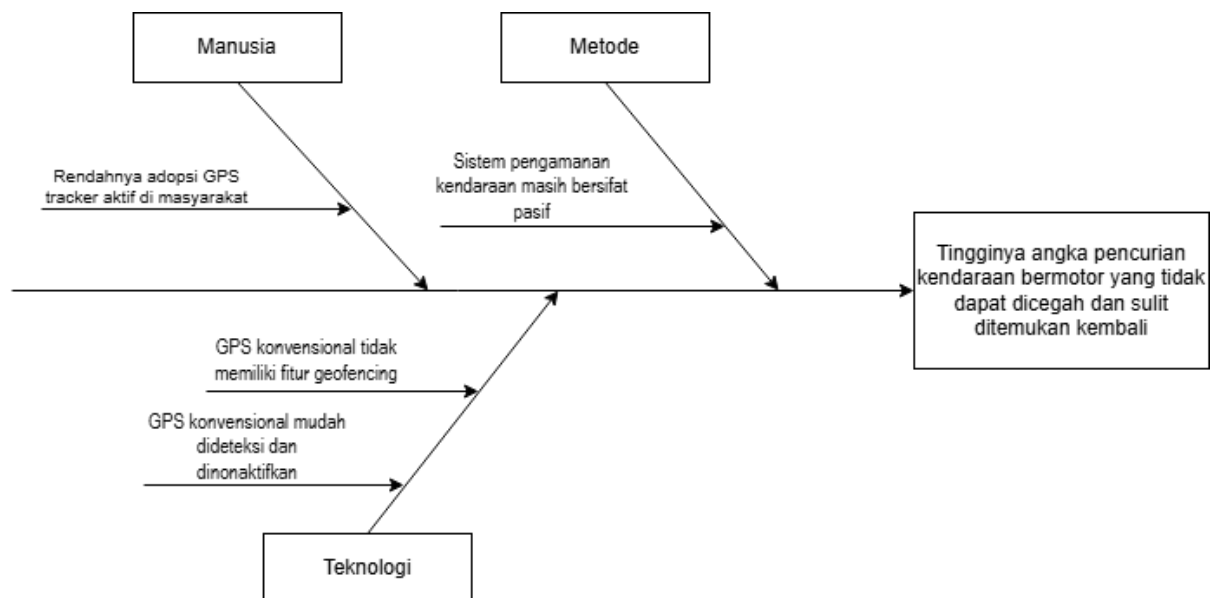
Tahapan dan Hasil	Langkah Penelitian	Literatur dan Referensi
kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional, kebutuhan non-fungsional, dan kebutuhan data sistem.	- Analisis kebutuhan fungsional (Functional Requirement). - Analisis kebutuhan non-fungsional. - Analisis kebutuhan data (Data Requirement).	
Tahap 4: Planning Tahap perencanaan dilakukan untuk menentukan ruang lingkup dan perencanaan pengembangan sistem yang akan dibangun. Hasil: Diperoleh perencanaan pengembangan perangkat keras dan perangkat lunak beserta arsitektur sistem yang akan dibangun.	- Menentukan ruang lingkup dan batasan sistem. - Menentukan komponen elektronik yang digunakan untuk perangkat keras. - Menentukan teknologi dan framework yang digunakan untuk website. - Menentukan arsitektur sistem secara keseluruhan.	[PRS20] [SOM16]
Tahap 5: Modeling Tahap pemodelan dilakukan untuk menggambarkan rancangan sistem yang akan dikembangkan secara detail. Hasil: Diperoleh skema rangkaian elektronik, model sistem website, rancangan basis data, dan rancangan antarmuka pengguna.	- Perancangan skema rangkaian elektronik perangkat keras menggunakan Fritzing. - Perancangan alur komunikasi data dari perangkat ke server. - Perancangan Use Case Diagram, Activity Diagram, dan Sequence Diagram. - Perancangan Entity Relationship Diagram (ERD). - Perancangan antarmuka pengguna website (UI/UX).	[PRS20]
Tahap 6: Construction Tahap implementasi dilakukan berdasarkan hasil perancangan yang telah dibuat. Hasil: Perangkat GPS tracker portabel dan non-invasive serta website pemantauan lokasi dengan fitur geofencing berbasis wilayah administratif berhasil dikembangkan.	- Perakitan rangkaian elektronik perangkat keras. - Penulisan firmware ESP32 menggunakan Arduino IDE dengan bahasa C++. - Implementasi library TinyGPS++ dan PubSubClient pada firmware. - Implementasi backend website menggunakan Laravel. - Implementasi WebSocket dan Pusher untuk pemantauan real-time. - Implementasi GeoJSON dan algoritma Point In Polygon untuk geofencing. - Implementasi notifikasi email (SMTP) dan WhatsApp (Fonnte).	[ESP21] [OAS14] [OTW23] [FET11] [RFC16] [ORO98]
Tahap 7: Deployment dan Pengujian Tahap ini dilakukan untuk memastikan seluruh sistem dapat berjalan dengan baik di lingkungan nyata. Hasil: Diperoleh hasil pengujian perangkat keras dan website serta evaluasi integrasi antar komponen sistem.	- Pengujian akurasi koordinat GPS dalam kondisi nyata. - Pengujian kestabilan koneksi pengiriman data melalui GSM. - Pengujian daya tahan baterai lithium. - Pengujian fungsional website menggunakan Black Box Testing. - Pengujian fitur geofencing wilayah administratif. - Pengujian notifikasi email dan WhatsApp. - Pengujian integrasi perangkat dengan website pemantauan.	[PRS20]
Tahap 8: Kesimpulan dan Saran Tahap akhir penelitian dilakukan dengan menyusun kesimpulan berdasarkan hasil implementasi dan pengujian sistem yang telah dilakukan. Hasil: Diperoleh kesimpulan sebagai jawaban dari setiap tujuan penelitian dan saran pengembangan sistem di masa mendatang.	- Analisis hasil penelitian dan pengujian sistem. - Penyusunan kesimpulan sebagai jawaban dari tujuan penelitian di BAB I. - Penyusunan saran pengembangan sistem lebih lanjut.	[SOM16] [PRS20]

3.2 Perumusan Masalah

Perumusan masalah bertujuan untuk mengidentifikasi masalah yang terdapat pada penelitian tugas akhir ini. Masalah dirumuskan menggunakan analisis sebab-akibat yang direpresentasikan dalam bentuk diagram fishbone.

3.2.1 Analisis Sebab Akibat

Analisis sebab akibat bertujuan untuk mengidentifikasi faktor-faktor yang menyebabkan tingginya angka pencurian kendaraan bermotor yang tidak dapat dicegah dan sulit ditemukan kembali. Diagram fishbone pada analisis ini menggunakan tiga kategori utama yaitu Manusia, Metode, dan Teknologi, yang masing-masing mewakili aspek penyebab yang berbeda namun saling berkontribusi terhadap akibat yang sama.



Gambar 3.1. Fishbone Sebab Akibat

Tabel 3.2. Deskripsi Fishbone Sebab Akibat

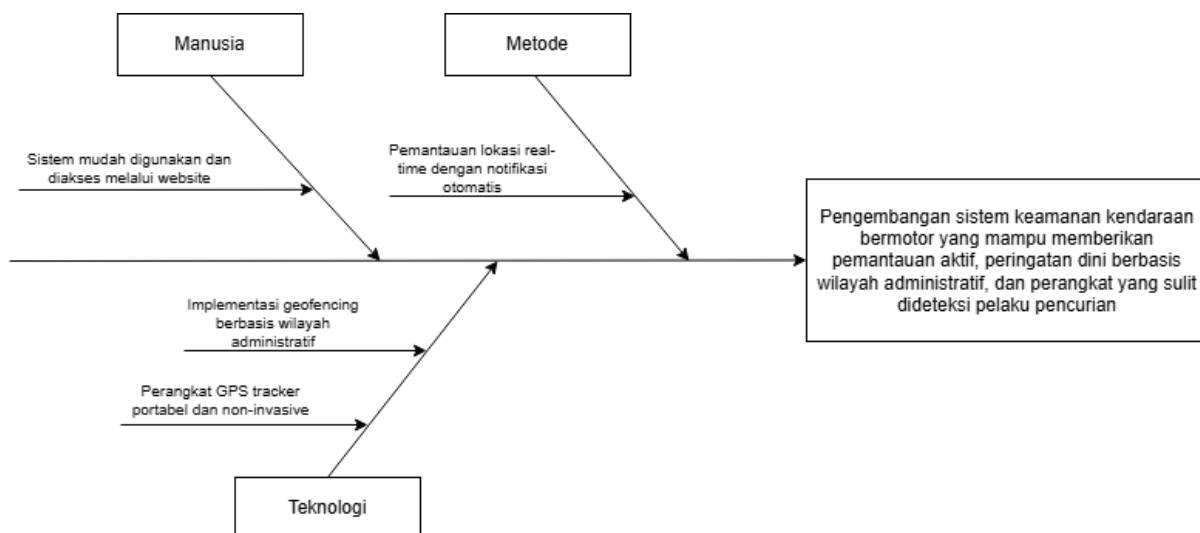
Aspek	Penyebab	Deskripsi Masalah
Manusia	Rendahnya adopsi GPS tracker aktif di masyarakat	Sebagian besar pemilik kendaraan bermotor belum menggunakan perangkat pelacak aktif, terbukti dari 1.494 kendaraan yang berhasil dicuri namun hanya satu unit yang terpantau melalui GPS aktif sehingga proses penemuan kendaraan menjadi sangat sulit.
Metode	Sistem pengamanan kendaraan masih bersifat pasif	Mekanisme pengamanan kendaraan bermotor saat ini masih bergantung pada laporan masyarakat sehingga proses penemuan kendaraan yang dicuri membutuhkan waktu lama dan tidak dapat dilakukan secara mandiri oleh pemilik kendaraan.
Teknologi	GPS konvensional tidak memiliki fitur geofencing	Perangkat GPS yang tersedia di pasaran saat ini belum dilengkapi fitur pembatasan wilayah administratif sehingga pemilik kendaraan tidak dapat menetapkan zona aman dan menerima peringatan otomatis ketika kendaraan melampaui batas wilayah yang ditentukan.

Aspek	Penyebab	Deskripsi Masalah
Teknologi	GPS konvensional mudah dideteksi dan dinonaktifkan	Perangkat GPS yang beredar saat ini umumnya dipasang secara permanen dengan memanfaatkan sistem kelistrikan kendaraan sehingga posisinya mudah dikenali dan rentan dinonaktifkan oleh pelaku pencurian sebelum atau sesaat setelah terjadi tindak kejahatan.

Berdasarkan tabel di atas, dapat disimpulkan bahwa akibat dari keempat penyebab tersebut adalah tingginya angka pencurian kendaraan bermotor yang tidak dapat dicegah dan sulit ditemukan kembali. Setiap penyebab yang teridentifikasi pada diagram fishbone ini bersumber langsung dari fakta dan data yang telah dipaparkan pada latar belakang dan identifikasi masalah di BAB I, sehingga keterkaitan antara kedua bagian tersebut dapat terjaga dengan baik.

3.2.2 Analisis Solusi Masalah

Analisis solusi masalah bertujuan untuk menggambarkan pendekatan penyelesaian yang ditawarkan oleh penelitian ini terhadap setiap penyebab yang telah teridentifikasi pada analisis sebab akibat sebelumnya. Setiap solusi yang diusulkan pada diagram fishbone ini memiliki keterkaitan langsung satu lawan satu dengan penyebab masalah yang bersangkutan, sehingga solusi yang dihasilkan benar-benar menjawab permasalahan yang ada bukan sekadar penambahan fitur yang tidak relevan.



Gambar 3.2. Fishbone Solusi Masalah

Tabel 3.3. Deskripsi Fishbone Solusi Masalah

Aspek	Penyebab	Deskripsi Masalah
Manusia	Sistem mudah digunakan dan diakses melalui website	Mengembangkan website pemantauan kendaraan dengan antarmuka yang sederhana dan mudah dipahami oleh masyarakat umum tanpa memerlukan keahlian teknis khusus,

Aspek	Penyebab	Deskripsi Masalah
		sehingga dapat mendorong adopsi penggunaan perangkat pelacak aktif yang lebih luas.
Metode	Pemantauan lokasi real-time dengan notifikasi otomatis	Mengembangkan sistem pemantauan lokasi kendaraan bermotor secara real-time yang mampu mendeteksi kondisi mencurigakan dan mengirimkan notifikasi otomatis melalui email dan WhatsApp kepada pemilik kendaraan tanpa bergantung pada laporan masyarakat.
Teknologi	Implementasi geofencing berbasis wilayah administratif	Mengimplementasikan fitur geofencing berbasis wilayah administratif menggunakan data GeoJSON dan algoritma Point in Polygon yang secara otomatis mengirimkan peringatan ketika kendaraan melampaui batas wilayah yang telah ditentukan.
Teknologi	Perangkat GPS tracker portabel dan non-invasive	Merancang perangkat GPS tracker yang dapat dipasang tanpa modifikasi permanen pada kendaraan sehingga posisinya dapat dipindahkan dan disembunyikan dengan mudah agar tidak dikenali dan dinonaktifkan oleh pelaku pencurian.

Berdasarkan tabel di atas, dapat disimpulkan bahwa keempat solusi tersebut menghasilkan satu kesatuan sistem yang utuh, yaitu Pengembangan Perangkat GPS Tracker Portabel dan Non-Invasive dengan Implementasi Geofencing Berbasis Wilayah Administratif untuk Sistem Keamanan Kendaraan Bermotor. Kedua solusi pada aspek Manusia dan Metode menjadi fokus utama pengembangan website yang dikerjakan oleh Aldi, sedangkan kedua solusi pada aspek Teknologi terbagi menjadi dua bagian yaitu geofencing yang dikembangkan oleh Aldi dan perangkat keras yang dikembangkan oleh Fakhri.

3.3 Kerangka Pemikiran Teoritis

Kerangka pemikiran teoritis merupakan gambaran alur berpikir yang menghubungkan antara permasalahan yang diidentifikasi, teori-teori yang digunakan sebagai landasan, dan solusi yang dihasilkan dalam penelitian ini. Kerangka pemikiran teoritis pada penelitian ini disusun berdasarkan landasan teori yang telah dipaparkan pada BAB II dan mengacu pada permasalahan yang telah diidentifikasi pada BAB I.

Permasalahan utama yang melatarbelakangi penelitian ini adalah tingginya angka pencurian kendaraan bermotor yang disebabkan oleh dua faktor utama yang saling berkaitan. Faktor pertama adalah keterbatasan perangkat GPS tracker konvensional yang bersifat invasif, mudah dideteksi, dan mudah dinonaktifkan oleh pelaku pencurian. Faktor kedua adalah belum tersedianya sistem pemantauan yang mampu memberikan peringatan secara proaktif berdasarkan batas wilayah administratif yang intuitif bagi pengguna.

Untuk menjawab faktor pertama, penelitian ini memanfaatkan konsep Internet of Things (IoT) sebagai landasan utama, di mana komponen-komponen fisik berupa mikrokontroler, modul GPS, dan modul komunikasi dihubungkan dan bekerja secara sinergis untuk

mengumpulkan dan mengirimkan data secara otomatis tanpa intervensi pengguna. Konsep Global Positioning System (GPS) menjadi dasar pemilihan modul NEO-6M sebagai penerima sinyal satelit yang menghasilkan data koordinat lokasi kendaraan dengan akurasi 2 hingga 5 meter. Mikrokontroler ESP32 dipilih sebagai unit pemrosesan utama berdasarkan karakteristiknya yang memiliki konsumsi daya rendah dan dua antarmuka UART yang memungkinkan komunikasi simultan dengan modul GPS dan modul GSM. Modul SIM800L dipilih sebagai media komunikasi data karena kemampuannya mengirimkan data melalui jaringan GSM tanpa bergantung pada Wi-Fi, sedangkan sistem manajemen daya berbasis baterai lithium yang dikelola oleh modul TP4056 menjadi komponen kunci yang memungkinkan perangkat beroperasi secara mandiri tanpa bergantung pada sistem kelistrikan kendaraan. Kombinasi seluruh komponen tersebut menghasilkan perangkat GPS tracker yang bersifat portabel dan non-invasive.

Untuk menjawab faktor kedua, penelitian ini memanfaatkan konsep geofencing sebagai mekanisme peringatan dini berbasis wilayah, dengan pendekatan geofencing polygon yang dipilih dibanding geofencing radius karena lebih mencerminkan batas wilayah nyata. Data GeoJSON administratif yang bersumber dari OpenStreetMap menjadi dasar penetapan zona aman berdasarkan wilayah kota. Algoritma Point in Polygon digunakan sebagai mekanisme deteksi posisi kendaraan terhadap zona yang telah ditetapkan. Protokol MQTT dipilih sebagai jembatan komunikasi data antara perangkat keras dan server karena sifatnya yang ringan dan efisien, sedangkan framework Laravel, PostgreSQL, WebSocket dan Pusher, serta Leaflet.js menjadi fondasi pengembangan website yang menampilkan data secara real-time. Reverse geocoding melalui Nominatim digunakan untuk melengkapi informasi alamat pada notifikasi yang dikirimkan melalui email dan WhatsApp kepada pemilik kendaraan.

Kombinasi seluruh teori dan teknologi tersebut menghasilkan satu sistem yang utuh dan terintegrasi, yaitu Perangkat GPS Tracker Portabel dan Non-Invasive dengan Implementasi Geofencing Berbasis Wilayah Administratif, yang menjadi solusi atas permasalahan keamanan kendaraan bermotor yang telah diidentifikasi sebelumnya.

3.3.1 Gambaran Produk Tugas Akhir

Produk yang dihasilkan dari penelitian ini adalah sebuah sistem keamanan kendaraan bermotor yang terdiri dari dua bagian utama yang saling terintegrasi, yaitu perangkat GPS tracker portabel dan non-invasive yang dikembangkan oleh Fakhri, serta website pemantauan lokasi kendaraan bermotor dengan implementasi geofencing berbasis wilayah administratif yang dikembangkan oleh Aldi. Kedua bagian ini bekerja secara sinergis membentuk satu sistem keamanan kendaraan bermotor yang utuh dan dapat diandalkan.

Secara fisik, perangkat GPS tracker terdiri dari beberapa komponen utama yang terintegrasi dalam satu rangkaian elektronik yang kompak, yaitu ESP32 sebagai mikrokontroler utama, modul GPS NEO-6M beserta antena GPS eksternal sebagai penerima sinyal satelit, modul SIM800L sebagai media komunikasi data melalui jaringan GSM, baterai lithium sebagai sumber daya mandiri, modul TP4056 sebagai pengisi daya, dan modul HT7333 sebagai penstabil tegangan. Perangkat ini bekerja dengan cara membaca data koordinat lokasi kendaraan secara berkala, memprosesnya menggunakan library TinyGPS++, kemudian mengirimkannya ke MQTT Broker menggunakan protokol MQTT melalui library PubSubClient. Perangkat dirancang agar dapat dipasang secara tersembunyi di berbagai posisi pada kendaraan tanpa modifikasi permanen, sehingga sulit dikenali dan dinonaktifkan oleh pelaku pencurian.

Secara fungsional, website pemantauan menerima data koordinat yang dikirimkan oleh perangkat GPS tracker melalui MQTT Broker, menyimpannya ke database PostgreSQL, dan menampilkannya secara real-time pada peta interaktif berbasis Leaflet.js melalui WebSocket dan Pusher. Website ini dilengkapi dengan fitur geofencing berbasis wilayah administratif yang memungkinkan pengguna menetapkan zona aman berdasarkan batas wilayah kota menggunakan data GeoJSON. Setiap kali kendaraan terdeteksi melampaui batas zona yang telah ditetapkan melalui algoritma Point in Polygon, sistem secara otomatis melakukan reverse geocoding menggunakan Nominatim dan mengirimkan notifikasi melalui email dan WhatsApp kepada pemilik kendaraan. Website juga menyediakan fitur riwayat perjalanan kendaraan, manajemen data kendaraan, serta tampilan status keaktifan perangkat GPS secara real-time.

Dengan kombinasi kedua bagian tersebut, sistem yang dihasilkan mampu memberikan solusi keamanan kendaraan bermotor yang lebih proaktif dibanding sistem pelacakan konvensional, baik dari sisi perangkat keras yang portabel dan non-invasive maupun dari sisi perangkat lunak yang dilengkapi geofencing berbasis wilayah administratif dan notifikasi otomatis.

3.4 Objek Penelitian

Objek penelitian pada tugas akhir ini adalah sistem keamanan kendaraan bermotor yang terdiri dari perangkat GPS tracker portabel dan non-invasive serta website pemantauan lokasi dengan implementasi geofencing berbasis wilayah administratif. Secara spesifik, objek penelitian mencakup tiga aspek utama sebagai berikut.

1. Perangkat Keras (Hardware)

Pada sisi perangkat keras, objek penelitian mencakup seluruh komponen elektronik yang membentuk perangkat GPS tracker, meliputi mikrokontroler ESP32,

modul GPS NEO-6M, antena GPS eksternal, modul SIM800L, baterai lithium, modul TP4056, dan modul HT7333, beserta rangkaian elektronik yang menghubungkan seluruh komponen tersebut dalam satu kesatuan perangkat yang fungsional, portabel, dan non-invasive.

2. Firmware

Pada sisi firmware, objek penelitian meliputi kode program yang ditulis menggunakan Arduino IDE dengan bahasa pemrograman C++ yang berjalan pada mikrokontroler ESP32, mencakup logika pembacaan data GPS, pemrosesan data NMEA menggunakan library TinyGPS++, pengelolaan koneksi GSM melalui modul SIM800L, dan pengiriman data koordinat ke server menggunakan protokol MQTT melalui library PubSubClient.

3. Website Pemantauan

Sementara itu, pada sisi perangkat lunak, objek penelitian mencakup aplikasi website yang dikembangkan menggunakan framework Laravel, meliputi backend yang menangani penerimaan data MQTT dan penyimpanan ke database PostgreSQL, implementasi fitur geofencing berbasis wilayah administratif menggunakan data GeoJSON dan algoritma Point in Polygon, implementasi pemantauan real-time menggunakan WebSocket dan Pusher, implementasi notifikasi melalui email dan WhatsApp, serta antarmuka pengguna yang dibangun menggunakan Leaflet.js untuk visualisasi peta interaktif.

4. Performa Sistem Secara Keseluruhan

Objek penelitian juga mencakup aspek-aspek kinerja sistem yang menjadi fokus pengujian, meliputi akurasi koordinat GPS yang dihasilkan dengan target presisi 2 hingga 5 meter, kestabilan koneksi pengiriman data melalui jaringan GSM dan MQTT, ketepatan deteksi geofencing berbasis algoritma Point in Polygon, kecepatan notifikasi yang dikirimkan kepada pengguna, serta integrasi keseluruhan antara perangkat keras dan website pemantauan dalam satu sistem yang utuh.

3.5 Analisis Kebutuhan Pengguna

Tahapan analisis kebutuhan pengguna bertujuan untuk mengidentifikasi pihak-pihak yang akan berinteraksi dengan sistem serta harapan dan kebutuhan mereka terhadap sistem yang akan dibangun.

3.5.1 Analisis Aktor

Aktor merupakan entitas yang berinteraksi secara langsung dengan sistem yang ada. Berikut adalah daftar aktor yang terlibat dalam Sistem GPS IoT dan Website dengan Fitur *Geofencing* Dinamis

Tabel 3.4. Analisis Aktor

No	Aktor	Deskripsi
1	Server/Backend	Sistem eksternal yang menerima dan memproses data koordinat yang dikirimkan oleh perangkat <i>GPS Tracker</i> melalui protokol MQTT
2	Pemilik Kendaraan	Pengguna yang memasang perangkat <i>GPS Tracker</i> pada kendaraan bermotor miliknya dan memantau lokasi kendaraan melalui website yang dikembangkan oleh rekan satu tim
3	Administrator	Pengguna dengan hak akses penuh yang bertanggung jawab mengelola data pengguna dan perangkat GPS yang terdaftar dalam sistem melalui panel admin, termasuk memantau ringkasan sistem melalui dashboard admin.

3.5.2 User Requirement

User Requirement (UR) merupakan daftar kebutuhan, keinginan, dan harapan setiap aktor terhadap sistem yang akan dibangun. Berikut adalah tabel User Requirement Sistem GPS IoT dan Website dengan Fitur *Geofencing* Dinamis.

Tabel 3.5. User Requirement

Kode	User Requirement
UR-01	Pengguna dapat memantau lokasi, tingkat kecepatan, arah gerak, serta sisa persentase baterai kendaraan secara real-time melalui website.
UR-02	Pengguna dapat menentukan zona geofencing berdasarkan wilayah administratif (provinsi/kabupaten/kecamatan), radius (titik pusat dan jarak), atau area bebas (polygon).
UR-03	Pengguna dapat melihat riwayat perjalanan kendaraan kapan saja melalui website
UR-04	Pengguna menerima notifikasi secara langsung pada dashboard website (in-app), serta melalui email dan WhatsApp ketika kendaraan keluar dari zona geofencing yang telah ditentukan.
UR-05	Pengguna mendapat pemberitahuan secara langsung pada dashboard website beserta alamat lengkap ketika kendaraan berhenti melebihi durasi yang ditentukan.
UR-06	Pengguna dapat mengatur sendiri durasi waktu berhenti sebelum notifikasi dikirimkan
UR-07	Pengguna dapat menambahkan dan mengelola data kendaraan yang dipantau
UR-08	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi berhenti sesuai kebutuhan
UR-09	Pengguna dapat melihat status perangkat GPS secara langsung di website, apakah aktif atau tidak
UR-10	Pengguna dapat menetapkan titik referensi (Home Location) untuk masing-masing kendaraan.

Kode	User Requirement
UR-11	Pengguna dapat mengunduh riwayat perjalanan dan riwayat berhenti kendaraan dalam bentuk dokumen PDF.
UR-12	Perangkat GPS dapat dipasang tanpa kabel dan mudah dipindahkan antar kendaraan
UR-13	Perangkat GPS mampu menghasilkan data koordinat lokasi secara akurat dengan presisi 2–5 meter
UR-14	Perangkat dapat ditempatkan secara tersembunyi di berbagai posisi pada kendaraan tanpa memerlukan modifikasi permanen
UR-15	Perangkat mampu beroperasi secara mandiri menggunakan baterai lithium tanpa bergantung pada sistem kelistrikan kendaraan
UR-16	Perangkat mampu mengirimkan data koordinat lokasi ke server secara berkala melalui jaringan GSM
UR-17	Admin dapat memantau ringkasan sistem (jumlah pengguna dan perangkat terdaftar, serta status online/offline perangkat) melalui dashboard admin.
UR-18	Admin dapat mengelola (menambahkan dan menghapus) data perangkat GPS yang terdaftar dalam sistem.
UR-19	Admin dapat mengelola (meninjau dan menghapus) data pengguna yang terdaftar dalam sistem.
UR-20	Admin dapat mengunduh data kredensial perangkat GPS yang baru ditambahkan dalam bentuk dokumen PDF.
UR-21	Pengguna dapat menetapkan batas kecepatan (overspeed) untuk masing-masing kendaraan.
UR-22	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi kecepatan berlebih (overspeed) sesuai kebutuhan.

3.6 Analisis *Hardware Requirement*

Analisis kebutuhan fungsional merupakan penjabaran dari User Requirement (UR) ke dalam *Hardware Requirement* (HR) yang harus dipenuhi oleh perangkat lunak. Setiap User Requirement (UR) dijabarkan menjadi satu atau bahkan lebih *Hardware Requirement* (HR) yang lebih spesifik dan terukur.

Tabel 3.6. Analisis *Hardware Requirement*

Kode UR	User Requirement	Kode HR	<i>Hardware Requirement</i>
UR-10	Perangkat GPS dapat dipasang tanpa kabel dan mudah dipindahkan antar kendaraan	HR-01	Baterai lithium sebagai sumber daya utama pengganti kabel kelistrikan kendaraan
		HR-02	Modul TP4056 untuk sistem pengisian daya baterai secara aman

Kode UR	User Requirement	Kode HR	Hardware Requirement
		HR-03	Modul HT7333 untuk menstabilkan tegangan yang masuk ke ESP32 dan komponen lainnya
UR-12	Perangkat GPS mampu menghasilkan data koordinat lokasi secara akurat dengan presisi 2–5 meter	HR-04	Modul GPS NEO-6M sebagai penerima sinyal satelit untuk menghasilkan data koordinat
		HR-05	Antena GPS eksternal untuk memperkuat penerimaan sinyal satelit agar akurasi mencapai 2–5 meter
UR-14	Perangkat dapat ditempatkan secara tersembunyi di berbagai posisi pada kendaraan tanpa memerlukan modifikasi permanen	HR-06	Perangkat dirancang dalam ukuran kompak dengan enclosure/casing pelindung sehingga dapat ditempatkan secara tersembunyi di berbagai posisi pada kendaraan tanpa modifikasi permanen
UR-15	Perangkat mampu mengirimkan data koordinat lokasi ke server secara berkala melalui jaringan GSM	HR-07	Modul SIM800L sebagai media pengiriman data koordinat ke server melalui jaringan GSM
UR-12, UR-16	Perangkat memproses data GPS dan mengirimkannya melalui jaringan GSM	HR-08	ESP32 sebagai mikrokontroler utama yang memproses data GPS dari modul NEO-6M dan mengelola komunikasi MQTT melalui modul SIM800L

Tabel 3.7. Analisis Perbandingan Pemilihan Modul Perangkat Keras

Komponen	Modul Yang diGunakan	Alternatif	Perbandingan	Alasan Memilih Modul yang diGunakan
Mikrokontroler	ESP32	ESP8266 / STM32	ESP32 memiliki kemampuan pemrosesan dan antarmuka komunikasi yang cukup untuk membaca GPS sekaligus mengendalikan modem GSM	ESP32 dipilih karena sesuai dengan kebutuhan pengolahan data GPS dan komunikasi GSM. Dua UART memungkinkan komunikasi serial dengan NEO-6M dan SIM800L secara lebih sesuai dengan rancangan perangkat.
GPS	NEO-6M	NEO-M8N	NEO-M8N mendukung multi-GNSS seperti GPS, GLONASS, Galileo, dan BeiDou serta memiliki sensitivitas dan kemampuan positioning lebih tinggi. NEO-6M hanya berfokus pada GPS, dengan akurasi sekitar 2,5 m pada kondisi spesifikasi u-blox.	NEO-6M sudah memenuhi kebutuhan penelitian, yaitu memperoleh koordinat kendaraan dengan target akurasi 2–5 m. Penggunaan NEO-M8N akan meningkatkan kemampuan GNSS, tetapi peningkatan tersebut bukan kebutuhan utama sistem karena data hanya digunakan sebagai input lokasi kendaraan.
Komunikasi seluler	SIM800L	SIM7670 / SIM7600	SIM7670/SIM7600 merupakan modul 4G LTE yang menyediakan kecepatan data jauh lebih tinggi dan fitur komunikasi lebih lengkap. SIM7672, misalnya, mendukung LTE Cat 1 hingga 10 Mbps downlink dan 5 Mbps uplink.	SIM800L dipilih karena kebutuhan komunikasi perangkat hanya berupa pengiriman data koordinat secara berkala melalui GPRS/MQTT. Sistem tidak membutuhkan bandwidth tinggi seperti video atau transfer data besar. Dengan demikian, kemampuan GPRS sudah mencukupi kebutuhan fungsional penelitian.

3.7 Analisis Software Requirement

Analisis kebutuhan fungsional merupakan penjabaran dari User Requirement (UR) ke dalam Software Requirement (SR) yang harus dipenuhi oleh perangkat lunak. Setiap User Requirement (UR) dijabarkan menjadi satu atau bahkan lebih Software Requirement (SR) yang lebih spesifik dan terukur.

Tabel 3.8. Analisis Software Requirement

Kode UR	User Requirement	Kode SR	Software Requirement
UR-01	Pengguna dapat memantau lokasi, tingkat kecepatan, arah gerak, serta sisa persentase baterai kendaraan secara real-time melalui website.	SR-01	Sistem membutuhkan WebSocket dan Pusher untuk mengirimkan data lokasi secara real-time ke browser.
		SR-02	Sistem membutuhkan Leaflet.js untuk menampilkan peta dan marker posisi kendaraan.
		SR-03	Sistem membutuhkan Laravel sebagai backend untuk mengelola data yang masuk dari perangkat GPS.
UR-02	Pengguna dapat menentukan zona geofencing berdasarkan wilayah administratif (provinsi/kabupaten/kecamatan), radius (titik pusat dan jarak), atau area bebas (polygon).	SR-04	Sistem membutuhkan data GeoJSON batas wilayah administratif untuk menggambarkan zona geofencing di peta.
		SR-05	Sistem membutuhkan algoritma Point in Polygon untuk mendeteksi apakah kendaraan berada di dalam atau luar zona geofencing.
UR-03	Pengguna dapat melihat riwayat perjalanan kendaraan kapan saja melalui website.	SR-06	Sistem membutuhkan PostgreSQL untuk menyimpan seluruh data koordinat perjalanan kendaraan secara permanen.
		SR-07	Sistem membutuhkan Leaflet.js untuk menggambar polyline riwayat perjalanan kendaraan di peta.
UR-04	Pengguna menerima notifikasi secara langsung pada dashboard website (in-app), serta melalui email dan WhatsApp ketika kendaraan keluar dari zona geofencing yang telah ditentukan.	SR-08	Sistem membutuhkan layanan SMTP untuk pengiriman notifikasi email.
		SR-09	Sistem membutuhkan API WhatsApp pihak ketiga untuk pengiriman notifikasi WhatsApp.
UR-05	Pengguna mendapat pemberitahuan secara langsung pada dashboard website beserta alamat lengkap ketika kendaraan berhenti melebihi durasi yang ditentukan.	SR-10	Sistem membutuhkan layanan Reverse Geocoding dari Nominatim OpenStreetMap untuk mengubah koordinat menjadi alamat lengkap.

Kode UR	User Requirement	Kode SR	Software Requirement
		SR-11	Sistem membutuhkan mekanisme untuk mendeteksi kondisi kendaraan berhenti berdasarkan durasi yang telah ditentukan.
UR-06	Pengguna dapat mengatur sendiri durasi waktu berhenti sebelum notifikasi dikirimkan.	SR-12	Sistem membutuhkan fitur konfigurasi pada database untuk menyimpan pengaturan durasi berhenti setiap pengguna.
UR-07	Pengguna dapat menambahkan dan mengelola data kendaraan yang dipantau.	SR-13	Sistem membutuhkan Eloquent ORM Laravel untuk operasi CRUD data kendaraan pada database PostgreSQL.
UR-08	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi berhenti sesuai kebutuhan.	SR-14	Sistem membutuhkan tabel konfigurasi di database untuk menyimpan status aktif atau nonaktif fitur notifikasi setiap pengguna.
UR-09	Pengguna dapat melihat status perangkat GPS secara langsung di website, apakah aktif atau tidak.	SR-15	Sistem membutuhkan mekanisme heartbeat untuk mendeteksi apakah perangkat GPS sedang aktif mengirim data atau tidak.
UR-10	Pengguna dapat menetapkan titik referensi (Home Location) untuk masing-masing kendaraan.	SR-16	Sistem membutuhkan fitur untuk menyimpan dan menampilkan koordinat Home Location pada masing-masing kendaraan.
UR-11	Pengguna dapat mengunduh riwayat perjalanan dan riwayat berhenti kendaraan dalam bentuk dokumen PDF.	SR-17	Sistem membutuhkan fitur untuk menghasilkan dokumen PDF berdasarkan data riwayat perjalanan dan riwayat berhenti kendaraan.
UR-17	Admin dapat memantau ringkasan sistem (jumlah pengguna dan perangkat terdaftar, serta status online/offline perangkat) melalui dashboard admin.	SR-18	Sistem membutuhkan dashboard admin yang menampilkan ringkasan jumlah pengguna, jumlah perangkat, serta status online/offline perangkat.
UR-18	Admin dapat mengelola (menambahkan dan menghapus) data perangkat GPS yang terdaftar dalam sistem.	SR-19	Sistem membutuhkan fitur bagi admin untuk menambahkan dan menghapus data perangkat GPS.
UR-19	Admin dapat mengelola (meninjau dan menghapus) data pengguna yang terdaftar dalam sistem.	SR-20	Sistem membutuhkan fitur bagi admin untuk menampilkan, mencari, dan menghapus data pengguna.
UR-20	Admin dapat mengunduh data kredensial perangkat GPS yang baru ditambahkan dalam bentuk dokumen PDF.	SR-21	Sistem membutuhkan fitur untuk menghasilkan dokumen PDF berisi kredensial (Device ID dan password) perangkat GPS yang baru ditambahkan.
UR-02	Pengguna dapat menentukan zona geofencing berdasarkan wilayah administratif (provinsi/kabupaten/kecamatan), radius (titik pusat dan jarak), atau area bebas (polygon).	SR-22	Sistem membutuhkan rumus Haversine untuk menghitung jarak kendaraan terhadap titik pusat pada zona geofencing bertipe radius.

Kode UR	User Requirement	Kode SR	Software Requirement
UR-21	Pengguna dapat menetapkan batas kecepatan (overspeed) untuk masing-masing kendaraan.	SR-23	Sistem membutuhkan mekanisme untuk membandingkan kecepatan kendaraan yang dikirimkan perangkat GPS dengan batas kecepatan yang dikonfigurasi per perangkat.
UR-22	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi kecepatan berlebih (overspeed) sesuai kebutuhan.	SR-24	Sistem membutuhkan tabel konfigurasi pada database untuk menyimpan nilai batas kecepatan dan status aktif/nonaktif notifikasi overspeed setiap perangkat.

3.8 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan tahap identifikasi seluruh fungsi dan perilaku sistem yang harus dipenuhi berdasarkan kebutuhan pengguna yang telah ditetapkan sebelumnya. Kebutuhan fungsional menggambarkan secara spesifik apa yang harus dapat dilakukan oleh sistem, bagaimana sistem merespons masukan dari pengguna maupun perangkat GPS, serta proses-proses internal yang berjalan di dalam sistem untuk menghasilkan keluaran yang diharapkan. Setiap kebutuhan fungsional diturunkan langsung dari kebutuhan perangkat lunak yang telah diidentifikasi pada subbab sebelumnya sehingga keterkaitan antar kebutuhan dapat terjaga dengan baik.

Berikut merupakan hasil analisis kebutuhan fungsional sistem yang diturunkan dari kebutuhan perangkat lunak:

Tabel 3.9. Analisis Kebutuhan Fungsional

Kode	Functional Requirement	Diturunkan dari
FR-01	Sistem menerima data koordinat dari perangkat GPS melalui protokol MQTT, memprosesnya melalui backend Laravel, dan meneruskannya ke browser secara real-time menggunakan WebSocket & Pusher	SR-01, SR-03
FR-02	Sistem menampilkan posisi kendaraan secara real-time pada peta interaktif berbasis Leaflet.js dengan marker yang diperbarui otomatis	SR-02
FR-03	Sistem menampilkan batas wilayah administratif (provinsi, kabupaten/kota, kecamatan, dan kelurahan) pada peta menggunakan data GeoJSON	SR-04
FR-04	Sistem memungkinkan pengguna memilih dan menyimpan zona geofencing berdasarkan wilayah administratif, radius (titik pusat dan jarak), atau area bebas (polygon) yang diinginkan	SR-04
FR-05	Sistem menjalankan algoritma Point in Polygon secara berkala untuk mendeteksi posisi kendaraan terhadap zona geofencing yang ditentukan	SR-05
FR-06	Sistem menyimpan seluruh data koordinat beserta timestamp perjalanan kendaraan ke database PostgreSQL secara otomatis	SR-06
FR-07	Sistem menampilkan riwayat perjalanan kendaraan dalam bentuk garis jalur (polyline) pada peta berdasarkan rentang waktu yang dipilih pengguna	SR-07
FR-08	Sistem mengirimkan notifikasi email secara otomatis kepada pengguna ketika kendaraan terdeteksi keluar dari zona geofencing	SR-08
FR-09	Sistem mengirimkan notifikasi WhatsApp secara otomatis kepada pengguna ketika kendaraan terdeteksi keluar dari zona geofencing	SR-08

Kode	Functional Requirement	Diturunkan dari
FR-10	Sistem mendeteksi kondisi kendaraan berhenti berdasarkan tidak adanya perubahan koordinat dalam durasi yang telah dikonfigurasi oleh pengguna	SR-10
FR-11	Sistem melakukan Reverse Geocoding untuk mengubah koordinat posisi berhenti menjadi alamat lengkap menggunakan Nominatim OpenStreetMap	SR-11
FR-12	Sistem mengirimkan notifikasi beserta alamat lengkap kepada pengguna ketika kendaraan berhenti melebihi durasi yang ditentukan	SR-11, FR-10
FR-13	Sistem menyimpan pengaturan durasi waktu berhenti yang ditentukan oleh masing-masing pengguna ke dalam database	SR-10
FR-14	Sistem menyediakan fitur untuk menambahkan data kendaraan baru beserta informasi identitas kendaraan	SR-13
FR-15	Sistem menyediakan fitur untuk mengubah dan menghapus data kendaraan yang telah didaftarkan	SR-13
FR-16	Sistem menyediakan tombol toggle untuk mengaktifkan atau menonaktifkan fitur notifikasi berhenti per pengguna, dan menyimpan status tersebut ke tabel konfigurasi database secara langsung	SR-14 (UR-08)
FR-17	Sistem menerima sinyal heartbeat dari perangkat GPS secara berkala sebagai indikator bahwa perangkat sedang aktif	SR-13
FR-18	Sistem menampilkan status perangkat GPS (aktif/tidak aktif) secara langsung di halaman website berdasarkan hasil deteksi heartbeat	SR-13
FR-19	Perangkat menggunakan baterai lithium sebagai sumber daya utama yang diisi melalui modul TP4056 dan distabilkan tegangannya oleh modul HT7333 sebelum dialirkan ke ESP32 dan komponen lainnya	HR-01, HR-02, HR-03
FR-20	Perangkat membaca data koordinat latitude dan longitude dari sinyal satelit melalui modul GPS NEO-6M yang diperkuat oleh antena GPS eksternal	HR-04, HR-05
FR-21	Perangkat memproses data NMEA yang dihasilkan modul GPS NEO-6M menggunakan library TinyGPS++ pada firmware ESP32 untuk mengekstrak nilai koordinat yang valid	HR-04, HR-08
FR-22	Perangkat mengelola koneksi internet melalui modul SIM800L menggunakan jaringan GSM dengan kartu SIM aktif yang terpasang	HR-07
FR-23	Perangkat mengirimkan data koordinat lokasi kendaraan ke MQTT Broker secara berkala menggunakan protokol MQTT melalui library PubSubClient pada firmware ESP32	HR-07, HR-08
FR-24	Perangkat mengirimkan sinyal heartbeat secara berkala ke server sebagai indikator bahwa perangkat sedang aktif dan terhubung	HR-08
FR-25	Perangkat dapat dipasang dan dilepas dari kendaraan tanpa memerlukan modifikasi permanen pada sistem kelistrikan kendaraan	HR-06
FR-26	Perangkat memiliki ukuran yang kompak sehingga dapat ditempatkan secara tersembunyi di berbagai posisi pada kendaraan yang sulit dijangkau	HR-06
FR-27	Sistem menampilkan dashboard admin berisi ringkasan jumlah pengguna, jumlah perangkat (terpakai/tersedia), dan status online/offline perangkat.	SR-18
FR-28	Admin dapat menambahkan perangkat GPS baru secara massal dengan Device ID dan password yang dibuat otomatis oleh sistem.	SR-19
FR-29	Admin dapat menghapus data perangkat GPS yang belum terhubung dengan akun pengguna.	SR-19
FR-30	Admin dapat meninjau dan menghapus data pengguna yang terdaftar dalam sistem, kecuali akun admin.	SR-20
FR-31	Sistem menghasilkan dokumen PDF berisi kredensial perangkat GPS yang baru ditambahkan oleh admin.	SR-21
FR-32	Sistem menjalankan perhitungan jarak menggunakan rumus Haversine secara berkala untuk mendeteksi apakah kendaraan berada di dalam atau di luar zona geofencing bertipe radius	SR-22
FR-33	Sistem membandingkan kecepatan kendaraan yang dikirimkan perangkat GPS dengan batas kecepatan yang dikonfigurasi, dan mendeteksi transisi dari kondisi normal ke kondisi melebihi batas (overspeed)	SR-23
FR-34	Sistem mengirimkan notifikasi kepada pengguna melalui channel yang dikonfigurasi ketika kendaraan terdeteksi melebihi batas kecepatan (overspeed)	SR-23, FR-33

Kode	Functional Requirement	Diturunkan dari
FR-35	Sistem menyediakan pengaturan bagi pengguna untuk menetapkan nilai batas kecepatan dan mengaktifkan atau menonaktifkan notifikasi overspeed untuk masing-masing kendaraan	SR-24

3.9 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan analisis kebutuhan yang tidak berhubungan langsung dengan fungsi utama dari sistem, namun sangat menentukan kualitas dan performa sistem secara keseluruhan.

Tabel 3.10. Analisis Kebutuhan Non-Fungsional

Kode	Atribut Kualitas	Keterangan
NF-01	Performance	Posisi kendaraan di peta harus diperbarui dalam waktu maksimal 3 detik setelah data diterima server. Dashboard utama harus dapat ditampilkan sepenuhnya dalam waktu maksimal 5 detik pada koneksi internet minimal 10 Mbps. Notifikasi email dan WhatsApp harus terkirim dalam waktu maksimal 10 detik setelah pelanggaran zona terdeteksi.
NF-02	Scalability	Sistem harus mampu menerima dan memproses data dari minimal 10 perangkat GPS secara bersamaan dan melayani minimal 10 pengguna aktif (koneksi WebSocket) secara bersamaan tanpa penurunan performa yang signifikan.
NF-03	Security	Seluruh komunikasi antara browser dan server menggunakan protokol HTTPS. Password pengguna disimpan dalam bentuk hash bcrypt dengan salt rounds minimal 10. Seluruh query database menggunakan parameterized query untuk mencegah SQL Injection. Setiap pengguna hanya dapat mengakses data kendaraan milik akunnya sendiri. Seluruh proses autentikasi pengguna (login, pembuatan akun setelah aktivasi perangkat, dan pemulihan password) memanfaatkan fitur autentikasi bawaan Laravel, dengan akses ke seluruh halaman pengelolaan dibatasi hanya untuk pengguna yang telah terautentikasi.
NF-04	Availability	Server aplikasi ditargetkan beroperasi dengan uptime minimal 95% selama periode pengujian.
NF-05	Reliability	Sistem melakukan retry pengiriman notifikasi email dan WhatsApp sebanyak 3 kali dengan jeda 30 detik jika pengiriman pertama gagal. Seluruh kegagalan pengiriman dicatat dalam log sistem.
NF-06	Compatibility	Sistem dapat diakses dan berjalan normal pada browser Google Chrome versi 90+, Mozilla Firefox versi 88+, dan Microsoft Edge versi 90+. Antarmuka bersifat responsif dan dapat diakses melalui perangkat desktop maupun smartphone.
NF-07	Usability	Seluruh antarmuka sistem, label, pesan error, dan notifikasi menggunakan Bahasa Indonesia. Setiap aksi permanen atau kritis (hapus kendaraan, hapus zona geofencing) menampilkan dialog konfirmasi sebelum dieksekusi untuk mencegah kesalahan yang tidak disengaja.
NF-08	Maintainability	Kode backend menggunakan arsitektur MVC (Model-View-Controller) agar mudah dipelihara dan dikembangkan. Seluruh konfigurasi sensitif (API key Fonnte, API key Pusher, kredensial database, kredensial SMTP) disimpan di file .env dan tidak di-hardcode dalam kode program.
NF-09	Interoperability	Sistem harus mampu menerima data koordinat dari perangkat GPS tracker yang dikembangkan oleh rekan satu tim melalui protokol MQTT, serta terintegrasi dengan layanan eksternal Nominatim OpenStreetMap untuk reverse geocoding dan API Fonnte untuk pengiriman notifikasi WhatsApp.

Kode	Atribut Kualitas	Keterangan
NF-10	Performance (Hardware)	Perangkat harus mampu mengirimkan data koordinat ke server dalam interval maksimal 5 detik sekali dalam kondisi sinyal GPS dan GSM yang stabil.
NF-11	Accuracy	Perangkat harus mampu menghasilkan data koordinat dengan akurasi posisi antara 2 hingga 5 meter dalam kondisi penerimaan sinyal satelit yang baik di area terbuka.
NF-12	Reliability (Hardware)	Perangkat harus mampu melakukan koneksi ulang secara otomatis ke MQTT Broker apabila koneksi GSM terputus, tanpa memerlukan restart perangkat secara manual.
NF-13	Portability	Perangkat harus dapat dipasang dan dilepas dari kendaraan tanpa menggunakan alat khusus dan tanpa meninggalkan bekas modifikasi pada kendaraan.
NF-14	Power Efficiency	Perangkat harus mampu beroperasi secara mandiri menggunakan baterai lithium dalam kondisi normal pengiriman data secara berkala.
NF-15	Durability	Perangkat harus mampu beroperasi secara stabil pada rentang suhu -10°C hingga 60°C, sesuai rentang operasional komponen ESP32 dan SIM800L.
NF-16	Compatibility (Hardware)	Perangkat harus mampu mengirimkan data koordinat dalam format yang dapat diterima dan diproses oleh backend website yang dikembangkan oleh rekan satu tim melalui protokol MQTT.

3.10 Analisis Kebutuhan Data

Analisis kebutuhan data bertujuan untuk mengidentifikasi data-data yang diperlukan oleh sistem dalam menjalankan fungsinya. Sistem ini menggunakan PostgreSQL yang menyimpan seluruh data dalam skema relasional secara terstruktur [RAM03].

Tabel 3.11. Analisis Kebutuhan Data

Kode	Data Requirement	Tipe Data	Keterangan
DR-01	Data koordinat lokasi kendaraan (latitude, longitude)	DECIMAL	Menyimpan posisi kendaraan yang diperoleh dari perangkat GPS dan digunakan untuk monitoring real-time, geofencing, dan riwayat perjalanan.
DR-02	Data timestamp koordinat	DATETIME	Menyimpan waktu saat koordinat lokasi dikirimkan oleh perangkat dan diterima oleh server.
DR-03	Data identitas perangkat GPS (device_id)	VARCHAR	Pengenal unik setiap perangkat GPS yang digunakan untuk menghubungkan perangkat dengan kendaraan yang dipantau.
DR-04	Data status sinyal GPS	INTEGER	Menyimpan jumlah satelit yang diterima perangkat sebagai indikator kualitas sinyal GPS.

Kode	Data Requirement	Tipe Data	Keterangan
DR-05	Data heartbeat perangkat GPS	DATETIME	Menyimpan waktu terakhir perangkat mengirimkan sinyal aktif ke server sebagai indikator status online/offline.
DR-06	Data GeoJSON batas wilayah administratif	JSON	Menyimpan data poligon wilayah administratif (provinsi, kabupaten/kota, kecamatan, dan kelurahan) yang digunakan dalam proses geofencing.
DR-07	Data zona geofencing kendaraan	JSON/VARCHAR	Menyimpan wilayah administratif yang dipilih pengguna sebagai zona geofencing untuk kendaraan tertentu.
DR-08	Data riwayat perjalanan kendaraan	DECIMAL, DATETIME	Menyimpan kumpulan data koordinat beserta timestamp yang digunakan untuk menampilkan jalur perjalanan kendaraan.
DR-09	Data posisi berhenti kendaraan	DECIMAL, DATETIME	Menyimpan koordinat dan waktu ketika kendaraan terdeteksi berhenti melebihi durasi tertentu.
DR-10	Data alamat hasil reverse geocoding	VARCHAR	Menyimpan alamat lengkap yang diperoleh dari konversi koordinat menggunakan layanan Nominatim OpenStreetMap.
DR-11	Data log notifikasi sistem	VARCHAR, DATETIME	Menyimpan riwayat notifikasi geofencing maupun notifikasi kendaraan berhenti beserta waktu pengirimannya.
DR-12	Data pengaturan durasi berhenti	INTEGER	Menyimpan konfigurasi durasi waktu berhenti yang ditentukan oleh pengguna sebelum notifikasi dikirimkan.
DR-13	Data akun pengguna	VARCHAR	Menyimpan informasi pengguna berupa nama, email, dan password yang telah dienkripsi (hashed).
DR-14	Data kendaraan yang didaftarkan	VARCHAR	Menyimpan informasi kendaraan berupa nama kendaraan, nomor polisi, merek kendaraan, dan device_id perangkat GPS yang terpasang.
DR-15	Data status notifikasi berhenti	BOOLEAN	Menyimpan status aktif atau nonaktif fitur notifikasi kendaraan berhenti pada masing-masing pengguna.
DR-16	Data format payload MQTT	JSON	Menyimpan struktur data yang dikirim perangkat ke MQTT Broker yang berisi device_id, latitude, longitude, timestamp, dan informasi pendukung lainnya.
DR-17	Data pengaturan batas kecepatan (overspeed) perangkat	INTEGER, BOOLEAN	Menyimpan nilai batas kecepatan (limit_kmh) dan status aktif/nonaktif notifikasi overspeed yang

Kode	Data Requirement	Type Data	Keterangan
			dikonfigurasi pengguna untuk masing-masing perangkat.
DR-18	Data status transisi kondisi overspeed perangkat	BOOLEAN	Menyimpan status kondisi kecepatan kendaraan sebelumnya (melebihi batas atau tidak) untuk mendeteksi transisi yang memicu notifikasi overspeed.

3.11 Analisis Kebutuhan Pengembangan Perangkat Lunak

Analisis kebutuhan pengembangan perangkat lunak merupakan tahap identifikasi seluruh tools, *framework*, dan lingkungan pengembangan yang dibutuhkan selama proses perancangan dan pembangunan sistem. Kebutuhan ini mencakup seluruh perangkat lunak yang digunakan oleh pengembang dalam menulis kode program, mengelola versi, menguji fungsionalitas, serta merancang skema database sistem. Berikut merupakan hasil analisis kebutuhan pengembangan perangkat lunak yang digunakan dalam penelitian ini:

Tabel 3.12. Analisis Kebutuhan Pengembangan Perangkat Lunak

Kode	Kebutuhan Pengembangan	Keterangan
SDR-01	Visual Studio Code	Code editor utama yang digunakan untuk pengembangan backend Laravel, frontend website, dan konfigurasi sistem
SDR-02	Arduino IDE	Lingkungan pengembangan yang digunakan untuk menulis, mengompilasi, dan mengunggah firmware ke mikrokontroler ESP32
SDR-03	Git & GitHub	Tools version control yang digunakan untuk mengelola perubahan source code selama proses pengembangan sistem
SDR-04	Composer	Package manager PHP yang digunakan untuk mengelola dependensi dan library pada framework Laravel
SDR-05	Laragon	Local development server yang digunakan sebagai lingkungan pengembangan PHP, Apache, dan PostgreSQL
SDR-06	Postman	Tools yang digunakan untuk melakukan pengujian API dan validasi komunikasi data pada backend
SDR-07	Google Chrome DevTools	Tools debugging yang digunakan untuk menguji antarmuka website, performa aplikasi, dan koneksi WebSocket
SDR-08	pgAdmin	Aplikasi yang digunakan untuk mengelola struktur database, tabel, relasi, serta memantau data pada PostgreSQL.
SDR-09	Laravel Framework	Framework PHP yang digunakan untuk membangun backend sistem monitoring, geofencing, notifikasi, dan manajemen data kendaraan

3.12 Analisis Kebutuhan *Deployment*

Analisis kebutuhan *deployment* merupakan tahap identifikasi seluruh komponen, layanan, dan konfigurasi yang diperlukan agar sistem dapat berjalan dengan baik di lingkungan produksi. Berbeda dengan kebutuhan pengembangan yang bersifat lokal dan hanya digunakan selama proses pembangunan sistem, kebutuhan *deployment* bersifat permanen dan menjadi prasyarat utama bagi sistem untuk dapat diakses oleh pengguna secara nyata. Berikut merupakan hasil analisis kebutuhan *deployment* yang digunakan dalam penelitian ini:

Tabel 3.13. Analisis Kebutuhan *Deployment*

Kode	Kebutuhan <i>Deployment</i>	Keterangan
SDPR-01	Web Server (Apache/Nginx)	Web server yang digunakan untuk menjalankan aplikasi Laravel dan menangani seluruh request HTTP dari pengguna
SDPR-02	PHP 8.1 atau lebih tinggi	Runtime PHP yang diperlukan untuk menjalankan framework Laravel pada lingkungan produksi
SDPR-03	PostgreSQL	Database server yang digunakan untuk menyimpan data pengguna, kendaraan, perangkat, lokasi, geofencing, riwayat perjalanan, riwayat berhenti, dan notifikasi.
SDPR-04	MQTT Broker (EMQX - broker.emqx.io)	MQTT Broker publik EMQX yang digunakan untuk menerima data koordinat dan heartbeat dari perangkat GPS Tracker
SDPR-05	Laravel Framework	Framework backend yang di-deploy pada server untuk mengelola logika bisnis sistem monitoring kendaraan
SDPR-06	Pusher WebSocket Service	Layanan WebSocket yang digunakan untuk mendistribusikan data lokasi kendaraan secara real-time ke browser pengguna
SDPR-07	SMTP Mail Service	Layanan SMTP yang digunakan untuk mengirimkan notifikasi email secara otomatis kepada pengguna
SDPR-08	API WhatsApp Fonnte	Layanan pihak ketiga yang digunakan untuk mengirimkan notifikasi WhatsApp secara otomatis kepada pengguna
SDPR-09	SSL/TLS Certificate (HTTPS)	Sertifikat keamanan yang digunakan untuk mengenkripsi komunikasi data antara browser, server, dan layanan eksternal
SDPR-10	Laravel Artisan Migration	Fasilitas deployment Laravel yang digunakan untuk membuat dan memperbarui struktur database sesuai rancangan sistem
SDPR-11	ESP32	Mikrokontroler utama yang telah diprogram dengan firmware final dan berfungsi sebagai pusat kendali perangkat GPS Tracker
SDPR-12	Modul GPS NEO-6M	Modul GPS yang digunakan untuk memperoleh koordinat lokasi kendaraan dari satelit GPS
SDPR-13	Antena GPS Eksternal	Antena aktif yang digunakan untuk meningkatkan kualitas penerimaan sinyal satelit GPS
SDPR-14	Modul SIM800L	Modul GSM yang digunakan untuk menghubungkan perangkat GPS Tracker ke jaringan seluler dan internet
SDPR-15	Kartu SIM GSM Aktif	Kartu SIM dengan paket data aktif yang digunakan oleh SIM800L untuk komunikasi data ke server
SDPR-16	Baterai Lithium	Sumber daya utama perangkat GPS Tracker yang memungkinkan perangkat bekerja secara portabel dan non-invasive
SDPR-17	Modul TP4056	Modul pengisian baterai yang digunakan untuk mengelola proses charging baterai lithium secara aman
SDPR-18	Modul HT7333	Modul low dropout regulator (LDO) yang digunakan untuk menstabilkan tegangan sebelum dialirkan ke ESP32 dan komponen lainnya

BAB 4

PERANCANGAN SISTEM

Bab ini menjelaskan perancangan sistem keamanan dan pemantauan kendaraan bermotor yang terdiri dari perangkat GPS Tracker portabel dan non-invasive serta website pemantauan lokasi kendaraan secara real-time. Perancangan yang dilakukan mencakup pemodelan berbasis skenario yang meliputi Business Use Case, Use Case, Skenario Use Case, dan Sequence Diagram, perancangan basis data menggunakan Entity Relationship Diagram (ERD), perancangan arsitektur sistem, perancangan blok diagram perangkat dan alur komunikasi sistem, serta perancangan antarmuka website.

4.1 Pemodelan Berbasis Skenario

Pemodelan berbasis skenario dilakukan untuk menggambarkan kebutuhan dan proses interaksi yang terdapat pada sistem GPS Tracker dan website pemantauan kendaraan. Pemodelan ini digunakan sebagai dasar dalam menentukan fungsi sistem, alur interaksi pengguna dengan sistem, serta proses yang terjadi dalam menjalankan setiap kebutuhan yang telah ditetapkan. Pemodelan berbasis skenario pada penelitian ini meliputi Business Use Case, Use Case, Skenario Use Case, dan Sequence Diagram.

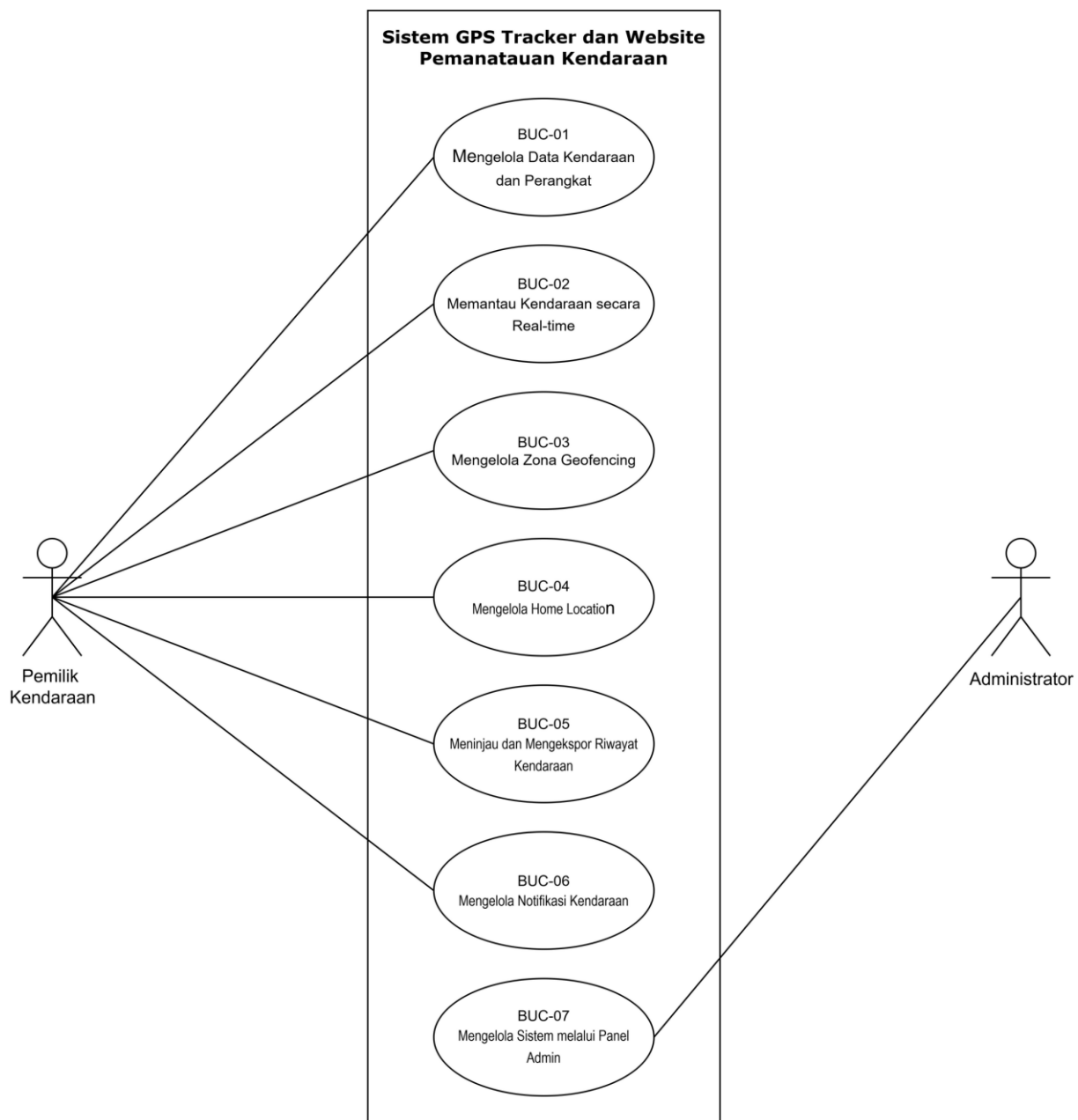
4.1.1 Business Use Case

Business Use Case digunakan untuk menggambarkan proses bisnis yang dilakukan oleh pengguna dalam menggunakan sistem GPS Tracker dan website pemantauan kendaraan. Pemodelan ini menunjukkan hubungan antara Pemilik Kendaraan sebagai aktor dengan proses-proses utama yang terdapat dalam sistem berdasarkan kebutuhan pengguna yang telah ditentukan pada tahap analisis. Business Use Case yang telah dirancang selanjutnya menjadi dasar dalam menentukan fungsi sistem dan penyusunan Use Case pada tahap perancangan.

Tabel 4.1. Business Use Case

Kode	Business Use Case	Aktor	Deskripsi
BUC-01	Mengelola Data Kendaraan dan Perangkat	Pemilik Kendaraan	Proses penambahan dan pengelolaan data kendaraan serta penambahan perangkat GPS tambahan yang digunakan untuk melakukan pemantauan kendaraan.
BUC-02	Memantau Kendaraan secara Real-time	Pemilik Kendaraan	Proses pemantauan posisi dan informasi kendaraan secara langsung melalui peta pada website serta melihat status perangkat GPS.
BUC-03	Mengelola Zona Geofencing	Pemilik Kendaraan	Proses menentukan, mengubah, menghapus, dan mengaktifkan atau menonaktifkan zona geofencing berdasarkan wilayah administratif, radius, atau area bebas berbentuk polygon, termasuk pengaturan notifikasi pelanggaran geofencing.
BUC-04	Mengelola Home Location	Pemilik Kendaraan	Proses menetapkan, mengubah, memindahkan, dan menghapus titik Home Location sebagai lokasi referensi untuk masing-masing kendaraan.
BUC-05	Meninjau dan Mengeksplor Riwat Kendaraan	Pemilik Kendaraan	Proses melihat riwayat perjalanan dan riwayat berhenti kendaraan berdasarkan periode tertentu, melakukan playback perjalanan, serta mengunduh riwayat dalam bentuk dokumen PDF.

Kode	Business Use Case	Aktor	Deskripsi
BUC-06	Mengelola Notifikasi Kendaraan	Pemilik Kendaraan	Proses mengatur durasi berhenti dan batas kecepatan (overspeed), mengaktifkan atau menonaktifkan notifikasi berhenti dan overspeed, serta menerima dan meninjau notifikasi terkait kondisi kendaraan, kecepatan berlebih, dan pelanggaran geofencing.
BUC-07	Mengelola Sistem melalui Panel Admin	Administrator	Proses admin memantau ringkasan sistem melalui dashboard, mengelola data perangkat GPS (menambahkan, menghapus, dan mengekspor kredensial), serta mengelola data pengguna yang terdaftar dalam sistem.



Gambar 4.1. Diagram Business Use Case

4.1.2 Use Case

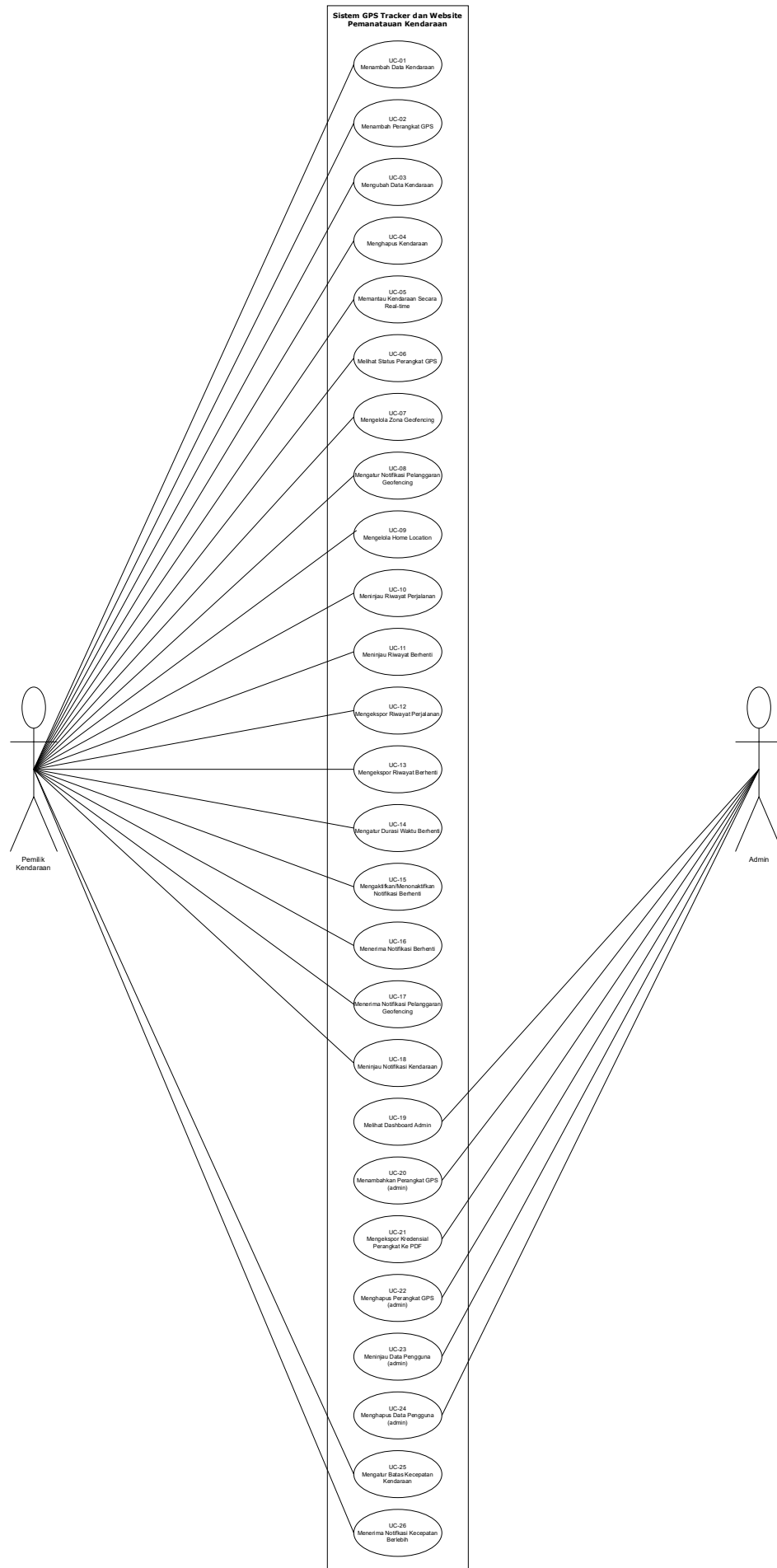
Use Case digunakan untuk menggambarkan fungsi-fungsi yang disediakan oleh sistem berdasarkan kebutuhan pengguna yang telah dianalisis. Pemodelan ini

menjelaskan hubungan antara Pemilik Kendaraan sebagai aktor dengan fungsi sistem yang digunakan untuk mengelola akun, mengaktivasi perangkat GPS, mengelola kendaraan dan geofencing, melakukan pemantauan kendaraan, mengatur notifikasi, serta melihat riwayat perjalanan. Use Case yang telah dirancang menjadi dasar untuk menjelaskan alur interaksi antara pengguna dengan sistem melalui skenario Use Case.

Tabel 4.2. Use Case

Kode	Use Case	Aktor	Deskripsi
UC-01	Menambahkan Data Kendaraan	Pemilik Kendaraan	Proses pengguna menambahkan data kendaraan yang akan dipantau dan menghubungkannya dengan perangkat GPS.
UC-02	Menambahkan Perangkat GPS	Pemilik Kendaraan	Proses pengguna menambahkan perangkat GPS baru ke akun yang telah dimiliki dengan memasukkan Device ID dan password perangkat.
UC-03	Mengubah Data Kendaraan	Pemilik Kendaraan	Proses pengguna memperbarui informasi kendaraan yang telah terdaftar pada sistem.
UC-04	Menghapus Kendaraan	Pemilik Kendaraan	Proses pengguna menghapus atau melakukan unbind kendaraan beserta keterkaitan perangkat dari akun.
UC-05	Memantau Kendaraan secara Real-time	Pemilik Kendaraan	Proses pengguna memantau posisi dan informasi kendaraan secara langsung melalui peta pada website.
UC-06	Melihat Status Perangkat GPS	Pemilik Kendaraan	Proses pengguna melihat status perangkat GPS berdasarkan kondisi perangkat dalam mengirimkan data ke sistem.
UC-07	Mengelola Zona Geofencing	Pemilik Kendaraan	Proses pengguna menambahkan, melihat, mengubah, menghapus, serta mengaktifkan atau menonaktifkan zona geofencing berdasarkan wilayah administratif, radius, atau polygon.
UC-08	Mengatur Notifikasi Pelanggaran Geofencing	Pemilik Kendaraan	Proses pengguna mengatur channel notifikasi pelanggaran geofencing yang tersedia untuk kendaraan dan menerima pemberitahuan ketika kendaraan keluar dari zona.
UC-09	Mengelola Home Location	Pemilik Kendaraan	Proses pengguna menetapkan, mengubah, memindahkan, dan menghapus Home Location sebagai titik referensi kendaraan.
UC-10	Meninjau Riwayat Perjalanan	Pemilik Kendaraan	Proses pengguna melihat riwayat perjalanan kendaraan berdasarkan periode tertentu dan melakukan playback perjalanan.
UC-11	Meninjau Riwayat Berhenti	Pemilik Kendaraan	Proses pengguna melihat daftar riwayat berhenti kendaraan berdasarkan data perjalanan yang tersimpan.
UC-12	Mengekspor Riwayat Perjalanan	Pemilik Kendaraan	Proses pengguna mengunduh riwayat perjalanan kendaraan dalam bentuk dokumen PDF.
UC-13	Mengekspor Riwayat Berhenti	Pemilik Kendaraan	Proses pengguna mengunduh riwayat berhenti kendaraan dalam bentuk dokumen PDF.
UC-14	Mengatur Durasi Waktu Berhenti	Pemilik Kendaraan	Proses pengguna menentukan durasi berhenti kendaraan yang menjadi dasar sistem dalam memberikan notifikasi.
UC-15	Mengaktifkan/Menonaktifkan Notifikasi Berhenti	Pemilik Kendaraan	Proses pengguna mengatur status aktif atau nonaktif fitur notifikasi berhenti.
UC-16	Menerima Notifikasi Berhenti	Pemilik Kendaraan	Proses sistem mendeteksi kendaraan berhenti melebihi durasi yang ditentukan dan mengirimkan pemberitahuan beserta alamat kendaraan kepada pengguna.
UC-17	Menerima Notifikasi Pelanggaran Geofencing	Pemilik Kendaraan	Proses sistem mendeteksi kendaraan keluar dari zona geofencing dan mengirimkan notifikasi melalui channel yang telah ditentukan.
UC-18	Meninjau Notifikasi Kendaraan	Pemilik Kendaraan	Proses pengguna melihat kumpulan notifikasi dan peringatan dari kendaraan yang dimiliki melalui panel notifikasi pada website.
UC-19	Melihat Dashboard Admin	Administrator	Proses admin melihat ringkasan sistem berupa jumlah pengguna, jumlah perangkat, dan status online/offline perangkat melalui dashboard admin.

Kode	Use Case	Aktor	Deskripsi
UC-20	Menambahkan Perangkat GPS (Admin)	Administrator	Proses admin menambahkan perangkat GPS baru secara massal dengan Device ID dan password yang dibuat otomatis oleh sistem.
UC-21	Mengekspor Kredensial Perangkat ke PDF	Administrator	Proses admin mengunduh daftar Device ID dan password perangkat GPS yang baru ditambahkan dalam bentuk dokumen PDF.
UC-22	Menghapus Perangkat GPS (Admin)	Administrator	Proses admin menghapus data perangkat GPS yang belum terhubung dengan akun pengguna.
UC-23	Meninjau Data Pengguna (Admin)	Administrator	Proses admin melihat dan mencari data pengguna yang terdaftar dalam sistem.
UC-24	Menghapus Data Pengguna (Admin)	Administrator	Proses admin menghapus data pengguna yang terdaftar dalam sistem, kecuali akun admin.
UC-25	Mengatur Batas Kecepatan Kendaraan	Pemilik Kendaraan	Proses pengguna menentukan nilai batas kecepatan (overspeed) kendaraan yang menjadi acuan sistem dalam memberikan notifikasi, serta mengaktifkan atau menonaktifkan notifikasi tersebut.
UC-26	Menerima Notifikasi Kecepatan Berlebih	Pemilik Kendaraan	Proses sistem mendeteksi kendaraan melebihi batas kecepatan yang ditentukan dan mengirimkan notifikasi kepada pengguna melalui channel yang telah dikonfigurasi.



Gambar 4.2. Diagram Use Case

4.1.3 Diagram Sequence

Diagram Sequence menggambarkan urutan interaksi antara pengguna dengan sistem dalam menjalankan setiap fungsi yang terdapat pada aplikasi AF GPS Tracker. Diagram ini menunjukkan aliran proses dari tindakan yang dilakukan oleh pengguna, proses yang dijalankan oleh sistem, hingga keluaran yang dihasilkan. Berikut merupakan Diagram Sequence untuk setiap Use Case yang telah didefinisikan.

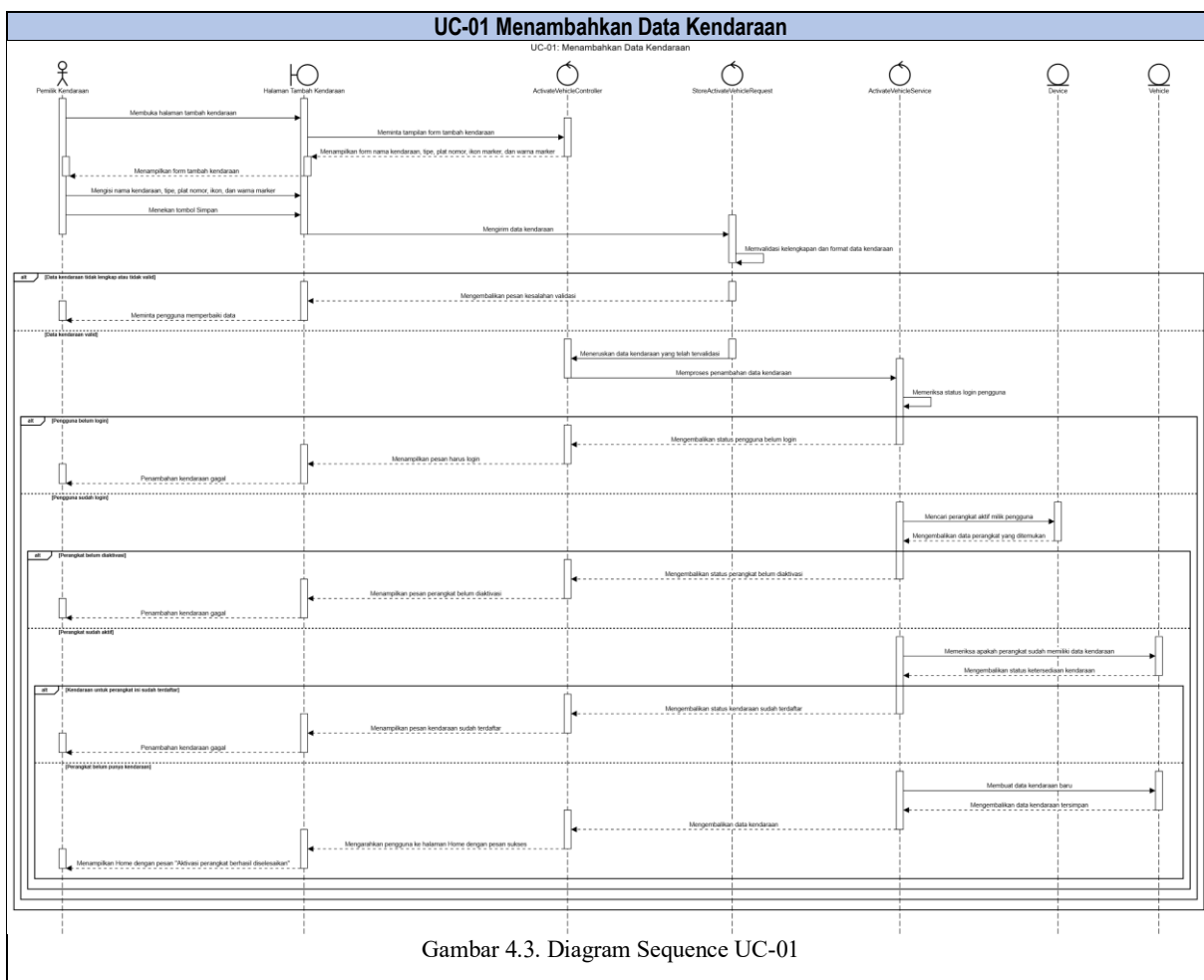
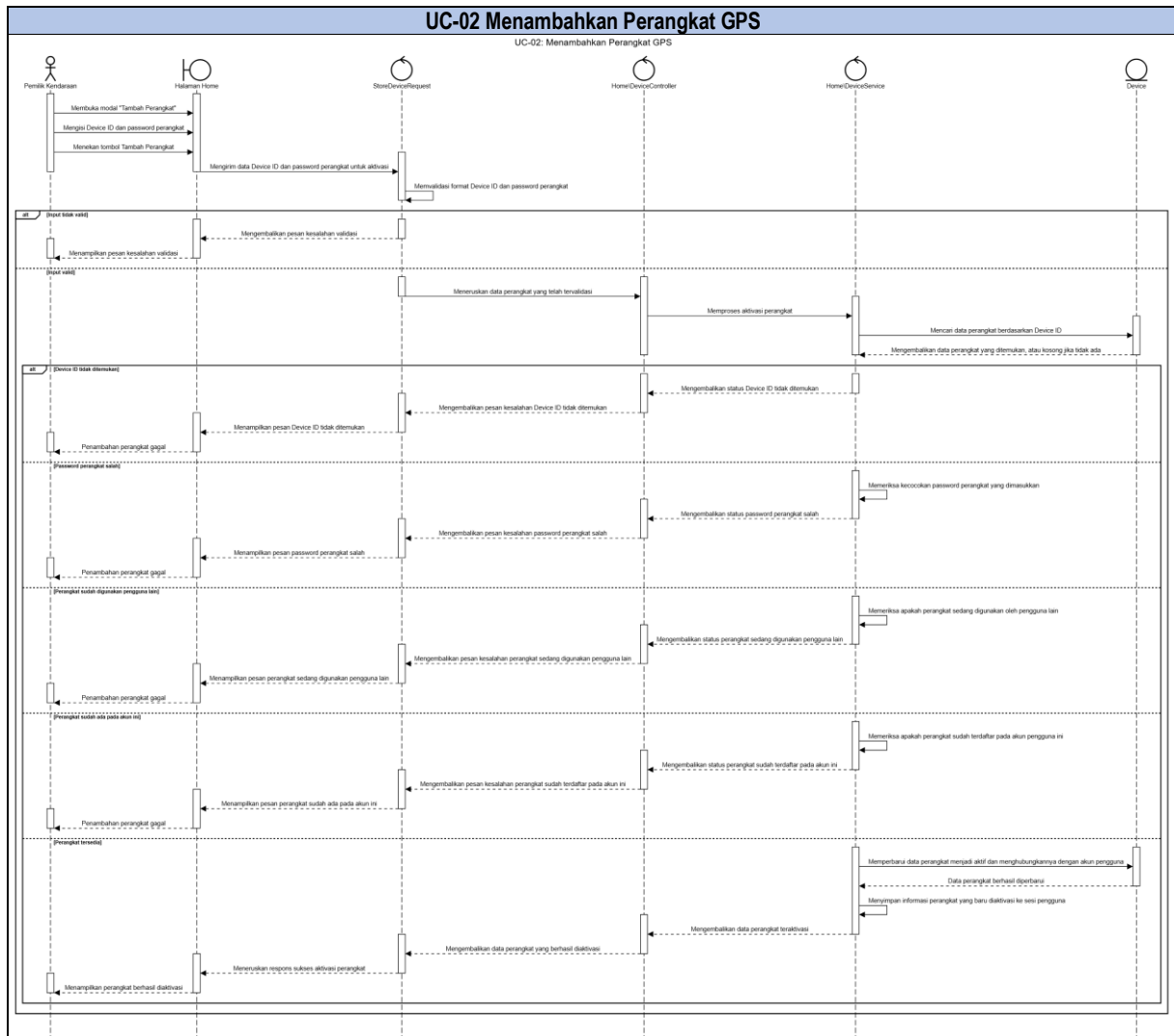
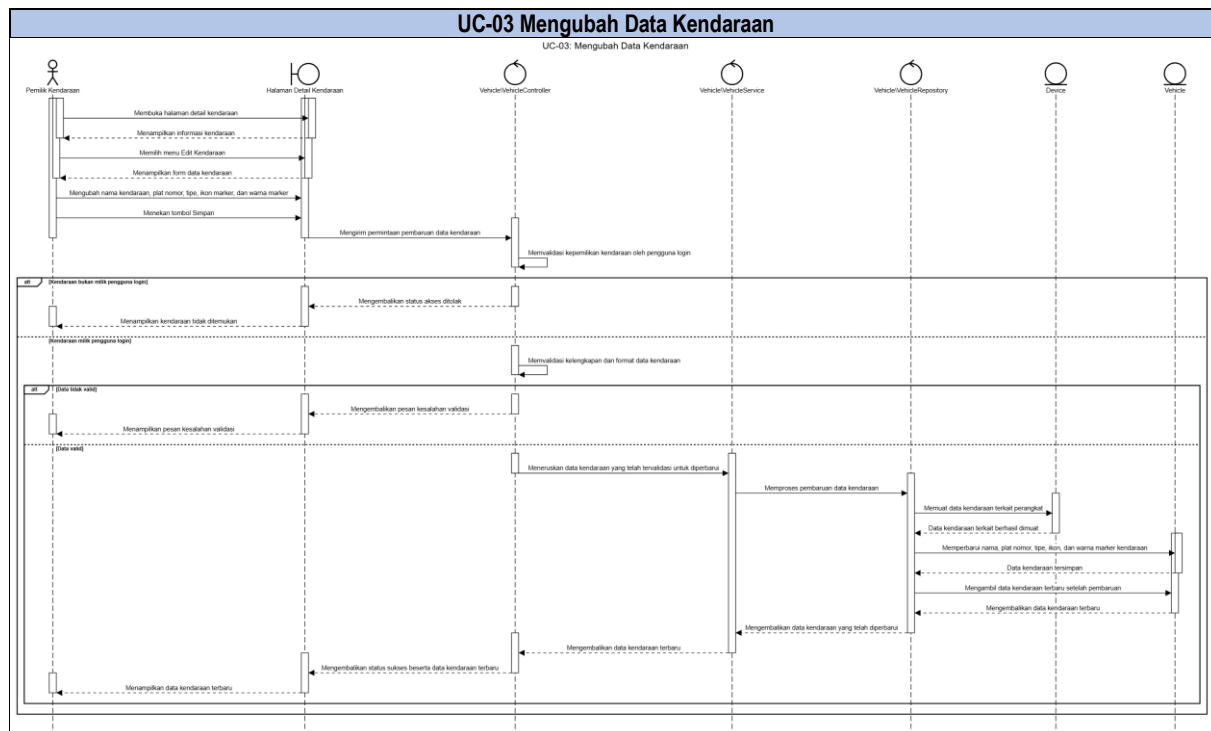


Diagram Sequence UC-01 menggambarkan alur interaksi ketika Pemilik Kendaraan menambahkan data kendaraan setelah perangkat GPS berhasil diaktivasi. Proses diawali ketika pengguna membuka halaman Tambah Kendaraan yang ditangani oleh ActivateVehicleController, kemudian mengisi nama kendaraan, tipe, nomor polisi, serta ikon dan warna marker yang divalidasi terlebih dahulu. Setelah data dinyatakan valid, ActivateVehicleService memeriksa keberadaan perangkat aktif milik pengguna serta memastikan perangkat tersebut belum memiliki data kendaraan lain. Apabila kedua syarat terpenuhi, sistem membuat data kendaraan baru pada model Vehicle di dalam sebuah transaksi database dan mengaitkannya dengan perangkat yang telah diaktivasi. Setelah proses penyimpanan berhasil, sistem menampilkan pesan bahwa aktivasi perangkat telah selesai dan mengarahkan pengguna ke halaman Home.



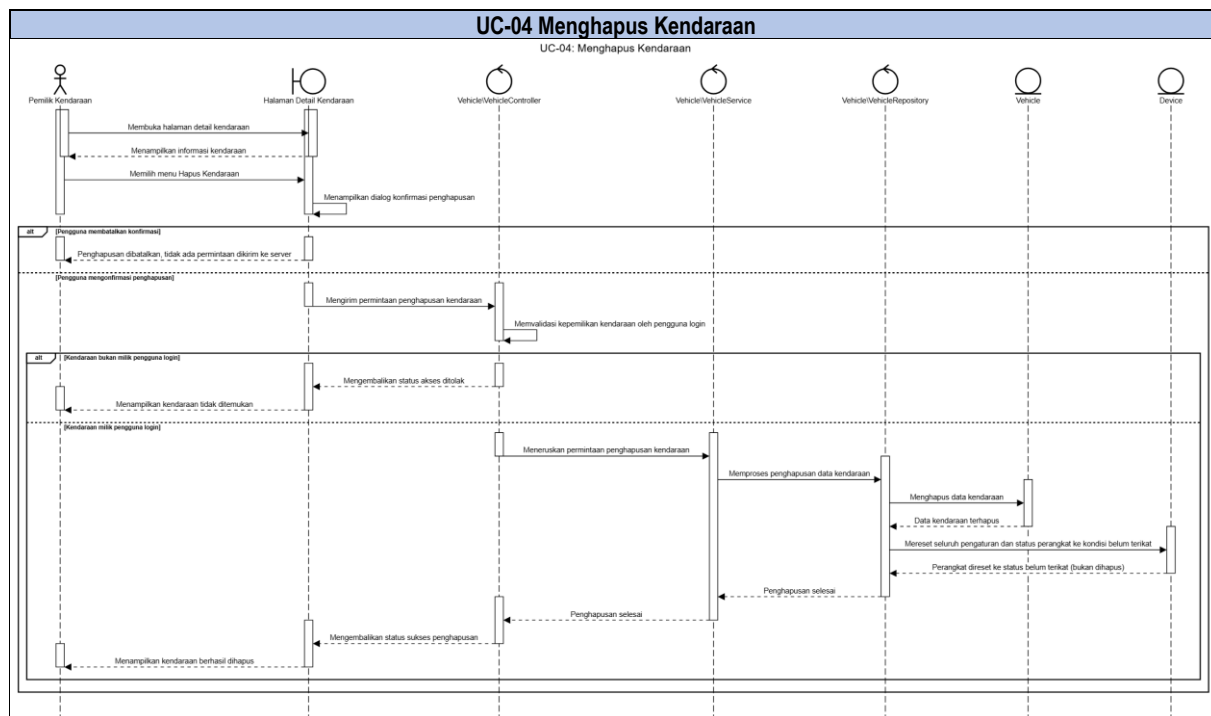
Gambar 4.4. Diagram Sequence UC-02

Diagram Sequence UC-02 menggambarkan alur interaksi ketika Pemilik Kendaraan menambahkan perangkat GPS tambahan melalui halaman Home. Proses diawali ketika pengguna membuka fitur Tambah Perangkat dan memasukkan Device ID beserta password perangkat, yang terlebih dahulu diperiksa formatnya sebelum dikirimkan kepada HomeDeviceController. Selanjutnya, HomeDeviceService memeriksa data perangkat langsung pada model Device di dalam sebuah transaksi database, meliputi keberadaan Device ID, kecocokan password, serta apakah perangkat telah terikat dengan akun lain maupun akun pengguna itu sendiri. Apabila seluruh pemeriksaan tersebut dinyatakan valid, sistem mengikat perangkat dengan akun pengguna yang sedang aktif. Sistem kemudian mengembalikan respons keberhasilan, dan pada sisi antarmuka pengguna, modal penambahan perangkat ditutup lalu digantikan dengan modal pengisian data kendaraan pada halaman yang sama tanpa memuat ulang halaman.



Gambar 4.5. Diagram Sequence UC-03

Diagram Sequence UC-03 menggambarkan alur interaksi ketika Pemilik Kendaraan mengubah informasi kendaraan yang telah terdaftar. Proses diawali ketika pengguna membuka halaman Detail Kendaraan, di mana perangkat pada alamat halaman secara otomatis diikl oleh Laravel melalui mekanisme route-model binding tanpa memerlukan pencarian data secara manual. Setelah kepemilikan perangkat diverifikasi oleh Vehicle\VehicleController, pengguna memilih menu Edit Kendaraan dan mengubah data yang divalidasi langsung pada method tersebut sebelum diteruskan kepada Vehicle\VehicleService. Apabila validasi gagal, sistem secara otomatis mengembalikan pesan kesalahan tanpa memproses perubahan lebih lanjut. Apabila valid, Vehicle\VehicleRepository memperbarui data kendaraan yang terhubung dengan perangkat tersebut, kemudian sistem menampilkan informasi bahwa data kendaraan berhasil diperbarui.



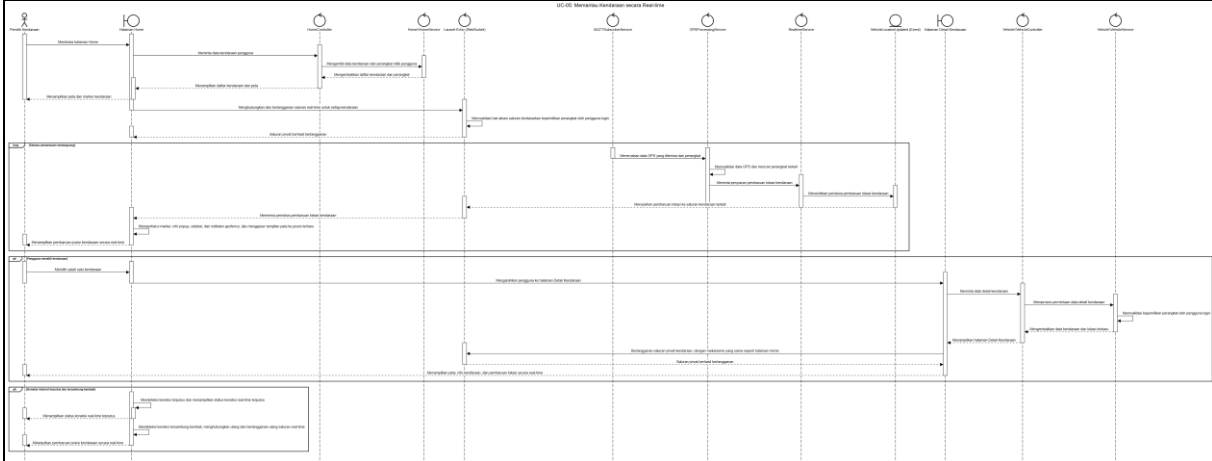
Gambar 4.6. Diagram Sequence UC-04

Diagram Sequence UC-04 menggambarkan alur interaksi ketika Pemilik Kendaraan menghapus kendaraan dari akunnya. Proses diawali ketika pengguna memilih menu Hapus Kendaraan pada halaman Detail Kendaraan, kemudian sistem menampilkan dialog konfirmasi pada sisi antarmuka. Apabila pengguna membatalkan konfirmasi, permintaan penghapusan tidak pernah dikirimkan ke server. Apabila

UC-04 Menghapus Kendaraan

pengguna mengonfirmasi, permintaan dikirimkan kepada Vehicle\VehicleController yang terlebih dahulu memverifikasi kepemilikan perangkat, kemudian diteruskan kepada Vehicle\VehicleService dan Vehicle\VehicleRepository untuk memproses penghapusan. Sistem menghapus data kendaraan beserta seluruh data terkait seperti geofence, notifikasi, riwayat perjalanan, riwayat berhenti, dan log perangkat, kemudian mengembalikan perangkat ke kondisi belum terikat pengguna tanpa menghapus perangkat itu sendiri, sehingga perangkat dapat diaktivasi kembali di kemudian hari. Sistem selanjutnya menampilkan notifikasi keberhasilan dan mengarahkan pengguna kembali ke halaman Home.

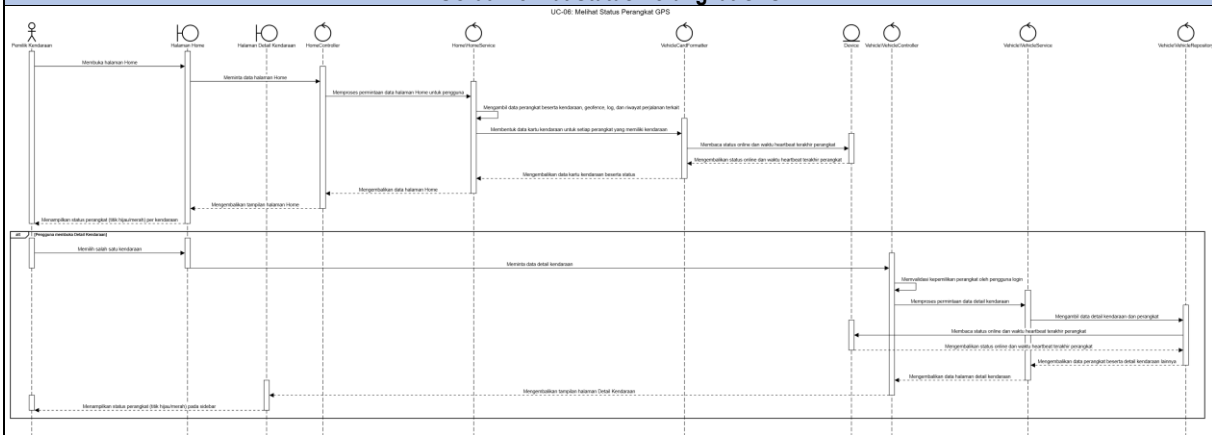
UC-05 Memantau Kendaraan Secara Real-time



Gambar 4.7. Diagram Sequence UC-05

Diagram Sequence UC-05 menggambarkan alur interaksi ketika Pemilik Kendaraan memantau kendaraan secara real-time. Proses diawali ketika pengguna membuka halaman Home, di mana HomeController melalui HomeService menampilkan daftar kendaraan beserta posisi terakhirnya pada peta. Untuk memperoleh pembaruan lokasi secara langsung, antarmuka pengguna membuka koneksi WebSocket melalui Laravel Echo pada kanal privat milik masing-masing perangkat. Secara terpisah dan berjalan tanpa bergantung pada aktivitas pengguna, setiap kali perangkat GPS mengirimkan data melalui MQTT, GPSProcessingService memproses data tersebut kemudian RealtimeService menyiarkan pembaruan lokasi melalui event yang diterima oleh seluruh browser yang sedang terhubung pada kanal tersebut, sehingga posisi kendaraan pada peta diperbarui secara otomatis tanpa perlu memuat ulang halaman. Apabila pengguna memilih salah satu kendaraan, sistem mengarahkan pengguna ke halaman Detail Kendaraan yang turut berlangganan pada kanal WebSocket yang sama untuk melanjutkan pembaruan posisi secara real-time.

UC-06 Melihat Status Perangkat GPS



Gambar 4.8. Diagram Sequence UC-06

Diagram Sequence UC-06 menggambarkan alur interaksi ketika Pemilik Kendaraan melihat status perangkat GPS pada halaman Home maupun halaman Detail Kendaraan. Status aktif atau tidak aktif suatu perangkat ditentukan oleh atribut `is_online` yang dihitung langsung pada model Device, berdasarkan apakah perangkat berstatus aktif dan waktu heartbeat terakhirnya masih berada dalam rentang lima menit terakhir. Pada halaman Home, HomeService membaca atribut tersebut melalui VehicleCardFormatter untuk ditampilkan pada setiap kartu kendaraan, sedangkan pada halaman Detail Kendaraan, Vehicle\VehicleRepository membaca atribut yang sama untuk ditampilkan pada bagian informasi perangkat. Karena keduanya membaca satu sumber perhitungan yang sama, status perangkat yang ditampilkan pada kedua halaman senantiasa konsisten. Apabila perangkat masih mengirimkan heartbeat dalam batas waktu tersebut, sistem menampilkan status aktif, sedangkan apabila melewati batas waktu tersebut, sistem menampilkan status tidak aktif.

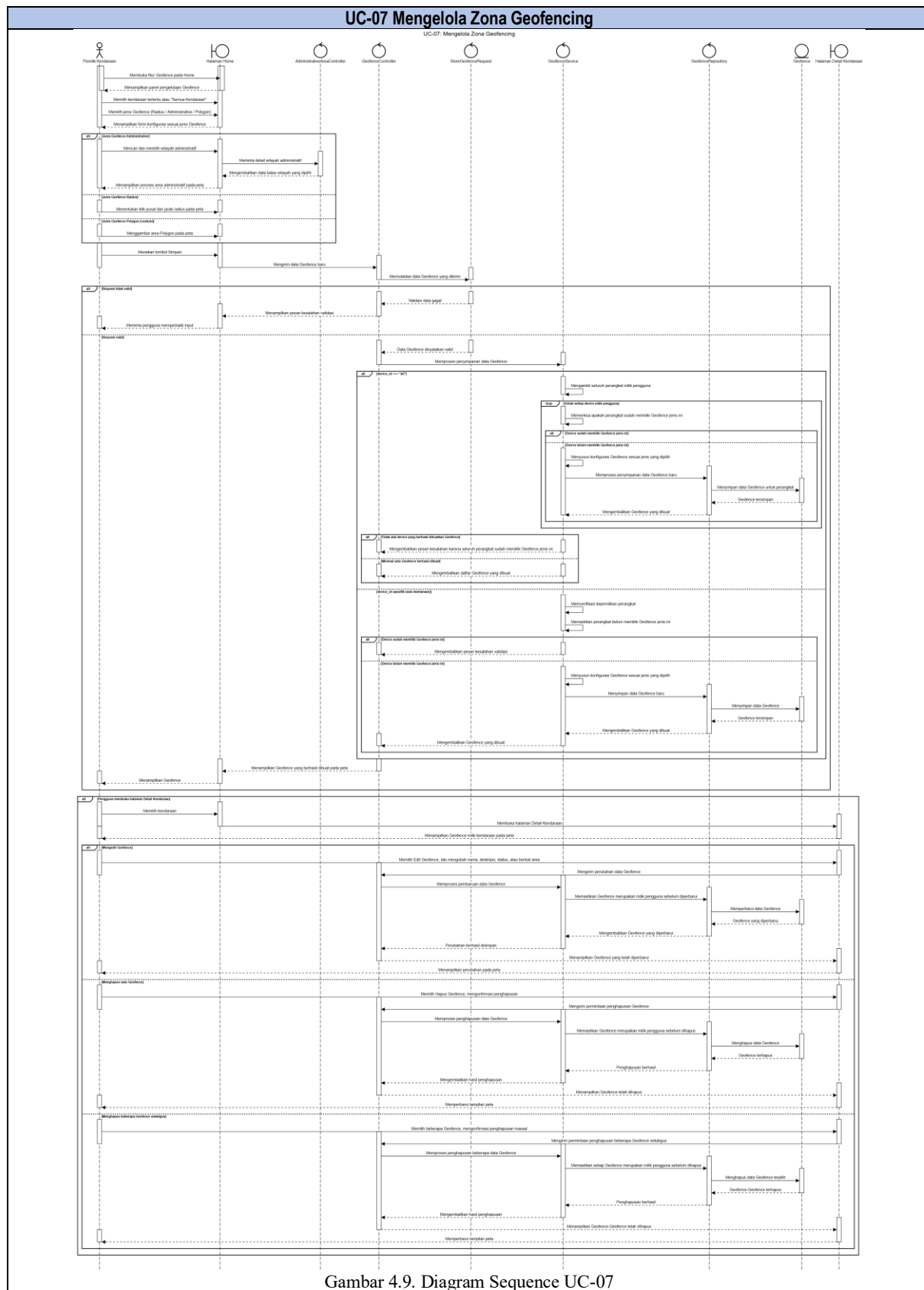
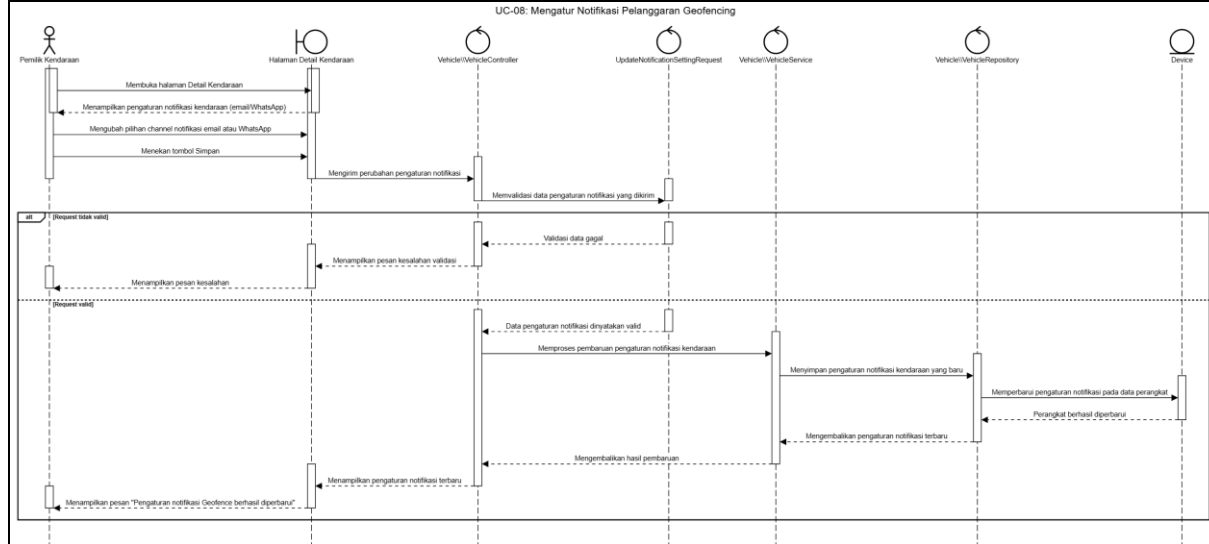


Diagram Sequence UC-07 menggambarkan alur interaksi ketika Pemilik Kendaraan mengelola zona geofencing. Proses diawali ketika pengguna membuka fitur Geofence pada Home, memilih kendaraan atau Semua Kendaraan, kemudian menentukan jenis geofence berupa wilayah administratif, radius, atau area bebas berbentuk polygon. Konfigurasi tersebut divalidasi sebelum diproses oleh GeofenceController dan GeofenceService, yang menyusun data konfigurasi sesuai jenis geofence yang dipilih. Apabila target berupa

UC-07 Mengelola Zona Geofencing

Semua Kendaraan, sistem membuat data geofence untuk setiap perangkat pengguna kecuali perangkat yang telah memiliki geofence dengan jenis yang sama, sesuai batasan satu geofence per jenis untuk setiap perangkat. Data yang telah divalidasi kemudian disimpan oleh GeofenceRepository ke dalam entitas Geofence dan ditampilkan pada peta. Pengguna juga dapat mengubah maupun menghapus geofence, baik satu per satu maupun sekaligus, melalui halaman Detail Kendaraan.

UC-08 Mengatur Notifikasi Pelanggaran Geofencing



Gambar 4.10. Diagram Sequence UC-08

Diagram Sequence UC-08 menggambarkan alur interaksi ketika Pemilik Kendaraan mengatur channel notifikasi pelanggaran geofencing pada suatu perangkat. Perlu dicatat bahwa pengaturan ini bukan merupakan atribut dari entitas Geofence, melainkan merupakan pengaturan notifikasi tingkat perangkat yang berlaku umum. Proses diawali ketika pengguna membuka halaman Detail Kendaraan dan memilih pengaturan notifikasi, kemudian menentukan status aktif email dan/atau WhatsApp. Data tersebut divalidasi sebelum diproses oleh Vehicle/VehicleController, kemudian Vehicle/VehicleService dan Vehicle/VehicleRepository memperbarui kolom notification_setting pada model Device. Setelah pengaturan berhasil disimpan, sistem menampilkan informasi bahwa pengaturan notifikasi Geofence telah diperbarui, dan pengaturan tersebut selanjutnya digunakan sistem untuk menentukan channel pengiriman ketika terjadi pelanggaran zona geofencing pada kendaraan tersebut.

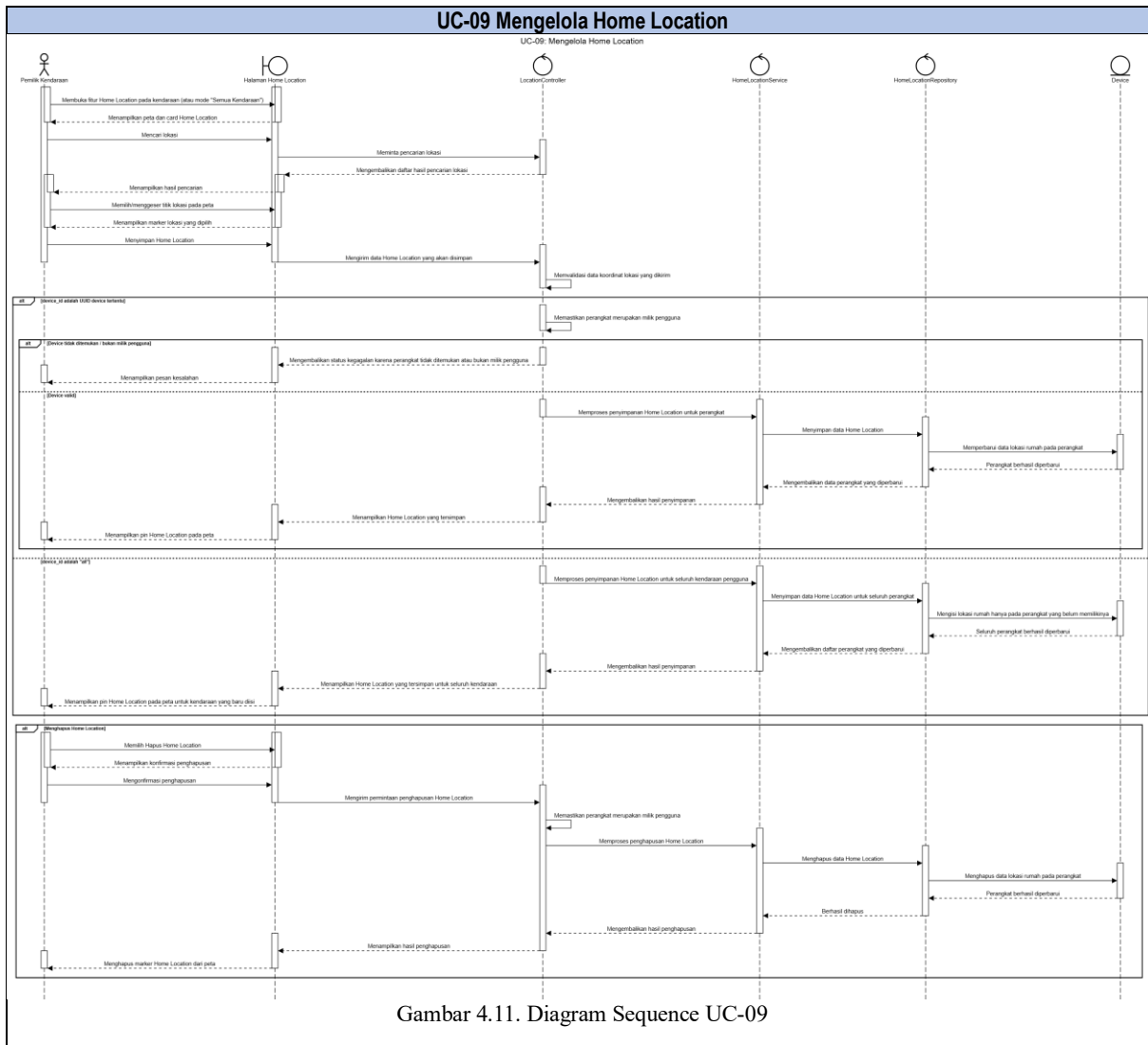


Diagram Sequence UC-09 menggambarkan alur interaksi ketika Pemilik Kendaraan mengatur Home Location. Proses diawali ketika pengguna melakukan pencarian lokasi dan menentukan titik pada peta, kemudian menyimpan koordinat tersebut untuk satu kendaraan tertentu atau sekaligus untuk seluruh kendaraan yang belum memiliki Home Location. Permintaan tersebut diproses oleh LocationController, kemudian diteruskan kepada HomeLocationService dan HomeLocationRepository, yang menyimpan koordinat tersebut langsung pada kolom home_location milik perangkat terkait tanpa memerlukan entitas tersendiri. Apabila terdapat geofence radius yang titik pusatnya mengikuti Home Location, sistem turut memperbarui titik pusat geofence tersebut secara otomatis. Setelah proses penyimpanan berhasil, sistem menampilkan Home Location pada peta. Pengguna juga dapat menghapus Home Location kapan saja, yang akan mengosongkan kembali kolom tersebut pada perangkat yang bersangkutan.

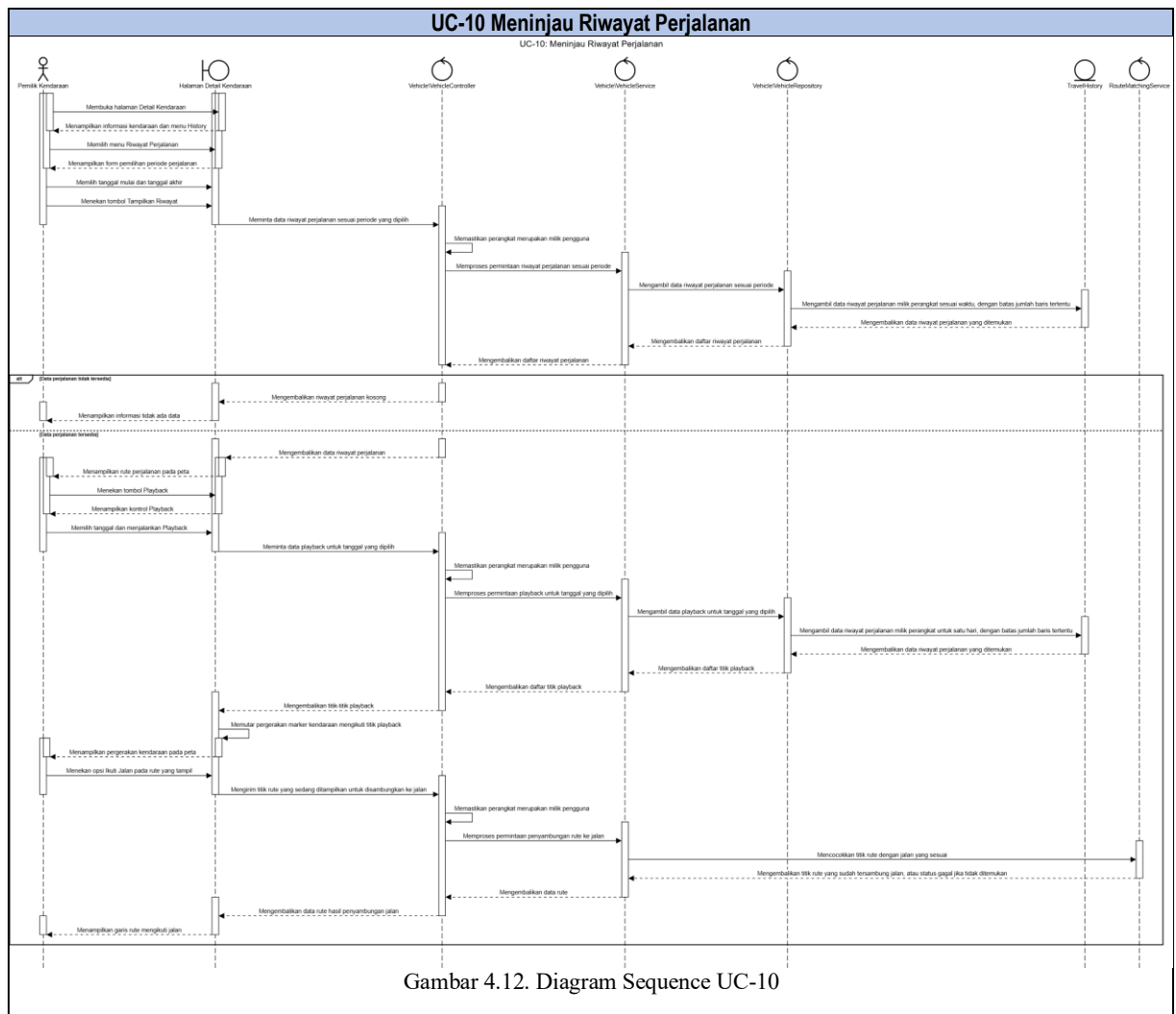


Diagram Sequence UC-10 menggambarkan alur interaksi ketika Pemilik Kendaraan meninjau riwayat perjalanan kendaraan. Proses diawali ketika pengguna membuka halaman Detail Kendaraan, memilih menu Riwayat Perjalanan, kemudian menentukan periode tanggal yang diinginkan. Permintaan tersebut diproses oleh Vehicle/HistoryController dan Vehicle/HistoryService, yang selanjutnya mengambil data melalui Vehicle/HistoryRepository dari entitas TravelHistory sesuai kendaraan dan periode yang dipilih. Apabila data tidak tersedia pada periode tersebut, sistem menampilkan informasi bahwa riwayat perjalanan tidak ditemukan. Apabila data tersedia, sistem menampilkan rute perjalanan pada peta dalam bentuk polyline. Pengguna juga dapat menjalankan fitur Playback pada rute yang ditampilkan, yang memutar kembali pergerakan kendaraan pada sisi antarmuka pengguna berdasarkan data yang telah diambil sebelumnya.

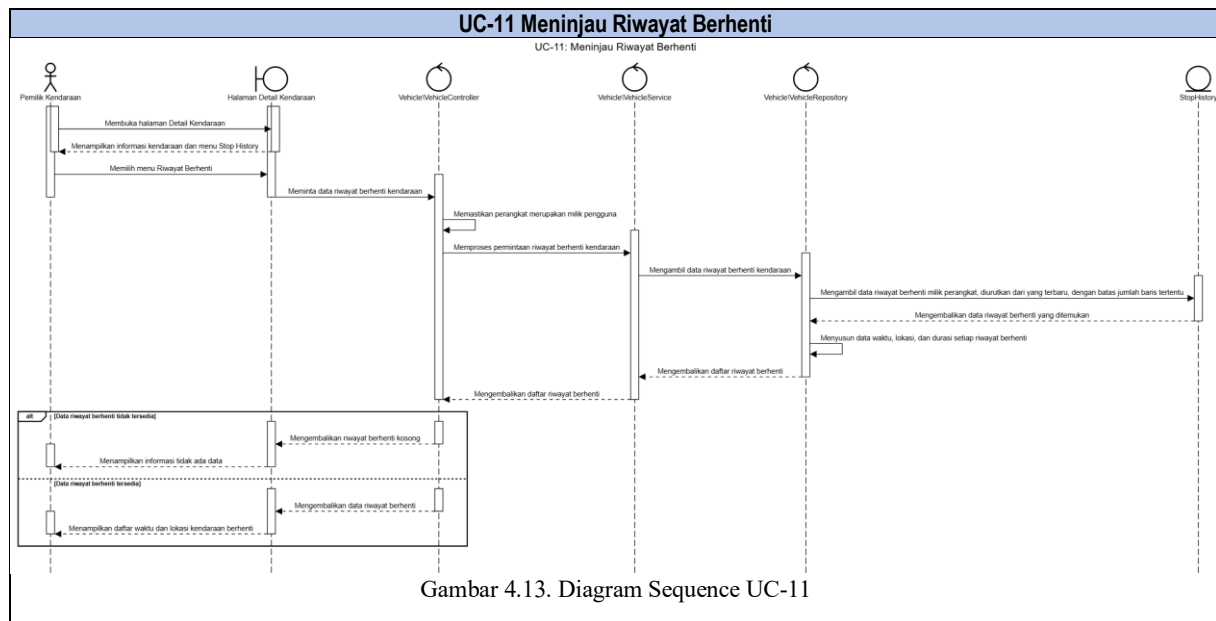


Diagram Sequence UC-11 menggambarkan alur interaksi ketika Pemilik Kendaraan meninjau riwayat berhenti kendaraan. Proses diawali ketika pengguna membuka halaman Detail Kendaraan dan memilih menu Riwayat Berhenti. Permintaan tersebut diproses oleh Vehicle/VehicleController dan Vehicle/VehicleService, kemudian Vehicle/VehicleRepository mengambil data dari entitas StopHistory milik kendaraan tersebut, diurutkan dari yang paling baru. Apabila belum terdapat data berhenti yang tercatat, sistem menampilkan informasi bahwa riwayat berhenti belum tersedia. Apabila data tersedia, sistem menampilkan daftar riwayat berhenti yang memuat informasi waktu mulai, durasi, serta lokasi kendaraan berhenti.

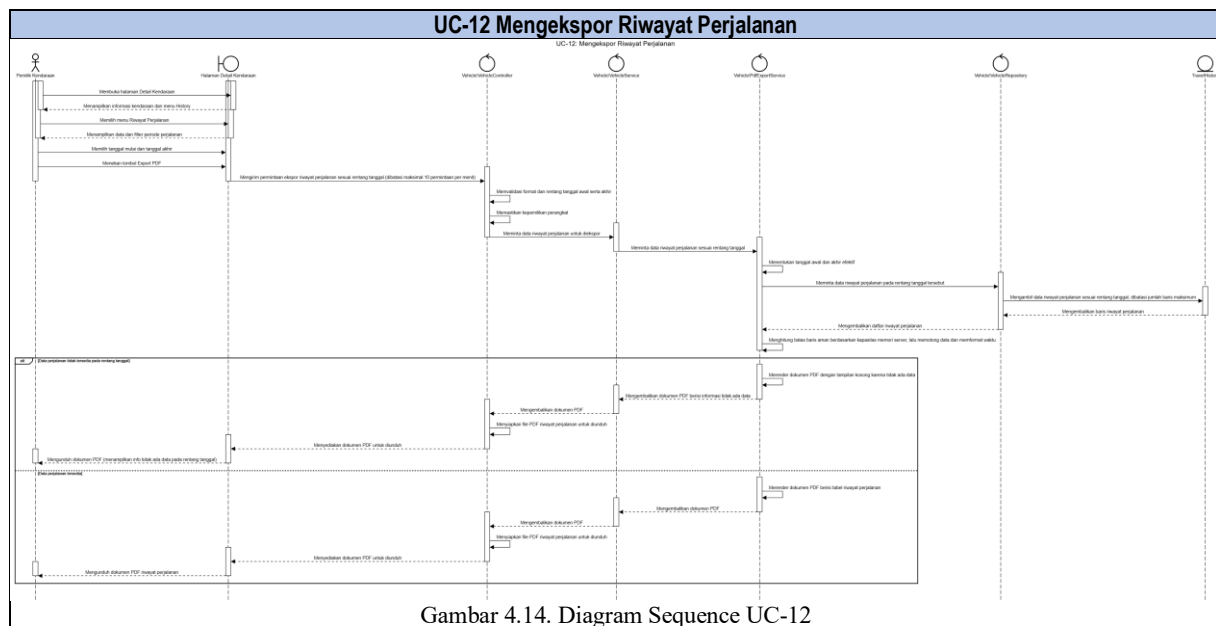
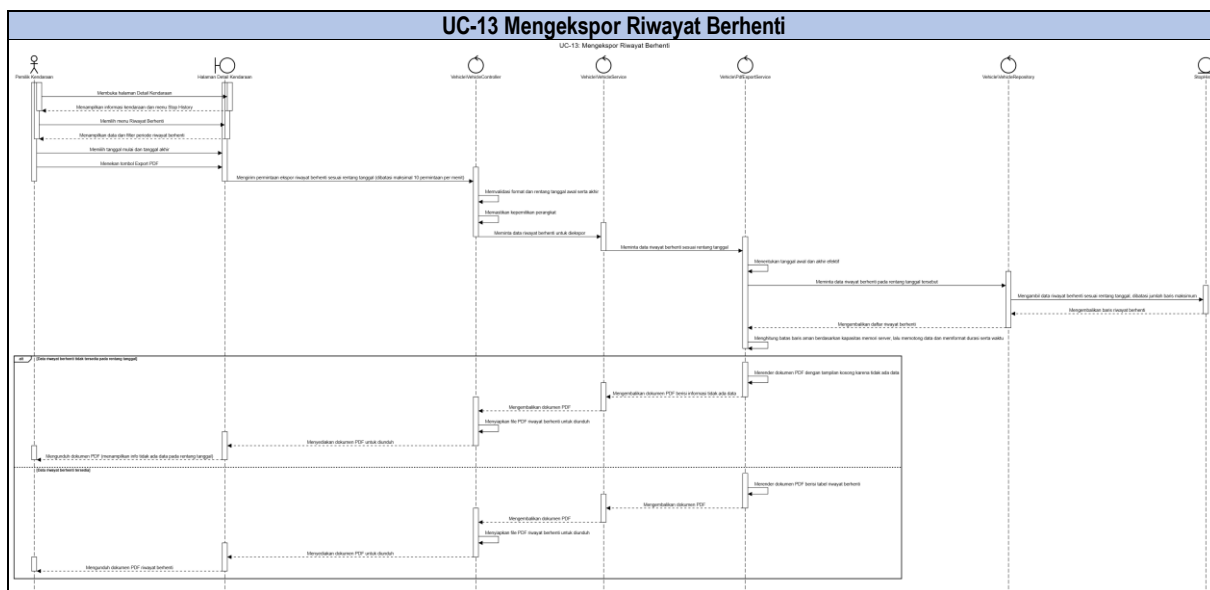
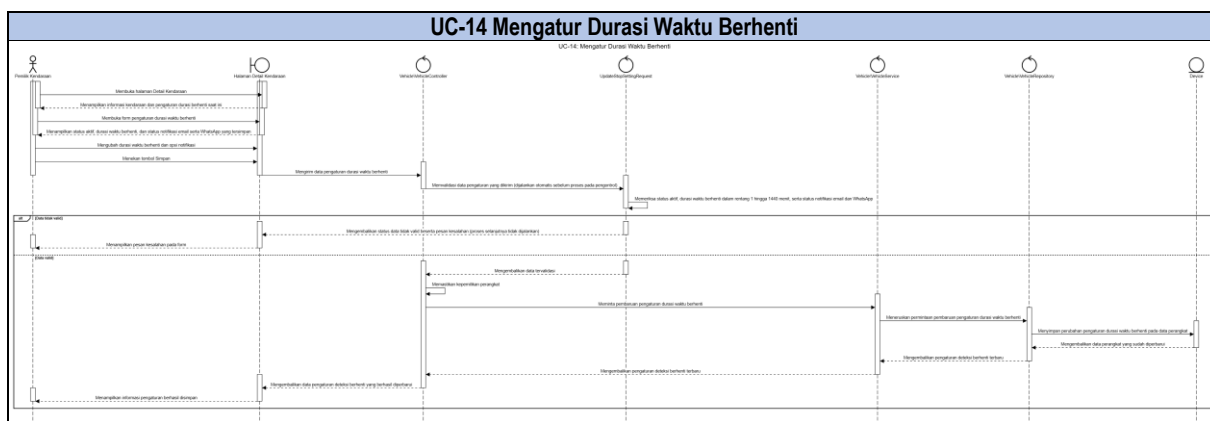


Diagram Sequence UC-12 menggambarkan alur interaksi ketika Pemilik Kendaraan mengekspor riwayat perjalanan ke dalam bentuk PDF. Proses diawali ketika pengguna menentukan periode perjalanan pada halaman Detail Kendaraan kemudian menekan tombol Export PDF. Permintaan tersebut diproses oleh Vehicle/VehicleController dan Vehicle/VehicleService, yang meneruskan permintaan kepada Vehicle/PdfExportService. Layanan tersebut mengambil kembali data riwayat perjalanan melalui Vehicle/VehicleRepository dari entitas TravelHistory, kemudian menyusun dokumen menggunakan tampilan Blade khusus yang dirender menjadi PDF melalui pustaka dompdf. Dokumen yang telah dihasilkan selanjutnya disediakan oleh sistem agar dapat diunduh oleh Pemilik Kendaraan.



Gambar 4.15. Diagram Sequence UC-13

Diagram Sequence UC-13 menggambarkan alur interaksi ketika Pemilik Kendaraan mengekspor riwayat berhenti ke dalam bentuk PDF. Proses ini serupa dengan proses ekspor riwayat perjalanan, yaitu pengguna menentukan periode pada halaman Detail Kendaraan kemudian menekan tombol Export PDF. Permintaan diproses oleh Vehicle\VehicleController dan Vehicle\VehicleService, kemudian diteruskan kepada Vehicle\PdfExportService yang mengambil data riwayat berhenti melalui Vehicle\VehicleRepository dari entitas StopHistory sesuai periode yang dipilih. Data tersebut selanjutnya disusun ke dalam dokumen PDF menggunakan tampilan Blade khusus dan pustaka dompdf, kemudian disediakan oleh sistem agar dapat diunduh oleh Pemilik Kendaraan.



Gambar 4.16. Diagram Sequence UC-14

Diagram Sequence UC-14 menggambarkan alur interaksi ketika Pemilik Kendaraan mengatur durasi waktu berhenti kendaraan. Nilai durasi yang tersimpan sebelumnya telah ditampilkan bersamaan dengan pemuatan awal halaman Detail Kendaraan, sehingga pengguna dapat langsung mengubah nilai tersebut tanpa permintaan tambahan. Setelah pengguna mengubah durasi dan menekan tombol Simpan, data dikirimkan kepada Vehicle\VehicleController dan divalidasi terlebih dahulu. Apabila nilai tidak valid, sistem secara otomatis mengembalikan pesan kesalahan sebelum permintaan diproses lebih lanjut. Apabila valid, Vehicle\VehicleService dan Vehicle\VehicleRepository memperbarui kolom stop_setting pada model Device, kemudian sistem menampilkan informasi bahwa pengaturan durasi berhenti telah berhasil disimpan.

Gambar 4.18. Diagram Sequence UC-16

Gambar 4.19. Diagram Sequence UC-17

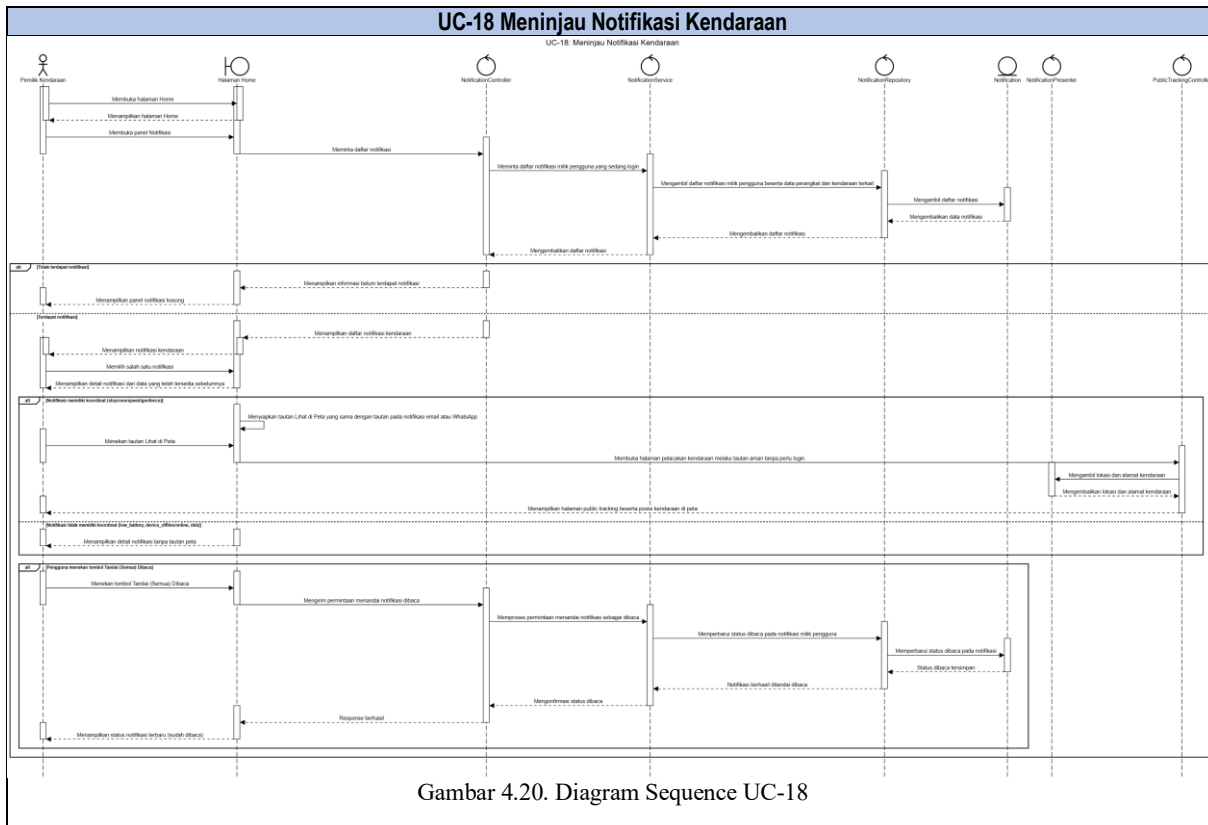


Diagram Sequence UC-18 menggambarkan alur interaksi ketika Pemilik Kendaraan meninjau notifikasi kendaraan melalui halaman Home. Proses diawali ketika pengguna membuka panel Notifikasi, kemudian NotificationController melalui NotificationService dan NotificationRepository mengambil daftar notifikasi milik pengguna dari entitas Notification. Pengguna dapat memilih salah satu notifikasi untuk melihat detailnya, dan apabila notifikasi tersebut memuat informasi koordinat, sistem menyediakan tautan Lihat di Peta berupa tautan bertanda tangan digital yang dibuat oleh NotificationPresenter, sehingga ketika dipilih akan mengarahkan pengguna menuju PublicTrackingController untuk menampilkan posisi kendaraan pada peta meskipun tanpa harus login. Pengguna juga dapat menandai notifikasi sebagai telah dibaca, baik satu per satu maupun sekaligus seluruhnya.

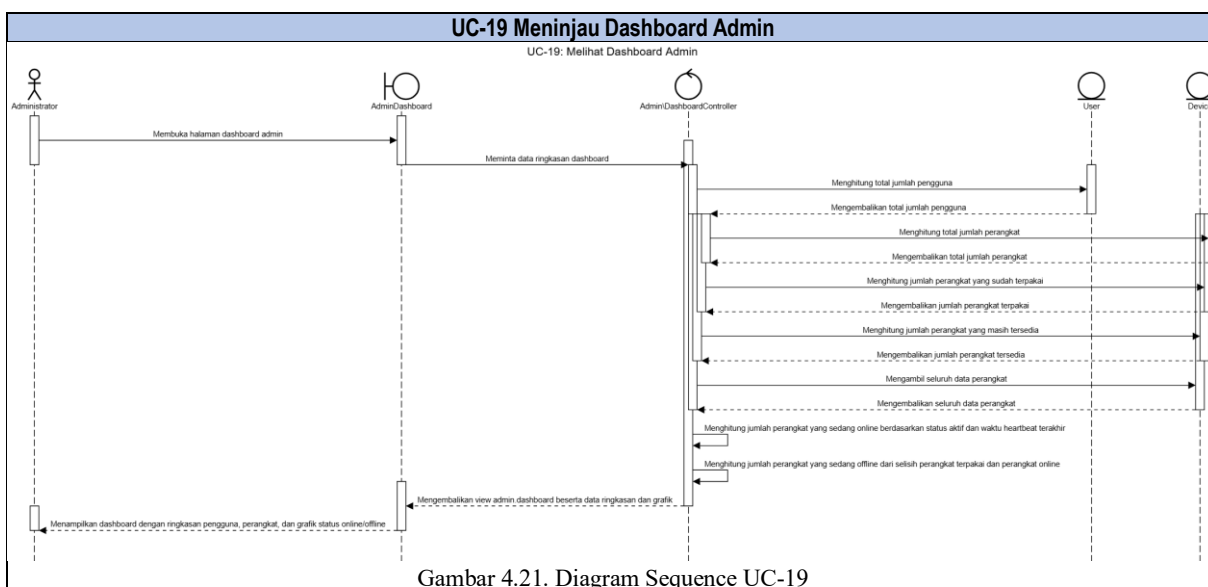
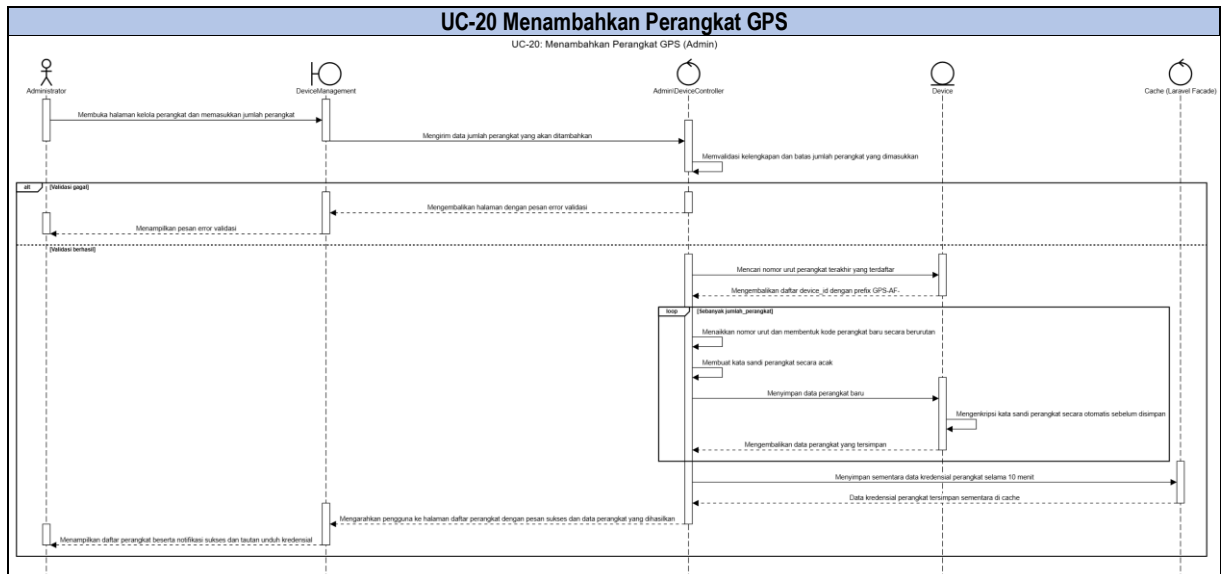
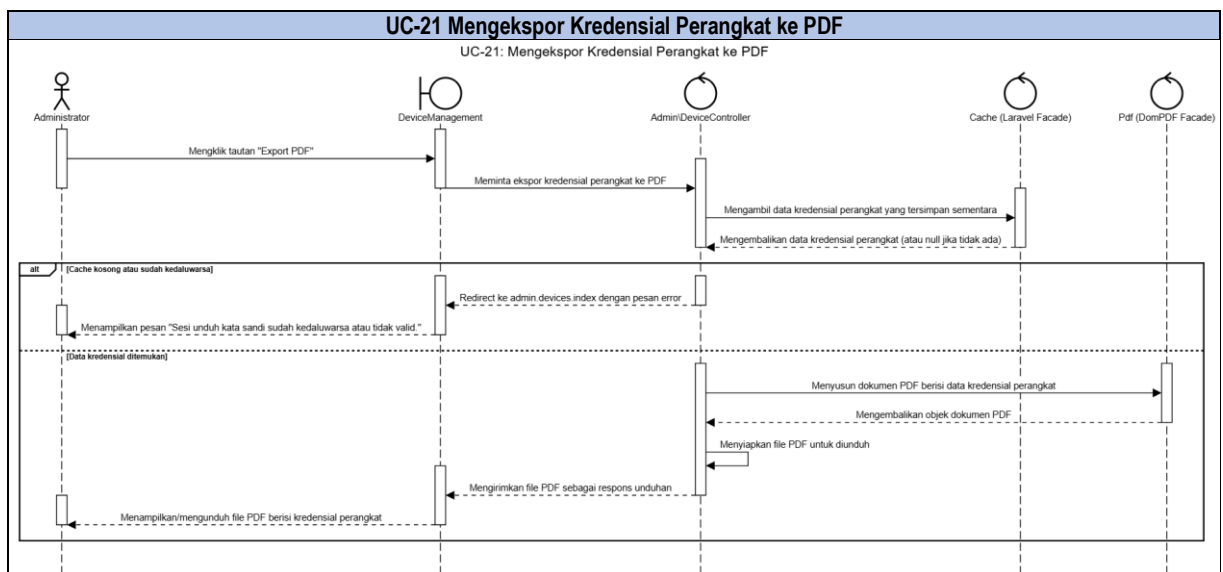


Diagram Sequence UC-19 menggambarkan alur interaksi ketika Administrator melihat dashboard admin. Proses diawali ketika Administrator membuka halaman Dashboard Admin, kemudian AdminDashboardController secara langsung mengambil jumlah pengguna dari model User serta jumlah dan status perangkat dari model Device, tanpa melalui lapisan service maupun repository terpisah. Status aktif atau tidak aktif setiap perangkat dihitung berdasarkan atribut is_online yang sama dengan yang digunakan pada halaman pemantauan kendaraan. Sistem kemudian menampilkan ringkasan jumlah pengguna, jumlah perangkat, serta grafik status online dan offline perangkat kepada Administrator, yang secara otomatis bernilai kosong apabila belum terdapat data pengguna maupun perangkat.



Gambar 4.22. Diagram Sequence UC-20

Diagram Sequence UC-20 menggambarkan alur interaksi ketika Administrator menambahkan perangkat GPS baru ke dalam sistem. Proses diawali ketika Administrator membuka halaman Kelola Perangkat dan memasukkan jumlah perangkat yang akan ditambahkan. Admin\DeviceController memvalidasi bahwa jumlah tersebut berada pada rentang satu hingga seratus, kemudian secara langsung membuat data perangkat baru sebanyak jumlah yang diminta pada model Device, tanpa melalui lapisan service maupun repository terpisah. Setiap perangkat memperoleh Device ID berurutan dengan awalan tertentu serta password acak yang dibuat oleh sistem. Data kredensial yang dihasilkan kemudian disimpan sementara pada cache selama sepuluh menit, dan Administrator ditampilkan daftar Device ID beserta password perangkat yang berhasil ditambahkan. Apabila jumlah perangkat tidak diisi atau berada di luar rentang yang ditentukan, sistem menampilkan pesan validasi dan meminta Administrator memperbaiki jumlah perangkat.



Gambar 4.23. Diagram Sequence UC-21

Diagram Sequence UC-21 menggambarkan alur interaksi ketika Administrator mengekspor kredensial perangkat GPS yang baru ditambahkan pada UC-20 ke dalam dokumen PDF. Proses diawali ketika Administrator menekan tombol Export PDF, kemudian Admin\DeviceController mengambil data kredensial perangkat dari cache sementara berdasarkan kunci sesi yang diberikan. Apabila data pada cache tidak ditemukan atau telah kedaluwarsa, sistem menampilkan pesan bahwa sesi unduh telah kedaluwarsa sehingga dokumen tidak dapat dibuat. Apabila data masih tersedia, sistem menyusun dokumen PDF berisi Device ID dan password perangkat menggunakan pustaka dompdf, kemudian menyediakan dokumen tersebut agar dapat diunduh oleh Administrator.

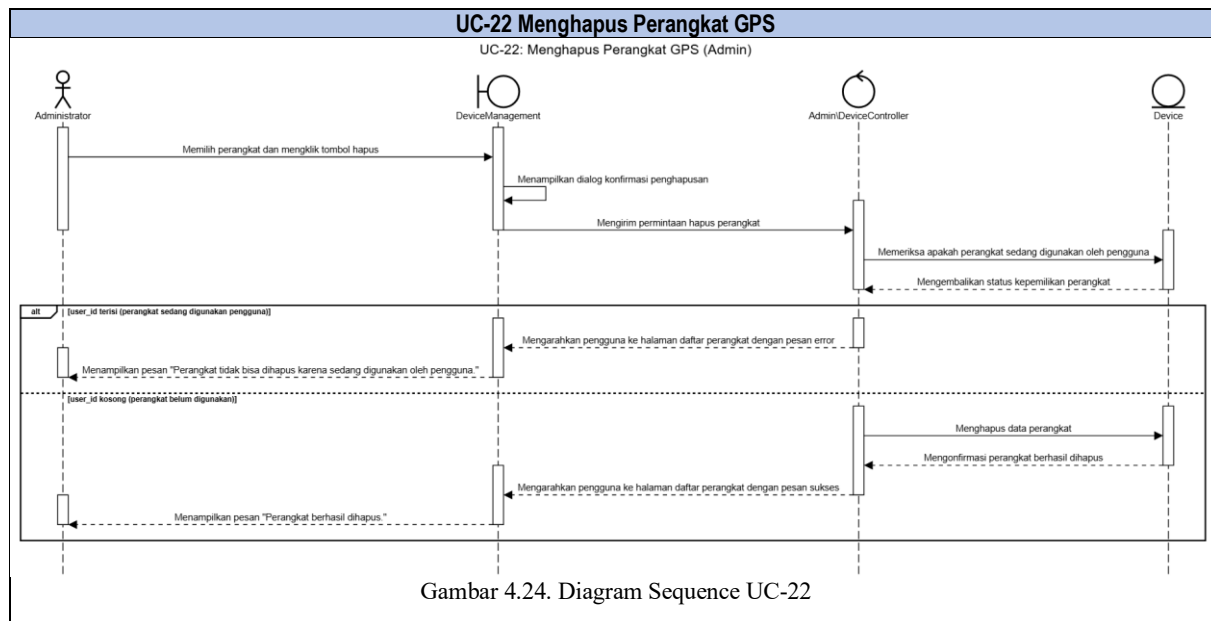


Diagram Sequence UC-22 menggambarkan alur interaksi ketika Administrator menghapus perangkat GPS dari sistem. Proses diawali ketika Administrator memilih perangkat pada halaman Kelola Perangkat, menekan tombol Hapus, dan mengonfirmasi penghapusan pada sisi antarmuka. Permintaan penghapusan kemudian dikirimkan kepada Admin\DeviceController, di mana perangkat pada alamat halaman secara otomatis diikat melalui route-model binding tanpa memerlukan pencarian data secara manual. Sistem selanjutnya memeriksa apakah perangkat tersebut masih terhubung dengan akun pengguna melalui kolom user_id pada perangkat. Apabila perangkat sedang digunakan oleh pengguna, sistem menolak penghapusan dan menampilkan pesan bahwa perangkat tidak dapat dihapus. Apabila perangkat belum terhubung dengan akun pengguna, sistem menghapus data perangkat secara langsung dan menampilkan daftar perangkat terbaru kepada Administrator.

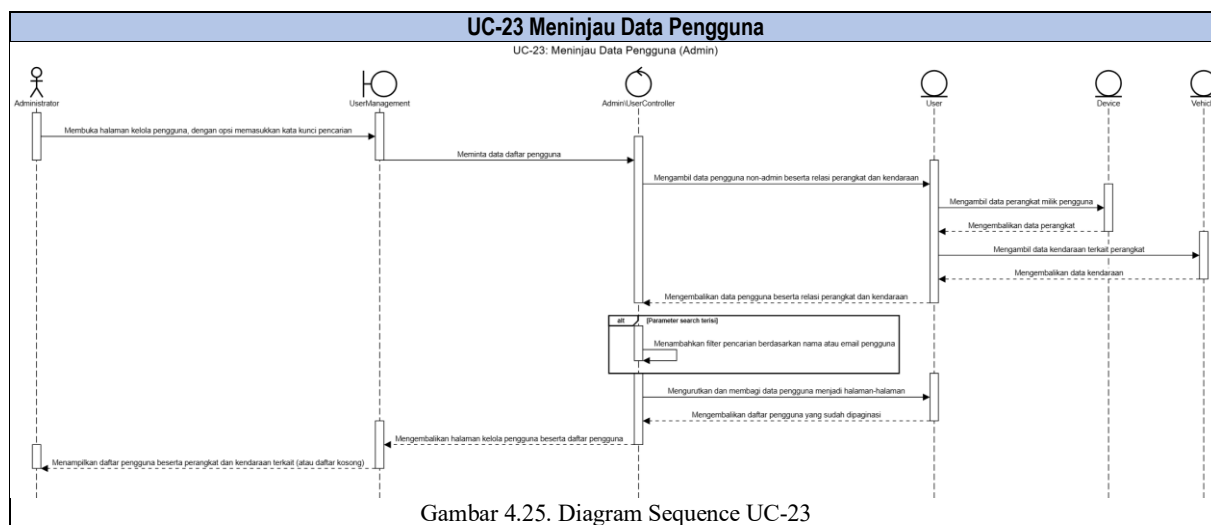


Diagram Sequence UC-23 menggambarkan alur interaksi ketika Administrator meninjau data pengguna pada halaman Kelola Pengguna. Proses diawali ketika Administrator membuka halaman Kelola Pengguna, kemudian Admin\UserController secara langsung mengambil seluruh data pengguna dari model User beserta perangkat dan kendaraan yang terkait melalui relasi Eloquent dalam satu permintaan, tanpa melalui lapisan service maupun repository terpisah. Administrator dapat pula memasukkan kata kunci berupa nama atau email pada kolom pencarian, yang diterapkan sebagai filter tambahan pada permintaan yang sama, bukan sebagai permintaan yang terpisah. Sistem kemudian menampilkan daftar pengguna yang sesuai dengan kata kunci pencarian, atau daftar seluruh pengguna apabila pencarian tidak digunakan.

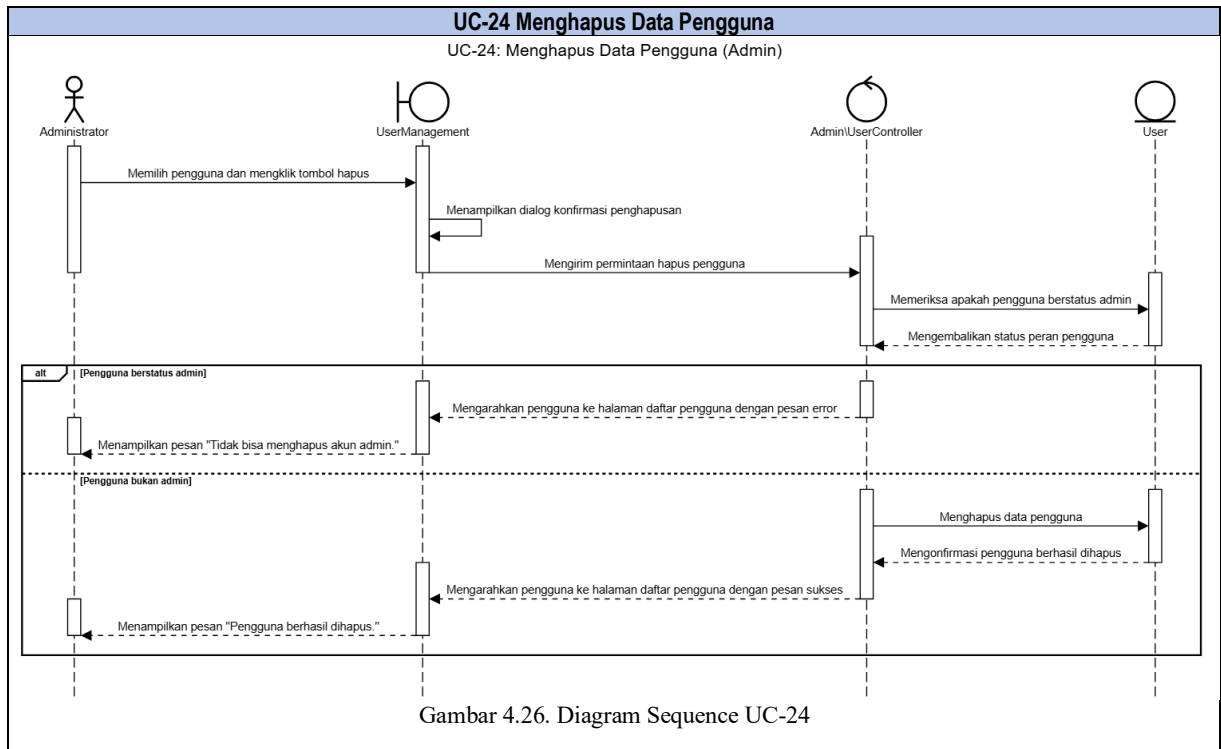
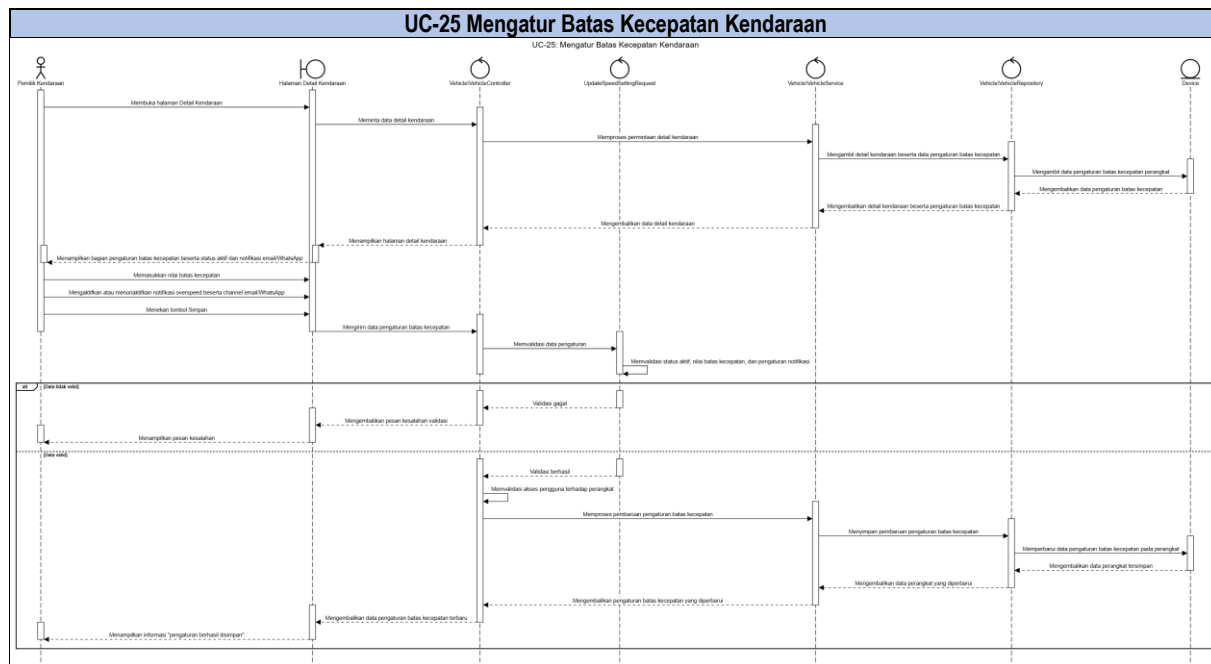
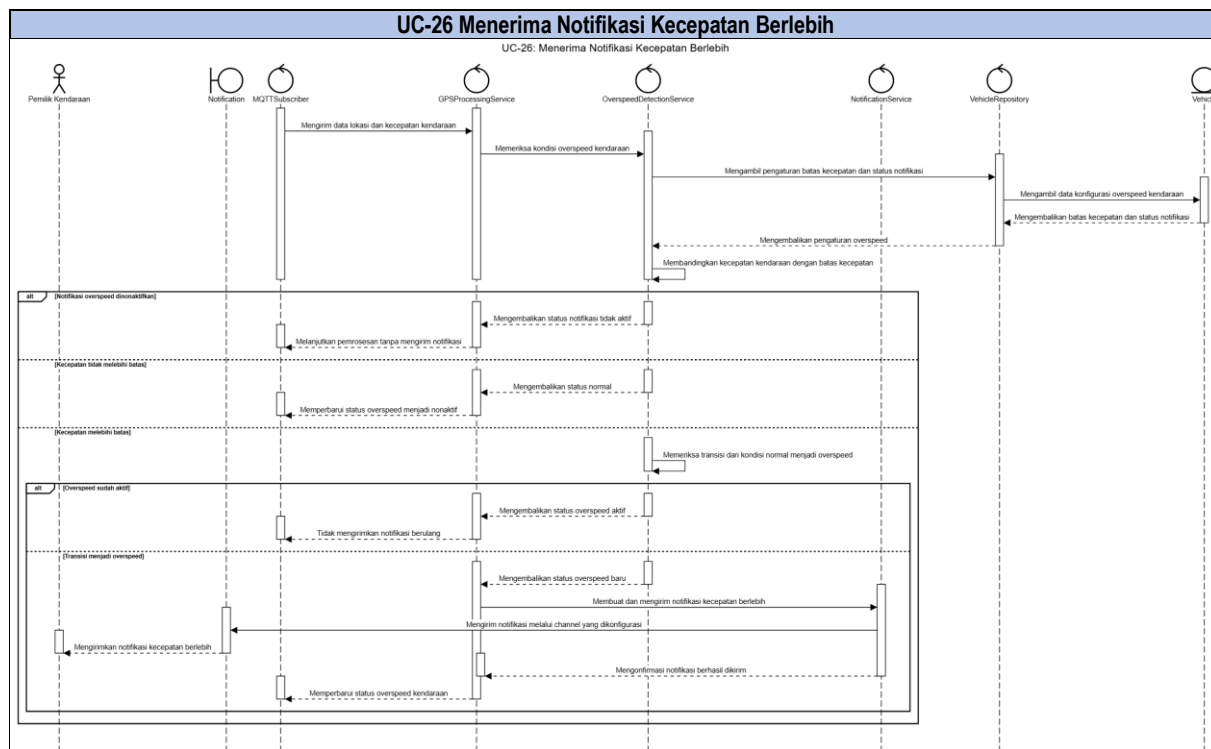


Diagram Sequence UC-24 menggambarkan alur interaksi ketika Administrator menghapus data pengguna pada halaman Kelola Pengguna. Proses diawali ketika Administrator memilih pengguna, menekan tombol Hapus, dan mengonfirmasi penghapusan pada sisi antarmuka. Permintaan tersebut dikirimkan kepada AdminUserController, yang memeriksa status akun pengguna secara langsung pada model User. Apabila akun yang dipilih merupakan akun admin, sistem menolak penghapusan dan menampilkan pesan bahwa akun admin tidak dapat dihapus. Apabila akun bukan merupakan akun admin, sistem menghapus data pengguna tersebut secara langsung. Penghapusan perangkat milik pengguna tidak dilakukan secara eksplisit oleh aplikasi, melainkan ditangani secara otomatis oleh basis data melalui aturan foreign key yang mengosongkan kolom user_id pada perangkat terkait, sehingga perangkat tetap tersedia dan dapat diaktivasi kembali oleh pengguna lain, sedangkan kendaraan yang terhubung dengan perangkat tersebut tidak ikut terhapus. Sistem selanjutnya menampilkan daftar pengguna terbaru kepada Administrator.



Gambar 4.27. Diagram Sequence UC-25

Diagram Sequence UC-25 menggambarkan alur interaksi ketika Pemilik Kendaraan mengatur batas kecepatan dan status notifikasi overspeed. Nilai batas kecepatan dan status notifikasi yang tersimpan sebelumnya telah ditampilkan bersamaan dengan pemuatan awal halaman Detail Kendaraan. Setelah pengguna memasukkan nilai batas kecepatan baru, mengaktifkan atau menonaktifkan notifikasi overspeed, dan menekan tombol Simpan, data dikirimkan kepada Vehicle/ServiceController dan divalidasi terlebih dahulu. Apabila nilai tidak valid, sistem secara otomatis mengembalikan pesan kesalahan. Apabila valid, Vehicle/Service dan Vehicle/Repository memperbarui kolom speed_setting pada model Device, kemudian sistem menampilkan informasi bahwa pengaturan batas kecepatan telah berhasil disimpan. Pengaturan tersebut selanjutnya digunakan oleh OverspeedDetectionService untuk mendeteksi kondisi kecepatan berlebih pada saat data lokasi kendaraan diterima.



Gambar 4.28. Diagram Sequence UC-26

Diagram Sequence UC-26 menggambarkan alur interaksi ketika sistem mendeteksi kendaraan melebihi batas kecepatan yang ditentukan dan mengirimkan notifikasi kepada Pemilik Kendaraan. Proses diawali ketika perangkat GPS mengirimkan data lokasi dan kecepatan melalui protokol MQTT, yang diterima oleh MQTTSubscriberService dan diproses lebih lanjut oleh GPSProcessingService. Selanjutnya,

UC-26 Menerima Notifikasi Kecepatan Berlebih
OverspeedDetectionService membandingkan kecepatan kendaraan yang diterima dengan batas kecepatan yang telah dikonfigurasi pada kolom speed_setting milik perangkat. Apabila kendaraan mengalami transisi dari kondisi normal menjadi melebihi batas kecepatan dan notifikasi overspeed telah diaktifkan, NotificationService menyimpan data notifikasi melalui NotificationRepository. Sistem selanjutnya menyiarkan notifikasi secara real-time pada antarmuka pengguna, serta memproses pengiriman notifikasi email dan/atau WhatsApp secara asinkron melalui antrean (queue) sesuai channel yang telah dikonfigurasi. Apabila kendaraan belum melebihi batas kecepatan atau notifikasi overspeed dinonaktifkan, sistem tidak mengirimkan notifikasi.

4.1.4 Skenario Use Case

Skenario Use Case merupakan penjelasan lebih rinci mengenai alur interaksi antara Pemilik Kendaraan dengan sistem dalam menjalankan setiap Use Case yang telah didefinisikan. Skenario ini menjelaskan kondisi awal sebelum proses dilakukan, tujuan dari setiap Use Case, kondisi setelah proses selesai, serta tahapan interaksi antara pengguna dan sistem. Selain skenario utama, skenario alternatif juga digunakan untuk menjelaskan kondisi yang dapat terjadi apabila proses utama tidak berjalan sesuai dengan kondisi yang diharapkan. Berikut merupakan skenario Use Case pada sistem AF GPS Tracker.

4.1.4.1. UC-01 Menambahkan Data Kendaraan

Rincian skenario dari proses menambahkan data kendaraan, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.3. Skenario UC-01

Identifikasi		
Use Case ID	UC-01	
Nama Use Case	Menambahkan Data Kendaraan	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan memiliki perangkat GPS yang telah diaktifasi.	
Postkondisi	Data kendaraan tersimpan dan terhubung dengan perangkat GPS.	
Tujuan	Mendaftarkan kendaraan yang akan dipantau oleh sistem.	
Skenario Utama		
Kondisi Awal: Halaman Data Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan berada pada halaman pengisian kendaraan.	Sistem menampilkan formulir kendaraan.
2	Pemilik Kendaraan mengisi nama kendaraan, nomor polisi, tipe, warna, dan ikon/marker kendaraan.	Sistem memvalidasi data.
3	Pemilik Kendaraan menekan tombol Simpan.	Sistem menyimpan data kendaraan dan menghubungkannya dengan perangkat GPS.
4		Sistem menampilkan informasi bahwa kendaraan berhasil ditambahkan.
5		Sistem mengarahkan pengguna ke Home.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data kendaraan belum lengkap.	Sistem menampilkan pesan validasi.
A2	Pengguna membatalkan proses.	Sistem tidak menyimpan data kendaraan.

4.1.4.2. UC-02 Menambahkan Perangkat GPS

Tabel berikut menjabarkan skenario Use Case untuk proses menambahkan perangkat GPS secara lebih rinci.

Tabel 4.4. Skenario UC-02

Identifikasi		
Use Case ID	UC-02	
Nama Use Case	Menambahkan Perangkat GPS	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan memiliki akun aktif.	
Postkondisi	Perangkat GPS baru berhasil terhubung dengan akun pengguna.	
Tujuan	Menambahkan perangkat GPS tambahan tanpa membuat akun pengguna baru.	
Skenario Utama		
Kondisi Awal: Halaman home/detail kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan memilih Tambah Perangkat pada sidebar.	Sistem menampilkan modal penambahan perangkat.
2	Pemilik Kendaraan memasukkan Device ID dan password perangkat.	Sistem menerima data perangkat.
3	Pemilik Kendaraan menekan tombol aktivasi.	Sistem memvalidasi perangkat.
4		Sistem menghubungkan perangkat dengan akun pengguna.
5		Sistem mengarahkan pengguna untuk mengisi data kendaraan.
6	Pemilik Kendaraan mengisi data kendaraan.	Sistem menyimpan kendaraan dan perangkat baru.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Device ID atau password salah.	Sistem menolak penambahan perangkat.
A2	Perangkat telah digunakan akun lain.	Sistem menampilkan informasi bahwa perangkat tidak dapat ditambahkan.

4.1.4.3. UC-03 Mengubah Data Kendaraan

Penjelasan langkah demi langkah mengenai proses mengubah data kendaraan, beserta kemungkinan skenario alternatifnya, disajikan pada tabel berikut.

Tabel 4.5. Skenario UC-03

Identifikasi		
Use Case ID	UC-03	
Nama Use Case	Mengubah Data Kendaraan	
Aktor	Pemilik Kendaraan	
Prekondisi	Data kendaraan telah terdaftar.	
Postkondisi	Informasi kendaraan berhasil diperbarui.	
Tujuan	Memperbarui informasi kendaraan sesuai kondisi terbaru.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka detail kendaraan.	Sistem menampilkan informasi kendaraan.
2	Pemilik Kendaraan memilih fitur Edit Kendaraan.	Sistem menampilkan formulir data kendaraan.
3	Pemilik Kendaraan mengubah data kendaraan.	Sistem memvalidasi perubahan.
4	Pemilik Kendaraan menyimpan perubahan.	Sistem memperbarui data kendaraan.
5		Sistem menampilkan data kendaraan yang telah diperbarui.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data perubahan tidak valid.	Sistem menampilkan pesan kesalahan.
A2	Pengguna membatalkan perubahan.	Sistem mempertahankan data sebelumnya.

4.1.4.4. UC-04 Menghapus Kendaraan

Skenario yang menggambarkan interaksi pengguna dan sistem pada proses menghapus kendaraan dijelaskan secara rinci melalui tabel berikut.

Tabel 4.6. Skenario UC-04

Identifikasi		
Use Case ID	UC-04	
Nama Use Case	Menghapus Kendaraan	
Aktor	Pemilik Kendaraan	
Prekondisi	Kendaraan telah terdaftar pada akun pengguna.	
Postkondisi	Kendaraan tidak lagi terhubung dengan akun pengguna.	
Tujuan	Menghapus keterkaitan kendaraan dan perangkat dari akun pengguna ketika kendaraan tidak lagi digunakan.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka detail kendaraan.	Sistem menampilkan informasi kendaraan.
2	Pemilik Kendaraan memilih Hapus Kendaraan.	Sistem menampilkan konfirmasi penghapusan.
3	Pemilik Kendaraan mengonfirmasi penghapusan.	Sistem melakukan proses unbind kendaraan.
4		Sistem menghapus keterkaitan kendaraan dari akun.
5		Sistem menampilkan informasi bahwa kendaraan berhasil dihapus.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Pengguna membatalkan penghapusan.	Sistem mempertahankan kendaraan.

4.1.4.5. UC-05 Memantau Kendaraan Secara Real-time

Tabel berikut merinci prekondisi, postkondisi, serta alur normal dan alternatif dari proses memantau kendaraan secara real-time.

Tabel 4.7. Skenario UC-05

Identifikasi		
Use Case ID	UC-05	
Nama Use Case	Memantau Kendaraan secara Real-time	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan memiliki kendaraan yang terdaftar.	
Postkondisi	Posisi dan informasi kendaraan ditampilkan serta diperbarui secara real-time.	
Tujuan	Memungkinkan pengguna memantau kondisi dan pergerakan kendaraan secara langsung.	
Skenario Utama		
Kondisi Awal: Halaman Home		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka Halaman Home.	Sistem menampilkan peta dan seluruh kendaraan pengguna.
2	Pemilik Kendaraan memilih kendaraan.	Sistem menampilkan posisi kendaraan pada peta.
3		Sistem menerima pembaruan data lokasi kendaraan.
4		Sistem memperbarui posisi marker kendaraan secara real-time.
5	Pemilik Kendaraan memilih detail kendaraan.	Sistem menampilkan peta khusus kendaraan dan informasi pemantauan.
6		Sistem menampilkan informasi lokasi dan kecepatan kendaraan terbaru.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data lokasi belum tersedia.	Sistem menampilkan informasi bahwa lokasi belum tersedia.
A2	Sistem tidak menerima pembaruan lokasi.	Sistem mempertahankan informasi posisi terakhir yang tersedia.
A3	Koneksi real-time terputus.	Sistem menampilkan status koneksi dan menunggu koneksi kembali.

4.1.4.6. UC-06 Melihat Status Perangkat GPS

Rincian skenario dari proses melihat status perangkat GPS, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.8. Skenario UC-06

Identifikasi		
Use Case ID	UC-06	
Nama Use Case	Melihat Status Perangkat GPS	
Aktor	Pemilik Kendaraan	
Prekondisi	Perangkat GPS telah terhubung dengan kendaraan.	
Postkondisi	Status perangkat ditampilkan pada website.	
Tujuan	Mengetahui apakah perangkat GPS sedang aktif atau tidak aktif.	
Skenario Utama		
Kondisi Awal: Halaman Home		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka Home atau Detail Kendaraan.	Sistem mengambil informasi status perangkat.
2		Sistem memeriksa data aktivitas/heartbeat perangkat.
3		Sistem menentukan status perangkat.
4		Sistem menampilkan status aktif atau tidak aktif.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Perangkat tidak mengirimkan heartbeat dalam batas waktu yang ditentukan.	Sistem menampilkan perangkat sebagai tidak aktif.

4.1.4.7. UC-07 Mengelola Zona Geofencing

Tabel berikut menjabarkan skenario Use Case untuk proses mengelola zona geofencing secara lebih rinci.

Tabel 4.9. Skenario UC-07

Identifikasi		
Use Case ID	UC-07	
Nama Use Case	Mengelola Zona Geofencing	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan memiliki kendaraan.	
Postkondisi	Zona geofencing berhasil dibuat atau diperbarui sesuai kebutuhan pengguna.	
Tujuan	Menentukan wilayah pemantauan kendaraan dan mengelola zona geofencing yang digunakan sistem.	
Skenario Utama		
Kondisi Awal: Halaman Home		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka fitur geofencing.	Sistem menampilkan peta dan geofence yang tersedia.
2	Pemilik Kendaraan memilih kendaraan yang akan menggunakan geofence.	Sistem menampilkan pilihan jenis geofence.
3	Pemilik Kendaraan memilih jenis geofence.	Sistem menampilkan konfigurasi sesuai jenis yang dipilih.
4	Pemilik Kendaraan menentukan wilayah administratif, titik dan radius, atau menggambar polygon.	Sistem menampilkan bentuk zona pada peta.
5	Pemilik Kendaraan menyimpan konfigurasi.	Sistem memvalidasi konfigurasi geofence.
6		Sistem menyimpan zona geofencing.
7		Sistem menampilkan geofence pada peta.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data geofence tidak lengkap.	Sistem meminta konfigurasi dilengkapi.
A2	Data wilayah administratif tidak tersedia.	Sistem menampilkan pesan bahwa wilayah tidak dapat dimuat.
A3	Pengguna membatalkan proses.	Sistem tidak menyimpan perubahan.

Identifikasi		
Use Case ID	UC-07	
A4	Pengguna menghapus geofence.	Sistem meminta konfirmasi dan menghapus geofence setelah dikonfirmasi.

4.1.4.8. UC-08 Mengatur Notifikasi Pelanggaran Geofencing

Penjelasan langkah demi langkah mengenai proses mengatur notifikasi pelanggaran geofencing, beserta kemungkinan skenario alternatifnya, disajikan pada tabel berikut.

Tabel 4.10. Skenario UC-08

Identifikasi		
Use Case ID	UC-08	
Nama Use Case	Mengatur Notifikasi Pelanggaran Geofencing	
Aktor	Pemilik Kendaraan	
Prekondisi	Geofence telah tersedia untuk kendaraan.	
Postkondisi	Pengaturan channel notifikasi geofence tersimpan.	
Tujuan	Menentukan channel yang digunakan untuk memberikan notifikasi ketika terjadi pelanggaran geofencing.	
Skenario Utama		
Kondisi Awal: Halaman Home		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka pengaturan geofence kendaraan.	Sistem menampilkan pengaturan notifikasi geofence.
2	Pemilik Kendaraan memilih channel notifikasi yang tersedia.	Sistem menerima konfigurasi.
3	Pemilik Kendaraan menyimpan pengaturan.	Sistem menyimpan konfigurasi notifikasi.
4		Sistem menampilkan informasi bahwa pengaturan berhasil disimpan.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Pengguna membatalkan perubahan.	Sistem mempertahankan pengaturan sebelumnya.
A2	Pengaturan gagal disimpan.	Sistem menampilkan pesan kesalahan.

4.1.4.9. UC-09 Mengelola Home Location

Skenario yang menggambarkan interaksi pengguna dan sistem pada proses mengelola Home Location dijelaskan secara rinci melalui tabel berikut.

Tabel 4.11. Skenario UC-09

Identifikasi		
Use Case ID	UC-09	
Nama Use Case	Mengelola Home Location	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan kendaraan telah tersedia.	
Postkondisi	Home Location kendaraan berhasil disimpan, diperbarui, atau dihapus.	
Tujuan	Menetapkan titik referensi lokasi untuk kendaraan.	
Skenario Utama		
Kondisi Awal: Halaman Detail kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan memilih fitur Home Location.	Sistem menampilkan peta dan card home location
2	Pemilik menekan tombol tambah	Sistem menampilkan formulir home location
3	Pemilik Kendaraan Mencari lokasi dan menentukan titik lokasi pada peta.	Sistem menampilkan marker Home Location.

Identifikasi		
Use Case ID	UC-09	
4	Pemilik Kendaraan menyimpan lokasi.	Sistem menyimpan koordinat Home Location kendaraan.
5		Sistem menampilkan Home Location pada peta.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Pengguna ingin memindahkan Home Location.	Sistem memperbarui koordinat berdasarkan titik baru.
A2	Pengguna memilih hapus Home Location.	Sistem meminta konfirmasi kemudian menghapus lokasi.
A3	Pengguna membatalkan proses.	Sistem mempertahankan lokasi sebelumnya.

4.1.4.10. UC-10 Meninjau Riwayat Perjalanan

Tabel berikut merinci prekondisi, postkondisi, serta alur normal dan alternatif dari proses meninjau riwayat perjalanan.

Tabel 4.12. Skenario UC-10

Identifikasi		
Use Case ID	UC-10	
Nama Use Case	Meninjau Riwayat Perjalanan	
Aktor	Pemilik Kendaraan	
Prekondisi	Data perjalanan kendaraan telah tersimpan pada sistem.	
Postkondisi	Riwayat perjalanan dan rute kendaraan ditampilkan pada peta.	
Tujuan	Mengetahui perjalanan kendaraan pada periode tertentu dan melihat kembali pergerakannya melalui playback.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka halaman riwayat perjalanan.	Sistem menampilkan pilihan kendaraan dan periode.
2	Pemilik Kendaraan memilih kendaraan dan rentang tanggal.	Sistem mengambil data perjalanan sesuai parameter.
3		Sistem menampilkan rute perjalanan dalam bentuk garis pada peta.
4	Pemilik Kendaraan memilih fitur Playback.	Sistem menampilkan kontrol playback.
5	Pemilik Kendaraan menjalankan playback.	Sistem menampilkan pergerakan kendaraan mengikuti rute perjalanan.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Tidak terdapat data pada periode yang dipilih.	Sistem menampilkan informasi bahwa riwayat tidak tersedia.
A2	Data riwayat gagal dimuat.	Sistem menampilkan pesan kesalahan.

4.1.4.11. UC-11 Meninjau Riwayat Berhenti

Rincian skenario dari proses meninjau riwayat berhenti, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.13. Skenario UC-11

Identifikasi	
Use Case ID	UC-11
Nama Use Case	Meninjau Riwayat Berhenti
Aktor	Pemilik Kendaraan
Prekondisi	Data perjalanan dan titik berhenti kendaraan tersedia.
Postkondisi	Daftar riwayat berhenti kendaraan ditampilkan.
Tujuan	Mengetahui lokasi dan waktu kendaraan berhenti berdasarkan data yang tersimpan.
Skenario Utama	
Kondisi Awal: Halaman Detail Kendaraan	

Identifikasi		
Use Case ID	UC-11	
No	Aktor	Sistem
1	Pemilik Kendaraan membuka halaman riwayat berhenti.	Sistem menampilkan pilihan kendaraan dan periode.
2	Pemilik Kendaraan memilih kendaraan dan periode.	Sistem mengambil data berhenti.
3		Sistem menampilkan daftar titik berhenti kendaraan.
4	Pemilik Kendaraan memilih salah satu data berhenti.	Sistem menampilkan informasi lokasi dan waktu berhenti.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Tidak terdapat data berhenti.	Sistem menampilkan informasi bahwa data berhenti tidak tersedia.
A2	Data gagal dimuat.	Sistem menampilkan pesan kesalahan.

4.1.4.12. UC-12 Mengekspor Riwayat Perjalanan

Tabel berikut menjabarkan skenario Use Case untuk proses mengekspor riwayat perjalanan secara lebih rinci.

Tabel 4.14. Skenario UC-12

Identifikasi		
Use Case ID	UC-12	
Nama Use Case	Mengekspor Riwayat Perjalanan	
Aktor	Pemilik Kendaraan	
Prekondisi	Data riwayat perjalanan tersedia.	
Postkondisi	Dokumen PDF riwayat perjalanan berhasil dihasilkan dan dapat diunduh.	
Tujuan	Menyediakan dokumentasi riwayat perjalanan dalam bentuk PDF.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka riwayat perjalanan.	Sistem menampilkan data perjalanan.
2	Pemilik Kendaraan menentukan kendaraan dan periode perjalanan.	Sistem menampilkan data sesuai pilihan.
3	Pemilik Kendaraan memilih Export PDF.	Sistem memproses data perjalanan.
4		Sistem menghasilkan dokumen PDF riwayat perjalanan.
5		Sistem menyediakan dokumen PDF untuk diunduh.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Tidak ada data perjalanan.	Sistem tidak menghasilkan dokumen dan menampilkan informasi bahwa data tidak tersedia.
A2	Proses pembuatan PDF gagal.	Sistem menampilkan pesan kesalahan.

4.1.4.13. UC-13 Mengekspor Riwayat Berhenti

Penjelasan langkah demi langkah mengenai proses mengekspor riwayat berhenti, beserta kemungkinan skenario alternatifnya, disajikan pada tabel berikut.

Tabel 4.15. Skenario UC-13

Identifikasi	
Use Case ID	UC-13
Nama Use Case	Mengekspor Riwayat Berhenti
Aktor	Pemilik Kendaraan
Prekondisi	Data riwayat berhenti tersedia.

Identifikasi		
Use Case ID	UC-13	
Postkondisi	Dokumen PDF riwayat berhenti berhasil dihasilkan dan dapat diunduh.	
Tujuan	Menyediakan dokumentasi riwayat berhenti kendaraan dalam bentuk PDF.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka riwayat berhenti.	Sistem menampilkan data riwayat berhenti.
2	Pemilik Kendaraan memilih kendaraan dan periode.	Sistem menampilkan data sesuai pilihan.
3	Pemilik Kendaraan memilih Export PDF.	Sistem memproses data berhenti.
4		Sistem menghasilkan dokumen PDF.
5		Sistem menyediakan dokumen untuk diunduh.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data berhenti tidak tersedia.	Sistem menampilkan informasi bahwa data tidak tersedia.
A2	Pembuatan PDF gagal.	Sistem menampilkan pesan kesalahan.

4.1.4.14. UC-14 Mengatur Durasi Waktu Berhenti

Skenario yang menggambarkan interaksi pengguna dan sistem pada proses mengatur durasi waktu berhenti dijelaskan secara rinci melalui tabel berikut.

Tabel 4.16. Skenario UC-14

Identifikasi		
Use Case ID	UC-14	
Nama Use Case	Mengatur Durasi Waktu Berhenti	
Aktor	Pemilik Kendaraan	
Prekondisi	Kendaraan telah terdaftar.	
Postkondisi	Durasi berhenti tersimpan dan digunakan sebagai batas notifikasi.	
Tujuan	Menentukan batas waktu kendaraan berhenti sebelum sistem memberikan notifikasi.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka pengaturan kendaraan.	Sistem menampilkan pengaturan durasi berhenti.
2	Pemilik Kendaraan memasukkan durasi berhenti.	Sistem memvalidasi nilai durasi.
3	Pemilik Kendaraan menyimpan pengaturan.	Sistem menyimpan durasi berhenti.
4		Sistem menampilkan informasi bahwa pengaturan berhasil disimpan.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Nilai durasi tidak valid.	Sistem menampilkan pesan validasi.
A2	Pengguna membatalkan perubahan.	Sistem mempertahankan pengaturan sebelumnya.

4.1.4.15. UC-15 Mengaktifkan/Menonaktifkan Notifikasi Berhenti

Tabel berikut merinci prekondisi, postkondisi, serta alur normal dan alternatif dari proses mengaktifkan atau menonaktifkan notifikasi berhenti.

Tabel 4.17. Skenario UC-15

Identifikasi	
Use Case ID	UC-15
Nama Use Case	Mengaktifkan/Menonaktifkan Notifikasi Berhenti
Aktor	Pemilik Kendaraan
Prekondisi	Kendaraan telah terdaftar dan pengaturan notifikasi tersedia.
Postkondisi	Status notifikasi berhenti tersimpan sesuai pilihan pengguna.

Identifikasi		
Use Case ID	UC-15	
Tujuan	Mengatur penggunaan fitur notifikasi berhenti.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka pengaturan notifikasi.	Sistem menampilkan status notifikasi berhenti.
2	Pemilik Kendaraan mengaktifkan atau menonaktifkan notifikasi.	Sistem menerima perubahan.
3	Pemilik Kendaraan menyimpan pengaturan.	Sistem menyimpan status notifikasi.
4		Sistem menampilkan informasi bahwa pengaturan berhasil diperbarui.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Pengguna membatalkan perubahan.	Sistem mempertahankan status sebelumnya.
A2	Penyimpanan gagal.	Sistem menampilkan pesan kesalahan.

4.1.4.16. UC-16 Menerima Notifikasi Berhenti

Rincian skenario dari proses menerima notifikasi berhenti, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.18. Skenario UC-16

Identifikasi		
Use Case ID	UC-16	
Nama Use Case	Menerima Notifikasi Berhenti	
Aktor	Pemilik Kendaraan	
Prekondisi	Notifikasi berhenti aktif dan durasi berhenti telah ditentukan. Kendaraan mengirimkan data lokasi.	
Postkondisi	Pemilik Kendaraan menerima pemberitahuan mengenai kendaraan yang berhenti melebihi batas waktu.	
Tujuan	Memberikan peringatan kepada pengguna ketika kendaraan berhenti terlalu lama.	
Skenario Utama		
No	Aktor	Sistem
1		Sistem menerima data lokasi kendaraan.
2		Sistem mendeteksi kendaraan dalam kondisi berhenti.
3		Sistem menghitung durasi kendaraan berhenti.
4		Sistem membandingkan durasi berhenti dengan batas yang telah ditentukan.
5		Sistem mendeteksi durasi berhenti telah melebihi batas.
6		Sistem memperoleh informasi alamat berdasarkan lokasi kendaraan.
7		Sistem membuat dan mengirimkan notifikasi berhenti.
8	Pemilik Kendaraan menerima notifikasi.	Sistem menampilkan informasi kendaraan berhenti beserta alamatnya.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Durasi berhenti belum melewati batas.	Sistem tidak mengirimkan notifikasi.
A2	Notifikasi berhenti dinonaktifkan.	Sistem tidak mengirimkan notifikasi berhenti.
A3	Alamat tidak berhasil diperoleh.	Sistem tetap memproses notifikasi menggunakan informasi lokasi yang tersedia.

4.1.4.17. UC-17 Menerima Notifikasi Pelanggaran Geofencing

Tabel berikut menjabarkan skenario Use Case untuk proses menerima notifikasi pelanggaran geofencing secara lebih rinci.

Tabel 4.19. Skenario UC-17

Identifikasi		
Use Case ID	UC-17	
Nama Use Case	Menerima Notifikasi Pelanggaran Geofencing	
Aktor	Pemilik Kendaraan	
Prekondisi	Geofence telah ditentukan dan kendaraan mengirimkan data lokasi.	
Postkondisi	Pemilik Kendaraan menerima notifikasi ketika kendaraan terdeteksi keluar dari zona geofencing.	
Tujuan	Memberikan peringatan kepada pengguna mengenai pelanggaran zona geofencing.	
Skenario Utama		
No	Aktor	Sistem
1		Sistem menerima koordinat kendaraan.
2		Sistem mengambil konfigurasi geofence kendaraan.
3		Sistem memeriksa posisi kendaraan terhadap zona geofencing.
4		Sistem mendeteksi kendaraan berada di luar zona.
5		Sistem membuat notifikasi pelanggaran geofencing.
6		Sistem mengirimkan notifikasi melalui channel yang telah dikonfigurasi.
7	Pemilik Kendaraan menerima notifikasi.	Sistem menyertakan informasi pelanggaran dan tautan untuk melihat posisi kendaraan pada peta.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Kendaraan masih berada di dalam zona.	Sistem tidak mengirimkan notifikasi pelanggaran.
A2	Geofence dinonaktifkan.	Sistem tidak memproses pelanggaran terhadap geofence tersebut.
A3	Pengiriman salah satu channel gagal.	Sistem mencatat status pengiriman channel tersebut.

4.1.4.18. UC-18 Meninjau Notifikasi Kendaraan

Penjelasan langkah demi langkah mengenai proses meninjau notifikasi kendaraan, beserta kemungkinan skenario alternatifnya, disajikan pada tabel berikut.

Tabel 4.20. Skenario UC-18

Identifikasi		
Use Case ID	UC-18	
Nama Use Case	Meninjau Notifikasi Kendaraan	
Aktor	Pemilik Kendaraan	
Prekondisi	Pengguna telah login dan terdapat notifikasi kendaraan pada sistem.	
Postkondisi	Pengguna dapat melihat informasi notifikasi kendaraan yang diterima.	
Tujuan	Memudahkan pengguna meninjau pemberitahuan dan peringatan dari seluruh kendaraan yang dimiliki.	
Skenario Utama		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka panel notifikasi pada Home.	Sistem menampilkan daftar notifikasi kendaraan.
2	Pemilik Kendaraan memilih salah satu notifikasi.	Sistem menampilkan detail notifikasi.
3	Pemilik Kendaraan melihat informasi notifikasi.	Sistem menampilkan jenis notifikasi, kendaraan, waktu, dan informasi terkait.
4	Pemilik Kendaraan memilih tautan Lihat di Peta jika tersedia.	Sistem mengarahkan pengguna ke posisi kendaraan pada peta.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Tidak terdapat notifikasi.	Sistem menampilkan informasi bahwa belum terdapat notifikasi.
A2	Detail notifikasi tidak dapat dimuat.	Sistem menampilkan pesan kesalahan.

4.1.4.19. UC-19 Melihat Dashboard Admin

Skenario yang menggambarkan interaksi pengguna dan sistem pada proses melihat dashboard admin dijelaskan secara rinci melalui tabel berikut.

Tabel 4.21. Skenario UC-19

Identifikasi		
Use Case ID	UC-19	
Nama Use Case	Melihat Dashboard Admin	
Aktor	Administrator	
Prekondisi	Admin telah login ke sistem.	
Postkondisi	Sistem menampilkan ringkasan data pengguna dan perangkat pada dashboard admin.	
Tujuan	Memberikan admin gambaran umum kondisi sistem secara cepat.	
Skenario Utama		
Kondisi Awal: Halaman Dashboard Admin		
No	Aktor	Sistem
1	Membuka halaman dashboard admin.	Mengambil data ringkasan jumlah pengguna, jumlah perangkat, dan status online/offline perangkat.
2		Menampilkan ringkasan data beserta grafik status perangkat pada dashboard.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Belum terdapat data pengguna atau perangkat.	Menampilkan dashboard dengan ringkasan dan grafik bernilai kosong.

4.1.4.20. UC-20 Menambahkan Perangkat GPS (Admin)

Tabel berikut merinci prekondisi, postkondisi, serta alur normal dan alternatif dari proses menambahkan perangkat GPS oleh admin.

Tabel 4.22. Skenario UC-20

Identifikasi		
Use Case ID	UC-20	
Nama Use Case	Menambahkan Perangkat GPS (Admin)	
Aktor	Administrator	
Prekondisi	Admin telah login dan berada pada halaman kelola perangkat.	
Postkondisi	Perangkat GPS baru tersimpan dalam sistem beserta Device ID dan password yang dibuat otomatis.	
Tujuan	Menyediakan stok perangkat GPS yang siap diaktivasi oleh pengguna.	
Skenario Utama		
Kondisi Awal: Halaman Kelola Perangkat		
No	Aktor	Sistem
1	Memasukkan jumlah perangkat yang akan ditambahkan.	Memvalidasi jumlah perangkat (1-100).
2	Menekan tombol Tambah Perangkat.	Menghasilkan Device ID unik dan password acak untuk setiap perangkat, lalu menyimpannya ke basis data.
3		Menampilkan daftar Device ID dan password perangkat yang berhasil ditambahkan.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Jumlah perangkat tidak diisi atau berada di luar rentang 1-100.	Menampilkan pesan validasi dan meminta admin memperbaiki jumlah perangkat.

4.1.4.21. UC-21 Mengekspor Kredensial Perangkat ke PDF

Rincian skenario dari proses mengeksport kredensial perangkat ke PDF, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.23. Skenario UC-21

Identifikasi		
Use Case ID	UC-21	
Nama Use Case	Mengekspor Kredensial Perangkat ke PDF	
Aktor	Administrator	
Prekondisi	Admin baru saja menambahkan perangkat GPS baru (UC-20) sehingga tersedia data kredensial pada cache sementara.	
Postkondisi	Dokumen PDF berisi Device ID dan password perangkat berhasil diunduh.	
Tujuan	Memudahkan admin mendistribusikan kredensial perangkat kepada pihak terkait.	
Skenario Utama		
Kondisi Awal: Halaman Kelola Perangkat		
No	Aktor	Sistem
1	Menekan tombol Export PDF pada hasil penambahan perangkat.	Mengambil data kredensial perangkat dari cache sementara berdasarkan kunci sesi.
2		Menyusun dan menyediakan dokumen PDF kredensial perangkat untuk diunduh.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data cache tidak ditemukan atau sudah kedaluwarsa.	Menampilkan pesan bahwa sesi unduh kedaluwarsa dan dokumen tidak dapat dibuat.

4.1.4.22. UC-22 Menghapus Perangkat GPS (Admin)

Tabel berikut menjabarkan skenario Use Case untuk proses menghapus perangkat GPS oleh admin secara lebih rinci.

Tabel 4.24. Skenario UC-22

Identifikasi		
Use Case ID	UC-22	
Nama Use Case	Menghapus Perangkat GPS (Admin)	
Aktor	Administrator	
Prekondisi	Admin telah login dan memiliki data perangkat GPS pada daftar perangkat.	
Postkondisi	Data perangkat GPS terhapus dari sistem.	
Tujuan	Menjaga data perangkat GPS tetap akurat dan relevan.	
Skenario Utama		
Kondisi Awal: Halaman Kelola Perangkat		
No	Aktor	Sistem
1	Memilih perangkat kemudian menekan Hapus.	Menampilkan konfirmasi penghapusan.
2	Mengonfirmasi penghapusan.	Memeriksa apakah perangkat sudah terhubung dengan akun pengguna.
3		Menghapus data perangkat dan menampilkan daftar perangkat terbaru.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Perangkat sedang digunakan oleh pengguna.	Menampilkan pesan bahwa perangkat tidak dapat dihapus.

4.1.4.23. UC-23 Meninjau Data Pengguna (Admin)

Penjelasan langkah demi langkah mengenai proses meninjau data pengguna oleh admin, beserta kemungkinan skenario alternatifnya, disajikan pada tabel berikut.

Tabel 4.25. Skenario UC-23

Identifikasi		
Use Case ID	UC-23	
Nama Use Case	Meninjau Data Pengguna (Admin)	
Aktor	Administrator	
Prekondisi	Admin telah login ke sistem.	
Postkondisi	Sistem menampilkan daftar pengguna yang terdaftar dalam sistem.	
Tujuan	Memungkinkan admin memantau data pengguna yang menggunakan sistem.	
Skenario Utama		
Kondisi Awal: Halaman Kelola Pengguna		
No	Aktor	Sistem
1	Membuka halaman kelola pengguna.	Menampilkan daftar pengguna beserta perangkat dan kendaraan terkait.
2	Memasukkan nama atau email pada kolom pencarian.	Menampilkan daftar pengguna yang sesuai dengan kata kunci pencarian.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Data pengguna tidak ditemukan sesuai kata kunci pencarian.	Menampilkan informasi bahwa pengguna tidak ditemukan.

4.1.4.24. UC-24 Menghapus Data Pengguna (Admin)

Skenario yang menggambarkan interaksi pengguna dan sistem pada proses menghapus data pengguna oleh admin dijelaskan secara rinci melalui tabel berikut.

Tabel 4.26. Skenario UC-24

Identifikasi		
Use Case ID	UC-24	
Nama Use Case	Menghapus Data Pengguna (Admin)	
Aktor	Administrator	
Prekondisi	Admin telah login dan memiliki data pengguna pada daftar pengguna.	
Postkondisi	Data pengguna terhapus dari sistem beserta keterkaitan perangkat yang dimilikinya.	
Tujuan	Menjaga data pengguna sistem tetap akurat dan relevan.	
Skenario Utama		
Kondisi Awal: Halaman Kelola Pengguna		
No	Aktor	Sistem
1	Memilih pengguna kemudian menekan Hapus.	Menampilkan konfirmasi penghapusan.
2	Mengonfirmasi penghapusan.	Memeriksa status akun pengguna (admin atau bukan).
3		Menghapus data pengguna beserta keterkaitan perangkat, dan menampilkan daftar pengguna terbaru.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Akun yang dipilih merupakan akun admin.	Menampilkan pesan bahwa akun admin tidak dapat dihapus.

4.1.4.25. UC-25 Mengatur Batas Kecepatan Kendaraan

Tabel berikut merinci prekondisi, postkondisi, serta alur normal dan alternatif dari proses mengatur batas kecepatan kendaraan.

Tabel 4.27. Skenario UC-25

Identifikasi	
Use Case ID	UC-25
Nama Use Case	Mengatur Batas Kecepatan Kendaraan
Aktor	Pemilik Kendaraan
Prekondisi	Kendaraan telah terdaftar.
Postkondisi	Batas kecepatan dan status notifikasi overspeed tersimpan dan digunakan sebagai acuan deteksi overspeed.

Identifikasi		
Use Case ID	UC-25	
Tujuan	Menentukan batas kecepatan kendaraan dan mengatur status notifikasi sebelum sistem memberikan peringatan overspeed.	
Skenario Utama		
Kondisi Awal: Halaman Detail Kendaraan		
No	Aktor	Sistem
1	Pemilik Kendaraan membuka pengaturan batas kecepatan.	Sistem menampilkan pengaturan batas kecepatan dan status notifikasi overspeed.
2	Pemilik Kendaraan memasukkan nilai batas kecepatan dan mengaktifkan atau menonaktifkan notifikasi overspeed.	Sistem memvalidasi nilai batas kecepatan.
3	Pemilik Kendaraan menyimpan pengaturan.	Sistem menyimpan batas kecepatan dan status notifikasi overspeed.
4		Sistem menampilkan informasi bahwa pengaturan berhasil disimpan.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Nilai batas kecepatan tidak valid (kosong atau negatif).	Sistem menampilkan pesan validasi.
A2	Pengguna membatalkan perubahan.	Sistem mempertahankan pengaturan sebelumnya.

4.1.4.26. UC-26 Menerima Notifikasi Kecepatan Berlebih

Rincian skenario dari proses menerima notifikasi kecepatan berlebih, mencakup kondisi awal, alur utama, dan alur alternatif, dijelaskan pada tabel berikut.

Tabel 4.28. Skenario UC-26

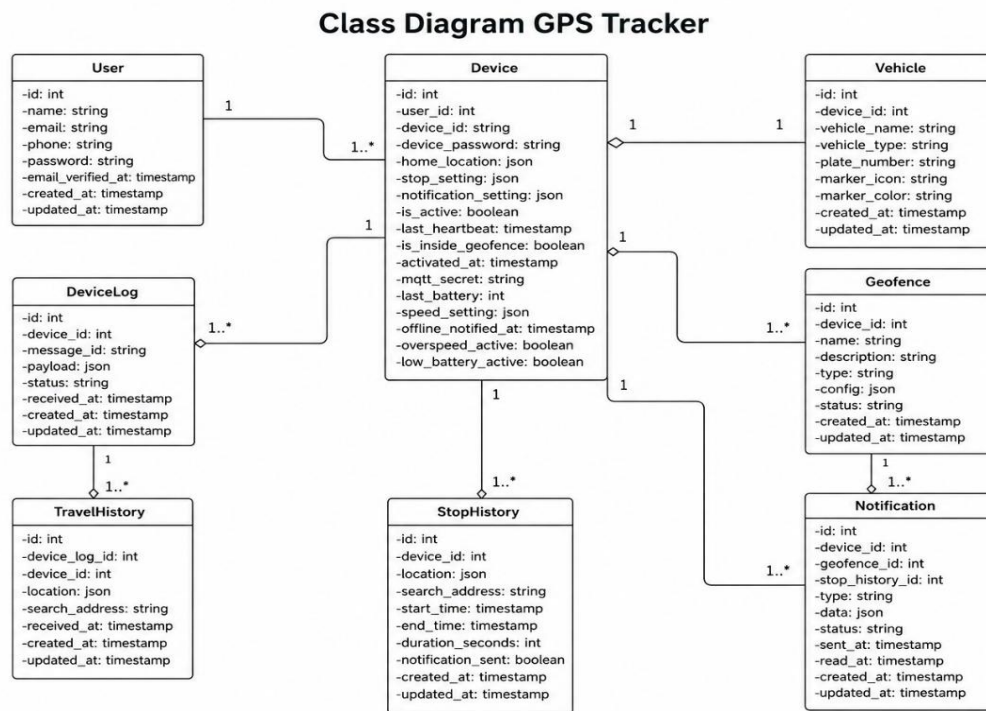
Identifikasi		
Use Case ID	UC-26	
Nama Use Case	Menerima Notifikasi Kecepatan Berlebih	
Aktor	Pemilik Kendaraan	
Prekondisi	Notifikasi overspeed aktif dan batas kecepatan telah ditentukan. Kendaraan mengirimkan data lokasi beserta kecepatan.	
Postkondisi	Pemilik Kendaraan menerima notifikasi ketika kendaraan terdeteksi melebihi batas kecepatan.	
Tujuan	Memberikan peringatan kepada pengguna ketika kendaraan melaju melebihi batas kecepatan yang ditentukan.	
Skenario Utama		
No	Aktor	Sistem
1		Sistem menerima data kecepatan kendaraan dari perangkat GPS.
2		Sistem mengambil pengaturan batas kecepatan (overspeed) kendaraan.
3		Sistem membandingkan kecepatan kendaraan dengan batas kecepatan yang dikonfigurasi.
4		Sistem mendeteksi kendaraan berada di atas batas kecepatan.
5		Sistem mendeteksi transisi dari kondisi normal menjadi melebihi batas (overspeed).
6		Sistem membuat dan mengirimkan notifikasi kecepatan berlebih melalui channel yang telah dikonfigurasi.
7	Pemilik Kendaraan menerima notifikasi.	Sistem menampilkan informasi kendaraan beserta kecepatan yang terdeteksi.
Skenario Alternatif		
Kode	Kondisi	Respons Sistem
A1	Kecepatan kendaraan tetap berada di atas batas pada titik data berikutnya.	Sistem tidak mengirimkan notifikasi berulang untuk episode overspeed yang sama.
A2	Kecepatan kendaraan kembali berada di bawah batas.	Sistem memperbarui status overspeed kendaraan menjadi nonaktif tanpa mengirimkan notifikasi tambahan.
A3	Notifikasi overspeed dinonaktifkan.	Sistem tidak mengirimkan notifikasi kecepatan berlebih.

4.2 Pemodelan Data

Pemodelan data merupakan tahap perancangan yang digunakan untuk menggambarkan struktur data serta hubungan antarobjek yang terdapat pada sistem AF GPS Tracker. Pemodelan data pada sistem ini terdiri dari Class Diagram dan Entity Relationship Diagram (ERD). Class Diagram digunakan untuk menggambarkan struktur class yang digunakan dalam sistem beserta atribut dan hubungan antarclass, sedangkan ERD digunakan untuk menggambarkan struktur basis data dan hubungan antarentitas yang digunakan dalam penyimpanan data sistem.

4.2.1 Class Diagram

Class Diagram menggambarkan struktur class yang terdapat pada sistem AF GPS Tracker beserta atribut dan hubungan antarclass. Class yang dimodelkan meliputi User, Device, Vehicle, DeviceLog, TravelHistory, StopHistory, Geofence, dan Notification. Hubungan antarclass menggambarkan keterkaitan antara pengguna, perangkat GPS, kendaraan, data lokasi, riwayat perjalanan, riwayat berhenti, geofence, serta notifikasi yang digunakan dalam sistem.

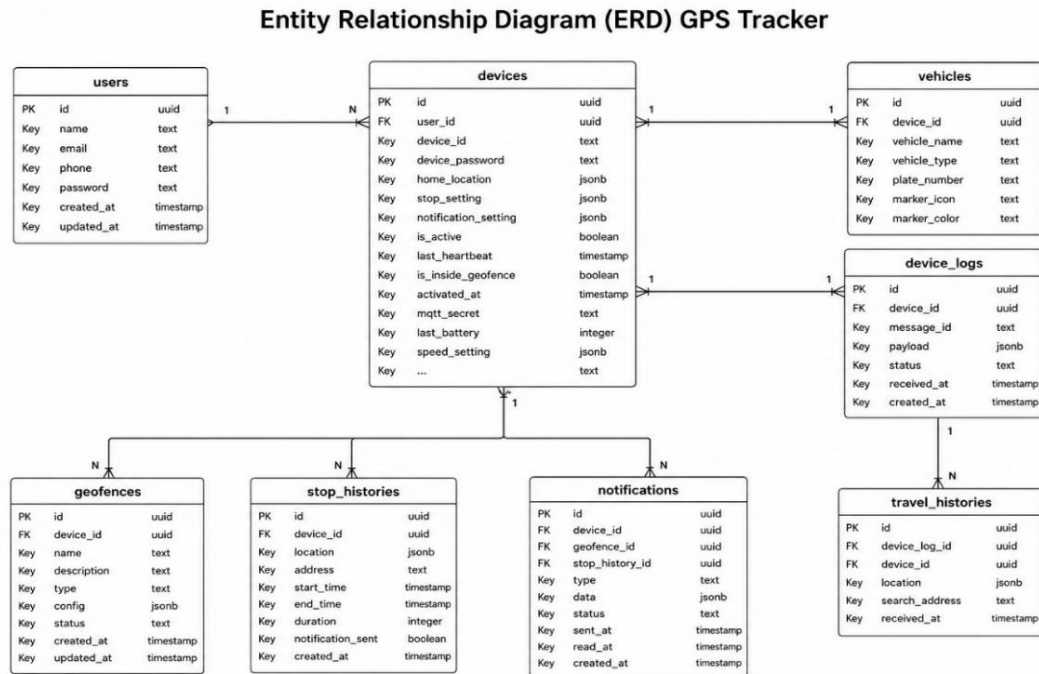


Gambar 4.29. Class Diagram

4.2.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan struktur basis data pada sistem AF GPS Tracker serta hubungan antarentitas yang digunakan dalam penyimpanan data. ERD menunjukkan entitas utama yang terdiri dari users, devices, vehicles, device_logs, travel_histories, stop_histories, geofences, dan

notifications, beserta atribut, primary key, foreign key, dan hubungan antarentitas. Pemodelan ini digunakan sebagai dasar dalam perancangan dan pengelolaan basis data sistem.



Gambar 4.30. ERD

4.3 Perancangan Perangkat Keras

Perancangan perangkat keras pada bagian ini mencakup rangkaian catu daya, rangkaian akuisisi GPS, rangkaian komunikasi GSM, serta mekanik perangkat, yang keempatnya membentuk satu kesatuan perangkat GPS tracker yang dikembangkan oleh Fakhri.

4.3.1 Perancangan Rangkaian Catu Daya

Perancangan rangkaian catu daya bertujuan untuk memenuhi kebutuhan perangkat keras HR-01, HR-02, dan HR-03 yang telah dianalisis pada BAB 3, yaitu penyediaan sumber daya listrik yang bersifat portable dan tidak bergantung pada sistem kelistrikan kendaraan. Hal ini sejalan dengan UR-13, di mana perangkat GPS tracker dituntut dapat dipasang tanpa kabel dan mudah dipindahkan antar kendaraan. Untuk memenuhi kebutuhan tersebut, rangkaian catu daya pada perangkat ini dirancang menggunakan tiga komponen utama, yaitu baterai lithium sebagai sumber energi, modul TP4056 sebagai sistem pengisian daya, dan modul HT7333 sebagai penstabil tegangan.

1. Baterai Lithium

Baterai lithium dipilih sebagai sumber daya utama menggantikan ketergantungan terhadap kabel kelistrikan kendaraan (aki motor), sehingga perangkat dapat berfungsi secara independen dan mudah dipindahkan tanpa memerlukan instalasi

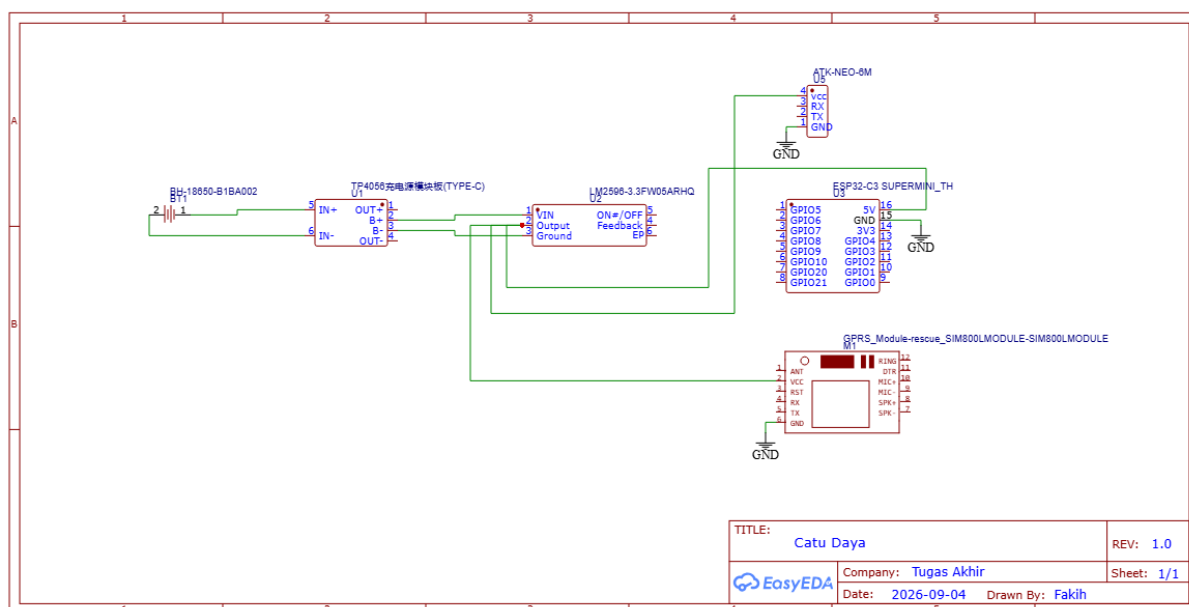
kabel tambahan pada kendaraan. Baterai yang digunakan pada perangkat ini adalah dua sel baterai lithium-ion tipe 18650 berkapasitas 3000 mAh yang dirangkai secara paralel dengan tegangan nominal 3,7 V, sehingga kapasitas total baterai menjadi 6000 mAh tanpa menaikkan tegangan keluaran, sesuai dengan modul pengisian daya satu sel (1S) yang digunakan pada perangkat ini.

2. Modul TP4056

Modul TP4056 digunakan sebagai sistem pengisian daya baterai lithium pada perangkat ini. Pada perangkat yang telah dirakit, modul yang digunakan merupakan modul HW-373 yang mengintegrasikan IC pengisian daya TP4056 dengan konektor input USB Type-C serta proteksi tambahan berupa over-charge protection, over-discharge protection, dan short-circuit protection dalam satu papan modul, sehingga proses pengisian daya baterai dapat dilakukan secara aman tanpa memerlukan rangkaian proteksi tambahan secara terpisah.

3. Modul HT7333

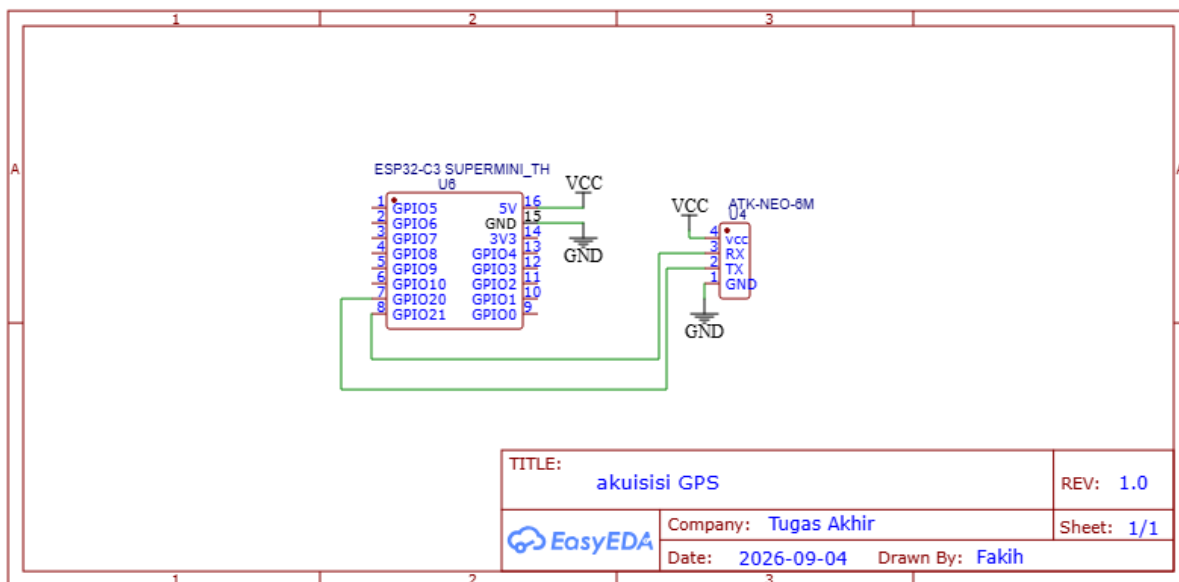
Modul HT7333 digunakan sebagai penstabil tegangan (low dropout regulator) yang berfungsi menurunkan dan menstabilkan tegangan keluaran baterai sebelum dialirkan ke ESP32 serta komponen lain pada perangkat, sehingga tegangan yang diterima tetap berada pada rentang operasional yang aman meskipun tegangan baterai berubah seiring proses pengosongan (discharge). Penggunaan modul ini bertujuan untuk menjaga kestabilan kerja mikrokontroler dan modul komunikasi selama perangkat beroperasi secara mandiri menggunakan daya baterai.



Gambar 4.31. Rangkaian Catu Daya

4.3.2 Perancangan Rangkaian Akuisisi GPS

Perancangan rangkaian akuisisi GPS bertujuan untuk memenuhi kebutuhan perangkat keras HR-04 dan HR-05 yang telah dianalisis pada BAB 3, yaitu penyediaan data koordinat lokasi kendaraan dengan tingkat akurasi 2 hingga 5 meter sesuai UR-14. Akuisisi data lokasi pada perangkat ini menggunakan modul GPS NEO-6M yang dilengkapi dengan antena GPS eksternal aktif untuk memperkuat penerimaan sinyal satelit. Modul GPS NEO-6M terhubung ke mikrokontroler ESP32 melalui komunikasi serial UART pada pin GPIO 16 (RX) dan GPIO 17 (TX) dengan kecepatan baud rate 9600 bps. Data yang diterima berupa kalimat NMEA yang selanjutnya diproses oleh firmware menggunakan pustaka TinyGPS++ untuk mengekstrak nilai latitude, longitude, kecepatan, arah (heading), serta jumlah satelit yang terdeteksi, sebelum data tersebut dikirimkan ke server.

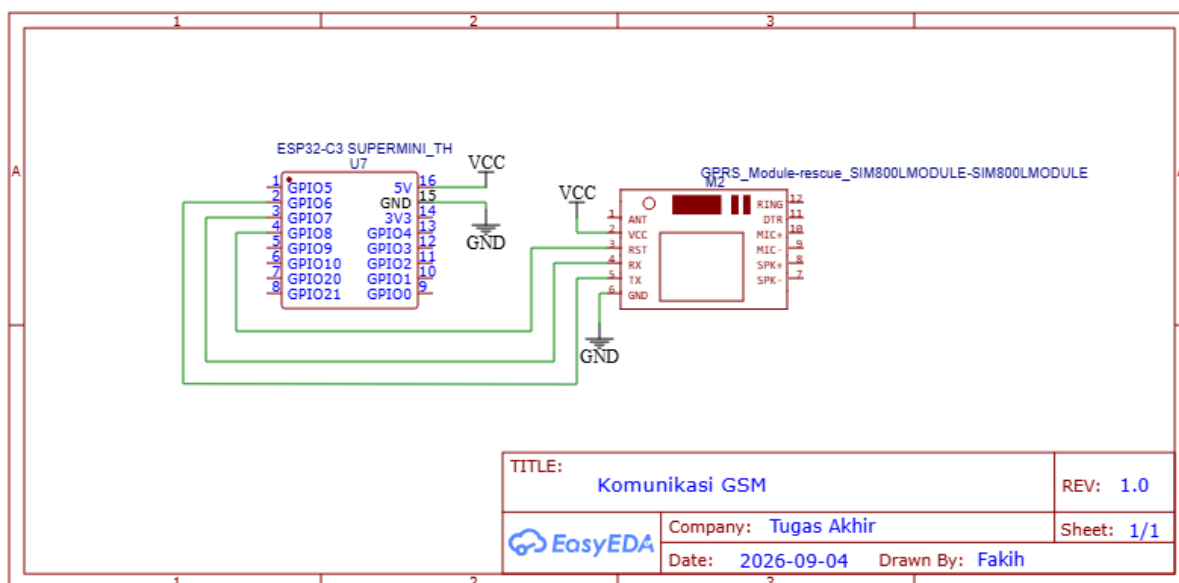


Gambar 4.32. Rangkaian Akuisisi GPS

4.3.3 Perancangan Rangkaian Komunikasi GSM

Perancangan rangkaian komunikasi GSM bertujuan untuk memenuhi kebutuhan perangkat keras HR-07 yang telah dianalisis pada BAB 3, yaitu penyediaan media pengiriman data koordinat lokasi ke server melalui jaringan seluler sesuai UR-16. Komunikasi GSM pada perangkat ini menggunakan modul SIM800L yang terhubung ke mikrokontroler ESP32 melalui komunikasi serial UART pada pin GPIO 25 (RX) dan GPIO 26 (TX) dengan kecepatan baud rate 9600 bps. Modul SIM800L memerlukan kartu SIM dengan layanan data (GPRS) aktif dan dikonfigurasi menggunakan Access Point Name (APN) sesuai operator seluler yang digunakan. Mengingat modul SIM800L memiliki karakteristik lonjakan arus (current spike) hingga sekitar 2 Ampere saat melakukan transmisi data, modul ini dicatu langsung dari baterai lithium tanpa melalui

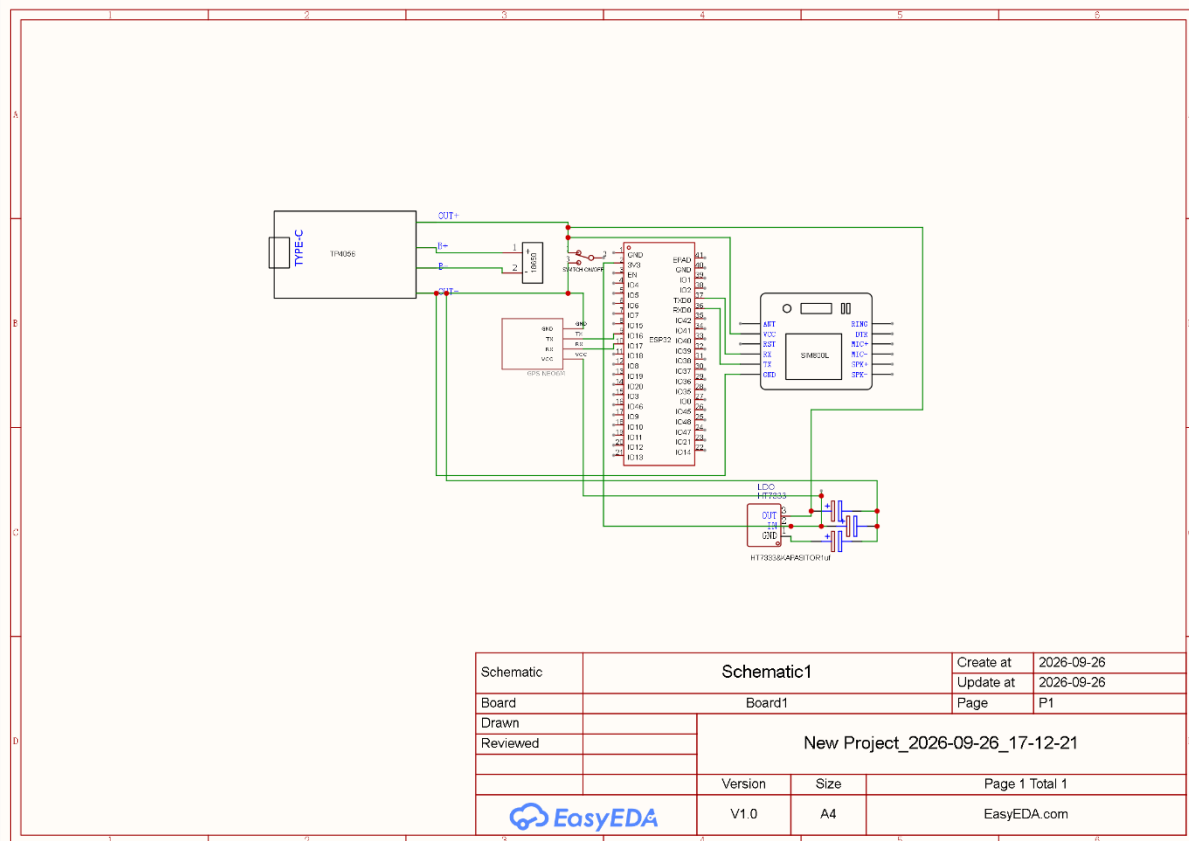
modul penstabil tegangan HT7333, sehingga mampu memenuhi kebutuhan arus sesaat tersebut tanpa menyebabkan penurunan tegangan (voltage drop) yang dapat membuat modul restart secara tidak terduga.



Gambar 4.33. Rangkaian Komunikasi GSM

4.3.4 Perancangan Mekanik

Perancangan mekanik bertujuan untuk memenuhi kebutuhan perangkat keras HR-06 yang telah dianalisis pada BAB 3, yaitu penyediaan bentuk fisik perangkat yang kompak sehingga dapat ditempatkan secara tersembunyi pada berbagai posisi di kendaraan tanpa memerlukan modifikasi permanen, sesuai dengan UR-15. Seluruh komponen elektronik perangkat, meliputi ESP32, modul SIM800L, modul GPS NEO-6M, modul pengisian daya, serta baterai lithium, dirangkai pada satu papan tembaga (copper board) yang berfungsi sebagaiudukan sekaligus bidang penghubung antar komponen secara point-to-point. Perangkat dilengkapi dengan saklar rocker (rocker switch) sebagai tombol daya utama untuk menyalakan dan mematikan perangkat secara manual, serta lapisan isolasi berupa selotip listrik (electrical tape) pada bagian sambungan solder yang terbuka untuk mencegah terjadinya hubung singkat (short-circuit) antar komponen. Rancangan mekanik yang ringkas dan tanpa casing tambahan memungkinkan perangkat memiliki dimensi fisik yang kecil sehingga mudah disembunyikan pada berbagai posisi di kendaraan bermotor.



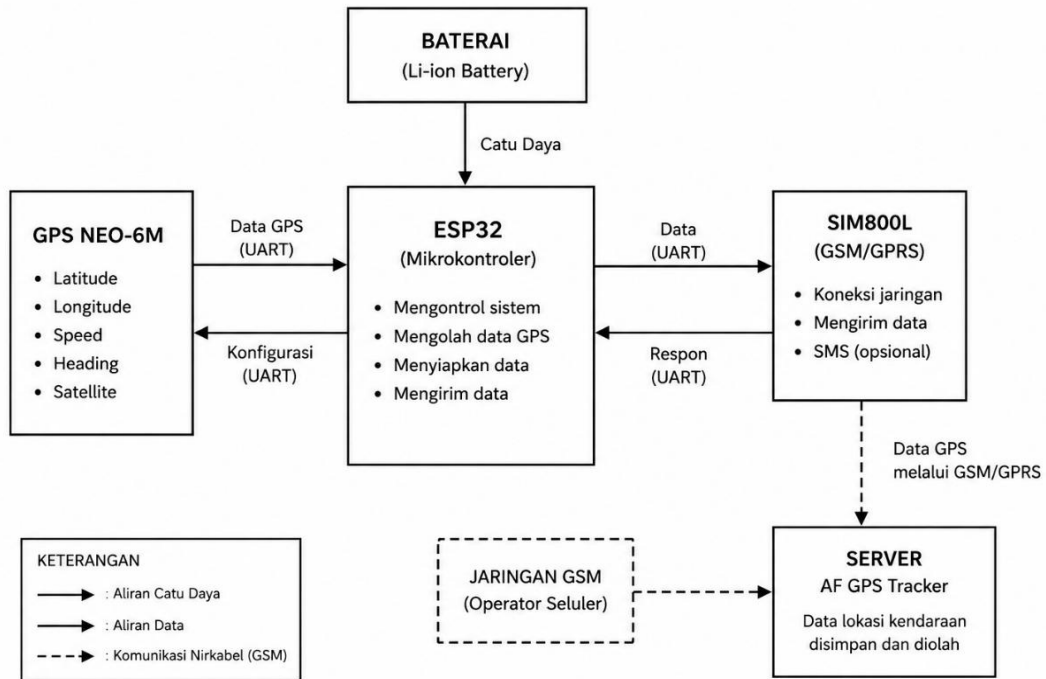
Gambar 4.34. Perancangan Mekanik

4.4 Arsitektur Sistem

Arsitektur sistem menggambarkan susunan komponen perangkat keras beserta alur komunikasi data yang menghubungkan perangkat GPS tracker dengan website pemantauan, sebagaimana diuraikan pada sub-bab berikut.

4.4.1 Blok Diagram Perangkat GPS Tracker

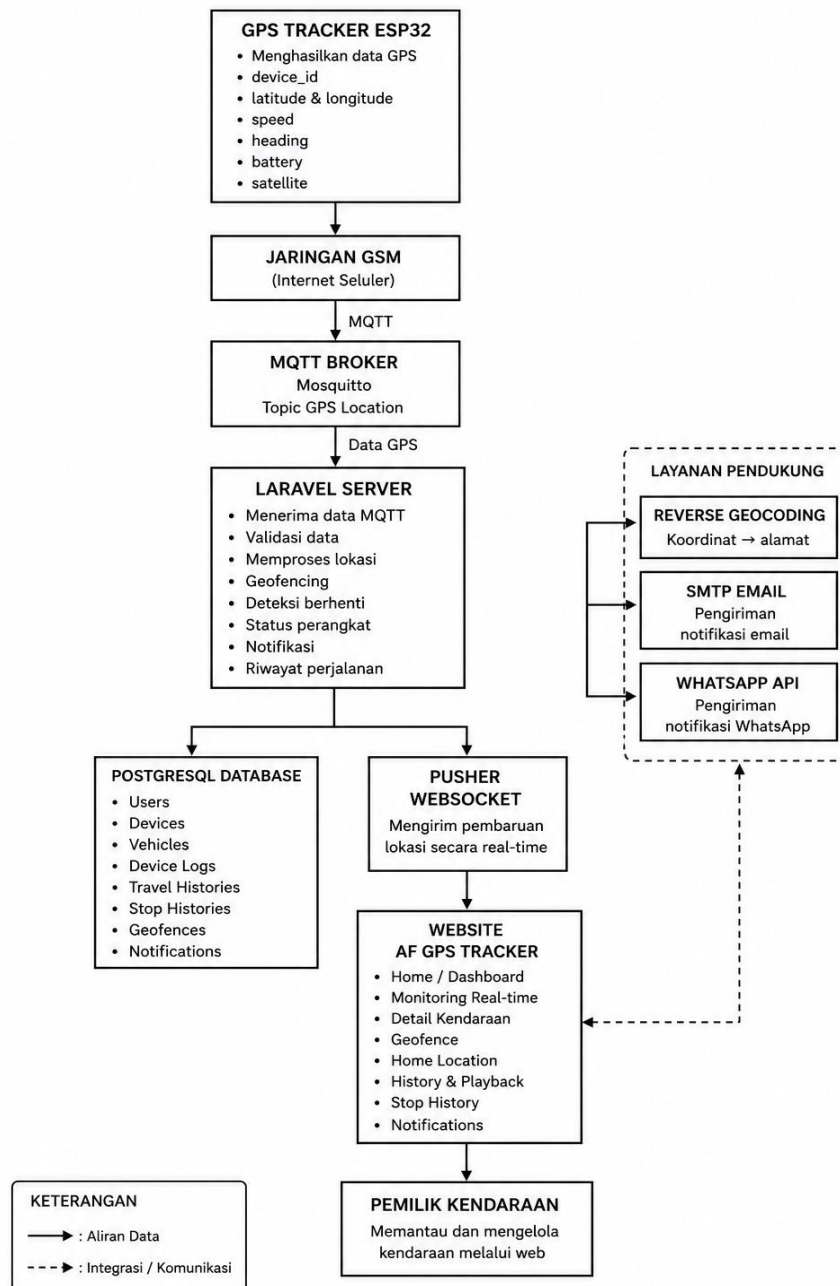
Blok diagram perangkat GPS Tracker menggambarkan hubungan antarkomponen elektronik yang membentuk satu kesatuan perangkat. Secara garis besar, perangkat terdiri atas empat blok utama, yaitu blok catu daya (baterai lithium, modul pengisian daya, dan modul penstabil tegangan), blok akuisisi lokasi (modul GPS NEO-6M beserta antena eksternal), blok komunikasi (modul SIM800L), serta blok pemrosesan (mikrokontroler ESP32) yang menjadi pusat kendali seluruh perangkat. Blok catu daya menyediakan daya listrik ke seluruh komponen, blok akuisisi lokasi mengirimkan data koordinat ke ESP32 melalui komunikasi serial, ESP32 kemudian memproses data tersebut dan meneruskannya ke blok komunikasi untuk dikirimkan ke server melalui jaringan GSM.



Gambar 4.35. Blok Diagram Perangkat GPS Tracker

4.4.2 Blok Diagram Alur Komunikasi Sistem

Blok diagram alur komunikasi sistem menggambarkan tahapan pengiriman data dari perangkat GPS Tracker hingga diterima dan diolah oleh website pemantauan. Modul GPS NEO-6M secara berkala mengirimkan data koordinat kepada ESP32 melalui komunikasi serial. ESP32 kemudian menyusun data tersebut ke dalam format JSON yang dilengkapi dengan tanda tangan digital HMAC-SHA256 untuk menjamin keaslian data, kemudian mengirimkannya ke MQTT Broker melalui modul SIM800L menggunakan jaringan GPRS. Backend Laravel pada sisi server bertindak sebagai Subscriber yang menerima data tersebut, memverifikasi tanda tangan digitalnya, kemudian memproses data melalui alur validasi, deteksi geofencing, dan penyimpanan riwayat sebelum akhirnya disiarkan secara real-time ke browser pengguna melalui Pusher.



Gambar 4.36. Blok Diagram Alur Komunikasi Sistem

4.5 Prototype

Perancangan Antar Muka menggambarkan rancangan tampilan visual setiap halaman sistem. Berikut adalah Rancangan – rancangan antarmuka dari web GPS Tracker.

4.5.1 Rancangan Antarmuka Halaman Login

Rancangan antarmuka halaman login menampilkan formulir masuk berupa kolom email dan password bagi pengguna yang telah memiliki akun, sebagaimana ditunjukkan pada gambar berikut.

Satu Platform, Kendali Penuh Kendaraan Anda

Trackio menghadirkan pelacakan lokasi real-time, riwayat perjalanan, dan notifikasi keamanan dalam satu platform yang mudah digunakan.

IMAGE / ILLUSTRATION

© 2026 Trackio. Seluruh hak cipta dilindungi.

Selamat Datang

Masuk untuk memulai monitoring kendaraan.

Email

Kata Sandi

☐ Ingat Saya [Lupa Kata Sandi?](#)

Masuk

atau

☐ Aktivasi Perangkat GPS

[Belum memiliki akun?](#)
Aktivasi perangkat GPS anda terlebih dahulu.

Gambar 4.37. Rancangan Antarmuka Halaman Login

4.5.2 Rancangan Antarmuka Halaman Aktivasi Perangkat

Halaman ini dirancang sebagai titik awal bagi pengguna baru untuk memvalidasi Device ID dan password perangkat GPS Tracker sebelum melanjutkan ke tahap pembuatan akun, seperti terlihat pada gambar berikut.

TRACKIO
Platform Pelacakan Kendaraan

Aktivasi Perangkat GPS Anda

Masukkan ID dan kata sandi perangkat untuk menghubungkan GPS tracker ke akun Trackio Anda.

IMAGE / ILLUSTRATION

© 2026 Trackio. Seluruh hak cipta dilindungi.

Aktivasi Perangkat

Masukkan Device ID dan Password Device yang terdapat pada perangkat GPS Anda.

ID Perangkat

Masukkan ID Perangkat

Kata Sandi Perangkat

Masukkan Kata Sandi Perangkat

Aktivasi

Kembali ke Login

Gambar 4.38. Rancangan Antarmuka Halaman Aktivasi Perangkat

4.5.3 Rancangan Antarmuka Halaman Lengkapi Profil

Setelah perangkat berhasil diaktivasi, rancangan antarmuka berikut menunjukkan formulir pelengkapan data akun berupa nama, email, nomor WhatsApp, dan password pengguna.

TRACKIO
Platform Pelacakan Kendaraan

Satu Langkah Lagi Menuju Aktivasi

Lengkapi data akun Anda untuk menyelesaikan proses aktivasi perangkat GPS.

Image / Illustration

© 2026 Trackio. Seluruh hak cipta dilindungi.

Lengkapi Profil

Lengkapi informasi pribadi Anda untuk melanjutkan proses aktivasi perangkat GPS.

Nama Lengkap

Masukkan nama lengkap

Email

Masukkan email

Nomor WhatsApp

08xxxxxxxxxxx

Kata Sandi

Minimal 8 karakter

Konfirmasi Kata Sandi

Masukkan ulang kata sandi

Simpan & Lanjutkan

Gambar 4.39. Rancangan Antarmuka Halaman Lengkapi Profil

4.5.4 Rancangan Antarmuka Halaman Lupa Password

Bagi pengguna yang tidak dapat mengingat kata sandi akunnya, rancangan antarmuka berikut menyediakan alur pemulihan password melalui alamat email yang telah terdaftar.

The wireframe shows a web page for 'AF GPS TRACKER' with the subtitle 'Platform Pelacakan Kendaraan'. The main heading is 'Reset Kata Sandi Anda Dengan Aman'. Below it, a subtext reads: 'Lupa kata sandi bukan masalah. Masukkan email Anda dan kami akan mengirimkan tautan untuk membuat kata sandi baru.' To the left of the form is a large grey rectangle labeled 'IMAGE / ILLUSTRATION'. The form itself is titled 'Lupa Kata Sandi' and contains the instruction: 'Masukkan email akun Anda, kami akan kirim tautan untuk membuat kata sandi baru.' It features an 'EMAIL' label, a text input field with the placeholder 'Masukkan Email', a dark blue button labeled 'Kirim Tautan Reset', and a link at the bottom that says 'Kembali ke halaman masuk'.

Gambar 4.40. Rancangan Antarmuka Halaman Lupa Password

4.5.5 Rancangan Antarmuka Halaman Informasi Kendaraan

Rancangan antarmuka berikut menampilkan daftar kendaraan yang telah terhubung dengan akun pengguna beserta informasi dasarnya.


TRACKIO
Platform Pelacakan Kendaraan

Lengkapi Informasi Kendaraan

Hubungkan kendaraan Anda dengan perangkat GPS untuk mulai memantau lokasi secara real-time.

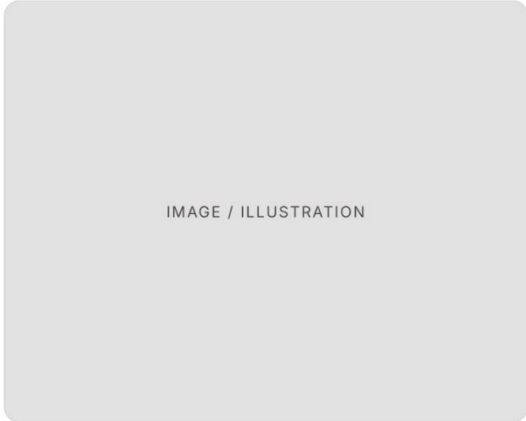


IMAGE / ILLUSTRATION

© 2026 Trackio. Seluruh hak cipta dilindungi.



Informasi Kendaraan

Lengkapi data kendaraan yang akan dihubungkan dengan perangkat GPS Tracker Anda.

Nama Kendaraan

Nomor Polisi

Jenis Kendaraan

Icon Marker

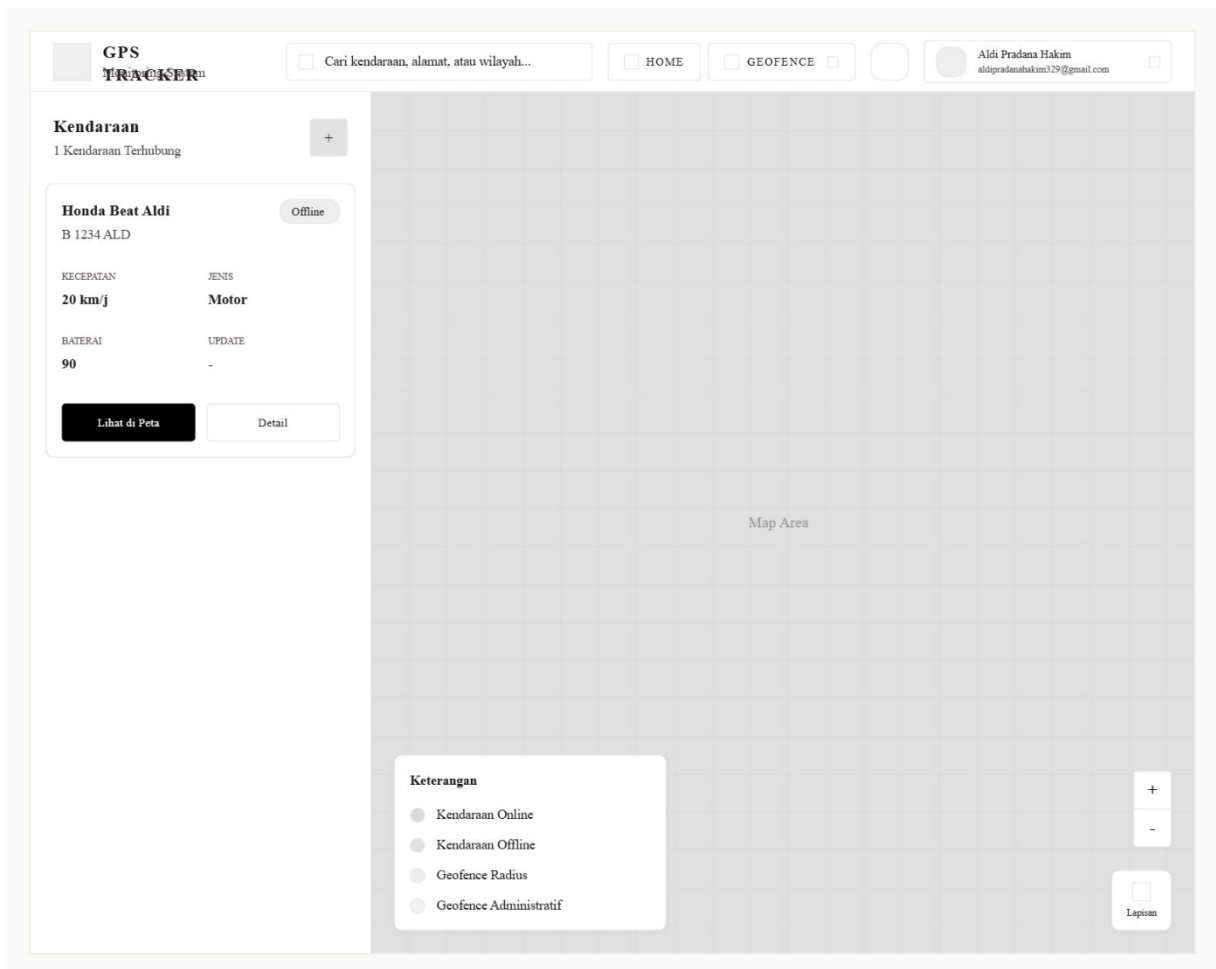
Warna Marker

Simpan & Lanjutkan →

Gambar 4.41. Rancangan Antarmuka Halaman Informasi Kendaraan

4.5.6 Rancangan Antarmuka Halaman Utama

Sebagai pusat pemantauan, rancangan antarmuka halaman utama menyajikan peta interaktif beserta posisi seluruh kendaraan milik pengguna, sebagaimana digambarkan berikut.



Gambar 4.42. Rancangan Antarmuka Halaman Utama

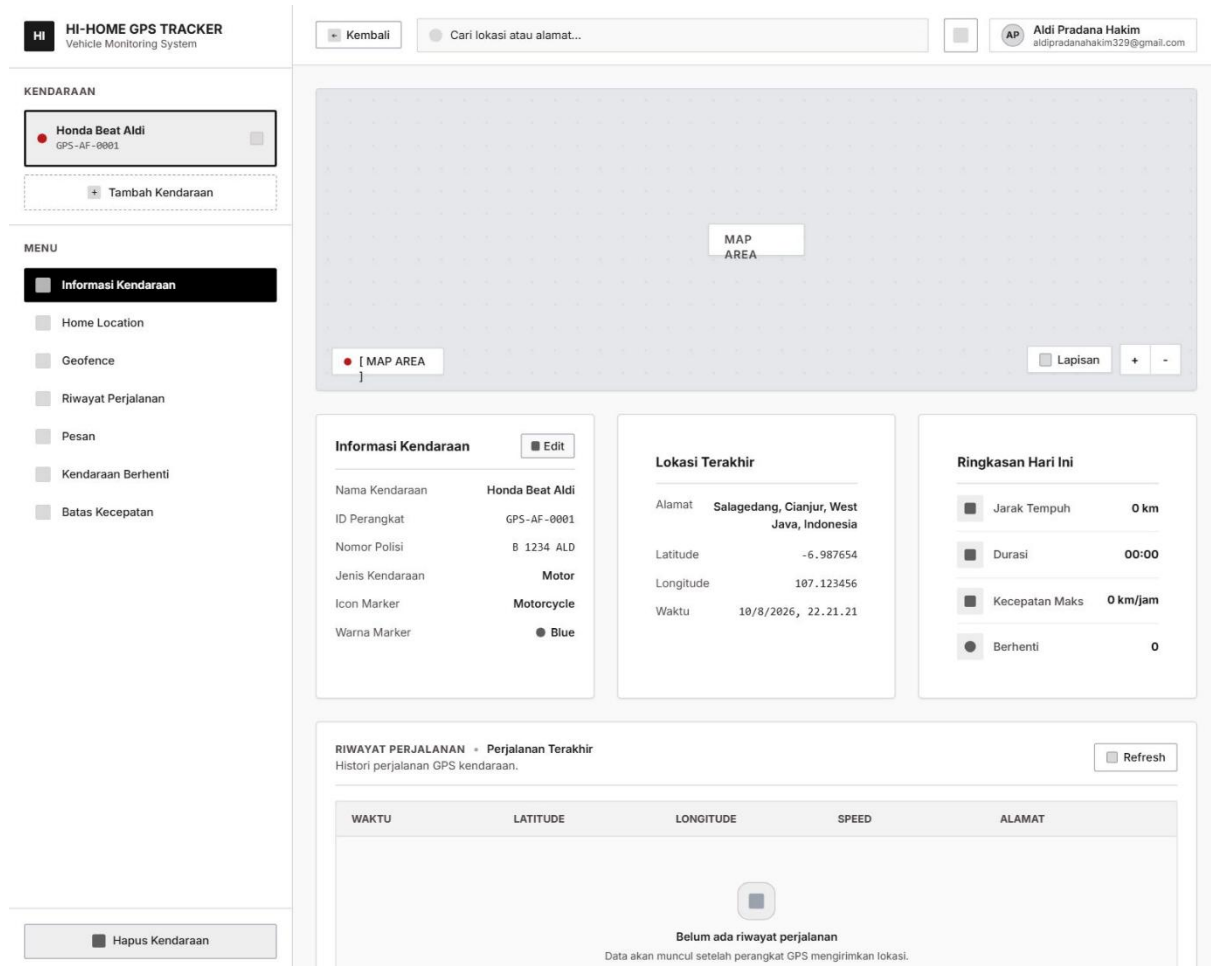
4.5.7 Rancangan Antarmuka Halaman Detail Kendaraan

Halaman Detail Kendaraan merupakan rancangan antarmuka yang ditampilkan setelah pengguna memilih salah satu kendaraan pada Halaman Utama, dan berfungsi sebagai pusat informasi serta pengaturan bagi kendaraan yang bersangkutan. Melalui halaman ini, pengguna dapat memantau kondisi kendaraan secara lebih rinci dibandingkan tampilan ringkas pada peta utama, sekaligus mengelola berbagai pengaturan yang berkaitan langsung dengan kendaraan tersebut tanpa harus berpindah ke halaman lain.

Rancangan antarmuka Halaman Detail Kendaraan terbagi menjadi beberapa bagian fungsional, yaitu Informasi Kendaraan, Home Location, Geofence, Riwayat Perjalanan, Pesan, Kendaraan Berhenti, dan Batas Kecepatan. Ketujuh bagian tersebut disusun dalam satu halaman yang sama agar pengguna dapat berpindah antarfungsi dengan mudah tanpa kehilangan konteks kendaraan yang sedang dipantau, sehingga pengalaman penggunaan aplikasi menjadi lebih efisien. Penjelasan rancangan masing-masing bagian tersebut diuraikan pada sub-bab berikut.

4.5.7.1. Informasi Kendaraan

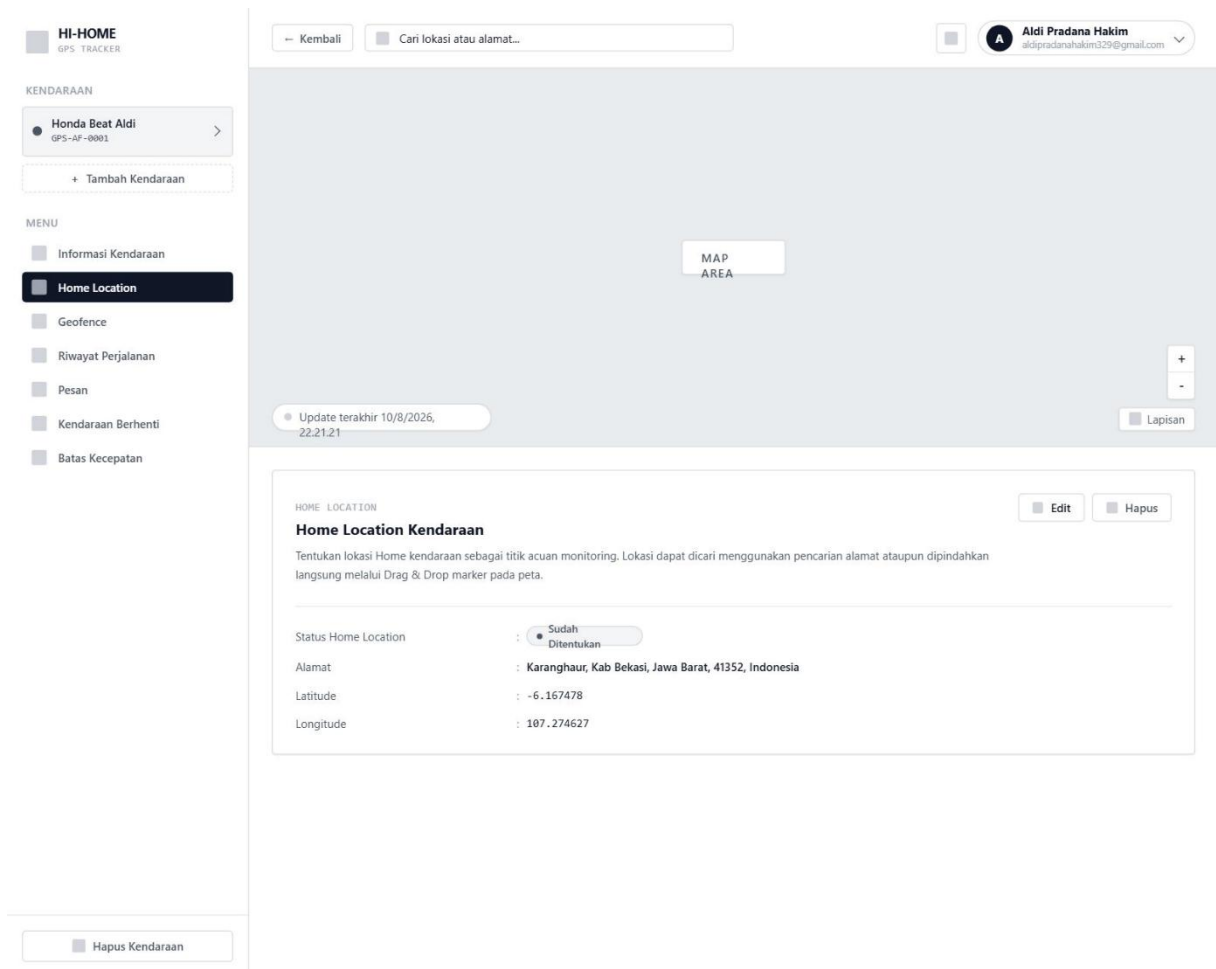
Bagian Informasi Kendaraan pada rancangan antarmuka Detail Kendaraan dirancang untuk menampilkan identitas kendaraan yang dipilih beserta status perangkat GPS yang terpasang.



Gambar 4.43. Rancangan Antarmuka Bagian Informasi Kendaraan

4.5.7.2. Home Location

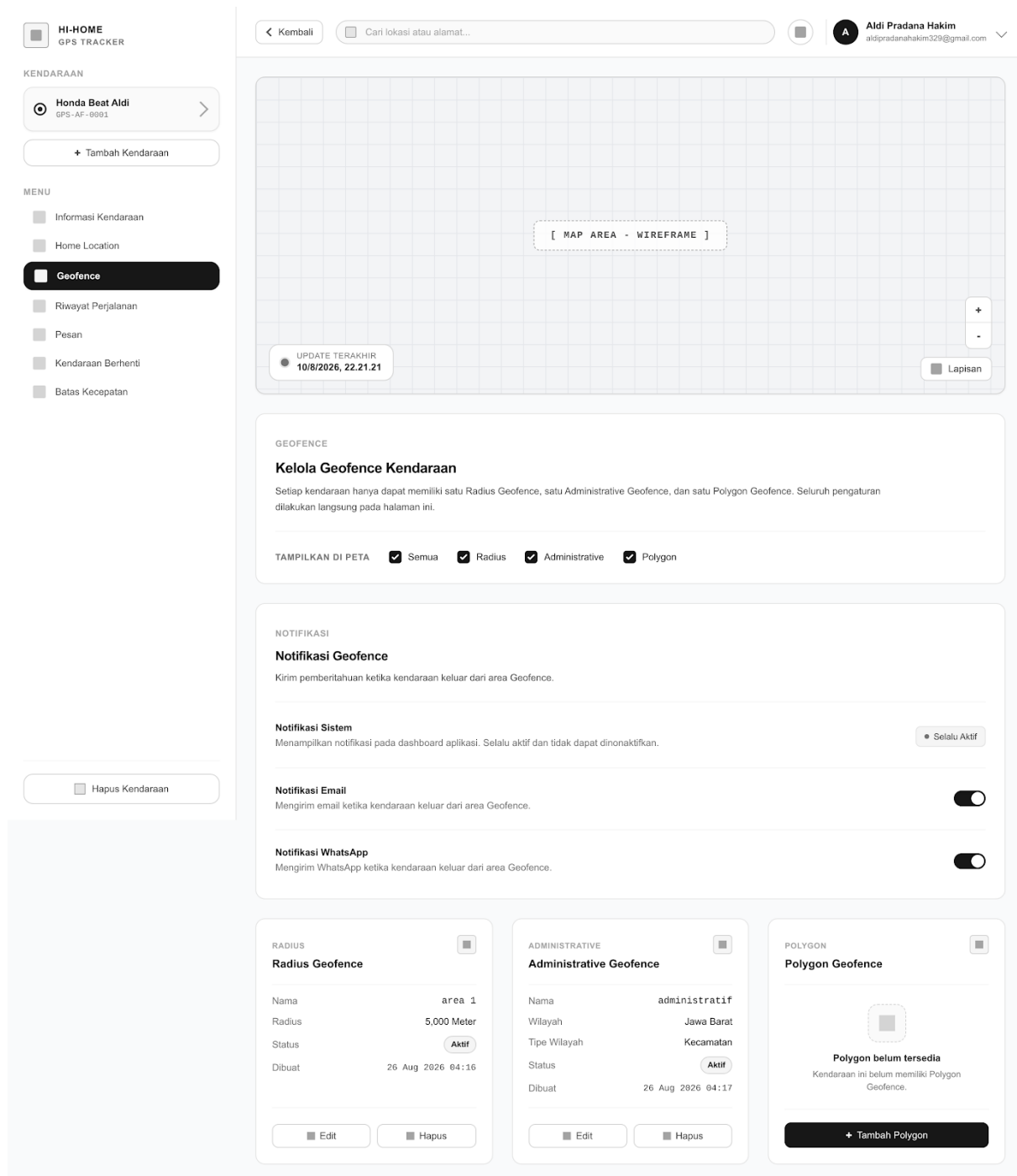
Rancangan bagian Home Location memungkinkan pengguna menentukan titik lokasi acuan atau rumah bagi kendaraan yang dipantau, seperti ditunjukkan pada gambar berikut.



Gambar 4.44. Rancangan Antarmuka Bagian Home Location

4.5.7.3. Geofence

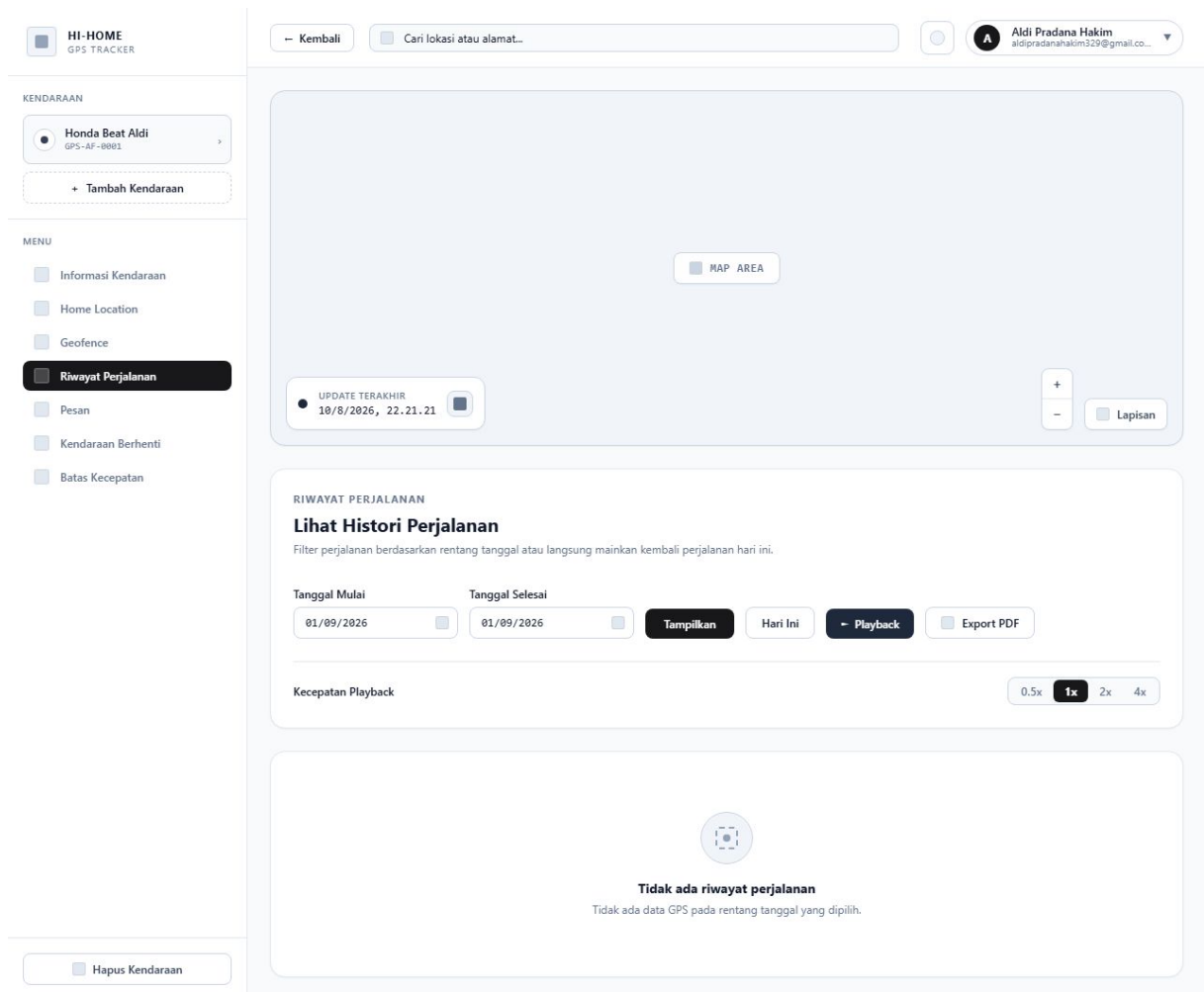
Rancangan bagian Geofence menampilkan daftar zona geofencing yang dapat dikonfigurasi pengguna untuk kendaraan tersebut.



Gambar 4.45. Rancangan Antarmuka Bagian Geofence

4.5.7.4. Riwayat Perjalanan

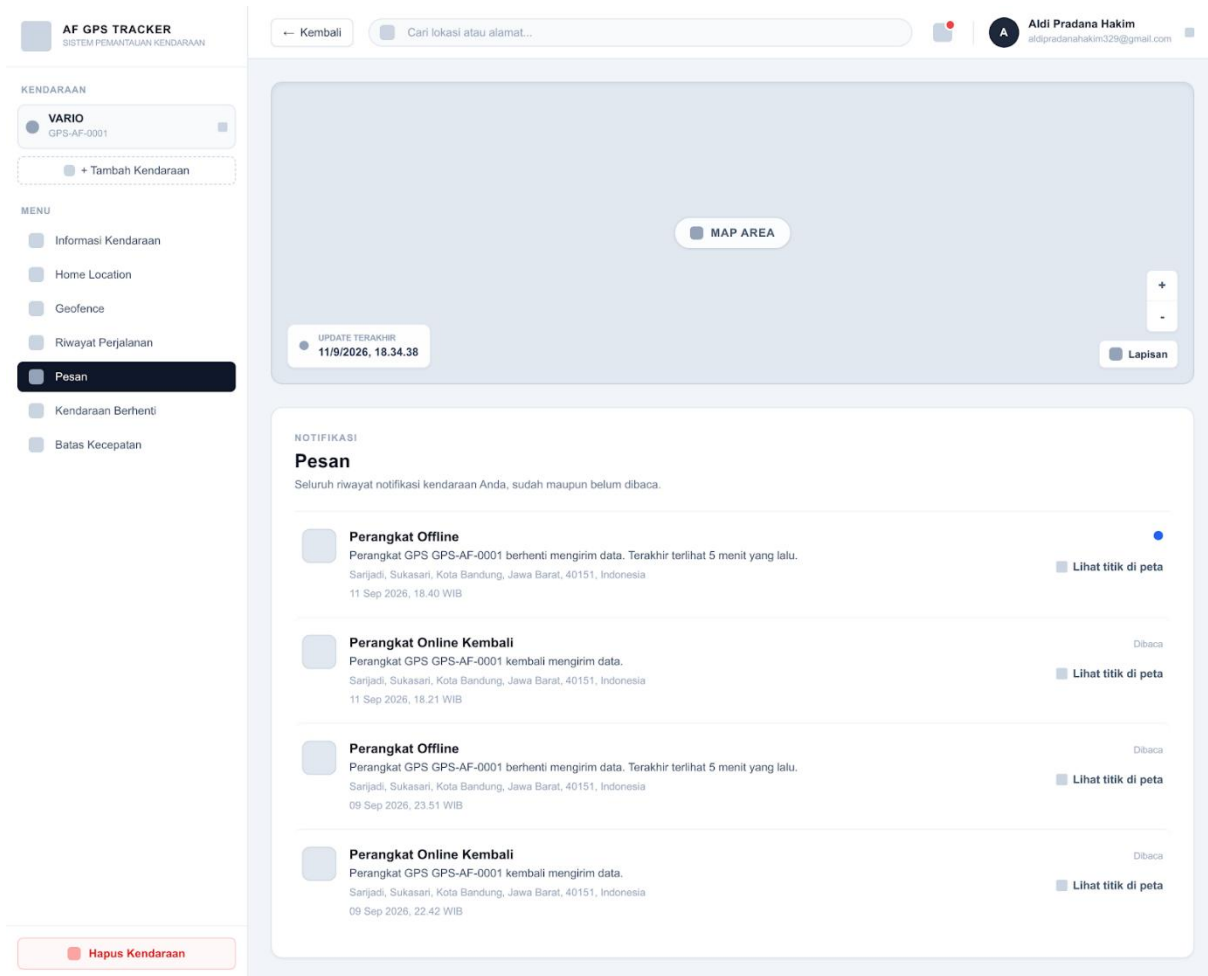
Bagian Riwayat Perjalanan pada rancangan antarmuka ini menyediakan tampilan rute pergerakan kendaraan beserta fitur pemutaran ulang (playback), sebagaimana digambarkan berikut.



Gambar 4.46. Rancangan Antarmuka Bagian Riwayat Perjalanan

4.5.7.5. Pesan

Rancangan bagian Pesan menampilkan kumpulan notifikasi yang akan diterima pengguna terkait kondisi kendaraan yang dipantau.



Gambar 4.47. Rancangan Antarmuka Bagian Pesan

4.5.7.6. Kendaraan Berhenti

Bagian Kendaraan Berhenti pada rancangan antarmuka ini dirancang untuk menampilkan riwayat kondisi kendaraan yang berhenti melebihi durasi tertentu.

HI-HOME

GPS TRACKER

KENDARAAN

Honda Beat Aldi

GPS-AF-8881

+ Tambah Kendaraan

MENU

Informasi Kendaraan

Home Location

Geofence

Riwayat Perjalanan

Pesan

Kendaraan Berhenti

Batas Kecepatan

Kembali

Cari lokasi atau alamat...

Aldi Pradana Hakim

aldipradanahakim329@gmail.com

MAP AREA

UPDATE TERAKHIR
10/8/2026, 22.21.21

+
-
Lapisan

STOP DETECTION

Pengaturan Stop Detection

Atur kapan kendaraan dianggap berhenti berdasarkan durasi kendaraan berada pada kecepatan 0 km/jam serta tentukan media notifikasi yang akan digunakan ketika Stop Detection terjadi.

Stop Detection

Durasi Kendaraan Berhenti

Notifikasi Sistem

Notifikasi WhatsApp

Aktif

5 Menit

Aktif

Aktif

Stop Detection

ON

Stop Timer

5 Menit

Email

ON

WhatsApp

ON

STOP SUMMARY

Ringkasan Stop Detection

Statistik kendaraan berhenti berdasarkan hasil Stop Detection.

Total Stop

3

Hari Ini

0

Durasi Terlama

10m

Total Waktu Stop

24m

STOP HISTORY

Riwayat Kendaraan Berhenti

Seluruh riwayat kendaraan berhenti yang berhasil dideteksi berdasarkan pengaturan Stop Detection.

Export PDF

Refresh

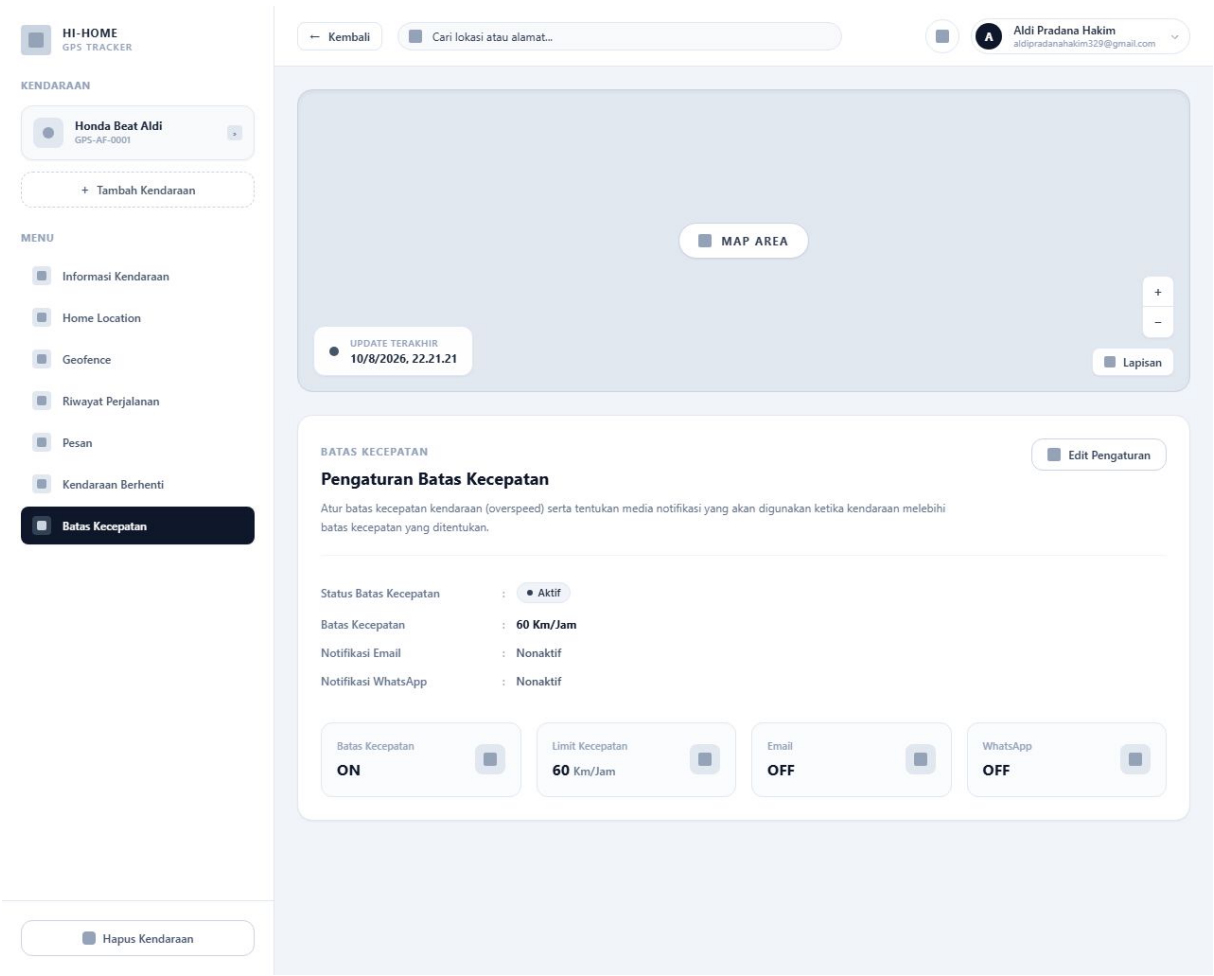
MULAI	SELESAI	DURASI	LOKASI	STATUS	AKSI
2026-08-10 22:16:29	2026-08-10 22:21:21	4m	Graha Indramas, 77, Jalan Alpda Karel Satsuit Tubun, RW 05, Silipi, Palmerah, West Jakarta, Special Capital Region of Jakarta, Jawa, 10260, Indonesia	Selesai	Lihat di Peta
2026-08-10 06:12:32	2026-08-10 06:22:32	10m	Sarijadi, Sukasari, Kota Bandung, Jawa Barat, 40152, Indonesia	Selesai	Lihat di Peta
2026-08-09 19:15:41	2026-08-09 19:25:41	10m	Sarijadi, Sukasari, Kota Bandung, Jawa Barat, 40152, Indonesia	Selesai	Lihat di Peta

Hapus Kendaraan

Gambar 4.48. Rancangan Antarmuka Bagian Kendaraan Berhenti

4.5.7.7. Batas Kecepatan

Rancangan bagian Batas Kecepatan memungkinkan pengguna menetapkan ambang batas kecepatan kendaraan beserta status notifikasi overspeed-nya.



Gambar 4.49. Rancangan Antarmuka Bagian Batas Kecepatan

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan proses penerapan hasil perancangan sistem AF GPS Tracker yang telah dibahas pada Bab IV menjadi sistem yang dapat digunakan untuk melakukan pemantauan kendaraan. Implementasi mencakup penerapan struktur proyek, antarmuka pengguna, kode program, basis data, serta proses deployment sistem. Selain implementasi perangkat lunak, pembahasan juga mencakup penerapan program pada perangkat GPS Tracker yang digunakan untuk memperoleh dan mengirimkan data lokasi kendaraan ke server. Setelah seluruh komponen sistem diimplementasikan, dilakukan pengujian untuk memastikan fungsi perangkat dan sistem dapat berjalan sesuai dengan kebutuhan yang telah ditentukan.

5.1 Implementasi

Sub-bab ini menjelaskan realisasi setiap bagian sistem AF GPS Tracker secara lebih rinci, meliputi lingkup lingkungan implementasi, struktur proyek, antarmuka, kode program pada backend maupun firmware perangkat, struktur basis data, hingga konfigurasi deployment yang digunakan pada lingkungan produksi.

5.1.1 Lingkup Implementasi

Lingkup implementasi menjelaskan lingkungan yang digunakan dalam proses penerapan sistem AF GPS Tracker, baik dari sisi perangkat keras maupun perangkat lunak. Lingkungan implementasi mencakup perangkat yang digunakan untuk menjalankan dan menguji sistem, perangkat GPS Tracker yang digunakan untuk memperoleh serta mengirimkan data lokasi kendaraan, serta perangkat lunak dan teknologi pendukung yang digunakan dalam pengembangan sistem.

Tabel 5.1. Implementasi Komponen

No	Komponen	Spesifikasi
1	Laptop Pengembang	Disesuaikan dengan spesifikasi laptop yang digunakan dalam pengembangan
2	Perangkat GPS Tracker	ESP32, GPS NEO-6M, SIM800L, baterai lithium, dan komponen pendukung
3	Smartphone	Digunakan untuk pengujian akses website dan penerimaan notifikasi
4	Jaringan	Jaringan internet dan jaringan GSM untuk komunikasi perangkat GPS Tracker dengan server

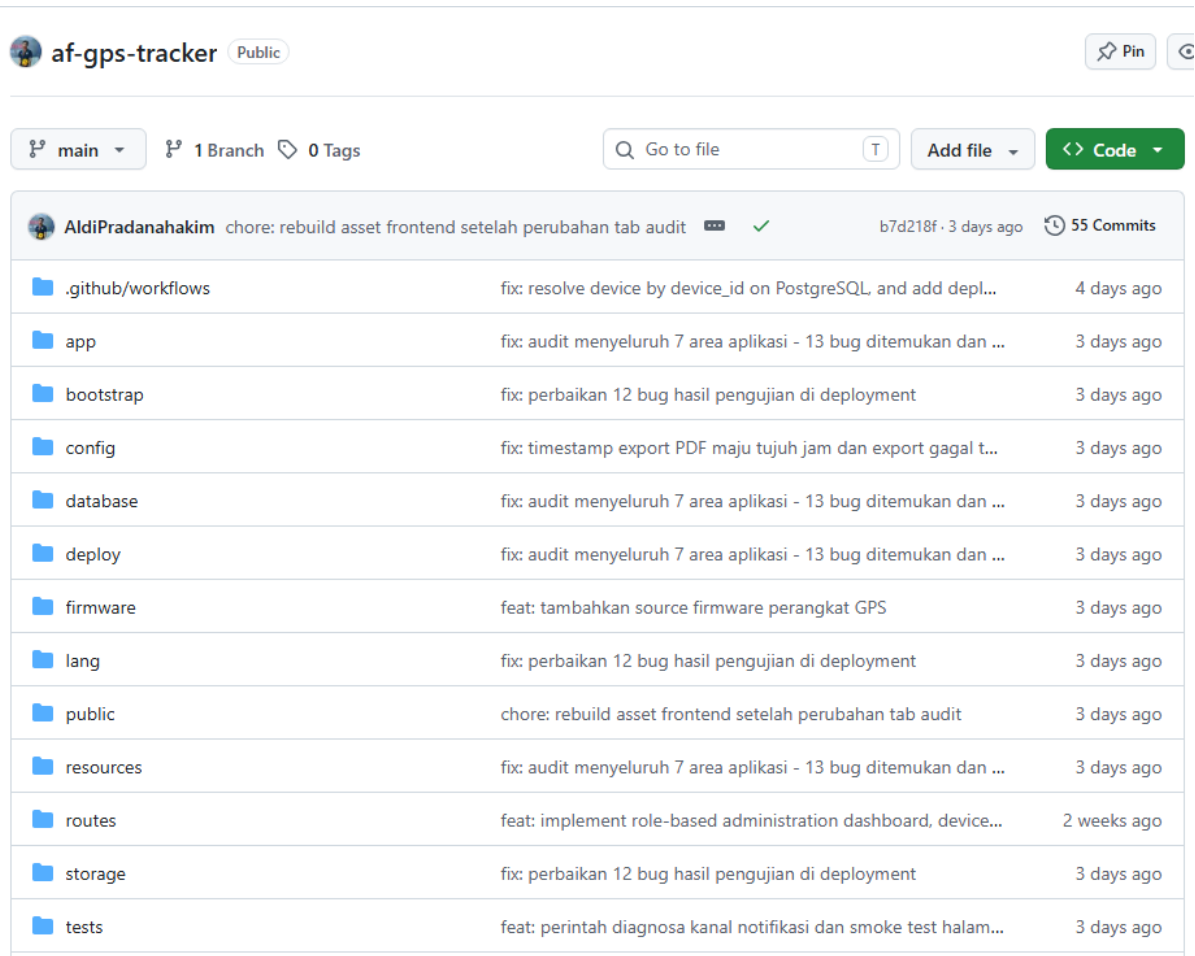
Tabel 5.2. Implementasi Perangkat Lunak/Teknologi

No	Perangkat Lunak/Teknologi	Versi/Keterangan
1	Visual Studio Code	Digunakan untuk pengembangan kode program
2	Laravel	Framework backend sistem
3	PHP	Bahasa pemrograman yang digunakan pada sisi backend

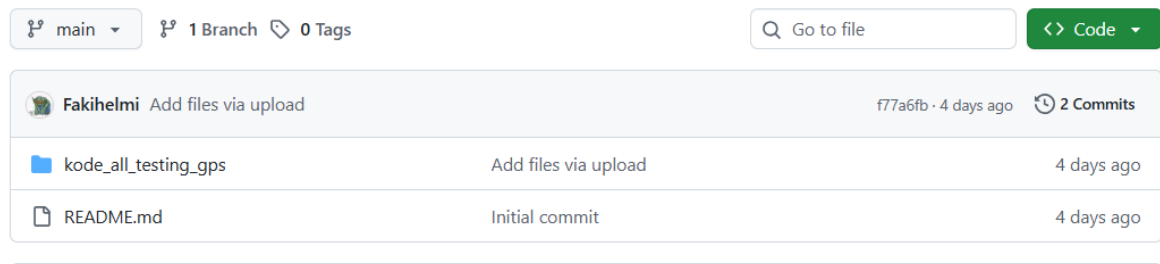
No	Perangkat Lunak/Teknologi	Versi/Keterangan
4	PostgreSQL	Sistem manajemen basis data
5	Arduino IDE	Digunakan untuk pemrograman perangkat ESP32
6	MQTT Broker (EMQX)	Digunakan sebagai MQTT Broker publik (broker.emqx.io) dalam komunikasi data GPS
7	Leaflet.js	Digunakan untuk menampilkan peta dan marker kendaraan
8	Pusher WebSocket	Digunakan untuk komunikasi data lokasi secara real-time
9	Google Chrome	Digunakan untuk menjalankan dan menguji website
10	Git dan GitHub	Digunakan untuk pengelolaan dan penyimpanan kode sumber

5.1.2 Implementasi Struktur Proyek

Proyek sistem AF GPS Tracker dikembangkan menggunakan framework Laravel dengan struktur direktori yang disusun secara modular. Struktur proyek digunakan untuk memisahkan komponen antarmuka, pengolahan data, logika sistem, akses basis data, serta proses pendukung lainnya. Pengembangan sistem menggunakan pola Controller, Service, Repository, dan Model sesuai dengan arsitektur yang telah dirancang pada tahap perancangan. Struktur proyek tersebut digunakan untuk mendukung pengembangan dan pengelolaan sistem secara terorganisir. Seluruh kode sumber dikelola menggunakan Git dan disimpan pada repository GitHub sebagai bagian dari proses version control selama pengembangan sistem dengan repository GitHubnya di alamat <https://github.com/AldiPradanahakim/af-gps-tracker.git>. Struktur proyek pada sisi perangkat lunak (website) yang dikembangkan oleh penulis ditunjukkan pada Gambar 5.1, sedangkan struktur proyek firmware pada sisi perangkat keras yang dikembangkan oleh rekan satu tim, Fakih, ditunjukkan pada Gambar 5.2. <https://github.com/AldiPradanahakim/af-gps-tracker.git> https://github.com/Fakihelmi/GPS_Tracker_io.git



Gambar 5.1. Struktur Proyek Software



Gambar 5.2. Struktur Proyek Hardware

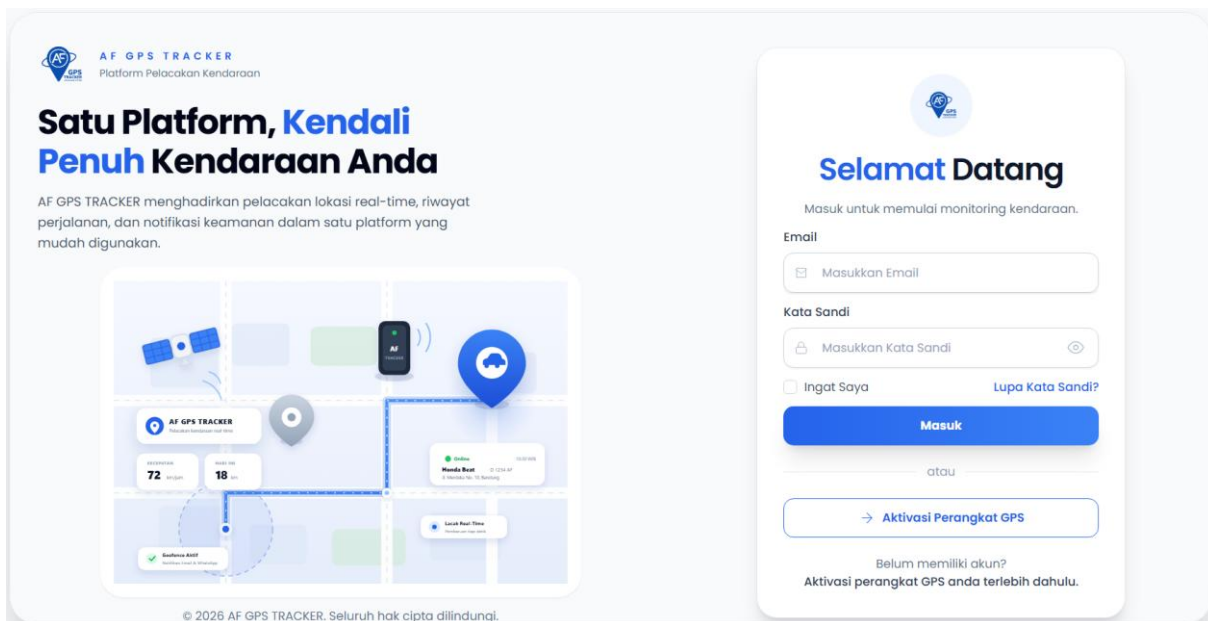
5.1.3 Implementasi Antarmuka Perangkat Lunak (Software)

Implementasi antarmuka menampilkan hasil penerapan rancangan antarmuka pengguna yang telah dibahas pada Bab IV ke dalam halaman website yang dapat diakses oleh pengguna. Antarmuka utama yang diimplementasikan meliputi halaman aktivasi perangkat dan pembuatan akun, halaman login, halaman Home sebagai peta pemantauan real-time, halaman Detail Kendaraan yang menampilkan posisi, kecepatan, riwayat perjalanan, dan playback perjalanan kendaraan, halaman pengelolaan zona geofencing, halaman pengaturan notifikasi, serta panel admin untuk pengelolaan perangkat dan

pengguna. Tangkapan layar setiap halaman disertakan pada bagian ini sebagai bukti visual hasil implementasi.

5.1.3.1. Implementasi Antarmuka Halaman Login

Halaman login menjadi pintu masuk utama bagi pengguna yang telah memiliki akun untuk dapat mengakses sistem AF GPS Tracker, dengan memasukkan alamat email dan kata sandi yang telah terdaftar sebelumnya.



Gambar 5.3. Implementasi Antarmuka Halaman Login

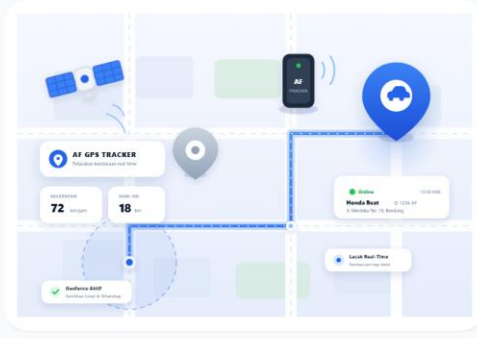
5.1.3.2. Implementasi Antarmuka Halaman Aktivasi Perangkat

Sebelum dapat menggunakan sistem, pengguna baru terlebih dahulu diarahkan ke halaman aktivasi perangkat untuk mendaftarkan perangkat GPS Tracker yang telah diperoleh sekaligus membuat akun pengguna melalui kode aktivasi yang tertera pada unit perangkat.


AF GPS TRACKER
Platform Pelacakan Kendaraan

Aktivasi Perangkat GPS Anda

Masukkan ID dan kata sandi perangkat untuk menghubungkan GPS tracker ke akun AF GPS TRACKER Anda.



© 2026 AF GPS TRACKER. Seluruh hak cipta dilindungi.



Aktivasi Perangkat

Masukkan Device ID dan Password Device yang terdapat pada perangkat GPS Anda.

ID Perangkat

Kata Sandi Perangkat

Aktivasi

[Kembali ke Login](#)

Gambar 5.4. Implementasi Antarmuka Halaman Aktivasi Perangkat

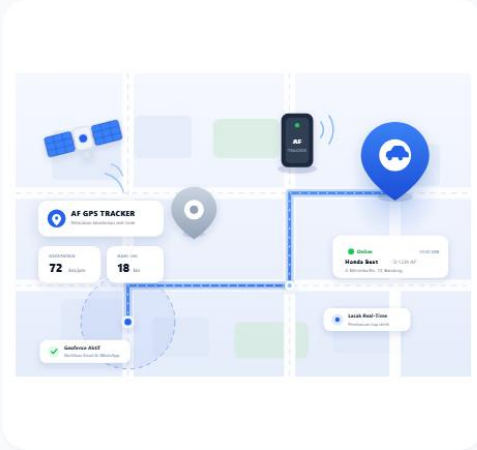
5.1.3.3. Implementasi Antarmuka Lengkapi Profil

Setelah proses aktivasi berhasil dilakukan, pengguna diarahkan menuju halaman ini untuk melengkapi data profil, seperti nama lengkap dan nomor WhatsApp yang selanjutnya digunakan sebagai tujuan pengiriman notifikasi dari sistem.


AF GPS TRACKER
Platform Pelacakan Kendaraan

Satu Langkah Lagi Menuju Aktivasi

Lengkapi data akun Anda untuk menyelesaikan proses aktivasi perangkat GPS.



© 2026 AF GPS TRACKER. Seluruh hak cipta dilindungi.



Lengkapi Profil

Lengkapi informasi pribadi Anda untuk melanjutkan proses aktivasi perangkat GPS.

Nama Lengkap

Email

Nomor WhatsApp

Kata Sandi

Kata sandi harus memuat:

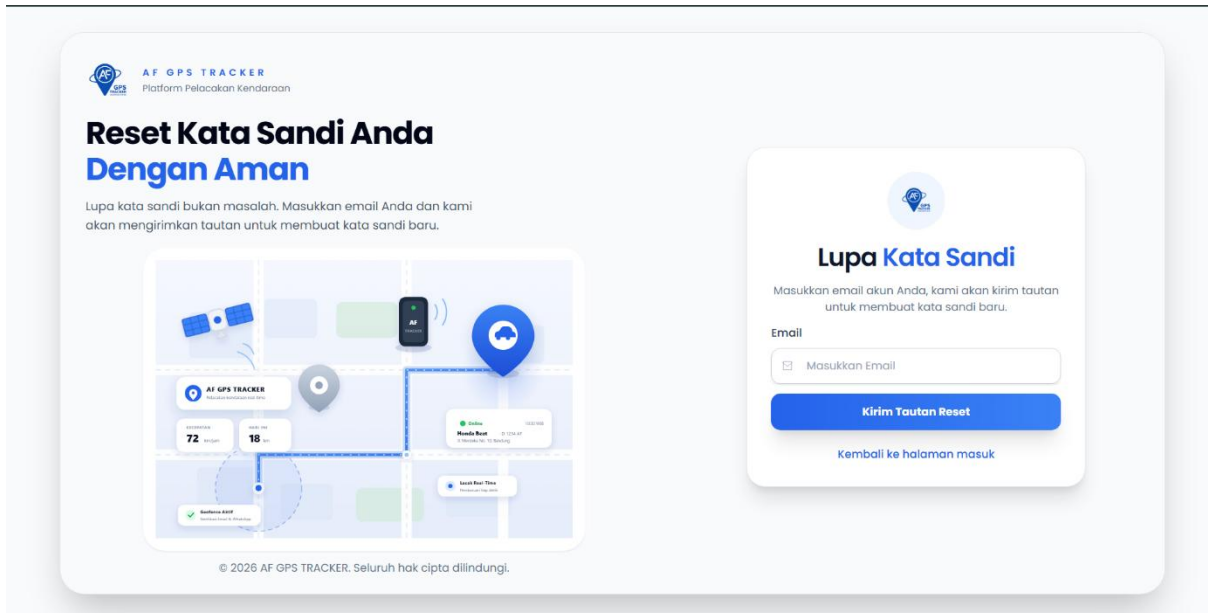
- Minimal 8 karakter
- Huruf besar (A-Z)
- Huruf kecil (a-z)
- Angka (0-9)
- Karakter khusus (contoh: ! @ # \$ %)

Contoh: Gps#Tracker2026

Gambar 5.5. Implementasi Antarmuka Halaman Lengkapi Profil

5.1.3.4. Implementasi Antarmuka Lupa Password

Fitur lupa password disediakan untuk mengakomodasi pengguna yang tidak dapat mengingat kata sandi akunnya, dengan proses reset yang dikirimkan melalui alamat email yang telah terdaftar.



Gambar 5.6. Implementasi Antarmuka Halaman Lupa Password

5.1.3.5. Implementasi Antarmuka Halaman Informasi Kendaraan

Pada halaman ini, pengguna dapat melihat daftar kendaraan yang telah terhubung dengan perangkat GPS Tracker miliknya beserta informasi dasar seperti nama dan jenis kendaraan.

AF GPS TRACKER
Platform Pelacakan Kendaraan

Lengkapi Informasi Kendaraan

Hubungkan kendaraan Anda dengan perangkat GPS untuk mulai memantau lokasi secara real-time.

Informasi Kendaraan

Lengkapi data kendaraan yang akan dihubungkan dengan perangkat AF GPS Tracker Anda.

Nama Kendaraan

Contoh : Motor Harian

Nomor Polisi

B - 1234 - ABC

Contoh: B 1234 ABC (Huruf-Angka-Huruf)

Jenis Kendaraan

Pilih Jenis Kendaraan

Icon Marker

Warna Marker

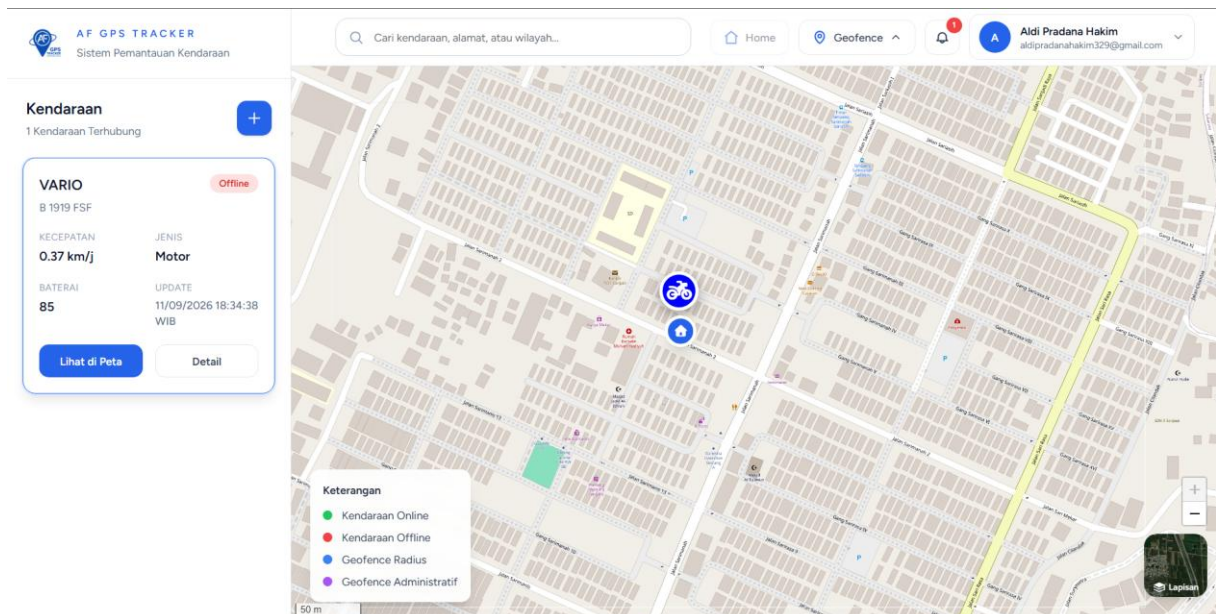
Hijau

© 2026 AF GPS TRACKER. Seluruh hak cipta dilindungi.

Gambar 5.7. Implementasi Antarmuka Halaman Informasi Kendaraan

5.1.3.6. Implementasi Antarmuka Halaman Utama

Sebagai pusat pemantauan, halaman utama menyajikan peta interaktif yang menampilkan posisi seluruh kendaraan pengguna secara real-time beserta ringkasan status masing-masing perangkat.



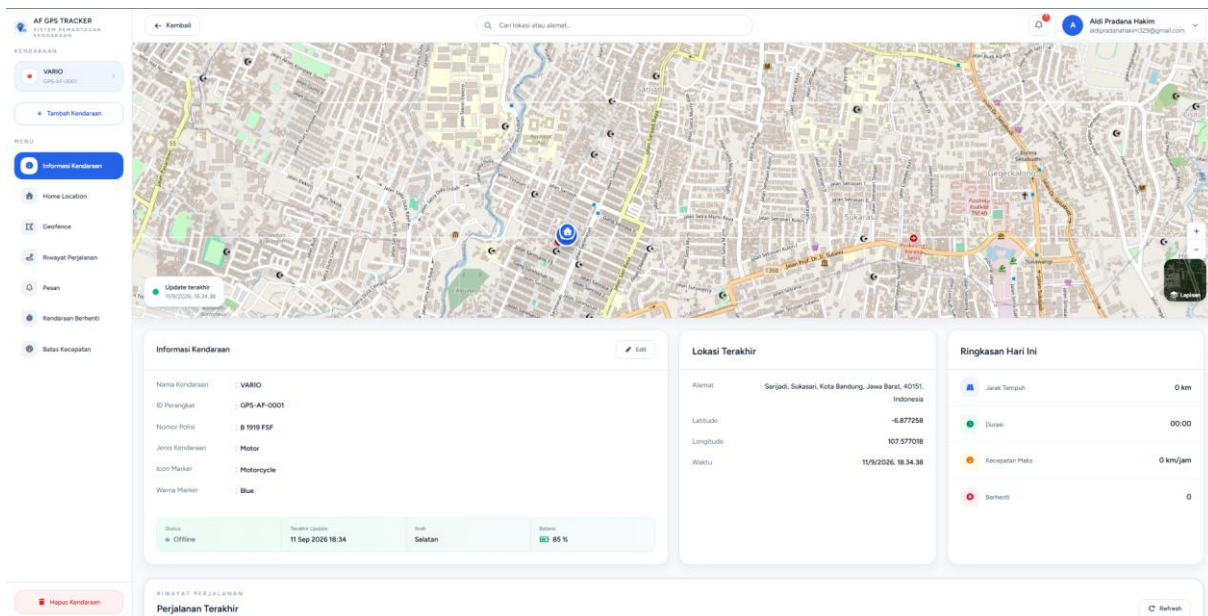
Gambar 5.8. Implementasi Antarmuka Halaman Utama

5.1.3.7. Implementasi Antarmuka Halaman Detail Kendaraan

Halaman Detail Kendaraan menyediakan informasi yang lebih mendalam mengenai satu kendaraan tertentu, yang terbagi menjadi beberapa bagian sebagaimana dijabarkan pada uraian berikut.

a. Informasi Kendaraan

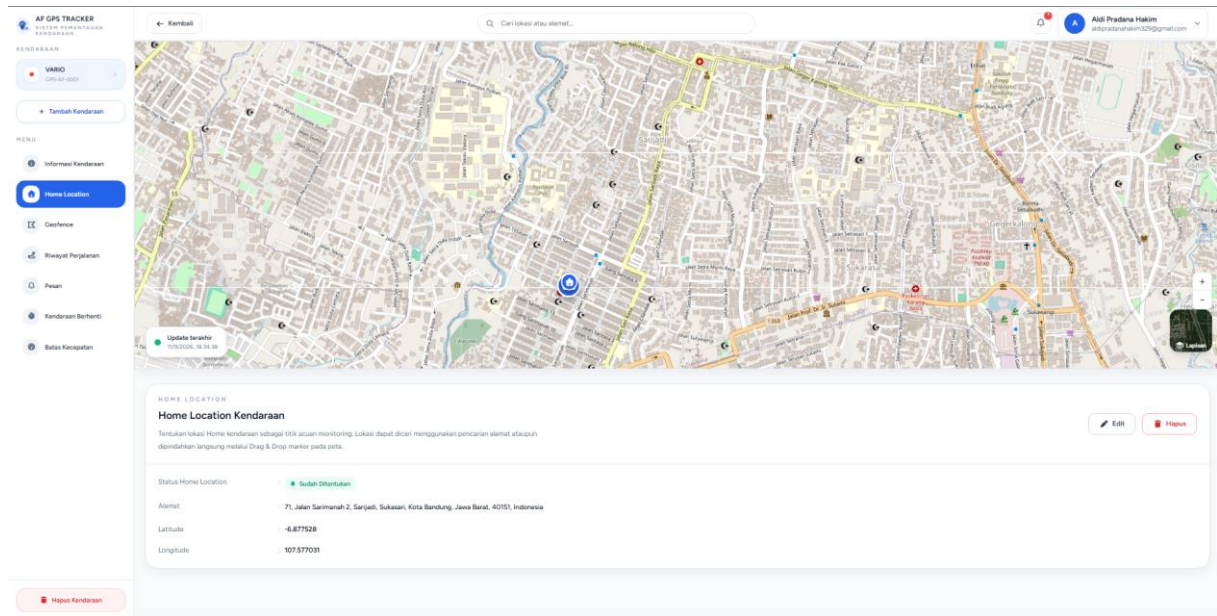
Bagian Informasi Kendaraan menampilkan data identitas kendaraan yang dipilih, mencakup nama, jenis kendaraan, serta status perangkat GPS yang terpasang di dalamnya.



Gambar 5.9. Implementasi Antarmuka Halaman Detail Informasi Kendaraan

b. Home Location

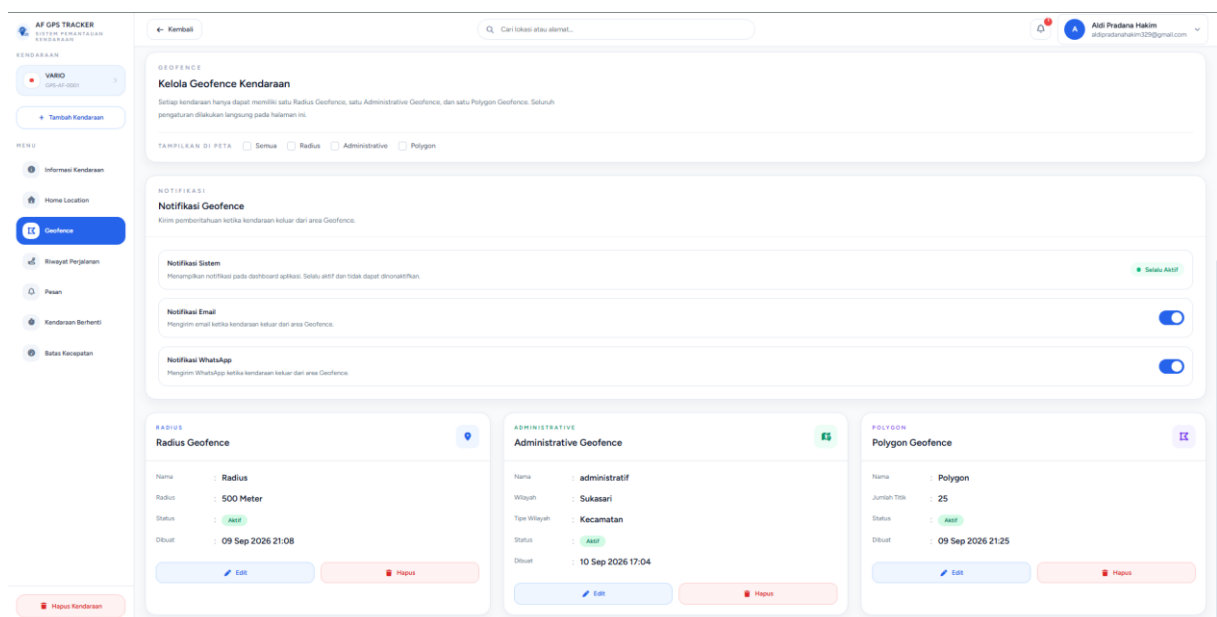
Pada bagian Home Location, pengguna dapat menentukan titik lokasi basis atau rumah kendaraan yang selanjutnya digunakan sebagai acuan pada beberapa fitur pemantauan lain dalam sistem.



Gambar 5.10. Implementasi Antarmuka Halaman Detail Home Location

c. Geofence

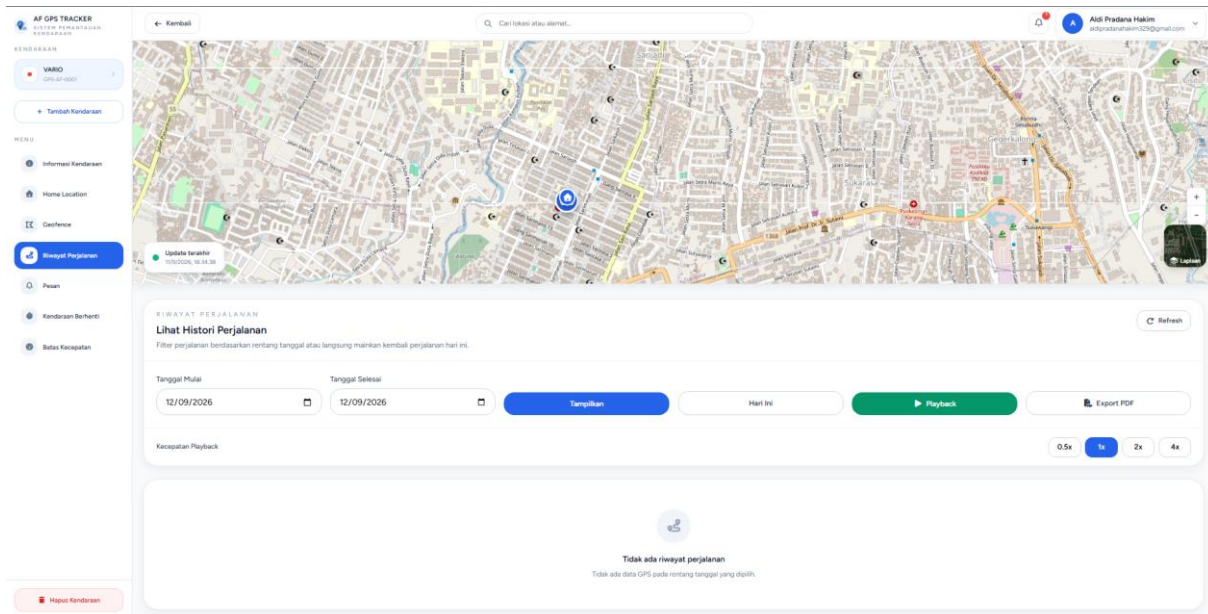
Bagian Geofence menampilkan daftar zona geofencing yang telah dikonfigurasi untuk kendaraan tersebut, baik berupa zona wilayah administratif, radius, maupun area bebas yang digambar langsung oleh pengguna.



Gambar 5.11. Implementasi Antarmuka Halaman Detail Geofence

d. Riwayat Perjalanan

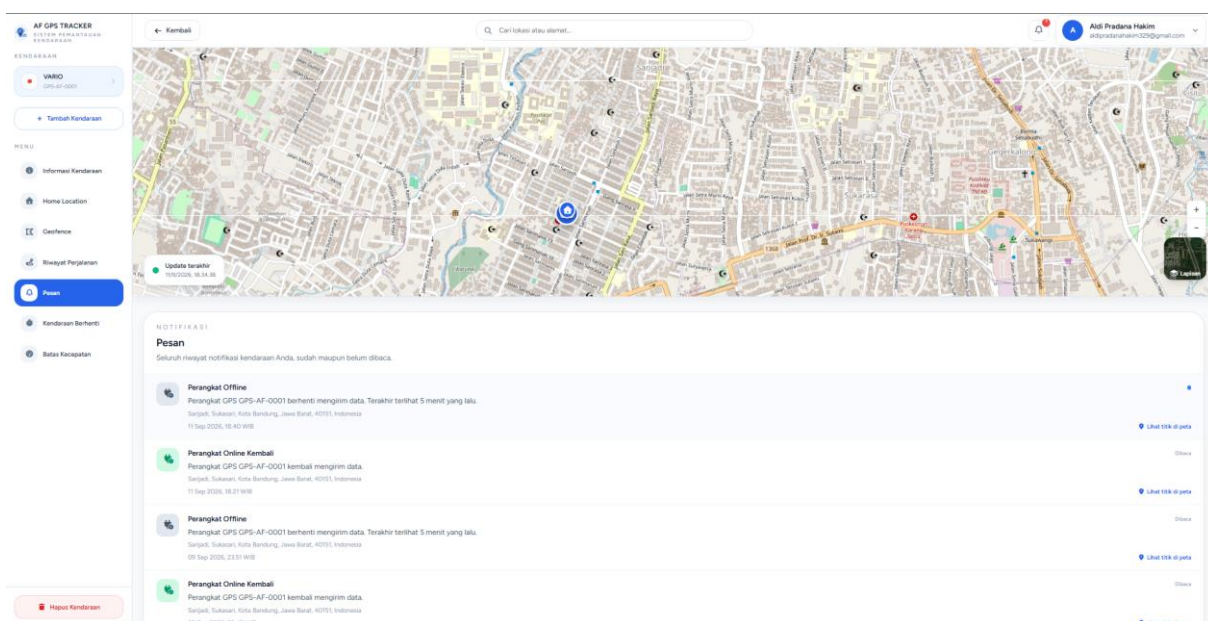
Bagian Riwayat Perjalanan menyajikan rekam jejak pergerakan kendaraan pada rentang waktu tertentu, lengkap dengan fitur pemutaran ulang (playback) untuk menampilkan kembali lintasan yang telah ditempuh.



Gambar 5.12. Implementasi Antarmuka Halaman Detail Riwayat Perjalanan

e. Pesan

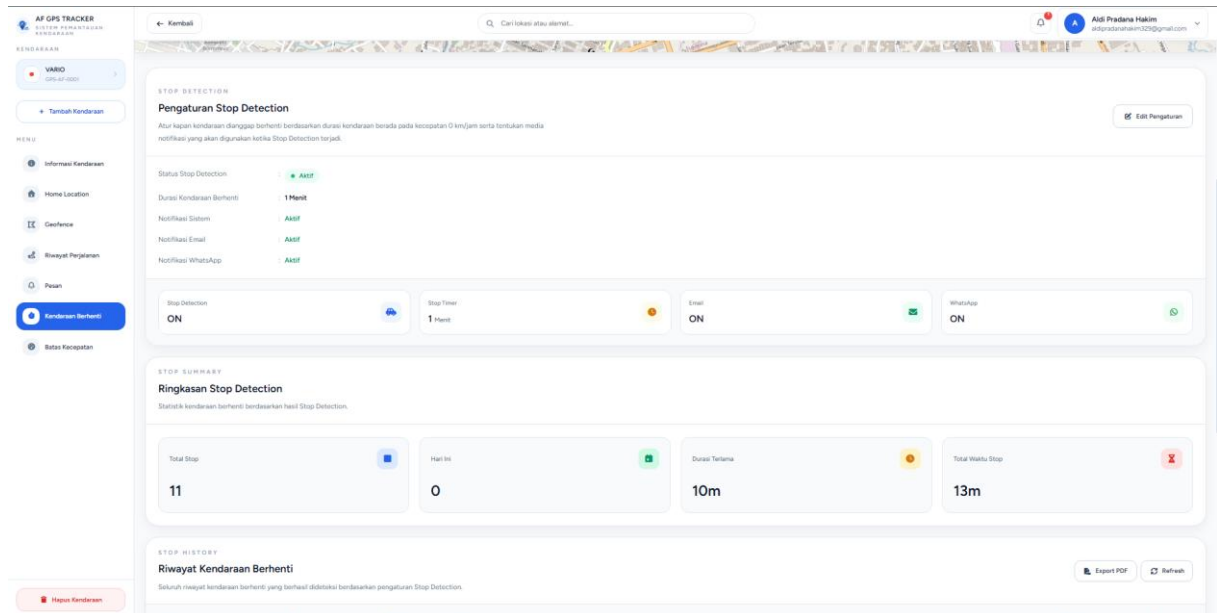
Pada bagian Pesan, pengguna dapat melihat kumpulan notifikasi yang pernah dikirimkan sistem terkait kendaraan yang bersangkutan, seperti peringatan pelanggaran geofencing, overspeed, maupun kendaraan berhenti.



Gambar 5.13. Implementasi Antarmuka Halaman Detail Pesan

f. Kendaraan Berhenti

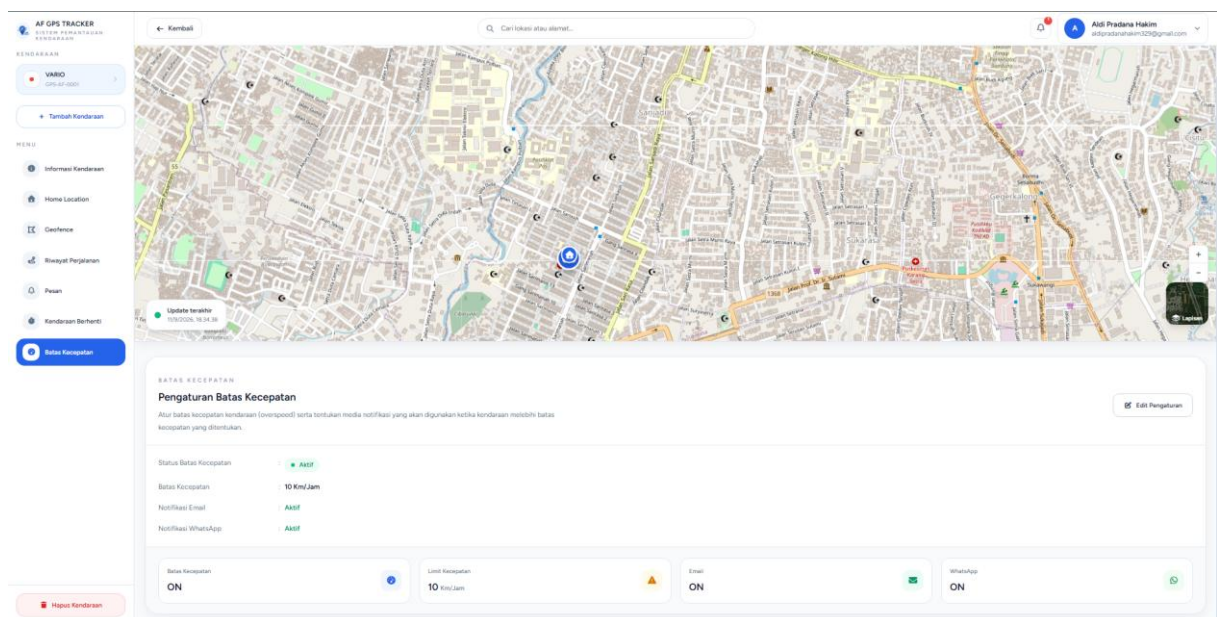
Bagian Kendaraan Berhenti menampilkan riwayat kondisi kendaraan yang berhenti melebihi durasi tertentu, termasuk waktu dan lokasi terjadinya.



Gambar 5.14. Implementasi Antarmuka Halaman Detail Kendaraan Berhenti

g. Batas Kecepatan

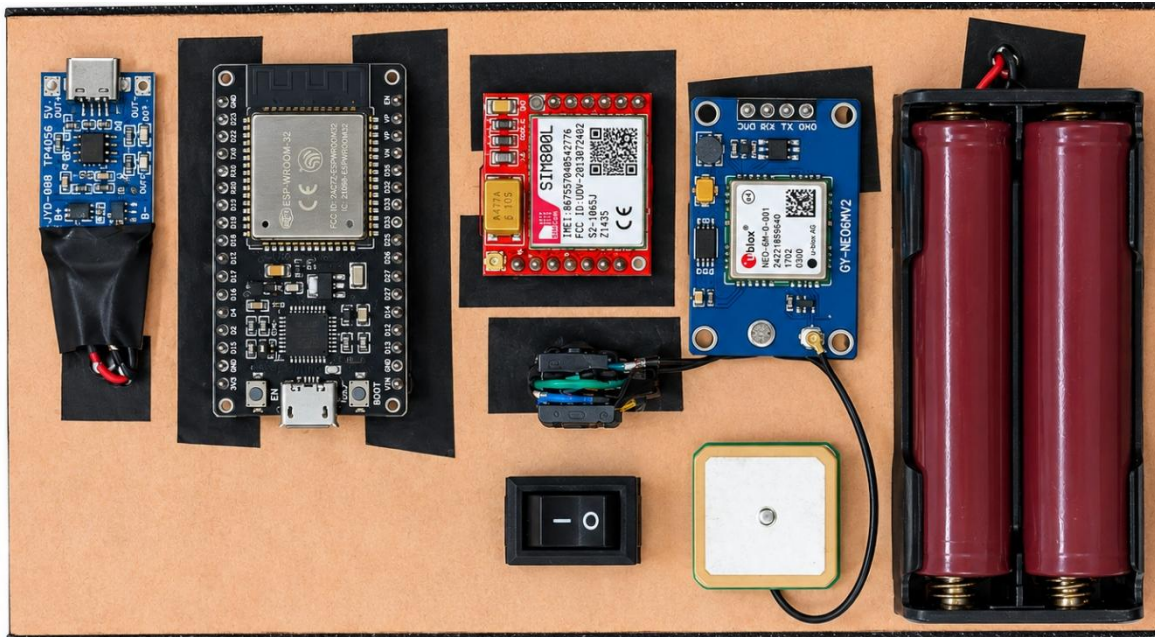
Melalui bagian Batas Kecepatan, pengguna dapat mengatur ambang batas kecepatan kendaraan yang apabila dilampaui akan memicu pengiriman notifikasi overspeed oleh sistem.



Gambar 5.15. Implementasi Antarmuka Halaman Detail Batas Kecepatan

5.1.4 Implementasi Antarmuka Perangkat Keras (Hardware)

Selain antarmuka perangkat lunak, implementasi juga mencakup wujud fisik perangkat keras GPS Tracker yang telah dirakit dan siap dipasang pada kendaraan. Perangkat tersebut terdiri atas mikrokontroler ESP32, modul GPS NEO-6M, modul komunikasi seluler SIM800L, serta baterai lithium sebagai sumber daya, yang dirangkai menjadi satu kesatuan unit sebagaimana ditunjukkan pada gambar berikut.



Gambar 5.16. Implementasi Antarmuka Perangkat Keras

5.1.5 Implementasi Kode Program Perangkat Lunak (Software)

Implementasi kode program perangkat lunak menjelaskan penerapan hasil perancangan pada Bab IV ke dalam bentuk kode program pada backend Laravel. Kode program yang ditampilkan pada bagian ini merupakan bagian representatif yang merepresentasikan logika utama dari setiap fitur, yaitu penerimaan dan pemrosesan data GPS, deteksi geofencing pada ketiga tipe zona (administratif, radius, dan area bebas), deteksi kecepatan berlebih (overspeed), penyiaran data secara real-time, pengiriman notifikasi, serta perhitungan riwayat perjalanan kendaraan.

5.1.5.1. Implementasi Penerimaan dan Validasi Data GPS melalui MQTT

Data koordinat, kecepatan, arah, dan level baterai kendaraan dikirimkan oleh perangkat GPS Tracker ke server melalui protokol MQTT dalam format JSON. Sebelum diproses lebih lanjut, GPSProcessingService melakukan validasi terhadap kelengkapan payload yang diterima, termasuk memastikan field kecepatan (speed) selalu tersedia sebagai salah satu data yang divalidasi.



```

$required = [
    'device_id', 'message_id', 'lat', 'lng',
    'speed', 'heading', 'battery', 'satellite',
    'received_at',
];

foreach ($required as $field) {
    if (! array_key_exists($field, $payload)) {
        throw new ArgumentException(
            sprintf('%s is required.', $field)
        );
    }
}

```

Gambar 5.17. Kode Program Validasi Data GPS Melalui MQTT

Validasi tersebut memastikan setiap payload yang diproses lebih lanjut oleh sistem selalu memiliki data kecepatan (speed) yang valid, sehingga informasi kecepatan kendaraan yang ditampilkan kepada pengguna maupun yang digunakan pada mekanisme deteksi overspeed selalu berasal dari data yang lengkap.

5.1.5.2. Implementasi Deteksi Geofencing Wilayah Administratif dan Area Bebas (Point in Polygon)

Untuk zona geofencing bertipe administratif maupun area bebas (custom polygon), sistem menggunakan algoritma Ray Casting (Point in Polygon) untuk menentukan apakah koordinat kendaraan berada di dalam atau di luar poligon zona yang telah ditentukan.

```

protected function pointInPolygon(float $latitude, float $longitude, array $polygon): bool
{
    $inside = false;
    $total = count($polygon);
    if ($total < 3) {
        return false;
    }

    $j = $total - 1;
    for ($i = 0; $i < $total; $i++) {
        $latI = (float) $polygon[$i][1];
        $lngI = (float) $polygon[$i][0];
        $latJ = (float) $polygon[$j][1];
        $lngJ = (float) $polygon[$j][0];

        $intersect = (($latI > $latitude) !== ($latJ > $latitude)) && (
            $longitude < ($lngJ - $lngI) * ($latitude - $latI)
            / (($latJ - $latI) ? 0.000000001) + $lngI
        );

        if ($intersect) {
            $inside = ! $inside;
        }
        $j = $i;
    }

    return $inside;
}

```

Gambar 5.18. Kode Program Geofencing Administratif dan PIP

Algoritma ini bekerja dengan menarik garis horizontal dari titik koordinat kendaraan dan menghitung jumlah perpotongan garis tersebut dengan sisi-sisi poligon; apabila jumlah perpotongan bernilai ganjil, maka titik dinyatakan berada di dalam zona. Pendekatan yang sama digunakan baik untuk zona wilayah administratif (provinsi, kabupaten/kota, kecamatan, dan kelurahan) maupun zona area bebas yang digambar langsung oleh pengguna pada peta.

5.1.5.3. Implementasi Deteksi Geofencing Radius (Rumus Haversine)

Untuk zona geofencing bertipe radius, sistem menghitung jarak antara koordinat kendaraan dengan titik pusat geofence menggunakan rumus Haversine, kemudian membandingkan hasilnya dengan nilai radius yang ditentukan pengguna.

```

public function contains(Geofence $geofence, float $latitude, float $longitude): bool
{
    $config = $this->validateConfig($geofence->config);

    $distance = $this->calculateDistance(
        $latitude, $longitude,
        $config['center']['lat'], $config['center']['lng']
    );

    return $distance ≤ $config['radius'];
}

public function calculateDistance(float $lat1, float $lng1, float $lat2, float $lng2): float
{
    $lat1 = deg2rad($lat1); $lng1 = deg2rad($lng1);
    $lat2 = deg2rad($lat2); $lng2 = deg2rad($lng2);

    $dLat = $lat2 - $lat1;
    $dLng = $lng2 - $lng1;

    $a = sin($dLat / 2) ** 2 + cos($lat1) * cos($lat2) * sin($dLng / 2) ** 2;
    $c = 2 * atan2(sqrt($a), sqrt(1 - $a));

    return self::EARTH_RADIUS * $c; // EARTH_RADIUS = 6371000 meter
}

```

Gambar 5.19. Kode Program Geofencing Radius

Kendaraan dinyatakan berada di dalam zona radius apabila jarak hasil perhitungan Haversine tidak melebihi nilai radius yang dikonfigurasi. Rumus dan pendekatan perhitungan jarak yang sama juga digunakan pada perhitungan total jarak tempuh riwayat perjalanan kendaraan.

5.1.5.4. Implementasi Deteksi Kecepatan Berlebih (Overspeed)

Selain menampilkan kecepatan kendaraan secara real-time, sistem juga memantau apakah kecepatan yang dikirimkan oleh perangkat GPS melebihi batas kecepatan yang dikonfigurasi pengguna untuk masing-masing kendaraan. Notifikasi overspeed hanya dikirimkan pada saat kecepatan kendaraan melintasi batas dari kondisi normal menjadi melebihi batas, bukan pada setiap titik data selama kendaraan masih berada di atas batas, untuk menghindari notifikasi yang berulang.

```

public function process(Device $device, array $payload): bool
{
    $setting = $this->extractSetting($device);
    if (! $setting['enabled']) {
        return false;
    }

    $speed = (float) ($payload['speed'] ?? 0);
    $isOverspeed = $speed > $setting['limit_kmh'];
    $wasOverspeed = (bool) $device->overspeed_active;

    if ($isOverspeed && ! $wasOverspeed) {
        $device->update(['overspeed_active' => true]);
        return true; // transisi → kirim notifikasi
    }

    if (! $isOverspeed && $wasOverspeed) {
        $device->update(['overspeed_active' => false]);
    }

    return false;
}

```

Gambar 5.20. Kode Program Kecepatan Berlebih

Status `overspeed_active` pada data perangkat digunakan sebagai penanda kondisi kecepatan sebelumnya, sehingga sistem dapat membedakan antara kondisi baru melebihi batas (transisi yang memicu notifikasi) dengan kondisi yang sudah berada di atas batas sejak titik data sebelumnya.

5.1.5.5. Implementasi Penyiaran Data Lokasi secara Real-Time

Setelah data lokasi (termasuk kecepatan) dan hasil pemeriksaan geofencing diproses, sistem langsung menyiarkan data tersebut ke browser pengguna melalui Pusher sebelum melakukan proses penyimpanan ke database, reverse geocoding, maupun pengiriman notifikasi. Hal ini dilakukan agar posisi dan kecepatan kendaraan pada peta selalu diperbarui secepat mungkin tanpa menunggu proses lain yang memakan waktu lebih lama.

```

class VehicleLocationUpdated implements ShouldBroadcastNow
{
    use Dispatchable, SerializesModels;

    public function __construct(
        public string $deviceId,
        public array $payload
    ) {}

    public function broadcastOn(): array
    {
        return [new PrivateChannel("vehicle.{ $this->deviceId }")];
    }

    public function broadcastAs(): string
    {
        return 'location.updated';
    }

    public function broadcastWith(): array
    {
        return $this->payload;
    }
}

```

Gambar 5.21. Kode Program Real-time

Event ini mengimplementasikan interface `ShouldBroadcastNow` sehingga disiarkan secara langsung tanpa melalui antrian (queue), pada kanal privat khusus untuk masing-masing perangkat (`vehicle.{deviceId}`) agar data lokasi suatu kendaraan hanya dapat diterima oleh pengguna yang berhak.

5.1.5.6. Implementasi Pengiriman Notifikasi WhatsApp

Notifikasi WhatsApp dikirimkan melalui API pihak ketiga Fonnte ketika terjadi pelanggaran geofencing, kendaraan berhenti melebihi durasi yang ditentukan, atau kendaraan terdeteksi melebihi batas kecepatan.

```

public function send(Notification $notification): void
{
    $token = config('services.fonnte.token');
    $phone = $this->normalizePhone($notification->device?->user?->phone);

    $response = Http::asForm()
        ->withHeaders(['Authorization' => $token])
        ->timeout(15)
        ->post(config('services.fonnte.url'), [
            'target' => $phone,
            'message' => $this->buildMessage($notification),
        ]);

    if (! $response->successful()) {
        throw new RuntimeException(
            "Fonnte API gagal dengan status {$response->status()}: {$response->body()}"
        );
    }
}

```

Gambar 5.22. Kode Program Notifikasi WhatsApp

Nomor tujuan diperoleh dari data pengguna yang terhubung dengan perangkat, dan permintaan dikirimkan menggunakan HTTP Client Laravel ke endpoint Fonnte dengan token API sebagai header otorisasi. Apabila permintaan gagal, sistem melempar exception yang ditangani pada lapisan pemanggil agar kegagalan satu kanal notifikasi tidak menghentikan proses notifikasi kanal lainnya.

5.1.5.7. Implementasi Perhitungan Riwayat Perjalanan (Trip)

Data riwayat perjalanan kendaraan yang tersimpan dikelompokkan menjadi beberapa trip berdasarkan riwayat berhenti, kemudian jarak tempuh setiap trip dihitung dengan menjumlahkan jarak antar titik koordinat yang berurutan menggunakan rumus Haversine.

```

protected function haversineKm(float $lat1, float $lng1, float $lat2, float $lng2): float
{
    $latDelta = deg2rad($lat2 - $lat1);
    $lngDelta = deg2rad($lng2 - $lng1);

    $a = sin($latDelta / 2) ** 2
        + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) * sin($lngDelta / 2) ** 2;

    $c = 2 * asin(sqrt($a));

    return self::EARTH_RADIUS_KM * $c;
}

```

Gambar 5.23. Kode Program Riwayat Perjalanan

Dengan menjumlahkan jarak antar-titik koordinat yang berurutan pada satu trip, sistem dapat menyajikan estimasi total jarak tempuh kendaraan pada halaman riwayat perjalanan tanpa memerlukan sensor odometer tambahan pada perangkat keras.

5.1.6 Implementasi Kode Program Perangkat Keras (Firmware ESP32)

Selain kode program pada sisi backend website, implementasi juga mencakup program (firmware) yang berjalan pada perangkat GPS Tracker. Firmware ditulis menggunakan Arduino IDE dalam bahasa C/C++ dan dijalankan pada mikrokontroler ESP32, dengan berkas sumber utama firmware/afgps_prod/afgps_prod.ino. Firmware ini bertugas membaca data koordinat dari modul GPS NEO-6M, membangun payload JSON yang ditandatangani secara digital, kemudian mengirimkannya ke server melalui modul SIM800L menggunakan protokol MQTT. Bagian ini menampilkan kode program yang merepresentasikan mekanisme utama firmware, yaitu koneksi jaringan GPRS, akuisisi data GPS, pengiriman data melalui MQTT dengan tanda tangan HMAC-SHA256, serta mekanisme siklus utama dan reconnect otomatis.

5.1.6.1. Implementasi Koneksi Jaringan GPRS melalui Modul SIM800L

Sebelum dapat mengirimkan data ke server, perangkat terlebih dahulu harus terhubung ke jaringan seluler dan mengaktifkan layanan data (GPRS) melalui modul SIM800L. Firmware menggunakan pustaka TinyGSM untuk berkomunikasi dengan modul SIM800L tanpa perlu menuliskan perintah AT command secara manual.

```
bool connectGPRS()
{
    Serial.print("Menunggu jaringan... ");

    if (!modem.waitForNetwork(60000L))
    {
        Serial.println("GAGAL");
        return false;
    }

    Serial.println("OK");
    Serial.print("Menghubungkan GPRS dengan APN: ");
    Serial.println(APN);

    if (!modem.gprsConnect(APN, "", ""))
    {
        Serial.println("GPRS GAGAL");
        return false;
    }

    Serial.println("GPRS BERHASIL");
    Serial.print("IP Address: ");
    Serial.println(modem.localIP());

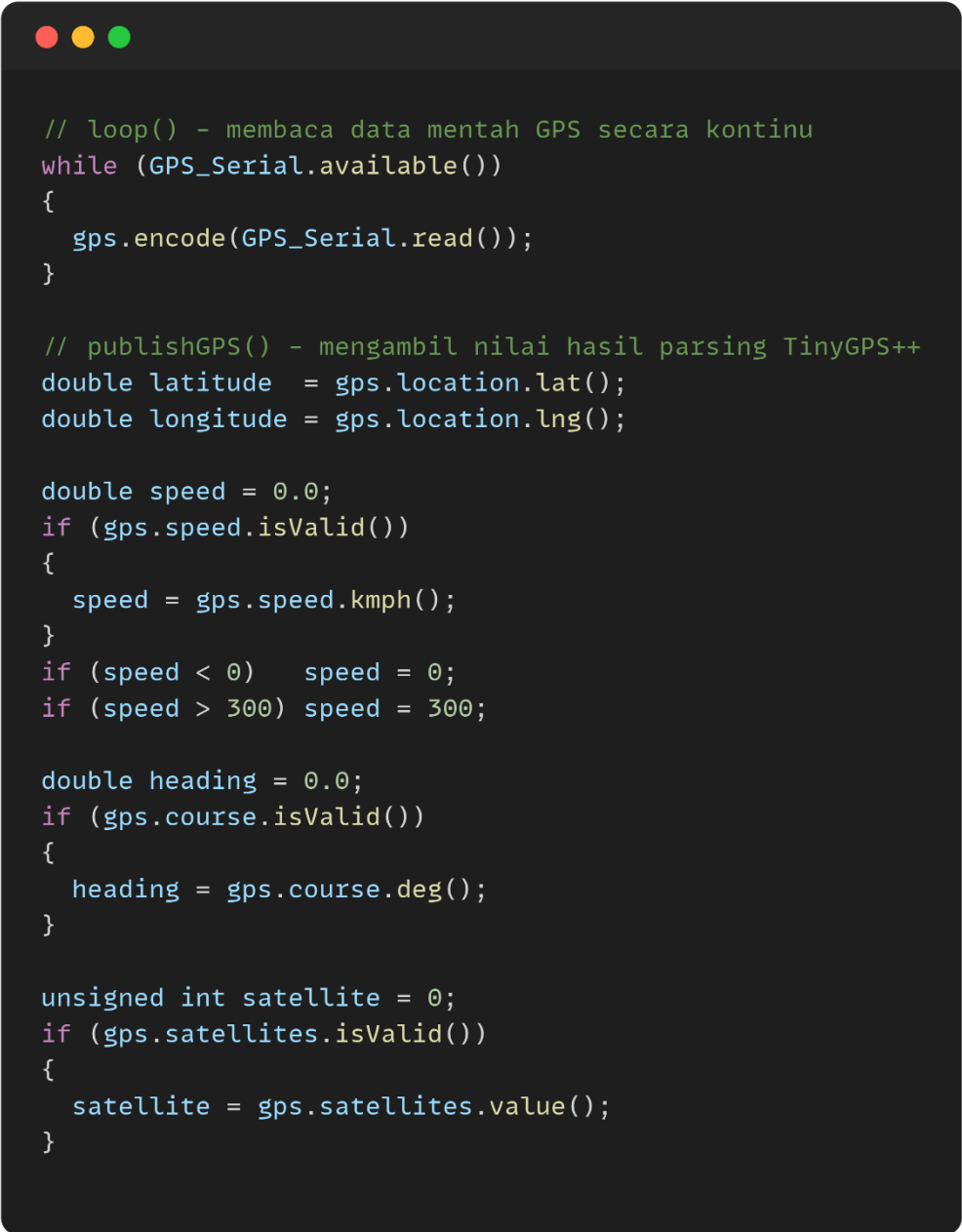
    return true;
}
```

Gambar 5.24. Kode Program Koneksi ke GPRS

Fungsi ini menunggu perangkat memperoleh sinyal jaringan seluler selama maksimal 60 detik, kemudian melakukan koneksi GPRS menggunakan Access Point Name (APN) yang telah dikonfigurasi. Apabila salah satu tahapan gagal, fungsi mengembalikan nilai false sehingga firmware dapat mencoba kembali pada kesempatan berikutnya tanpa perlu me-restart perangkat secara manual.

5.1.6.2. Implementasi Akuisisi Data GPS melalui Modul NEO-6M

Data koordinat lokasi kendaraan diperoleh dari modul GPS NEO-6M yang mengirimkan data secara kontinu dalam format kalimat NMEA melalui jalur serial. Firmware menggunakan pustaka TinyGPS++ untuk mengurai (parsing) data mentah tersebut menjadi nilai latitude, longitude, kecepatan, arah, dan jumlah satelit yang dapat langsung digunakan.



```

// loop() - membaca data mentah GPS secara kontinu
while (GPS_Serial.available())
{
    gps.encode(GPS_Serial.read());
}

// publishGPS() - mengambil nilai hasil parsing TinyGPS++
double latitude = gps.location.lat();
double longitude = gps.location.lng();

double speed = 0.0;
if (gps.speed.isValid())
{
    speed = gps.speed.kmph();
}
if (speed < 0)    speed = 0;
if (speed > 300) speed = 300;

double heading = 0.0;
if (gps.course.isValid())
{
    heading = gps.course.deg();
}

unsigned int satelllite = 0;
if (gps.satellites.isValid())
{
    satelllite = gps.satellites.value();
}

```

Gambar 5.25. Kode Program Akuisisi Data GPS

Modul GPS NEO-6M terhubung ke ESP32 melalui komunikasi serial UART pada pin GPIO 16 (RX) dan GPIO 17 (TX) dengan kecepatan 9600 bps. Nilai kecepatan yang diperoleh dari TinyGPS++ (`gps.speed.kmph()`) dibatasi (clamping) pada rentang 0 hingga 300 km/j sebelum dikirimkan, sebagai lapisan validasi tambahan terhadap kemungkinan nilai anomali dari modul GPS.

5.1.6.3. Implementasi Pengiriman Data Lokasi melalui MQTT dengan Tanda Tangan HMAC-SHA256

Data yang telah diperoleh disusun menjadi payload JSON kemudian dikirimkan ke server melalui protokol MQTT menggunakan pustaka PubSubClient, dengan koneksi data yang disediakan oleh modul SIM800L. Untuk menjamin keaslian data dan mencegah

pemalsuan payload oleh pihak yang tidak berwenang, setiap payload dilengkapi dengan tanda tangan digital HMAC-SHA256 yang dihitung dari kombinasi nilai-nilai data menggunakan kunci rahasia yang hanya diketahui oleh perangkat dan server.

```
bool connectMQTT()
{
    mqtt.setServer(MQTT_BROKER, MQTT_PORT);

    // WAJIB: payload bertanda tangan berukuran sekitar 300 byte, melewati
    // buffer bawaan PubSubClient yang hanya 256 byte.
    mqtt.setBufferSize(512);

    String clientId = "ESP32-" + String(DEVICE_ID) + "-" + String(random(0xffff), HEX);

    if (mqtt.connect(clientId.c_str(), MQTT_USERNAME, MQTT_PASSWORD))
    {
        Serial.println("MQTT CONNECTED!");
        return true;
    }

    Serial.print("MQTT GAGAL, state = ");
    Serial.println(mqtt.state());
    return false;
}

// Penyusunan payload (disederhanakan)
String payload = "{";
payload += "\"device_id\": \"";    payload += DEVICE_ID;    payload += "\"";
payload += "\"lat\": \"";        payload += sLatitude;    payload += "\"";
payload += "\"lng\": \"";        payload += sLongitude;    payload += "\"";
payload += "\"speed\": \"";      payload += sSpeed;      payload += "\"";
payload += "\"battery\": \"";    payload += sBattery;    payload += "\"";
payload += "\"received_at\": \""; payload += receivedAt;    payload += "\"";
payload += "\"signature\": \"";  payload += signature;    payload += "\"";
payload += "}";

bool success = mqtt.publish(MQTT_TOPIC, payload.c_str());
```

Gambar 5.26. Kode Program Pengiriman Data Lokasi

Signature dihitung menggunakan fungsi HMAC-SHA256 bawaan pustaka mbedTLS pada ESP32, atas gabungan nilai `message_id`, `lat`, `lng`, `speed`, `heading`, `battery`, `satellite`, dan `received_at`. Pada sisi server, `MQTTSignatureService` melakukan perhitungan ulang dan membandingkannya dengan nilai signature yang diterima sebelum payload diproses lebih lanjut oleh `GPSProcessingService`, sehingga payload yang tidak memiliki signature valid akan ditolak. Perlu dicatat bahwa nilai `battery` pada firmware ini masih berupa nilai tetap (placeholder) karena rangkaian pembacaan tegangan baterai (ADC/voltage divider) belum diimplementasikan pada revisi perangkat keras saat ini; pengujian fitur deteksi baterai lemah pada sisi server tetap dapat dilakukan dengan mengirimkan nilai `battery` secara manual melalui payload uji, sebagaimana disajikan pada sub-bab Pengujian. Pembacaan tegangan baterai secara langsung dicatat sebagai pengembangan lanjutan pada revisi perangkat keras berikutnya.

5.1.6.4. Implementasi Siklus Utama dan Reconnect Otomatis

Fungsi `loop()` pada firmware menjalankan pembacaan data GPS secara terus-menerus, sementara pengiriman data ke server dibatasi oleh jeda waktu (`PUBLISH_INTERVAL`) menggunakan penghitung waktu berbasis `millis()`. Firmware juga secara aktif memantau status koneksi GPRS dan MQTT pada setiap iterasi, dan akan mencoba menyambung ulang secara otomatis apabila koneksi terputus.

```
void loop()
{
  while (GPS_Serial.available())
  {
    gps.encode(GPS_Serial.read());
  }

  if (!mqtt.connected())
  {
    Serial.println("MQTT TERPUTUS");

    if (!modem.isGprsConnected())
    {
      Serial.println("GPRS terputus. Mencoba reconnect GPRS...");
      if (!connectGPRS())
      {
        delay(5000);
        return;
      }
    }

    if (!connectMQTT())
    {
      delay(5000);
      return;
    }

    Serial.println("MQTT berhasil reconnect!");
  }

  mqtt.loop();

  if (millis() - lastPublish ≥ PUBLISH_INTERVAL)
  {
    lastPublish = millis();
    publishGPS();
  }

  delay(10);
}
```

Gambar 5.27. Kode Program Siklus Utama dan Reconnect Otomatis

Pada firmware produksi, `PUBLISH_INTERVAL` diatur selama 10.000 milidetik (10 detik), sehingga perangkat mengirimkan data lokasi kurang lebih setiap 10 detik selama koneksi tersedia. Mekanisme reconnect memeriksa status GPRS terlebih dahulu sebelum mencoba menyambungkan ulang MQTT, dengan jeda 5 detik antar percobaan,

sehingga perangkat dapat pulih secara mandiri dari gangguan sinyal tanpa memerlukan intervensi manual berupa restart perangkat.

5.1.7 Implementasi Database

Implementasi basis data dilakukan menggunakan PostgreSQL yang dikelola melalui Laravel Artisan Migration, sehingga struktur skema basis data dapat dibuat dan diperbarui secara terversi mengikuti perubahan rancangan Entity Relationship Diagram (ERD) pada Bab IV. Tabel 5.3 berikut menunjukkan tabel utama yang digunakan pada sistem AF GPS Tracker.

Tabel 5.3. Implementasi Database

No	Tabel	Deskripsi
1	users	Menyimpan data akun pengguna, termasuk status admin (is_admin).
2	devices	Menyimpan data perangkat GPS, kredensial aktivasi, status heartbeat, serta pengaturan berhenti dan overspeed.
3	vehicles	Menyimpan data kendaraan yang terhubung dengan perangkat GPS.
4	device_logs	Menyimpan log mentah data lokasi yang diterima dari perangkat.
5	travel_histories	Menyimpan titik koordinat riwayat perjalanan kendaraan beserta kecepatan dan alamat hasil reverse geocoding.
6	stop_histories	Menyimpan data riwayat kondisi kendaraan berhenti.
7	geofences	Menyimpan konfigurasi zona geofencing bertipe administratif, radius, atau custom polygon.
8	notifications	Menyimpan log notifikasi yang dikirimkan kepada pengguna.

5.2 Pengujian

Sub-bab ini menjelaskan proses pengujian yang dilakukan terhadap sistem AF GPS Tracker secara menyeluruh, mencakup pengujian perangkat lunak (website) menggunakan metode black-box testing terhadap seluruh use case yang telah dirancang pada Bab IV, serta pengujian perangkat keras (firmware) untuk memastikan setiap modul fungsi pada perangkat GPS Tracker bekerja sesuai dengan yang diharapkan.

5.2.1 Rencana Pengujian

Pengujian sistem AF GPS Tracker dilakukan menggunakan metode black-box testing untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan yang telah dianalisis pada Bab III dan dirancang pada Bab IV. Pengujian dilakukan terhadap seluruh use case, yaitu UC-01 sampai dengan UC-26, dengan setiap use case diuji menggunakan satu atau lebih skenario yang mencakup kondisi normal (positif) maupun kondisi tidak normal (negatif). Setiap skenario pengujian diberi kode Test Case ID dengan format TC-<nomor UC>-<nomor urut>, misalnya TC-07-01 untuk skenario pertama pada UC-05. Pengujian dilakukan secara manual pada browser Google Chrome dalam lingkungan produksi (production) sistem yang telah dideploy, dengan mengacu pada dokumen Test Case Black Box Testing yang telah disusun sebelumnya.

Tabel 5.4. Rancangan Pengujian Perangkat Lunak

No	Kode UC	Nama Use Case	Jumlah Kasus Uji
1	UC-01	Menambahkan data kendaraan	3
2	UC-02	Menambahkan perangkat GPS	3
3	UC-03	Mengubah data kendaraan	3
4	UC-04	Menghapus kendaraan	2
5	UC-05	Memantau kendaraan secara real-time	5
6	UC-06	Melihat status perangkat GPS	2
7	UC-07	Mengelola zona geofencing	7
8	UC-08	Mengatur notifikasi pelanggaran geofencing	3
9	UC-09	Mengelola home location	4
10	UC-10	Meninjau riwayat perjalanan	4
11	UC-11	Meninjau riwayat berhenti	4
12	UC-12	Mengekspor riwayat perjalanan	3
13	UC-13	Mengekspor riwayat berhenti	3
14	UC-14	Mengatur durasi waktu berhenti	3
15	UC-15	Mengaktifkan atau menonaktifkan notifikasi berhenti	4
16	UC-16	Menerima notifikasi berhenti	4
17	UC-17	Menerima notifikasi pelanggaran geofencing	4
18	UC-18	Meninjau notifikasi kendaraan	5
19	UC-19	Melihat dashboard admin	2
20	UC-20	Menambahkan perangkat GPS oleh admin	2
21	UC-21	Mengekspor kredensial perangkat ke PDF	2
22	UC-22	Menghapus perangkat GPS oleh admin	2
23	UC-23	Meninjau data pengguna oleh admin	3
24	UC-24	Menghapus data pengguna oleh admin	2
25	UC-25	Mengatur batas kecepatan kendaraan	3
26	UC-26	Menerima notifikasi kecepatan berlebih	4
Total			86

Selain pengujian perangkat lunak, dilakukan pula pengujian black-box terhadap perangkat keras (firmware) GPS Tracker yang terdiri atas ESP32, modul GPS NEO-6M, dan modul GSM/GPRS SIM800L. Skenario pengujian perangkat keras tidak mengacu pada use case Bab IV, karena Bab IV hanya memodelkan interaksi pengguna melalui website. Skenario disusun berdasarkan fungsi-fungsi yang telah diimplementasikan pada firmware (firmware/afgps_prod/afgps_prod.ino) serta aturan validasi pada sisi server yang menerima data dari perangkat (GPSProcessingService dan MQTTSignatureService). Rencana pengujian perangkat keras disusun ke dalam 11 modul fungsi firmware dengan total 27 kasus uji, sebagaimana dirangkum pada tabel berikut.

Tabel 5.5. Rencana Pengujian Perangkat Keras

No	Kode Modul	Nama Modul Firmware	Jumlah Kasus Uji
1	HW-01	Inisialisasi Perangkat dan Modem SIM800L	2
2	HW-02	Koneksi Jaringan GPRS	3
3	HW-03	Koneksi ke MQTT Broker	2
4	HW-04	Akuisisi Sinyal GPS (GPS NEO-6M)	3
5	HW-05	Pengiriman Data Lokasi dengan Tanda Tangan HMAC-SHA256	5
6	HW-06	Penyaringan Perpindahan Posisi (Position Threshold)	2
7	HW-07	Deteksi Pesan Duplikat (Message ID)	2
8	HW-08	Deteksi Baterai Lemah	2
9	HW-09	Reconnect Otomatis saat Koneksi Terputus	3
10	HW-10	Pengiriman Data Berkala (Publish Interval)	1

No	Kode Modul	Nama Modul Firmware	Jumlah Kasus Uji
11	HW-11	Deteksi Perangkat Offline/Online (Heartbeat)	2
Total			27

5.2.2 Hasil Pengujian Perangkat Lunak (Software)

Bagian ini menampilkan hasil pengujian black-box untuk setiap use case yang telah diuji. Kolom “Hasil Aktual” menunjukkan hasil yang benar-benar diperoleh pada saat pengujian dijalankan, dan kolom “Status” menunjukkan apakah hasil aktual telah sesuai dengan hasil yang diharapkan.

5.2.2.1. UC-01 Menambahkan Data Kendaraan

Tabel 5.6. Test case UC-01

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-01-01	Menambahkan kendaraan dengan data valid	Mengisi nama kendaraan, nomor polisi, tipe, warna, dan marker kemudian menekan Simpan	Sistem menyimpan kendaraan dan menghubungkannya dengan perangkat GPS	Kendaraan berhasil disimpan dan sistem mengarahkan pengguna ke halaman Home	Berhasil
TC-01-02	Menambahkan kendaraan dengan data tidak lengkap	Mengosongkan salah satu data kendaraan kemudian menyimpan	Sistem menampilkan pesan validasi	Sistem menolak dan menampilkan pesan bahwa nama kendaraan harus diisi	Berhasil
TC-01-03	Membatalkan penambahan kendaraan	Mengisi data kendaraan kemudian membatalkan proses	Sistem tidak menyimpan data kendaraan	Data kendaraan tidak tersimpan	Berhasil

5.2.2.2. UC-02 Menambahkan Perangkat GPS

Tabel 5.7. Test case UC-02

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-02-01	Menambahkan perangkat dengan data valid	Memasukkan Device ID dan password perangkat yang valid kemudian melakukan aktivasi	Sistem menghubungkan perangkat dengan akun pengguna	Perangkat baru berhasil terhubung dengan akun pengguna	Berhasil
TC-02-02	Menambahkan perangkat dengan Device ID/password salah	Memasukkan data perangkat yang tidak valid	Sistem menolak penambahan perangkat	Sistem menolak dan menampilkan pesan bahwa data perangkat tidak valid	Berhasil
TC-02-03	Menambahkan perangkat yang telah digunakan	Memasukkan perangkat yang telah terhubung dengan akun lain	Sistem menampilkan informasi bahwa perangkat tidak dapat ditambahkan	Sistem menampilkan pesan bahwa perangkat sudah digunakan oleh akun lain	Berhasil

5.2.2.3. UC-03 Mengubah Data Kendaraan

Tabel 5.8. Test case UC-03

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-03-01	Mengubah data kendaraan dengan data valid	Membuka Detail Kendaraan, memilih Edit, mengubah data, kemudian menyimpan	Sistem memperbarui data kendaraan	Data kendaraan berhasil diperbarui sesuai data terbaru	Berhasil
TC-03-02	Mengubah kendaraan dengan data tidak valid	Mengubah data kendaraan dengan nilai yang tidak sesuai kemudian menyimpan	Sistem menampilkan pesan kesalahan	Data tidak disimpan dan sistem menampilkan pesan untuk mengisi data yang masih kosong	Berhasil
TC-03-03	Membatalkan perubahan kendaraan	Mengubah data kemudian membatalkan proses	Sistem mempertahankan data sebelumnya	Data kendaraan tetap seperti sebelum diubah	Berhasil

5.2.2.4. UC-04 Menghapus Kendaraan

Tabel 5.9. Test case UC-04

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-04-01	Menghapus kendaraan	Membuka Detail Kendaraan, memilih Hapus Kendaraan, kemudian mengonfirmasi	Sistem menghapus keterkaitan kendaraan dari akun	Data kendaraan berhasil dihapus dan hilang dari database	Berhasil
TC-04-02	Membatalkan penghapusan kendaraan	Memilih Hapus Kendaraan kemudian membatalkan konfirmasi	Sistem mempertahankan kendaraan	Data kendaraan tidak jadi dihapus dan tetap ada	Berhasil

5.2.2.5. UC-05 Memantau Kendaraan secara Real-time

Tabel 5.10. Test case UC-05

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-05-01	Memantau kendaraan secara real-time	Membuka Home dan memilih kendaraan yang terdaftar	Sistem menampilkan posisi kendaraan pada peta	Sistem berhasil menampilkan posisi kendaraan sesuai data lokasi yang dikirim perangkat GPS	Berhasil
TC-05-02	Menerima pembaruan lokasi	Membiarkan perangkat mengirim data lokasi baru	Posisi marker kendaraan diperbarui secara real-time	Posisi marker kendaraan diperbarui secara real-time setiap ada data lokasi baru	Berhasil
TC-05-03	Data lokasi belum tersedia	Membuka kendaraan yang belum memiliki data lokasi	Sistem menampilkan informasi bahwa lokasi belum tersedia	Sistem menampilkan informasi bahwa data lokasi belum tersedia	Berhasil
TC-05-04	Koneksi real-time terputus	Memutus koneksi real-time saat monitoring berlangsung	Sistem menampilkan status koneksi dan	Ketika koneksi diputus, sistem menampilkan	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
			menunggu koneksi kembali	status perangkat menjadi offline	
TC-05-05	Menampilkan dan memperbarui informasi kecepatan kendaraan secara real-time	Membuka halaman Home/Detail Kendaraan pada kendaraan yang perangkat GPS-nya aktif mengirim data, kemudian memperhatikan nilai kecepatan yang ditampilkan saat kendaraan bergerak dengan kecepatan yang berbeda-beda	Sistem menampilkan nilai kecepatan kendaraan (km/j) sesuai data yang dikirimkan perangkat GPS dan memperbarui nilai tersebut secara real-time setiap kali data lokasi baru diterima	Nilai kecepatan kendaraan tampil sesuai kecepatan aktual perangkat dan selalu diperbarui secara real-time setiap ada data baru dari perangkat GPS	Berhasil

5.2.2.6. UC-06 Melihat Status Perangkat GPS

Tabel 5.11. Test case UC-06

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-06-01	Melihat perangkat aktif	Membuka Home/Detail Kendaraan ketika perangkat masih mengirim heartbeat	Sistem menampilkan status perangkat aktif	Perangkat tampil dengan status online dan aktif	Berhasil
TC-06-02	Melihat perangkat tidak aktif	Membuka Home/Detail Kendaraan ketika perangkat tidak mengirim heartbeat sesuai batas waktu	Sistem menampilkan status perangkat tidak aktif	Perangkat tampil dengan status offline dan tidak aktif	Berhasil

5.2.2.7. UC-07 Mengelola Zona Geofencing

Tabel 5.12. Test case UC-07

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-07-01	Menambahkan geofence administratif	Memilih kendaraan, memilih jenis administratif, menentukan wilayah, kemudian menyimpan	Sistem menyimpan dan menampilkan geofence administratif	Data berhasil ditambahkan dan sistem menampilkan area administratif pada peta	Berhasil
TC-07-02	Menambahkan geofence radius	Memilih tipe radius, menentukan titik dan radius, kemudian menyimpan	Sistem menyimpan dan menampilkan area radius pada peta	Data berhasil disimpan dan sistem menampilkan area radius pada peta	Berhasil
TC-07-03	Menambahkan geofence polygon	Memilih tipe polygon, menggambar area pada peta, kemudian menyimpan	Sistem menyimpan dan menampilkan area polygon	Data berhasil disimpan dan sistem menampilkan area polygon pada peta	Berhasil
TC-07-04	Konfigurasi geofence tidak lengkap	Tidak melengkapi konfigurasi geofence kemudian menyimpan	Sistem meminta konfigurasi dilengkapi	Data tidak disimpan dan sistem meminta konfigurasi dilengkapi	Berhasil
TC-07-05	Wilayah administratif tidak tersedia	Memilih wilayah administratif yang tidak dapat dimuat	Sistem menampilkan pesan bahwa	Sistem menampilkan pesan bahwa data	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
			wilayah tidak dapat dimuat	wilayah tidak ditemukan	
TC-07-06	Membatalkan pengelolaan geofence	Melakukan perubahan geofence kemudian membatalkan proses	Sistem tidak menyimpan perubahan	Konfigurasi geofence tetap tidak berubah	Berhasil
TC-07-07	Menghapus geofence	Memilih geofence kemudian memilih hapus dan mengonfirmasi	Sistem menghapus geofence	Data geofence berhasil dihapus dan sistem menampilkan notifikasi berhasil	Berhasil

5.2.2.8. UC-08 Mengatur Notifikasi Pelanggaran Geofencing

Tabel 5.13. Test case UC-08

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-08-01	Mengatur channel notifikasi geofence	Membuka pengaturan geofence, memilih channel notifikasi, kemudian menyimpan	Sistem menyimpan konfigurasi channel notifikasi	Sistem berhasil menyimpan konfigurasi channel notifikasi	Berhasil
TC-08-02	Membatalkan pengaturan notifikasi	Mengubah channel kemudian membatalkan perubahan	Sistem mempertahankan pengaturan sebelumnya	Pengaturan channel notifikasi menggunakan toggle aktif/nonaktif langsung tanpa tombol Batal terpisah, sehingga pengaturan sebelumnya tetap berlaku hingga toggle diubah	Berhasil
TC-08-03	Penyimpanan konfigurasi gagal	Melakukan penyimpanan ketika terjadi kegagalan proses	Sistem menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan saat penyimpanan gagal	Berhasil

5.2.2.9. UC-09 Mengelola Home Location

Tabel 5.14. Test case UC-09

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-09-01	Menambahkan Home Location	Membuka fitur Home Location, menekan Tambah, mencari lokasi, menentukan titik, kemudian menyimpan	Sistem menyimpan koordinat Home Location dan menampilkan marker pada peta	Koordinat Home Location berhasil tersimpan dan marker tampil pada peta	Berhasil
TC-09-02	Memindahkan Home Location	Memilih Home Location kemudian menentukan titik baru	Sistem memperbarui koordinat berdasarkan titik baru	Koordinat Home Location berhasil diperbarui	Berhasil
TC-09-03	Menghapus Home Location	Memilih Hapus Home Location kemudian mengonfirmasi	Sistem menghapus Home Location	Home Location berhasil dihapus dan marker hilang dari peta	Berhasil
TC-09-04	Membatalkan pengelolaan Home Location	Melakukan perubahan kemudian	Sistem mempertahankan	Data Home Location tetap seperti sebelumnya	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
		membatalkan proses	Home Location sebelumnya		

5.2.2.10. UC-10 Meninjau Riwayat Perjalanan

Tabel 5.15. Test case UC-10

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-10-01	Melihat riwayat perjalanan	Memilih kendaraan dan periode perjalanan kemudian meminta data	Sistem menampilkan rute perjalanan pada peta	Sistem berhasil menampilkan rute perjalanan pada peta	Berhasil
TC-10-02	Menjalankan playback perjalanan	Memilih fitur Playback kemudian menjalankannya	Sistem menampilkan pergerakan kendaraan mengikuti rute	Sistem berhasil menampilkan pergerakan kendaraan mengikuti rute perjalanan	Berhasil
TC-10-03	Tidak terdapat data perjalanan	Memilih periode yang tidak memiliki data	Sistem menampilkan informasi bahwa riwayat tidak tersedia	Sistem menampilkan informasi bahwa riwayat perjalanan tidak tersedia	Berhasil
TC-10-04	Gagal memuat riwayat	Melakukan permintaan riwayat ketika data gagal dimuat	Sistem menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan saat data gagal dimuat	Berhasil

5.2.2.11. UC-11 Meninjau Riwayat Berhenti

Tabel 5.16. Test case UC-11

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-11-01	Melihat riwayat berhenti	Memilih kendaraan dan periode riwayat berhenti	Sistem menampilkan daftar titik berhenti	Sistem berhasil menampilkan daftar titik berhenti	Berhasil
TC-11-02	Melihat detail titik berhenti	Memilih salah satu data berhenti	Sistem menampilkan informasi lokasi dan waktu berhenti	Sistem berhasil menampilkan informasi lokasi dan waktu berhenti	Berhasil
TC-11-03	Tidak terdapat data berhenti	Memilih periode tanpa data berhenti	Sistem menampilkan informasi bahwa data berhenti tidak tersedia	Sistem menampilkan informasi bahwa data berhenti tidak tersedia	Berhasil
TC-11-04	Gagal memuat data berhenti	Membuka riwayat ketika data gagal dimuat	Sistem menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan saat data gagal dimuat	Berhasil

5.2.2.12. UC-12 Mengekspor Riwayat Perjalanan

Tabel 5.17. Test case UC-12

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-12-01	Export riwayat perjalanan	Memilih kendaraan dan periode kemudian menekan Export PDF	Sistem menghasilkan dokumen PDF dan menyediakan file untuk diunduh	Dokumen PDF riwayat perjalanan berhasil dihasilkan dan dapat diunduh	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-12-02	Export tanpa data perjalanan	Melakukan Export PDF pada periode tanpa data	Sistem tidak menghasilkan dokumen dan menampilkan informasi bahwa data tidak tersedia	Sistem menampilkan informasi bahwa data tidak tersedia untuk periode tersebut	Berhasil
TC-12-03	Gagal membuat PDF	Menjalankan export ketika proses pembuatan PDF mengalami kegagalan	Sistem menampilkan pesan kesalahan	Proses pembuatan PDF tetap berhasil diselesaikan pada seluruh pengujian yang dilakukan	Berhasil

5.2.2.13. UC-13 Mengekspor Riwayat Berhenti

Tabel 5.18. Test case UC-13

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-13-01	Export riwayat berhenti	Memilih kendaraan dan periode kemudian menekan Export PDF	Sistem menghasilkan PDF riwayat berhenti dan menyediakan file untuk diunduh	Dokumen PDF riwayat berhenti berhasil dihasilkan dan dapat diunduh	Berhasil
TC-13-02	Export tanpa data berhenti	Melakukan Export PDF pada periode tanpa data	Sistem menampilkan informasi bahwa data tidak tersedia	Sistem menampilkan informasi bahwa data tidak tersedia untuk periode tersebut	Berhasil
TC-13-03	Gagal membuat PDF	Menjalankan export ketika pembuatan PDF gagal	Sistem menampilkan pesan kesalahan	Proses pembuatan PDF tetap berhasil diselesaikan pada seluruh pengujian yang dilakukan	Berhasil

5.2.2.14. UC-14 Mengatur Durasi Waktu Berhenti

Tabel 5.19. Test case UC-14

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-14-01	Mengatur durasi berhenti dengan nilai valid	Membuka pengaturan durasi, memasukkan nilai valid, kemudian menyimpan	Sistem menyimpan durasi berhenti sebagai batas notifikasi	Durasi berhenti berhasil tersimpan dengan nilai yang valid	Berhasil
TC-14-02	Memasukkan durasi tidak valid	Memasukkan nilai durasi yang tidak sesuai kemudian menyimpan	Sistem menampilkan pesan validasi	Sistem menampilkan pesan validasi untuk nilai durasi yang tidak sesuai	Berhasil
TC-14-03	Membatalkan perubahan durasi	Mengubah nilai durasi kemudian membatalkan proses	Sistem mempertahankan pengaturan sebelumnya	Pengaturan durasi berhenti tetap seperti sebelumnya	Berhasil

5.2.2.15. UC-15 Mengaktifkan/Menonaktifkan Notifikasi Berhenti

Tabel 5.20. Test case UC-15

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-15-01	Mengaktifkan notifikasi berhenti	Membuka pengaturan notifikasi dan mengaktifkan fitur	Sistem menyimpan status notifikasi sebagai aktif	Status notifikasi berhenti berhasil tersimpan sebagai aktif	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
		kemudian menyimpan			
TC-15-02	Menonaktifkan notifikasi berhenti	Membuka pengaturan notifikasi dan menonaktifkan fitur kemudian menyimpan	Sistem menyimpan status notifikasi sebagai nonaktif	Status notifikasi berhenti berhasil tersimpan sebagai nonaktif	Berhasil
TC-15-03	Membatalkan perubahan status	Mengubah status notifikasi kemudian membatalkan proses	Sistem mempertahankan status sebelumnya	Status notifikasi berhenti tetap seperti sebelumnya	Berhasil
TC-15-04	Penyimpanan status gagal	Mengubah status dan melakukan penyimpanan ketika proses gagal	Sistem menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan saat penyimpanan status gagal	Berhasil

5.2.2.16. UC-16 Menerima Notifikasi Berhenti

Tabel 5.21. Test case UC-16

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-16-01	Kendaraan berhenti melebihi batas waktu	Mengatur notifikasi aktif, menentukan durasi berhenti, kemudian membuat kendaraan berhenti melebihi durasi tersebut	Sistem mendeteksi kondisi berhenti, memperoleh alamat, dan mengirimkan notifikasi kepada pengguna	Sistem berhasil mendeteksi kondisi berhenti, memperoleh alamat, dan mengirimkan notifikasi kepada pengguna	Berhasil
TC-16-02	Durasi berhenti belum melewati batas	Membuat kendaraan berhenti tetapi belum melewati durasi yang ditentukan	Sistem tidak mengirimkan notifikasi	Sistem tidak mengirimkan notifikasi karena durasi belum melewati batas	Berhasil
TC-16-03	Notifikasi berhenti dinonaktifkan	Menonaktifkan notifikasi kemudian membuat kendaraan berhenti melebihi batas	Sistem tidak mengirimkan notifikasi berhenti	Sistem tidak mengirimkan notifikasi karena fitur telah dinonaktifkan	Berhasil
TC-16-04	Alamat tidak berhasil diperoleh	Membuat kondisi kendaraan berhenti ketika proses reverse geocoding tidak berhasil	Sistem tetap memproses notifikasi menggunakan informasi lokasi yang tersedia	Sistem tetap memproses dan mengirimkan notifikasi menggunakan informasi lokasi yang tersedia	Berhasil

5.2.2.17. UC-17 Menerima Notifikasi Pelanggaran Geofencing

Tabel 5.22. Test case UC-17

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-17-01	Kendaraan keluar dari geofence	Menentukan geofence aktif kemudian mengirimkan data lokasi kendaraan di luar zona	Sistem mendeteksi pelanggaran dan mengirimkan notifikasi melalui channel yang dikonfigurasi	Sistem berhasil mendeteksi pelanggaran dan mengirimkan notifikasi melalui channel yang dikonfigurasi	Berhasil
TC-17-02	Kendaraan masih berada dalam geofence	Mengirimkan data lokasi kendaraan yang masih berada di dalam zona	Sistem tidak mengirimkan notifikasi pelanggaran	Sistem tidak mengirimkan notifikasi karena	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
				kendaraan masih berada dalam zona	
TC-17-03	Geofence dinonaktifkan	Menonaktifkan geofence kemudian mengirimkan lokasi kendaraan di luar area	Sistem tidak memproses pelanggaran terhadap geofence tersebut	Sistem tidak memproses pelanggaran karena geofence telah dinonaktifkan	Berhasil
TC-17-04	Salah satu channel notifikasi gagal	Membuat kondisi salah satu channel notifikasi tidak dapat digunakan ketika terjadi pelanggaran	Sistem mencatat status pengiriman channel tersebut	Sistem tetap mencatat status pengiriman pada channel yang gagal tanpa menghentikan channel lain	Berhasil

5.2.2.18. UC-18 Meninjau Notifikasi Kendaraan

Tabel 5.23. Test case UC-18

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-18-01	Melihat daftar notifikasi	Membuka panel Notifikasi pada Home	Sistem menampilkan daftar notifikasi kendaraan	Sistem berhasil menampilkan daftar notifikasi kendaraan	Berhasil
TC-18-02	Melihat detail notifikasi	Memilih salah satu notifikasi	Sistem menampilkan jenis notifikasi, kendaraan, waktu, dan informasi terkait	Sistem berhasil menampilkan detail notifikasi secara lengkap	Berhasil
TC-18-03	Melihat posisi kendaraan dari notifikasi	Membuka notifikasi yang memiliki tautan Lihat di Peta kemudian memilih tautan tersebut	Sistem mengarahkan pengguna ke posisi kendaraan pada peta	Sistem berhasil mengarahkan pengguna ke posisi kendaraan pada peta	Berhasil
TC-18-04	Tidak terdapat notifikasi	Membuka panel notifikasi ketika tidak ada data notifikasi	Sistem menampilkan informasi bahwa belum terdapat notifikasi	Sistem menampilkan informasi bahwa belum terdapat notifikasi	Berhasil
TC-18-05	Detail notifikasi gagal dimuat	Memilih notifikasi ketika detail tidak dapat dimuat	Sistem menampilkan pesan kesalahan	Detail notifikasi tetap berhasil dimuat pada seluruh pengujian yang dilakukan	Berhasil

5.2.2.19. UC-19 Melihat Dashboard Admin

Tabel 5.24. Test case UC-19

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-19-01	Melihat dashboard dengan data tersedia	Membuka halaman dashboard admin	Sistem menampilkan ringkasan jumlah pengguna, jumlah perangkat, dan status online/offline perangkat beserta grafik	Sistem berhasil menampilkan ringkasan dan grafik status perangkat	Berhasil
TC-19-02	Melihat dashboard saat belum ada data pengguna/perangkat	Membuka dashboard pada kondisi sistem belum memiliki data	Sistem menampilkan dashboard dengan ringkasan dan grafik bernilai kosong	Sistem berhasil menampilkan dashboard dengan ringkasan dan grafik bernilai kosong	Berhasil

5.2.2.20. UC-20 Menambahkan Perangkat GPS (Admin)

Tabel 5.25. Test case UC-20

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-20-01	Menambahkan perangkat dengan jumlah valid	Memasukkan jumlah perangkat kemudian menekan Tambah Perangkat	Sistem menghasilkan Device ID dan password baru sejumlah yang diminta, lalu menampilkan daftarnya	Sistem berhasil menghasilkan Device ID dan password baru sesuai jumlah yang diminta	Berhasil
TC-20-02	Menambahkan perangkat dengan jumlah tidak valid	Mengosongkan atau mengisi jumlah perangkat di luar rentang 1-100 kemudian menekan Tambah Perangkat	Sistem menampilkan pesan validasi	Sistem menampilkan pesan validasi karena format input hanya menerima angka dalam rentang yang valid	Berhasil

5.2.2.21. UC-21 Mengekspor Kredensial Perangkat ke PDF (Admin)

Tabel 5.26. Test case UC-21

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-21-01	Export PDF setelah penambahan perangkat berhasil	Menekan tombol Export PDF pada hasil penambahan perangkat	Dokumen PDF kredensial perangkat berhasil diunduh	Dokumen PDF kredensial perangkat berhasil diunduh	Berhasil
TC-21-02	Export PDF saat sesi cache kedaluwarsa	Menekan tombol Export PDF setelah kunci cache tidak valid/kedaluwarsa (lebih dari 10 menit)	Sistem menampilkan pesan bahwa sesi unduh kedaluwarsa dan dokumen tidak dapat dibuat	Sistem menampilkan pesan bahwa sesi unduh telah kedaluwarsa	Berhasil

5.2.2.22. UC-22 Menghapus Perangkat GPS (Admin)

Tabel 5.27. Test case UC-22

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-22-01	Menghapus perangkat yang belum digunakan	Memilih perangkat yang belum terikat akun pengguna, menekan Hapus, lalu mengonfirmasi	Perangkat terhapus dan daftar perangkat diperbarui	Perangkat berhasil terhapus dan daftar perangkat diperbarui	Berhasil
TC-22-02	Menghapus perangkat yang sudah digunakan pengguna	Memilih perangkat yang sudah terikat akun pengguna, menekan Hapus, lalu mengonfirmasi	Sistem menolak penghapusan dan menampilkan pesan bahwa perangkat tidak dapat dihapus	Sistem menolak penghapusan karena perangkat masih terikat dengan akun pengguna	Berhasil

5.2.2.23. UC-23 Meninjau Data Pengguna (Admin)

Tabel 5.28. Test case UC-23

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-23-01	Melihat daftar pengguna	Membuka halaman kelola pengguna	Daftar pengguna beserta perangkat dan kendaraan terkait ditampilkan	Sistem berhasil menampilkan daftar pengguna beserta perangkat dan kendaraan terkait	Berhasil

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-23-02	Mencari pengguna dengan kata kunci valid	Memasukkan nama atau email pengguna yang terdaftar pada kolom pencarian	Daftar pengguna tersaring sesuai kata kunci pencarian	Daftar pengguna berhasil tersaring sesuai kata kunci pencarian	Berhasil
TC-23-03	Mencari pengguna dengan kata kunci tidak ditemukan	Memasukkan kata kunci yang tidak cocok dengan pengguna manapun	Sistem menampilkan informasi bahwa pengguna tidak ditemukan	Sistem menampilkan informasi bahwa pengguna tidak ditemukan	Berhasil

5.2.2.24. UC-24 Menghapus Data Pengguna (Admin)

Tabel 5.29. Test case UC-24

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-24-01	Menghapus akun pengguna biasa	Memilih pengguna (bukan admin), menekan Hapus, lalu mengonfirmasi	Akun pengguna beserta keterkaitan perangkatnya terhapus, daftar pengguna diperbarui	Akun pengguna berhasil terhapus dan daftar pengguna diperbarui	Berhasil
TC-24-02	Mencoba menghapus akun admin	Memilih akun dengan status admin, menekan Hapus, lalu mengonfirmasi	Sistem menolak penghapusan dan menampilkan pesan bahwa akun admin tidak dapat dihapus	Sistem menolak penghapusan karena akun berstatus admin	Berhasil

5.2.2.25. UC-25 Mengatur Batas Kecepatan Kendaraan

Tabel 5.30. Test case UC-25

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-25-01	Mengatur batas kecepatan dengan nilai valid	Membuka pengaturan batas kecepatan pada Detail Kendaraan, memasukkan nilai valid (misal 80 km/j), kemudian menyimpan	Sistem menyimpan nilai batas kecepatan sebagai acuan deteksi overspeed	Nilai batas kecepatan berhasil tersimpan sesuai yang dimasukkan	Berhasil
TC-25-02	Memasukkan batas kecepatan tidak valid	Memasukkan nilai batas kecepatan yang tidak sesuai (kosong atau negatif) kemudian menyimpan	Sistem menampilkan pesan validasi	Sistem menampilkan pesan validasi untuk nilai yang tidak sesuai	Berhasil
TC-25-03	Mengaktifkan/menonaktifkan notifikasi overspeed	Mengaktifkan atau menonaktifkan toggle notifikasi overspeed kemudian menyimpan	Sistem menyimpan status notifikasi overspeed sebagai aktif/nonaktif	Status notifikasi overspeed berhasil tersimpan sesuai pilihan	Berhasil

5.2.2.26. UC-26 Menerima Notifikasi Kecepatan Berlebih

Tabel 5.31. Test case UC-26

Test Case ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-26-01	Kendaraan melebihi batas kecepatan	Mengaktifkan notifikasi overspeed, menentukan batas kecepatan, kemudian mengirimkan data kecepatan kendaraan yang melebihi batas tersebut	Sistem mendeteksi transisi ke kondisi overspeed dan mengirimkan notifikasi kepada pengguna melalui channel yang dikonfigurasi	Sistem berhasil mendeteksi transisi overspeed dan mengirimkan notifikasi	Berhasil
TC-26-02	Kecepatan kendaraan tetap di atas batas pada titik data berikutnya	Mengirimkan data lokasi lanjutan dengan kecepatan yang masih di atas batas setelah notifikasi pertama terkirim	Sistem tidak mengirimkan notifikasi berulang untuk episode overspeed yang sama	Sistem tidak mengirimkan notifikasi berulang selama kecepatan belum turun di bawah batas	Berhasil
TC-26-03	Kecepatan kembali normal	Mengirimkan data lokasi dengan kecepatan yang telah turun di bawah batas setelah kondisi overspeed	Sistem memperbarui status overspeed menjadi nonaktif tanpa mengirimkan notifikasi tambahan	Status overspeed kendaraan berhasil diperbarui menjadi nonaktif	Berhasil
TC-26-04	Notifikasi overspeed dinonaktifkan	Menonaktifkan notifikasi overspeed kemudian mengirimkan data kecepatan yang melebihi batas	Sistem tidak mengirimkan notifikasi overspeed	Sistem tidak mengirimkan notifikasi karena fitur telah dinonaktifkan	Berhasil

5.2.3 Hasil Pengujian Perangkat Keras (Hardware)

Pengujian perangkat keras dilakukan terhadap setiap modul fungsi firmware dengan mengamati keluaran pada Serial Monitor perangkat serta data yang diterima dan diproses oleh server (basis data Device, DeviceLog, TravelHistory, dan Notification). Hasil pengujian tiap modul disajikan pada tabel berikut.

5.2.3.1. HW-01 Inisialisasi Perangkat dan Modem SIM800L

Tabel 5.32. Test case HW-01

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-01-01	Modem merespons AT command saat baterai/saklar SIM800L dinyalakan	Menyalakan saklar daya SIM800L kemudian mengamati Serial Monitor saat boot	Serial Monitor menampilkan "Modem OK!" dan proses lanjut ke koneksi GPRS	Modem merespons dan proses lanjut ke tahap koneksi GPRS sesuai yang diharapkan	Berhasil
TC-HW-01-02	Modem tidak merespons saat hanya diberi daya lewat USB tanpa baterai	Menyalakan ESP32 hanya melalui kabel USB tanpa memasang/menyalakan baterai SIM800L	Serial Monitor menampilkan "Modem tidak merespons!" beserta instruksi	Serial Monitor menampilkan pesan modem tidak merespons	Berhasil

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
			memasang baterai dan menyalakan saklar	beserta instruksi yang sesuai	

5.2.3.2. HW-02 Koneksi Jaringan GPRS

Tabel 5.33. Test case HW-02

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-02-01	Koneksi GPRS berhasil dengan sinyal dan paket data tersedia	Menunggu proses waitForNetwork() dan gprsConnect(APN) pada saat boot	Serial menampilkan "GPRS BERHASIL" beserta IP Address yang diperoleh	GPRS berhasil terhubung dan IP Address tampil pada Serial Monitor	Berhasil
TC-HW-02-02	Koneksi GPRS gagal karena tidak ada sinyal jaringan	Menyalakan perangkat di lokasi tanpa sinyal GSM (area tertutup/basement)	Serial menampilkan "Menunggu jaringan... GAGAL" setelah 60 detik waktu tunggu (waitForNetwork timeout)	Serial Monitor menampilkan status gagal setelah waktu tunggu terlampaui	Berhasil
TC-HW-02-03	Koneksi GPRS gagal karena kuota data habis	Menyalakan perangkat dengan SIM card yang terdaftar jaringan namun kuota data habis	Serial menampilkan "GPRS GAGAL" setelah gprsConnect() ditolak	Serial Monitor menampilkan status GPRS gagal sesuai yang diharapkan	Berhasil

5.2.3.3. HW-03 Koneksi ke MQTT Broker

Tabel 5.34. Test case HW-03

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-03-01	Koneksi MQTT berhasil dengan username/password valid (firmware produksi)	Menjalankan connectMQTT() menggunakan kredensial device dari secrets.h	Serial menampilkan "MQTT CONNECTED!"	Koneksi MQTT berhasil dan status CONNECTED tampil pada Serial Monitor	Berhasil
TC-HW-03-02	Koneksi MQTT gagal dengan username/password tidak valid	Menjalankan connectMQTT() dengan MQTT_USERNAME/PASSWORD yang salah	Serial menampilkan "MQTT GAGAL, state = ..." (kode penolakan otentikasi dari broker)	Serial Monitor menampilkan kode kegagalan otentikasi sesuai yang diharapkan	Berhasil

5.2.3.4. HW-04 Akuisisi Sinyal GPS (GPS NEO-6M)

Tabel 5.35. Test case HW-04

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-04-01	GPS memperoleh fix lokasi dan waktu valid di area terbuka	Mengaktifkan perangkat di area outdoor dengan pandangan langit terbuka, menunggu proses akuisisi satelit	gps.location.isValid() dan gps.date/time bernilai valid; data lokasi siap dikirim ke server	Modul GPS berhasil memperoleh fix lokasi dan waktu yang valid di area terbuka	Berhasil
TC-HW-04-02	GPS belum memperoleh fix di area tertutup	Mengaktifkan perangkat di dalam ruangan beton/area tertutup	Serial menampilkan "GPS BELUM VALID"; fungsi publishGPS()	Sistem tidak mengirim data location sampai sinyal GPS valid,	Berhasil

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
			berhenti tanpa mengirim data	sesuai yang diharapkan	
TC-HW-04-03	Lokasi valid namun waktu GPS belum tersinkronisasi	Mengamati kondisi transisi saat modul baru memperoleh fix posisi tetapi data waktu satelit belum lengkap	Serial menampilkan "WAKTU GPS BELUM VALID"; data belum dikirim sampai waktu GPS valid	Sistem menahan pengiriman data sampai waktu GPS tersinkronisasi, sesuai yang diharapkan	Berhasil

5.2.3.5. HW-05 Pengiriman Data Lokasi dengan Tanda Tangan HMAC-SHA256

Tabel 5.36. Test case HW-05

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-05-01	Payload valid dengan signature benar diterima dan diproses server	Mengirim payload JSON lengkap (device_id, message_id, lat, lng, speed, heading, battery, satellite, received_at, signature) melalui MQTT publish	mqtt.publish() bernilai true ("MQTT PUBLISH BERHASIL!"); server memverifikasi signature dan memproses heartbeat, baterai, serta lokasi	Payload diterima dan diproses penuh oleh server sesuai yang diharapkan	Berhasil
TC-HW-05-02	Payload dengan signature tidak valid ditolak server	Mengirim payload dengan field signature yang diubah/acak (bukan hasil HMAC yang benar)	Server menolak dengan pesan "signature is invalid." melalui MQTTSignatureService::verify(), data tidak diproses	Server menolak payload dengan pesan signature invalid sesuai yang diharapkan	Berhasil
TC-HW-05-03	Payload tanpa field signature ditolak server	Mengirim payload JSON tanpa menyertakan field "signature"	Server menolak dengan pesan "signature is required."	Server menolak payload tanpa field signature sesuai yang diharapkan	Berhasil
TC-HW-05-04	Payload dengan nilai numerik di luar rentang valid ditolak	Mengirim payload dengan salah satu field numerik melewati batas yang diizinkan GPSProcessingService::validatePayload()	Server menolak dengan pesan sesuai field, contoh "lat is out of range." atau "battery is out of range."	Server menolak payload dengan nilai di luar rentang sesuai yang diharapkan	Berhasil
TC-HW-05-05	Payload dengan received_at di luar jendela waktu yang diperbolehkan ditolak	Mengirim payload dengan field received_at jauh berbeda dari waktu server	Server menolak dengan pesan "received_at is outside the acceptable time window." (lebih dari 1 hari lampau atau lebih dari 5 menit ke depan)	Server menolak payload dengan received_at di luar jendela waktu sesuai yang diharapkan	Berhasil

5.2.3.6. HW-06 Penyaringan Perpindahan Posisi (Position Threshold)

Tabel 5.37. Test case HW-06

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-06-01	Perpindahan posisi $\geq 0,5$ meter tersimpan	Mengirim dua payload berurutan dengan titik	DeviceLog dan TravelHistory baru	Data riwayat baru tersimpan sesuai yang diharapkan	Berhasil

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
	sebagai riwayat baru	koordinat berjarak $\geq 0,5$ meter	tersimpan pada payload kedua		
TC-HW-06-02	Perpindahan posisi $< 0,5$ meter (perangkat diam) tidak menambah riwayat baru	Mengirim dua payload berurutan dengan titik koordinat hampir sama (perangkat diam di tempat)	DeviceLog/TravelHistory tidak bertambah, namun heartbeat dan baterai tetap diperbarui	Riwayat tidak bertambah namun heartbeat dan baterai tetap diperbarui, sesuai yang diharapkan	Berhasil

5.2.3.7. HW-07 Deteksi Pesan Duplikat (Message ID)

Tabel 5.38. Test case HW-07

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-07-01	Payload dengan message_id baru diproses secara normal	Mengirim payload dengan message_id yang belum pernah diterima sebelumnya untuk device tersebut	Payload diproses penuh (heartbeat, baterai, riwayat, notifikasi jika relevan)	Payload diproses penuh sesuai yang diharapkan	Berhasil
TC-HW-07-02	Payload dengan message_id yang sama (duplikat) diabaikan	Mengirim ulang payload dengan message_id identik dengan payload sebelumnya (mis. akibat publish ulang jaringan tidak stabil)	Server tidak memproses ulang riwayat/notifikasi baru (DeviceLogService::isDuplicate() bernilai true)	Server mengabaikan payload duplikat sesuai yang diharapkan	Berhasil

5.2.3.8. HW-08 Deteksi Baterai Lemah

Tabel 5.39. Test case HW-08

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-08-01	Baterai turun ke $\leq 20\%$ memicu notifikasi baterai lemah	Mengirim payload dengan battery = 20 setelah sebelumnya berada di atas ambang batas	Notifikasi baterai lemah dibuat dan status low_battery_active menjadi true	Notifikasi baterai lemah berhasil dibuat sesuai yang diharapkan	Berhasil
TC-HW-08-02	Baterai di atas ambang batas tidak memicu notifikasi berulang	Mengirim payload dengan battery = 45 pada kondisi status low_battery_active masih false	Tidak ada notifikasi baterai lemah baru yang dibuat	Tidak ada notifikasi baru yang dibuat sesuai yang diharapkan	Berhasil

5.2.3.9. HW-09 Reconnect Otomatis saat Koneksi Terputus

Tabel 5.40. Test case HW-09

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-09-01	Koneksi MQTT terputus sementara GPRS masih aktif, perangkat reconnect MQTT otomatis	Menghentikan/merestart broker MQTT sesaat kemudian mengaktifkannya kembali, mengamati loop()	Serial menampilkan "MQTT TERPUTUS" lalu "MQTT berhasil reconnect!" tanpa perlu restart perangkat	Perangkat berhasil reconnect MQTT secara otomatis tanpa restart manual	Berhasil

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-09-02	Koneksi GPRS terputus (sinyal hilang) lalu kembali, perangkat mencoba reconnect GPRS	Membawa perangkat ke area tanpa sinyal sesaat lalu kembali ke area bersinyal	Serial menampilkan "GPRS terputus. Mencoba reconnect GPRS..." lalu berhasil terhubung kembali dan reconnect MQTT	Perangkat berhasil reconnect GPRS dan MQTT secara otomatis	Berhasil
TC-HW-09-03	Reconnect gagal berulang saat tidak ada sinyal dalam waktu lama	Membiarkan perangkat tetap berada di area tanpa sinyal dalam waktu lama	Serial menampilkan "Reconnect GPRS gagal." berulang dengan jeda 5 detik antar percobaan, perangkat tidak crash/hang	Perangkat tetap mencoba reconnect tanpa crash/hang sesuai yang diharapkan	Berhasil

5.2.3.10. HW-10 Pengiriman Data Berkala (Publish Interval)

Tabel 5.41. Test case HW-10

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-10-01	Data terkirim setiap kurang lebih 10 detik pada firmware produksi	Mengamati Serial Monitor selama 1 menit dan menghitung jumlah proses publishGPS()	Payload terkirim setiap kurang lebih 10 detik sesuai PUBLISH_INTERVAL = 10000 ms pada afgps_prod.ino	Payload terkirim secara berkala kurang lebih setiap 10 detik sesuai yang diharapkan	Berhasil

5.2.3.11. HW-11 Deteksi Perangkat Offline/Online (Heartbeat)

Tabel 5.42. Test case HW-11

ID Uji	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-HW-11-01	Perangkat berhenti mengirim data > 5 menit ditandai offline dan notifikasi terkirim	Mematikan perangkat, menunggu lebih dari 5 menit, kemudian menjalankan perintah device:health-check	Status perangkat menjadi tidak aktif (is_online=false) sesuai offline_threshold_minutes=5 pada config/mqtt.php; notifikasi perangkat offline terkirim	Perangkat ditandai offline dan notifikasi terkirim sesuai yang diharapkan	Berhasil
TC-HW-11-02	Perangkat kembali online setelah sebelumnya offline	Menyalakan kembali perangkat yang sebelumnya berstatus offline_notified_at terisi	Notifikasi perangkat kembali online dibuat oleh sistem	Notifikasi perangkat kembali online berhasil dibuat sesuai yang diharapkan	Berhasil

5.2.4 Rangkuman Hasil Pengujian

Tabel berikut merangkum hasil pengujian black-box yang telah dilakukan terhadap seluruh use case, dikelompokkan berdasarkan Business Use Case (BUC) sebagaimana telah dipetakan pada Bab IV.

Tabel 5.43. Rangkuman Hasil Pengujian Perangkat Lunak

Kode	Modul Pengujian (Business Use Case)	Jumlah Kasus Uji	Berhasil	Gagal	Persentase Keberhasilan
BUC-01	Mengelola Data Kendaraan dan Perangkat (UC-01, UC-02, UC-03, UC-04)	11	11	0	100%
BUC-02	Memantau Kendaraan secara Real-time (UC-05, UC-06)	7	7	0	100%
BUC-03	Mengelola Zona Geofencing (UC-07, UC-08)	10	10	0	100%
BUC-04	Mengelola Home Location (UC-09)	4	4	0	100%
BUC-05	Meninjau dan Mengekspor Riwayat Kendaraan (UC-10, UC-11, UC-12, UC-13)	14	14	0	100%
BUC-06	Mengelola Notifikasi Kendaraan (UC-14, UC-15, UC-16, UC-17, UC-18, UC-25, UC-26)	27	27	0	100%
BUC-07	Mengelola Sistem melalui Panel Admin (UC-19, UC-20, UC-21, UC-22, UC-23, UC-24)	13	13	0	100%
Total		86	86	0	100%

Berdasarkan tabel rangkuman di atas, seluruh 86 kasus uji black-box yang dilaksanakan pada sistem AF GPS Tracker dinyatakan berhasil (100%), termasuk kasus uji tambahan TC-05-05 (tampilan kecepatan real-time) serta TC-25-01 sampai TC-26-04 (pengaturan dan notifikasi batas kecepatan/overspeed). Hasil ini menunjukkan bahwa seluruh fungsi yang telah dianalisis pada Bab III dan dirancang pada Bab IV telah berjalan sesuai dengan yang diharapkan pada lingkungan produksi.

Tabel berikut merangkum hasil pengujian black-box perangkat keras (firmware), dikelompokkan berdasarkan modul fungsi firmware sebagaimana disajikan pada sub-bab Rencana Pengujian.

Tabel 5.44. Rangkuman Pengujian Perangkat Keras

Kode Modul	Nama Modul Firmware	Jumlah Kasus Uji	Berhasil	Gagal	Persentase Keberhasilan
HW-01	Inisialisasi Perangkat dan Modem SIM800L	2	2	0	100%
HW-02	Koneksi Jaringan GPRS	3	3	0	100%
HW-03	Koneksi ke MQTT Broker	2	2	0	100%
HW-04	Akuisisi Sinyal GPS (GPS NEO-6M)	3	3	0	100%
HW-05	Pengiriman Data Lokasi dengan Tanda Tangan HMAC-SHA256	5	5	0	100%
HW-06	Penyaringan Perpindahan Posisi (Position Threshold)	2	2	0	100%
HW-07	Deteksi Pesan Duplikat (Message ID)	2	2	0	100%
HW-08	Deteksi Baterai Lemah	2	2	0	100%

Kode Modul	Nama Modul Firmware	Jumlah Kasus Uji	Berhasil	Gagal	Persentase Keberhasilan
HW-09	Reconnect Otomatis saat Koneksi Terputus	3	3	0	100%
HW-10	Pengiriman Data Berkala (Publish Interval)	1	1	0	100%
HW-11	Deteksi Perangkat Offline/Online (Heartbeat)	2	2	0	100%
Total		27	27	0	100%

Berdasarkan tabel rangkuman di atas, seluruh 27 kasus uji black-box perangkat keras yang dilaksanakan pada firmware GPS Tracker dinyatakan berhasil (100%). Dengan demikian, seluruh 27 kasus uji perangkat keras bersama dengan 86 kasus uji perangkat lunak yang telah dipaparkan sebelumnya (total 113 kasus uji) menunjukkan bahwa seluruh fungsi utama sistem AF GPS Tracker, baik pada sisi perangkat keras maupun perangkat lunak, telah berjalan sesuai dengan kebutuhan yang telah dianalisis pada Bab III dan dirancang pada Bab IV.

5.2.5 Pengujian User Acceptance Testing (UAT)

User Acceptance Testing (UAT) dilakukan untuk memvalidasi kelayakan sistem AF GPS Tracker dari sudut pandang pengguna akhir setelah seluruh rangkaian pengujian black-box selesai dilaksanakan. Pengujian ini menggunakan Skala Guttman dengan dua pilihan jawaban, yaitu "Ya" dan "Tidak", yang diisi oleh responden melalui kuesioner setelah responden mencoba secara langsung fitur-fitur yang terdapat pada sistem. UAT dilakukan terhadap dua kelompok pengguna, yaitu kelompok Pengguna/Pemilik Kendaraan dan kelompok Admin, dengan pernyataan pada setiap kelompok disusun berdasarkan fungsi-fungsi yang telah diuji pada pengujian black-box sebelumnya.

Pernyataan-pernyataan berikut digunakan untuk menilai kelayakan sistem dari sudut pandang kelompok Pengguna/Pemilik Kendaraan, sebagaimana ditunjukkan pada tabel berikut.

Tabel 5.45. Pengujian UAT Pengguna

No	Pernyataan	Ya	Tidak	Keterangan
1	Pengguna dapat menambahkan data kendaraan baru dengan mudah.	✓		
2	Pengguna dapat menambahkan perangkat GPS ke kendaraan dengan mudah.	✓		
3	Pengguna dapat mengubah data kendaraan sesuai kebutuhan dengan mudah.	✓		
4	Pengguna dapat menghapus data kendaraan yang tidak lagi digunakan dengan mudah.	✓		

No	Pernyataan	Ya	Tidak	Keterangan
5	Sistem menampilkan posisi kendaraan secara real-time dengan jelas pada peta.	✓		
6	Pengguna dapat melihat status aktif atau tidak aktifnya perangkat GPS dengan mudah.	✓		
7	Fitur pengaturan zona geofencing (administratif, radius, maupun polygon) mudah digunakan.	✓		
8	Pengguna dapat mengatur channel notifikasi pelanggaran geofencing (email dan/atau WhatsApp) dengan mudah.	✓		
9	Fitur Home Location membantu pengguna menentukan titik acuan lokasi kendaraan dengan mudah.	✓		
10	Pengguna dapat meninjau riwayat perjalanan kendaraan beserta fitur playback dengan mudah.	✓		
11	Pengguna dapat meninjau riwayat berhenti kendaraan dengan mudah.	✓		
12	Pengguna dapat mengekspor riwayat perjalanan kendaraan ke dalam bentuk PDF dengan mudah.	✓		
13	Pengguna dapat mengekspor riwayat berhenti kendaraan ke dalam bentuk PDF dengan mudah.	✓		
14	Pengguna dapat mengatur sendiri durasi waktu berhenti sebelum notifikasi dikirimkan.	✓		
15	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi berhenti sesuai kebutuhan.	✓		
16	Sistem mengirimkan notifikasi berhenti beserta alamat lengkap kendaraan dengan akurat.	✓		
17	Sistem mengirimkan notifikasi pelanggaran geofencing dengan tepat ketika kendaraan keluar dari zona yang ditentukan.	✓		
18	Pengguna dapat meninjau seluruh notifikasi kendaraan pada satu halaman dengan mudah.	✓		
19	Pengguna dapat mengatur batas kecepatan kendaraan sesuai kebutuhan dengan mudah.	✓		

No	Pernyataan	Ya	Tidak	Keterangan
20	Sistem mengirimkan notifikasi ketika kendaraan melaju melebihi batas kecepatan yang telah ditentukan.	✓		
21	Secara keseluruhan, sistem AF GPS Tracker membantu proses pemantauan kendaraan secara efektif.	✓		

Pernyataan-pernyataan berikut digunakan untuk menilai kelayakan sistem dari sudut pandang kelompok Admin, sebagaimana ditunjukkan pada tabel berikut.

Tabel 5.46. Pengujian UAT Admin

No	Pernyataan	Ya	Tidak	Keterangan
1	Admin dapat memantau ringkasan jumlah perangkat GPS dan pengguna terdaftar dengan mudah melalui dashboard.	✓		
2	Admin dapat menambahkan perangkat GPS baru beserta kredensialnya dengan mudah.	✓		
3	Admin dapat mengekspor kredensial perangkat GPS yang baru ditambahkan ke dalam bentuk PDF dengan mudah.	✓		
4	Admin dapat menghapus perangkat GPS yang sudah tidak digunakan dengan mudah.	✓		
5	Admin dapat meninjau data pengguna sistem dengan mudah.	✓		
6	Admin dapat menghapus data pengguna sistem dengan mudah apabila diperlukan.	✓		
7	Secara keseluruhan, panel admin membantu proses pengelolaan sistem AF GPS Tracker secara efektif.	✓		

5.2.5.1. Rangkuman Hasil User Acceptance Testing (UAT)

Tabel berikut menyajikan rangkuman hasil pengujian UAT yang telah dilaksanakan terhadap kedua kelompok responden.

Tabel 5.47. Rangkuman Hasil UAT

No	Kelompok Penguji	Jumlah Item Pernyataan	Ya	Tidak	Persentase
1	Pengguna/Pemilik Kendaraan	21	21	0	100%
2	Admin	7	7	0	100%
Total		28	28	0	100%

Berdasarkan tabel rangkuman di atas, seluruh responden dari kelompok Pengguna/Pemilik Kendaraan maupun Admin menjawab "Ya" pada seluruh pernyataan yang diujikan, sehingga diperoleh persentase keberhasilan sebesar 100% pada kedua kelompok responden.

BAB 6

KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan yang diperoleh dari hasil perancangan, pembangunan, dan pengujian sistem AF GPS Tracker, termasuk hasil pengujian lapangan yang dilakukan pada kondisi penggunaan sebenarnya, beserta saran yang dapat menjadi bahan pertimbangan bagi pengembangan penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan hasil perancangan, pembangunan, dan pengujian sistem AF GPS Tracker yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Sistem pemantauan lokasi kendaraan bermotor secara real-time telah berhasil dibangun sebagai upaya pencegahan pencurian. Perangkat GPS tracker memperoleh koordinat kendaraan melalui modul GPS NEO-6M dan mengirimkannya ke server setiap kurang lebih 10 detik melalui jaringan GPRS dan protokol MQTT, sehingga posisi kendaraan yang sebelumnya hanya dapat diketahui setelah kendaraan berpindah lokasi kini dapat ditampilkan pada peta secara langsung dan terus diperbarui. Setiap payload dilindungi tanda tangan digital HMAC-SHA256 dan divalidasi di sisi server, sedangkan penyaringan perpindahan posisi serta deteksi pesan duplikat menjaga agar data yang diproses benar-benar mencerminkan pergerakan kendaraan. Sistem juga menampilkan status aktif atau tidak aktifnya perangkat dan mengirimkan peringatan dini berupa notifikasi pelanggaran geofencing, kecepatan berlebih, kendaraan berhenti melebihi durasi, baterai lemah, serta perangkat offline dan kembali online melalui email, WhatsApp, dan notifikasi dalam aplikasi, sehingga tindakan pencegahan dapat dilakukan sebelum kendaraan berpindah jauh dari lokasi semula.
2. Sistem pelacakan kendaraan bermotor berbasis GPS telah berhasil dibangun untuk memudahkan proses pencarian kendaraan yang dicuri. Data koordinat beserta waktu penerimaannya tersimpan sebagai riwayat perjalanan dan riwayat berhenti yang dapat ditinjau per kendaraan dan per periode waktu, dilengkapi fitur playback pergerakan kendaraan mengikuti rute, informasi lokasi dan waktu berhenti, serta ekspor ke dalam dokumen PDF. Ketersediaan riwayat tersebut membuat proses pencarian kendaraan yang sebelumnya bergantung pada perkiraan dan ingatan pemilik kini dapat dilakukan dengan menelusuri kembali jalur perjalanan kendaraan, mengetahui titik berhenti terakhir, dan memperkirakan arah perpindahannya. Informasi tersebut juga dapat dijadikan bukti pendukung ketika kejadian dilaporkan kepada pihak berwenang.

3. Fitur pembatasan wilayah administratif (geofencing) telah berhasil dibangun sebagai mekanisme peringatan dini. Sistem menyediakan tiga jenis zona, yaitu zona wilayah administratif Indonesia (provinsi, kabupaten/kota, kecamatan, dan kelurahan) yang bersumber dari data GeoJSON, zona radius, serta zona polygon yang digambar langsung oleh pengguna. Penentuan posisi kendaraan terhadap zona administratif dan polygon dihitung menggunakan algoritma Point in Polygon, sedangkan zona radius dihitung menggunakan rumus Haversine, dengan data masukan berupa koordinat berakurasi 2 hingga 5 meter dari perangkat. Setiap zona dapat diaktifkan atau dinonaktifkan dan dilengkapi konfigurasi kanal notifikasi, sehingga peringatan yang sebelumnya hanya dapat diberikan setelah pengguna menyadari sendiri perpindahan kendaraan kini terkirim otomatis melalui email dan/atau WhatsApp beserta alamat lokasi yang mudah dipahami. Kegagalan pada satu kanal tidak menghentikan pengiriman kanal lainnya, dan seluruh fungsi pengelolaan zona beserta konfigurasi notifikasi geofencing berhasil dijalankan pada 10 kasus uji black-box dengan tingkat keberhasilan 100%.
4. Perangkat pelacak kendaraan yang bersifat portabel dan tidak merusak (non-invasive) telah berhasil dirancang. Perangkat dibangun menggunakan mikrokontroler ESP32, modul GPS NEO-6M dengan antena eksternal aktif, dan modul komunikasi SIM800L, dengan sumber daya mandiri berupa dua sel baterai lithium-ion tipe 18650 berkapasitas 3000 mAh yang dirangkai paralel sehingga menghasilkan kapasitas total 6000 mAh pada tegangan nominal 3,7 V. Pengisian daya menggunakan modul HW-373 berbasis IC TP4056 dengan konektor USB Type-C serta proteksi over-charge, over-discharge, dan short-circuit, sedangkan penstabil tegangan menggunakan modul HT7333. Seluruh komponen dirangkai secara point-to-point pada satu papan tembaga yang sekaligus berfungsi sebagaiudukan, dilengkapi saklar daya utama dan lapisan isolasi, sehingga perangkat yang sebelumnya menuntut pemasangan invasif pada sistem kelistrikan kendaraan dan karenanya mudah dikenali kini dapat beroperasi mandiri, dipasang tanpa memotong atau mengubah jalur kelistrikan kendaraan, mudah dipindahkan antar kendaraan, serta dapat ditempatkan secara tersembunyi. Keberhasilan tersebut dibuktikan oleh 27 kasus uji black-box perangkat keras dengan tingkat keberhasilan 100%.
5. Perangkat pelacak kendaraan yang mudah digunakan, terjangkau, dan dapat mendorong adopsi yang lebih luas di masyarakat telah berhasil dihasilkan. Keterjangkauan didukung oleh penggunaan komponen yang umum tersedia di pasaran tanpa komponen khusus maupun rangkaian tambahan, sedangkan kemudahan penggunaan diwujudkan melalui pengisian daya dengan konektor USB

Type-C yang lazim digunakan, saklar daya utama yang sederhana, pemasangan yang tidak memerlukan keahlian teknis khusus, serta seluruh fungsi pemantauan yang disediakan dalam bentuk aplikasi berbasis web yang dapat diakses melalui browser pada berbagai perangkat tanpa proses instalasi. Dengan demikian, sistem yang sebelumnya menuntut perangkat khusus dan pemasangan rumit kini dapat digunakan oleh masyarakat umum hanya dengan mengaktifkan perangkat melalui website. Kelayakan tersebut dibuktikan melalui pengujian black-box terhadap 86 kasus uji perangkat lunak dan 27 kasus uji perangkat keras (total 113 kasus uji) yang seluruhnya berhasil dengan tingkat keberhasilan 100%, serta User Acceptance Testing terhadap 28 pernyataan yang seluruhnya dijawab "Ya" (100%).

6.2 Saran

Berdasarkan hasil penelitian dan pengembangan sistem AF GPS Tracker yang telah dilakukan, terdapat beberapa aspek yang dapat dikembangkan pada penelitian selanjutnya, yaitu sebagai berikut:

1. Pengembangan ketahanan terhadap kondisi lingkungan. Perangkat GPS Tracker pada penelitian ini belum diuji secara khusus terhadap ketahanan air dan kondisi cuaca. Oleh karena itu, penelitian selanjutnya dapat mengembangkan desain casing dan sistem perlindungan perangkat agar memiliki tingkat ketahanan terhadap air yang lebih baik, sehingga perangkat dapat digunakan pada berbagai kondisi cuaca dan lingkungan.
2. Optimalisasi penggunaan baterai. Penggunaan baterai 18650 pada perangkat GPS Tracker masih menjadi salah satu aspek yang dapat dikembangkan, khususnya dari sisi ukuran dan efisiensi ruang pada perangkat portabel. Penelitian selanjutnya dapat mempertimbangkan penggunaan jenis baterai dengan ukuran yang lebih kecil namun tetap memiliki kapasitas dan kemampuan arus yang sesuai dengan kebutuhan perangkat, serta melakukan pengujian ketahanan baterai secara terukur untuk mengetahui lama operasi perangkat pada kondisi penggunaan sebenarnya. Hal tersebut diharapkan dapat mengurangi ukuran perangkat dan membuat desain GPS Tracker menjadi lebih ringkas serta efisien.
3. Penyederhanaan dan integrasi komponen perangkat keras berbasis PCB kustom. Perangkat GPS Tracker pada penelitian ini masih menggunakan modul-modul yang terpisah, seperti mikrokontroler, modul GPS, dan modul GSM, serta dirakit secara point-to-point menggunakan kabel penghubung pada satu papan tembaga (copper board). Oleh karena itu, penelitian selanjutnya dapat merancang Printed Circuit Board (PCB) kustom serta menggunakan komponen atau chip yang lebih terintegrasi (all-in-one). Hal ini diharapkan dapat memperkecil dimensi fisik

perangkat, mengurangi kompleksitas rangkaian, meningkatkan kestabilan koneksi perangkat keras, serta menekan biaya produksi apabila akan dikembangkan ke dalam skala industri atau manufaktur massal.

4. Pengembangan aplikasi mobile sebagai media pemantauan. Sistem pemantauan pada penelitian ini masih diakses melalui aplikasi berbasis web. Oleh karena itu, penelitian selanjutnya dapat mengembangkan aplikasi mobile pada perangkat Android dan iOS agar pemantauan lokasi kendaraan dapat dilakukan lebih praktis, notifikasi dapat diterima secara langsung melalui push notification tanpa bergantung pada email dan WhatsApp, serta perangkat dapat memanfaatkan fitur bawaan telepon pintar seperti pemantauan di latar belakang dan widget informasi status kendaraan.
5. Penambahan fitur baru pada perangkat lunak. Fitur pemantauan pada penelitian ini masih terbatas pada penyajian informasi lokasi, geofencing, riwayat perjalanan, dan notifikasi, sehingga belum mencakup tindakan lanjutan terhadap kendaraan maupun integrasi dengan pihak eksternal. Penelitian selanjutnya dapat menambahkan fitur baru, seperti kontrol perangkat dari jarak jauh untuk mengaktifkan mode hemat daya dan mengatur interval pengiriman data, berbagi akses pemantauan kendaraan kepada beberapa pengguna, serta integrasi pelaporan dengan instansi berwenang. Penambahan fitur tersebut diharapkan dapat meningkatkan nilai guna sistem dalam penanganan tindak pencurian kendaraan bermotor.

DAFTAR PUSTAKA

- [AGA11] V. Agafonkin, "Leaflet — An Open-Source JavaScript Library for Interactive Maps." Accessed: 2023. [Online]. Available: <https://leafletjs.com>
- [ALI24] Aliyah, N. Hartono, and A. A. Muin, "Penggunaan User Acceptance Testing (UAT) pada pengujian sistem informasi pengelolaan keuangan dan inventaris barang," *SWITCH: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 2, pp. 42–58, Mar. 2025, doi: 10.62951/switch.v3i1.330.
- [ASH22] M. Y. Ashadi, S. Ariyani, B. S. Rintyarna, and N. K. Wardati, "Desain sistem keamanan sepeda motor dengan memanfaatkan GPS Tracker berbasis IoT," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 4, no. 2, pp. 152–159, 2022.
- [BAH14] A. Bahga and V. Madiseti, *Internet of Things: A Hands-On Approach*. VPT, 2014.
- [DET26] Tim detikcom, "Terungkap ribuan motor di Jaksel bakal ‘dimutilasi’ diekspor ke Afrika," *detikNews*. Accessed: May 12, 2026. [Online]. Available: <https://news.detik.com/berita/d-8485153/terungkap-ribuan-motor-di-jaksel-bakal-dimutilasi-diekspor-ke-afrika>
- [ESP21] Espressif Systems, *ESP32 Series Datasheet*, version 3.6. Shanghai: Espressif Systems, 2021.
- [FET11] I. Fette and A. Melnikov, "The WebSocket Protocol," RFC 6455, Internet Engineering Task Force, 2011, doi: 10.17487/RFC6455.
- [FON23] Fonnte, *Fonnte API Documentation*. 2023. [Online]. Available: <https://fonnte.com/docs>
- [GOO23] OpenStreetMap Foundation, *Nominatim Documentation*. 2023. [Online]. Available: <https://nominatim.org/release-docs/latest/>
- [HAR25] A. D. Harahap, Indarto, and F. Humam, "Rancang bangun sistem monitoring dan kontrol jarak jauh pada rental motor berbasis IoT dengan geofencing," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, pp. 8086–8093, 2025.
- [KAP96] E. D. Kaplan, Ed., *Understanding GPS: Principles and Applications*. Boston: Artech House, 1996.
- [KLE01] J. Klensin, "Simple Mail Transfer Protocol," RFC 2821, Internet Engineering Task Force, 2001.
- [KUP11] A. Küpper, U. Bareth, and B. Freese, "Geofencing and Background Tracking — The Next Features in LBSs," in *INFORMATIK 2011 – Informatik schafft Communities*, 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Bonn, 2011.
- [MUN24] G. Munawar, "Sistem monitoring kendaraan logistik secara real-time berbasis Internet of Things (IoT)," *Indonesian Journal of Computer Science and Engineering (IJCSE)*, vol. 1, no. 2, pp. 20–26, 2024, doi: 10.70656/ijcse.v1i02.118.
- [OAS14] A. Banks and R. Gupta, Eds., *MQTT Version 3.1.1*, OASIS Standard. Burlington: OASIS Open, 2014. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
- [ORO98] J. O'Rourke, *Computational Geometry in C*, 2nd ed. Cambridge: Cambridge University Press, 1998.
- [OSM23] OpenStreetMap Foundation, *OpenStreetMap Wiki*. 2023. [Online]. Available: <https://wiki.openstreetmap.org>
- [OTW23] T. Otwell, *Laravel Documentation*, version 10.x. 2023. [Online]. Available: <https://laravel.com/docs/10.x>

- [PRS20] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York: McGraw-Hill Education, 2020.
- [PUS23] Pusher Ltd., *Pusher Channels Documentation*. London: Pusher Ltd., 2023. [Online]. Available: <https://pusher.com/docs/channels>
- [PUT25] D. F. Putri and D. Widhiantoro, "Implementasi metode geofencing untuk sistem pelacak lokasi: Studi literatur sistematis," in *Prosiding Seminar Nasional Inovasi Vokasi (SNIV)*, vol. 4, pp. 785–791, 2025.
- [RAH19] A. C. Rahman, I. W. A. Arimbawa, and A. H. Jatmika, "Implementasi Internet of Things pada sistem informasi pelacakan kendaraan bermotor menggunakan GPS berbasis web," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, vol. 1, no. 1, pp. 121–130, 2019, doi: 10.29303/jtika.v1i1.10.
- [RAM03] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. New York: McGraw-Hill, 2003.
- [RFC16] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, "The GeoJSON Format," RFC 7946, Internet Engineering Task Force, 2016, doi: 10.17487/RFC7946.
- [SIM13] SIMCom Wireless Solutions, *SIM800L Hardware Design*, version 1.09. Shanghai: SIMCom Wireless Solutions Co., Ltd., 2013.
- [SOM16] I. Sommerville, *Software Engineering*, 10th ed. Boston: Pearson Education, 2016.
- [UBL10] u-blox AG, *NEO-6 u-blox 6 GPS Modules Data Sheet*, GPS.G6-HW-09005-E. Thalwil: u-blox AG, 2010.
- [UMI22] Uminingsih, M. N. Ichsanudin, M. Yusuf, and S. Suraya, "Pengujian fungsional perangkat lunak sistem informasi perpustakaan dengan metode black box testing bagi pemula," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.

LAMPIRAN

A. UAT Admin

FORMULIR PENGUJIAN PENERIMAAN PENGGUNA (USER ACCEPTANCE TESTING / UAT) UNTUK ADMIN

A. INFORMASI DOKUMEN

Nomor Dokumen	UAT-AFGPS-ADMIN-001
Nama Sistem	AF GPS Tracker – Sistem Pemantauan Kendaraan Berbasis Web
Versi Sistem	1.0.0
Tanggal Pengujian	24 September 2026
Penguji (Admin)	Aldi Perdana Kham

B. PETUNJUK PENGISIAN

1. Bacalah setiap pernyataan pengujian dengan seksama.
2. Lakukan pengujian langsung pada sistem AF GPS Tracker sesuai pernyataan yang tertera.
3. Beri tanda centang (✓) pada kolom "Ya" jika fitur berfungsi sesuai pernyataan, atau pada kolom "Tidak" jika tidak berfungsi.
4. Isi kolom Catatan jika terdapat temuan, kendala, atau masukan terhadap fitur yang diuji.
5. Isi kesimpulan dan tanda tangan pada bagian akhir formulir.

C. DAFTAR PERNYATAAN PENGUJIAN

No	Fitur yang Diuji	Pernyataan Admin	Ya	Tidak	Catatan
1	Dashboard Admin	Admin dapat memantau ringkasan jumlah perangkat GPS dan pengguna terdaftar dengan mudah melalui dashboard.	✓		
2	Pengelolaan Perangkat GPS	Admin dapat menambahkan perangkat GPS baru beserta kredensialnya dengan mudah.	✓		
3	Ekspor Kredensial Perangkat	Admin dapat mengekspor kredensial perangkat GPS yang baru ditambahkan ke dalam bentuk PDF dengan mudah.	✓		
4	Penghapusan Perangkat GPS	Admin dapat menghapus perangkat GPS yang sudah tidak digunakan dengan mudah.	✓		
5	Data Pengguna	Admin dapat meninjau data pengguna sistem dengan mudah.	✓		
6	Penghapusan Data Pengguna	Admin dapat menghapus data pengguna sistem dengan mudah apabila diperlukan.	✓		
7	Kesesuaian Panel Admin secara Keseluruhan	Secara keseluruhan, panel admin membantu proses pengelolaan sistem AF GPS Tracker secara efektif.	✓		

D. KESIMPULAN PENGUJIAN

Jumlah Item Diuji	7 Item
Jumlah Jawaban "Ya"	7 Item
Jumlah Jawaban "Tidak"	0 Item
Persentase Penerimaan	100%

Kesimpulan

- ☒ Sistem DITERIMA dan layak digunakan
☐ Sistem DITERIMA dengan catatan perbaikan
☐ Sistem DITOLAK dan perlu perbaikan menyeluruh

Catatan Tambahan

.....

.....

E. PERNYATAAN DAN TANDA TANGAN

Saya yang bertanda tangan di bawah ini menyatakan bahwa pengujian telah dilakukan secara langsung dan hasil yang tercatat dalam formulir ini adalah benar adanya.

Bandung, 24 Januari 2026

Penguji



(.....nkq.....)

B. UAT Pengguna

FORMULIR PENGUJIAN PENERIMAAN PENGGUNA (USER ACCEPTANCE TESTING / UAT) UNTUK PENGGUNA

A. INFORMASI DOKUMEN

Nomor Dokumen	UAT-AFGPS-PENGGUNA-001
Nama Sistem	AF GPS Tracker – Sistem Pemantauan Kendaraan Berbasis Web
Versi Sistem	1.0.0
Tanggal Pengujian	24 September 2026
Penguji (Pengguna/Pemilik Kendaraan)	Hedi Sukur

B. PETUNJUK PENGISIAN

1. Bacalah setiap pernyataan pengujian dengan seksama.
2. Lakukan pengujian langsung pada sistem AF GPS Tracker sesuai pernyataan yang tertera.
3. Beri tanda centang (✓) pada kolom "Ya" jika fitur berfungsi sesuai pernyataan, atau pada kolom "Tidak" jika tidak berfungsi.
4. Isi kolom Catatan jika terdapat temuan, kendala, atau masukan terhadap fitur yang diuji.
5. Isi kesimpulan dan tanda tangan pada bagian akhir formulir.

C. DAFTAR PERNYATAAN PENGUJIAN

No	Fitur yang Diuji	Pernyataan Pengguna	Ya	Tidak	Catatan
1	Menambahkan Data Kendaraan	Pengguna dapat menambahkan data kendaraan baru dengan mudah.	✓		
2	Menambahkan Perangkat GPS	Pengguna dapat menambahkan perangkat GPS ke kendaraan dengan mudah.	✓		
3	Mengubah Data Kendaraan	Pengguna dapat mengubah data kendaraan sesuai kebutuhan dengan mudah.	✓		
4	Menghapus Data Kendaraan	Pengguna dapat menghapus data kendaraan yang tidak lagi digunakan dengan mudah.	✓		
5	Pemantauan Kendaraan Real-time	Sistem menampilkan posisi kendaraan secara real-time dengan jelas pada peta.	✓		
6	Status Perangkat GPS	Pengguna dapat melihat status aktif atau tidak aktifnya perangkat GPS dengan mudah.	✓		
7	Pengelolaan Zona Geofencing	Fitur pengaturan zona geofencing (administratif, radius, maupun polygon) mudah digunakan.	✓		
8	Notifikasi Pelanggaran Geofencing	Pengguna dapat mengatur channel notifikasi pelanggaran geofencing (email dan/atau WhatsApp) dengan mudah.	✓		
9	Pengelolaan Home Location	Fitur Home Location membantu pengguna menentukan titik acuan lokasi kendaraan dengan mudah.	✓		
10	Riwayat Perjalanan	Pengguna dapat meninjau riwayat perjalanan kendaraan beserta fitur playback dengan mudah.	✓		
11	Riwayat Berhenti	Pengguna dapat meninjau riwayat berhenti kendaraan dengan mudah.	✓		
12	Ekspor Riwayat Perjalanan	Pengguna dapat mengeksport riwayat perjalanan kendaraan ke dalam bentuk PDF dengan mudah.	✓		
13	Ekspor Riwayat Berhenti	Pengguna dapat mengeksport riwayat berhenti kendaraan ke dalam bentuk PDF dengan mudah.	✓		

No	Fitur yang Diuji	Pernyataan Pengguna	Ya	Tidak	Catatan
14	Pengaturan Durasi Waktu Berhenti	Pengguna dapat mengatur sendiri durasi waktu berhenti sebelum notifikasi dikirimkan.	✓		
15	Pengaturan Notifikasi Berhenti	Pengguna dapat mengaktifkan atau menonaktifkan notifikasi berhenti sesuai kebutuhan.	✓		
16	Notifikasi Berhenti	Sistem mengirimkan notifikasi berhenti beserta alamat lengkap kendaraan dengan akurat.	✓		
17	Notifikasi Pelanggaran Geofencing	Sistem mengirimkan notifikasi pelanggaran geofencing dengan tepat ketika kendaraan keluar dari zona yang ditentukan.	✓		
18	Riwayat Notifikasi Kendaraan	Pengguna dapat meninjau seluruh notifikasi kendaraan pada satu halaman dengan mudah.	✓		
19	Pengaturan Batas Kecepatan	Pengguna dapat mengatur batas kecepatan kendaraan sesuai kebutuhan dengan mudah.	✓		
20	Notifikasi Kecepatan Berlebih	Sistem mengirimkan notifikasi ketika kendaraan melaju melebihi batas kecepatan yang telah ditentukan.	✓		
21	Kesesuaian Sistem secara Keseluruhan	Secara keseluruhan, sistem AF GPS Tracker membantu proses pemantauan kendaraan secara efektif.	✓		

D. KESIMPULAN PENGUJIAN

Jumlah Item Diuji	21 Item
Jumlah Jawaban "Ya"	21 Item
Jumlah Jawaban "Tidak"	0 Item
Persentase Penerimaan	100.00 %

Kesimpulan

- ☒ Sistem DITERIMA dan layak digunakan
☐ Sistem DITERIMA dengan catatan perbaikan
☐ Sistem DITOLAK dan perlu perbaikan menyeluruh

Catatan Tambahan

E. PERNYATAAN DAN TANDA TANGAN

Saya yang bertanda tangan di bawah ini menyatakan bahwa pengujian telah dilakukan secara langsung dan hasil yang tercatat dalam formulir ini adalah benar adanya.

Bandung, 24 September 2024

Penguji

(.....)