



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

SISTEM KEAMANAN KENDARAAN (MOBILE SECURITY VEHICLE'S) BERBASIS ARDUINO

MUHAMMAD ISA SENOAJI
NRP 0511164000078

Dosen Pembimbing
Ir. Muchammad Husni, M.Kom

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

SISTEM KEAMANAN KENDARAAN (MOBILE SECURITY VEHICLE'S) BERBASIS ARDUINO

MUHAMMAD ISA SENOAJI
NRP 0511164000078

Dosen Pembimbing
Ir. Muchammad Husni, M.Kom

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

MOBILE SECURITY VEHICLE'S BASED ON ARDUINO

MUHAMMAD ISA SENOAJI
NRP 0511164000078

Supervisor
Ir. Muchammad Husni, M.Kom

DEPARTMENT OF INFORMATICS
Faculty of Electronics and Smart Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SISTEM KEAMANAN KENDARAAN (MOBILE SECURITY VEHICLE'S) BERBASIS ARDUINO

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Arsitektur dan Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh :
MUHAMMAD ISA SENOAJI
NRP : 05111640000078

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Ir. Muchammad Husni, M.Kom
NIP: 19600221 198403 1001



A handwritten signature in black ink, appearing to read 'U. Husni', is written over a light grey rectangular background.

(pembimbing)

SURABAYA
JUNI, 2020

[Halaman ini sengaja dikosongkan]

SISTEM KEAMANAN KENDARAAN (MOBILE SECURITY VEHICLE'S) BERBASIS ARDUINO

Nama Mahasiswa : Muhammad Isa Senoaji
NRP : 05111640000078
Jurusan : Departemen Informatika FTEIC-ITS
Dosen Pembimbing : Ir. Muchammad Husni, M.Kom

ABSTRAK

Tugas akhir ini adalah merancang dan membangun sebuah alat Sistem Keamanan Kendaraan Motor menggunakan microcontroller Arduino, jaringan internet beserta modul-modul yang terpasang di dalamnya. Rancang bangun alat ini ditujukan untuk mengamankan kendaraan Motor dari pencurian dan memantau kondisi kendaraan. Diharapkan dengan adanya Sistem ini bisa memberi manfaat apabila diterapkan dalam kehidupan nyata.

Tantangan utama tugas akhir ini adalah menggabungkan microcontroller, beberapa modul, jaringan internet dan kelistrikan kendaraan agar saling terkoordinir dengan baik sehingga menjadi sebuah sistem keamanan kendaraan motor sesuai yang diharapkan.

Hasil dari Tugas Akhir ini adalah terbentuknya sistem keamanan kendaraan motor berbasis arduino yang mana sistem ini dapat memenuhi beberapa kebutuhan fungsional antara lain dapat memantau lokasi kendaraan, mengendalikan alarm, dan kelistrikan kendaraan melalui aplikasi web yang dapat digunakan pada multi-platform.

Kata kunci: *Arduino, Keamanan Kendaraan, Alarm, Internet of Things, Multi-platform, Web.*

MOBILE SECURITY VEHICLE'S SYSTEM BASED ON ARDUINO

Student Name : Muhammad Isa Senoaji
Student ID : 05111640000078
Major : Informatics Department FTEIC-ITS
Supervisor : Ir. Muchammad Husni, M.Kom

ABSTRACT

This final project is to design and build a device of Mobile Security Vehicle's using Arduino microcontroller along with motorcycle, network and moduls installed in it. The design of this device is intended to secure people's motorcycle from theft and monitor motorcycle condition. It is expected that with the Mobile Security Vehicle's system can provide benefits when applied in real life.

The main challenge of this final task is to combine microcontroller, some module, networks and motorcycle electricity to be well coordinated so that it becomes an mobile security vehicle's system as expected.

The result of this Final Project is Mobile Security Vehicle's System Based on Arduino where the system can satisfy some functional needs such as can monitoring motorcycle location, controlling alarm, and motorcycle electricity to avoid theft through web application that can be used in multi-platform.

Keywords: *Arduino, Vehicle Security, Alarm, Internet of Things, Multi-platform, Web*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

SISTEM KEAMANAN KENDARAAN BERBASIS ARDUINO

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Ayah, Umi dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Muchammad Husni beserta keluarga selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Shinta Syafiera Ramadhanti yang selalu ada untuk menemani dan memberi support penulis di saat susah, sedih, dan bahagia.
4. Seluruh keluarga besar Remaja Masjid Baitul Haq Keputih Surabaya yang selalu menemani dan membantu penulis.
5. Rekan-rekan tim tugas Akhir IoT yang selalu berbagi pengalaman dan saran selama mengerjakan Tugas Akhir.
6. Akbar Noto Ponco selaku rekan yang selalu mengingatkan, memarahi dan mengajak mabar Penulis selama pengerjaan Tugas Akhir ini.
7. Teman-teman angkatan 2016 jurusan Teknik Informatika ITS yang telah menemani dan mendukung penulis selama kuliah.
8. Google.com yang membantu penulis mengerjakan Tugas Akhir.
9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2020

Muhammad Isa Senoaji

DAFTAR ISI

| | |
|--|-------|
| LEMBAR PENGESAHAN..... | v |
| ABSTRAK | vii |
| ABSTRACT | viii |
| KATA PENGANTAR..... | ix |
| DAFTAR ISI..... | xi |
| DAFTAR GAMBAR | xv |
| DAFTAR TABEL | xvi |
| DAFTAR KODE SUMBER | xviii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Permasalahan..... | 2 |
| 1.3 Batasan Permasalahan | 2 |
| 1.4 Tujuan..... | 2 |
| 1.5 Manfaat..... | 2 |
| 1.6 Metodologi | 3 |
| 1.7 Sistematika Penulisan..... | 5 |
| BAB II TINJAUAN PUSTAKA..... | 8 |
| 2.1 Arduino Uno..... | 8 |
| 2.2 Modul Relay 2 Channel (HL-52S) | 9 |
| 2.3 Modul GPS (Neo-7M)..... | 10 |
| 2.4 Sensor Tegangan (ACS 712)..... | 10 |
| 2.5 Modul SIM800L..... | 11 |
| 2.6 Komunikasi Serial | 12 |
| 2.7 Arduino <i>Software</i> IDE..... | 13 |
| 2.8 ArduinoJSON | 14 |
| 2.9 <i>Framework</i> Laravel | 15 |
| 2.10 REST API..... | 16 |
| 2.11 Kelistrikan Kendaraan | 17 |
| BAB III ANALISIS DAN PERANCANGAN SISTEM..... | 18 |
| 3.1 Analisis | 18 |
| 3.1.1 Analisis Permasalahan..... | 18 |

| | | |
|--------|---|----|
| 3.1.2 | Deskripsi Umum..... | 18 |
| 3.2 | Perancangan Sistem..... | 19 |
| 3.2.1 | Lingkungan Implementasi | 19 |
| 3.2.2 | Perancangan Arsitektur Jaringan | 20 |
| 3.2.3 | Perancangan Sistem..... | 20 |
| 3.2.4 | Perancangan Alat..... | 22 |
| 3.2.5 | Perancangan Alat..... | 23 |
| 3.2.6 | Rancangan Pemasangan Alat..... | 24 |
| 3.2.7 | Kebutuhan Fungsional Sistem | 25 |
| 3.2.8 | Diagram Kasus Penggunaan | 26 |
| 3.2.9 | Perancangan Aplikasi | 27 |
| 3.2.10 | Perancangan Antarmuka Aplikasi | 27 |
| BAB IV | IMPLEMENTASI SISTEM..... | 30 |
| 4.1 | Implementasi Alat | 30 |
| 4.1.1 | Gambar Rangkaian alat | 30 |
| 4.2 | Implementasi <i>Coding</i> Alat..... | 31 |
| 4.2.1 | Kode Sumber Pembacaan Sensor | 31 |
| 4.2.3 | Kode Sumber <i>Request</i> Perintah | 33 |
| 4.2.4 | Kode Sumber Pengiriman Status Alat | 34 |
| 4.2.5 | Kode Sumber Logika Sistem..... | 34 |
| 4.2.6 | Kode Sumber Menghidupkan dan Mematikan Relay..... | 35 |
| 4.2.7 | Kode Sumber Menghidupkan dan Mematikan Buzzer | 36 |
| 4.3 | Implementasi Pembuatan Aplikasi Web..... | 37 |
| 4.3.1 | Kode Sumber Mengecek Status Alat..... | 37 |
| 4.3.2 | Kode Sumber Mengecek Perintah | 37 |
| 4.3.3 | Kode Sumber Menyimpan Data Status Alat..... | 37 |

| | |
|---|-----------|
| 4.3.6 Implementasi Tampilan Aplikasi Web..... | 38 |
| 4.3.7 Implementasi Pemasangan Alat | 41 |
| BAB V PENGUJIAN DAN EVALUASI | 44 |
| 5.1 Skenario Pengujian..... | 44 |
| 5.2 Pengujian Alat..... | 44 |
| 5.3 Evaluasi Pengujian Fungsionalitas | 48 |
| BAB VI KESIMPULAN DAN SARAN..... | 49 |
| 5.1 Kesimpulan..... | 49 |
| 5.2 Saran..... | 49 |
| DAFTAR PUSTAKA..... | 51 |
| BIODATA PENULIS..... | 54 |

[Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Arduino Uno [2] | 8 |
| Gambar 2.2 Modul Relay 2 Channel [3] | 9 |
| Gambar 2.3 Modul GPS [4] | 10 |
| Gambar 2.4 Sensor Tegangan [5]..... | 10 |
| Gambar 2.5 Modul SIM800L [6] | 11 |
| Gambar 2.6 Komunikasi Serial [7]..... | 12 |
| Gambar 2.7 Arduino <i>Software</i> IDE | 13 |
| Gambar 2.8 Framework Laravel [9]..... | 15 |
| Gambar 2.9 Ilustrasi REST API [10] | 17 |
| Gambar 3.1 Rancangan Arsitektur Jaringan..... | 20 |
| Gambar 3.2 <i>Flowchart Diagram</i> Sistem | 21 |
| Gambar 3.3 Cover Plastik | 22 |
| Gambar 3.4 Kabel Jumper..... | 22 |
| Gambar 3.5 <i>DC to DC Stepdown</i> | 23 |
| Gambar 3.6 <i>Block Diagram</i> Sistem..... | 24 |
| Gambar 4.1 Rangkaian Alat | 31 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 3.1 Lingkungan Implementasi | 19 |
| Tabel 3.2 Kebutuhan Fungsional Sistem | 25 |
| Tabel 3.3 Keterangan Kode Kasus Penggunaan..... | 26 |
| Tabel 5.1 Hasil Pengujian Eksekusi Perintah Ketika Diam | 45 |
| Tabel 5.2 Pengujian Eksekusi Perintah Ketika Bergerak | 46 |
| Tabel 5.3 Pengujian Keberhasilan Aplikasi dalam Memonitor Status Kendaraan | 47 |
| Tabel 5.6 Evaluasi Pengujian Fungsionalitas | 48 |

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

| | |
|--|----|
| Kode Sumber 4.1 Pembacaan Sensor | 32 |
| Kode Sumber 4.2 <i>Request</i> dan <i>Parsing</i> Respon Perintah..... | 33 |
| Kode Sumber 4.3 Pengiriman Status Alat | 34 |
| Kode Sumber 4.4 Logika Sistem..... | 35 |
| Kode Sumber 4.5 Menghidupkan dan Mematikan Relay..... | 36 |
| Kode Sumber 4.6 Menghidupkan dan Mematikan Buzzer..... | 36 |

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Kendaraan bermotor merupakan kebutuhan seluruh masyarakat dunia. Kendaraan menjadi kebutuhan primer dalam berbagai kegiatan maupun aktifitas manusia, mulai dari penggunaan pribadi hingga penggunaan bisnis. Disamping besarnya kebutuhan manusia akan kendaraan, terdapat masalah besar yang mana sampai saat ini masih dapat dikatakan krusial, yaitu pada bagian keamanan kendaraan. Menurut data dari Polrestabes Surabaya, tahun 2017 terdapat 2.105 pencurian kendaraan di Jawa Timur. Dari kasus kehilangan kendaraan tersebut, kendaraan yang dapat ditemukan kembali 57 kendaraan saja atau kurang dari 3 persennya. Hal ini yang menjadi perhatian penulis dalam perancangan keamanan kendaraan bermotor.

Sistem Keamanan Kendaraan berbasis Arduino dengan Kendali *via* Android merupakan sebuah perangkat *mobile* yang nantinya akan ditanam pada kendaraan yang mampu menghidupkan, mematikan mesin kendaraan, *tracking* posisi kendaraan serta membunyikan kendaraan (alarm). Perangkat ini nantinya akan dikendalikan menggunakan ponsel android sebagai kontrol pada perangkat yang ditanam pada kendaraan. Sebagai contoh, bilamana pengguna mengaktifkan layanan matikan paksa / *force shut down* pada aplikasi maka secara otomatis kendaraan tidak akan bisa menyala. Konsep Sistem Keamanan Kendaraan ini nantinya dipadukan dengan konsep *Internet of Things* (IoT) sehingga pengguna dapat memantau lokasi dan status kendaraannya di manapun dia berada.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana cara merancang Sistem Keamanan Kendaraan berbasis Arduino?
2. Bagaimana cara membuat aplikasi *multi-platform* untuk memantau dan mengontrol kendaraan dengan menggunakan konsep *Internet of Things*?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, antara lain:

1. Alat Keamanan Kendaraan dibuat menggunakan *Microcontroller* Arduino
2. Rancang bangun sistem terbatas pada pemasangan disepeda motor.
3. Alat dianggap memiliki koneksi internet.
4. Keamanan tingkat lanjut dalam pengoperasian aplikasi web.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Membuat alat keamanan kendaraan berbasis arduino
2. Membangun aplikasi *multi-platform* untuk memantau dan mengontrol kondisi kondisi kendaraan
3. Sebagai syarat kelulusan untuk mendapatkan gelar Sarjana 1.

1.5 Manfaat

Dengan dibuatnya tugas akhir ini diharapkan bisa memberi manfaat yang besar dan mengurangi rasa khawatir bagi pengguna kendaraan ketika tidak digunakan atau dipinjam orang lain.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi latar belakang pembuatan tugas akhir, rumusan masalah, batasan masalah, tujuan pembuatan, manfaat, metodologi hingga jadwal kegiatan pembuatan tugas akhir. Selain itu proposal tugas akhir ini memberikan ringkasan dari tugas akhir. Proposal tugas akhir juga berisi tinjauan pustaka yang digunakan sebagai referensi pembuatan tugas akhir ini.

b. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan program yaitu mengenai *Internet of Things (IoT)*, Arduino, sensor, modul-modul, Android, API dan Web.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat alat yang merupakan rancangan dari sistem keamanan kendaraan berbasis arduino beserta rancangan aplikasi berbasis web sebagai penunjangnya.

d. Implementasi perangkat keras

Implementasi perangkat keras dibangun menggunakan Arduino sebagai *microcontrollernya* disertai dengan beberapa modul yang terpasang yaitu modul Relay 2 Channel dengan Octocoupler, Buzzer, modul SIM800L, modul GPS Neo-7M, sensor tegangan, sensor getaran dan *Power Supply*.

e. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat keras dan perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah
- c. Batasan Tugas Akhir
- d. Tujuan
- e. Metodologi
- f. Sistematika Penulisan

2. Tinjauan Pustaka

3. Desain dan Implementasi

4. Pengujian dan Evaluasi

5. Kesimpulan dan Saran

6. Daftar Pustaka

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dipaparkan mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini, di antaranya adalah Arduino Uno, Modul Relay, Modul GPS, Sensor Tegangan, Modul WiFi, Komunikasi Serial, ArduinoJSON, Framework Laravel, REST API dan beberapa teori lainnya yang menunjang dalam pengerjaan penelitian.

2.1 Arduino Uno



Gambar 2.1 Arduino Uno [1]

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 *pin input* dari *output digital* dimana 6 *pin input* tersebut dapat digunakan sebagai output PWM dan 6 *pin input analog*, 16 MHz *osilator* kristal, koneksi USB, *jack power*, *ICSP header*, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC ke *adaptor-DC* atau baterai untuk menjalankannya. [1]

Untuk membangun Sistem Keamanan Kendaraan ini tentunya diperlukan sebuah *microcontroller* untuk mengatur jalannya sistem agar saling terkoordinir dengan baik.

Dalam pengerjaan Tugas Akhir kali ini Arduino Uno dipilih sebagai *microcontroller* dikarenakan *compatible* dengan beberapa komponen yang nantinya digunakan. Selain itu arduino harganya relatif lebih murah dan didukung dengan ratusan *library* yang ada. Pemrograman arduino tergolong mudah digunakan dan termasuk *Open-Source* baik dari segi *software* maupun dari segi *hardware*.

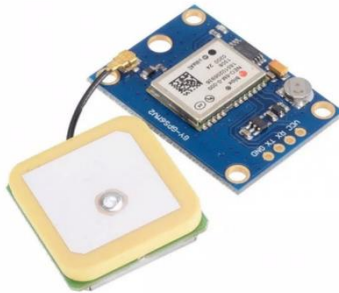
2.2 Modul Relay 2 Channel (HL-52S)



Gambar 2.2 Modul Relay 2 Channel [2]

Relay HL-52S memiliki 2 relay dengan tingkat 10A @250, 125 V AC dan 10A @30, 28 V DC. Konektor *output* tegangan tinggi mempunyai 3 pin, yang tengah adalah pin biasa, dan pin sisi lainnya merupakan pin untuk membuka arus dan menutup arus. Disisi lain terdapat 2 pasang pin. Pasangan pin pertama memiliki 4 pin, satu *ground*, pin VCC untuk menghidupkan modul, dan 2 input pun In1 dan In2. Pasangan pin lainnya memiliki 3 pin dengan *jumper* diantara JDVcc dan pin VCC. Dengan konfigurasi seperti ini, *electromagnet* dari relay akan langsung ditenagai dari *Board* Arduino dan jika terdapat kerusakan/kesalahan pada relay, maka *microcontroller* dapat mengalami kerusakan. [2]

2.3 Modul GPS (Neo-7M)

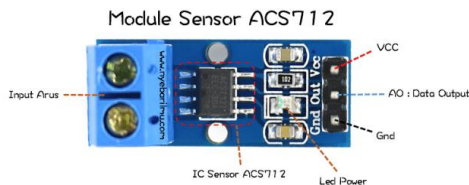


Gambar 2.3 Modul GPS [3]

NEO-7M GPS merupakan perangkat tambahan pada mikrokontroler seperti Arduino agar Arduino dapat mengirim lokasi dimana berada. Modul ini membutuhkan daya sebesar 3.3v dan memiliki 4 pin yaitu VCC, GND, TX, dan RX. Dalam proses *retrieve* data lokasi, modul akan mencari satelit untuk terhubung terlebih dahulu, lalu bila sudah terjadi koneksi maka akan ditandai dengan kedipan led merah pada modul. Untuk dapat digunakan Bersama Arduino, modul ini menggunakan komunikasi serial. [3]

Untuk membangun Sistem Keamanan Kendaraan ini diperlukan modul untuk mendeteksi lokasi kendaraan. Dalam pengerjaan Tugas Akhir kali ini modul GPS dipilih karena modul ini terbilang murah dan *compatible* dengan *microcontroller* arduino.

2.4 Sensor Tegangan (ACS 712)



Gambar 2.4 Sensor Tegangan [4]

ACS712 merupakan modul yang berfungsi sebagai pendeteksi arus/tegangan pada kelistrikan. Modul ini dapat menyaring data tegangan terendah dan tegangan tertinggi, modul ini menggunakan data analog sehingga dibutuhkan perhitungan pada code untuk benar-benar dapat digunakan. [4]

Untuk membangun Sistem Keamanan Kendaraan ini diperlukan sensor untuk mendeteksi arus tegangan listrik kendaraan. Sensor ini dapat digunakan untuk mendeteksi apakah terdapat tegangan arus listrik pada kendaraan atau tidak.

Dalam pengerjaan Tugas Akhir kali ini sensor tegangan dipilih karena sensor ini terbilang murah dan *compatible* dengan *microcontroller* arduino.

2.5 Modul SIM800L

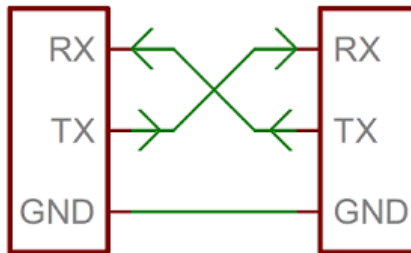


Gambar 2.5 Modul SIM800L [5]

SIM800L merupakan *shield* untuk kartu SIM yang berfungsi sebagai perangkat tambahan *microcontroller* seperti Arduino agar dapat terhubung langsung dengan operator jaringan dan membuat koneksi TCP/IP maupun koneksi jaringan telfon. Modul ini membutuhkan daya sekitar 3.7-4.4v. [5]

Untuk membangun Sistem Keamanan Kendaraan ini diperlukan adanya koneksi internet untuk mengirimkan data status pada alat menuju *server* dan menerima perintah dari server. Maka digunakan modul SIM800L.

2.6 Komunikasi Serial



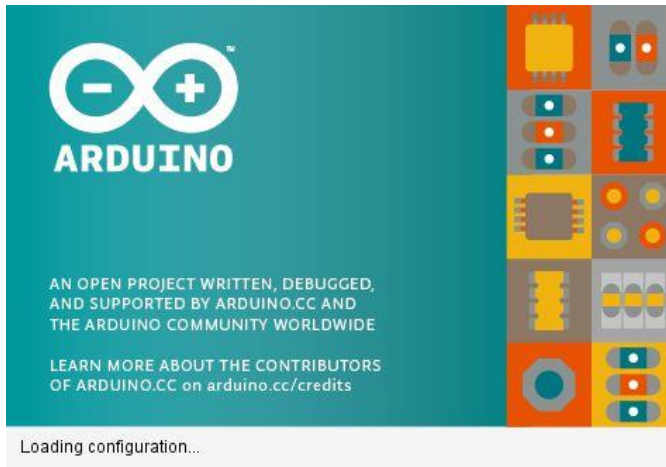
Gambar 2.6 Komunikasi Serial [6]

Komunikasi Serial merupakan salah satu cara komunikasi antar modul yang sering digunakan pada modul-modul *microcontroller* seperti arduino. Komunikasi pada pin TX/RX menggunakan TTL *Logic Level* (5V atau 3.3V tergantung *board* yang digunakan). Komunikasi serial juga dapat digunakan antara modul dan monitor serial pengguna sehingga proses pengembangan dapat di ukur se-detail mungkin. [6]

Untuk membangun Sistem Keamanan Kendaraan ini diperlukan komunikasi antar modul, antara lain Arduino dengan modul GPS, Arduino dan modul WiFi dan lain-lain.

Dalam pengerjaan Tugas Akhir kali ini komunikasi serial dipilih karena mudah dan *compatible* dengan arduino.

2.7 Arduino Software IDE



Gambar 2.7 *Arduino Software IDE* [7]

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, *IC microcontroller* Arduino telah ditanamkan suatu program bernama *Bootloader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software

Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Arduino IDE digunakan untuk mengembangkan pembuatan fungsional alat pada sistem keamanan kendaraan ini. Versi yang digunakan untuk mengembangkan sistem ini adalah Arduino IDE versi 1.8.10. IDE ini tergolong ringan dan mudah untuk dipelajari.

2.8 ArduinoJSON

ArduinoJSON adalah sebuah format untuk berbagi data berupa JSON pada Arduino. Seperti dapat kita lihat dari namanya, JSON diturunkan dari bahasa pemrograman JavaScript, akan tetapi format ini tersedia bagi banyak bahasa lain termasuk Python, Ruby, PHP, dan Java. JSON biasanya dilafalkan seperti nama "Jason." [8]

JSON menggunakan ekstensi .json saat ia berdiri sendiri. Saat didefinisikan di dalam format file lain (seperti di dalam .html), ia dapat tampil didalam tanda petik sebagai JSON string, atau ia dapat dimasukkan kedalam sebuah variabel. Format ini sangat mudah untuk ditransfer antar server web dengan klien atau browser.

Karena sangat mudah dibaca dan ringan, JSON memberikan alternatif lebih baik dari XML dan membutuhkan formatting yang tidak banyak.

2.9 Framework Laravel



Gambar 2.8 Framework Laravel [9]

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan website berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*.

- Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
- View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
- Controller, Controller merupakan bagian yang menjembatani *model* dan *view*.

- Beberapa fitur yang terdapat di Laravel :
- *Bundles*, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
 - *Eloquent ORM*, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “*active record*” yang mengatasi masalah pada hubungan objek *database*.
 - *Application Logic*, merupakan bagian dari aplikasi, menggunakan *controller* atau bagian *Route*.
 - *Reverse Routing*, mendefinisikan relasi atau hubungan antara *Link* dan *Route*.
 - *Restful controllers*, memisahkan logika dalam melayani HTTP GET and POST.
 - *Class Auto Loading*, menyediakan *loading* otomatis untuk *class* PHP.
 - *View Composer*, adalah kode unit logikal yang dapat dieksekusi ketika *view* sedang dimuat.
 - *IoC Container*, memungkinkan obyek baru dihasilkan dengan pembalikan *controller*.
 - *Migration*, menyediakan sistem kontrol untuk skema *database*.
 - *Unit Testing*, banyak tes untuk mendeteksi dan mencegah regresi.
 - *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman. [9]

Dalam pengerjaan Tugas Akhir kali ini *Framework* Laravel digunakan untuk penunjang rancang bangun aplikasi web.

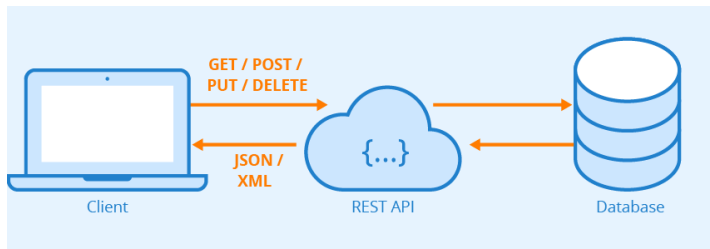
Versi Laravel yang digunakan pada pengerjaan Tugas Akhir kali ini adalah versi 5.8.

2.10 REST API

REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan

mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data. [10]

Dalam pengerjaan Tugas Akhir kali ini metode REST API digunakan untuk menerima dan mengirim data dari alat Keamanan Kendaraan kendaraan pusat server.



Gambar 2.9 Ilustrasi REST API [10]

2.11 Kelistrikan Kendaraan

Kelistrikan merupakan hal utama dan merupakan kebutuhan primer kendaraan. Tanpa kelistrikan, kendaraan tidak akan bisa berfungsi. Terdapat 2 macam tipe kelistrikan pada kendaraan, yaitu tipe arus AC (bolak balik) dan tipe arus DC (satu arah). Biasanya pada kendaraan berlampu bohlam, kelistrikan masih menggunakan arus DC, sedangkan kendaraan berlampu LED menggunakan arus AC.

Pada pengerjaan tugas akhir ini, manipulasi kelistrikan kendaraan diperlukan guna mengamankan kendaraan. Oleh karena itu, tujuan manipulasi kelistrikan kendaraan diterapkan pada alat Keamanan Kendaraan berbasis Arduino.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan keamanan kendaraan berbasis arduino ini. Mulai dari masalah yang melatar-belakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan alat dan aplikasi, yang meliputi perancangan alat, perancangan implementasi pada kendaraan dan tampilan antar muka halaman aplikasi.

3.1 Analisis

Tahap analisis meliputi analisis masalah, analisis kebutuhan, deskripsi umum sistem, dan kasus penggunaan sistem yang dibuat.

3.1.1 Analisis Permasalahan

Kendaraan merupakan kebutuhan yang wajib dimiliki oleh hampir setiap orang. Faktor keamanan kendaraan hingga saat ini menjadi masalah utama bagi masyarakat yang memiliki kendaraan. Hal ini dibuktikan dengan tingginya angka kehilangan kendaraan sepanjang tahun. Pada saat memarkirkan atau meminjamkan kendaraan, mayoritas orang merasa cemas dengan kendaraannya khususnya bagi orang yang tinggal bersama seperti kontrakan ataupun kos-kosan yang minim lahan parkir. Karena kondisi tersebut, biasanya kendaraan diparkir begitu saja di halaman atau teras rumah. Hal ini juga dirasakan bagi penyewa kendaraan, biasanya mereka khawatir akan kehilangan atau kendaraannya dibawa kabur oleh si Penyewa.

3.1.2 Deskripsi Umum

Berdasarkan permasalahan yang ada pada pembahasan domain permasalahan, solusi yang ditawarkan adalah pembuatan pengaman kendaraan berbasis arduino. Alat pengaman ini dapat

menjadikan kendaraan dalam beberapa *mode*, yaitu *mode* normal dan *mode anti-theft*. Dalam *mode* normal, aktifitas kelistrikan kendaraan akan normal seperti sedia kala dan dapat digunakan secara normal. Namun bila dalam keadaan *mode anti-theft*, kelistrikan kendaraan akan terputus total sehingga kendaraan tidak akan bisa dioperasikan, dan juga bila kendaraan tersenggol maka alarm akan aktif. Dalam pembuatan rancangan ini akan digunakan beberapa modul yaitu modul relay, gps dan alarm serta sensor tegangan untuk mendeteksi kondisi kelistrikan kendaraan dan getaran. Selain itu dibuat sebuah aplikasi berbasis Web yang dapat memantau dan mengontrol alat yang dipasang pada kendaraan secara *realtime*.

3.2 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan rancangan dan arsitektur yang akan digunakan untuk penelitian. Juga menggambarkan bagaimana proses-proses yang harus dijalani. Proses-proses ini akan dipresentasikan dengan diagram alir (*flowchart*) pada setiap uji coba yang dilakukan.

3.2.1 Lingkungan Implementasi

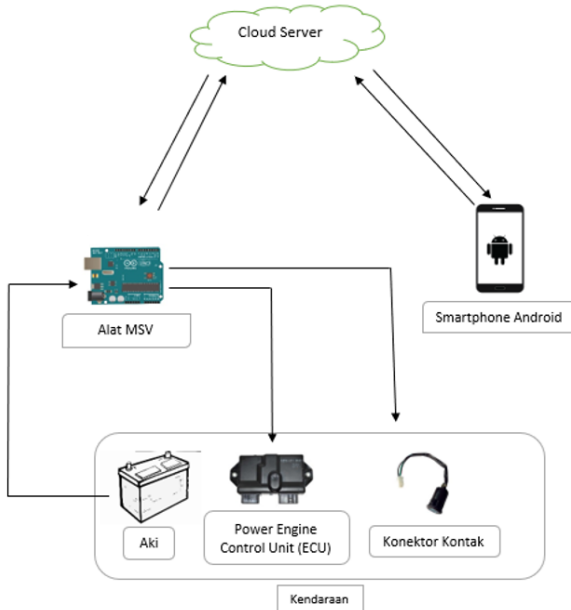
Lingkungan implementasi untuk pengerjaan tugas akhir ini adalah ditunjukkan pada **Tabel 3.1** berikut:

Tabel 3.1 Lingkungan Implementasi

| Perangkat | Jenis Perangkat | Spesifikasi |
|-----------------|-----------------|--------------------------------|
| Perangkat keras | Prosesor | AMD Ryzen 5 3.7Ghz |
| | RAM | 4 GB |
| | System Type | 64-bit <i>Operating System</i> |
| Perangkat lunak | Sistem Operasi | CentOS dengan cPanel |
| | IDE | Arduino <i>Software IDE</i> |

3.2.2 Perancangan Arsitektur Jaringan

Berikut ini adalah rancangan arsitektur jaringan yang akan dibuat dalam sistem keamanan kendaraan berbasis arduino dapat dilihat pada **Gambar 3.1**

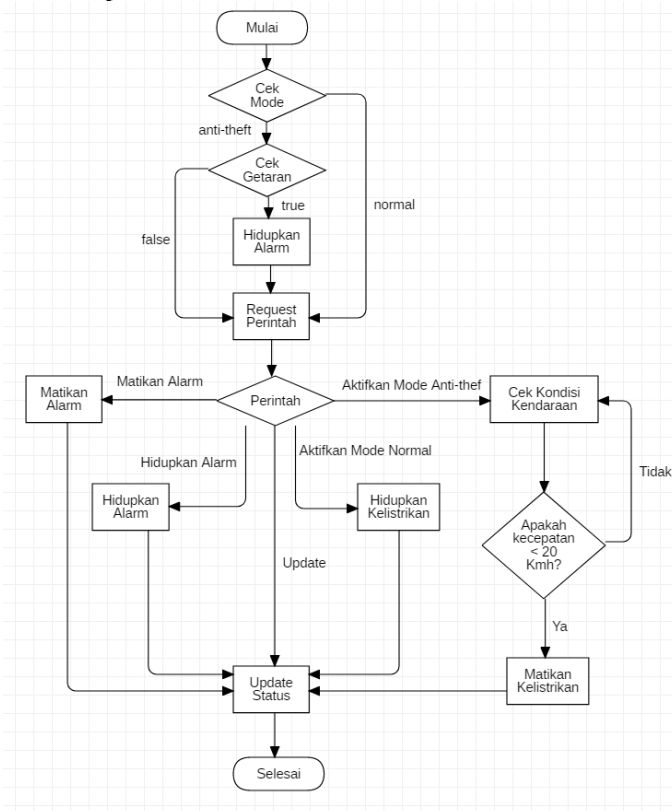


Gambar 3.1 Rancangan Arsitektur Jaringan

3.2.3 Perancangan Sistem

Pada tahap ini akan dijelaskan perancangan kerja sistem kendaraan berbasis arduino. Sistem memiliki 2 *mode* yaitu *mode normal* dan *mode anti-theft*. *Mode normal* yaitu dimana kelistrikan kendaraan aktif seperti sedia kala sehingga kendaraan dapat digunakan. *Mode anti-theft* yaitu dimana kelistrikan kendaraan dalam kondisi dimatikan sehingga kendaraan tidak dapat digunakan, dan juga mendeteksi getaran guna mengaktifkan alarm. Sistem juga memiliki *mode* kendali *buzzer* untuk membunyikan kendaraan.

Sistem bekerja pada 2 kondisi, yaitu pada keadaan berhenti/terparkir atau berjalan. Hal ini dikarenakan memungkinkan pengguna lupa mengaktifkan *mode anti-theft* lalu terjadi kendaraan tercuri. Penerapan mematikan kelistrikan kendaraan secara langsung bilamana kendaraan dalam keadaan berjalan maka akan membahayakan pengemudi. Untuk itu diambil kecepatan batas atas kendaraan untuk dapat dimatikan kelistrikannya yaitu 20 Km/h. *Flowchart Diagram* kerja sistem dapat dilihat pada **Gambar 3.2**.



Gambar 3.2 *Flowchart Diagram* Sistem

3.2.4 Perancangan Alat

Pada tahap ini akan dijelaskan perancangan alat untuk membangun sistem keamanan kendaraan berbasis arduino. Dalam membangun alat ini digunakan beberapa bahan yang diperlukan yaitu :

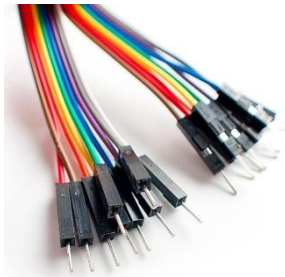
- a. Cover Plastik



Gambar 3.3 Cover Plastik

Plastik digunakan untuk membuat alat keamanan kendaraan ini dikarenakan kokoh dan kuat bila dibandingkan dengan kertas *duplex*, kertas karton ataupun kardus. Selain itu saat ini sudah terdapat teknologi cetak 3D dimana *design* 3D bisa dicetak dengan mudah dan presisi.

- b. Kabel *Jumper*



Gambar 3.4 Kabel *Jumper*

Kabel *jumper* diperlukan untuk koneksi antar *microcontroller* Arduino dengan modul-modul. Kabel *jumper* juga mudah digunakan serta dapat digunakan berkali-kali.

c. DC to DC *Stepdown*



Gambar 3.5 *DC to DC Stepdown*

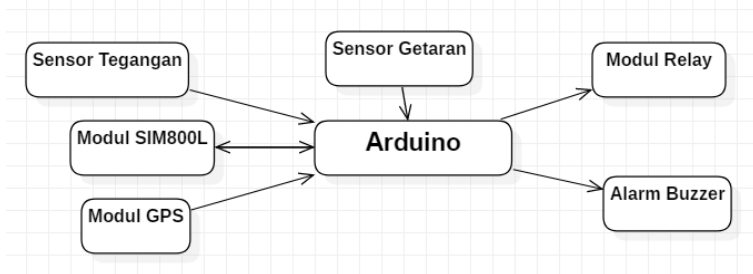
DC to DC stepdown digunakan untuk menurunkan tegangan listrik dari Aki ke alat. modul ini harganya cukup terjangkau dan bisa dibeli dengan mudah melalui toko *online*.

3.2.5 Perancangan Alat

Pada tahap ini akan dijelaskan perancangan alat untuk membangun pengaman kendaraan berbasis arduino. Dalam membangun alat ini digunakan beberapa komponen yang diperlukan yaitu :

- d. Arduino Uno
- e. Papan *Breadboard*
- f. Kabel *Jumper*
- g. Sensor Tegangan (ACS-712)
- h. Sensor Getaran
- i. Modul SIM800L
- j. Modul Relay 2 *Channel*
- k. Alarm/*Buzzer*
- l. Modul GPS (Neo 7M)
- m. *DC to DC Stepdown*

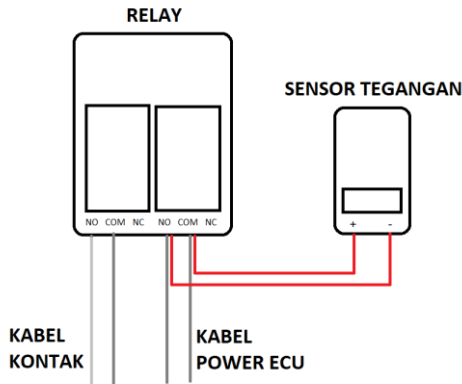
Berikut ini adalah gambar diagram blok rangkaian komponen pada sistem kendaraan berbasis arduino dapat dilihat pada **Gambar 3.6**.



Gambar 3.6 Block Diagram Sistem

3.2.6 Rancangan Pemasangan Alat

Pada tahap ini akan dijelaskan rancangan pemasangan alat Keamanan Kendaraan berbasis Arduino pada kendaraan. Perlu diketahui bahwa alat Keamanan Kendaraan berbasis Arduino ini bekerja pada kelistrikan kendaraan. Oleh karena itu, dalam pemasangan alat terdapat 2 sistem kelistrikan kendaraan yang harus di hubungkan ke alat. Kelistrikan pertama yaitu kabel yang terhubung pada kontak kendaraan, dan kelistrikan kedua yaitu kabel yang terhubung dengan pemasok listrik pada Sistem Injeksi / ECU. Kabel kelistrikan tersebut dihubungkan pada relay 2 channel pada slot NO (*Normally Open*) dan COM (*Common*) yaitu secara *default* relay dalam kondisi aktif/terhubung. Lalu pada kabel kelistrikan yang terhubung pada injeksi diberikan kabel *jumper* menuju sensor Tegangan sebagai pendeteksi tegangan bilamana mesin kendaraan dalam keadaan menyala. Diagram Pemasangan Alat dapat dilihat pada gambar 3.7.



Gambar 3.7 Rancangan Pemasangan

3.2.7 Kebutuhan Fungsional Sistem

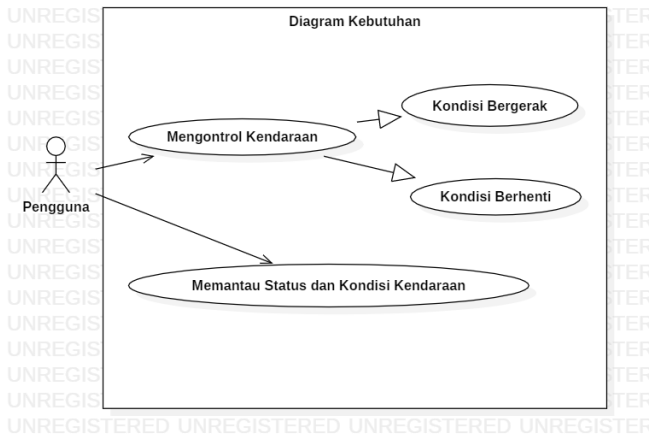
Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Adapun kebutuhan fungsional sistem keamanan ini dideskripsikan dalam **Tabel 3.2**

Tabel 3.2 Kebutuhan Fungsional Sistem

| Kode Kebutuhan | Kebutuhan Fungsional | Deskripsi |
|----------------|---|---|
| F-001 | Memantau status kendaraan | Memantau kondisi kendaraan, yaitu <i>mode</i> , status kelistrikan, alarm dan lokasi kendaraan. |
| F-002 | Mengontrol kendaraan ketika kondisi berhenti | Mengontrol <i>mode</i> mesin dan status alarm dalam kondisi kendaraan berhenti |
| F-003 | Mengontrol kendaraan ketika kondisi berjalan | Mengontrol <i>mode</i> mesin dan status alarm dalam kondisi kendaraan berjalan |
| F-004 | Mengaktifkan alarm dalam <i>mode anti-theft</i> | Dalam <i>mode anti-theft</i> sistem alarm dapat aktif ketika kendaraan tersenggol |

3.2.8 Diagram Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Berikut adalah diagram kasus penggunaan dari sistem keamanan kendaraan berbasis arduino dapat dilihat di **Gambar 3.8**



Gambar 3.8 UseCase Diagram

Dari gambar diagram kasus penggunaan yang ada pada **Gambar 3.8** keterangan kode kasus penggunaan dapat dilihat pada **Tabel 3.3**

Tabel 3.3 Keterangan Kode Kasus Penggunaan

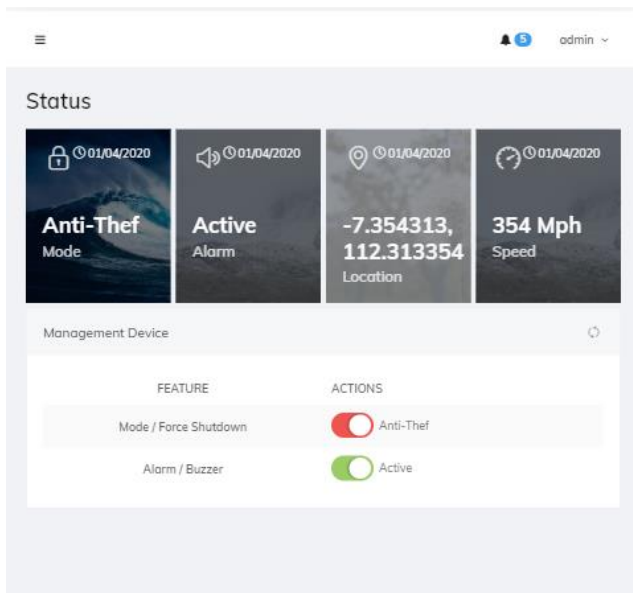
| Kode Kasus Penggunaan | Kasus Penggunaan | Aktor |
|-----------------------|---|----------|
| UC-001 | Mengontrol Kendaraan dalam kondisi berhenti | Pengguna |
| UC-002 | Mengontrol Kendaraan dalam kondisi bergerak | Pengguna |
| UC-003 | Memantau kondisi dan status kendaraan | Pengguna |

3.2.9 Perancangan Aplikasi

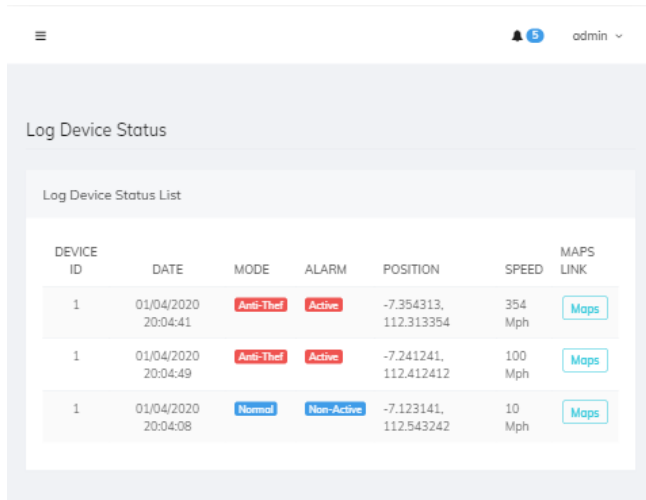
Pada tahap ini akan dijelaskan perancangan aplikasi untuk penunjang keamanan kendaraan berbasis arduino ini. Dalam membangun aplikasi ini akan digunakan Kerangka Kerja / *framework* Laravel yang akan di *deploy* pada *cloud* yang berbasis cPanel.

3.2.10 Perancangan Antarmuka Aplikasi

Berikut adalah perancangan antarmuka aplikasi keamanan kendaraan ini. Dari rancangan ini pengguna dapat melakukan pemantauan kondisi dan status kendaraan, serta lokasi kendaraan. Pengguna juga dapat melihat *log status* kendaraan pada halaman *log device* Rancangan antarmuka aplikasi dapat dilihat pada **Gambar 3.9** dan **Gambar 3.10**



Gambar 3.9 Rancangan antarmuka aplikasi



The screenshot displays a web application interface for 'Log Device Status'. At the top, there is a navigation menu icon on the left and a user profile 'admin' with a notification bell icon on the right. The main content area is titled 'Log Device Status' and contains a 'Log Device Status List' table. The table has seven columns: 'DEVICE ID', 'DATE', 'MODE', 'ALARM', 'POSITION', 'SPEED', and 'MAPS LINK'. There are three data rows. The first two rows show 'Anti-Theft' mode with 'Active' alarms, while the third row shows 'Normal' mode with a 'Non-Active' alarm. Each row includes a 'Maps' button for further details.

| DEVICE ID | DATE | MODE | ALARM | POSITION | SPEED | MAPS LINK |
|-----------|------------------------|------------|------------|--------------------------|------------|----------------------|
| 1 | 01/04/2020 20:04:41 | Anti-Theft | Active | -7.354313, 112.313354 | 354 Mph | Maps |
| 1 | 01/04/2020 20:04:49 | Anti-Theft | Active | -7.241241, 112.412412 | 100 Mph | Maps |
| 1 | 01/04/2020 20:04:08 | Normal | Non-Active | -7.123141, 112.543242 | 10 Mph | Maps |

Gambar 3.10 Rancangan antarmuka log status aplikasi

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI SISTEM

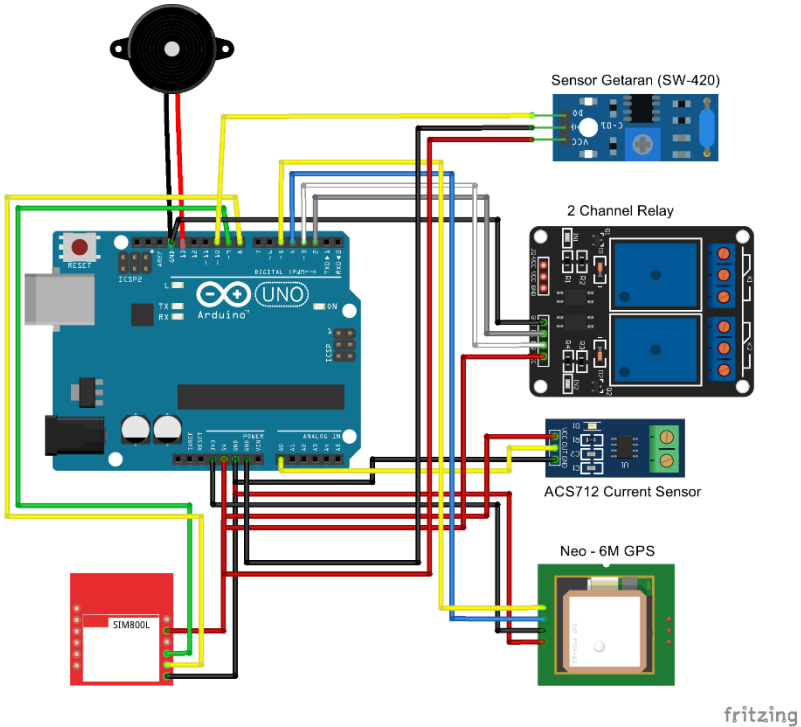
Bab ini membahas implementasi dari perancangan sistem keamanan kendaraan berbasis arduino yang di dalamnya akan dibahas mengenai lingkungan pengembangan sistem, implementasi sistem dan antarmuka pengguna.

4.1 Implementasi Alat

Pada bagian ini akan dijelaskan secara detail mengenai implementasi alat pada sistem keamanan kendaraan berbasis arduino.

4.1.1 Gambar Rangkaian alat

Pada tahap implementasi ini akan dijabarkan bagaimana proses-proses merangkai alat kendaraan berbasis arduino ini. Gambar rangkaian alat dapat dilihat pada **Gambar 4.1**.



Gambar 4.1 Rangkaian Alat

4.2 Implementasi *Coding* Alat

Pada sub bab Implementasi *Coding* Alat akan menjelaskan tentang pembuatan perangkat yang telah direncanakan pada Bab III. Berikut ini adalah implementasi *coding* terhadap alat. Dalam proses *coding* dikerjakan menggunakan Arduino IDE. Detail pembahasan mengenai kode sumber yang dibuat dalam sistem akan dijelaskan pada sub-bab berikut.

4.2.1 Kode Sumber Pembacaan Sensor

Bagian ini adalah implementasi dari pembacaan sensor pada Sistem Keamanan Kendaraan. Dalam hal ini ada beberapa sensor

yang digunakan yaitu sensor tegangan (ACS-712), GPS (Neo-6M), dan sensor getaran. Kode sumber dari pembacaan sensor tertera pada **Kode Sumber 4.1**

```
while (GPS.available() > 0){
  if (Tiny.encode(GPS.read())){
    if (Tiny.location.isValid())
    {
      lat = Tiny.location.lat();
      lang = Tiny.location.lng();
      speed = Tiny.speed.mps();
    }
  }
}

int CheckMesin(){
  float temp;
  int val = analogRead(A0);
  temp = val / 4.092;
  float val2 = (temp / 10);
  if(val2 >= 1){
    return 1;
  }
  else{
    return 0;
  }
}

int CheckGetaran(){
  if(digitalRead(pinGetaran)){
    return 1;
  }
  else{
    return 0;
  }
}
```

Kode Sumber 4.1 Pembacaan Sensor

4.2.3 Kode Sumber *Request* Perintah

Bagian ini adalah implementasi dari request perintah dari alat yang tercatat diserver. Dalam hal ini Arduino akan me-*request* perintah ke server melalui modul SIM800L. Nantinya respon data yang diterima modul akan digunakan sebagai parameter status alat. Terdapat 2 jenis respon perintah dari server, yaitu "*update*" bilamana tidak terdapat perintah yang belum dijalankan dan data berupa JSON bila terdapat perintah yang belum dijalankan. Bila jenis perintah berupa JSON maka nantinya akan di-*parsing* menjadi bentuk ArduinoJSON melalui kode sumber yang tertera pada **Kode Sumber 4.2**

```
String url;

Sim800l.println(url);
waitUntilResponse("OK");

Sim800l.println("AT+HTTPACTION=0");
waitUntilResponse("+HTTPACTION:");

Sim800l.println("AT+HTTPREAD");
waitUntilResponse("OK");

int firstIndex = responseString.indexOf("{");
int lastIndex = responseString.indexOf("}");
if(firstIndex > 0 || lastIndex > 0 ){
    String subResp = responseString;
    if(subResp.indexOf("update") < 0){
        DeserializationError error =
deserializeJson(doc,subResp);
        if(error) {
            Serial.print(F("deserializeJson() failed: "));
            Serial.println(error.c_str());
            return;
        }
    }
}
```

Kode Sumber 4.2 *Request* dan *Parsing* Respon Perintah

4.2.4 Kode Sumber Pengiriman Status Alat

Bagian ini adalah implementasi dari pengiriman data status alat ke server bilamana sistem melakukan *request* update status ataupun perintah alat lainnya. Kode sumber tertera pada **Kode Sumber 4.3**

```
Void updateStatus{
    url =
    "AT+HTTTPARA=\"URL\", \"http://msv.cydeveloper.com/api/create?";
    url += "buzzer=" + String(CheckBuzzer()) + "&";
    url += "relay=" + String(CheckRelay()) + "&";
    url += "lat=" + String(lat,6) + "&";
    url += "lang=" + String(lang,6) + "&";
    url += "speed=" + String(speed)+ "&";
    url += "mesin=" + String(CheckMesin()+ "&";
    url += "id_command=" + String(id_command)+ "&";
    url += "id=" + String(id)+"\"";
    Sim8001.println(url);
    waitUntilResponse("OK");
    Sim8001.println("AT+HTTPACTION=0");
    waitUntilResponse("+HTTTPACTION:");
}
```

Kode Sumber 4.3 Pengiriman Status Alat

4.2.5 Kode Sumber Logika Sistem

Bagian ini adalah implementasi dari logika sistem Keamanan Kendaraan untuk menentukan perintah yang diterima dari server. Kode sumber tertera pada **Kode Sumber 4.4**

```

if(subResp.indexOf("update") < 0){
    if(doc["command"] == "relay"){
        if(doc["value"] == 1){
            TurnOnRelay();
        }
        else {
            if(speed < 5)
                TurnOffRelay();
            else
                lookForSpeed = true;
        }
    }
}

else if(doc["command"] == "buzzer"){
    if(doc["value"] == 1){
        TurnOnBuzzer();
    }
    else {
        TurnOffBuzzer();
    }
}
updateStatus();
}
else {
    updateStatus();
}
if(CheckRelay()){
    if(CheckGetaran() && speed < 1)
        TurnOnBuzzer()
}
}

```

Kode Sumber 4.3 Logika Sistem

4.2.6 Kode Sumber Menghidupkan dan Mematikan Relay

Bagian ini adalah implementasi dari menghidupkan dan mematikan relay bilamana perintah mengaktifkan / mematikan *mode Anti-theft* dilakukan serta mengecek status relay apakah dalam kondisi menyala atau mati. Kode sumber tertera pada **Kode Sumber 4.5**

```

void TurnOnBuzzer(){
    digitalWrite(buzzer, HIGH);
}

void TurnOffBuzzer(){
    digitalWrite(buzzer, LOW);
}

bool CheckRelay(){
    bool r1 = digitalRead(relay1);
    bool r2 = digitalRead(relay2);
    if (r1 && r2){
        return true;
    }
    else{
        return false;
    }
}

```

Kode Sumber 4.6 Menghidupkan dan Mematikan Relay

4.2.7 Kode Sumber Menghidupkan dan Mematikan *Buzzer*

Bagian ini adalah implementasi dari menghidupkan dan mematikan *buzzer* bilamana perintah mengaktifkan / mematikan alarm dilakukan. Kode sumber tertera pada **Kode Sumber 4.6**

```

bool CheckBuzzer(){
    bool data = digitalRead(buzzer);
    return data;
}

void TurnOnBuzzer(){
    digitalWrite(buzzer, HIGH);
}

void TurnOffBuzzer(){
    digitalWrite(buzzer, LOW);
}

```

Kode Sumber 4.4 Menghidupkan dan Mematikan *Buzzer*

4.3 Implementasi Pembuatan Aplikasi Web

Pada bagian ini akan dijelaskan mengenai implementasi pembuatan Aplikasi Web keamanan kendaraan berbasis arduino. Aplikasi ini dirancang menggunakan IDE *Sublime Text 3* dengan kerangka kerja (*framework*) Laravel versi 5.8.

4.3.1 Kode Sumber Mengecek Status Alat

Bagian ini adalah implementasi koding untuk mengecek status alat sesuai yang *disetting* pada server menggunakan API Laravel. Rangkaian kode dapat dilihat pada **Kode Sumber 4.7**

```
public function index(){
    $recent = LogDeviceStatus::latest()->first();
    return view('admin.index',compact('recent'));
}
```

Kode Sumber 4.7 Cek Status Alat

4.3.2 Kode Sumber Mengecek Perintah

Bagian ini adalah implementasi koding untuk mengecek perintah yang belum dijalankan pada alat. Rangkaian kode dapat dilihat pada **Kode Sumber 4.8**

```
public function commandCheck(Request $request){
    $command = Command::where('device_id',$request->id)-
    >where('status',0)->select(['id','command','value'])-
    >first();
    if(is_null($command)){
        return "{update}";
    }else{
        return json_encode($command);
    }
}
```

Kode Sumber 4.8 Mengecek Perintah

4.3.3 Kode Sumber Menyimpan Data Status Alat

Bagian ini adalah implementasi koding untuk menyimpan data status yang dikirimkan alat ke server. Rangkaian kode dapat dilihat pada **Kode Sumber 4.9**.

```

public function store(Request $request){
    try {
        if ($request->has('id_command')) {
            $command = Command::find($request->id_command);
            $command->status = true;
            $command->save();
        }
        $data = new LogDeviceStatus();
        $data->device_id = $request->id;
        $data->relay = $request->relay;
        $data->buzzer = $request->buzzer;
        $data->lat = $request->lat;
        $data->lang = $request->lang;
        $data->speed = $request->speed;
        $data->mesin = $request->mesin;
        $data->save();
        return 1;
    } catch (\Exception $e) {
        return $e->getMessage();
    }
}

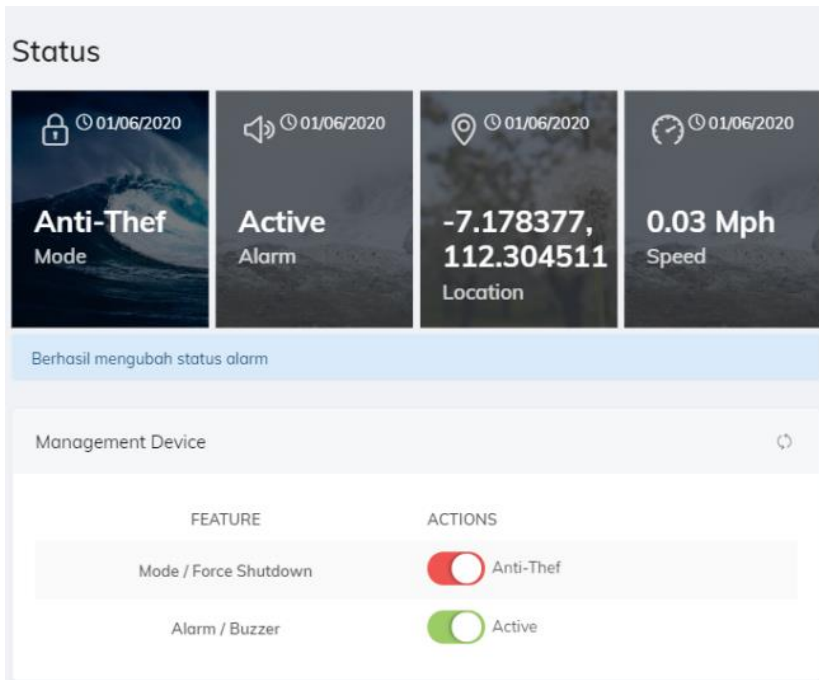
```

Kode Sumber 4.9 Menyimpan data status alat

4.3.6 Implementasi Tampilan Aplikasi Web

Bagian ini menampilkan hasil implementasi tampilan aplikasi Web Sistem Keamanan Kendaraan berbasis Arduino. Aplikasi web ini terdiri dari 2 halaman yaitu halaman *dashboard* dan halaman *log status device*.

Halaman *dashboard* berisi kondisi terbaru kendaraan yang dikirim alat berisi status *mode* mesin, status alarm, lokasi kendaraan, dan kecepatan kendaraan. Selain itu terdapat kontrol kendaraan yang berfungsi untuk kontrol *mode* mesin dan kontrol alarm kendaraan. Gambar halaman *dashboard* terdapat pada Gambar 4.2.



Gambar 4.2 Tampilan Halaman Dashboard

Halaman *Log Status Device* adalah halaman catatan/log status kendaraan yang dikirim alat yang terpasang. Halaman ini berisi status lengkap yang dikirimkan alat, yaitu tanggal pengiriman data, status *mode* mesin kendaraan, status alarm, koordinat lokasi kendaraan, kecepatan kendaraan, status mesin serta tombol untuk melihat posisi kendaraan pada *Google Maps*. Gambar halaman *log status device* dapat dilihat pada Gambar 4.3.

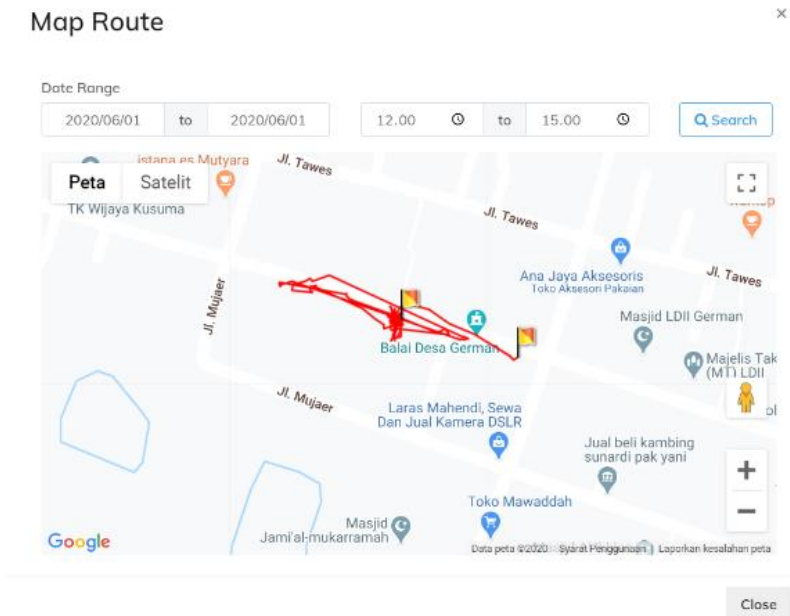
Log Device Status

Log Device Status List

| DEVICE ID | DATE | MODE | ALARM | POSITION | SPEED | MAPS LINK |
|-----------|------------------------|-----------|------------|--------------------------|-----------------|----------------------|
| 1 | 06/06/2020 11:06:54 | Anti-Thef | Active | -7.178377, 112.304511 | 0.03 Mph | Maps |
| 1 | 06/06/2020 11:06:33 | Anti-Thef | Active | -7.178377, 112.304511 | 0.036011 Mph | Maps |
| 1 | 01/06/2020 17:06:39 | Normal | Non-Active | -7.178267, 112.30451 | 0.07 Mph | Maps |
| 1 | 01/06/2020 17:06:12 | Anti-Thef | Non-Active | -7.178211, 112.30454 | 0.69 Mph | Maps |

Gambar 4.3 Tampilan Halaman Log Status

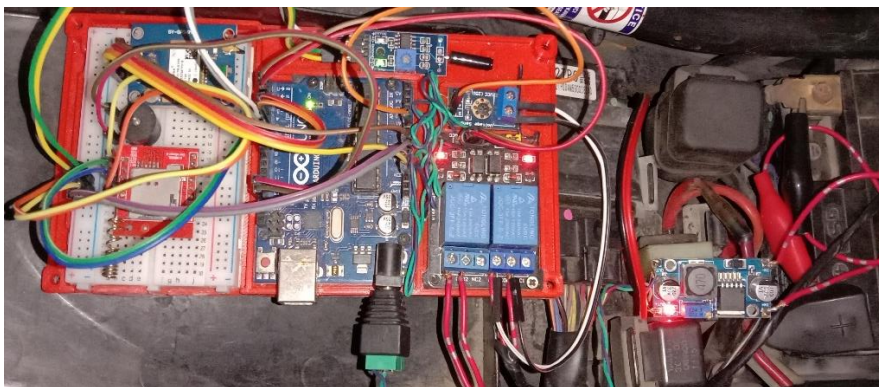
Selain itu, terdapat juga tombol yang berfungsi sebagai riwayat perjalanan kendaraan berdasarkan pada rentang tanggal dan jam yang dipilih. Riwayat perjalanan kendaraan akan terlihat dengan garis merah yang terhubung berdasarkan data koordinat kendaraan yang tersimpan di server. Adapun dua gambar bendera merepresentasikan titik mulai dan titik akhir posisi kendaraan berdasarkan rentang waktu yang dipilih. Gambar riwayat perjalanan/*tracking map* terdapat pada Gambar 4.3.



Gambar 4.4 Tampilan Tracking Maps

4.3.7 Implementasi Pemasangan Alat

Pada bagian ini akan dijelaskan mengenai hasil implementasi pemasangan alat keamanan kendaraan berbasis arduino. Pemasangan dilakukan pada kendaraan motor berkapasitas 150cc dengan sistem pembakaran injeksi. Kabel kelistrikan yang terhubung pada kontak dihubungkan pada *channel* 1 relay dan kabel kelistrikan yang terhubung pada suplai listrik menuju *Engine Control Unit* / ECU dihubungkan pada *channel* 2 relay dengan status *Normally Open* / NO. Lalu pada *channel* 2 relay diberikan kabel jumper yang dipasang pada sensor tegangan.



Gambar 4.3 Implementasi Pemasangan Alat

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan analisa sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya sistem keamanan kendaraan berbasis arduino ini.

5.1 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dibagi menjadi dua yaitu pengujian alat dan pengujian perangkat lunak. Pengujian alat dilakukan dengan cara memberi rangsangan pada sensor dan melihat respon yang diberikan oleh sistem.

5.2 Pengujian Alat

Pengujian fungsionalitas alat dilakukan dengan melakukan skenario yang sama dengan rancangan alur proses aplikasi sebagai tolak ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada alat dapat dijabarkan pada subbab berikut :

5.2.1 Pengujian Respon Alat pada Perintah saat Keadaan Diam

Dalam membuat Sistem Keamanan Kendaraan ini terdapat kasus dimana kendaraan dalam keadaan diam/terparkir. Rincian hasil pengujian respon alat dapat dilihat pada **Tabel 5.1**.

Tabel 5.1 Hasil Pengujian Eksekusi Perintah Ketika Diam

| Nomor | UC-001 |
|------------------------------|---|
| Nama | Pengujian Respon Alat terhadap Perintah ketika dalam kondisi diam/terparkir |
| Tujuan Pengujian | Menguji respon sistem ketika alat menerima perintah ketika kondisi diam/terparkir |
| Skenario 1 | Pengguna mengaktifkan <i>mode Anti-theft</i> pada aplikasi |
| Kondisi Awal | Status kendaraan dalam <i>mode</i> Normal |
| Langkah Pengujian | Pengguna mengaktifkan <i>mode Anti-theft</i> |
| Hasil Yang Diharapkan | Status kendaraan menjadi <i>mode Anti-theft</i> (relay terputus) |
| Hasil Yang Didapat | Status kendaraan berubah menjadi <i>mode anti-theft</i> |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Status kendaraan dalam <i>mode anti-theft</i> |
| Skenario 2 | Pengguna menonaktifkan <i>mode Anti-theft</i> pada aplikasi |
| Kondisi Awal | Status kendaraan dalam <i>mode Anti-theft</i> |
| Langkah Pengujian | Pengguna menonaktifkan <i>mode Anti-theft</i> |
| Hasil Yang Diharapkan | Status kendaraan menjadi <i>mode</i> Normal (relay terhubung) |
| Hasil Yang Didapat | Status kendaraan berubah menjadi <i>mode</i> normal |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Status kendaraan dalam <i>mode</i> Normal |
| Skenario 3 | Alarm akan aktif bila tersenggol dalam keadaan <i>anti-theft</i> |
| Kondisi Awal | Status kendaraan dalam <i>mode Anti-theft</i> dan alarm non-aktif |
| Langkah Pengujian | Pengguna menggerakkan kendaraan |
| Hasil yang Diharapkan | Alarm menjadi aktif |
| Hasil Yang Didapat | Alarm aktif |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Alarm menjadi aktif |

5.2.2 Pengujian Respon Alat pada Perintah saat Keadaan Bergerak

Dalam membuat Sistem Keamanan Kendaraan ini terdapat kasus dimana kendaraan dalam keadaan bergerak atau sedang digunakan, hal ini dikarenakan terdapat kasus yaitu kendaraan di curi ketika pemilik belum mengaktifkan *mode anti-theft*. Rincian hasil pengujian respon alat dapat dilihat pada **Tabel 5.2**.

Tabel 5.2 Pengujian Eksekusi Perintah Ketika Bergerak

| Nomor | UC-002 |
|------------------------------|--|
| Nama | Pengujian Respon Alat terhadap Perintah ketika dalam kondisi bergerak/digunakan |
| Tujuan Pengujian | Menguji respon sistem ketika alat menerima perintah ketika kendaraan sedang bergerak/digunakan |
| Skenario 1 | Pengguna mengaktifkan <i>mode Anti-theft</i> pada aplikasi |
| Kondisi Awal | Status kendaraan dalam <i>mode Normal</i> |
| Langkah Pengujian | Pengguna mengaktifkan <i>mode Anti-theft</i> |
| Hasil Yang Diharapkan | Status kendaraan menjadi <i>mode Anti-theft</i> (relay terputus) |
| Hasil Yang Didapat | Status kendaraan berubah menjadi <i>mode anti-theft</i> |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Status kendaraan dalam <i>mode anti-theft</i> |
| Skenario 2 | Pengguna menonaktifkan <i>mode Anti-theft</i> pada aplikasi |
| Kondisi Awal | Status kendaraan dalam <i>mode Anti-theft</i> |
| Langkah Pengujian | Pengguna menonaktifkan <i>mode Anti-theft</i> |
| Hasil Yang Diharapkan | Status kendaraan menjadi <i>mode Normal</i> (relay terhubung) |
| Hasil Yang Didapat | Status kendaraan berubah menjadi <i>mode normal</i> |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Status kendaraan dalam <i>mode Normal</i> |

5.3 Pengujian Aplikasi

Pengujian dilakukan untuk menentukan keberhasilan aplikasi dalam memonitor status kendaraan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada **Tabel 5.3**.

Tabel 5.3 Pengujian Keberhasilan Aplikasi dalam Memonitor Status Kendaraan

| | |
|------------------------------|---|
| Nomor | UC-003 |
| Nama | Pengujian Keberhasilan Aplikasi dalam Memonitor Status Kendaraan |
| Tujuan Pengujian | Menguji keberhasilan alat dalam memonitor data status kendaraan |
| Skenario 1 | Pengguna melihat status kendaraan dihalaman <i>dashboard</i> |
| Kondisi Awal | Data status kendaraan belum muncul |
| Langkah Pengujian | Pengguna melihat halaman <i>dashboard</i> |
| Hasil Yang Diharapkan | Halaman aplikasi dapat menampilkan status kondisi kendaraan |
| Hasil Yang Didapat | Halaman aplikasi dapat menampilkan status kondisi kendaraan |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Halaman aplikasi menampilkan status kondisi kendaraan |
| Skenario 2 | Pengguna melihat <i>log</i> / riwayat status kendaraan dihalaman <i>Log Device</i> |
| Kondisi Awal | Halaman <i>log device</i> belum terbuka |
| Langkah Pengujian | Pengguna membuka halaman <i>log device</i> |
| Hasil Yang Diharapkan | Halaman aplikasi dapat menampilkan <i>log</i> status kondisi kendaraan |
| Hasil Yang Didapat | Halaman aplikasi dapat menampilkan <i>log</i> status kondisi kendaraan |
| Hasil Pengujian | Berhasil |
| Kondisi Akhir | Halaman aplikasi menampilkan <i>log</i> status kondisi kendaraan |

5.3 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada **Tabel 5.6**. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5.4 Evaluasi Pengujian Fungsionalitas

| No. | Kode Kasus Penggunaan | Nama Kasus Penggunaan | Skenario | Hasil |
|-----|-----------------------|--|------------|----------|
| 1 | UC-001 | Pengujian Kontrol Kendaraan dalam kondisi berhenti | Skenario 1 | Berhasil |
| | | | Skenario 2 | Berhasil |
| | | | Skenario 3 | Berhasil |
| 2 | UC-002 | Pengujian kontrol Kendaraan dalam kondisi bergerak | Skenario 1 | Berhasil |
| | | | Skenario 2 | Berhasil |
| 3 | UC-003 | Pengujian Aplikasi dalam memonitoring status kendaraan | Skenario 1 | Berhasil |
| | | | Skenario 2 | Berhasil |

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

5.1 Kesimpulan

Dari hasil pengamatan yang dilakukan selama proses perancangan, implementasi dan pengujian yang dilakukan dapat diambil kesimpulan sebagai berikut :

1. Sistem keamanan kendaraan berhasil dibuat menggunakan *microcontoller* Arduino beserta komponen-komponen yang tersusun di dalamnya yaitu berupa modul SIM800L, modul GPS, modul relay 2 *channel* dan sensor tegangan. Hal ini ditunjukkan dengan berhasilnya pengujian respon sistem terhadap kontrol yang diberikan.
2. Aplikasi Web untuk memantau dan mengontrol serta mengamankan kendaraan berhasil dibuat dengan menggunakan *framework* Laravel dengan penunjang sistem *cloud*.

5.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem pada masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang dilakukan :

1. Untuk menerapkan Sistem Keamanan Kendaraan dalam kehidupan diperlukan adanya koneksi internet serta jaringan yang lebih kuat dan cepat guna menghindari delay data yang berlebihan.
2. Disarankan untuk penelitian selanjutnya menggunakan pengaman kelistrikan lanjutan, seperti sekering guna menghindari adanya arus pendek pada alat.

3. Disarankan untuk penelitian selanjutnya menghubungkan nyala lampu sein dengan alarm sehingga mempermudah kendaraan untuk ditemukan.

DAFTAR PUSTAKA

- [1] "reichelt," [Online]. Available: <https://www.reichelt.com/de/en/Single-Board-Microcontroller/ARDUINO-UNO-DIP/3/index.html?ACTION=3&GROUPID=8243&ARTICLE=154902>. [Accessed 19 June 2020].
- [2] "Modul Relay 2 Channel untuk Arduino," [Online]. Available: <http://www.worldofrobotic.com/product/2524-modul-relay-2-channel-untuk-arduino.html>. [Accessed 21 June 2020].
- [3] "Neo 6M Datasheet," [Online]. Available: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf). [Accessed 21 June 2020].
- [4] "Karakteristik Sensor Arus ACS 712," 09 Nov 2018. [Online]. Available: <https://www.andalanelektro.id/2018/11/karakteristik-sensor-suhu-acs-712.html>. [Accessed 21 June 2020].
- [5] D. Ardianto, "Sim800l GSM to Arduino," [Online]. Available: <http://www.belajarduino.com/2016/05/sim800l-gsmgprs-module-to-arduino.html>. [Accessed 21 June 2020].
- [6] "Komunikasi Serial Arduino Uno," 31 July 2017. [Online]. Available: <https://www.nyebarilmu.com/komunikasi-serial-arduino-uno-via-komputer/>. [Accessed 22 June 2020].
- [7] S. Arduino, "Mengenal Arduino Software (IDE)," 16 Mar 2016. [Online]. Available: <https://www.sinuarduino.com/artikel/mengenal-arduino-software-ide/>. [Accessed 02 June 2020].

- [8] "Mengenal Format JSON," 19 Oct 2017. [Online]. Available: <https://www.codepolitan.com/mengenal-format-json-59e8152dd0e51>. [Accessed 16 June 2020].
- [9] "Pengertian dan Keunggulan Framework Laravel," 07 June 2016. [Online]. Available: <https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>. [Accessed 22 June 2020].
- [10] Kiddy, "RESTFul API," 29 April 2019. [Online]. Available: <https://medium.com/@kiddy.xyz/restful-api-apaan-tuh-dbcfa434761e>. [Accessed 22 June 2020].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, **Muhammad Isa Senoaji**, lahir di Jakarta, 19 Maret 1998. Penulis menempuh pendidikan sekolah menengah pertama di SMP Negeri 86 Jakarta dan melanjutkan pendidikannya di SMA Sulthon Aulia Boarding School. Selanjutnya penulis melanjutkan pendidikan sarjana di Departemen Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi guru ngaji di Masjid

Baitul Haq Keputih Surabaya dan aktif pada beberapa kepanitiaan seperti LKMM Pra-TD, Schematics, Remaja Masjid, Forum Pemuda Surabaya dan kegiatan lainnya. Dalam bidang professional, Penulis juga aktif dalam mengembangkan start-up software house serta pekerjaan freelance sebagai full-stack developer. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Arsitektur dan Jaringan Komputer (AJK). Penulis dapat dihubungi melalui email: isasenoaji@gmail.com.