



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

RANCANG BANGUN PROTOTYPE ALAT PELACAKAN MOTOR DENGAN SISTEM PERINGATAN DINI MENGGUNAKAN MIKROKONTROLER ARDUINO

GURUH ADI DARMA
NRP 5110 100 230

Dosen Pembimbing I
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc.

Dosen Pembimbing II
Hudan Studiawan, S.Kom, M.Kom

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TUGAS AKHIR - KI141502

**RANCANG BANGUN PROTOTIPE ALAT
PELACAKAN MOTOR DENGAN SISTEM
PERINGATAN DINI MENGGUNAKAN
MIKROKONTROLER ARDUINO**

**GURUH ADI DARMA
NRP 5110 100 230**

**Dosen Pembimbing I
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc.**

**Dosen Pembimbing II
Hudan Studiawan, S.Kom, M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

PROTOTYPE OF TRACKING DEVICE FOR MOTORCYCLE WITH EARLY WARNING SYSTEM USING ARDUINO MICROCONTROLLER

**GURUH ADI DARMA
NRP 5110 100 230**

**First Supervisor
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc.**

**Second Supervisor
Hudan Studiawan, S.Kom, M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN PROTOTIPE ALAT PELACAKAN MOTOR DENGAN SISTEM PERINGATAN DINI MENGGUNAKAN MIKROKONTROLER ARDUINO TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

GURUH ADI DARMA
NRP . 5110 100 230

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr.Eng. Radityo Anggoro, S.Kom., M.Kom. (Pembimbing 1)
NIP: 198410162008121002

2. Hudan Studiawan, S.Kom., M.Kom. (Pembimbing 2)
NIP: 198705112012121003



SURABAYA
Januari, 2017

[Halaman ini sengaja dikosongkan]

PROTOTYPE ALAT PELACAKAN MOTOR DENGAN SISTEM PERINGATAN DINI MENGUNAKAN MIKROKONTROLER ARDUINO

Nama : Guruh Adi Darma
NRP : 5110 100 230
Jurusan : Teknik Informatika – FTif ITS
Dosen pembimbing I : Dr. Eng. Radityo Anggoro, S.kom.,
M.Sc.
Dosen Pembimbing II : Hudan Studiawan, S.Kom., M.Kom.

ABSTRAK

Kendaraan bermotor pada saat ini adalah suatu sarana yang hampir selalu dipakai setiap hari pada semua lapisan masyarakat, entah itu berbentuk kendaraan bermotor milik pribadi ataupun kendaraan bermotor angkutan umum. Di Indonesia sendiri jalan raya pun setiap harinya telah dipenuhi oleh lalu lintas kendaraan bermotor yang sangat tinggi. Salah satu kendaraan bermotor yang mendominasi di setiap jalan raya adalah sepeda motor. Sepeda motor banyak diminati oleh masyarakat dikarenakan mudahnya akses untuk memiliki sepeda motor tersebut, mulai dari uang muka yang murah atau bahkan tidak membayar uang muka sama sekali hingga kredit yang mudah. Sayang sekali di saat meningkatnya tingkat konsumtif masyarakat terhadap sepeda motor diikuti pula dengan maraknya kejadian pencurian terhadap sepeda motor.

Dari banyaknya kejadian tersebut teretuslah ide untuk membuat suatu prototipe alat untuk melacak sepeda motor yang berpindah tempat tanpa sepengetahuan pemilik sepeda motor. Dalam proses pembuatan prototipe alat ini, penulis menggunakan sebuah *shield* GPS/GPRS/GSM v3.0 dari dfrobot yang sudah terdapat sensor GPS di dalamnya untuk mengetahui koordinat

posisi kendaraan bermotor yang sudah diberikan prototipe alat ini nantinya. Untuk membantu *shield* tersebut dalam pengolahan data dan pengiriman data menuju *server* maka dibutuhkan sebuah mikrokontroler Arduino Uno. Hasil yang didapat dari sensor GPS tersebut akan ditampilkan dalam bentuk *mobile web*.

Prototipe alat ini nantinya akan di uji coba mulai dari uji coba fungsionalitas hingga uji coba performa dengan beberapa scenario yang telah dirancang. Berdasarkan uji coba performa dan fungsionalitas, alat ini memerlukan waktu sekitar 5 menit atau bahkan lebih untuk *locking* posisi GPS dan paling lama 5 detik untuk mengirimkan notifikasi kepada pemilik apabila kendaraan telah berpindah tempat.

Kata kunci: Sepeda motor, Berpindah tempat, GPS, dan Mikrokontroler Arduino.

PROTOTYPE OF TRACKING DEVICE FOR MOTORCYCLE WITH EARLY WARNING SYSTEM USING ARDUINO MICROCONTROLLER

Name : Guruh Adi Darma
NRP : 5110 100 230
Department : Teknik Informatika – FTIf ITS
Supervisor I : Dr. Eng. Radityo Anggoro, S.Kom.,
M.Sc.
Supervisor II : Hudan Studiawan, S.Kom., M.Kom.

ABSTRACT

At this time, motor vehicle is almost always been used every day by people on every stratum of society, be it in the form of motor vehicles belonging to private or public transportation vehicles. In Indonesia alone, every day on the highway has been met by motor vehicle traffic is very high. One motor vehicle that predominate in each road are motorcycles. Motorcycles much in demand by the public due to easier access to have a motorcycle, ranging from inexpensive down payment or even not make a down payment at all to easy credit. Unfortunately, in a time of rising public consumption levels for motorcycles followed with the widespread incidence of theft of motorcycles.

The number of theft of motorcycles occurrences has inspired the idea to create a prototype tool to track migrating motorcycle without the knowledge of the owner of a motorcycle. In the process of prototyping tool, the authors use a shield GPS / GPRS / GSM v3.0 of dfrobot that already contained GPS sensor to determine the coordinates of the position of the motorcycle which has been given a prototype of this tool later. To help those shields in data processing and data transmission to the server, it takes an

Arduino Uno microcontroller. The results obtained from the GPS sensor will be displayed in the form of mobile web.

The prototype of this tool will be tested ranging from test functionality to test the performance with some scenarios that have been designed. Based on test performance and functionality, this tool takes about 5 minutes or even more to locking the GPS position and a maximum of 5 seconds to send a notification to the owner if the vehicle has been moved.

Keyword: Motorcycle, Migrating, GPS, and Microcontroller Arduino

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah Subhanahu Wata'alla, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “ **RANCANG BANGUN PROTORIPE ALAT PELACAKAN MOTOR DENGAN SISTEM PERINGATAN DINI MENGGUNAKAN MIKROKONTROLER ARDUINO** ” dengan baik.

Dalam kesempatan kali ini, penulis ingin mengucapkan terima kasih sebanyak-banyaknya serta apresiasi setinggi-tingginya kepada pihak-pihak yang turut serta dan ikut andil membantu penulis dalam menyelesaikan Tugas Akhir ini, maupun dalam proses menempuh pendidikan pada jenjang Strata Satu Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Penulis mengucapkan terima kasih kepada:

1. Allah SWT yang selalu melimpahkan rahmat, rezeki, dan nikmat kepada seluruh hamba-Nya.
2. Papa, Almarhumah Mama, Kakak Perempuan, dan drg. Deavita Dinari yang selalu memberikan semangat, doa, serta dorongan yang tiada henti kepada penulis untuk segera menyelesaikan Tugas Akhir ini.
3. Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. dan bapak Hudan Studiawan, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan kepercayaan, dukungan dan bimbingan, nasehat, perhatian, dan segala hal yang telah diberikan kepada penulis.
4. Bapak Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc. selaku dosen wali yang telah memberikan nasihat kepada penulis dalam merancang program studi pada setiap semester serta membantu penulis apabila mengalami kesusahan.

5. Bapak Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. selaku ketua jurusan Teknik Informatika ITS, Bapak Dr.Eng. Radityo Anggoro S.Kom, M.Sc. selaku koordinator TA, Ibu Isye Arieshanti, S.Kom, M.Phil selaku koordinator KP penulis dan segenap Bapak/Ibu dosen Teknik Informatika yang telah memberikan ilmu kepada penulis.
6. Pak Yudi, Pak Moerdiono dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis saat menimba ilmu di Teknik Informatika ITS.
7. Sahabat saya Samodro Bagus Prasetyanto S.Kom., MBA. beserta I Gusti Ngurah Ady Kusuma S.Kom.,M.Kom. yang telah membantu banyak dalam proses pengerjaan Tugas Akhir ini hingga detik-detik terakhir, tanpa kalian mungkin saya masih kesusahan mengerjakan ini sendirian.
8. Alief,varistha,yudha,irham,ilmal,rasyid,naufal,febrian,valent ,serta seluruh keluarga besar laboratorium GCL sebelum berganti nama menjadi IGS seperti sekarang, terima kasih banyak telah menjadi teman dekat sekaligus keluarga selama berada di TC.
9. Teman-teman seperjuangan bidang minat KBJ serta para admin lab yang lain terima kasih atas kebersamaanya selama ini.
10. Teman-teman angkatan 2010, baik yang sudah lulus maupun masih berjuang untuk gelar S.Kom., terima kasih banyak untuk kalian dan tetap semangat untuk yang masih berjuang.
Permintaan maaf terbesar dari penulis atas segala kekurangan yang masih terdapat dari diri penulis, baik yang terdapat pada buku ini maupun selama proses penulisan hingga selesainya buku ini. Saran dan kritik yang membangun sangat penulis harapkan.

Surabaya, Januari 2017

Guruh Adi Darma

DAFTAR ISI

LEMBAR PENGESAHAN	v
Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
Daftar gambar	xvii
daftar tabel	xxi
1. BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan dan Manfaat	3
1.4.1. Tujuan	3
1.4.2. Manfaat	3
1.5. Metodologi.....	3
2. BAB II.....	7
2.1. Mikrokontroler Arduino.....	7
2.2. IDE Arduino.....	8
2.3. Global Positioning System.....	10
2.4. Shield GPS/GPRS/GSM v3.0 DFRobot	11
2.5. Regulator Step Down XL4015.....	12
3. BAB III	15
3.1. Deskripsi Umum Sistem	15

3.2.	Arsitektur Umum Sistem	16
3.3.	Perancangan Umum Sistem	17
3.4.	Perancangan Perangkat Keras	18
3.4.1.	Perancangan Shield GPS/GSM/GPRS DFRobot	18
3.4.2.	Perancangan Regulator Step Down XL4015	19
3.5.	Perancangan Diagram Alir Sistem	20
3.6.	Diagram Alir Aplikasi Sistem	21
3.6.1.	Diagram Alir Mendeteksi Lokasi	21
3.6.2.	Diagram Alir Mikrokontroler	23
3.6.3.	Diagram Alir Pengiriman Data pada GPRS	24
3.6.4.	Diagram Alir Penerimaan Data Pada Server	25
3.7.	Perancangan Kebutuhan Arus Listrik	27
3.8.	Perancangan Antar Muka Aplikasi	28
4.	BAB IV	29
4.1.	Lingkungan Implementasi	29
4.1.1.	Lingkungan Implementasi Perangkat Keras	29
4.1.2.	Lingkungan Implementasi Perangkat Lunak	29
4.2.	Implementasi Perangkat Keras	30
4.2.1.	Implementasi Perangkat Shield GPS/GPRS/GSM v3.0	30
4.2.2.	Implementasi Perangkat Regulator XL4015	31
4.3.	Implementasi Perangkat Lunak	32
4.3.1.	Implementasi Pada Mikrokontroler Arduino	33
4.3.2.	Implementasi Pada Aplikasi Web	37
5.	BAB V	45

5.1.	Uji Coba Fungsionalitas.....	45
5.1.1.	Lingkungan Uji Coba.....	46
5.1.2.	Uji Coba Deteksi Lokasi pada GPS	46
5.2.	Uji Coba Sistem Peringatan Dini	60
6.	BAB VI.....	73
6.1.	Kesimpulan	73
6.2.	Saran	73
	daftar pustaka	75
7.	LAMPIRAN.....	77
	BIODATA PENULIS	87

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Perangkat Mikrokontroler Arduino.....	8
Gambar 2.2. Antar muka IDE Arduino.....	9
Gambar 2.3. GPS/GPRS/GSM Module 3.0	12
Gambar 2.4. Regulator XL4015 step-down	13
Gambar 3.1. Arsitektur Umum Sistem.....	16
Gambar 3.2. Shield GPS/GSM/GPRS DFRobot dengan Arduino	19
Gambar 3.3. Perancangan Regulator Step Down XL4015	20
Gambar 3.4. Diagram Alir Sistem	21
Gambar 3.5. Diagram alir mendeteksi lokasi.....	22
Gambar 3.6. Diagram alir mikrokontroler	23
Gambar 3.7. Diagram Alir Pengiriman Data	25
Gambar 3.8. Diagram Alir Penerimaan Data Pada Server.....	26
Gambar 3.9. Tampilan pada Mobile Web.....	28
Gambar 4.1. Arduino berserta GPS Shield dan Regulator Terpasang di Motor.....	31
Gambar 4.2. Regulator XL4015 terpasang di motor.....	32
Gambar 4.3. Inisialisasi Pada Mikrokontroler Arduino	33
Gambar 4.4. Proses Membaca Data GPS Arduino	34
Gambar 4.5. inisialisasi GPS pada mikrokontroler Arduino.....	35
Gambar 4.6. Pengiriman data menuju server	36
Gambar 4.7. Penerimaan data dan penyimpanan ke database	38
Gambar 4.8. Proses menampilkan data pada aplikasi mobile web	40
Gambar 4.9. Fungsi untuk mengirim email	42
Gambar 4.10. Implementasi halaman web untuk menampilkan lokasi.....	43
Gambar 5.1. Lokasi Awal Uji Coba.....	49

Gambar 5.2. Titik Koordinat Posisi Awal.....	50
Gambar 5.3. Posisi Arduino Mendapat Koordinat Pertama.....	51
Gambar 5.4. Kondisi Sekitar Perempatan Dharmawangsa	52
Gambar 5.5. Kondisi Sekitar Jalan Nias	53
Gambar 5.6. Titik koordinat yang kendaraan bermotor.....	54
Gambar 5.7. Perempatan Jalan Ngagel Jaya Selatan	55
Gambar 5.8. Titik koordinat di jalan ngagel	56
Gambar 5.9. Titik koordinat di jalan nginden semolo	57
Gambar 5.10. Jalan Semolowaru Tengah	58
Gambar 5.11. Titik koordinat di jalan semolowaru	59
Gambar 5.12. Titik Koordinat Keseluruhan.....	60
Gambar 5.13. Isi email yang dikirim	62
Gambar 5.14 Titik Awal dan Akhir Uji Coba Pertama.....	63
Gambar 5.15 Koordinat Uji Coba Pertama	64
Gambar 5.16 Kiriman email Uji Coba Pertama	65
Gambar 5.17 Titik Awal dan Akhir uji coba kedua	66
Gambar 5.18 Koordinat Uji Coba Kedua.....	67
Gambar 5.19 Kiriman email Uji Coba Kedua.....	68
Gambar 5.20 Titik Awal dan Akhir Uji Coba Ketiga	69
Gambar 5.21 Koordinat Uji Coba Ketiga	70
Gambar 5.22 Kiriman email Uji Coba Ketiga	70
Gambar 7.1. Inisialisasi variabel yang dibutuhkan	77
Gambar 7.2. Fungsi setup arduino + gsm shield.....	78
Gambar 7.3. Fungsi utama pada arduino yang selalu looping	79
Gambar 7.4. Konfigurasi GPRS pada arduino untuk koneksi internet	80
Gambar 7.5. Fungsi untuk mengirim koordinat ke web server...	82
Gambar 7.6. Fungsi untuk batasan perpindahan motor.....	82
Gambar 7.7. Konfigurasi GPS pada arduino untuk mendapatkan koordinat	82

Gambar 7.8. Fungsi untuk mendapatkan raw data posisi longitude dan latitude.....	83
Gambar 7.9. Fungsi untuk memproses raw data longitude dan latitude menjadi longitude dan latitude yang dapat di tampilkan dalam map.....	85

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Struktur Tabel Basis Data	18
Tabel 3.2. Kebutuhan Suplai Listrik	27
Tabel 5.1. Hasil Uji Coba Kecepatan Lock GPS	47
Tabel 5.2. Hasil Uji Coba Informasi Lokasi GPS	48
Tabel 5.3. Hasil uji coba pengiriman data	61
Tabel 5.4 Hasil Uji Coba Sistem Peringatan Dini.....	71

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab pendahuluan akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir yang meliputi latar belakang, rumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir, serta metodologi dan sistematika penulisan buku Tugas Akhir

1.1. Latar Belakang

Indonesia merupakan Negara dengan tingkat konsumtif yang tinggi terutama pada kendaraan bermotor [1]. Tingginya kepemilikan kendaraan bermotor di Indonesia ternyata diikuti pula dengan tingginya tingkat pencurian kendaraan bermotor (curanmor) yang kerap kali terjadi di kota-kota besar hingga pelosok desa yang berada di Indonesia. Tingginya tingkat pencurian kendaraan bermotor (curanmor) di Indonesia membuat warga Negara Indonesia yang ingin membeli atau memiliki kendaraan bermotor menjadi was-was dengan adanya ketidaknyamanan dalam hal keamanan pada saat meninggalkan kendaraan bermotor mereka ketika mereka hendak berbelanja di *mini-market* atau sekedar membeli makanan pada sebuah rumah makan. Untuk itu, diperlukan sebuah alat atau aplikasi untuk memantau posisi sebuah kendaraan bermotor yang dapat memberikan *early warning* kepada pemilik/*user* dari kendaraan bermotor tersebut.

Aplikasi pemantau/*tracker* untuk kendaraan bermotor ini dirancang untuk dapat melakukan pemantauan pergerakan sebuah kendaraan bermotor pada saat kendaraan bermotor tersebut berada di kondisi hidup ataupun mati. Data yang dibutuhkan pada aplikasi ini adalah koordinat posisi kendaraan bermotor. Dibutuhkan sensor *Global Positioning System* (GPS) yang dipasang pada mikrokontroler Arduino dalam perancangan arsitektur sistem ini. Data dari koordinat titik posisi kendaraan bermotor ini akan dikirim oleh mikrokontroler Arduino ke aplikasi. Tujuannya adalah agar posisi koordinat titik kendaraan bermotor yang terbaru

dapat dibaca dan diolah dalam sebuah aplikasi web. Untuk mengirimkan data dari mikrokontroler arduino ke aplikasi, dibutuhkan sebuah alat yang bernama GSM shield. Pengiriman data didahului dengan melakukan koneksi pada aplikasi dan proses selanjutnya adalah pengiriman data oleh GSM shield.

Aplikasi nantinya akan digunakan oleh klien. Untuk mendapatkan data koordinat titik posisi kendaraan bermotor, mikrokontroler akan mengirim data kepada aplikasi. Ketika Arduino mengirim data, secara otomatis aplikasi akan mengolah data yang dikirim dan akan di tampilkan pada aplikasi Google Maps. Tujuan aplikasi ini yakni untuk memonitor posisi kendaraan bermotor yang bergerak tanpa sepengetahuan *user*. Diharapkan dengan aplikasi ini dapat dilakukan pemantauan posisi kendaraan bermotor secara *real-time* serta dapat dilakukan *tracking* oleh sang pemilik motor untuk mencari keberadaan motor yang dicuri atau berpindah tempat.

1.2. Rumusan Masalah

Adapun yang menjadi rumusan masalah dalam Tugas Akhir ini adalah sebagai berikut:

- a) Bagaimana cara mendapatkan posisi dengan menggunakan sensor Arduino?
- b) Bagaimana cara menentukan kapan sistem peringatan dini dijalankan?
- c) Bagaimana cara mengolah dan mengirim data ke aplikasi.

1.3. Batasan Masalah

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada Tugas Akhir ini, di antaranya sebagai berikut:

1. Perangkat menggunakan satu mikrokontroler Arduino.
2. Sensor yang digunakan adalah sensor GPS yang menjadi satu dengan GSM *shield*.
3. Menggunakan komunikasi GPRS pada saat mengirimkan data sensor ke *server*.

4. Perpindahan kendaraan bermotor di tandai hanya apabila arduino mengirimkan data koordinat yang berbeda dari koordinat yang paling awal dikirim.
5. Keakuratan posisi kendaraan bermotor terbatas mengikuti perangkat GSM *shield* dari DFRobot.
6. Keakuratan posisi akan sangat terbatas atau tidak mendapat koordinat sama sekali apabila memasuki ruangan karena GPS susah untuk *locking position* apabila di dalam ruangan.
7. Aplikasi berbasis *mobile web interface* dengan bahasa PHP dan JavaScript.

1.4. Tujuan dan Manfaat

1.4.1. Tujuan

Tujuan dari pembuatan Tugas Akhir adalah membuat sebuah alat yang dapat menjadi sistem peringatan dini apabila kendaraan bermotor diambil atau berpindah tempat tanpa sepengetahuan pemilik menggunakan mikrokontroler Arduino Uno. Prototipe alat dapat mengetahui perpindahan kendaraan motor secara *real-time* dan memberi notifikasi pada pemilik kendaraan bermotor apabila berpindah posisi.

1.4.2. Manfaat

Manfaat yang diharapkan dari pengerjaan Tugas Akhir ini adalah untuk memberikan bentuk solusi pelacakan kendaraan bermotor yang berpindah tempat dan diharapkan dapat mengurangi tingkat pencurian kendaraan bermotor (curanmor) di Indonesia yang masih masuk ke dalam taraf yang cukup tinggi dengan menggunakan sensor berbasis mikrokontroler Arduino yang hemat energi sehingga tidak terlalu memakan banyak daya yang tersedia di setiap kendaraan bermotor.

1.5. Metodologi

Adapun langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Penyusunan proposal Tugas Akhir ini bertujuan untuk membuat sistem yang memberikan peringatan dini kepada pemilik motor apabila terjadi pergerakan pada motor sang pemilik.

2. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan Tugas Akhir sekaligus mempelajarinya. Mulai dari pengumpulan literatur, diskusi, perancangan alat, modul Arduino Uno, modul GSM/GPRS/GPS DFRobot dan dokumentasi internet.

3. Perancangan Sistem

Tahap ini merupakan perancangan sistem dengan menggunakan studi literatur. Dengan bekal teori dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses perancangan sistem.

4. Implementasi Perangkat Lunak

Tahap ini merupakan implementasi rancangan alat yang telah dibuat. Tahap ini merupakan bentuk realisasi tahap sebelumnya dengan menggunakan GSM *shield* yang sudah memiliki sensor GPS di dalamnya untuk memantau posisi kendaraan bermotor.

5. Pengujian dan Evaluasi

Pada tahap ini akan dilakukan pengujian terhadap sistem yang dibangun. Untuk pengujian, maka akan disimulasikan motor yang bergerak dengan keadaan mati ataupun hidup untuk menampilkan analisa data dari hasil perpindahan koordinat posisi kendaraan bermotor yang dipakai sebagai bahan uji coba. Setelah itu akan dicatat kinerja perangkat dan kendala apa saja yang dapat mempengaruhi kinerja perangkat.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini disusun laporan Tugas Akhir sebagai dokumentasi pelaksanaan Tugas Akhir, yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah didapatkan selama pengerjaan Tugas Akhir ini.

1.5.1. Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi latar belakang, permasalahan, tujuan, batasan permasalahan, metodologi, dan sistematika penulisan dari Tugas Akhir yang dibuat.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi dasar teori yang mendukung pembahasan Tugas Akhir ini. Dasar teori yang dibahas meliputi kebutuhan perangkat mikrokontroler, penggunaan basis data, penggunaan *shield* GPS/GPRS/GSM v3.0 DFRobot, dan regulator *step down* XL4015.

BAB III. PERANCANGAN PERANGKAT

Bab ini berisi tentang perancangan sistem, flowchart dan perancangan antarmuka yang akan dibuat. Perancangan yang dibahas meliputi perancangan perangkat keras, input dan output data, serta pengiriman data, diagram alir metode pengiriman data dengan menggunakan GPRS, dan perancangan antarmuka perangkat lunak.

BAB IV. IMPLEMENTASI PERANGKAT

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* dari fungsi

input data dan pengiriman data pada mikrokontroler, antarmuka perangkat lunak dan *screenshot* aplikasi.

BAB V. EVALUASI DAN UJI COBA

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan beberapa jenis pengujian pada saat kendaraan bermotor berhenti maupun bergerak, performa maupun dalam sebuah kasus. Pengujian dilakukan pada beberapa tempat.

BAB VI. PENUTUP

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak atau alat untuk ke depannya.

BAB II

TINJAUAN PUSTAKA

Pada bab tinjauan pustaka akan dijelaskan beberapa hal mengenai teori yang berkaitan dengan sistem yang diimplementasikan. Bab ini bertujuan untuk memberikan gambaran umum kepada pembaca mengenai sistem yang akan dibuat selain menunjang pembuatan sistem dalam analisa kebutuhan sistem.

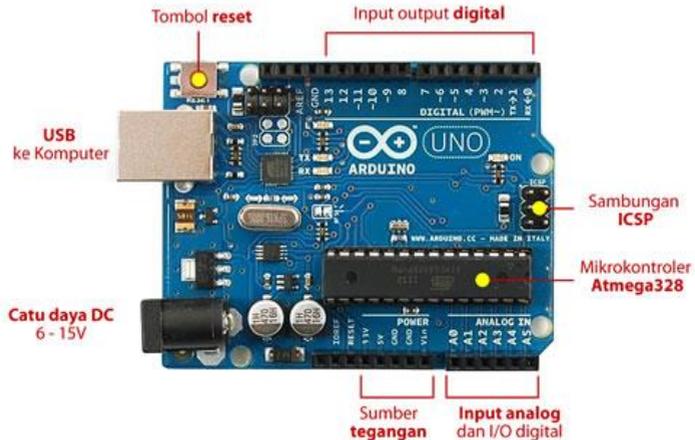
2.1. Mikrokontroler Arduino

Arduino merupakan sebuah mikrokontroler berbasis *single-board* yang bersifat *open-source*. Arduino dirancang untuk menggantikan tugas komputer sebagai alat untuk menerima inputan dari kondisi lingkungan sekitar. Arduino lebih ditujukan untuk mengendalikan perangkat keras dibandingkan komputer.

Arduino dikembangkan oleh tim pengembang yang terdiri dari: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, dan Nicholas Zambetti. Mikrokontroler Arduino dapat dipasangkan dengan bermacam-macam sensor. Adapun sensor yang dapat dipasangkan pada Arduino seperti sensor gerak, ultrasonik, panas, suara dan yang lainnya.

Untuk berkomunikasi dengan komputer dapat menggunakan berbagai macam cara, antara lain menggunakan kabel serial, ethernet shield, GPRS/GSM module, wifi-shield, bluetooth dan lain-lain [2]. Arduino Uno merupakan board mikrokontroler dengan menggunakan ATmega328 sebagai basisnya. Di dalam Arduino Uno terdapat 14 pin sebagai *input/output digital* (6 pin diantaranya dapat digunakan sebagai *output PWM*), 6 pin *analog input*, 16 MHz *ceramics resonator*, sebuah koneksi USB, sebuah *socket power*, dan sebuah tombol *reset*. Arduino dapat dihidupkan dengan menggunakan kabel USB, sebuah *AC-to-DC adapter*, maupun baterai bertegangan 7 hingga 12 V. Program kapasitas memori Arduino Uno adalah 32 Kb dan 0.5 dari kapasistas memori digunakan untuk *bootloader*, serta memiliki SRAM sebesar 2 Kb

dan 1 Kb sebagai EEPROM [3]. Perangkat mikrokontroler Arduino dapat dilihat pada Gambar 2.1.

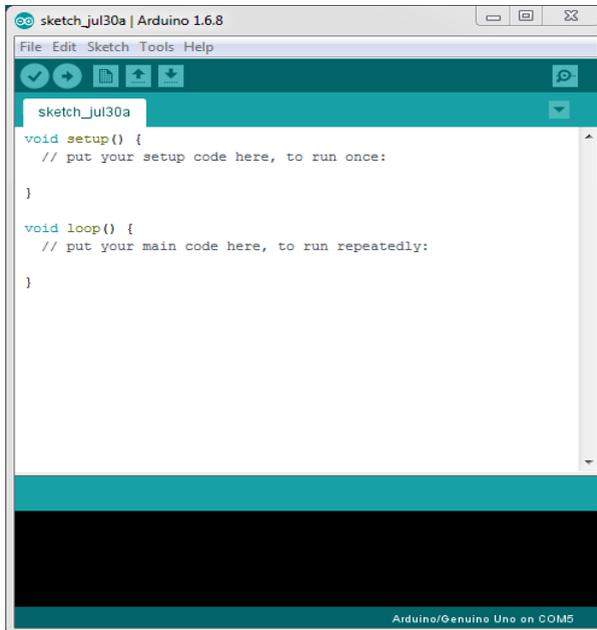


Gambar 2.1 Perangkat Mikrokontroler Arduino

Aplikasi Arduino sudah terdapat *serial monitor* yang digunakan sebagai aplikasi uji coba untuk melihat saat arduino menerima data koordinat dan mengirimnya menuju *server* menggunakan *GSM shield*. Lampu LED pada TX/RX akan menyala ketika proses pengiriman dan penerimaan data pada mikrokontroler Arduino.

2.2. IDE Arduino

IDE Arduino ini adalah sebuah perangkat lunak yang memudahkan kita mengembangkan aplikasi mikrokontroler mulai dari menuliskan kode sumber program, kompilasi, upload hasil kompilasi, dan uji coba secara terminal serial. Arduino ini bisa dijalankan di komputer dengan berbagai macam *platform* seperti Windows, Macintosh, ataupun linux karena IDE Arduino ini menggunakan basis bahasa Java dalam pemrogramannya [4].



Gambar 2.2. Antar muka IDE Arduino

Bahasa pemrograman yang digunakan oleh IDE Arduino didalam mengembangkan aplikasi mikrokontroler adalah C/C++ for Arduino. Tentunya terdapat style khusus yang membedakannya dari C/C++ biasa, yaitu:

- *void main(void)* sebagai fungsi program utama diganti dengan *void loop()*. Perbedaannya pada bahasa C biasa tidak terjadi *loop*, jadi harus ada *looping* yang ditambahkan misalnya *while(1){.....}*. Dalam Arduino secara otomatis fungsi *loop()* akan kembali lagi dari awal jika sudah dieksekusi hingga intruksi paling bawah.
- Ditambahkan fungsi *void setup(void)*, fungsi ini digunakan untuk inisialisasi mikrokontroler sebelum fungsi utama *loop()* dieksekusi.
- Tidak direpotkan dengan setting register-register, karena Arduino sudah memasukkannya kedalam *library*-nya dan

secara otomatis disesuaikan dengan jenis *board* Arduino berkenaan jenis mikrokontrolernya. Jadi *setup* untuk perangkat kerasnya menjadi mudah.

Tampak *user interface* aplikasi IDE Arduino yang telah *terinstall* pada laptop dengan *platform* Windows dapat dilihat pada Gambar 2.2.

Terdapat 5 tombol berurutan dari kiri ke kanan yang berada di *toolbar* yang dimana fungsinya adalah sebagai berikut:

- Memeriksa kode yang telah ditulis, apakah ada kesalahan pada saat kompilasi atau tidak
- Memeriksa kode yang telah ditulis sekaligus mengunggah kode tersebut ke dalam mikrokontroler Arduino.
- Membuat *sketch* kode baru.
- Membuka daftar *sketch* kode yang sudah tersimpan
- Menyimpan *sketch* kode yang telah dibuat.

2.3. Global Positioning System

Global Positioning System adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India. Sistem ini dikembangkan oleh Departemen Pertahanan Amerika Serikat, dengan nama lengkapnya adalah NAVSTAR GPS (kesalahan umum adalah bahwa NAVSTAR adalah sebuah singkatan, ini adalah salah, NAVSTAR adalah nama yang diberikan oleh John Walsh, seorang penentu kebijakan penting dalam program GPS). Kumpulan satelit ini diurus oleh 50th Space Wing Angkatan Udara Amerika Serikat. Biaya perawatan sistem ini sekitar US\$750 juta per tahun, termasuk penggantian satelit lama, serta riset dan pengembangan [5].

Akurasi atau ketepatan perlu mendapat perhatian bagi penentuan koordinat sebuah titik atau lokasi. Koordinat posisi ini akan selalu mempunyai 'faktor kesalahan', yang lebih dikenal dengan 'tingkat akurasi'. Misalnya, suatu alat menunjukkan sebuah titik koordinat dengan akurasi 15 meter, artinya posisi sebenarnya bisa berada dimana saja dalam radius 15 meter dari titik koordinat (lokasi) tersebut. Makin kecil angka akurasi (artinya akurasi makin tinggi), maka posisi alat akan menjadi semakin tepat. Harga alat juga akan meningkat seiring dengan kenaikan tingkat akurasi yang bisa dicapainya [5].

Pada pemakaian sehari-hari, tingkat akurasi ini lebih sering dipengaruhi oleh faktor sekeliling yang mengurangi kekuatan sinyal satelit. Karena sinyal satelit tidak dapat menembus benda padat dengan baik, maka ketika menggunakan alat, penting sekali untuk memperhatikan luas langit yang dapat dilihat.

2.4. *Shield* GPS/GPRS/GSM v3.0 DFRobot

Shield GPS/GPRS/GSM v3.0 adalah perangkat tambahan untuk mikrokontroler Arduino. Alat ini telah memiliki mesin *Quad-band* yang mampu bekerja pada frekuensi 1800 MHz dimana pada tipe EGSM mampu bekerja hingga 900 MHz dan pada tipe DCS GSM mampu bekerja hingga 850 MHz. Selain itu, alat ini didukung oleh teknologi GPS untuk navigasi satelit. Sehingga dapat diketahui posisi alat ini berada. Alat ini menggunakan chip SIM908 sehingga dapat mengirim pesan, menelpon, dan dapat menggunakan jaringan GSM. Chip yang tertanam ini dibuat oleh SIMCom dengan dikontrol melalui perintah AT untuk melakukan komunikasi baik untuk mengirim pesan, menelpon, dan menggunakan jaringan internet. Alat yang ditunjukkan pada Gambar 2.2 membutuhkan sumber daya tegangan untuk mengoperasikan teknologi GPS antara 3.0 V – 4.5 V sedangkan untuk GSM antara 3.2 V – 4.8 V.

Dengan ditambahkannya alat ini maka informasi posisi kendaraan bermotor akan dapat dipantau [6]. Alat ini dapat dilihat pada Gambar 2.3.



Gambar 2.3. GPS/GPRS/GSM Module 3.0

2.5. Regulator Step Down XL4015

XL 4015 *Step Down* adalah sebuah alat yang pada umumnya disebut dengan nama regulator. Fungsi utama dari alat ini adalah untuk mengatur tegangan maupun amper dari suatu *input* listrik menuju *output* yang sudah di tentukan berapa besaran tegangan dan amper yang diinginkan. Spesifikasi dari XL4015 adalah sebagai berikut:

- *Input Voltage range: 5-36v.*
- *Output Voltage Range: 1,25-32v.*
- *Output amper: 0-5A.*
- *Output power: 75W.*
- *High Efficiency up to 96%*
- Memiliki fungsi *thermal shutdown.*
- Memiliki fungsi pembatasan amper.
- Memiliki perlindungan terhadap korsleting.
- P x L x T = 68,2 x 38,8 x 15 mm

Dengan memakai alat yang ditunjukkan pada Gambar 2.4 ini diharapkan Arduino memiliki pasokan daya sesuai dengan yang dibutuhkan.



Gambar 2.4. Regulator XL4015 step-down

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN PERANGKAT LUNAK

Perancangan merupakan bagian terpenting dari pembuatan perangkat keras dan perangkat lunak yang berupa perencanaan – perencanaan secara teknis perangkat keras yang dibuat serta aplikasinya. Dalam bab perancangan perangkat lunak secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir. Berawal dari deskripsi umum hingga perancangan proses, alur dan implementasinya.

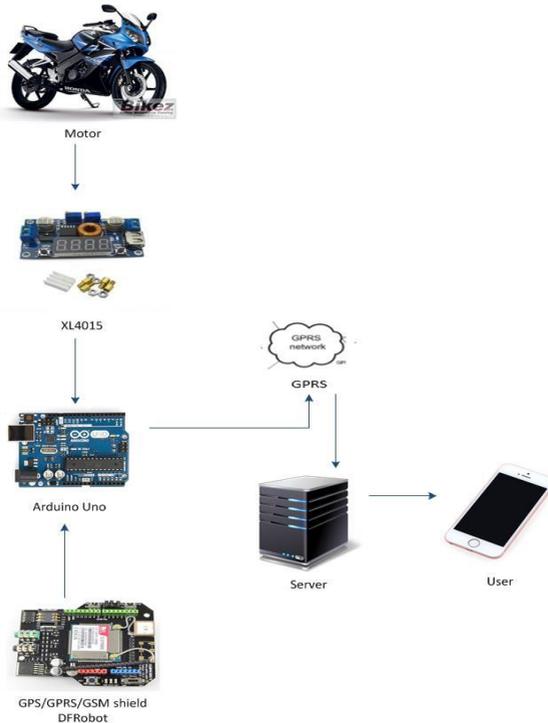
3.1. Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibangun prototype alat pelacakan kendaraan bermotor yang disertai sistem peringatan dini dengan menggunakan sensor GPS yang sudah menjadi satu dengan *GSM shield* dari dfrobot. Sistem peringatan dini akan aktif ketika motor dipindahkan posisinya yang menyebabkan *GPS shield* mengirimkan titik koordinat yang berbeda sehingga akan mengirimkan email kepada *user* sebagai bentuk notifikasi bahwa kendaraan bermotor miliknya telah berpindah tempat.

Pada Tugas Akhir ini dibutuhkan sensor untuk mengetahui apakah posisi kendaraan bermotor berpindah tempat atau tidak, yaitu sensor GPS yang telah tertanam menjadi satu dengan *GSM shield* milik dfrobot yang akan dihubungkan dengan mikrokontroler arduino sebagai pengolah data yang kemudian akan dikirimkan dengan menggunakan GPRS. Untuk pengiriman data penulis menggunakan sebuah perangkat GSM yang telah memiliki teknologi GPRS. Pengiriman data dari GPRS menuju ke hostingan *server* aplikasi dilakukan secara terus – menerus. Pada *server* aplikasi telah diberi batas tertentu yang akan menentukan apakah perlu mengirim notifikasi berupa email atau tidak. Jika ada indikasi kendaraan motor berpindah tempat, maka *server* akan mengirim email notifikasi kepada *user* agar dapat mengetahui bahwa kendaraan miliknya telah berpindah tempat.

3.2. Arsitektur Umum Sistem

Rancangan arsitektur dari sistem yang dibuat terlihat pada Gambar 3.1 yang terdapat beberapa komponen dan sensor utama.



Gambar 3.1. Arsitektur Umum Sistem

Pada umumnya rancangan arsitektur dari sistem ini adalah memasang mikrokontroler Arduino Uno yang telah dilengkapi dengan GSM *shield* dari DFRobot yang telah dilengkapi dengan GPS sehingga dapat menentukan titik koordinat dimana kendaraan bermotor berada sekaligus mengirimkan posisi kendaraan bermotor tersebut menuju *server*. Untuk catu daya dari mikrokontroler ini menggunakan sebuah aki yang terdapat di

motor yang akan dipasangkan dengan sebuah regulator untuk mengatur daya yang masuk ke dalam mikrokontroler.

a. GPS/GPRS/GSM Shield DFRobot

Shield ini berfungsi untuk memberikan informasi lokasi dimana alat ini berada dan juga berfungsi untuk mengirim data koordinat yang di dapat menuju *server* aplikasi. Alat ini bekerja dengan menggunakan sinyal yang dikirimkan melalui 24 satelit ke bumi. Sinyal ini berjenis gelombang mikro. Sehingga didapatkan lokasi terkini alat ini berada.

b. Mikrokontroler Arduino

Mikrokontroler Arduino berguna untuk memproses nilai yang didapatkan oleh sensor GPS. Pada mikrokontroler data dari sensor GPS akan dikirimkan ke *server* dengan menggunakan perangkat *shield* dari dfrobot. Lalu pada *server* akan di proses menjadi titik *pin point* pada peta dan *server* akan memberi notifikasi berupa email apabila posisi kendaraan bermotor berpindah tempat.

3.3. Perancangan Umum Sistem

Perancangan basis data menggambarkan struktur tabel yang akan digunakan dalam penyimpanan kebutuhan data sistem pada basis data. Pada Tabel 3.1 menggambarkan struktur tabel dari basis data yang digunakan dalam sistem ini.

Terdapat 2 tabel dengan masing-masing atribut yang digunakan untuk menyimpan kebutuhan sistem. Pada tabel Arduino di dalamnya terdapat *Id_arduino*, nama, dan email. *Email* disini adalah alamat *email* yang akan dipakai sebagai alamat pengiriman notifikasi pada saat kendaraan bermotor berpindah tempat. Sedangkan pada tabel koordinat berfungsi untuk menyimpan titik-titik koordinat posisi *longitude* dan *latitude* dari mikrokontroler Arduino sehingga dapat diketahui posisi dari mikrokontroler Arduino tersebut sedang berada dimana.

Tabel 3.1 Struktur Tabel Basis Data

Arduino		
Id_arduino	VarChar(250)	<pk>
nama	VarChar(250)	
email	VarChar(250)	
Koordinat		
Id_koordinat	BigInt(20)	<pk>
Id_arduino	VarChar(250)	
Latitude	VarChar(250)	
Longitude	VarChar(250)	

3.4. Perancangan Perangkat Keras

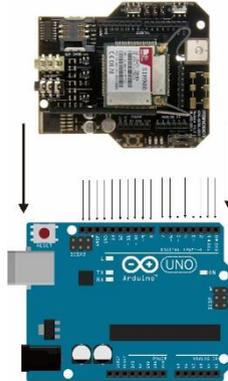
Pada sistem ini, perangkat keras yang digunakan sebagai berikut:

1. 1 buah mikrokontroler Arduino.
2. 1 buah *shield* GPS/GSM/GPRS v3.0 milik DFRobot.
3. 1 buah aki motor.
4. 1 buah Regulator *Step-Down* XL4015.
5. 1 buah SIM Telkomsel.
6. 1 buah laptop ACER 3820TG.

3.4.1. Perancangan Shield GPS/GSM/GPRS DFRobot

Pada Gambar 3.2 merupakan sebuah perancangan *shield* GSM/GPRS/GPS milik DFRobot dengan mikrokontroler Arduino. Untuk dapat memasang alat ini, mikrokontroler Arduino harus ditumpuk dan menjadi satu dengan GSM/GPRS/GPS *shield* milik DFRobot.

Sehingga akan menjadi satu kesatuan alat untuk dapat melakukan berbagai fungsi seperti mengirimkan data melalui kartu SIM yang dipasangkan atau juga mengambil data posisi titik koordinat dimana alat ini sedang berada.

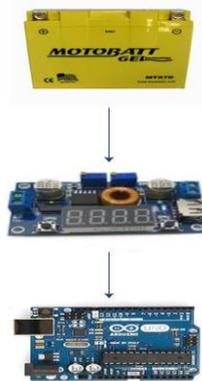


Gambar 3.2. Shield GPS/GSM/GPRS DFRobot dengan Arduino

Pada *shield* GPS/GSM/GPRS DFRobot ini memiliki beberapa fungsi yang dapat digunakan oleh mikrokontroler Arduino yaitu dapat memberikan lokasi terkini alat ini berada dengan menggunakan GPS, dan dapat mengirimkan data yang telah diolah oleh Arduino kedalam sistem pengiriman yang bernama GPRS. Alat ini bekerja dengan menggunakan AT command sehingga untuk dapat mengirimkan data membutuhkan sebuah kartu SIM (*subscriber identity Module*). Alat ini dapat bekerja dengan menggunakan suplay tenaga 12 Volt untuk dapat mengoperasikan GPS dan GPRS.

3.4.2. Perancangan Regulator Step Down XL4015

Regulator *step down* XL4015 diperlukan untuk membatasi tegangan yang diperlukan untuk catu daya mikrokontroler Arduino dari sebuah aki motor yang tegangannya dapat naik dan turun pada saat digunakan. Untuk memasang regulator ini diperlukan 2 kabel untuk *input* daya dari aki dan 2 kabel untuk *output* daya menuju arduino.



Gambar 3.3. Perancangan Regulator Step Down XL4015

Pada Gambar 3.3 di tunjukkan bahwa regulator *step down* XL4015 ditempatkan di antara aki motor dengan Arduino. Dengan pemasangan alat ini kita dapat mengatur tegangan yang akan masuk menuju *board* Arduino Uno menjadi stabil pada tegangan 12 Volt yang pada awalnya tegangan tersebut berubah-ubah mengikuti aki motor pada saat motor dinyalakan.

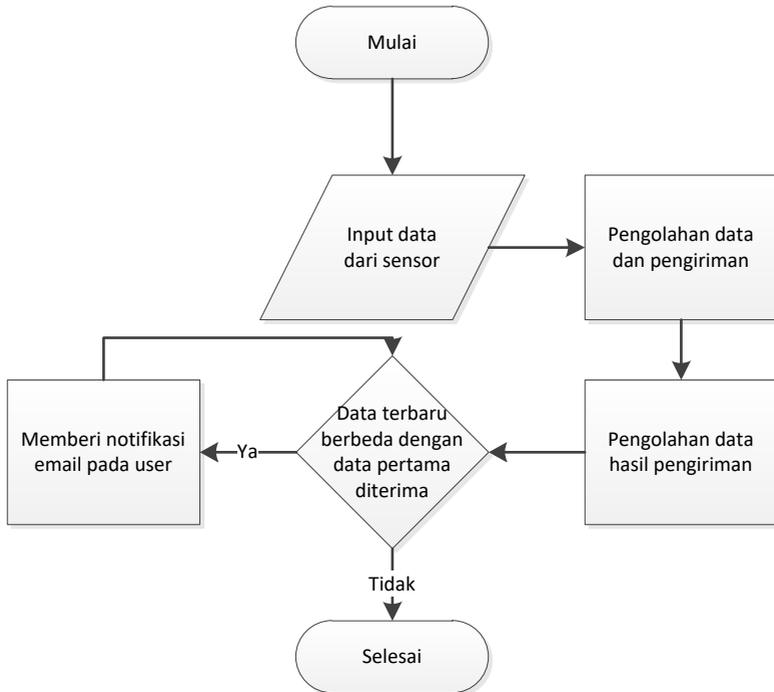
3.5. Perancangan Diagram Alir Sistem

Pada diagram alir prototype alat pelacakan motor dengan sistem peringatan dini ini menggambarkan urutan fungsionalitas sistem secara keseluruhan. Diagram alir sistem dapat dilihat pada Gambar 3.4.

Sistem diawali dengan membaca input data dari sensor GPS. Kemudian data yang dihasilkan oleh sensor akan diolah oleh mikrokontroler Arduino sehingga menghasilkan data *analog*. Data tersebut akan dikirim menuju *server* menggunakan perangkat komunikasi GSM yang sudah menjadi satu dengan GPS milik DFRobot.

Setelah data dikirim, *server* akan menerima data tersebut dan menyimpan data koordinat yang di terima ke dalam *database* yang sudah disediakan. Apabila data terbaru yang diterima oleh *server*

berbeda dengan data yang pertama kali diterima, maka server akan memberikan notifikasi berupa *email* kepada pemilik kendaraan bermotor.



Gambar 3.4. Diagram Alir Sistem

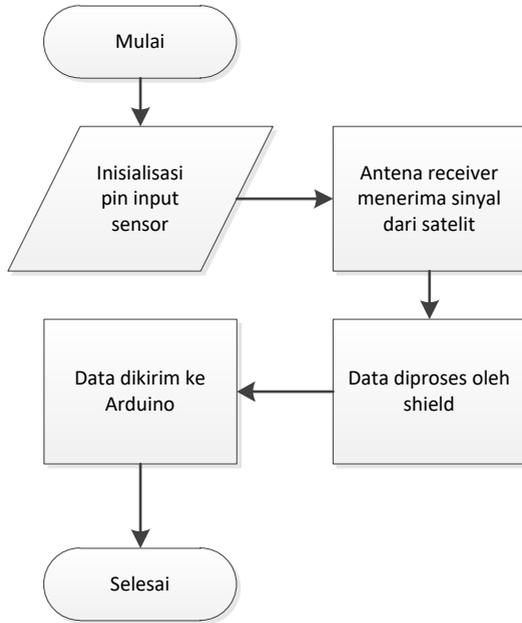
3.6. Diagram Alir Aplikasi Sistem

Alur setiap proses yang terdapat pada aplikasi digambarkan pada diagram alir, untuk memudahkan pemahaman secara garis besar proses yang ada pada sistem.

3.6.1. Diagram Alir Mendeteksi Lokasi

Untuk mendapatkan lokasi secara *realtime* dibutuhkan alat GPS. alat ini akan memberikan informasi lengkap dengan bantuan

satelit. Data hasil dari pengolahan GPS akan dikirimkan ke *microcontroller* Arduino untuk diproses. Diagram alir mendeteksi lokasi dapat dilihat pada Gambar 3.5.

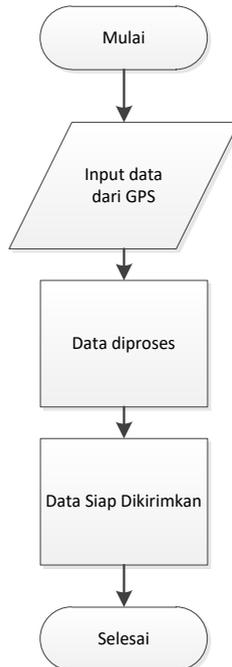


Gambar 3.5. Diagram alir mendeteksi lokasi

Untuk awal mulanya di perlukan inisialisasi pin input pada arduino agar dapat bekerja sama dengan GPRS/GPS *shield* dari DFRobot yang kemudian setelah inisialisasi dilakukan barulah GPRS/GPS *shield* dari DFRobot dapat berfungsi. Pada awalnya dimulai dengan diterimanya sinyal satelit oleh antenna *receiver* yang dimiliki oleh GPRS/GPS *shield* dari DFRobot yang dimana sinyal satelit tersebut akan diproses oleh *shield* dari DFRobot menjadi *raw data* dan kemudian dikirim menuju Arduino untuk dilakukan proses selanjutnya.

3.6.2. Diagram Alir Mikrokontroler

Setelah lokasi berhasil dideteksi dan sinyal satelit yang berhasil diterima oleh GPRS/GPS *shield* telah diproses juga oleh GPRS/GPS *shield*, maka hasil data *output* dari sensor GPRS/GPS *shield* dari DFRobot selanjutnya akan diproses pada mikrokontroler Arduino.



Gambar 3.6. Diagram alir mikrokontroler

Mikrokontroler ini akan mengolah data yang masih berbentuk *raw* data yang diperoleh dari sensor GPRS/GPS *shield* milik DFRobot dan setelah itu data yang telah diproses akan dikirimkan kepada *server* menggunakan jaringan GPRS yang juga dilakukan oleh GPRS/GPS *shield* milik DFRobot. Untuk melakukan koneksi ke dalam jaringan GPRS, digunakan kartu SIM (*Subscriber*

Identity Module) dengan Telkomsel sebagai provider yang digunakan.

Diagram alir mikrokontroler untuk memproses data lokasi yang diterima oleh GPRS/GPS *shield* sehingga menjadi data yang akan dikirim menuju server dapat dilihat pada Gambar 3.6.

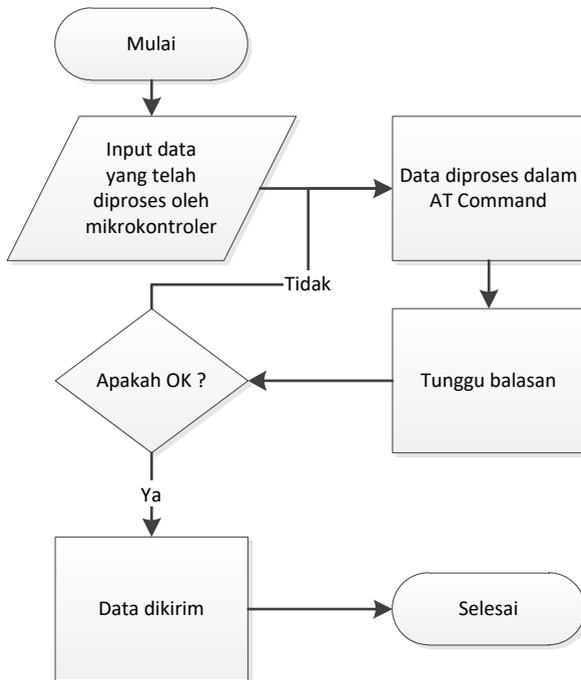
Pada saat mikrokontroler Arduino menerima data yang diberikan oleh GPRS/GPS *shield* milik DFRobot, mikrokontroler Arduino akan memproses data yang masih berbentuk *raw* tersebut menjadi data yang siap untuk dikirim menuju *server* yang nantinya akan dikirim melalui GPRS/GPS *shield* milik DFRobot.

3.6.3. Diagram Alir Pengiriman Data pada GPRS

Pada proses pengiriman data ini, sebelumnya didapatkan data keseluruhan dari mikrokontroler Arduino dan selanjutnya data tersebut akan diproses dalam AT COMMAND untuk dapat dikirimkan melalui GPRS. Diagram alir mengirim data ini dibutuhkan sinyal provider pada jangkauan BTS dapat dilihat pada Gambar 3.7.

Data yang sudah diproses oleh mikrokontroler Arduino akan menjadi inputan yang akan dikirim menuju *server*. Data tersebut akan diproses dalam AT Command yang nantinya menunggu balasan dari *server*. Apabila sudah OK, maka data akan segera dikirim menuju *server* yang digunakan melalui GPRS oleh GPRS/GPS/GSM *shield* dari DFRobot yang sebelumnya sudah dipasangkan kartu SIM dari provider yang menyediakan jaringan GPRS.

Pada tahap ini data titik koordinat posisi *longitude* dan *latitude* dari mikrokontroler Arduino sudah diproses oleh Arduino sehingga merupakan data yang sudah bisa dibaca oleh *server* dan juga dapat ditampilkan pada peta yang ada di aplikasi *mobile web*. Data dikirim dalam bentuk URL dan akan di-*parsing* pada saat berada di *server*.



Gambar 3.7. Diagram Alir Pengiriman Data

3.6.4. Diagram Alir Penerimaan Data Pada Server

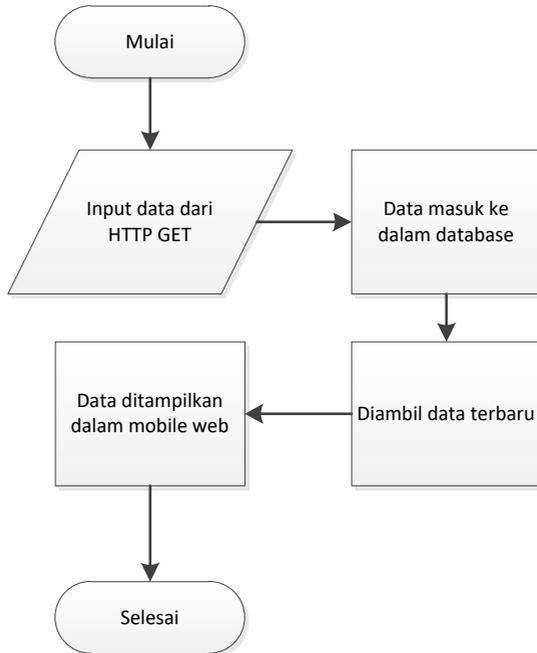
Setelah data dikirim oleh mikrokontroler Arduino melalui GPRS/GPS *shield* DFRobot menuju server, data diterima oleh *server* melalui jaringan GPRS. Supaya *server* dapat melakukan penerimaan data yang dikirim oleh mikrokontroler Arduino melalui GPRS/GPS *shield* DFRobot, maka *server* harus terhubung dengan koneksi internet.

Setelah komputer mendapatkan domain untuk dapat melakukan penerimaan data dari GPRS, maka data yang dikirim oleh mikrokontroler Arduino melalui GPRS/GPS *shield* DFRobot akan diproses dan data tersebut selanjutnya akan masuk ke dalam database dan nantinya data tersebut akan ditampilkan dalam

bentuk *pin point* pada google maps yang terdapat pada tampilan *mobile web* sehingga dapat memudahkan pengguna untuk *monitoring* lokasi dari kendaraan bermotor miliknya melalui aplikasi *mobile web* yang telah disediakan.

Proses yang terjadi dalam *server* ini hanya sebatas dalam penyimpanan data yang didapat dari hasil kiriman mikrokontroler Arduino yang dikirim melalui GPRS/GPS *shield* milik DFRobot dalam *database* yang tersedia dan kemudian ditampilkan dalam peta yang terdapat pada aplikasi *mobile web*.

Diagram alir inialisasi koneksi pada aplikasi *web* dapat dilihat pada Gambar 3.8.



Gambar 3.8. Diagram Alir Penerimaan Data Pada Server

Data diterima oleh server dari inputan HTTP GET yang dilakukan oleh mikrokontroler Arduino. Kemudian data yang diterima oleh *server* akan langsung dimasukkan ke dalam *database* yang sudah disediakan. Setelah data masuk ke dalam *database*, barulah di ambil data yang paling baru untuk kemudian ditampilkan ke dalam aplikasi *mobile web* yang dapat di akses oleh pengguna.

3.7. Perancangan Kebutuhan Arus Listrik

Perancangan kebutuhan arus listrik merupakan hal yang penting dalam membangun sistem ini. Hal ini diperlukan karena masing-masing alat membutuhkan suplai listrik atau daya yang berbeda sehingga semua komponen harus mendapatkan suplai daya minimum yang baik agar dapat menjalankan sistem yang ada. Apabila kebutuhan suplai listrik tidak terpenuhi, maka sistem ini tidak dapat berjalan dengan baik atau bahkan tidak akan berfungsi sebagaimana seharusnya.

Maka perlu dipersiapkan kebutuhan suplai listrik untuk masing-masing komponen yang terdapat dalam sistem yang akan digunakan. Kebutuhan suplai listrik atau daya masing-masing komponen yang diperlukan dalam sistem yang akan dipakai dapat dilihat pada Tabel 3.2.

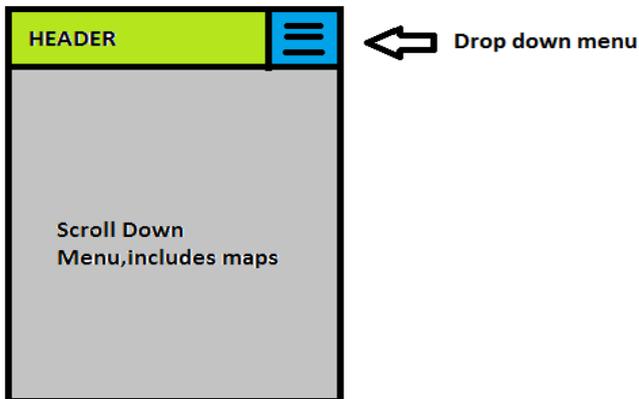
Tabel 3.2. Kebutuhan Suplai Listrik

Nama Perangkat	Kebutuhan Suplai Listrik	
	Arus Listrik	Tegangan Listrik
Arduino Uno	50mA	7-12v
GPS/GPRS/GSM <i>shield</i>	100mA	6-12v
Total	150mA	

3.8. Perancangan Antar Muka Aplikasi

Sistem berjalan secara otomatis dan *user* hanya menggunakan web aplikasi ini sebagai alat bantu untuk memantau posisi kendaraan bermotor miliknya dan juga untuk menghapus maps yang sudah terdapat *pin point* pada saat pelacakan sebelumnya. Rancangan antar muka aplikasi dibuat dalam bentuk aplikasi *mobile web*.

Pada Gambar 3.9 merupakan rancang bangun web aplikasi untuk mengetahui lokasi alat ini berada. Data untuk menampilkan lokasi ini didapatkan dari alat GPS yang telah dikirimkan melalui GPRS pada mikrokontroler Arduino. Pemberian lokasi ini ditampilkan secara *realtime* sehingga dapat mengetahui dimana posisi alat deteksi ini berada sekarang. Terdapat *marker* atau *pinpoint* pada peta yang telah tersedia di rancang bangun ini. *Marker* atau *pinpoint* ini merupakan *library* dari Google Maps untuk memudahkan *user* dalam mengetahui lokasi kendaraan bermotor.



Gambar 3.9. Tampilan pada Mobile Web

BAB IV IMPLEMENTASI

Bab implementasi akan membahas mengenai implementasi sistem pada perangkat keras dan perangkat lunak dari perancangan sistem yang telah dibahas pada bab 3.

4.1. Lingkungan Implementasi

Untuk mengimplementasi perangkat lunak ini digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah hostingan (*server*), mikrokontroler Arduino dan *shield* GPS/GSM/GPRS DFRobot. Spesifikasi dari perangkat-perangkat tersebut adalah sebagai berikut:

1. 1 buah regulator XL4015 *step-down*.
2. 1 buah Arduino Uno, ATMEGA 328P dengan 32 KB *flash* memori dan 2 KB memori SRAM.
3. 1 buah *shield* GPS/GSM/GPRS v3.0 milik DFRobot dengan power supply 6-12 V, Quad-Band 850/900/1800/1900 MHz, GPRS multi slot class 10, navigasi satelit dengan GPS.

4.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

1. Microsoft Windows 7 sebagai sistem operasi.
2. Arduino *Development Kit* versi 1.0.5 sebagai IDE untuk mengimplementasikan aplikasi Arduino.
3. Web di *hosting* di hostinger.co.id yang memakai:
 1. Apache HTTP Server sebagai aplikasi *web server* untuk menghasilkan halaman web yang benar kepada *user* berdasarkan kode PHP

2. MySQL sebagai aplikasi *database server* untuk mengolah dan menyimpan data Microsoft Visio 2010 untuk merancang diagram alir.

4.2. Implementasi Perangkat Keras

Pembuatan implementasi perangkat keras ini diawali dengan sebuah prototipe untuk menguji apakah perangkat keras dapat berfungsi sebagai pengambilan data sensor. Pada sistem ini, perangkat keras yang digunakan adalah sebagai berikut:

1. 1 buah mikrokontroler Arduino.
2. 1 buah regulator XL4015 *step-down*
3. 1 buah *shield* GPS/GSM/GPRS v3.0 DFRobot.
4. 1 buah SIM telkomsel.
5. 1 buah aki motor.

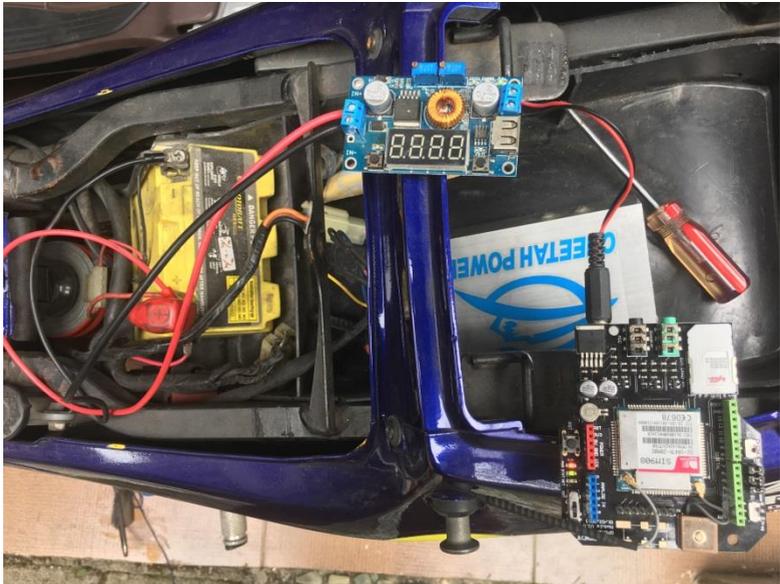
Untuk implementasi tiap-tiap perangkat keras dapat dilihat dalam subbab berikut ini.

4.2.1. Implementasi Perangkat *Shield* GPS/GPRS/GSM v3.0

Shield GPS/GPRS/GSM v3.0 DFRobot disini digunakan untuk mengambil koordinat posisi kendaraan bermotor berada dan juga sebagai media pengiriman data koordinat posisi kendaraan bermotor dari Arduino menuju *server*. Data yang diterima oleh *shield* GPS/GPRS/GSM v3.0 DFRobot ini adalah berupa data *longitude* dan *latitude* posisi kendaraan bermotor yang telah dipasangi prototype alat ini.

Data yang telah diterima dari sensor GPS akan diproses oleh mikrokontroler Arduino dan kemudian data akan dikirim menuju *server* menggunakan jaringan GPRS dari kartu SIM yang digunakan. Pada Tugas Akhir ini penulis menggunakan kartu SIM dengan Telkomsel sebagai *provider*nya.

Implementasi *shield* GPS/GPRS/GSM v3.0 DFRobot dapat dilihat pada Gambar 4.1.



Gambar 4.1. Arduino beserta GPS Shield dan Regulator Terpasang di Motor

4.2.2. Implementasi Perangkat Regulator XL4015

Perangkat regulator ini digunakan pada prototipe alat pelacakan motor dengan sistem peringatan dini ini sebagai sebuah alat atau perangkat yang membatasi atau mengatur besarnya tegangan dan amper yang masuk menuju mikrokontroler Arduino yang sudah dipasang dengan *shield* GPS/GPRS/GSM v3.0 dari DFRobot.

Perangkat ini dibutuhkan dikarenakan *input* catu daya untuk mikrokontroler Arduino berasal dari sebuah aki motor yang tegangannya naik dan turun setiap saat sehingga memerlukan sebuah perangkat regulator untuk menstabilkan atau membatasi arus listrik yang mengalir menuju mikrokontroler Arduino. Pada Tugas Akhir ini digunakan catu daya sebesar 12 V untuk tegangannya dengan kuat arus sebesar 1000 mA. Perangkat ini

nantinya dihubungkan dengan aki sepeda motor menggunakan 2 buah kabel yang di baut di aki sepeda motor dan untuk output menuju Arduino menggunakan 2 buah kabel yang dihubungkan dengan colokan *charger* yang sesuai dengan yang ada pada mikrokontroler Arduino.

Implementasi dari perangkat ini dapat dilihat pada Gambar 4.2.



Gambar 4.2. Regulator XL4015 terpasang di motor

4.3. Implementasi Perangkat Lunak

Implementasi pada perangkat lunak terbagi menjadi dua bagian, yaitu mikrokontroler Arduino dan aplikasi berbasis web. Setiap bagian akan dijelaskan lebih lanjut pada masing-masing subbab.

4.3.1. Implementasi Pada Mikrokontroler Arduino

1. Inisialisasi Arduino dan Pin yang digunakan.

Proses inisialisasi dilakukan di awal ketika perangkat pertama kali dinyalakan. Proses inisialisasi hanya dilakukan sekali saat perangkat menyala.

	<code>//Init the driver pins for GSM</code>
	<code>function</code>
1	<code>pinMode(3, OUTPUT);</code>
2	<code>pinMode(4, OUTPUT);</code>
3	<code>pinMode(5, OUTPUT);</code>
	<code>//Output GSM Timing</code>
4	<code>digitalWrite(5, HIGH);</code>
5	<code>delay(1500);</code>
6	<code>digitalWrite(5, LOW);</code>
7	<code>Serial.begin(9600);</code>

Gambar 4.3. Inisialisasi Pada Mikrokontroler Arduino

Penjelasan kode sumber dari Gambar 4.3 adalah sebagai berikut:

- `pinMode` adalah inisialisasi pin yang digunakan untuk mendapatkan data dari *shield* GPS/GPRS/GSM DFRobot.
- `digitalWrite` pada pin nomor 5 adalah *output* GSM agar dapat berjalan. Inisialisasi awal diisi dengan nilai *high*, kemudian *low*.
- `Serial.begin` adalah fungsi untuk menjalankan Arduino serta memasukkan nilai *baudrate* menjadi 9600.
- Nilai *baudrate* adalah nilai yang digunakan mikrokontroler arduino untuk berkomunikasi dengan laptop yang terhubung dengan kabel USB.

2. Membaca Data GPS

Proses membaca *input*-an dari sensor pada sistem ini merupakan hal yang penting karena *input* dari sistem ini berasal dari sensor yang ada, yaitu sensor GPS.

1	<code>Serial.println("AT+CGPSINF=0");</code>
2	<code>read_String();</code>
3	<code>strtok(inData, ",");</code>
4	<code>strcpy(longitude, strtok(NULL,</code> <code>",,")); // Gets longitude</code>
5	<code>strcpy(latitude, strtok(NULL,</code> <code>",,")); // Gets latitude</code>
6	<code>Result();</code>

Gambar 4.4. Proses Membaca Data GPS Arduino

Penjelasan kode sumber dari Gambar 4.4 adalah sebagai berikut:

- AT CGPSINF adalah sebuah AT commands yang akan memberikan umpan balik informasi lokasi GPS.
- `read_String` merupakan fungsi untuk menyimpan informasi yang diterima setelah AT commands dijalankan.
- Pada baris ke 3 hingga 5 adalah proses untuk mendapatkan data *latitude* dan *longitude* dari informasi yang didapat setelah AT commands dijalankan.
- `result` adalah sebuah fungsi untuk memroses konversi nilai *latitude* dan *longitude* dari informasi yang didapat. Fungsi ini akan mengambil 5 angka dari raw data yang didapat dan di proses menjadi nilai *longitude* dan *latitude* yang sesungguhnya.

3. Proses inisialisasi GPS

Proses membaca *input*-an dari GPS merupakan bagian terpenting dalam pengambilan nilai ini. *Input* ini telah diinisialisasi pada *microcontroller* Arduino sehingga dapat diambil nilai tersebut.

	<code>//Configuracion en Inicializacion</code>
	<code>GPS</code>
1	<code>delay(1000);</code>
2	<code>Serial.println("AT+CGPSIPR=9600");//</code> <code>(set the baud rate)</code>
3	<code>delay(5000);</code>
4	<code>Serial.println("AT+CGPSPWR=1");</code>
5	<code>// (turn on GPS power supply</code> <code>delay(5000);</code>
6	<code>Serial.println("AT+CGPSRST=1");</code>
7	<code>// (reset GPS in autonomy mode)</code> <code>delay(10000); //delay para</code> <code>esperar señal del GPS</code>

Gambar 4.5. inisialisasi GPS pada mikrokontroler Arduino

Penjelasan kode sumber pada Gambar 4.5 adalah sebagai berikut:

- AT CGPSIPR adalah AT *commands* untuk *setup* nilai *baudrate*.
- AT CGPSPWR adalah AT *commands* untuk mengaktifkan *power supply* untuk GPS.
- AT CGPSRST adalah AT *commands* untuk me-*reset* GPS dari *autonomy mode*.

4. Proses mengirim data melalui GPRS

Proses untuk mengirimkan data dari seluruh nilai yang ada pada mikrokontroler Arduino akan dilakukan oleh GPRS. Proses ini sebelumnya akan dilakukan ketika nilai tersebut telah tersedia. Dibutuhkan sebuah SIM untuk dapat

melakukan pengiriman data ke jaringan. Penulis menggunakan SIM dari *provider* Telkomsel.

1	<code>if(corlat <-7 && corlon >112)</code>
2	<code>{</code> <code>if(corlon > longSekitar corlon <</code> <code>longSekitar2 corlat > latSekitar corlat <</code> <code>latSekitar2)</code> <code>{</code>
3	<code>Serial.print("AT+HTTTPARA=\"URL\", \"http://guru</code> <code>hadi.16mb.com/index.php/server/lokasi?id_arduin</code> <code>o=123&latitude=");</code>
4	<code>Serial.print(corlat, 5);</code>
5	<code>Serial.print("&longitude=");</code>
6	<code>Serial.print(corlon, 5);</code>
7	<code>Serial.print("\r\n\r\n");</code>
8	<code>delay(1000);</code>
9	<code>Serial.println("AT+HTTPACTION=0"); //</code>
10	<code>GET action</code> <code>delay(1000);</code> <code>}</code> <code>}</code>

Gambar 4.6. Pengiriman data menuju server

Penjelasan kode sumber pada Gambar 4.6 adalah sebagai berikut:

- Pada baris 1 dan 2 adalah untuk menentukan bahwa koordinat yang akan dikirim tidak berada di luar Indonesia dan bukan perpindahan jarak dekat.
- Pada baris 3 hingga baris 7 adalah proses penetapan nilai yang didapat ke dalam parameter HTTP yang nantinya akan dipanggil.
- AT HTTPACTION adalah AT *commands* yang berfungsi untuk memanggil parameter HTTP yang telah ditentukan.

4.3.2. Implementasi Pada Aplikasi Web

1. Proses penyimpanan data dalam database

Proses penyimpanan data ini dilakukan apabila sudah ada data koordinat *longitude* dan *latitude* yang diterima oleh *server* yang berasal dari mikrokontroler Arduino. Proses ini mem-*parsing* data yang awalnya berupa sebuah URL menjadi masing-masing data *longitude* dan *latitude*.

```

public function lokasi()
{
1   $koordinat = array(
2     'id_koordinat' => '',
3     'id_arduino'=>$this->input->get('id_arduino'),
4     'longitude'=>$this->input->get('longitude'),
5     'latitude'=>$this->input->get('latitude'),
        );

6   $data = null;
7   if($this->input->get('id_arduino') && $this-
->input->get('longitude') && $this->input-
->get('latitude'))
        {
8       $data = $this->ta_model-
->add_koordinat($koordinat);
        }

9       if($data!=null)
        {
10          $temp = $this->ta_model-
->get_koordinat($this->input->get('id_arduino'),
-$this->input->get('latitude'), $this->input-
->get('longitude'));

                //tambahan
11          $temp1 = $this->ta_model-
->get_koordinat2($this->input-
->get('id_arduino'));
12          if($temp1->latitude!=$temp-
->latitude && $temp1->longitude != $temp-
->longitude)

```

13	<pre> { \$this->sendmail(\$arduino- >email); } } } } </pre>
----	---

Gambar 4.7. Penerimaan data dan penyimpanan ke database

Penjelasan kode sumber pada Gambar 4.7 adalah sebagai berikut:

- Pada baris 1 sampai 5 adalah proses pemisahan data yang pada awalnya berbentuk sebuah *url* agar didapatkan id Arduino, *longitude*, dan *latitude* dari data yang dikirimkan oleh mikrokontroler Arduino.
- Pada baris 6 sampai 8 adalah proses memasukkan data yang telah dipisah menuju *database* yang telah disediakan.
- Pada baris 9 hingga 12 adalah proses membandingkan titik koordinat *longitude* dan *latitude* yang diperoleh pertama kali dengan titik koordinat *longitude* dan *latitude* yang paling akhir diperoleh oleh mikrokontroler Arduino.
- Pada baris 13 dibutuhkan apabila terdapat perbedaan antara titik koordinat *latitude* dan *longitude* awal kali di terima oleh *server* dengan yang paling akhir diterima oleh *server* maka *server* akan mengirimkan *email* menuju pemilik kendaraan bermotor tersebut.

2. Proses penampilan data menuju *mobile web*

Data yang telah dikirim oleh mikrokontroler Arduino menuju *server* melalui GPRS/GPS *shield* DFRobot dan telah dimasukkan menuju *database* yang telah disediakan sebelumnya akan ditampilkan dalam bentuk *mobile web* yang dapat di akses oleh

ponsel yang memiliki sistem operasi iOS, Android, ataupun Windows.

```

1 public function index()
2     {
3         $this->load->library('googlemaps');
4
5         $list=$this->ta_model->get_koordinatnew("123");
6
7         if($list!=null
8         {
9             foreach($list as $l)
10                {
11
12                    $config['center'] = ".$l->latitude.",
13                    ".$l->longitude;
14                    $config['zoom'] = 'auto';
15                    $this->googlemaps->initialize($config);
16
17                    $i = 1;
18                    foreach($list as $l)
19                    {
20                        $marker = array();
21
22                        $marker['position'] = ".$l->latitude.", ".$l->longitude;
23
24                        $marker['icon'] =
25                        'http://chart.apis.google.com/chart?chst=d_map_
26                        pin_letter&chld=".$i++."|9999FF|000000';
27
28                        $this->googlemaps->add_marker($marker);
29
30                    }
31                }
32            else
33            {
34                $config['center'] = '.-7.256072.', '112.74259';

```

17	<code>\$config['zoom'] = 'auto';</code>
18	<code>\$this->googlemaps->initialize(\$config);</code> <code> }</code>
19	<code>\$data['map'] = \$this->googlemaps->create_map();</code>
20	<code>\$data['arduino'] = \$this->ta_model->get_arduino('123');</code>
21	<code>\$this->load->view('map_view', \$data);</code> <code> }</code>

Gambar 4.8. Proses menampilkan data pada aplikasi mobile web

Penjelasan kode sumber pada Gambar 4.8 adalah sebagai berikut:

- Pada baris 1 hingga 2 berfungsi untuk memuat *library* dari Google Maps yang akan digunakan pada aplikasi *mobile web*.
- Pada baris ke 3 berfungsi untuk melakukan *query* data titik koordinat *longitude* dan *latitude* yang terdapat didalam *database*.
- Pada baris ke 4 berfungsi untuk memastikan apabila *database* titik koordinat *latitude* dan *longitude* telah terisi dari data yang dikirimkan oleh mikrokontroler Arduino.
- Pada baris 5 hingga 8 berfungsi agar peta pada aplikasi *mobile web* selalu menunjukkan posisi atau *marker* terakhir dari posisi kendaraan bermotor yang telah dipasangkan mikrokontroler Arduino.
- Pada baris 9 hingga 14 berfungsi untuk menampilkan posisi atau *marker* pada setiap titik koordinat

longitude dan *latitude* yang terdapat didalam *database*.

- Pada baris 15 hingga 18 berfungsi apabila ternyata *database* dari titik koordinat posisi *latitude* dan *longitude* masih kosong sehingga peta hanya akan menampilkan tampak peta kota Surabaya secara keseluruhan.
- Pada baris 19 hingga 21 berfungsi untuk menampilkan peta yang telah siap untuk ditampilkan pada aplikasi *mobile web*.

3. Fungsi untuk mengirimkan *email* kepada pemilik kendaraan bermotor

Fungsi ini dibutuhkan untuk mengirimkan *email* kepada pemilik kendaraan bermotor yang telah dipasangkan mikrokontroler Arduino agar pemilik kendaraan bermotor tersebut mengetahui apabila kendaraan bermotor miliknya berpindah tempat dari posisi awal dia meletakkan kendaraan bermotor miliknya.

Fungsi ini digunakan atau dipanggil di baris 13 pada Gambar 4.7.

1	<code>public function sendmail(\$email)</code>
	<code>{</code>
2	<code> \$this->load->library('email'); // load email</code>
	<code> library</code>
3	<code> \$this->email->from('admin@guruhadi.16mb.com',</code>
	<code> 'Motorcycle Thief Tracker');</code>
4	<code> \$this->email->to(\$email);</code>
5	<code> \$this->email->subject('ALERT!!!');</code>
6	

7	<pre> \$this->email->message('Lokasi kendaraan anda berpindah, buka http://guruhadi.16mb.com untuk mengetahui lokasi kendaraan anda saat ini'); \$this->email->send(); } </pre>
---	---

Gambar 4.9. Fungsi untuk mengirim email

Penjelasan kode sumber pada Gambar 4.9 adalah sebagai berikut:

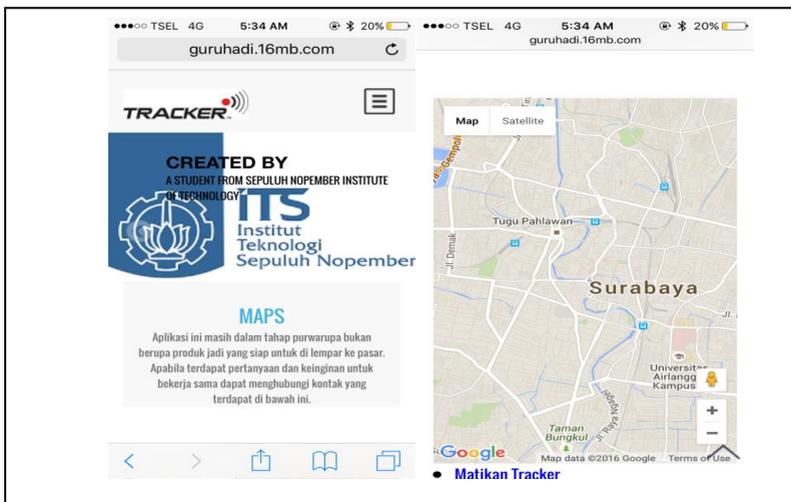
- Pada baris 1 adalah deklarasi dari fungsi 'sendmail'.
- Pada baris ke 2 berfungsi untuk memuat *library email* yang akan digunakan.
- Pada baris ke 3 untuk menentukan alamat *email* yang digunakan untuk mengirim *email* menuju *email* pemilik dari kendaraan bermotor yang telah dipasangkan mikrokontroler Arduino.
- Pada baris ke 4 menentukan alamat *email* yang akan dikirimkan, alamat *email* tersebut diambil dari database yang sesuai dengan id Arduino yang digunakan.

4. Tampilan muka pada aplikasi *mobile web*.

Implementasi antar muka perangkat lunak yang digunakan oleh pengguna hanya terdapat pada aplikasi *mobile web*. Pada aplikasi *mobile web* tersebut memiliki satu halaman antar muka yang dapat di *scroll-down* untuk melihat keseluruhan tampilan halaman antar muka pada aplikasi *mobile web* yang ada. Halaman tersebut digunakan sebagai alat bantu pelacakan posisi dimana kendaraan bermotor tersebut sedang berada.

Pada implementasinya Halaman pada aplikasi web ini digunakan untuk menampilkan lokasi dari prototipe alat pelacakan motor ini berada dan tombol apabila kita ingin menghapus *history* dari titik-titik koordinat tempat kendaraan bermotor tersebut telah berpindah. Aplikasi *mobile web* untuk tampilan antar-muka dengan pengguna ini menggunakan Google Maps untuk

menunjukkan peta tempat dimana kendaraan bermotor yang telah dipasangkan alat ini sedang berada. Implementasi antar muka perangkat lunak dapat dilihat pada Gambar 4.10



Gambar 4.10. Implementasi halaman web untuk menampilkan lokasi

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Pada bab uji coba dan evaluasi akan dibahas mengenai uji coba dari segi fungsionalitas dan performa dari aplikasi. Uji coba fungsionalitas dan performa akan dibagi ke dalam beberapa skenario uji coba.

5.1. Uji Coba Fungsionalitas

Uji coba fungsionalitas merupakan sebuah pengujian terhadap jalannya fungsi utama yang ada pada aplikasi. Pada aplikasi ini, uji coba terbagi menjadi dua bagian, yaitu perangkat GPS, dan sistem peringatan dini apabila kendaraan berpindah posisi. Fungsionalitas utama dari sistem ini terdapat pada mikrokontroler Arduino dan perangkat GPRS. Arduino berfungsi sebagai media untuk membaca data dari sensor serta pengolahan data yang telah didapat dari sensor. Perangkat GPRS berfungsi sebagai media pengiriman data dari mikrokontroler beserta perangkat sensor ke aplikasi web sebagai hasil pelacakan keberadaan kendaraan bermotor. Pada uji coba ini, terdapat skenario untuk menguji fungsionalitas sistem, meliputi:

1. Deteksi posisi oleh GPS

Pendeteksian posisi digunakan aplikasi untuk membantu pelacakan dimana posisi kendaraan bermotor yang ditinggal secara *real-time*.

2. Sistem Peringatan dini

Sistem peringatan dini ini digunakan aplikasi untuk membantu *user* mengetahui adanya pergerakan kendaraan bermotor miliknya dengan cara memberikan notifikasi berupa *email* kepada *user*.

5.1.1. Lingkungan Uji Coba

Pada subbab lingkungan uji coba, dijelaskan mengenai gambaran lingkungan yang digunakan sebagai uji coba aplikasi. Uji coba dilakukan pada lokasi terbuka agar GPS dapat dengan mudah mendapat *locking* posisi pada satelit. Lingkungan uji coba mencakup beberapa spesifikasi, yaitu:

1. Rumah makan cepat saji *McDonalds* mulyosari bagian *outdoor*.
2. Alfaekspres Jl. Arief Rahman Hakim
3. Jalan raya dharmahusada hingga jalan semolowaru tengah X.
4. Perjalanan memutar di daerah jalan raya mulyosari.
5. Perangkat sensor.
6. 1 buah laptop untuk memantau serial monitor.
7. 1 buah *handphone* untuk melihat apakah data masuk ke *database server*.

Seperti yang telah dijelaskan, untuk lokasi uji coba berada di lokasi terbuka. Pada tempat terbuka digunakan untuk menguji alat GPS berdasarkan informasi yang diterima oleh sinyal satelit. Hasil data pada pengujian akan dikirimkan melalui sistem GPRS.

5.1.2. Uji Coba Deteksi Lokasi pada GPS

Pada uji coba perangkat GPS dilakukan pendeteksian informasi lokasi yang dikirimkan oleh sinyal satelit. Tujuan dari uji coba deteksi untuk mengetahui kecepatan memberikan informasi lokasi yang dikirimkan oleh sinyal satelit dan seberapa akurat perangkat GPS dalam memberikan informasi lokasi terkini.

Skenario uji coba akan dibagi menjadi 2 kegiatan yaitu pada saat GPS melakukan perjalanan dan ketika berhenti di satu titik lokasi. Dalam uji coba pendeteksian lokasi pada perangkat GPS yang diuji adalah waktu saat alat dinyalakan hingga menampilkan data informasi lokasi yang dikirimkan oleh sinyal satelit ke *receiver* pada perangkat GPS dan keakuratan nilai yang diterima.

Hasil uji coba dijelaskan sebagai berikut:

1. Uji Coba Statis satu lokasi

Uji coba ini dilakukan hanya pada satu lokasi tertentu dengan menggunakan dua kegiatan yang berbeda, kemudian akan diketahui perbedaan tingkat akurasi dan kecepatan pemberian informasi yang dihasilkan oleh perangkat GPS. hasil uji coba dapat dilihat pada Tabel 5.1.

Tabel 5.1. Hasil Uji Coba Kecepatan Lock GPS

No	Tempat	Waktu	Keterangan
1	McDonalds Mulyosari	323 detik	Diam,malam hari
2	Alfaekspres Ir. Soekarno	284 detik	Diam,malam hari
3	Area Parkir McDonalds Mulyosari	127 detik	Diam,sore hari
4	Jl. Dharmahasada menuju Rs. Dr. Soetomo	197 detik	Bergerak,siang hari,hujan
5	Jl. Nginden Semolo menuju Terminal Bratang	183 detik	Bergerak,siang hari,hujan

Tabel 5.2. Hasil Uji Coba Informasi Lokasi GPS

No.	Latitude	Longitude	Keterangan
1.	-7.266871	112.796411	McDonalds Mulyosari
2.	-7.289951	112.776926	Alfaekspres Ir. Soekarno
3.	-7.266818	112.796253	Area Parkir McDonalds Mulyosari
4.	-7.265922	112.760971	Jl. Dharmahasada menuju Rs. Dr. Soetomo
5.	-7.299690	112.764871	Jl. Nginden Semolo menuju Terminal Bratang

Pada Tabel 5.1 merupakan tabel hasil uji coba kecepatan untuk *lock* GPS dengan sinyal satelit. Waktu yang dibutuhkan memiliki rentan nilai waktu 49- 196 detik. Hal ini disebabkan dari lokasi pengujian dan sumber tenaga yang dibutuhkan oleh GPS. nilai waktu ini akan memberikan tingkat kecepatan GPS dalam memberikan informasi lokasi. Pada tabel 5.2 merupakan hasil *output* nilai dari GPS yang diuji berbagai tempat.

2. Uji Coba Progressif beberapa lokasi

Pada uji coba progressif beberapa lokasi ini alat dipasangkan pada motor yang dipakai untuk kegiatan uji coba yang kemudian akan dibawa berkeliling di jalanan kota Surabaya.

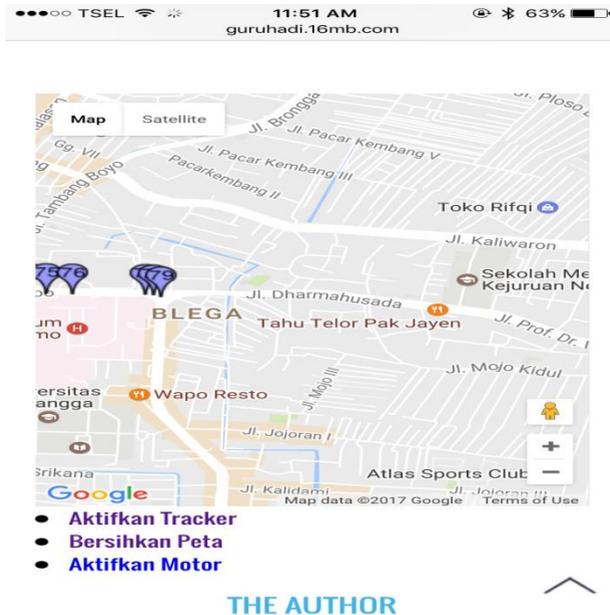
Pada awalnya penulis berangkat dari rumah menuju titik lokasi awal untuk memulai uji coba ini. Penulis memilih lokasi SPBU Pertamina yang berada di Jalan Raya Dharmahasada sebagai titik lokasi awal untuk memulai uji coba ini. Pada saat sampai di titik awal lokasi untuk memulai uji coba ini keadaan sekitar sedang hujan dan berawan.



Gambar 5.1. Lokasi Awal Uji Coba

Pada Gambar 5.1 dapat terlihat lokasi awal uji coba sedang hujan dan berawan. Pada lokasi inilah penulis memasang mikrokontroler Arduino yang telah dipasangkan dengan GPRS/GPS *shield* dari DFRobot. Setelah mikrokontroler Arduino dipasangkan, penulis langsung mengendarai motor untuk melalui rute uji coba yang sudah disiapkan.

Karena penulis langsung mengendarai motor tanpa menunggu mikrokontroler untuk locking posisi awal, maka titik koordinat posisi *longitude* dan *latitude* awal belum terdeteksi dan data titik koordinat posisi *longitude* dan *latitude* awal tidak terkirim dan tersimpan didalam *server*. *Screenshot* data titik koordinat posisi *longitude* dan *latitude* yang pertama kali terdeteksi dan terkirim ke *server* dapat dilihat pada Gambar 5.2.



Gambar 5.2. Titik Koordinat Posisi Awal

Pada Gambar 5.2 dapat dilihat bahwa titik koordinat posisi *longitude* dan *latitude* awal yang terkirim dan tersimpan pada *server* bukan di SPBU pertamina yang bertempat di jalan Raya Dharmahusada tempat awal penulis memasang mikrokontroller Arduino namun titik awal koordinat di dapatkan pada pertigaan antara Jalan Raya Dharmahusada, Jalan Prof. Dr. Moestopo, dan Jalan Karang Menjangan.

Titik awal koordinat posisi *longitude* dan *latitude* yang terkirim dan tersimpan di *server* di tandai dengan *marker* yang memiliki nomor 79 atau dalam hal ini *marker* dengan angka terbesar adalah *marker* posisi awal dimana mikrokontroler Arduino pertama kali berhasil mendapatkan koordinat dari posisi kendaraan bermotor.



Gambar 5.3. Posisi Arduino Mendapat Koordinat Pertama

Posisi pada saat Arduino mendapatkan titik koordinat posisi *longitude* dan *latitude* pertama kali dapat dilihat pada Gambar 5.3.

Selanjutnya penulis melanjutkan rute perjalanan menuju perempatan antara Jalan Prof. Dr. Moestopo, Jalan Dharmawangsa, dan Jalan Tambang Boyo yang dimana terdapat RSUD dr. Soetomo dan RS Husada Utama di dekat perempatan tersebut.

Cuaca masih hujan gerimis di sekitar penulis. Kondisi sekitar jalan pada perempatan antara Jalan Prof. Dr. Moestopo, Jalan Dharmawangsa, dan Jalan Tambang Boyo dapat dilihat pada Gambar 5.4 yang penulis ambil di bawah ini.



Gambar 5.4. Kondisi Sekitar Perempatan Dharmawangsa

Selanjutnya penulis melanjutkan perjalanan melewati Jalan Gubeng Masjid yang terletak tepat di depan stasiun kereta api gubeng, Surabaya. Setelah melalui jalan gubeng masjid, di lanjutkan perjalanan menuju Jalan Nias dan kemudian Jalan Raya Kertajaya.

Sayangnya penulis tidak sempat mengambil dokumentasi di Jalan Raya Kertajaya dikarenakan tidak memungkinkannya keadaan untuk penulis mengambil dokumentasi. Namun pada saat sedang berada di jalan nias dapat dilihat dokumentasinya pada Gambar 5.5. Kondisi jalan nias pada saat dilewati hujan gerimis dan berawan, namun di Jalan Nias dan Jalan Kertajaya yang akan dilewati juga memiliki pepohonan yang cukup lebat.

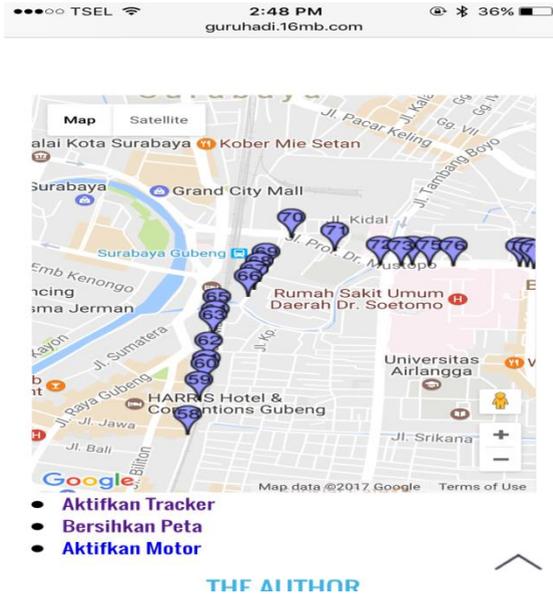


Gambar 5.5. Kondisi Sekitar Jalan Nias

Pada saat melewati Jalan Prof. Dr. Moestopo, Jalan Gubeng Masjid, dan Jalan Nias mikrokontroler Arduino dapat bekerja dengan baik dalam mendapatkan titik-titik koordinat posisi *longitude* dan *latitude* posisi kendaraan bermotor saat bergerak menyusuri jalanan tersebut.

Namun pada saat akan memasuki jalan kertajaya hingga perempatan pertama jalan kertajaya, mikrokontroler Arduino tidak melakukan *update* titik koordinat posisi *longitude* dan *latitude* kendaraan bermotor yang telah dipasangkan mikrokontroler Arduino. Hal ini dikarenakan kendaraan bermotor yang telah dipasangkan mikrokontroler Arduino terjebak macet dan berjalan sangat lambat hingga berhenti sehingga koordinat perpindahan

yang terjadi di anggap masih dalam lingkup koordinat sekitar dan tidak mengalami perpindahan koordinat yang signifikan.



Gambar 5.6. Titik koordinat yang kendaraan bermotor

Pada Gambar 5.6 dapat dilihat titik-titik koordinat posisi *longitude* dan *latitude* kendaraan bermotor yang telah dipasangkan mikrokontroller Arduino pada saat melewati jalan Prof. Dr. Moestopo, jalan gubeng masjid, dan jalan nias.

Dapat dilihat pada gambar terdapat *marker* yang memiliki nomor 58. Pada *marker* yang memiliki nomor 58 itulah mikrokontroller Arduino yang dipasangkan pada kendaraan bermotor mulai tidak mengirimkan data menuju *server* dikarenakan motor yang digunakan untuk uji coba terjebak macet yang mengakibatkan motor bergerak sangat lambat hingga berhenti sehingga perpindahan koordinat tidak signifikan dan dianggap masih dalam lingkup koordinat sekitar.

Kemudian perjalanan dilanjutkan dari jalan kertajaya menuju Jalan Pucang Anom Timur, Jalan Ngagel Jaya, Jalan Ngagel Jaya Utara, Jalan Ngagel Tama, dan Jalan Ngagel Jaya Selatan.

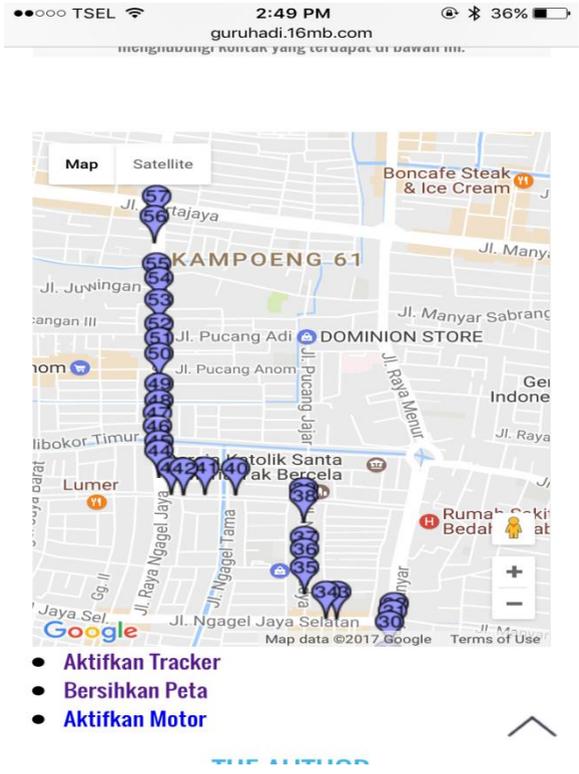


Gambar 5.7. Perempatan Jalan Ngagel Jaya Selatan

Pada perempatan jalan antara Jalan Ngagel Jaya Selatan dengan Jalan Raya Manyar, dan Jalan Manyar Rejo kondisi sekitar masih berawan walaupun hujan sudah mulai berhenti seperti yang ditunjukkan pada Gambar 5.7.

Pada saat melewati Jalan Pucang Anom Timur dan Jalan Ngagel Jaya mikrokontroller Arduino dapat mendapatkan titik posisi koordinat *longitude* dan *latitude* dari kendaraan bermotor dengan mudah tanpa ada hambatan yang berarti. Namun ketika kendaraan bermotor yang telah dipasangkan mikrokontroller Arduino mulai melewati Jalan Ngagel Jaya Utara dan Jalan Ngagel Madya, motor yang digunakan untuk uji coba kembali harus

berjalan dengan lambat hingga berhenti pada perempatan antara Jalan Ngagel Jaya Utara dan Jalan Ngagel Madya dikarenakan adanya *traffic light* di daerah tersebut.



Gambar 5.8. Titik koordinat di jalan ngagel

Pada Gambar 5.8 dapat dilihat bahwa ada perpindahan yang cukup jauh dari *marker* nomor 40 dan *marker* nomor 39. Hal ini disebabkan karena motor yang dipakai untuk uji coba berjalan dengan lambat sehingga titik koordinat yang diterima oleh mikrokontroler Arduino perbedaannya tidak signifikan dan masih

dianggap masih di dalam lingkup koordinat sekitar oleh mikrokontroler Arduino.

Namun setelah motor yang digunakan uji coba melewati *traffic light* tersebut posisi koordinat *longitude* dan *latitude* didapatkan kembali dengan normal.



Gambar 5.9. Titik koordinat di jalan nginden semolo

Kemudian perjalanan dilanjutkan menuju jalan raya manyar dan kemudian jalan nginden semolo yang juga terdapat beberapa pepohonan yang lebat.

Pada Gambar 5.9 dapat dilihat bahwa jarak antar *marker* pengiriman titik koordinat posisi *longitude* dan *latitude* kendaraan bermotor sedikit melebar seperti antar *marker* yang bernomor 18 dan 17, 16 dan 15, juga 15 dan 14.

Hal ini disebabkan oleh kecepatan yang membuat koordinat perpindahan antar *longitude* dan *latitude* yang didapat oleh

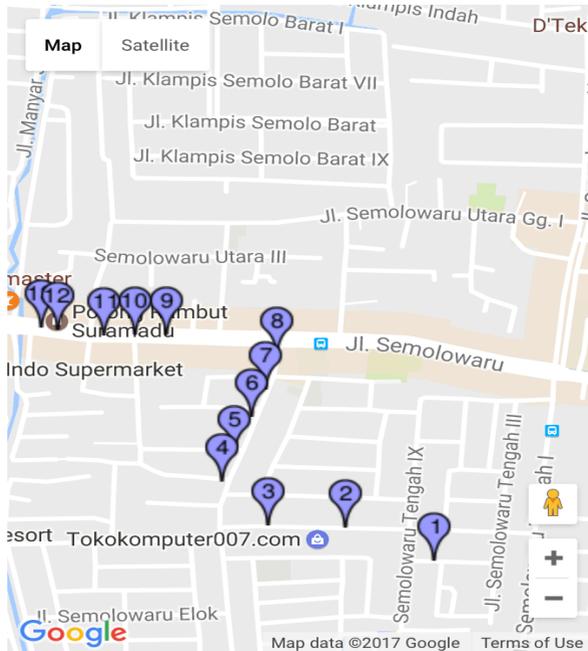
mikrokontroler Arduino sedikit lebih jauh antar *marker* perpindahannya. Namun demikian mikrokontroler Arduino masih dapat memberikan informasi lokasi dari kendaraan bermotor yang digunakan untuk uji coba.



Gambar 5.10. Jalan Semolowaru Tengah

Perjalanan dilanjutkan menuju Jalan Semolowaru, Jalan Semolowaru Selatan II, hingga sampai di depan rumah penulis di Semolowaru Tengah X seperti yang dapat dilihat di Gambar 5.10.

Pada Gambar 5.11 dapat dilihat titik posisi koordinat *longitude* dan *latitude* kendaraan bermotor antara *marker* bernomor 4, 3, 2, dan 1 agak berjauhan dibandingkan yang lainnya. Hal ini disebabkan karena sinyal telkomsel sebagai kartu provider yang digunakan untuk mengirim data melalui jaringan GPRS agak sedikit jelek untuk di area semolowaru. *Marker* bernomor 1 menunjukkan posisi terakhir atau posisi terbaru dari kendaraan bermotor yang digunakan untuk uji coba.

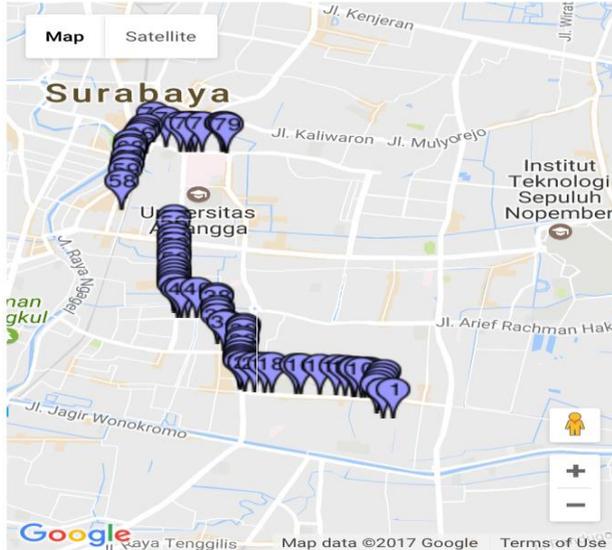


- **Aktifkan Tracker**
- **Bersihkan Peta**
- **Aktifkan Motor**

THE AUTHOR

Gambar 5.11. Titik koordinat di jalan semolowaru

Pada Gambar 5.12 dapat dilihat keseluruhan titik lokasi koordinat *longitude* dan *latitude* yang didapat pada saat pelaksanaan uji coba yang dimulai di Jalan Raya Dharmahasada dan berakhir di Jalan Semolowaru Tengah X.



- Aktifkan Tracker
- Bersihkan Peta
- Aktifkan Motor

THE AUTHOR

Aplikasi ini merupakan suatu Tugas Akhir dari mahasiswa yang

Gambar 5.12. Titik Koordinat Keseluruhan

5.2. Uji Coba Sistem Peringatan Dini

Uji coba sistem peringatan dini ini dilakukan untuk melihat perilaku yang dihasilkan oleh aplikasi *web* pada Tugas Akhir. Uji coba ini bertujuan untuk mengetahui seberapa besar kekuatan sistem, serta batasan-batasan apa saja yang menjadi dasar dari sistem.

Pada uji coba sistem peringatan dini ini diharapkan aplikasi dapat memberikan notifikasi *email* kepada *user* apabila kendaraan bermotor berpindah tempat.

Pada uji coba sistem peringatan dini ini, aplikasi akan menerima dua data titik koordinat yang berbeda dan akan di hitung berapa waktu yang diperlukan untuk mengirimkan notifikasi kepada *user* melalui *email* pada saat data koordinat kedua di terima oleh *server*.

1. Uji Coba kecepatan pengiriman *email*.

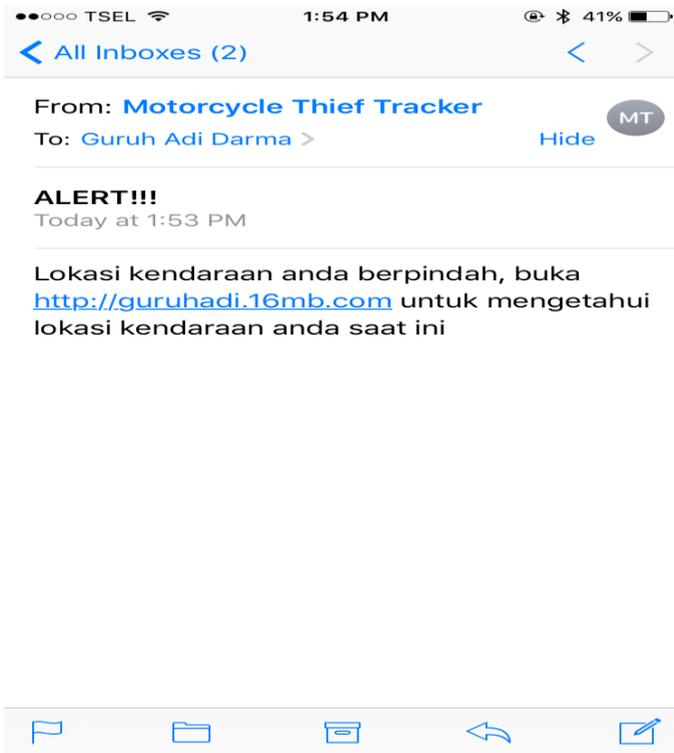
Pada hasil uji coba kecepatan pengiriman *email*, dilakukan percobaan sebanyak 5 kali dengan 2 titik koordinat yang berbeda dan dihitung waktu yang diperlukan untuk mengirim notifikasi *email* kepada *user*. Dari hasil uji coba didapatkan hasil pada Tabel 5.3

Tabel 5.3. Hasil uji coba pengiriman data

Percobaan	Waktu yang diperlukan
1	2 detik
2	5 detik
3	3 detik
4	5 detik
5	1 detik

Hasil uji coba ini dapat dilihat pada Tabel 5.3 dimana dilakukan 5 kali uji coba perpindahan lokasi dan dihitung waktunya dari saat titik koordinat *longitude* dan *latitude* kedua didapat hingga *email* diterima oleh pengguna kendaraan bermotor yang digunakan untuk uji coba ini.

Pada Gambar 5.13 adalah isi dari *email* yang dikirimkan oleh *server* menuju alamat *email* dari pemilik kendaraan bermotor yang telah dipasangkan mikrokontroller Arduino yang digunakan sebagai objek dari uji coba.



Gambar 5.13. Isi email yang dikirim

2. Uji Coba Perpindahan Kendaraan melebihi batas aman

Pada uji coba perpindahan kendaraan melebihi batas aman ini akan dilakukan perpindahan pada motor yang digunakan untuk uji coba melewati batas aman kendaraan tersebut sehingga di anggap berpindah tempat, dimana batas aman yang di tentukan adalah 20 meter. Sehingga apabila motor bergerak melebihi 20 meter maka mikrokontroler Arduino akan secara otomatis mengirimkan data koordinat terbaru menuju *server* dan *server* akan otomatis mengirimkan *email* notifikasi kepada pengguna.

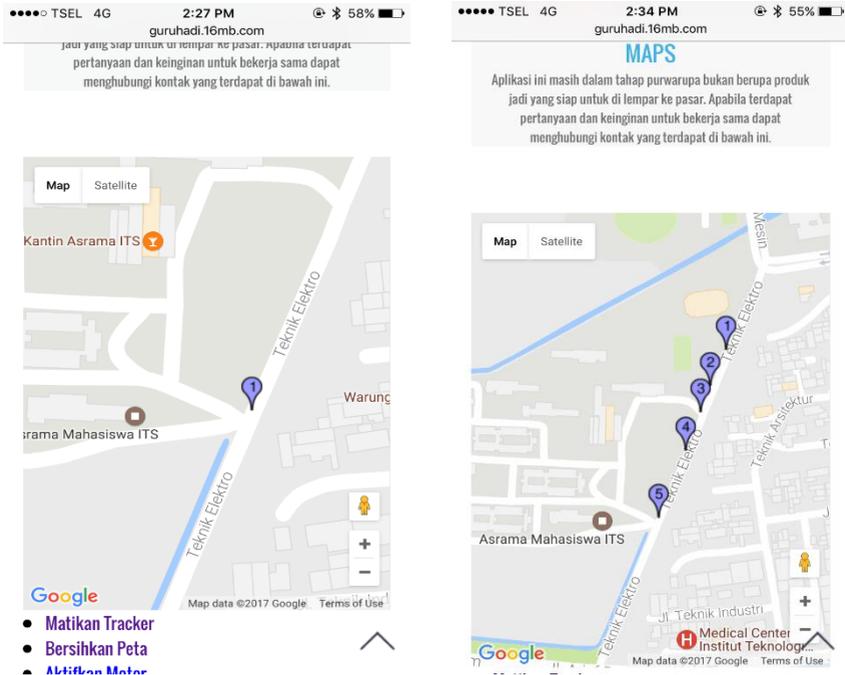
Uji coba dilakukan di 3 tempat yang berbeda. Yang pertama dilakukan uji coba di depan asrama mahasiswa ITS hingga lapangan futsal PLN ITS. Keadaan pada saat uji coba dapat dilihat dalam Gambar 5.14



Gambar 5.14 Titik Awal dan Akhir Uji Coba Pertama

Pada Gambar 5.14 adalah titik awal memulai uji coba dan titik akhir dari uji coba yang dilaksanakan. Titik awal uji coba yang pertama ini dimulai di depan asrama mahasiswa ITS dan diakhiri pada tempat di depan lapangan futsal PLN ITS.

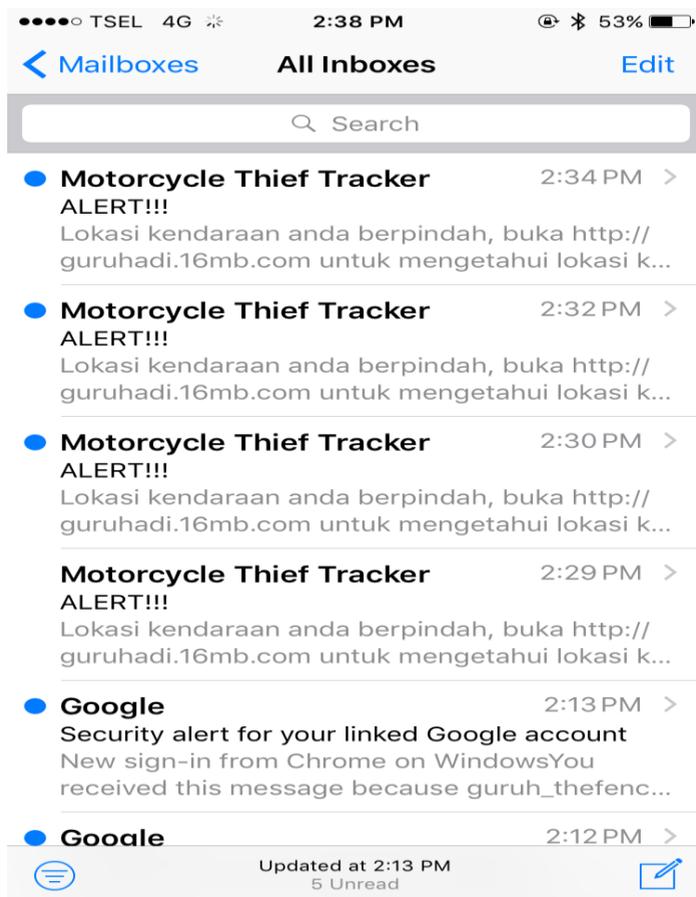
Pada uji coba di tempat pertama ini di dapatkan 4 titik perpindahan seperti yang dapat dilihat pada Gambar 5.15 yang berarti motor telah mengalami perpindahan yang melebihi lingkup koordinat sekitar sehingga mikrokontroller Arduino mengirimkan titik koordinat yang baru menuju *server*.



Gambar 5.15 Koordinat Uji Coba Pertama

Kemudian saat terjadi perpindahan koordinat dari motor yang digunakan untuk uji coba maka *server* akan mengirimkan *email* yang berisi pemberitahuan apabila posisi motor telah berpindah tempat dari tempat awal.

Isi dari *email* yang masuk dapat dilihat pada gambar 5.13 dan pada Gambar 5.16 kita dapat melihat *email* yang masuk ke pemilik kendaraan bermotor setelah terjadi adanya perpindahan motor yang digunakan sebagai uji coba.



Gambar 5.16 Kiriman *email* Uji Coba Pertama

Lokasi kedua dalam uji coba ini dilakukan di depan danau jurusan Statistika sebagai titik awal memulai uji coba dan berakhir di jalan depan jurusan Teknik Elektro. Lokasi dari uji coba yang kedua ini dapat dilihat pada Gambar 5.17

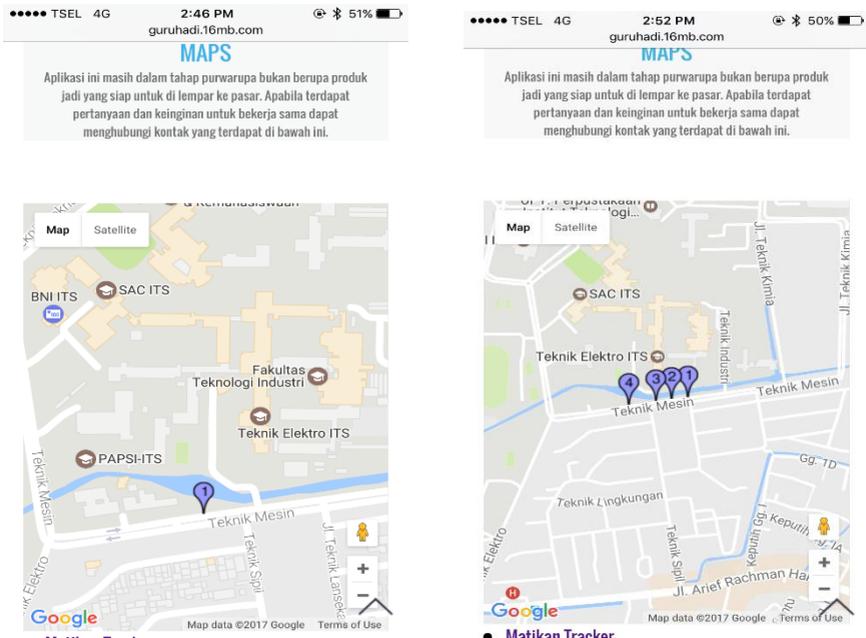


Gambar 5.17 Titik Awal dan Akhir uji coba kedua

Pada uji coba kedua yang di lakukan di jalan di depan danau jurusan Statistika hingga jalan di depan jurusan Teknik Elektro ini di dapatkan 3 kali perpindahan dari motor yang digunakan untuk uji coba.

Yang berarti juga motor telah 4 kali melakukan perpindahan yang memiliki titik koordinat *longitude* dan *latitude* yang melebihi lingkup dari koordinat sekitar yang telah ditetapkan sebelumnya.

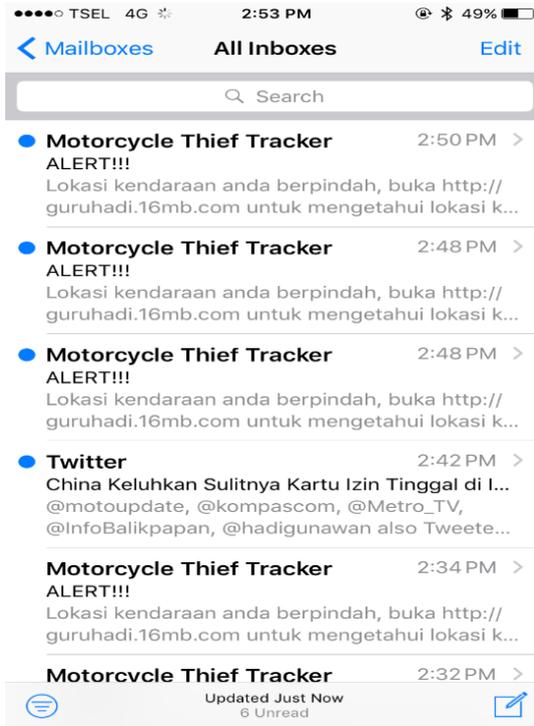
Pada Gambar 5.18 dapat dilihat dimana titik awal mikrokontroler Arduino mendapatkan titik koodinat *longitude* dan *latitude* awal dari posisi motor yang digunakan sebagai uji coba dan juga titik- titik koordinat *longitude* dan *latitude* posisi perpindahan motor yang digunakan dalam uji coba.



Gambar 5.18 Koordinat Uji Coba Kedua

Setelah terjadi perpindahan dari motor yang digunakan untuk uji coba, pengguna atau pemilik kendaraan bermotor kembali mendapatkan *email* yang berisi notifikasi bahwa kendaraan bermotor miliknya telah berpindah tempat dari tempat awal posisi titik koordinat yang didapatkan.

Email yang masuk ke alamat *email* pemilik kendaraan bermotor tersebut dapat dilihat pada Gambar 5.19



Gambar 5.19 Kiriman *email* Uji Coba Kedua

Kemudian uji coba dilanjutkan menuju tempat yang ketiga sekaligus terakhir yaitu di lingkungan jurusan Teknik Informatika. Uji coba dilakukan di gerbang jurusan Teknik Informatika melewati parkiran mobil mahasiswa dan dosen jurusan Teknik Informatika dan terakhir di parkiran motor mahasiswa jurusan Teknik Informatika.

Titik awal uji coba dilakukan di gerbang jurusan Teknik Informatika dan titik terakhir uji coba berada pada parkiran motor

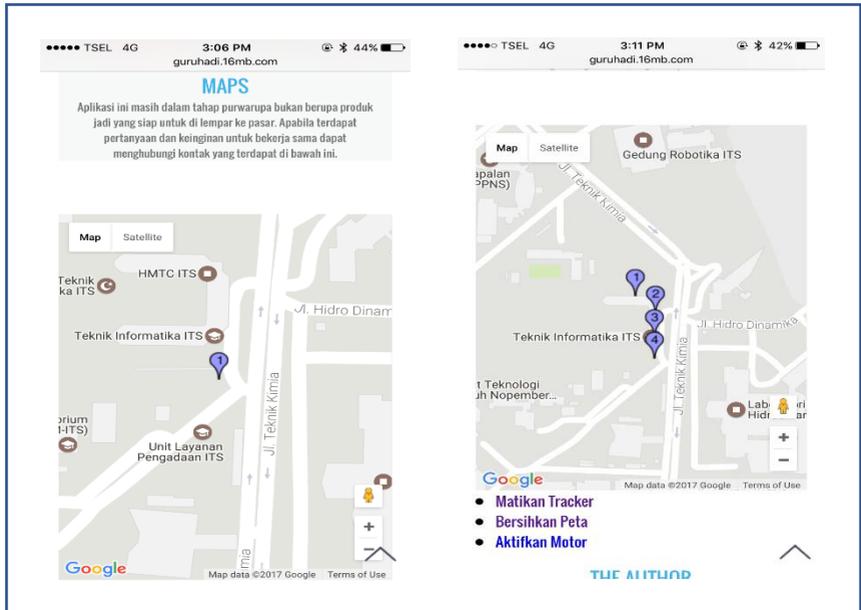
mahasiswa jurusan Teknik Informatika. Titik awal dan akhir dari uji coba terakhir ini dapat dilihat pada Gambar 5.20



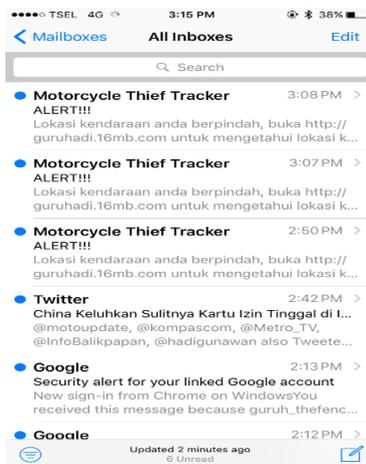
Gambar 5.20 Titik Awal dan Akhir Uji Coba Ketiga

Pada uji coba ketiga sekaligus terakhir ini didapatkan sejumlah 3 kali perpindahan seperti pada uji coba kedua. Di ujicoba yang terakhir ini motor melakukan 3 kali perpindahan titik posisi koordinat *longitude* dan *latitude* yang melebihi dari lingkup titik koordinat sekitar yang telah ditetapkan.

Titik koordinat awal dilakukan uji coba dan titik-titik koordinat perpindahan yang dilakukan dalam uji coba yang dilakukan di area sekitar jurusan Teknik Informatika sekaligus *email* yang dikirim menuju alamat *email* pemilik kendaraan bermotor ini dapat dilihat pada Gambar 5.21 dan juga Gambar 5.22



Gambar 5.21 Koordinat Uji Coba Ketiga



Gambar 5.22 Kiriman email Uji Coba Ketiga

Pada uji coba perpindahan kendaraan melebihi batas aman yang dilakukan di 3 tempat yang berbeda ini didapatkan hasil dari uji coba pada Tabel 5.4

Tabel 5.4 Hasil Uji Coba Sistem Peringatan Dini

<i>Percobaan</i>	<i>Koordinat yang didapat</i>		<i>Status koordinat</i>	<i>Status pengiriman notifikasi</i>
<i>I</i>	-7.28901	112.79254	Berhasil didapatkan ·	Sukses.
	-7.28830	112.79281		
	-7.28790	112.79296		
	-7.28762	112.79306		
	-7.28724	112.79322		
<i>2</i>	-7.28604	112.79513	Berhasil didapatkan ·	Sukses.
	-7.28596	112.79567		
	-7.28590	112.79598		
	-7.28587	112.79630		
<i>3</i>	-7.28008	112.79760	Berhasil didapatkan ·	Sukses.
	-7.27979	112.79760		
	-7.27948	112.79761		
	-7.27926	112.79739		

Berdasarkan Tabel 5.4 diketahui prototipe alat dapat mengambil dan mengirim titik koordinat posisi kendaraan bermotor dan mengirimkan notifikasi berupa *email* menuju pemilik kendaraan bermotor apabila kendaraan bermotor miliknya berpindah tempat melebihi koordinat ruang lingkup sekitar.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan yang didapat selama pengerjaan Tugas Akhir ini beserta saran-saran untuk pengembangan Tugas Akhir ini di masa depan

6.1. Kesimpulan

Dari hasil pengamatan selama masa pengerjaan Tugas Akhir ini dapat diambil beberapa kesimpulan, yaitu:

1. Arduino mendapatkan titik koordinat posisi dengan memanfaatkan GPRS/GPS/GSM *shield* milik DFRobot.
2. Sistem peringatan dini dijalankan pada saat perpindahan posisi motor keluar dari koordinat ruang lingkup sekitar.
3. Data diolah menggunakan mikrokontroler Arduino yang menghasilkan titik koordinat *longitude* dan *latitude* yang kemudian dikirimkan ke server memanfaatkan jaringan GPRS.

Dengan demikian prototipe mikrokontroler Arduino dalam penelitian Tugas Akhir ini dapat digunakan sebagai sistem peringatan dini untuk kendaraan bermotor yang berpindah tanpa sepengetahuan pemilik kendaraan bermotor tersebut.

6.2. Saran

Saran ini dibuat untuk menambahkan fitur-fitur yang dapat ditambahkan pada prototipe alat pelacakan motor dengan sistem peringatan dini ini untuk pengembangan alat di masa yang akan datang.

1. Pada penelitian selanjutnya dapat ditambahkan fitur user pada aplikasi *web* agar dapat dipakai *multi-user*.
2. Menambah fitur keamanan seperti dapat mematikan kendaraan bermotor dari aplikasi *web*.
3. Mengganti GPS sensor agar dapat mendapat lokasi dan *locking* posisi dengan tepat dan cepat.

4. Menambah fitur *on/off* untuk tracker agar pada saat tidak digunakan mikrokontroler tidak mengirim secara terus-menerus posisi koordinat mikrokontroler

DAFTAR PUSTAKA

- [1] R. Rahadiyansyah, August 2015. [Online]. Available: <http://oto.detik.com/otoshow/read/2015/08/29/151350/3004585/1506/audi-indonesia-paling-konsumtif-di-asean>.
- [2] Arduino, "Arduino," Maret 2013. [Online]. Available: <http://www.arduino.cc/>. [Accessed 7 Maret 2014].
- [3] Arduino, "Arduino-Introduction," Arduino, 2016. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Accessed 14 11 2013].
- [4] B. Siswoyo, "Belaja," June 2012. [Online]. Available: <http://bsiswoyo.lecture.ub.ac.id/2012/06/belajar-arduino-pengantar/>.
- [5] Wikipedia, October 2015. [Online]. Available: https://id.wikipedia.org/wiki/Sistem_Pemosisi_Global#cite_note-GPS_overview_from_JPO-3.
- [6] DFrobot, "GPS/GPRS/GSM Module V 3.0," DFRobot, shanghai, 2014.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

```
1  const int anPin = 3;
2
3  char aux_str[30];
4  char aux;
5
6  int sum = 0;
7  int avgrange = 60;
8  char latitude[20];
9  char longitude[20];
10 char lastLatitude[20];
11 char lastLongitude[20];
12 float corlat;
13 float corlon;
14 float latSekitar = 0, latSekitar2 = 0;
15 float longSekitar = 0, longSekitar2 = 0;
16
17 char inChar;
18 int index;
19 char inData[200];
```

Gambar 7.1. Inisialisasi variabel yang dibutuhkan

```
1  void setup()
2  {
3      //Init the driver pins for GSM function
4      pinMode(3, OUTPUT);
5      pinMode(4, OUTPUT);
6      pinMode(5, OUTPUT);
```

```

7 //Output GSM Timing
8 digitalWrite(5,HIGH);
9 delay(1500);
10 digitalWrite(5,LOW);
11
12 Serial.begin(9600);
    // Use these commands instead of the
hardware switch 'UART select' in order to
enable each mode
13 // If you want to use both GMS and GPS.
enable the required one in your code and
disable the other one for each access.
14 digitalWrite(3,LOW);//enable GSM TX, RX
digitalWrite(4,HIGH);//disable GPS TX,
16 RX
17 Serial.println("Starting up...");
18 delay(10000);
19
20 start_GSM();
21
22 delay(5000);
23
24 start_GPS();
25 delay(5000);
26
27 }

```

Gambar 7.2. Fungsi setup arduino + gsm shield

```

1
2 void loop()
3 {
4 //readHTTP();

```

5	<code>//delay(1000);</code>
6	<code>read_GPS();</code>
7	<code>delay(1000);</code>
8	<code>send_GPRS();</code>
9	<code>delay(1000);</code>
10	<code>getKoorSekitar();</code>
11	<code>delay(1000);</code>
12	<code>//send_SMS();</code>
13	<code>//delay(1000);</code>
14	<code>}</code>

Gambar 7.3. Fungsi utama pada arduino yang selalu looping

1	
2	<code>void start_GSM()</code>
3	<code>{</code>
4	<code> //Configuracion GPRS Claro Argentina</code>
5	<code> Serial.println("AT");</code>
6	<code> delay(5000);</code>
7	<code> Serial.println("AT+SAPBR=3,1,\"APN\", \"tel</code>
8	<code> konsel\");</code>
9	<code> delay(5000);</code>
10	<code> Serial.println("AT+SAPBR=3,1,\"CONTYPE\", \"</code>
11	<code> \"GPRS\"); //diganti dengan CONTYPE</code>
12	<code> delay(5000);</code>
13	<code> Serial.println("AT+SAPBR=1,1");</code>
14	<code> delay(5000);</code>
15	<code> Serial.println("AT+SAPBR=2,1");</code>
16	<code> delay(5000);</code>
17	<code> Serial.println("AT+HTTPINIT");</code>

16	delay(5000);
17	Serial.println("AT+HTTTPARA=\"CID\",1");
18	delay(5000);
19	}

Gambar 7.4. Konfigurasi GPRS pada arduino untuk koneksi internet

1	void send_GPRS()
2	{
3	if(corlat <-7 && corlon >112)
4	{
	if(corlon > longSekitar corlon <
	longSekitar2 corlat > latSekitar
5	corlat < latSekitar2)
6	{
	Serial.print("AT+HTTTPARA=\"URL\", \"http://g
	uruhadi.16mb.com/index.php/server/lokasi?id_
7	arduino=123&latitude=");
8	//Serial.print(latitude);
9	Serial.print(corlat, 5);
10	Serial.print("&longitude=");
11	//Serial.print(longitude);
12	Serial.print(corlon, 5);
13	Serial.print("\n\r\n\r\n");
14	delay(1000);
	Serial.println("AT+HTTPACTION=0"); //
15	GET action
16	delay(1000);
17	}
18	Else
19	{

```
20     Serial.print("&latitude");
21     Serial.print(corlat, 5);
22     Serial.print("\r\n");
23     Serial.print("&longitude");
24     Serial.print(corlon, 5);
25     Serial.print("\r\n");
26     Serial.print("&latitude Sekitar + :");
27     Serial.print(latSekitar, 5);
28     Serial.print("\r\n");
29     Serial.print("&latitude Sekitar - :");
30     Serial.print(latSekitar2, 5);
31     Serial.print("\r\n");
32     Serial.print("&longitude Sekitar + :");
33     Serial.print(longSekitar, 5);
34     Serial.print("\r\n");
35     Serial.print("&longitude Sekitar - :");
36     Serial.print(longSekitar2, 5);
37     Serial.print("\r\n");
38     Serial.print("Koordinat invalid");
39     Serial.print("\r\n");
40 }
41 }
42 Else
43 {
44     //Serial.print(latitude);
45     Serial.print(corlat, 5);
46     Serial.print("\r\n");
47     //Serial.print(longitude);
48     Serial.print(corlon, 5);
49     Serial.print("\r\n");
```

```

50     Serial.print("Koordinat invalid");
51     Serial.print("\r\n");
52 }
53 }

```

Gambar 7.5. Fungsi untuk mengirim koordinat ke web server

```

1 void getKoorSekitar()
2 {
3     latSekitar= corlat + 0.0002;
4     latSekitar2= corlat - 0.0002;
5     longSekitar= corlon + 0.0002;
6     longSekitar2= corlon - 0.0002;
7 }

```

Gambar 7.6. Fungsi untuk batasan perpindahan motor

```

1 void start_GPS()
2 {
3     //Configuracion en Inicializacion GPS
4     //Serial.print("AT");
5     delay(1000);
6     Serial.println("AT+CGPSIPR=9600");//
7     (set the baud rate)
8     delay(5000);
9     Serial.println("AT+CGPSPWR=1"); // (
10    turn on GPS power supply)
11    delay(5000);
12    Serial.println("AT+CGPSRST=1"); // (
13    reset GPS in autonomy mode)
14    delay(10000); //delay para esperar
15    señal del GPS}

```

Gambar 7.7. Konfigurasi GPS pada arduino untuk mendapatkan koordinat

```

1  void read_GPS(){
2
3
4      Serial.println("AT+CGPSINF=0");
5
6      read_String();
7
8      strtok(inData, ",");
9      strcpy(longitude, strtok(NULL, ",")); //
10     Gets longitude
11     strcpy(latitude, strtok(NULL, ",")); //
12     Gets latitude
13
14     result();
15 }

```

Gambar 7.8. Fungsi untuk mendapatkan raw data posisi longitude dan latitude

```

1  void result(){
2
3      double newlat =
4      Datatransfer(latitude, 5);
5      double newlon =
6      Datatransfer(longitude, 5);
7      corlat = decimalgps(newlat);
8      corlon = decimalgps(newlon);
9  }
10
11 double decimalgps(double rawdata){
12     int degrees = (int)(rawdata / 100);
13     double minutes = rawdata -
14     (degrees*100);

```

```
12 double mindecimal = minutes / 60.0;
13 double total = degrees + mindecimal;
14
15 return total;
16 }
17
18 double Datatransfer(char *data_buf, char
19 num){ //konversi data ke tipe float
20
21 /*data_buf: data array input
22 double temp=0.0;
23 //num: jumlah angka di kanan titik decimal
24 unsigned char I,j;
25 if(data_buf[0]=='-'){
26 //konversi ke bilangan negatif
27 i=1;
28
29 while(data_buf[i]!='.')
30 //proses array input
31 temp=temp*10+(data_buf[i++]-0x30);
32 for(j=0;j<num;j++)
33 temp=temp*10+(data_buf[++i]-0x30);
34 for(j=0;j<num;j++)
35 temp=temp/10;
36
37 temp=0-temp;
38 }
39 else{
40 //konversi ke bilangan positif
41 i=0;
42 while(data_buf[i]!='.')
43 temp=temp*10+(data_buf[i++]-0x30);
```

```
38     for(j=0;j<num;j++)
39         temp=temp*10+(data_buf[++i]-0x30);
40     for(j=0;j<num;j++)
41         temp=temp/10 ;
42     }
43     return temp;
44 }
45
46 void ShowSerialData()
47 {
48     while(Serial.available()!=0)
49         Serial.write(Serial.read());
50 }
```

Gambar 7.9. Fungsi untuk memproses raw data longitude dan latitude menjadi longitude dan latitude yang dapat di tampilkan dalam map

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Guruh Adi Darma, lahir di Surabaya pada 6 Agustus 1992. Penulis adalah anak terakhir dari dua bersaudara. Penulis menempuh pendidikan formal di SD Al-Falah Tropodo pada saat kelas I hingga III lalu kemudian melanjutkan pendidikan hingga kelas VI SD di SD Al-Falah Taman Mayangkara, Surabaya. Penulis melanjutkan pendidikan di SMP Al-Falah Deltasari Waru yang kemudian disusul menuju SMAN 5 Surabaya. Secara singkat, penulis pernah mengenyam pendidikan di SD Al-Falah(1998-2004), SMP Al-Falah(2004-2007), dan SMAN 5 Surabaya(2007-2010). Pada tahun 2010, penulis diterima dan melanjutkan pendidikan S1 di Jurusan Teknik Informatika Fakultas Teknologi Informasi pada Institut Teknologi Sepuluh Nopember Surabaya angkatan 2010 yang terdaftar dengan NRP 5110100230. Di jurusan Teknik Informatika, penulis mengambil bidang minat *Net Centric Computing* atau lebih dikenal dengan nama NCC. Selama menempuh kuliah, penulis juga aktif dalam organisasi kemahasiswaan seperti Himpunan Mahasiswa Teknik Computer (HMTC) dan berkesempatan untuk menjadi staff dan staff ahli departemen minat bakat. Penulis dapat dihubungi melalui alamat email guruhadidarma@icloud.com.