



TUGAS AKHIR - KI141502

RANCANG BANGUN *MOBILE EMBEDDED SYSTEM* UNTUK IDENTIFIKASI DAN PEMETAAN KONDISI PERMUKAAN JALAN PADA AREA PERKOTAAN DENGAN ANALISIS DATA *DUAL-ACCELEROMETER SENSOR*

AHMAD FAUZAN MUFID
NRP 5112100203

Dosen Pembimbing
Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
Hudan Studiawan, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - K1141502

RANCANG BANGUN *MOBILE EMBEDDED SYSTEM* UNTUK IDENTIFIKASI DAN PEMETAAN KONDISI PERMUKAAN JALAN PADA AREA PERKOTAAN DENGAN ANALISIS DATA *DUAL-ACCELEROMETER* SENSOR

AHMAD FAUZAN MUFID
NRP 5112100203

Dosen Pembimbing
Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
Hudan Studiawan, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESES- KI141502
MOBILE EMBEDDED SYSTEM DESIGN FOR
IDENTIFICATION AND MAPPING THE ROAD
SURFACE CONDITIONS IN URBAN AREAS
WITH DATA ANALYSIS-ACCELEROMETER
DUAL SENSOR

AHMAD FAUZAN MUFID
NRP 5112100203

Advisor

Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
Hudan Studiawan, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2016

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN *MOBILE EMBEDDED SYSTEM* UNTUK IDENTIFIKASI DAN PEMETAAN KONDISI PERMUKAAN JALAN PADA AREA PERKOTAAN DENGAN ANALISIS DATA *DUAL-ACCELEROMETER* SENSOR

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Rumpun Mata Kuliah Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

AHMAD FAUZAN MUFID

NRP. 5112 100 203

Disetujui oleh Dosen Pembimbing Tugas Akhir
Waskitho Wibisono, S.Kom., M.Eng.
NIP: 197410222000031001



(Pembimbing 1)

Hudan Studiawan, S.Kom., M.Kom
NIP: 198705112012121003

(Pembimbing 2)

**SURABAYA
JUNI, 2016**

(Halaman ini sengaja dikosongkan)

**RANCANG BANGUN *MOBILE EMBEDDED SYSTEM*
UNTUK IDENTIFIKASI DAN PEMETAAN KONDISI
PERMUKAAN JALAN PADA AREA PERKOTAAN
DENGAN ANALISIS DATA *DUAL-ACCELEROMETER*
SENSOR**

Nama Mahasiswa : Ahmad Fauzan Mufid
NRP : 5112100203
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Waskitho Wibisono, S.Kom., M.Eng.,
Ph.D
Dosen Pembimbing II : Hudan Studiawan, S.Kom.,M.Kom.

ABSTRAK

Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas umum. Di jalan terdapat terdapat lubang, banyak kecelakaan terjadi karena kurangnya informasi mengenai kondisi jalan untuk pengguna jalan. Kurangnya informasi mengenai kondisi jalan membuat pengguna jalan kurang waspada sehingga kecelakaan lebih mungkin terjadi. Saat ini jalan yang berlubang diketahui jika ada laporan dari masyarakat, bahkan perbaikan jalan berlubang pun dilakukan sendiri oleh masyarakat. Dengan adanya layanan untuk menentukan lokasi jalan berlubang ini, diharapkan dapat membantu pemerintah dalam menentukan pengambilan kebijakan perbaikan jalan berlubang.

Pada penelitian sebelumnya sistem yang dibuat hanya menentukan terdapat lonjakan. Penelitian tersebut menggunakan sensor accelerometer dan GPS pada smartphone untuk pengambilan data. Pada penelitian ini, penulis ingin membuat sistem yang dapat menentukan lubang pada jalan. Untuk mendapatkan data aktivitas saat di jalan digunakan sensor accelerometer, lalu untuk mendapatkan lokasi jalan digunakan GPS/GPRS/GSM Shield V3.0. Untuk membantu mengontrol

diperlukan bantuan teknologi mikrokontroler yaitu Arduino uno. Untuk menentukan apakah jalan tersebut terdapat lubang digunakan batas yang didapatkan dari nilai selisih perubahan accelerometer. Hasil yang telah didapatkan dikirimkan ke server menggunakan GPRS. Hasil penentuan ditampilkan dalam bentuk peta pada web.

Kemudian sistem akan diuji coba mulai dari uji fungsionalitas hingga uji performa dengan menggunakan beberapa skenario yang telah ditentukan. Berdasarkan hasil uji coba performa, keakuratan keseluruhan sistem dalam menentukan keputusan yang benar yaitu 83,33% dengan delay waktu pengiriman dari rangkaian alat ke server lebih dari 11 detik.

Kata kunci: Jalan, Lubang, Mikrokontroler, Accelerometer, ADXL345, Sensor.

MOBILE EMBEDDED SYSTEM FOR IDENTIFICATION AND MAPPING THE ROAD SURFACE CONDITIONS IN URBAN AREAS WITH DATA ANALYSIS DUAL-ACCELEROMETER SENSOR

Student Name : Ahmad Fauzan Mufid
NRP : 5112 100 203
Major : Teknik Informatika FTIf-ITS
Advisor I : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
Advisor II : Hudan Studiawan, S.Kom.,M.Kom.

ABSTRACT

The road is a land transport infrastructure covering all parts of the road, including complementary buildings and equipment intended for general traffic. On the road there is a hole, many accidents occur due to lack of information about the road conditions to road users. Lack of information on road conditions make the road less vigilant so that more accidents may occur. Currently the potholes known if there are reports from the public, even repair potholes were carried out by the communities themselves. With the service to determine the location of potholes, is expected to assist the government in determining the policy making repairs potholes

In previous studies the systems made only determines there is a spike. These studies use accelerometer sensor and GPS on a smartphone for data retrieval. In this study, the authors wanted to create a system that can determine the hole on the road. To get the current activity data used road accelerometer sensor, and then to find out the location, the author use of GPS / GPRS / GSM Shield V3.0. To help control the necessary assistance that is Arduino Uno microcontroller technology. To determine whether the path is a hole used boundary obtained from the differences of the accelerometer changes. The results that have been obtained is

sent to the server using GPRS. The result of the determination shown in the form of maps on the web.

Then the system will be tested ranging from test to test the functionality of the performance by using a couple of scenarios that have been specified. Based on the results of performance testing, the overall accuracy of the system in determining the correct decision that is 83,33% with a delay time of delivery of a suite of tools to the server more than 11 seconds.

Keywords: Road, Potholes, Speed Bump, Microcontroller, Accelerometer, ADXL345, Sensor.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur, kehadiran Allah Subhanahu wa ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun *Mobile Embedded System* untuk Identifikasi dan Pemetaan Kondisi Permukaan Jalan pada Area Perkotaan dengan Analisis Data *Dual-Accelerometer Sensor*”.

Pengerjaan Tugas Akhir ini adalah momen bagi penulis untuk mengeluarkan seluruh kemampuan, hasrat, dan keinginan yang terpendam di dalam hati mulai dari masuk kuliah hingga lulus sekarang ini, lebih tepatnya di jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak. Melalui lembar ini, penulis ingin secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah melimpahkan rahmat, hidayah, dan inayah-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan baik.
2. Junjungan kita Nabi Muhammad SAW yang telah menjadi inspirasi, contoh yang baik bagi penulis sehingga tetap termotivasi dalam mengerjakan Tugas Akhir.
3. Kedua orang tua penulis yang telah mencurahkan kasih sayang, perhatian, dan doa kepada penulis.
4. Adik penulis yang memberi semangat dengan cara yang unik.
5. Waskitho Wibisono, S.Kom., M.Eng., Ph.D. dan Pak Hudan Studiawan, S.Kom.,M.Kom. selaku dosen pembimbing yang senantiasa memberikan arahan.
6. Pak Victor Hariadi, S.Si., M.Kom., selaku dosen wali penulis. Terima kasih atas dukungan, nasihat, dan pembelajaran hidup yang diberikan kepada penulis selama penulis menjalani kuliah di jurusan Teknik Informatika ITS

7. Google yang menjadi jembatan antara saya dan pengetahuan yang amat luas.
8. Regin Iqbal, Zola Mahendra, Yarjuna Rohmat, Yusuf Nugoroho, M. Satrio Ramadhana Yudhono, Ripas Filqadar dan Putu Adhi Purwanto.
9. Teman-teman lab NCC (Net Cet Centric) yang saling mendukung dalam memperjuangkan Tugas Akhir.
10. Teman-teman administrator lab NCC (Net Cet Centric) yang memberikan semangat dalam pengerjaan Tugas Akhir.
11. Seluruh teman Teknik Informatika ITS angkatan 2012 yang telah menemani dan memberi pengalaman berharga bagi penulis sejak maba sampai lulus.
12. Teman-teman angkatan 2009, 2010, 2011, 2012, 2013, 2014, 2015 yang sudah memberikan pengalaman selama kuliah di TC ini.
13. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tugas Akhir ini, penulis mengucapkan terima kasih.

Penulis telah berusaha sebaik mungkin dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, 5 Juni 2016
Penulis

Ahmad Fauzan Mufid

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
1 BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	6
2 BAB II TINJAUAN PUSTAKA.....	9
2.1 Mikrokontroler Arduino	9
2.2 Arduino Uno.....	9
2.3 GPS/GPRS/GSM <i>Shield</i> V3.0	11
2.4 Sensor <i>Accelerometer</i> ADXL345.....	12
2.5 Arduino IDE	14
2.6 <i>Activity Recognition</i>	16
2.6.1 Data Collection	16
2.6.2 Feature Extraction.....	16
2.6.3 Data Interpretation	16
2.7 Data Training.....	16
3 BAB III ANALISIS DAN PERANCANGAN	19
3.1 Deskripsi Umum Sistem.....	19
3.2 Arsitektur Umum Sistem.....	19
3.3 Perancangan Antarmuka Sistem.....	22
3.4 Perancangan Kebutuhan Arus Listrik.....	22
3.5 Perancangan Perangkat Keras	23

3.6	Diagram Alir Aplikasi Sistem	24
3.6.1	Diagram Alir Data Keseluruhan Sistem.....	25
3.6.2	Diagram Alir Menentukan Lubang	26
3.6.3	Diagram Alir Pengiriman Data	27
3.6.4	Diagram Alir Menampilkan Data pada <i>Website</i>	28
3.7	Perancangan Basis Data Sistem.....	29
4	BAB IV IMPLEMENTASI.....	33
4.1	Lingkungan Implementasi	33
4.2	Implementasi Perangkat Keras	34
4.3	Implementasi Perangkat Lunak pada Perangkat Keras...36	
4.4	Implementasi Perangkat Lunak untuk Membaca Sensor <i>Accelerometer</i> ADXL345.....	37
4.5	Implementasi Perangkat Lunak untuk Mendapatkan Nilai GPS.....	39
4.6	Implementasi Perangkat Lunak untuk Mengirim Data...40	
4.7	Implementasi Mendapatkan dan Menggunakan <i>Data Training</i>	42
4.8	Implementasi Menentukan <i>Threshold</i>	47
4.9	Implementasi Menentukan Lubang	48
4.10	Implementasi Rangkaian pada Sepeda Motor	50
4.11	Implementasi Penerimaan Data pada <i>Server</i>	51
4.12	Implementasi Menampilkan Peta pada <i>Website</i>	51
5	BAB V PENGUJIAN DAN EVALUASI	53
5.1	Uji Coba Fungsionalitas	53
5.1.1	Lingkungan Uji Coba.....	53
5.1.2	Uji Coba Data dari Perangkat Bergerak ke <i>Server</i> ...54	
5.2	Uji Coba Performa.....	56
5.2.1	Uji Coba Akurasi Melewati Lubang	56
5.2.2	Uji Coba Akurasi Melewati Jalan Normal	57
5.2.3	Uji Coba Melewati Polisi Tidur	59
5.3	Hasil Uji Coba	60
5.4	Uji Coba Delay Waktu Pengiriman	62
5.5	Evaluasi Pengujian.....	63
5.5.1	Evaluasi Pengujian Fungsionalitas.....	63
5.5.2	Evaluasi Pengujian Performa	63

6	BAB VI KESIMPULAN DAN SARAN	65
6.1	Kesimpulan.....	65
6.2	Saran.....	65
	DAFTAR PUSTAKA.....	67
	LAMPIRAN	69
	BIODATA PENULIS.....	77

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Arduino Uno.....	9
Gambar 2.2 GPS/GSM/GPRS Shield Arduino V3.0.....	11
Gambar 2.3 Sensor <i>accelerometer</i> ADXL345	13
Gambar 2.4 Arduino IDE	15
Gambar 3.1 Arsitektur Sistem	20
Gambar 3.2 Antarmuka lokasi lubang.....	22
Gambar 3.3 Skema Perancangan Perangkat Keras.....	24
Gambar 3.4 Diagram Alir Keseluruhan Sistem.....	25
Gambar 3.5 Diagram Alir Menentukan Lubang.....	26
Gambar 3.6 Diagram Alir Pengiriman Data.....	27
Gambar 3.7 Diagram Alir Menampilkan Data pada <i>Website</i>	28
Gambar 4.1 Perangkat Keras Sistem identifikasi dan pemetaan kondisi jalan	35
Gambar 4.2 Rangkaian Perangkat Keras Sistem Sistem identifikasi dan pemetaan kondisi jalan	35
Gambar 4.3 Implementasi <i>Setup</i> Sensor <i>Accelerometer</i> ADXL345	37
Gambar 4.4 Implementasi Mendapatkan Nilai Sensor <i>Accelerometer</i> ADXL345.....	38
Gambar 4.5 Implementasi Mendapatkan Nilai pada GPS.....	40
Gambar 4.6 Implementasi Perangkat Lunak Mengirim Data.....	42
Gambar 4.7 Menyimpan Data Kondisi Jalan pada <i>SD Card</i>	44
Gambar 4.8 Modul <i>Micro SD Card</i> pada Rangkaian Alat.....	44
Gambar 4.9 Grafik Perubahan Nilai <i>Accelerometer</i> Ketika Melewati Lubang.....	46
Gambar 4.10 Grafik Nilai Selisih Setiap Perubahan Nilai <i>Accelerometer</i> Melewati Lubang	46
Gambar 4.11 Kumpulan Nilai Absolute Selisih Kondisi Jalan Berlubang	48
Gambar 4.12 Implementasi Menentukan Jalan Berlubang.....	49
Gambar 4.13 Implementasi Rangkaian Depan pad Sepeda Motor	50

Gambar 4.14 Implementasi Rangkaian Belakang pada Sepeda Motor	50
Gambar 4.15 Implementasi Penerimaan Data pada Server	51
Gambar 4.16 Tampilan Antarmuka Peta Kondisi Jalan pada <i>Website</i>	52
Gambar 5.1 <i>Database</i> Menerima Data dari Perangkat bergerak	55
Gambar 5.2 Peta Menampilkan Kondisi Jalan dengan Penanda	56
Gambar 0.1 Inisialisasi Variabel Global.....	69
Gambar 0.2 Fungsi-fungsi untuk Mengonversi Nilai <i>Raw</i> GPS Menjadi Float.....	70
Gambar 0.3 Fungsi untuk Menginisialisasi GPS dan GSM	71
Gambar 0.4 Fungsi Setup untuk Mempersiapkan Sensor dan GPS/GPRS/GSM <i>Shield</i> V3.0	72
Gambar 0.5 Fungsi untuk Penentuan Keputusan.....	74
Gambar 0.6 Fungsi untuk Menerima Data	75

DAFTAR TABEL

Tabel 2.1 Arduino Data Sheet	10
Tabel 2.2 AT Command pada GPS/GPRS/GSM Shield V3.0	12
Tabel 2.3 Deskripsi Pin pada ADXL345	14
Tabel 3.1 Perancangan Kebutuhan Daya	23
Tabel 3.2 Perancangan Tabel <i>Final</i>	29
Tabel 3.3 Perancangan Tabel <i>Front</i>	30
Tabel 3.4 Perancangan Tabel <i>Back</i>	31
Tabel 4.1 Data Training Melewati Lubang	45
Tabel 5.1 Lingkungan Uji Coba	54
Tabel 5.2 Data dari Perangkat Bergerak ke <i>Server</i>	55
Tabel 5.3 Skenario Uji Coba Akurasi Melewati Lubang	56
Tabel 5.4 Uji Coba Melewati Lubang Rangkaian Bagian Depan	57
Tabel 5.5 Uji Coba Melewati Lubang Rangkaian Bagian Belakang	57
Tabel 5.6 Uji Coba Melewati Lubang Keputusan <i>Final</i>	57
Tabel 5.7 Skenario Uji Coba Akurasi Melewati Jalan Normal ...	58
Tabel 5.8 Uji Coba Melewati Jalan Normal Rangkaian Bagian Depan	58
Tabel 5.9 Uji Coba Melewati Jalan Normal Rangkaian Bagian Belakang	58
Tabel 5.10 Uji Coba Melewati Jalan Normal Keputusan <i>Final</i> ..	59
Tabel 5.11 Skenario Uji Coba Akurasi Melewati Polisi Tidur ...	59
Tabel 5.12 Uji Coba Melewati Polisi Tidur Rangkaian Bagian Depan	60
Tabel 5.13 Uji Coba Melewati Polisi Tidur Rangkaian Bagian Belakang	60
Tabel 5.14 Uji Coba Melewati Polisi Tidur Keputusan Final	60
Tabel 5.15 Representasi Keputusan dengan Kondisi Sebenarnya	61
Tabel 5.16 Hasil Uji Coba Sistem	61
Tabel 5.17 Data dari Rangkaian Bagian Depan	62
Tabel 5.18 Data dari Rangkaian Bagian Belakang	62
Tabel 5.19 Rangkuman Hasil Pengujian Fungsional	63

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

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

1.1 Latar Belakang

Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas umum. Di jalan terdapat lubang, banyak kecelakaan terjadi karena kurangnya informasi mengenai kondisi jalan untuk pengguna jalan. Kurangnya informasi mengenai kondisi jalan membuat pengguna jalan kurang waspada sehingga kecelakaan lebih mungkin terjadi. Saat ini jalan yang berlubang diketahui jika ada laporan dari masyarakat, bahkan perbaikan jalan berlubang pun dilakukan sendiri oleh masyarakat. Dengan adanya layanan untuk menentukan lokasi jalan berlubang ini, diharapkan dapat membantu pemerintah dalam menentukan pengambilan kebijakan perbaikan jalan berlubang.

Pada penelitian sebelumnya sistem yang dibuat hanya menentukan jalan mulus, kasar, atau terdapat lonjakan. Penelitian tersebut menggunakan sensor *accelerometer* dan GPS pada smartphone untuk pengambilan data. Pada penelitian ini, penulis ingin membuat sistem yang dapat menentukan terdapat lubang pada jalan. Untuk mendapatkan data jalan digunakan sensor *accelerometer*, lalu untuk mendapatkan lokasi jalan digunakan GPS *shield*. Untuk membantu mengontrol diperlukan bantuan teknologi mikrokontroler yaitu Arduino uno. Untuk menentukan apakah jalan tersebut terdapat lubang digunakan batas yang didapatkan dari *threshold* yang didapatkan dari selisih perubahan nilai *accelerometer*. Hasil yang telah didapatkan dikirimkan ke *server* menggunakan GSM *Shield*. Hasil penentuan ditampilkan dalam bentuk peta pada *web*.

Pada penelitian ini, penulis ingin membuat sistem yang dapat mendapatkan data jalan dengan menggunakan sensor *accelerometer* yang diletakkan pada sepeda motor. Kemudian sistem dapat menganalisa data untuk menentukan lubang. Hasil dari data yang didapat beserta lokasi dikirimkan ke *web*. Data yang diterima di ditampilkan dalam peta. Untuk mengatur penerimaan dan pengiriman data diperlukan mikrokontroler. Sensor yang dibutuhkan berupa *accelerometer* ADXL345 untuk mengukur percepatan statis gravitasi dalam merasakan kemiringan, modul GPS/GPRS/GSM V3.0 untuk mendapatkan lokasi lubang dengan modul GPS dan mengirim data ke web dengan modul GSM/GPRS, sehingga dapat ditampilkan di web secara *realtime*. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu peralatan yang digunakan, di sini penulis menggunakan sensor *accelerometer*, modul GPS/GPRS/GSM V3.0, Arduino, di sini penulis juga menentukan di jalan terdapat lubang dan adanya hasil yang ditampilkan pada web berupa peta lokasi jalan berlubang.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merekam data kondisi jalan datar, jalan berlubang menggunakan sensor *accelerometer* ADXL345?
2. Bagaimana mengekstrak data dari hasil rekaman *accelerometer* ADXL345 dan GPS?
3. Bagaimana membedakan jalan berlubang dan normal dari data yang telah didapatkan?
4. Bagaimana menentukan lokasi jalan berlubang?
5. Bagaimana akurasi sistem dalam menentukan lokasi jalan berlubang?
6. Bagaimana menampilkan lokasi jalan berlubang dalam peta pada *web*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Lokasi pemetaan pada area perkotaan.
2. Data yang dikirimkan ke *server* merupakan data hasil *accelerometer* dan lokasi jalan dari *GPS shield*.
3. Hasil terdapat 2 kelas yaitu jalan berlubang dan normal.
4. Data dikirimkan ke *server* ketika motor berada di jalan.
5. Koneksi internet tidak putus.
6. Kecepatan sepeda motor stabil pada 20 km/jam sampai dengan 30 km/jam.
7. Tidak mengirimkan data kondisi jalan normal ke *server*.
8. Lubang memiliki kedalaman lebih dari 15 cm.
9. Lubang tidak berada pada jalan yang belum diaspal.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah.

1. Membuat sistem yang dapat mengidentifikasi jalan berlubang.
2. Sistem yang dibuat dapat menampilkan lokasi jalan yang berlubang.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai sarana untuk mengetahui kondisi jalan, sehingga memungkinkan pengguna layanan untuk lebih berhati-hati dalam memilih jalan yang dilewati dengan terlebih dahulu memeriksa lokasi jalan yang berlubang. Layanan ini juga dapat membantu pemerintah melakukan pemetaan kondisi jalan yang berlubang secara efisien. Sehingga mempermudah pemerintah dalam menentukan pengambilan kebijakan perbaikan jalan berlubang.

1.6 Metodologi

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

1. Penyusunan Proposal Penelitian

Pada tahap awal penyusunan proposal, penulis memulai pengerjaan Tugas Akhir dengan melakukan penyusunan proposal Tugas Akhir. Pada proposal tersebut berisi mengenai mekanisme sistem identifikasi dan pemetaan kondisi permukaan jalan.

2. Studi Literatur

Tahap ini adalah tahap pengumpulan informasi dan pembelajaran materi yang akan digunakan dalam pengerjaan Tugas Akhir. Karena pada pengerjaan Tugas akhir dibutuhkan berbagai komponen perangkat keras, maka penulis melakukan studi literatur mengenai penggunaan perangkat keras tersebut. Studi literatur yang termasuk diskusi dan pemahaman mengenai topik Tugas Akhir adalah sebagai berikut:

1. Mekanisme mikrokontroler Arduino.
2. Penerimaan dan pengiriman data dari *server*.
3. Penerimaan dan pengiriman data dari mikrokontroler Arduino ke *server*.
4. Pembacaan data pada sensor *accelerometer* ADXL345 dan modul DS 1307.
5. Proses mikrokontroler dapat melakukan membaca dan menerima gps melalui GPS/GPRS/GSM *Shield* V3.0.
6. Proses mikrokontroler dapat melakukan koneksi GPRS melalui GPS/GPRS/GSM *Shield* V3.0.
7. Pengiriman data dari GPS/GPRS/GSM *Shield* V3.0 ke *server*.
8. Menerima data pada *server*.
9. Menampilkan peta pada *website*

3. Perancangan Sistem

Pada tahap ini dilakukan analisis terhadap sistem serta perancangan sistem yang akan dibuat. Hal ini ditujukan untuk dapat merumuskan solusi yang data ditempuh untuk melakukan implementasi pada sistem yang akan dibuat serta kemungkinan yang dapat terjadi ketika implementasi sistem berlangsung.

4. Implementasi Perangkat Lunak

Untuk implementasi sistem identifikasi dan pemetaan kondisi permukaan jalan ini, implementasi perangkat lunak pada penelitian ini adalah sebagai berikut:

- Implementasi pada *server*
Pada implementasi perangkat lunak ada sisi *server* menggunakan bahasa pemrograman PHP dengan kerangka kerja Codeigniter 3.0.6, basis data menggunakan MySQL untuk menyimpan hasil dan lokasi dari mikrokontroler dan menampilkan peta menggunakan kerangka kerja leaflet.
- Sensor dan kontroler
Kontroler adalah pelaksana dari perintah mikrokontroler Arduino, sedangkan sensor adalah pengambil data dari apa yang terjadi di lingkungannya. Keduanya adalah perangkat *input* dan *output* bagi Arduino Uno. Arduino diprogram dengan kumpulan fungsi bahasa pemrograman C/C++. Untuk menuliskan logika-logika dari skema rangkaian yang terhubung dengan perangkat keras Arduino Uno, dibutuhkan Arduino IDE.

5. Pengujian dan Evaluasi

Sistem akan diuji setelah selesai diimplementasikan menggunakan skenario yang sudah dipersiapkan yang sesuai

dan cocok dengan sistem yang telah diimplementasikan. Pengujian dan evaluasi diperlukan untuk mengetahui kesalahan yang terjadi dan memperbaiki kesalahan tersebut.

6. Penyusunan Buku Penelitian

Pada tahap ini disusun laporan penelitian sebagai dokumentasi pelaksanaan penelitian, yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan

Buku penelitian ini bertujuan untuk mendapatkan gambaran dari pengerjaan penelitian dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku penelitian ini terdiri atas beberapa bagian yaitu :

BAB I PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan penelitian ini sehingga dapat berjalan dengan baik.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai desain dan perancangan perangkat lunak. Desain perangkat lunak meliputi desain data, arsitektur, dan proses.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat output yang dihasilkan oleh aplikasi, dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran-saran untuk pengembangan aplikasi pada masa mendatang.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan penelitian.

Lampiran

Merupakan bab tambahan yang berisi kode-kode sumber yang penting pada aplikasi ini.

(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi penjelasan dasar teori yang menjadi dasar pembuatan Tugas Akhir ini.

2.1 Mikrokontroler Arduino

Arduino adalah sebuah alat untuk membuat komputer yang dapat lebih merasakan dan mengontrol lingkungannya. Mikrokontroler ini merupakan komputer fisik berbasis *open source* yang menyediakan lingkungan pengembangan software yang dapat dihasilkan untuk *board* Arduino berupa Arduino IDE.

2.2 Arduino Uno

Arduino Uno yang ditunjukkan pada Gambar 2.1 adalah salah satu jenis mikrokontroler berbasis ATmega 328. Mikrokontroler ini menggunakan bahasa pemrograman Processing yang mengkombinasikan basa pemrograman C++ dan Java.



Gambar 2.1 Arduino Uno

Tabel 2.1 Arduino Data Sheet

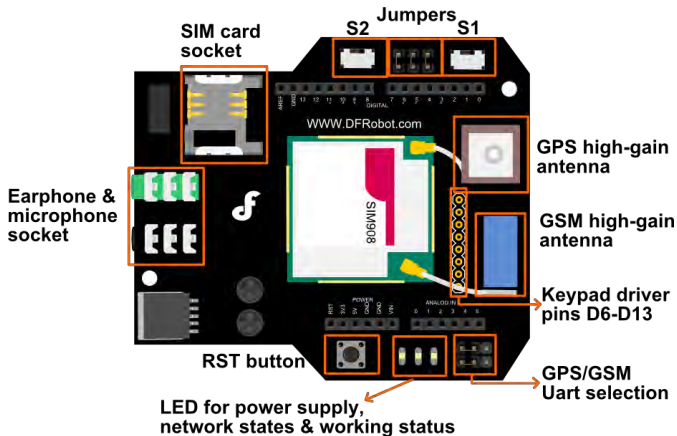
Mikrokontroler	ATmega328
<i>Operating Voltage</i>	5 V
<i>Input Voltage (recommended)</i>	7-12 V
<i>Input Voltage (limits)</i>	6-20 V
<i>Digital I/O Pins</i>	14 (6 pin merupakan <i>Pulse Width Modulation output</i>)
<i>Analag Input Pins</i>	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
<i>Flash Memory</i>	32 KB (ATmega328) dimana 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz

Mikrokontroler ATmega328 memiliki 14 pin digital *input/output* (6 pin digunakan sebagai PWM output, 6 analog *input*, 16 MHz sumber *clock signal* untuk digital *circuits*, sebuah konektor USB, sebuah *socket* listrik, sebuah konektor ICSP, dan tombol *reset*. Arduino Uno dapat diberikan *power* melalui konektor USB atau dengan *power supply* eksternal [2]. Pada penelitian ini, Arduino Uno digunakan sebagai mikrokontroler untuk mengatur sensor *accelerometer* ADXL345 agar berjalan dengan baik.

Arduino Uno yang memiliki detail spesifikasi seperti pada Tabel 2.1, digunakan sebagai mikrokontroler untuk mengatur sistem identifikasi permukaan jalan dalam membaca, menghasilkan keputusan dan mengirimkan data agar berjalan dengan baik.

2.3 GPS/GPRS/GSM Shield V3.0

Arduino GPS/GPRS/GSM *Shield* seperti yang ditunjukkan pada Gambar 2.2 GPS/GPR/GSM *shield* dari DFRobot. *Shield* ini dapat bekerja dengan mesin Quad-band GSM/GPRS pada frekuensi 900MHz EGSM/DCS 1800MHz dan GSM850 MHz/PCS 1900MHz. *Shield* ini juga mendukung teknologi GPS untuk navigasi satelit. Itu mungkin bagi robot dan sistem kontrol untuk mengirim pesan dan menggunakan jaringan GSM. Hal ini dikontrol melalui perintah AT (GSM07.07, 07.05 dan SIMCOM enhanced AT Commands). Dan desain *shield* ini memungkinkan untuk menggunakan fungsi GSM & GPS langsung dengan komputer dan Arduino. Ini termasuk *high-gain* antena SMD untuk GPS & GSM.



Gambar 2.2 GPS/GSM/GPRS Shield Arduino V3.0 [3]

Shield GPS/GPRS/GSM ini menggunakan chip SIM908 yang diterapkan dari SIMCom. Menampilkan antarmuka fungsi standar industri dan GPS, kombinasi dari kedua teknologi memungkinkan barang, kendaraan dan orang yang akan dilacak dengan mulus di setiap lokasi dan kapan saja dengan cakupan sinyal [4].

Dalam penggunaan *shield* ini dibutuhkan *command* khusus yang disebut AT *command*. AT *command* akan diprogram dan *shield*

akan membaca perintah sesuai dengan *command* dalam program. AT *command* sendiri memiliki format penulisan yang sudah diatur dari pabrik pembuat *shield*. Beberapa contoh AT *command* yang digunakan pada sistem ini ini dapat dilihat pada Tabel 2.2.

Tabel 2.2 AT Command pada GPS/GPRS/GSM Shield V3.0 [5]

No.	AT Command	Keterangan
1	AT+CGSPWR	Menyalakan GPS
2	AT+CGPSRST	Mengatur Mode GPS
3	AT+CGPSINF	Mendapatkan informasi lokasi
4	AT+CGREG=?	Mengecek staus jaringan
5	AT+HTTPIPINIT	Inisiasi mode HTTP
6	AT+HTTPTERM	Menyudahi mode HTTP
7	AT+HTTPPARAM	Mengatur URL yang akan dikunjungi

Pada penelitian ini, GPS/GPRS/GSM *Shield* digunakan sebagai pendukung Arduino untuk pengiriman data ke *server* melalui internet dan menentukan posisi kendaraan.

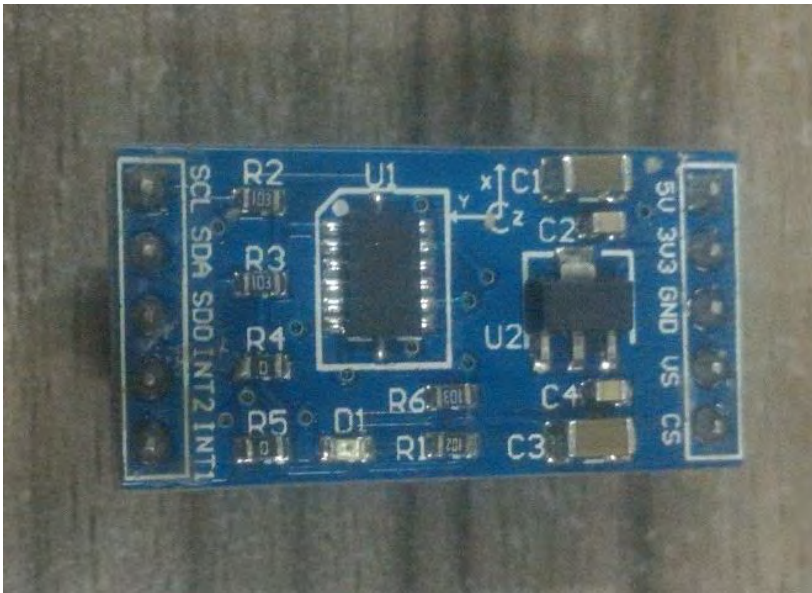
2.4 Sensor Accelerometer ADXL345

Sensor *accelerometer* ADXL345 seperti yang ditunjukkan pada Gambar 2.3. ADXL345 merupakan salah satu tipe *accelerometer* yang menggunakan prinsip Micro Electro Mechanical System (MEMS). MEMS sendiri menurut artinya adalah sebuah sistem mikro dengan ukuran komponen penyusun mikron. MEMS jenis ini umumnya terdiri dari dua komponen dasar, yaitu elemen *sensing* dan unit transduser.

Accelerometer ADXL345 adalah produk dengan jenis Analog *Device*. Sensor jenis ini mengukur percepatan linier dalam tiga sumbu(x, y dan z) dengan resolusi tinggi yaitu bisa mencapai 13-bit. Pada sensitivitasnya memiliki pilihan *range* pengukuran mulai dari $\pm 2g$, $\pm 4g$, $\pm 8g$ hingga $\pm 16g$. Di mana tiap 1g merupakan satu satuan percepatan rata-rata gravitasi bumi yaitu $9,8m/s^2$. Dalam penelitian ini menggunakan tingkat sensitivitas sebesar $\pm 4g$. Pada sensor ini terdapat beberapa fungsi penginderaan khusus yang disediakan. Kegiatan dan aktivitas merasakan mendeteksi ada atau kurangnya

gerak dan jika percepatan pada sumbu manapun melebihi tingkat yang telah diatur. Penginderaan jatuh bebas mendeteksi jika perangkat jatuh. Fungsi-fungsi ini dapat dipetakan ke salah satu dari dua interrupt pin output. Paten tertunda 32-tingkat *first in, first out* (FIFO) buffer yang terintegrasi dapat digunakan untuk menyimpan data untuk meminimalkan intervensi host processor [6].

Serial I/O pada jenis *accelerometer* ini yang dihubungkan dengan mikrokontroler, dalam penelitian ini menggunakan mikrokontroler Arduino uno. ADXL345 mempunyai dua metode antarmuka yaitu SPI dan I2C di mana keduanya merupakan antarmuka digital.



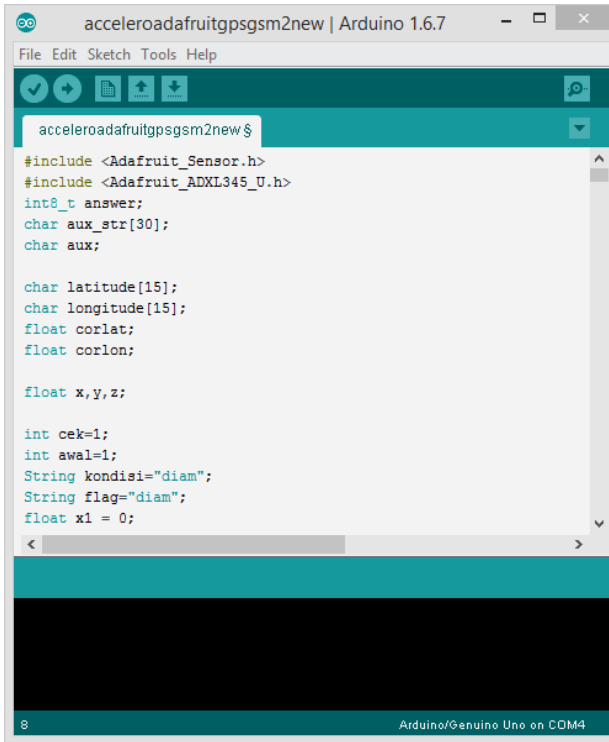
Gambar 2.3 Sensor *accelerometer* ADXL345

Tabel 2.3 Deskripsi Pin pada ADXL345 [7]

Nama Pin	Fungsi
VDD I/O	Power Supply untuk digital interface
GND	Ground
VS	Power supply
CS	Chip Select
SCL/SCLK	Serial Communications Clock. SCL untuk I2C dan SCL untuk SPI
SDA/SDI/SDIO	Serial Data (I2C)/ Serial Data Inut (SPI 4-wire)/ Serial Data Input Output (SPI 3-wire)
SDO/ALT ADDR	Serial Data Output (SPI 4-wire)/ Alternate I2C Address
INT1	Interrupt 1
INT2	Interrupt 2

2.5 Arduino IDE

Lingkungan pengembangan Arduino terdiri dari *text editor* untuk menulis kode Arduino, area pesan, *console text*, *toolbar* dengan tombol untuk fungsi secara umum digunakan pada kode Arduino, dan beberapa menu. Arduino IDE yang ditunjukkan pada Gambar 2.4 ini akan terkoneksi dengan perangkat keras Arduino ketika Arduino disambungkan dengan kabel USB untuk mengunggah program dan saling berkomunikasi. Program untuk Arduino ditulis pada sketch. Sketch yang disimpan dalam bentuk file .ino [8].



Gambar 2.4 Arduino IDE

Terdapat pula *console* untuk menampilkan pesan dan informasi *error* kompilasi program. Pada toolbar terdapat beberapa tombol berikut:

- *Verify* untuk melakukan cek *error* pada kode program.
- *Upload* untuk kompilasi kode program dan mengunggah kode program ke hardware Arduino.
- *New* untuk membuat sketch baru.
- *Open* untuk membuka kode program yang sudah tersimpan sebelumnya.
- *Save* untuk menyimpan kode program.
- *Open serial monitor* untuk membuka serial monitor Arduino.

2.6 *Activity Recognition*

Activity recognition merupakan teknik yang digunakan untuk proses pendeteksian aktivitas pengguna. Adapun tahap-tahap dari *activity recognition* dapat dijelaskan sebagai berikut [9]:

2.6.1 **Data Collection**

Untuk melakukan sebuah identifikasi aktivitas dikatakan melewati lubang, diperlukan data yang nantinya akan digunakan untuk pengolahan. Pada umumnya data diambil dengan menggunakan sebuah *device* berupa sensor. Data ini diharapkan dapat digunakan untuk dilakukan proses identifikasi yang bertujuan untuk menggambarkan sebuah entitas tertentu.

2.6.2 **Feature Extraction**

Setelah memperoleh data dari sensor *accelerometer*, proses selanjutnya adalah ekstraksi data. Data yang akan diproses terlebih dahulu diolah sedemikian hingga agar data tersebut tidak terlalu konvergen atau terlalu divergen untuk dilakukan proses penentuan lubang. Salah satunya menggunakan teknik *data sampling*.

2.6.3 **Data Interpretation**

Hal ini merupakan tahap paling penting dalam proses identifikasi aktivitas pengguna. Setelah memiliki data yang sudah diekstraksi maka langkah selanjutnya adalah melakukan pendeteksian.

2.7 **Data Training**

Data *training* merupakan sekumpulan data yang biasanya disajikan dalam bentuk tabel yang memuat informasi tertentu. Informasi tersebut digunakan untuk mencari hubungan tertentu dari suatu data *training*.

Untuk mendapatkan nilai *threshold*, maka dibandingkan antara data *training* sensor *accelerometer* saat kondisi pengguna normal dengan sensor saat kondisi melewati lubang. Dengan menggunakan perbandingan terhadap nilai *threshold*, dapat dibedakan antara aktivitas normal dengan aktivitas yang menyerupai melewati lubang.

Secara umum kerja menentukan lubang adalah sebagai berikut:

- 1) Pertama, sensor *accelerometer* menerima data berupa sumbu x , y , dan z .
- 2) Dari sumbu x , y , dan z yang di dapatkan, dicari nilai selisih untuk masing masing sumbu. Dari nilai selisih masing-masing sumbu ditentukan nilai absolut.
- 3) Beberapa nilai absolut dikumpulkan, dari nilai absolut masing-masing sumbu yang dikumpulkan kemudian dicari sumbu yang memiliki nilai yang paling aktif atau paling banyak mengalami perubahan. Kemudian tentukan nilai maksimum dan minimum untuk sumbu tersebut.
- 4) Nilai maksimum dan minimum dari sumbu yang telah ditentukan dijadikan sebagai nilai *threshold*.

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Bab ini akan menjelaskan mengenai dasar dari perancangan sistem yang akan dibangun pada Tugas Akhir. Perancangan yang dibahas adalah mengenai deskripsi umum sistem, proses perancangan, alur, dan implementasinya.

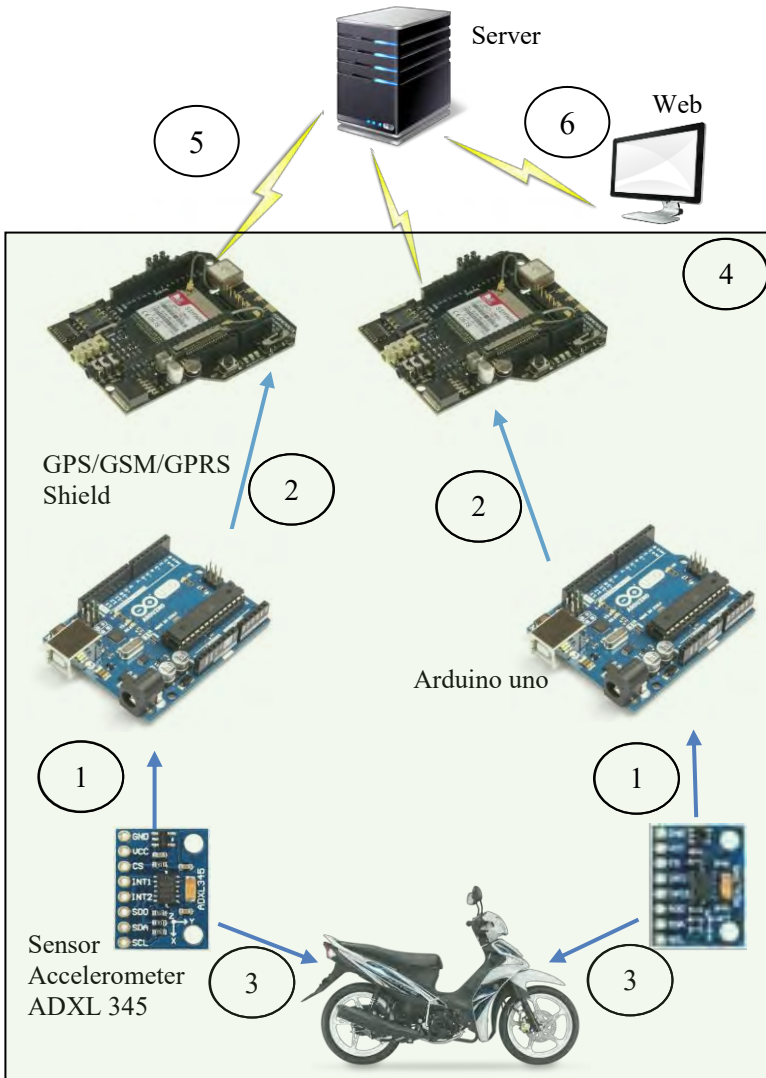
3.1 Deskripsi Umum Sistem

Pada penelitian ini akan dibangun sistem identifikasi dan pemetaan kondisi permukaan jalan secara *real-time* menggunakan mikrokontroler Arduino, GPS/GPRS/GSM *Shield* Arduino kemudian data dikirim ke *server* yang kemudian di *web* akan ditampilkan lokasi jalan berlubang berdasarkan data GPS yang dikirimkan ke *server*.

Sistem identifikasi dan pemetaan kondisi permukaan jalan ini berfungsi untuk memantau aktivitas pengguna ketika melewati lubang. Sensor *accelerometer* mampu mendeteksi percepatan linier sumbu x, sumbu y dan sumbu z. Kemudian data *real-time* dari sensor *accelerometer* akan dilakukan perhitungan untuk menentukan apakah perubahan gerakan yang terjadi merupakan perubahan yang mengindikasikan pengguna melewati lubang.

3.2 Arsitektur Umum Sistem

Rancangan arsitektur keseluruhan sistem pada sistem identifikasi kondisi jalan yang akan dibangun seperti pada Gambar 3.1. Sistem yang dibangun terdapat dua rangkaian, salah satu rangkaian diletakkan pada bagian depan sepeda motor dan rangkaian yang lainnya diletakkan pada bagian sepeda motor. Saat sepeda motor berjalan, masing-masing rangkaian akan memberikan hasil keputusan kondisi jalan.



Gambar 3.1 Arsitektur Sistem

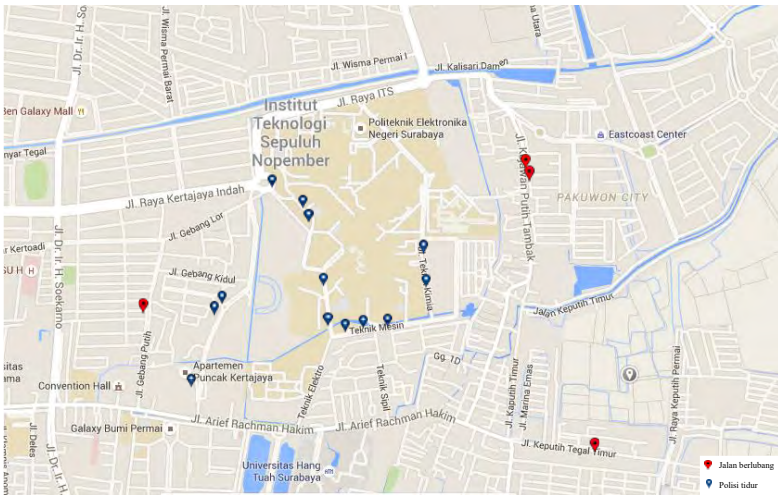
Berdasarkan pada Gambar 3.1, sistem identifikasi kondisi permukaan jalan ini memiliki alur proses yang dijabarkan sesuai nomor sebagai berikut:

1. Sensor *accelerometer* ADXL345 dihubungkan dengan mikrokontroller Arduino uno, tingkat sensitivitas pada ADXL345 diset $\pm 4g$
2. GPS/GPRS/GSM *Shield* V.3.0 dipasangkan pada Arduino, setelah itu dilakukan konfigurasi perangkat yaitu mengatur apn dari *provider* yang digunakan.
3. Sensor *accelerometer* ADXL345, GPS/GPRS/GSM *Shield* V.3.0 dan Arduino yang telah jadi satu di letakkan pada bagian depan dan belakang sepeda motor.
4. pada tahap ini perangkat bergerak sistem identifikasi dan pemetaan kondisi permukaan jalan siap untuk digunakan untuk mendapatkan data sensor dari aktivitas pengguna sebelum jatuh dan saat terjatuh. hasil dari *training* data akan dikelompokkan berdasarkan kondisi suatu kejadian dan mencari hubungan tertentu dari suatu data *training*. data akan diproses sedemikian rupa dengan membandingkan dengan nilai *threshold* yang telah didapatkan sehingga dapat menentukan bahwa pengguna sedang melewati lubang.
5. Perangkat bergerak akan mengirimkan data letak posisi kejadian tersebut berupa koordinat *longitude* dan *latitude* yang didapatkan dari GPS dengan perintah AT Command yaitu *POST* HTTP ke *server*, pengiriman data dilakukan dengan cara menambahkan paramater nilai GPS dan hasil perhitungan pada URL dengan menggunakan teknologi GPRS dari provider yang digunakan.
6. *Website* identifikasi dan pemetaan kondisi permukaan jalan akan melakukan *request* pula berupa HTTP GET pada *server* untuk mendapatkan data *latitude* dan

longitude untuk mendapatkan lokasi dimana pengguna melewati lubang.

3.3 Perancangan Antarmuka Sistem

Pada penelitian ini akan dirancang antarmuka untuk pengguna yakni antarmuka berupa peta yang menampilkan lokasi pengguna melewati lubang ketika dibuka.



Gambar 3.2 Antarmuka lokasi lubang

3.4 Perancangan Kebutuhan Arus Listrik

Perancangan kebutuhan arus listrik adalah salah satu proses penting yang harus dilakukan dalam membangun sistem. Hal ini terjadi karena masing-masing komponen dalam suatu sistem membutuhkan daya yang berbeda-beda, agar seluruh komponen bisa mendapatkan arus listrik minimum yang baik dan dapat secara optimum menjalankan sistem, maka perlu dipersiapkan kebutuhan arus listrik masing-masing komponen. Perancangan kebutuhan arus listrik dapat dilihat pada Tabel 3.1

Pada penggunaan GPRS untuk mengirimkan data dari sensor ke *server* kebutuhan arus listrik yang digunakan hanya pada saat pengiriman berlangsung, begitu juga dengan penggunaan GSM.

Tabel 3.1 Perancangan Kebutuhan Daya

Nama Perangkat	Kebutuhan Arus Listrik	
	Arus Listrik	Tegangan Listrik
Arduino Uno	40 mA	5V DC
GSM/GPRS/GPS Shield	100 mA	6-12V DC
Sensor <i>Accelerometer</i> ADXL345	15 mA	5~24V DC
Total	155 mA	-

3.5 Perancangan Perangkat Keras

Perancangan perangkat keras secara umum menjelaskan mengenai perancangan penempatan perangkat keras yang digunakan untuk membangun sistem. Rangkaian perangkat keras pada sistem ini seperti pada Gambar 3.3, terdapat komponen yang digunakan agar sistem dapat berjalan sebagai berikut:

1. Dua buah sensor *accelerometer* ADXL345
2. Dua buah Arduino Uno
3. Dua buah GPS/GPRS/GSM *Shield* V3.0
4. Dua buah baterai 9V.

GPRS/GSM/GPS *Shield* V3.0 ditempatkan diatas Arduino Uno, sehingga pemasangan 4 buah kabel *accelerometer* ADXL345 ditempatkan pada *shield* tersebut. Empat buah kabel tersebut yaitu Ground Arduino, VCC Arduino, pin A4(SDA) dan pin A5(SCL) sebagai input analog. Sedangkan pengiriman data pada Arduino melalui GPRS *Shield* yang berguna menghubungkan Arduino dengan *server*, untuk mengirimkan data berupa lokasi saat pengguna melewati lubang melalui mode

GSM, pada penelitian ini digunakan provider “Tri” untuk mengirimkan data dengan GPRS.



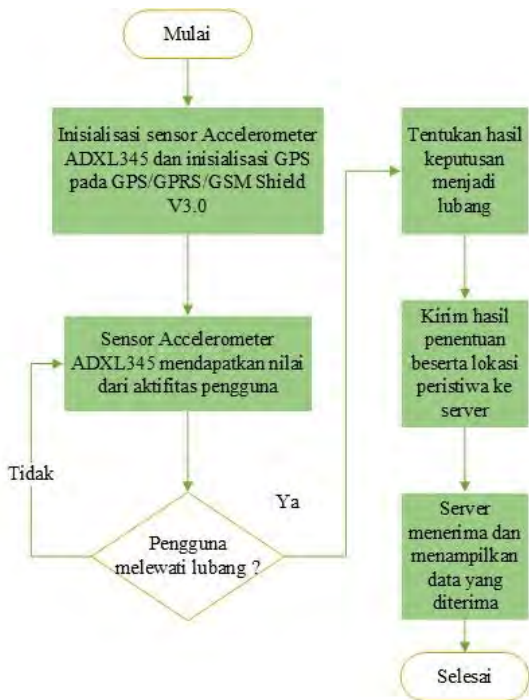
Gambar 3.3 Skema Perancangan Perangkat Keras

3.6 Diagram Alir Aplikasi Sistem

Diagram alir aplikasi sistem dibuat untuk memudahkan dalam merancang dan memahami seluruh proses yang terjadi di dalam sistem. Diagram alir aplikasi sistem terdiri dari diagram alir keseluruhan sistem, diagram alir sensor *accelerometer* ADXL345, diagram alir pengiriman data ke *server*, diagram alir penampilan data pada *website*, dan diagram alir *GPS Shield* untuk mencari lokasi jatuh.

3.6.1 Diagram Alir Data Keseluruhan Sistem

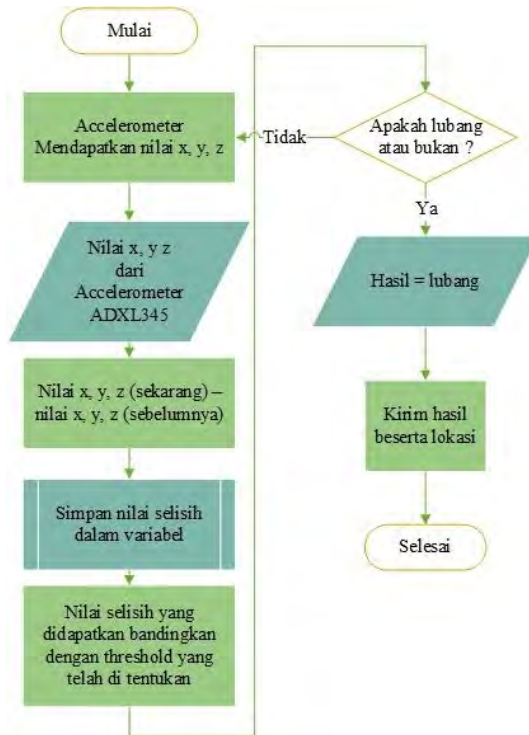
Perancangan alir data keseluruhan sistem dilakukan untuk dapat lebih memahami dan memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, serta menunjukkan tentang fungsi-fungsi utama atau proses yang ada. Pada Gambar 3.4, sistem diawali dengan menginisialisasi GPS dan sensor *accelerometer* ADXL345, kemudian sensor mulai merekam aktivitas pengguna, Jika terdapat perubahan aktivitas yang menunjukkan pengguna melewati lubang perangkat bergerak akan mencatat koordinat lokasi kejadian. Data lokasi kejadian berupa koordinat *latitude* dan *longitude* akan dikirimkan ke *server*, dan data tersebut juga akan ditampilkan di *web* dalam bentuk peta yang berisi tanda. Tanda dalam peta menunjukkan lokasi lubang.



Gambar 3.4 Diagram Alir Keseluruhan Sistem

3.6.2 Diagram Alir Menentukan Lubang

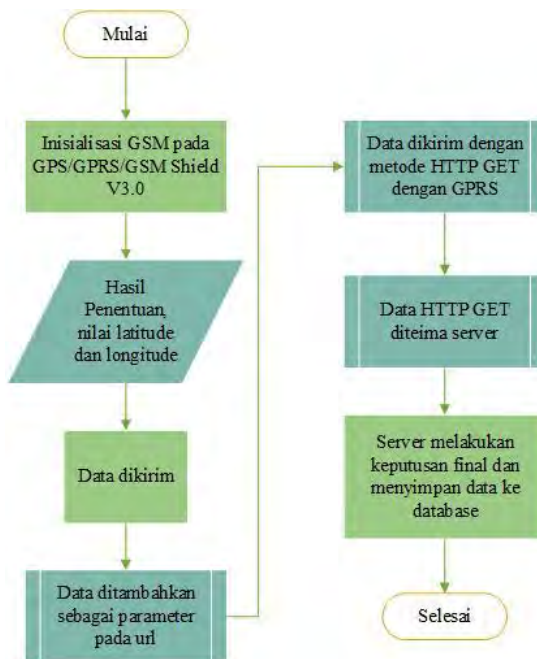
Pada Gambar 3.5 dalam proses mendeteksi seluruh aktivitas pengguna dan untuk mengetahui adanya peristiwa yang menunjukkan melewati lubang dilakukan oleh sensor *accelerometer* ADXL345. Sistem identifikasi dan pemetaan kondisi permukaan jalan ini menggunakan tingkat sensitivitas sebesar $\pm 4g$ pada *accelerometer* ADXL345. Dari data yang didapatkan *accelerometer* ADXL345 dilakukan perhitungan untuk menentukan apakah pada aktifitas pengguna terjadi peristiwa melewati lubang, kemudian menentukan apakah dalam peristiwa tersebut terdapat lubang.



Gambar 3.5 Diagram Alir Menentukan Lubang

3.6.3 Diagram Alir Pengiriman Data

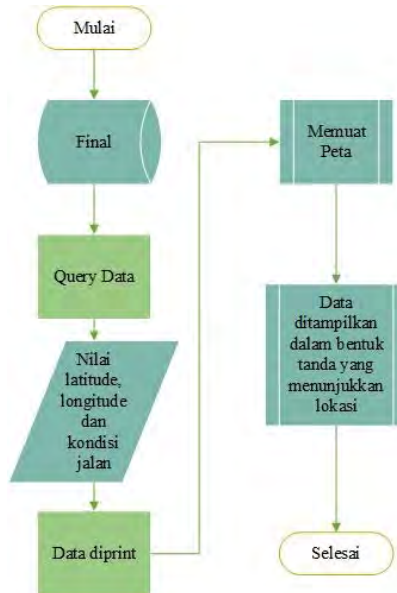
Gambar 3.6 menunjukkan diagram alir pengiriman data aliran sensor dan waktu dengan GPS/GPRS/GSM *Shield* V3.0 melalui jaringan GPRS sesuai *provider SIM Card* yang digunakan. Setelah dilakukan inisialisasi RTC, *Serial Port*, dan *Serial Digital*, maka tahap yang akan dilakukan oleh Arduino selanjutnya adalah melakukan koneksi GPRS. Hal ini dilakukan agar Arduino dapat berkomunikasi dengan *server* dan dapat saling berkirir data. Koneksi GPRS bisa didapatkan melalui GPS/GPRS/GSM *Shield* V3.0 yang telah dipasang bersama *SIM Card*. Masing-masing *SIM Card* memiliki *Access Point Names (APN)* yang berbeda beda, sehingga perlu diperhatikan dalam inisialisasi nama GPRS dan *APN SIM Card*.



Gambar 3.6 Diagram Alir Pengiriman Data

3.6.4 Diagram Alir Menampilkan Data pada Website

Proses aliran data dalam menjalankan *website* identifikasi dan pemetaan kondisi permukaan jalan dengan melakukan *request* HTTP GET kepada *server*, data disimpan ke dalam *database* dengan kondisi yang diberikan pada *server* yang menerima data dari dua rangkaian *mobile*. Data yang diterima dari dua rangkaian digabungkan di dalam tabel *final* pada *database*. Untuk menampilkan peta pada *website* dilakukan *query* pada data yang tersimpan di dalam *database* pada *server* secara langsung. Pada *web* digunakan *framework* Codeigniter 3.0.6. Data yang ditampilkan pada *website* adalah peta yang berisikan tanda, dimana tanda menunjukkan lokasi lubang. Diagram aliran data ditunjukkan oleh Gambar 3.7. Dalam saat menampilkan peta dilakukan pemanggilan pada *framework* leaflet. Leaflet merupakan *framework* untuk menampilkan peta pada *website*.



Gambar 3.7 Diagram Alir Menampilkan Data pada Website

3.7 Perancangan Basis Data Sistem

Basis data yang digunakan pada aplikasi *website* Sistem identifikasi dan pemetaan kondisi permukaan jalan ini adalah MySQL. Pada Sistem identifikasi dan pemetaan kondisi permukaan jalan ini diperlukan satu tabel *master*, yang digunakan untuk menyimpan data hasil keputusan akhir dari data uji coba yang akan di tampilkan dalam peta pada *website*. Tabel *master* tersebut akan dinamakan tabel *final*. Selain tabel *master* diperlukan dua tabel transaksi, yaitu tabel yang digunakan untuk menyimpan data yang diterima *server* yang dikirim oleh perangkat *mobile*. Tabel transaksi pertama akan dinamakan tabel *front*, tabel transaksi kedua akan dinamakan tabel *back*. Perancangan tabel master dapat dilihat pada Tabel 3.2, sedangkan untuk perancangan tabel transaksi dapat dilihat pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.2 Perancangan Tabel *Final*

No.	Nama Atribut	Tipe Data	Keterangan
1	id_final	int	Primary_key untuk tabel final.
2	latitude_final	float	Nilai latitude yang akan digunakan untuk menentukan lokasi pada peta.
3	longitude_final	float	Nilai longitude yang akan digunakan untuk menentukan lokasi pada peta.
4	altitude_final	float	Nilai altitude yang akan digunakan untuk menentukan lokasi pada peta.
5	status_final	varchar (20)	Hasil keputusan akhir yang berupa kondisi jalan.

6	waktu_final	timestamp	Waktu keputusan akhir dibuat.
7	hasildari_final	varchar (10)	Menunjukkan hasil keputusan akhir berasal dari penggabungan dua sumber (<i>front</i> dan <i>back</i>) atau tidak.

Tabel 3.3 Perancangan Tabel *Front*

No.	Nama Atribut	Tipe Data	Keterangan
1	id_front	int	Primary_key untuk tabel front.
2	latitude_front	float	Nilai latitude yang diterima dari perangkat <i>mobile</i> bagian depan sepeda motor.
3	longitude_front	float	Nilai longitude yang diterima dari perangkat <i>mobile</i> bagian depan sepeda motor.
4	altitude_front	float	Nilai altitude yang diterima dari perangkat <i>mobile</i> bagian depan sepeda motor.
5	status_front	varchar (20)	Hasil keputusan yang diterima dari perangkat <i>mobile</i> bagian depan sepeda motor.
6	waktu_front	timestamp	Waktu data diterima dari perangkat <i>mobile</i> bagian depan sepeda motor.

Tabel 3.4 Perancangan Tabel *Back*

No.	Nama Atribut	Tipe Data	Keterangan
1	id_back	int	Primary_key untuk tabel back.
2	latitude_back	float	Nilai latitude yang diterima dari perangkat <i>mobile</i> bagian belakang sepeda motor.
3	longitude_back	float	Nilai longitude yang diterima dari perangkat <i>mobile</i> bagian belakang sepeda motor.
4	altitude_back	float	Nilai altitude yang diterima dari perangkat <i>mobile</i> bagian belakang sepeda motor.
5	status_back	varchar (20)	Hasil keputusan yang diterima dari perangkat <i>mobile</i> bagian belakang sepeda motor.
6	waktu_back	timestamp	Waktu data diterima dari perangkat <i>mobile</i> bagian belakang sepeda motor.

(Halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan sistem identifikasi dan pemetaan kondisi permukaan jalan yang telah dijabarkan pada BAB III ANALISIS DAN PERANCANGAN. Cakupan implementasi dari perancangan sistem meliputi perangkat sistem *mobile* dalam membaca dan menentukan kondisi jalan, proses pengiriman data dan pengimplementasian antarmuka. Bahasa pemrograman yang digunakan adalah bahasa pemrograman arduino, dan PHP.

4.1 Lingkungan Implementasi

Lingkungan implementasi Tugas Akhir dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

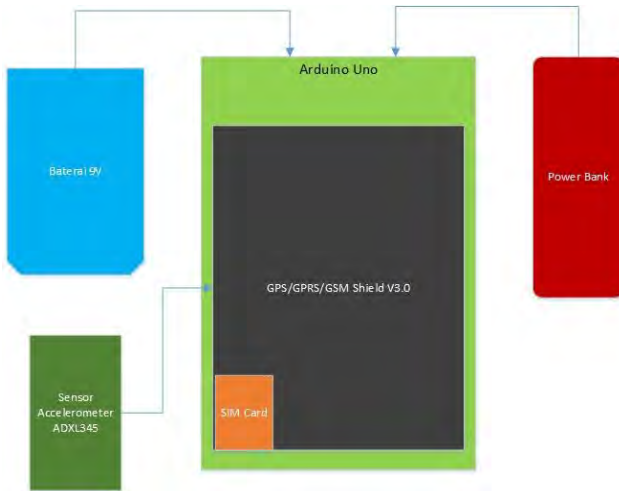
Perangkat	Spesifikasi
Perangkat Keras	Processor: Intel ® Core™ i3-3240M CPU @ 3.40 GHz Perangkat Sistem: Arduino Uno, GPS/GPRS/GSM <i>Shield</i> Arduino V3.0 , Sensor <i>Accelerometer</i> ADXL345
Perangkat Lunak	Sistem Operasi: Windows 8.1 Perangkat Pengembang: Arduino IDE, MySQL, Sublime Perangkat Pembantu: Microsoft Visio 2016, Microsoft Word 2013

4.2 Implementasi Perangkat Keras

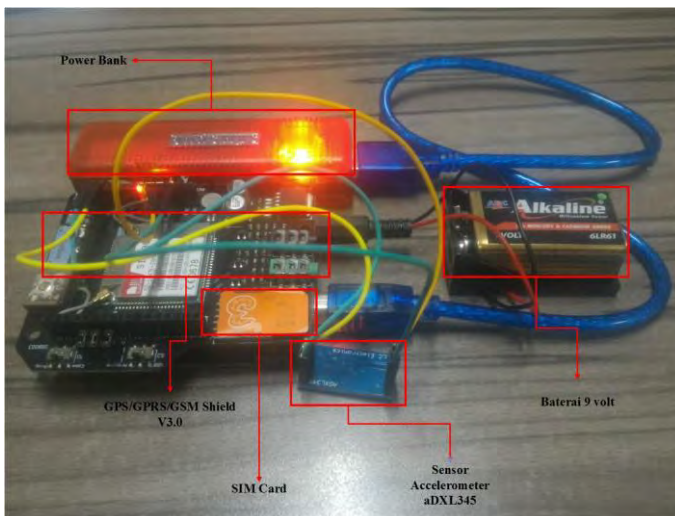
Implementasi perangkat keras sistem identifikasi dan pemetaan kondisi permukaan jalan adalah sebagai berikut:

- 2 (dua) buah Arduino Uno.
- 2 (dua) buah Sensor *Accelerometer* ADXL345.
- 2 (dua) buah GPS/GPRS/GSM *Shield* Arduino V3.0.
- 2 (dua) buah Baterai 9 volt.
- 2 (dua) buah *Power Bank*.
- Beberapa jumper.

Arduino Uno berperan sebagai pusat kontrol dan perhitungan pada sistem identifikasi dan pemetaan kondisi permukaan jalan, pusat penerimaan data sensor *accelerometer* ADXL345. Data yang diterima dari sensor *accelerometer* ADXL345 berupa nilai sumbu x, y dan z untuk mendapatkan hasil keputusan dan data GPS berupa latitude, longitude, altitude dari GPS/GSM/GPRS *shield* V3.0 yang akan dikirimkan melalui jaringan GPRS pada GPS/GSM/GPRS *shield* Arduino V3.0 menggunakan GPRS dari *provider SIM Card* yang digunakan, pada penelitian ini digunakan *SIM Card* dengan *provider* Tri. Perancangan perangkat keras dapat dilihat pada Gambar 4.1 dan rangkaian perangkat keras yang diimplementasikan sesuai dengan perancangan perangkat keras dapat dilihat pada Gambar 4.2 dan rangkaian perangkat keras tersebut dibuat dua buah. Rangkaian perangkat keras yang sudah selesai dibuat dimasukkan kedalam suatu *case* untuk menghindari kerusakan pada alat. Dua buah rangkaian perangkat keras tersebut disematkan pada sepeda motor, satu buah disematkan pada bagian depan sepeda motor dan satu buah yang lainnya disematkan pada bagian belakang sepeda motor. Rangkaian bagian depan dan belakang masing-masing akan mengirimkan data keputusan ke *server*, data dari kedua rangkaian akan diterima oleh *server* dan keputusan *final* diambil.



Gambar 4.1 Perangkat Keras Sistem identifikasi dan pemetaan kondisi jalan



Gambar 4.2 Rangkaian Perangkat Keras Sistem Sistem identifikasi dan pemetaan kondisi jalan

4.3 Implementasi Perangkat Lunak pada Perangkat Keras

Implementasi perangkat lunak di Arduino pada sistem identifikasi dan pemetaan kondisi permukaan jalan terdapat beberapa fungsi, yaitu sebagai pusat penerimaan sensor *accelerometer* ADXL345, sebagai pusat penerimaan GPS dari GPS/GPRS/GSM *Shield* Arduino V3.0, sebagai pengirim data sensor ke *server* menggunakan GPS/GPRS/GSM *Shield* Arduino V3.0, dan sebagai pusat perhitungan untuk menentukan kondisi jalan.

Sebelum mengimplementasikan keseluruhan perangkat lunak pada perangkat keras, yang penting untuk diperhatikan adalah melakukan uji coba kerja GPS/GPRS/GSM *Shield* Arduino V3.0. Dalam melakukan pengiriman data ke *server* dibutuhkan GSM pada GPS/GPRS/GSM *Shield* Arduino V3.0 yang mampu melakukannya dengan baik di jaringan GPRS. Untuk itu, dapat dilakukan pengujian perintah AT Command pada GPS/GPRS/GSM *Shield* Arduino V3.0 dengan AT Command Tester, yaitu layanan pengujian online yang disediakan oleh m2msupport.

Untuk implementasi penggunaan GSM dan GPRS pada GPS/GPRS/GSM *Shield* Arduino V3.0, yang perlu diperhatikan adalah catu daya yang digunakan, yaitu minimal mendapatkan catu daya 9 - 12 V dan kuat arus 1000mA. Pada penelitian ini, digunakan catu daya 9 V dengan kuat arus 1000 mA. Pengujian dengan menggunakan AT Command Tester penting dilakukan karena dapat membantu dalam mengetahui satu demi satu perintah yang dijalankan GPS/GPRS/GSM *Shield* Arduino V3.0 dan dapat mengetahui kendala yang terjadi pada GPS/GPRS/GSM *Shield* Arduino V3.0. Selain catu daya ada hal lain yang perlu diperhatikan, yaitu kartu *provider* yang digunakan. Berdasarkan hasil uji coba, kartu *provider* yang digunakan dan berfungsi dengan baik pada GPS/GPRS/GSM *Shield* Arduino V3.0 untuk wilayah Indonesia yaitu kartu *provider* "Tri".

4.4 Implementasi Perangkat Lunak untuk Membaca Sensor *Accelerometer* ADXL345

Implementasi perangkat lunak sensor *accelerometer* ADXL345 pada Arduino dilakukan dengan menggunakan dengan menggunakan *library* dari *Adafruit*, *library adafruit* digunakan untuk menentukan *range* pengukuran mulai dari $\pm 2g$, $\pm 4g$, $\pm 8g$ hingga $\pm 16g$. Di mana tiap $1g$ merupakan satu satuan percepatan rata-rata gravitasi bumi yaitu $9,8m/s^2$. Dalam penelitian ini menggunakan tingkat sensitivitas sebesar $\pm 4g$. Selain itu *library* *Adafruit* digunakan untuk proses kalibrasi sumbu x, y, z pada sensor *accelerometer* ADXL345 [9]. Pada penelitian ini digunakan *baud rate* 9600 untuk Arduino dan GPS/GPRS/GSM *Shield* Arduino V3.0. *Baud rate* adalah banyaknya symbol yang ditransfer selama 1 detik untuk transmisi data serial [10].

Implementasi perangkat lunak untuk membaca Sensor *accelerometer* ADXL345 pada Arduino dapat dilihat pada Gambar 4.3 dan Gambar 4.4. Sebelum membaca nilai dari *accelerometer* ADXL345, harus dilakukan setup untuk mengaktifkan sensor *accelerometer* ADXL345. Prosedur setup *accelerometer* ADXL345 dapat dilihat pada Gambar 4.3.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: -
<pre>Function setup { Serial.begin(baudrate) If(!accelerometerbegin) then Print "no ADXL345 detected" While(1) endif SetRangeAccelerometer(ADXL345_RANGE_4_G) }</pre>
Keluaran:-

Gambar 4.3 Implementasi Setup Sensor *Accelerometer* ADXL345

Setelah diinisialisasi *accelerometer* ADXL345 dapat dilakukan proses membaca nilai dari sumbu x, y dan z pada *accelerometer* ADXL345. Selama sensor *accelerometer* ADXL345 tidak mati, meskipun terjadi perubahan ataupun tidak terjadi perubahan, sensor *accelerometer* ADXL345 akan terus membaca percepatan linier sumbu x, y dan z. Nilai sumbu x, y, dan z yang terbaca masih belum benar oleh karena itu dilakukan kalibrasi perangkat lunak untuk memperbaiki pembacaan sensor *accelerometer* ADXL345. Kecepatan *accelerometer* ADXL345 dalam membaca nilai sumbu x, y dan z dapat diatur dengan memberikan jeda waktu dalam proses membaca nilai sesuai dengan kebutuhan. Pada penelitian jeda waktu yang digunakan yaitu 250 ms. Jeda waktu 250 ms diambil berdasarkan uji coba pengambilan data yang disimpan menggunakan *micro* SD, jika data yang diambil menggunakan jeda waktu kurang dari 250 ms maka banyak data hilang saat pengumpulan data, karena *latency* kecepatan *write micro* SD tidak dapat lebih cepat dari 250 ms. Prosedur pembacaan nilai dari sumbu x, y dan z *accelerometer* ADXL345 dapat dilihat pada Gambar 4.4.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: -
<pre>Function Loop { Variable float x, y, z x <- acceleration.getEvent.x() y <- acceleration.getEvent.y() z <- acceleration.getEvent.z() delay(time) }</pre>
Keluaran: Nilai Sumbu x, y, z

Gambar 4.4 Implementasi Mendapatkan Nilai Sensor *Accelerometer* ADXL345

4.5 Implementasi Perangkat Lunak untuk Mendapatkan Nilai GPS

Implementasi perangkat lunak untuk mendapatkan nilai GPS pada GPS/GPRS/GSM *Shield* Arduino V3. AT Command digunakan pada saat menginisialisasi dan mengaktifkan GPS. Berikut ini adalah macam-macam AT *Command* yang digunakan untuk mengaktifkan GPS.

- AT+CGPSIPR: Digunakan untuk mengatur nilai *baud rate*.
- AT+CGPSPWR: Digunakan untuk menyalakan *power supply* dari GPS.
- AT+CGPSRST: Digunakan untuk *reset* GPS dari *autonomy mode*.
- AT+CGPSSTATUS?: Digunakan untuk memeriksa status GPS.

Perintah AT tersebut kemudian dibungkus didalam kode program pada Arduino untuk inisialisasi dan mengaktifkan GPS. Implementasi perangkat lunak untuk mendapatkan nilai GPS dapat dilihat pada Gambar 4.5.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: -
<pre> Function Start_GPS { AT+CGPSIPR=9600 AT+CGPSPWR=1 AT+CGPSRST=1 AT+CGPSSTATUS? } Function setup { Start_GPS() Enable GPS mode Disable GSM mode } </pre>

<pre> Function Loop { latitude = Getlatitude(); longitude = Getlongitude(); altitude = Getaltitude(); } </pre>
Keluaran: Nilai latitude, longitude dan altitude

Gambar 4.5 Implementasi Mendapatkan Nilai pada GPS

Data GPS yang didapatkan dari GPS/GSM/GPRS *Shield* Arduino V3.0 berupa latitude, longitude dan altitude. Data posisi yang berupa latitude, longitude dan altitude didapatkan setelah GPS/GSM/GPRS *Shield* Arduino V3.0 terhubung dengan beberapa satelit.

4.6 Implementasi Perangkat Lunak untuk Mengirim Data

Data yang akan dikirimkan yaitu, data posisi dari GPS yang berupa latitude, longitude dan altitude, data hasil keputusan kondisi jalan yang berupa lubang, serta asal dari rangkaian alat yang akan mengirim data menunjukkan data yang diterima *server* berasal dari rangkaian alat bagian depan atau belakang sepeda motor berupa *front* atau *back*.

Berikut ini adalah macam-macam perintah AT yang digunakan untuk mengirimkan data ke *server* melalui HTTP :

- AT+CREG: registrasi jaringan *SIM Card*, respon yang diberikan adalah “OK”.
- AT+SAPBR: melakukan konfigurasi profil GPRS berdasarkan IP APN *SIM Card*, respon yang diberikan adalah “OK”.
- AT+HTTPINIT: inisialisasi layanan HTTP, respon yang diberikan adalah “OK”.
- AT+HTTPPARA: menetapkan nilai parameter HTTP, respon yang diberikan adalah “OK”.
- AT+HTTPACTION: HTTP *Method Action*, respon yang diberikan adalah “OK”.

- AT+HTTPTERM: memutus layanan HTTP, respon yang diberikan adalah “OK”.

Perintah AT tersebut kemudian dibungkus didalam kode program pada Arduino untuk inialisasi dan proses pengiriman data. Implementasi perangkat lunak untuk mengirim data ditunjukkan oleh Gambar 4.6.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: latitude, longitude, altitude,
<pre> Function Start_GSM() { Serial.println("AT+CREG?") AT+SAPBR=3,1,\"APN\", \"3gprs\" AT+SAPBR=3,1,\"USER\", \"3gprs\" AT+SAPBR=3,1,\"PWD\", \"3gprs\" AT+SAPBR=3,1,\"Contype\", \"GPRS\" AT+SAPBR=1,1 AT+HTTPIINIT AT+HTTPPARA=\"CID\",1\" AT+CGPSPWR=1 AT+CGPSRST=1 AT+CGPSSTATUS? } Function setup { Start_GSM() Enable GSM mode Disable GPS mode } Function Loop { If(status=lubang) then latitude = Getlatitude() longitude = Getlongitude() altitude = Getaltitude() AT+HTTPPARA=\"URL\", \"fauzan.yusufnugroho.com/Getdata? latitude = \"); print(colat) </pre>

```

    print("&longitude =")
    print(corlon)
    print("&altitude=")
    print(coralto)
print("&status=")
    print(kondisi)
print("&asal=")
    print(asal)
    AT+HTTPACTION=0
endif
}

```

Keluaran: Nilai latitude, longitude dan altitude

Gambar 4.6 Implementasi Perangkat Lunak Mengirim Data

Dalam implementasi pengiriman data, pengiriman data hanya akan terjadi jika nilai status bernilai lubang, jika status tidak memiliki nilai tersebut maka proses pengiriman data tidak akan dilakukan.

4.7 Implementasi Mendapatkan dan Menggunakan *Data Training*

Data *training* didapatkan dengan menyimpan data sumbu x , y , dan z yang telah dibaca oleh *accelerometer* ADXL345 saat pengguna melakukan aktifitas. Data training yang telah didapatkan digunakan untuk menentukan nilai *threshold*. Untuk menyimpan nilai sumbu x , y , dan z yang dibaca oleh *accelerometer* ADXL345 pada rangkaian alat ditambahkan modul *micro SD card*. Implementasi rangkaian alat yang telah ditambahkan modul *micro SD card* dapat dilihat pada . Pada saat pembacaan dan penyimpanan data training posisi diletakkannya rangkaian alat harus sama dengan posisi saat rangkaian digunakan untuk menentukan kondisi jalan. Rangkaian alat diletakkan pada bagian depan sepeda motor dan bagian belakang sepeda motor.

Pada penelitian ini dikumpulkan data sensor ketika pengguna sedang melewati lubang saat menggunakan sepeda motor. Data diambil menggunakan sensor *accelerometer* yang kemudian nilai dari sumbu x , y , z , $x_{i+1}-x_i$, $y_{i+1}-y_i$, dan $z_{i+1}-z_i$,

disimpan ke dalam *micro* SD. Data training melewati lubang dapat dilihat pada Tabel 4.1 dan grafiknya pada Gambar 4.9. Pada Gambar 4.10 merupakan nilai selisih dari data training yang telah didapatkan.

Prosedur Pengumpulan Data Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: -
<pre> Awal <- 1; Function Loop { Variable float x, y, z x <- acceleration.getEvent.x() y <- acceleration.getEvent.y() z <- acceleration.getEvent.z() File = SD.open(filename, FILE_WRITE); if (awal==1) then x1 <- 0; x2 <- x; xtemporary<-x; y1 <- 0; y2 <- y; ytemporary<-y; z1<-0; z2 <- z; ztemporary <- z; awal <- 0; else x1 <- xtemporary; xtemporary <- x; x2 <- x; xdifference <- absolute(x2-x1); y1 <- ytemporary; ytemporary<-y; y2 <- y; ydifference<-absolute (y2-y1); z1 <- ztemporary; ztemporary<-z; z2 <- z; zdifference <- absolute (z2-z1); if(File) then </pre>

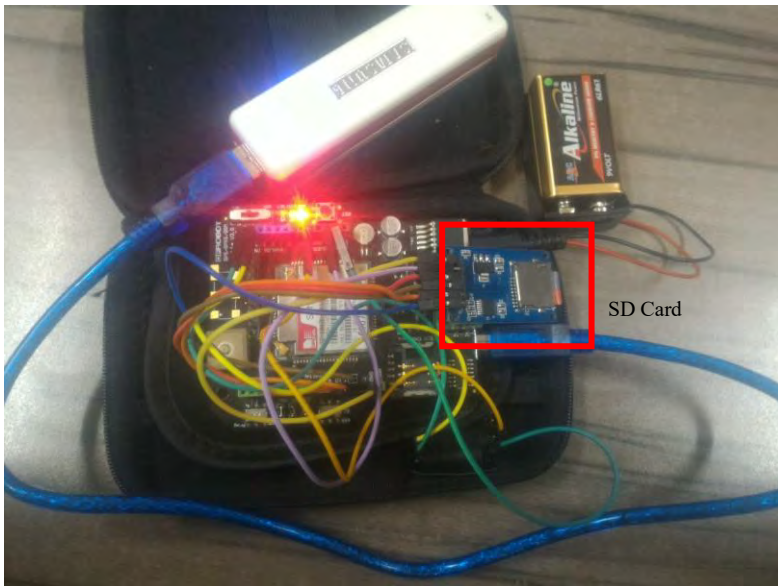
```

File.print(xdifference);
File.print(",");
File.print(ydifference);
File.print(",");
File.print(zdifference);
File.print(",");
File.println("lubang");
dataFile.close();
endif
endif
delay(time);
}

```

Keluaran: sumbu x, sumbu y, sumbu z, $X_{i+1}-X_i$, $Y_{i+1}-Y_i$, $Z_{i+1}-Z_i$

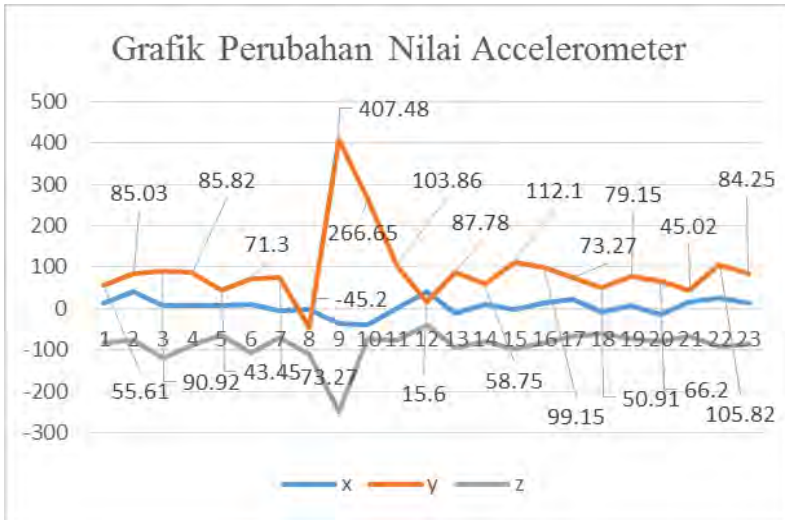
Gambar 4.7 Menyimpan Data Kondisi Jalan pada SD Card



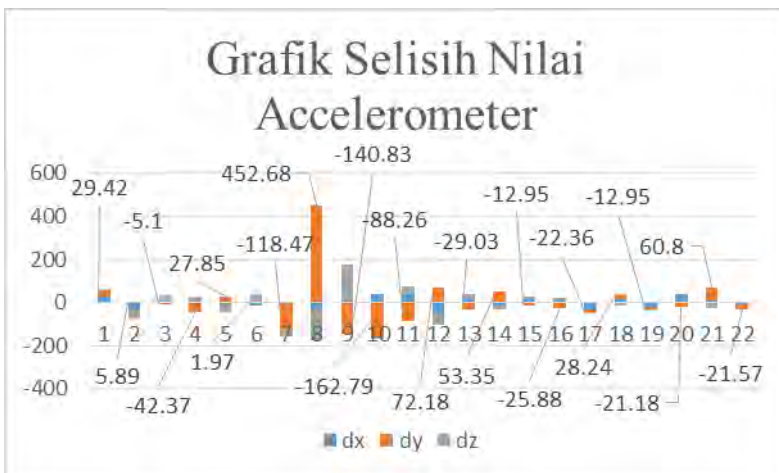
Gambar 4.8 Modul Micro SD Card pada Rangkaian Alat

Tabel 4.1 Data Training Melewati Lubang

Sumbu X	Sumbu X	Sumbu X	$X_{i+1}-X_i$	$Y_{i+1}-Y_i$	$Z_{i+1}-Z_i$
13.94	55.61	-81.33	26.68	29.42	3.92
40.62	85.03	-77.41	-34.13	5.89	-41.97
6.49	90.92	-119.38	-0.39	-5.1	31.38
6.1	85.82	-88	1.96	-42.37	23.54
8.06	43.45	-64.46	1.96	27.85	-41.98
10.02	71.3	-106.44	-14.51	1.97	35.31
-4.49	73.27	-71.13	1.96	-118.47	-38.44
-2.53	-45.2	-109.57	-34.13	452.68	-139.65
-36.66	407.48	-249.22	-2.74	-140.83	175.34
-39.4	266.65	-73.88	39.62	-162.79	-3.14
0.22	103.86	-77.02	40.79	-88.26	36.48
41.01	15.6	-40.54	-51.39	72.18	-54.13
-10.38	87.78	-94.67	21.58	-29.03	16.08
11.2	58.75	-78.59	-14.12	53.35	-19.61
-2.92	112.1	-98.2	14.9	-12.95	15.69
11.98	99.15	-82.51	10.2	-25.88	14.12
22.18	73.27	-68.39	-29.81	-22.36	6.67
-7.63	50.91	-61.72	12.94	28.24	-12.94
5.31	79.15	-74.66	-21.18	-12.95	-3.93
-15.87	66.2	-78.59	30.6	-21.18	10.99
14.73	45.02	-67.6	9.41	60.8	-25.11
24.14	105.82	-92.71	-10.2	-21.57	6.28
13.94	84.25	-86.43			



Gambar 4.9 Grafik Perubahan Nilai Accelerometer Ketika Melewati Lubang



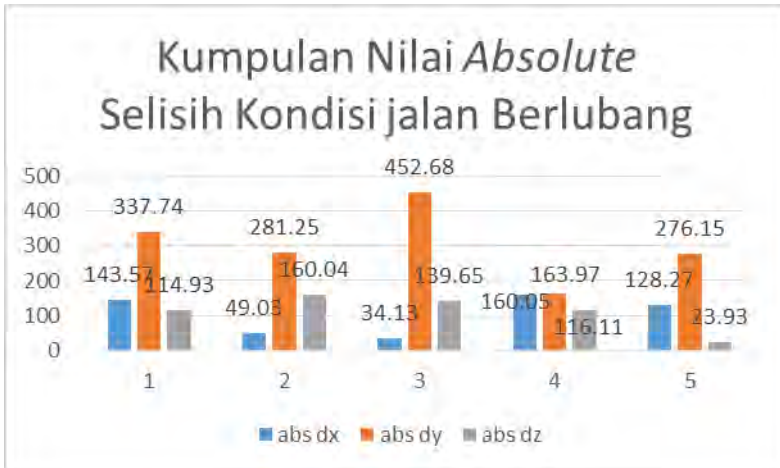
Gambar 4.10 Grafik Nilai Selisih Setiap Perubahan Nilai Accelerometer Melewati Lubang

Pada Gambar 4.10 merupakan grafik perubahan nilai *accelerometer* yang didapat pada Tabel 4.1, dari data perubahan yang telah didapat kemudian dihitung selisih nilai perubahan tiap sumbu yang ada, setiap data yang didapat dikurangkan dengan nilai yang didapatkan sebelumnya. Maka perhitungan dilakukan menjadi $X_{i+1} - X_i$. X merepresentasikan sumbu *accelerometer*, I merepresentasikan urutan didapaknya data. Melalui nilai selisih perubahan akan dicari *threshold* berdasarkan nilai yang menunjukkan peristiwa melewati lubang.

4.8 Implementasi Menentukan *Threshold*

Untuk mendapatkan nilai *threshold*, maka dilakukan cara sebagai berikut:

1. Dari data *training* yang telah didapatkan untuk kondisi saat pengguna melewati jalan normal atau berlubang diambil selisih nilai perubahan percepatan [11].
2. Untuk kondisi saat pengguna melewati jalan normal dilihat berapa nilai maksimum dan minimum selisih perubahan kecepatan.
3. Untuk kondisi saat pengguna melewati lubang, dari data yang didapat dipilih bagian nilai selisih perubahan percepatan yang menunjukkan bahwa pengguna melewati lubang.
4. Dari nilai selisih tersebut ditentukan nilai *absolute Difference* = $|X_{i+1} - X_i|$.
5. Nilai *absolute* yang menunjukkan kondisi lubang dikumpulkan dapat dilihat pada Gambar 4.11, dari sumbu x , y , z diambil salah satu yang mengalami perubahan paling besar dibandingkan dengan sumbu yang lain, pada rangkaian ini, diambil nilai maksimum dan minimum dari sumbu y . Nilai tersebut digunakan sebagai nilai *threshold*. Nilai minimum untuk batas bawah sedangkan nilai maksimum untuk batas atas.



Gambar 4.11 Kumpulan Nilai Absolute Selisih Kondisi Jalan Berlubang

4.9 Implementasi Menentukan Lubang

Sebelum mengirimkan data ke *server*, ada nilai status kondisi jalan yang harus didapatkan. Secara umum kerja menentukan lubang adalah sebagai berikut:

- 1) Pertama, sensor *accelerometer* menerima data berupa sumbu x , y , dan z .
- 2) Jika perubahan pembacaan sensor memenuhi kondisi pertama maka aktivitas pengguna diklasifikasikan sebagai aktivitas statis atau berjalan normal. Sebaliknya, maka aktivitas pengguna akan diasumsikan sebagai transisi dinamis.

Selanjutnya dilakukan pengecekan transisi dinamis pengguna, apakah transisi pergerakan pengguna dari aktivitas statis menuju aktivitas transisi dinamis mengindikasikan keadaan melewati lubang. Untuk membedakan keadaan melewati jalan normal dengan keadaan melewati lubang, maka dibandingkan antara *nilai* selisih perubahan percepatan nilai x , y , z dengan *threshold*. Untuk mendapatkan nilai *threshold*, maka

dibandingkan antara data *training* sensor *accelerometer* kondisi pengguna melewati jalan normal dengan sensor *accelerometer* dan saat kondisi pengguna melewati lubang. Implementasi untuk mendapatkan nilai status kondisi jalan dapat dilihat pada Gambar 4.12.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: sumbu x, sumbu y, sumbu z,
<pre> i <- 1 flag <- normal Function Loop { if (i=1) then y1 <- 0 y2 <- sumbu y ytemporary <- sumbu y else y1 <- ytemporary ytemporary <- sumbu y y2 <- sumbu y ydifference <- absolute(y2-y1) if(flag!=lubang) then if(ydifference >= floorthresholdylubang and ydifference <= ceilingthresholdylubang) then status <- lubang flag <-lubang endif endif endif endif else flag <- Normal endif i <- 0 endif </pre>
Keluaran:Nilai status kondisi jalan

Gambar 4.12 Implementasi Menentukan Jalan Berlubang

4.10 Implementasi Rangkaian pada Sepeda Motor

Untuk melakukan penentuan lubang pada kondisi jalan yang sebenarnya rangkaian diimplementasikan pada sepeda motor, rangkaian yang telah di implementasikan pada sepeda motor dapat dilihat pada dan .



Gambar 4.13 Implementasi Rangkaian Depan pada Sepeda Motor



Gambar 4.14 Implementasi Rangkaian Belakang pada Sepeda Motor

4.11 Implementasi Penerimaan Data pada *Server*

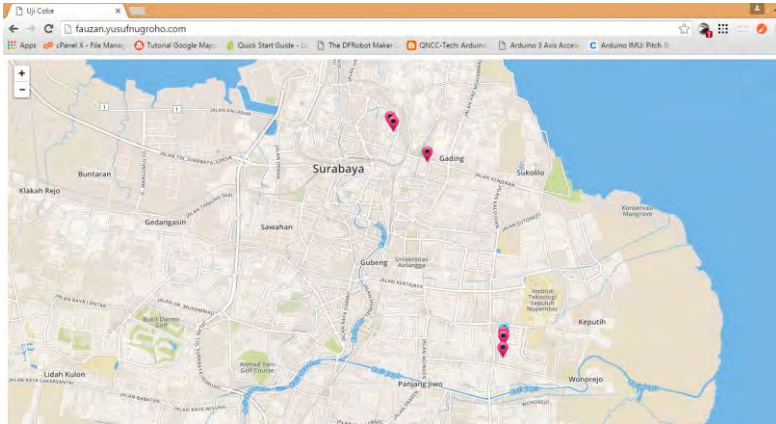
Data yang dikirimkan oleh rangkaian alat ketika berada di jalan akan diterima oleh *server*. Implementasi pada *server* untuk menerima data tersebut menggunakan bahasa pemrograman PHP dan menggunakan *framework* Codeigniter 3.0.6. Dari data yang diterima diambil parameternya. Untuk masing-masing parameter dimasukkan ke dalam variabel yang sesuai. Data dari parameter yang didapatkan berupa hasil data yang dikirim dari rangkaian bagian depan dan belakang. Setelah data rima data tersebut dimasukkan ke *database* yang telah dibuat. Implementasi penerimaan data oleh *server* dapat dilihat pada Gambar 4.15.

Prosedur Sistem Identifikasi dan Pemetaan Kondisi Permukaan Jalan
Masukan: latitude, longitude, altitude, status, asal
<pre>getParameter (\$_GET[\$parameter]) latitude <- \$_GET['latitude'] longitude <- \$_GET['longitude'] altitude <- \$_GET['altitude'] status <- \$_GET['status'] asal <- \$_GET['asal']</pre>
Keluaran: latitude, longitude, altitude, status, asal

Gambar 4.15 Implementasi Penerimaan Data pada Server

4.12 Implementasi Menampilkan Peta pada *Website*

Data yang telah diterima dan masuk pada *database* akan ditampilkan pada *website* berupa peta yang memiliki tanda berwarna merah yang menunjukkan kondisi jalan berlubang. Untuk menampilkan peta pada *website* digunakan *framework* Leaflet. Tampilan antarmuka peta pada *website* dapat dilihat pada Gambar 4.16.



Gambar 4.16 Tampilan Antarmuka Peta Kondisi Jalan pada *Website*

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem identifikasi dan pemetaan kondisi permukaan jalan yang dikembangkan. Sistem akan diuji coba fungsionalitas dan performa dengan menjalankan skenario yang sudah ditentukan. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini. Uji coba akan dilakukan dengan dua tipe pengujian yaitu uji coba fungsionalitas dan uji coba performa. Pengujian fungsionalitas meliputi uji coba setiap bagian perangkat keras yang dirangkai pada Arduino dan uji coba keseluruhan sistem. Pengujian performa meliputi akurasi sensor *accelerometer* ADXL345, dan sistem menentukan jalan berlubang.

5.1 Uji Coba Fungsionalitas

Uji coba fungsionalitas dilakukan untuk mengetahui fungsi dasar masing-masing komponen baik perangkat keras yang dirangkai berjalan sebagaimana mestinya.

5.1.1 Lingkungan Uji Coba

Pada subbab ini akan dijelaskan mengenai lingkungan uji coba sistem identifikasi dan pemetaan kondisi permukaan jalan. Uji coba sensor *accelerometer* ADXL345 dilakukan dengan dihubungkan secara langsung dengan GPS/GPRS/GSM *Shield* Arduino V3.0. Kemudian rangkaian diletakkan pada sepeda motor yang akan berjalan. Data GPS dari GPS/GPRS/GSM *Shield* Arduino V3.0 harus didapatkan terlebih dahulu sebelum sepeda motor mulai berjalan dan sensor *accelerometer* membaca perubahan aktivitas sehingga proses ujicoba dapat dijalankan dengan lancar. Pengiriman data yang dilakukan melalui jaringan GPRS yang didapatkan dari jaringan *SIM Card* yang berada di GPS/GPRS/GSM *Shield* Arduino V3. Lingkungan uji coba memiliki spesifikasi seperti pada Tabel 5.1.

Tabel 5.1 Lingkungan Uji Coba

Perangkat Keras (<i>Server</i>)	
Hosting	NiagaHoster
Perangkat Bergerak	
Sepeda Motor	Suzuki Shogun 125 RR
Alat	Rangkaian dari: <ul style="list-style-type: none"> - Sensor <i>accelerometer</i> ADXL345 - Arduino Uno - GPS/GPRS/GSM <i>Shield</i> Arduino V3.0. - Baterai 9 volt - Power bank

5.1.2 Uji Coba Data dari Perangkat Bergerak ke *Server*

Pada sistem identifikasi dan pemetaan kondisi permukaan jalan, saat pengguna sedang berjalan terdapat proses pembacaan perubahan aktivitas oleh *accelerometer* ADXL345 dan juga pembacaan GPS oleh GPS/GPRS/GSM *Shield* Arduino V3.0. Dari data *accelerometer* kemudian dibandingkan dengan *threshold* untuk menentukan keputusan. Hasil keputusan dikirimkan ke *server* beserta dengan data posisi yang didapatkan dari GPS. Data yang dikirim diterima oleh *server* dan dimasukkan ke *database*, dari data yang tersimpan di *database* dilakukan *query* dan ditampilkan dalam peta. Untuk itu dilakukan uji coba apakah data yang dibaca dan hasil dari keputusan berhasil diterima oleh *server*. Untuk skenario uji coba perangkat bergerak dapat dilihat pada Tabel 5.2.

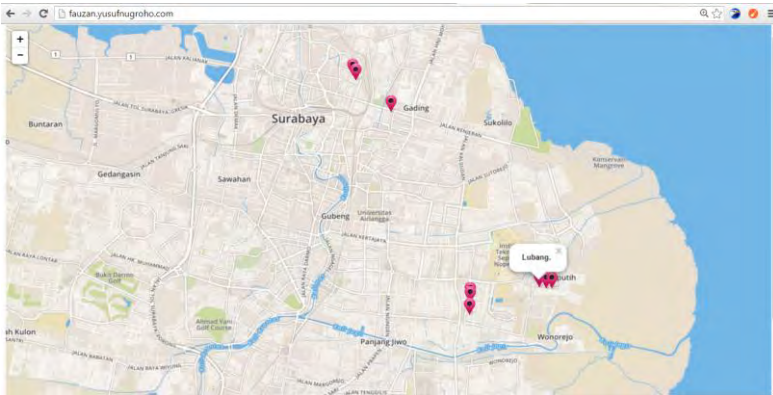
Tabel 5.2 Data dari Perangkat Bergerak ke *Server*

Nama	Uji coba data dari perangkat bergerak ke <i>server</i>
Tujuan	Mendapatkan data dari perangkat bergerak pada <i>server</i>
Skenario	Pengguna mengaktifkan sistem sistem identifikasi dan pemetaan kondisi permukaan jalan pada sepeda motor yang telah disematkan rangkain perangkat bergerak dan berjalan pada beberapa jalan.
Hasil Uji Coba	<i>Server</i> menerima data yang dikirimkan dari perngkat bergerak.

Hasil uji coba data dari perangkat bergerak ke *server* dapat dilihat pada Gambar 5.1. Pada uji coba ini, pengguna dapat melihat data yang dikirimkan ke *server* di tampilkan oleh penanda pada peta, peta dapat dilihat pada Gambar 5.1. Data yang ditampilkan berupa data yang didapatkan dari kedua rangkaian alat yang mengirimkan hasil keputusannya masing-masing, kemudian data diterima dan digabungkan pada server. Data yang terdapat pada tabel *final* ditampilkan pada *website* yang berupa tanda menunjukkan posisi lubang pada peta.

id_final	latitude_final	longitudo_final	altitudo_final	status_final	waktu_final	hasildari_final
3987331	-7.23402	112.75383	22.22213	lubang	2016-06-11 18:16:47	gabung
3987332	-7.23541	112.75477	22.22213	lubang	2016-06-11 18:20:53	gabung
3987333	-7.24399	112.76443	22.22216	lubang	2016-06-11 18:54:33	gabung
3987334	-7.29331	112.78612	22.22216	PT	2016-06-11 19:30:51	gabung
3987335	-7.29451	112.78598	22.22216	lubang	2016-06-11 19:35:08	gabung
3987336	-7.29575	112.78595	22.22216	lubang	2016-06-11 19:47:04	gabung
3987337	-7.29904	112.78584	22.22216	lubang	2016-06-11 19:53:57	gabung

Gambar 5.1 Database Menerima Data dari Perangkat bergerak



Gambar 5.2 Peta Menampilkan Kondisi Jalan dengan Penanda

5.2 Uji Coba Performa

Uji coba performa sistem dilakukan untuk mengetahui seberapa akurat dan tepat sistem yang dibangun. Uji coba performa sistem yang dilakukan meliputi berapa keputusan yang dihasilkan benar sesuai dengan kondisi jalan sebenarnya.

5.2.1 Uji Coba Akurasi Melewati Lubang

Uji performa akurasi sistem ini dilakukan untuk mengetahui akurasi ketepatan sistem dalam menentukan kondisi jalan yang berlubang sesuai dengan keadaan sebenarnya. Uji akurasi sensor perlu dilakukan karena lubang merupakan kondisi jalan yang akan ditampilkan di peta. Untuk skenario uji coba akurasi sistem dalam menentukan lubang dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario Uji Coba Akurasi Melewati Lubang

Nama	Uji coba akurasi melewati lubang
Tujuan	Mengetahui akurasi sistem menentukan lubang
Skenario	Uji coba dilakukan dengan menjalankan sepeda motor melewati lubang di jalan

Tabel 5.4 Uji Coba Melewati Lubang Rangkaian Bagian Depan

Uji Coba	Hasil Keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	Lubang	Lubang	6	5
2	-	Lubang	6	1

Tabel 5.5 Uji Coba Melewati Lubang Rangkaian Bagian Belakang

Uji Coba	Hasil Keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	Lubang	Lubang	6	5
2	-	Lubang	6	1

Tabel 5.6 Uji Coba Melewati Lubang Keputusan *Final*

Uji Coba	Hasil keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	Lubang	Lubang	6	5
2	-	Lubang	6	1

5.2.2 Uji Coba Akurasi Melewati Jalan Normal

Uji performa akurasi sistem ini dilakukan untuk mengetahui akurasi ketepatan sistem dalam menentukan kondisi jalan yang tidak terdapat lubang sesuai dengan keadaan sebenarnya. Uji akurasi sensor perlu dilakukan karena jalan normal merupakan kondisi jalan yang akan tidak terdapat penanda saat ditampilkan di peta. Uji coba dilakukan dengan cara melewati jalan normal yang tidak terdapat lubang. Untuk skenario uji coba akurasi sistem dalam dapat dilihat pada Tabel 5.7.

Tabel 5.7 Skenario Uji Coba Akurasi Melewati Jalan Normal

Nama	Uji coba akurasi melewati Jalan Normal
Tujuan	Mengetahui akurasi sistem dengan melihat pada jalan yang tidak terdapat lubang, alat masih mendeteksi terdapatnya lubang
Skenario	Uji coba dilakukan dengan menjalankan sepeda motor jalan normal yang tidak terdapat lubang.

Tabel 5.8 Uji Coba Melewati Jalan Normal Rangkaian Bagian Depan

Uji Coba	Hasil Keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	Lubang	Normal	0	2
2	Normal	Normal	0	378

Tabel 5.9 Uji Coba Melewati Jalan Normal Rangkaian Bagian Belakang

Uji Coba	Hasil Keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	Lubang	Normal	0	2
2	Normal	Normal	0	376

Tabel 5.10 Uji Coba Melewati Jalan Normal Keputusan Final

Uji Coba	Hasil Keputusan Sistem	Kondisi Jalan Sebenarnya	Jumlah Lubang	Jumlah Keputusan
1	lubang	Normal	0	2
2	Normal	Normal	0	376

5.2.3 Uji Coba Melewati Polisi Tidur

Uji coba ini dilakukan sebagai bentuk evaluasi dan pembandingan terhadap sistem yang telah dibuat, apakah mampu dalam menentukan kondisi selain lubang. Nilai *threshold* didapatkan dengan cara yang sama seperti mendapatkan nilai *threshold* lubang. Uji coba dilakukan dengan menambahkan kondisi jika memenuhi *threshold* polisi tidur yang telah ditentukan pada kode menentukan lubang. Uji performa akurasi sistem ini dilakukan untuk mengetahui akurasi ketepatan sistem dalam menentukan kondisi jalan yang terdapat polisi tidur sesuai dengan keadaan sebenarnya. Untuk skenario uji coba akurasi sistem dalam menentukan polisi tidur dapat dilihat pada Tabel 5.11 Skenario Uji Coba Akurasi Melewati Polisi Tidur.

Tabel 5.11 Skenario Uji Coba Akurasi Melewati Polisi Tidur

Nama	Uji coba akurasi melewati Polisi Tidur
Tujuan	Mengetahui akurasi sistem menentukan Polisi Tidur
Skenario	Uji coba dilakukan dengan menjalankan sepeda motor melewati polisi tidur

Tabel 5.12 Uji Coba Melewati Polisi Tidur Rangkaian Bagian Depan

Uji Coba	Hasil keputusan sistem	Kondisi Jalan Sebenarnya	Jumlah Polisi Tidur	Jumlah Keputusan
1	Lubang	Polisi Tidur	3	3
2	Normal	Polisi Tidur	3	0

Tabel 5.13 Uji Coba Melewati Polisi Tidur Rangkaian Bagian Belakang

Uji Coba	Hasil keputusan sistem	Kondisi Jalan Sebenarnya	Jumlah Polisi Tidur	Jumlah Keputusan
1	Lubang	Polisi Tidur	6	3
2	Normal	Polisi Tidur	6	0

Tabel 5.14 Uji Coba Melewati Polisi Tidur Keputusan Final

Uji Coba	Hasil keputusan sistem	Kondisi Jalan Sebenarnya	Jumlah Polisi Tidur	Jumlah Keputusan
1	Lubang	Polisi Tidur	6	3
2	Normal	Polisi Tidur	6	0

5.3 Hasil Uji Coba

Dari analisis uji coba yang telah dilakukan maka didapatkan hasil akurasi untuk alat rangkaian depan, rangkaian belakang dan keputusan *final* pada *server*. Untuk menentukan hasil uji coba performa maka dibuat pembagian seperti pada Tabel 5.15, Hasil dapat dilihat pada Tabel 5.16.

Tabel 5.15 Representasi Keputusan dengan Kondisi Sebenarnya

Keputusan	Kondisi Sebenarnya	
	Lubang	Normal
Lubang	TP	FN
Normal	FP	TN

TP = *True Positive*

FN = *False Negative*

FP = *False Positive*

TN = *True Negative*

Tabel 5.16 Hasil Uji Coba Sistem

Kode Pengujian	True Positives	True Negative	False Positives	False Negatives
Front	5	378	1	2
Back	5	376	1	2
Final	5	376	1	2

$$\text{Presisi} = \frac{TP}{(TP + FP)} \times 100\% = \frac{5}{(5 + 1)} \times 100\% = 83,33\%$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \times 100\% = \frac{5}{(376 + 2)} \times 100\% = 99,47\%$$

$$\text{Akurasi} = \frac{TP + TN}{(TP + TN + FP + FN)} \times 100\%$$

$$= \frac{5 + 376}{(5 + 376 + 1 + 2)} \times 100\% = 99,73\%$$

Berdasarkan data dan perhitungan diatas, nilai presisi dari sistem yaitu 83,33%, untuk nilai *recall* sistem yaitu 99,47%, dan

nilai keakuratan keseluruhan sistem dalam menentukan keputusan yang benar yaitu 99,73 %.

5.4 Uji Coba Delay Waktu Pengiriman

Berdasarkan data yang diterima di *server* dari rangkaian depan dan belakang delay waktu pengiriman dapat dilihat. Data yang diterima dari rangkaian bagian depan dapat dilihat pada Tabel 5.17 dan data yang diterima dari rangkaian belakang dapat dilihat pada Tabel 5.18 .

Tabel 5.17 Data dari Rangkaian Bagian Depan

Id_front	Latitude_front	Longitude_front	Altitude_front	Status_front	Waktu_front
1	-7.77000	112.32100	20.22300	lubang	2016-06-11 18:15:52
53	-7.29654	112.78593	22.22216	lubang	2016-06-11 19:30:11
57	-7.29001	112.78582	22.22216	lubang	2016-06-11 19:46:32
88	-7.29177	112.80673	8.17279	lubang	2016-07-17 10:45:41
90	-7.29187	112.80835	8.17279	lubang	2016-07-17 11:05:42
92	-7.29151	112.80488	8.17279	lubang	2016-07-17 11:19:47

Tabel 5.18 Data dari Rangkaian Bagian Belakang

Id_front	Latitude_front	Longitude_front	Altitude_front	Status_front	Waktu_front
1	-7.77000	112.32100	20.22300	lubang	2016-06-11 18:16:47
37	-7.29654	112.78593	22.22216	lubang	2016-06-11 19:30:51
41	-7.29001	112.78582	22.22216	lubang	2016-06-11 19:47:04
55	-7.29177	112.80673	8.17279	lubang	2016-07-17 10:45:51
58	-7.29187	112.80835	8.17279	lubang	2016-07-17 11:06:09
60	-7.29151	112.80488	8.17279	lubang	2016-07-17 11:20:01

Dari selisih waktu yang diterima dari rangkaian depan dan bagian belakang dapat dilihat *delay* waktu pengiriman dari alat ke *server*. Delay waktu pengiriman ditentukan berdasarkan selisih yang diterima yaitu lebih dari 11 detik.

5.5 Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian akurasi maka pada subbab ini akan menjelaskan evaluasi pengujian seperti berikut.

5.5.1 Evaluasi Pengujian Fungsionalitas

Hasil pengujian fungsionalitas dapat dilihat pada subbab sebelumnya. Berdasarkan Gambar 5.1 dan Gambar 5.2 dapat dilihat hasil uji coba fungsi pada aplikasi dapat dijalankan dengan baik. Sehingga dapat ditarik kesimpulan bahwa fungsionalitas aplikasi telah bekerja sesuai dengan harapan dan menampilkan proses serta hasil yang benar.

Tabel 5.19 Rangkuman Hasil Pengujian Fungsional

Nama	Hasil
Pengujian menerima data dari rangkaian <i>mobile</i>	Berhasil
Pengujian melihat peta di <i>website</i>	Berhasil

5.5.2 Evaluasi Pengujian Performa

Berdasarkan hasil uji coba pada Tabel 5.16 dapat dilihat bahwa hasil pengujian lubang memiliki presisi 83,33%. hasil yang baik dapat ditemui pada pengujian jalan normal yang memiliki *recall* 99,47%. Berdasarkan hasil uji coba melewati polisi tidur dapat dilihat hasil keputusan yang dihasilkan saat menentukan kondisi jalan polisi tidur terdeteksi sebagai lubang oleh sistem dalam menentukan keputusan.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

const int chipSelect = 4;

Adafruit_ADXL345_Unified accel =
  Adafruit_ADXL345_Unified(12345);
int cek=1;
int awal=1;
String kondisi="diam";
String flag="diam";
float y1 = 0;
float y2 = 0;
float ytemporary = 0;
float ydifference = 0;
```

Gambar 0.1 Inisialisasi Variabel Global

```
double Datatransfer(char *data_buf, char num){
  double temp=0.0;
  unsigned char i, j;
  if(data_buf[0]!='-'){
    i=1;

    while(data_buf[i]!='.')
      temp=temp*10+(data_buf[i++]-0x30);
    for(j=0; j<num; j++)
      temp=temp*10+(data_buf[++i]-0x30);
    for(j=0; j<num; j++)
      temp=temp/10;

    temp=0-temp;
  }
  else{
    i=0;
    while(data_buf[i]!='.')

```

```

        temp=temp*10+(data_buf[i++]-0x30);
    for(j=0;j<num;j++)
        temp=temp*10+(data_buf[++i]-0x30);
    for(j=0;j<num;j++)
        temp=temp/10 ;
    }
    return temp;
}

double decimalgps(double rawdata){
    int degrees = (int)(rawdata / 100);
    double minutes = rawdata - (degrees*100);
    double mindecimal = minutes / 60.0;
    double total = degrees + mindecimal;

    return total;
}

void result(){

    double newlat = Datatransfer(latitude,5);
    double newlon = Datatransfer(longitude,5);
    double newalt = Datatransfer(altitude,5);
    corlat = decimalgps(newlat);
    corlon = decimalgps(newlon);
    coralt = decimalgps(newalt);
}

```

Gambar 0.2 Fungsi-fungsi untuk Mengonversi Nilai Raw GPS Menjadi Float

```

void start_GSM(){
    //Configuracion GPRS Claro Argentina
    Serial.println("AT");
    delay(2000);
    Serial.println("AT+CREG?");
    delay(2000);
    Serial.println("AT+SAPBR=3,1,\"APN\", \"3gprs\"");
    delay(2000);
    Serial.println("AT+SAPBR=3,1,\"USER\", \"3gprs\"");
    delay(2000);
    Serial.println("AT+SAPBR=3,1,\"PWD\", \"3gprs\"");
    delay(2000);
}

```



```

Serial.println("AT+SAPBR=3,1,\"Contype\", \"GPRS\");
delay(2000);
Serial.println("AT+SAPBR=1,1");
delay(10000);
Serial.println("AT+HTTPIPINIT");
delay(2000);
Serial.println("AT+HTTTPARA=\"CID\",1");
delay(2000);
}

void start_GPS(){
  //Configuracion en Inicializacion GPS
  Serial.print("AT");
  delay(1000);
  Serial.println("AT+CGPSIPR=9600");// (set the baud
rate)
  delay(1000);
  Serial.println("AT+CGPSPWR=1"); // (turn on GPS
power supply)
  delay(1000);
  Serial.println("AT+CGPSRST=1"); // (reset GPS in
autonomy mode)
  delay(1000);
  Serial.println("AT+CGPSSTATUS?"); //cek status GPS
  delay(10000); //delay para esperar señal del GPS
}

```

Gambar 0.3 Fungsi untuk Menginisialisasi GPS dan GSM

```

void setup(void)
{

  pinMode(3,OUTPUT);//The default digital driver pins
for the GSM and GPS mode
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  digitalWrite(5,HIGH);
  delay(1500);
  digitalWrite(5,LOW);
}

```

```

Serial.begin(9600);

digitalWrite(3,LOW);//enable GSM TX, RX
digitalWrite(4,HIGH);//disable GPS TX, RX

    delay(3000);

    start_GSM();

    delay(5000);

    start_GPS();

    delay(5000);//GPS ready

Serial.println("AT");
delay(2000);
Serial.println("AT+CGSPWR=1");
delay(1000);
Serial.println("AT+CGPSRST=1");
delay(1000);

digitalWrite(4,LOW);//Enable GPS mode
digitalWrite(3,HIGH);//Disable GSM mode
delay(2000);
Serial.println("Oops, no ADXL345 detected ... Check
your wiring!");
    while(1);
}
/* Set the range to whatever is appropriate for your
project */
// accel.setRange(ADXL345_RANGE_16_G);
// accel.setRange(ADXL345_RANGE_8_G);
// accel.setRange(ADXL345_RANGE_4_G);
// accel.setRange(ADXL345_RANGE_2_G);
}
}

```

Gambar 0.4 Fungsi Setup untuk Mempersiapkan Sensor dan GPS/GPRS/GSM Shield V3.0

```

void loop(){
    read_GPS();
}

```

```

if(delaygps==1){
    delay(2000);
}
result();
Serial.println("tes2");//fungsi akurasi data mentah
GPS
Serial.println(corlat, 5);
Serial.println(corlon, 5);
Serial.println(coralt, 5);

if(delaygps==1){
    delay(2000);
}
sensors_event_t event;
accel.getEvent(&event);
y=event.acceleration.y*10+36;
Serial.print(" ");Serial.println("m/s^2 ");
if(corlat< -7 && corlon > 111 && cek==1)
{
    kondisi="lubang";
    send_GPRS();
    cek=0;
    kondisi="lubang";
    delaygps=0;
}
if (awal==1){
    y1=0;
    y2=y;
    ytemporary=y;
    awal=0;
}
else{
    y1=ytemporary;
    ytemporary=y;
    y2=y;
    ydifference=abs(y2-y1);
    if(flag!="lubang" && corlat< -7 && corlon >112){
        kondisi="lubang";
        flag="lubang";
        if(ydifference >= 163.97
        && ydifference <= 452.68){
Serial.print("AT+HTTTPARA=\"URL\", \"fauzan.yusufnugroh
o.com/GetData?latitude=");

```

```

        Serial.print(corlat, 5);
        Serial.print("&longitude=");
        Serial.print(corlon, 5);
        Serial.print("&altitude=");
        Serial.print(coralt, 5);
        Serial.print("&status=");
        Serial.print("lubang");
        Serial.print("&asal=");
        Serial.print("front");
        Serial.println("\n");
        delay(2000);
        Serial.println("AT+HTTPACTION=0");
        delay(2000);
    }

}

else{
    kondisi="normal";
    flag="normal";
    Serial.print("masuk normal");
}
}
}
delay(250);
}

```

Gambar 0.5 Fungsi untuk Penentuan Keputusan

```

if(!empty($_GET['latitude'])&&
!empty($_GET['longitude']) &&
!empty($_GET['altitude']) && !empty($_GET['status'])
&&!empty($_GET['asal'])) {

function getParameter($par, $default = null){
    if (isset($_GET[$par]) && strlen($_GET[$par]))
return $_GET[$par];
    elseif (isset($_POST[$par]) && strlen($_POST[$par]))
        return $_POST[$par];
        else return $default;
    }
}
}
$latitude = getParameter("latitude");

```

```
$longitude = getParameter("longitude");
$altitude = getParameter("altitude");
$status = getParameter("status");
$asal = getParameter("asal");
$file = 'data.txt';
//$detailwaktu = date("Y-m-d H:i:s");
if (!file_put_contents($file, $latitude
, FILE_APPEND | LOCK_EX))
    echo "\n\t Error saving Data\n";
else echo "\n\t Data Save\n";
```

Gambar 0.6 Fungsi untuk Menerima Data

(Halaman ini sengaja dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, maka dapat disimpulkan sebagai berikut.

1. Sensor *accelerometer* dapat merekam perubahan percepatan pengendara saat di jalan dan mendapatkan data perubahan percepatan saat pengendara melewati lubang.
2. Data dari *accelerometer* didapatkan nilai sumbu x , y dan z. Untuk GPS didapatkan nilai *latitude*, *longitude*, dan *altitude* yang dikonversi menjadi nilai *float* dari *raw* data yang didapatkan.
3. Saat melewati lubang terdapat perbedaan yang didapatkan dari nilai perubahan sensor *accelerometer*.
4. Peta dapat menampilkan lokasi lubang juga mengestimasi posisinya menggunakan GPS.
5. Algoritma penentuan lubang digunakan adalah *threshold* yang ditentukan berdasar analisis selisih perubahan nilai *accelerometer* dari data yang telah didapatkan lebih dulu. Sistem memiliki presisi 83,33%.

6.2 Saran

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

1. Pada penelitian selanjutnya dapat dilakukan pada kendaraan mobil dan pada *range* kecepatan yang lebih luas dan ditambahkan sensor yang merekam kecepatan kendaraan, untuk menambah variasi *threshold*.
2. Sensor *accelerometer* yang digunakan memiliki sensitivitas yang sama.
3. Rangkaian sistem dihubungkan dengan *accumulator* sepeda motor atau mobil untuk menghemat biaya yang digunakan untuk membeli baterai dan *power bank*. Selain itu dengan dihubungkannya sistem dengan *accumulator* maka memungkinkan untuk mengirim lokasi jalan normal ke *server*, sehingga dapat dilakukan pembaharuan peta terhadap jalan yang telah diperbaiki.
4. Sensor yang digunakan tidak hanya *accelerometer* sehingga dapat meningkatkan akurasi bahkan menentukan berbagai tipe kondisi jalan.
5. Untuk pengumpulan tidak menggunakan SD *card* atau *Micro SD*. Kemampuan *write* SD *card* tidak cukup cepat, jika delay kurang dari 250 ms akan banyak data yang tidak tersimpan atau hilang.

DAFTAR PUSTAKA

- [1] [Online]. Available:
<http://arduino.cc/en/Main/arduinoBoardUno>. [Diakses 5 6 2016].
- [2] [Online]. Available:
<http://www.dfrobot.com/wiki/images/d/d2/TEL0051ppinout.png>. [Diakses 5 6 2016].
- [3] [Online]. Available:
[http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_\(SKU:TEL0051\)](http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)). [Accessed 5 6 2016].
- [4] [Online]. Available:
http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908_AT. [Diakses 5 6 2016].
- [5] [Online]. Available:
<http://www.analog.com/en/products/mems/mems-accelerometers/adxl345.html>. [Diakses 5 6 2016].
- [6] [Online]. Available:
<https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>. [Diakses 5 6 2016].
- [7] [Online]. Available:
<https://www.arduino.cc/en/Guide/Windows>. [Diakses 5 6 2016].
- [8] [Online]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/adxl345-digital-accelerometer.pdf>. [Diakses 9 6 2016].
- [9] [Online]. Available:
<https://www.arduino.cc/en/Serial/Begin>. [Diakses 9 6 2016].
- [10] Jakob Eriksson and Lewis Girod and Bret Hull and Ryan Newton and Samuel Madden and Hari Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," *The Sixth Annual International*

conference on Mobile Systems, Applications and Services (MobiSys 2008), pp. 1-11, 2008.

BIODATA PENULIS



Ahmad Fauzan Mufid, lahir pada tanggal 14 Februari 1994 di Martapura. Penulis adalah anak ke-1 dari 2 bersaudara. Penulis menempuh pendidikan di SDN Banjarbaru Utara 1 (2000-2006), MTsN Model Darrussalam Martapura (2006-2009), SMAN 1 Banjarbaru (2009-2012). Pada tahun 2012, penulis menempuh pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya. Di jurusan

Teknik Informatika, penulis mengambil rumpun mata kuliah Komputasi Berbasis Jaringan. Selama menempuh kuliah, penulis aktif terlibat dalam organisasi mahasiswa di antaranya, sebagai staff Riset dan Teknologi Himpunan Mahasiswa Teknik Computer-Informatika ITS periode 2013/2014, sebagai staff *Information Media* Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi ITS periode 2013/2014, sebagai staff ahli Riset dan Teknologi Himpunan Mahasiswa Teknik Computer-Informatika ITS periode 2014/2015, sebagai staff ahli Syria Keluarga Muslim Informatika ITS. Juga turut bergabung menjadi anggota UKM Tennis 2012, 2013 dan 2014. Pada beberapa acara kampus, penulis juga beberapa kali aktif menjadi panitia. Penulis dapat dihubungi melalui alamat email [fauzan14mufid\[at\]gmail.com](mailto:fauzan14mufid[at]gmail.com).