



TUGAS AKHIR - KI141502

**PENGEMBANGAN SISTEM CERDAS
PENGATURAN RUTE KENDARAAN DARURAT
MENGGUNAKAN METODE FUZZY
BERDASARKAN JARAK DAN URUTAN
PRIORITAS**

**MUHAMMAD FAIEZ ANANDA
NRP 5113100055**

Dosen Pembimbing I
Henning Titi Ciptaningtyas, S.Kom., M.Kom.

Dosen Pembimbing II
Ir. Muhammad Husni, M.Kom.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**PENGEMBANGAN SISTEM CERDAS
PENGATURAN RUTE KENDARAAN DARURAT
MENGGUNAKAN METODE FUZZY
BERDASARKAN JARAK DAN URUTAN
PRIORITAS**

**MUHAMMAD FAIEZ ANANDA
NRP 5113100055**

**Dosen Pembimbing I
Henning Titi Ciptaningtyas, S.Kom., M.Kom.**

**Dosen Pembimbing II
Ir. Muhammad Husni, M.Kom.**

**Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

DEVELOPMENT OF INTELLIGENT EMERGENCY VEHICLE ROUTE CONTROL SYSTEM USING FUZZY METHOD BASED ON DISTANCE AND PRIORITY LEVEL

MUHAMMAD FAIEZ ANANDA

NRP 5113100055

First Advisor

Henning Titi Ciptaningtyas, S.Kom., M.Kom.

Second Advisor

Ir. Muchammad Husni, M.Kom.

Department of Informatics

Faculty of Information Technology and Communication

Sepuluh Nopember Institute of Technology

Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM CERDAS PENGATURAN RUTE KENDARAAN DARURAT MENGGUNAKAN METODE FUZZY BERDASARKAN JARAK DAN URUTAN PRIORITAS

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Arsitektur dan Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

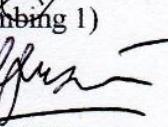
Oleh:

MUHAMMAD FAIEZ ANANDA

NRP: 5113100055

Disetujui oleh Pembimbing Tugas Akhir

1. Henning Titi C., S.Kom. M.Kom
(NIP. 198407082010122001) (Pembimbing 1)

2. Ir. Muhammad Husni, M.Kom
(NIP. 196002211984031001) (Pembimbing 2)


**SURABAYA
JANUARI, 2018**

[Halaman ini sengaja dikosongkan]

PENGEMBANGAN SISTEM CERDAS PENGATURAN RUTE KENDARAAN DARURAT MENGGUNAKAN METODE FUZZY BERDASARKAN JARAK DAN URUTAN PRIORITAS

**Nama Mahasiswa : MUHAMMAD FAIEZ ANANDA
NRP : 5113100055
Jurusan : Informatika FTIK-ITS
Dosen Pembimbing 1 : Henning Titi C., S.Kom., M.Kom.
Dosen Pembimbing 2 : Ir. Muchammad Husni, M.Kom.**

Abstrak

Kendaraan darurat seperti mobil pemadam kebakaran dan mobil ambulans adalah kendaraan yang membutuhkan rute dan waktu tercepat saat beroperasi. Untuk mempercepat sampainya kendaraan darurat ke lokasi kejadian, kendaraan darurat sudah dilengkapi oleh lampu sirine dan suara sirine. Namun, karena perilaku pemakai lalu lintas yang lain dan tingginya tingkat kemacetan di kota besar, kendaraan darurat sering terlambat tiba di tempat kejadian dan mengakibatkan jatuhnya korban jiwa dan kerugian harta yang tidak sedikit. Untuk mengatasi masalah tersebut, perlu ada Intelligent Transportation System (ITS) untuk mengatur rute kendaraan darurat agar dapat berkerja lebih cepat.

Pengembangan ITS pada Tugas Akhir ini akan menerapkan metode fuzzy dalam bentuk fuzzy control system untuk menentukan prioritas kendaraan darurat. Nilai prioritas yang dihasilkan dari fuzzy control system digunakan sebagai parameter utama dalam mengatur rute dan merekayasa lampu lalu lintas di persimpangan terutama jika ada collision antar kendaraan darurat yang terjadi. Google Maps Directions API akan digunakan untuk mencari rute tercepat untuk kendaraan darurat.

Dari hasil uji coba yang telah dilakukan, didapatkan bahwa penerapan metode fuzzy dapat membantu mengatasi

masalah collision antar kendaraan darurat yang terjadi di persimpangan dengan penentuan prioritas sebelum melakukan rekayasa lampu lalu lintas dengan kecepatan waktu eksekusi penentuan prioritas rata-rata 13,385 ms dan kecepatan waktu eksekusi pengaturan rute rata-rata 215,638 ms. Penggunaan Google Maps Directions API untuk mencari rute dan penerapan formula Haversine untuk mencari rumah sakit atau pusat pemadam kebakaran terdekat dapat membantu kinerja kendaraan darurat dalam menangani berbagai insiden dengan kecepatan waktu eksekusi rata-rata 19,24 ms.

Kata kunci: *Intelligent Transportation System (ITS), fuzzy control system, kendaraan darurat, prioritas, rekayasa lalu lintas, Google Maps API.*

DEVELOPMENT OF INTELLIGENT EMERGENCY VEHICLE ROUTE CONTROL SYSTEM USING FUZZY METHOD BASED ON DISTANCE AND PRIORITY LEVEL

Student's Name	: MUHAMMAD FAIEZ ANANDA
Student's ID	: 5113100055
Department	: Informatics FTIK-ITS
First Advisor	: Henning Titi C., S.Kom., M.Kom.
Second Advisor	: Ir. Muhammad Husni, M.Kom.

Abstract

Emergency vehicles such as firefighters and ambulances are types of vehicles that need fastest route and time while working. To speed up the arrival of emergency vehicles to location of the incident, emergency vehicles are already equipped by sirens. However, due to the behaviour of other traffic users and the high level of traffic congestion in large cities, emergency vehicles are often late when arriving to location of the incident and resulting in casualties and loss of property. To overcome this problems, an Intelligent Transportation System (ITS) is needed to set the route of emergency vehicles in order to work faster.

Development of ITS in this undergraduate thesis will implements the fuzzy method in form of fuzzy control system to determine the priority of emergency vehicle. The priority value generated by fuzzy control system will be used as the main parameter when controlling the routes and traffic lights at the intersection, especially if there is a collision occurred between emergency vehicles. Google Maps Directions API will be used to find the fastest route for emergency vehicles.

From the experiment results, the implementation of fuzzy method can help overcome the problem of collision between emergency vehicles occurred at the intersection with priority determination before controlling the traffic lights with an average

priority determination execution time of 13.385 ms and average execution time of 215.638 ms for route control. The use of Google Maps Directions API to find the fastest route and the implementation of Haversine formula to find the nearest hospital or fire station can help the performance of emergency vehicle in handling incidents with an average execution time of 19.24 ms.

Keywords: ***Intelligent Transportation System (ITS), fuzzy control system, emergency vehicle, priority, traffic control , Google Maps API.***

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "**PENGEMBANGAN SISTEM CERDAS PENGATURAN RUTE KENDARAAN DARURAT MENGGUNAKAN METODE FUZZY BERDASARKAN JARAK DAN URUTAN PRIORITAS**". Tugas Akhir ini merupakan salah satu syarat dalam menempuh ujian sidang guna memperoleh gelar Sarjana Komputer. Dengan penggerjaan Tugas Akhir ini, penulis bisa belajar banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Selesainya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan dari beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Orang tua penulis Bapak Amril Hasan dan Ibu Yustina serta keluarga penulis yang selalu memberikan dukungan doa, moral, dan material yang tak terhingga kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Ibu Henning Titi Ciptaningtyas, S.Kom., M.Kom. dan Bapak Ir. Muchammad Husni, M.Kom selaku dosen pembimbing penulis yang telah membimbing dan memberikan motivasi, dan nasehat dalam menyelesaikan Tugas Akhir ini.
4. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS.
5. Bapak Dr. Radityo Anggoro, S.Kom., M.Sc. selaku koordinator Tugas Akhir.

6. Seluruh dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di ITS.
7. Teman-teman angkatan 2013, yang sudah mendukung penulis selama perkuliahan.
8. Sahabat penulis yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu dan berjuang bersama-sama penulis.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan sehingga dengan kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Januari 2018

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak.....	vii
<i>Abstract</i>	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan.....	3
1.4 Tujuan	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.6.1 Penyusunan Proposal Tugas Akhir	4
1.6.2 Studi Literatur	4
1.6.3 Implementasi Perangkat Lunak.....	4
1.6.4 Pengujian dan Evaluasi.....	5
1.6.5 Penyusunan Buku	5
1.7 Sistematika Penulisan Laporan	5
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Intelligent Transportation System</i>	7
2.2 <i>Fuzzy Control System</i>	7
2.3 Formula Haversine	8
2.4 PHP	9
2.5 MySQL.....	10
2.6 JSON	10
2.7 Google Maps API.....	11
2.8 GPS	12
2.9 Python	12
2.10 NumPy.....	13
2.11 SciPy	14
2.12 Skfuzzy.....	14

2.13	Android	15
2.14	Klasifikasi Bahaya Kebakaran	16
2.15	Klasifikasi Ambulans	18
2.16	Tugas Pokok Pemadam Kebakaran	18
BAB III PERANCANGAN PERANGKAT LUNAK.....	19	
3.1	Data	19
3.1.1	Data Masukan	19
3.1.2	Data Keluaran	20
3.2	Dekripsi Umum Sistem	20
3.3	<i>Fuzzy Control System</i> untuk Penentuan Prioritas	24
3.3.1	<i>Membership Function</i> Tingkat Bahaya Insiden	26
3.3.2	<i>Membership Function</i> Jarak	27
3.3.3	<i>Membership Function</i> Jumlah Korban	28
3.3.4	<i>Membership Function</i> Prioritas	29
3.3.5	<i>Fuzzy Rule</i>	29
3.4	Perancangan Antarmuka	30
BAB IV IMPLEMENTASI.....	33	
4.1	Lingkungan Implementasi	33
4.2	Implementasi	34
4.2.1	<i>Database</i>	34
4.2.2	Proses <i>Login</i>	39
4.2.3	Pelaporan Insiden	40
4.2.4	Pencarian Rute	43
4.2.4.1	<i>Kelas Route</i>	43
4.2.4.2	Proses Pencarian Rute	44
4.2.5	Penghitungan Prioritas	45
4.2.6	Pengaturan Rute	46
BAB V HASIL UJI COBA DAN EVALUASI	47	
5.1	Lingkungan Pengujian	47
5.2	Data Pengujian	48
5.3	Skenario Uji Coba	49
5.3.1	Skenario Uji Coba Performa Pencarian Rute	49
5.3.1.1	Skenario Uji Coba Pencarian Rute 1	51
5.3.1.2	Skenario Uji Coba Pencarian Rute 2	53

5.3.2 Skenario Uji Coba Performa Penghitungan Prioritas	
5.3.2.1 Skenario Uji Coba Penghitungan Prioritas 1	56
5.3.2.2 Skenario Uji Coba Penghitungan Prioritas 2	58
5.3.3 Skenario Uji Coba Performa Pengaturan Rute	60
5.3.3.1 Skenario Uji Coba Pengaturan Rute 1	63
5.3.3.2 Skenario Uji Coba Pengaturan Rute 2	64
5.3.3.3 Skenario Uji Coba Pengaturan Rute 3	66
5.3.3.4 Skenario Uji Coba Pengaturan Rute 4	69
5.4 Evaluasi Umum Skenario Uji Coba	73
BAB VI KESIMPULAN DAN SARAN	75
6.1 Kesimpulan	75
6.2 Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN	Error! Bookmark not defined.
BIODATA PENULIS	95

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Diagram Blok <i>Fuzzy Control System</i>	8
Gambar 2.2 Logo PHP[7].....	9
Gambar 2.3 Logo MySQL[8].....	11
Gambar 2.4 Logo JSON[9]	11
Gambar 2.5 Logo Google Maps[10]	13
Gambar 2.6 Logo Python[12].....	13
Gambar 2.7 Logo NumPy[13].....	13
Gambar 2.8 Logo SciPy[13].....	14
Gambar 2.9 Logo skfuzzy[14].....	15
Gambar 2.10 Logo Android[15].....	16
Gambar 3.1 Diagram Alir Proses Pencarian Rute	22
Gambar 3.2 Diagram Alir Proses Pengaturan Rute	23
Gambar 3.3 Arsitektur Sistem	24
Gambar 3.4 <i>Membership Function</i> Tingkat Bahaya Insiden.....	26
Gambar 3.5 <i>Membership Function</i> Jarak	27
Gambar 3.6 <i>Membership Function</i> Jumlah Korban	28
Gambar 3.7 <i>Membership Function</i> Prioritas	29
Gambar 3.8 Desain Tampilan Antarmuka <i>Login</i>	30
Gambar 3.9 Desain Tampilan Antarmuka Lapor Insiden	31
Gambar 3.10 Desain Tampilan Antarmuka Tampilkan Rute	32
Gambar 5.1 Contoh Hasil Pencarian Rute pada Aplikasi.....	50
Gambar 5.2 Lokasi Persimpangan.....	61
Gambar 5.3 Respons Server Sukses	62
Gambar 5.4 Respons Server Gagal 1.....	62
Gambar 5.5 Respons Server Gagal 2.....	62

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Tabel Klasifikasi Bahaya Kebakaran	16
Tabel 3.1 Tabel Bobot Jenis Insiden untuk Ambulans	25
Tabel 3.2 Tabel Bobot Jenis Insiden untuk Pemadam Kebakaran	25
Tabel 3.3 Tabel Bobot Isu Medis	25
Tabel 3.4 Tabel Bobot Bahaya Kebakaran	25
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	33
Tabel 4.2 Tabel-tabel pada <i>Database</i>	35
Tabel 4.3 Atribut Tabel am_incident_data	35
Tabel 4.4 Atribut Tabel pb_incident_data	36
Tabel 4.5 Atribut Tabel emergency_services	37
Tabel 4.6 Atribut Tabel vehicles	37
Tabel 4.7 Atribut Tabel traffic_lights	38
Tabel 4.8 Atribut Tabel users	38
Tabel 5.1 Spesifikasi Lingkungan Pengujian	47
Tabel 5.2 Tabel Lokasi Rumah Sakit dan Pusat Pemadam Kebakaran	51
Tabel 5.3 Tabel Lokasi Insiden Pencarian Rute	52
Tabel 5.4 Hasil Uji Coba Pencarian Rute 1	53
Tabel 5.5 Hasil Uji Coba Pencarian Rute 2	54
Tabel 5.6 Tabel Kondisi Insiden untuk Ambulans	55
Tabel 5.7 Tabel Kondisi Insiden untuk Pemadam Kebakaran	56
Tabel 5.8 Hasil Uji Coba Penghitungan Prioritas 1	57
Tabel 5.9 Hasil Uji Coba Penghitungan Prioritas 2	59
Tabel 5.10 Tabel Lokasi Lampu Lalu Lintas	61
Tabel 5.11 Hasil Uji Coba Pengaturan Rute 1	63
Tabel 5.12 Tabel Kondisi Pengaturan Rute	64
Tabel 5.13 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 1	65
Tabel 5.14 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 2	65
Tabel 5.15 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 3	65

Tabel 5.16 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 4	65
Tabel 5.17 Hasil Uji Coba Pengaturan Rute 2	66
Tabel 5.18 Tabel Kondisi Pengaturan Rute 3	67
Tabel 5.19 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 1	67
Tabel 5.20 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 2	67
Tabel 5.21 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 3	68
Tabel 5.22 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 4	68
Tabel 5.23 Hasil Uji Coba Pengaturan Rute 3	68
Tabel 5.24 Tabel Kondisi Pengaturan Rute 4	69
Tabel 5.25 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 1	70
Tabel 5.26 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 2	70
Tabel 5.27 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 3	71
Tabel 5.28 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 4	71
Tabel 5.29 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 5	71
Tabel 5.30 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 6	72
Tabel 5.31 Hasil Uji Coba Pengaturan Rute 4	72

DAFTAR KODE SUMBER

Pseudocode 4.1 Kode Program <i>Login</i>	39
Pseudocode 4.2 Kode Program Penentuan Jenis Pelaporan Insiden	40
Pseudocode 4.3 Kode Program Pengambilan Lokasi Kendaraan Darurat.....	41
Pseudocode 4.4 Kode Program Pelaporan Insiden.....	42
Pseudocode 4.5 Kode Program Kelas <i>Route</i>	43
Pseudocode 4.6 Kode Program Menampilkan Rute.....	44
Pseudocode 4.7 Kode Program Penghitungan Prioritas	45
Pseudocode 7.1 Kode Program Penyimpanan Data Insiden untuk Ambulans	55
Pseudocode 7.2 Kode Program Penyimpanan Data Insiden untuk Pemadam Kebakaran.....	39
Pseudocode 7.3 Kode Program Pencarian Rute	40
Pseudocode 7.4 Kode Program Inisialisasi <i>Membership Function</i>	41
Pseudocode 7.5 Kode Program Pengaturan Rute	42

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan darurat seperti mobil pemadam kebakaran dan mobil ambulans adalah kendaraan yang membutuhkan rute terpendek dan waktu perjalanan tersingkat dari posisi awal menuju ke lokasi kejadian (dalam hal ini lokasi kebakaran atau lokasi kecelakaan). Selain rute terpendek, mencari lokasi rumah sakit terdekat juga menjadi faktor penting.

Untuk mempercepat sampainya kendaraan darurat ke lokasi kejadian dan kembalinya ambulans dari lokasi kejadian ke rumah sakit, kendaraan darurat sudah dilengkapi oleh lampu sirine dan suara sirine. Namun, karena perilaku pemakai lalu lintas yang lain dan tingginya tingkat kemacetan di kota besar, kendaraan darurat sering terlambat tiba di tempat kejadian dan mengakibatkan jatuhnya korban jiwa dan kerugian harta yang tidak sedikit.

Untuk menangani masalah tersebut, maka akan diimplementasikan sistem transportasi cerdas yang akan merekayasa lampu lalu lintas pada jalur yang akan dilalui oleh kendaraan darurat. Kendaraan darurat akan diatur lewat *server* yang nantinya bertindak sebagai unit utama yang menentukan rute terpendek dan tercepat untuk kendaraan darurat[1], mencari rumah sakit atau pusat pemadam kebakaran terdekat, dan juga merekayasa lampu lalu lintas di sepanjang jalur yang dilalui oleh kendaraan darurat.

Penelitian sebelumnya sudah ada sistem pengaturan lampu lalu lintas menjadi hijau ketika sistem mendeteksi kendaraan darurat sudah masuk dalam jarak trigger lampu lalu lintas. Sistem pengaturan lalu lintas tersebut terdiri dari mikrokontroler Arduino sebagai *trigger* untuk sistem, *smartphone* berbasis Android sebagai pengontrol, dan GPS yang terdapat pada *smartphone* berguna sebagai penentu posisi mobil darurat. Arduino *master* mengkondisikan Arduino yang lain agar menjadi kondisi darurat

atau tidak. Perangkat Android diletakkan di mobil darurat bertujuan untuk mengirim pesan darurat ketika mobil darurat masuk jarak *trigger* lampu lalu lintas yang posisi dan jarak *trigger*-nya sudah diatur sebelumnya di dalam sistem[2]. Pada tugas akhir ini, akan dikembangkan sistem pengaturan prioritas kendaraan darurat untuk merekayasa lampu lalu lintas pada rute yang dilalui kendaraan darurat dengan menggunakan *fuzzy control system*. Sehingga setiap kendaraan darurat memiliki prioritas masing-masing yang nantinya akan digunakan sebagai penentu jika terjadi *collision* di persimpangan[3].

Dalam tugas akhir ini, akan menggunakan Google Maps Directions API untuk menentukan rute tercepat yang akan dilalui oleh kendaraan darurat, yang nantinya rute tersebut akan dikirimkan ke *device* berbasis Android di kendaraan darurat. Selain itu, *fuzzy control system* akan diimplementasikan untuk membantu proses rekayasa lampu lalu lintas pada persimpangan. Sehingga dalam kasus ini, kendaraan darurat tidak banyak membuang waktu karena terjebak oleh lampu merah jika ada *collision* yang terjadi antar kendaraan darurat. *Fuzzy control system* ini akan membantu untuk mengatur rekayasa lampu lalu lintas yang tepat berdasarkan prioritas kendaraan darurat yang mengalami *collision*. Hal ini tentunya akan mempercepat waktu tempuh kendaraan darurat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mekanisme pencarian rute tercepat yang akan dilalui oleh kendaraan darurat?
2. Bagaimana pengaturan prioritas kendaraan darurat jika terjadi *collision* antar kendaraan darurat di sebuah persimpangan?
3. Bagaimana manajemen rekayasa lampu lalu lintas yang tepat pada persimpangan yang akan dilalui kendaraan darurat?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan sebagai berikut:

1. Mengatur prioritas kendaraan darurat saat ada *collision* pada sebuah persimpangan berdasarkan urgensitas (misal ambulans yang membawa pasien kritis memiliki prioritas lebih tinggi daripada ambulans yang membawa jenazah).
2. Jumlah *collision* yang akan ditangani pada sebuah persimpangan maksimal sebanyak empat kendaraan.
3. Jenis persimpangan yang digunakan adalah persimpangan berbentuk empat lengan.
4. Lokasi rumah sakit dan pusat pemadam kebakaran telah disediakan terlebih dahulu.
5. Lokasi awal *user* aplikasi kendaraan darurat diasumsikan berada di lokasi rumah sakit atau pusat pemadam kebakaran terdekat.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Mencari rute tercepat untuk kendaraan darurat ke lokasi tujuan.
2. Melakukan pengaturan prioritas kendaraan darurat saat ada *collision* antar kendaraan darurat di sebuah persimpangan.
3. Melakukan manajemen rekayasa lampu lalu lintas yang tepat pada persimpangan yang akan dilalui kendaraan darurat.

1.5 Manfaat

Manfaat dari pembuatan Tugas Akhir ini antara lain:

1. Mempersingkat waktu tempuh kendaraan darurat saat menuju lokasi kejadian.
2. Mengurangi timbulnya korban jiwa maupun kerugian harta akibat terlambatnya kendaraan darurat.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam penggerjaan Tugas Akhir ini adalah sebagai berikut.

1.6.1 Penyusunan Proposal Tugas Akhir

Tahap awal Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Proposal berisi tentang deskripsi pendahuluan dari Tugas Akhir yang akan dibuat. Prosposal juga berisi tentang garis besar penggerjaan Tugas Akhir sehingga memberikan gambaran untuk dapat mengerjakan Tugas Akhir sesuai dengan *timeline* yang dibuat.

1.6.2 Studi Literatur

Pada tahap ini dilakukan untuk mencari informasi dan studi literatur apa saja yang dapat dijadikan sebagai referensi untuk membantu penggerjaan Tugas Akhir ini. Tahap ini merupakan tahap untuk memahami semua metode yang akan dikerjakan, sehingga memberi gambaran selama penggerjaan Tugas Akhir. Informasi didapatkan dari buku dan literatur yang berhubungan dengan metode yang digunakan. Informasi yang dicari adalah *fuzzy control system*, formula Haversine untuk pencarian lokasi terdekat, dan juga fitur-fitur yang disediakan oleh Google Maps API. Tugas akhir ini juga mengacu pada literatur jurnal dengan judul “*Automatic Ambulance Rescue System*” yang diterbitkan pada tahun 2012[1].

1.6.3 Implementasi Perangkat Lunak

Implementasi merupakan tahap untuk mengimplementasikan metode-metode yang sudah diajukan pada proposal Tugas Akhir. Untuk membangun sistem yang telah dirancang sebelumnya, implementasi dilakukan dengan menggunakan beberapa bahasa pemrograman yaitu PHP versi

5.6.3 dan MySQL versi 5.6.21 sebagai bahasa pemrograman bagian *server* dan *database*, Python versi 2.7 sebagai bahasa pemrograman untuk *fuzzy controller* dengan library skfuzzy 0.2, Java sebagai bahasa pemrograman untuk aplikasi Android. Perangkat lunak yang digunakan adalah PyCharm Community Edition 2017.1 dan Android Studio 3.0.1 sebagai IDE.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini sistem yang telah dikembangkan akan diuji coba dengan menggunakan skenario uji coba yang telah dibuat. Uji coba dilakukan dengan menjalankan sistem dengan tujuan mengetahui kemampuan metode yang digunakan dan mengevaluasi hasil Tugas Akhir dengan jurnal pendukung yang digunakan. Hasil evaluasi mencakup performa dari metode yang telah diimplementasikan dalam sistem pengaturan rute kendaraan darurat.

1.6.5 Penyusunan Buku

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir yang mencangkup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

- 1. Bab I. Pendahuluan**

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.

- 2. Bab II. Tinjauan Pustaka**

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis

besar, bab ini berisi tentang *Intelligent Transportation System* (ITS) dan *fuzzy control system*.

3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai perancangan *server* dari sistem pengaturan rute kendaraan darurat yang menggunakan *fuzzy control system* sebagai metode penentuan prioritas dan juga antarmuka pada aplikasi kendaraan darurat berbasis Android.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk *Pseudocode* yang berupa *Pseudocode* dari metode *fuzzy control system* dan formula Haversine yang digunakan pada *server*, implementasi *database*, dan implementasi aplikasi kendaraan darurat berbasis Android.

5. Bab V. Pengujian dan Evaluasi

Bab ini berisikan hasil uji coba dari sistem pengaturan rute kendaraan darurat yang sudah dikembangkan. Uji coba dilakukan dengan menggunakan skenario uji coba yang telah dibuat. Hasil evaluasi mencakup performa sistem dalam pencarian rute, mengatasi *collision*, dan merekayasa lalu lintas.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

8. Lampiran

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode program secara keseluruhan.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam Tugas Akhir ini. Teori-teori tersebut adalah *fuzzy control system*, formula Haversine, dan beberapa teori lain yang mendukung pembuatan Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 *Intelligent Transportation System*

Intelligent Transportation System (ITS) adalah sebuah aplikasi tingkat lanjut yang bertujuan untuk menyediakan layanan inovatif yang berkaitan dengan berbagai mode transportasi dan manajemen lalu lintas. Penggunaan ITS memungkinkan berbagai pengguna untuk mendapatkan informasi yang lebih baik dan membuat jaringan transportasi menjadi lebih aman, terkoordinasi, dan ‘lebih cerdas’[4].

Pembuatan ITS untuk kendaraan darurat merupakan tujuan utama Tugas Akhir ini. Dengan adanya ITS yang baik untuk kendaraan darurat, diharapkan kinerja dari kendaraan darurat dalam menangani berbagai insiden menjadi lebih cepat.

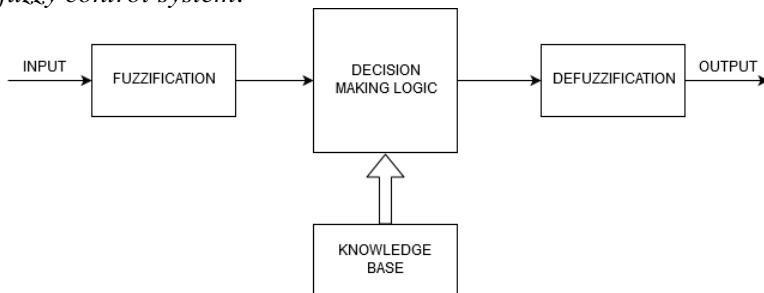
2.2 *Fuzzy Control System*

Fuzzy control system adalah sebuah metode pengaturan yang mengaplikasikan logika *fuzzy* untuk menirukan pikiran manusia. *Fuzzy control system* menggambarkan sebuah protokol dengan menggunakan aturan IF-THEN. Keambiguan dalam definisi istilah linguistik direpresentasikan dengan menggunakan himpunan *fuzzy*, yang batasnya saling tumpang tindih. Dalam himpunan *fuzzy*, elemen *domain* tertentu dapat secara bersamaan

menjadi milik beberapa himpunan (dengan derajat keanggotaan yang berbeda)[5].

Penggunaan *fuzzy control system* sebagai *decision support system* memiliki berbagai kelebihan. *Fuzzy control system* dapat menangani masukan yang bersifat ambigu dan mengolahnya berdasarkan *knowledge base* yang tersedia dengan logika *fuzzy*. Daripada menggunakan persamaan matematis yang kompleks, logika *fuzzy* menggunakan deskripsi linguistik untuk menentukan hubungan antara informasi masukan dan aksi keluaran.

Pada Tugas Akhir ini, *fuzzy control system* digunakan dalam menentukan prioritas kendaraan darurat. Keambiguan dalam menentukan kendaraan mana yang memiliki prioritas lebih tinggi membuat penggunaan *fuzzy control system* sangat berguna untuk menanggulangi masalah jika ada *collision* antar kendaraan darurat di sebuah persimpangan jalan. Gambar 2.1 merupakan diagram *fuzzy control system*.



Gambar 2.1 Diagram Blok Fuzzy Control System

2.3 Formula Haversine

Formula Haversine biasa digunakan untuk menghitung jarak antara dua titik di permukaan bumi menggunakan garis lintang (*latitude*) dan garis bujur (*longitude*) sebagai variabel masukan. Formula Haversine adalah persamaan penting pada bidang navigasi, karena formula ini dapat memberikan jarak antara dua titik pada permukaan bola (bumi) berdasarkan *latitude* dan *longitude*[6].

Pada Tugas Akhir ini, formula Haversine digunakan untuk pencarian rumah sakit atau pusat pemadam kebakaran yang terdekat dari lokasi kejadian. Persamaan 2.1 merupakan formula Haversine dimana φ adalah *latitude*, λ adalah *longitude*, R adalah jari-jari bumi (6.371 km), dan d adalah jarak antara dua titik.

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \text{atan}2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c \quad (2.1)$$

2.4 PHP

PHP (akronim rekursif dari *PHP: Hypertext Preprocessor*) adalah bahasa *scripting open-source* yang biasanya banyak digunakan untuk pengembangan web dan dapat ditanamkan ke dalam kode HTML. PHP sangat mudah dipahami bagi pemula, tetapi PHP juga menyediakan berbagai macam fitur tingkat lanjut yang berguna bagi *programmer* profesional[7].

Pada Tugas Akhir ini, pemrograman menggunakan PHP lebih terfokus pada *server-side scripting*. Dimana *script-script* PHP di *server* digunakan sebagai penghubung antara aplikasi Android kendaraan darurat dengan *database* dan *script* Python untuk *fuzzy control system*. Versi PHP yang digunakan pada Tugas Akhir ini adalah versi 5.6.3. Gambar 2.2 merupakan logo PHP.



Gambar 2.2 Logo PHP[7]

2.5 MySQL

MySQL merupakan *relational database management system* (RDBMS) *open-source* yang berbasis *structured query language* (SQL). MySQL dapat berjalan di berbagai platform, termasuk Linux, UNIX, dan Windows. MySQL biasanya digunakan dalam pengembangan aplikasi berbasis web, dan digunakan oleh perusahaan-perusahaan besar seperti Facebook, Twitter, YouTube, Yahoo!, dan lain-lain[8].

Selain bersifat *open-source*, MySQL dipilih karena dapat diintegrasikan dengan bahasa pemrograman Python yang digunakan sebagai *fuzzy control system* dalam Tugas Akhir ini. MySQL yang digunakan dalam Tugas Akhir ini adalah versi 5.6.21. Gambar 2.3 merupakan logo MySQL.



Gambar 2.3 Logo MySQL[8]

2.6 JSON

JSON merupakan singkatan dari *JavaScript Object Notation* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman Javascript. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun sehingga membuat JSON menjadi ideal sebagai bahasa pertukaran data[9].

Pada Tugas Akhir ini, JSON digunakan sebagai format pertukaran data antara aplikasi Android dengan *server*, antara Google Maps API dengan aplikasi Android untuk pencarian rute, dan juga antara *server* dengan Google Maps API untuk me-request data *geocoding*. Gambar 2.4 merupakan logo JSON.



Gambar 2.4 Logo JSON[9]

2.7 Google Maps API

Google Maps API (*Application Programming Interface*) menyediakan sejumlah utilitas untuk memanipulasi peta dan menambahkan konten di dalam peta melalui berbagai layanan. Google Maps API tersedia untuk Android, iOS, *browser web*, dan melalui layanan *web* HTTP. Google Maps API dikelompokkan menurut *platform*-nya yaitu Android, iOS, dan *web*. Selain itu, Google Maps juga menyediakan berbagai macam API yang bisa dipilih sesuai kegunaannya[10].

Pada Tugas Akhir ini, Google Maps API yang digunakan adalah Directions API untuk pencarian rute, Geocoding API untuk pencarian data alamat, Android API untuk menyediakan peta di aplikasi Android, dan Places API untuk menyediakan data tempat. Gambar 2.5 merupakan logo Google Maps.



Gambar 2.5 Logo Google Maps[10]

2.8 GPS

Global Positioning System atau yang biasa disingkat GPS adalah sistem navigasi berbasis satelit. GPS memungkinkan sebuah penerima GPS untuk mendapatkan lokasinya di permukaan bumi 24 jam per hari dengan akurasi sangat tinggi. Selain menentukan lokasi, GPS juga dapat digunakan untuk mengkalkulasi kecepatan, jarak perjalanan, dan lain-lain dari posisi 2-D (*latitude* dan *longitude*) yang telah didapatkan sebelumnya[11].

Pada Tugas Akhir ini, GPS digunakan untuk menentukan posisi kendaraan darurat yang nantinya *latitude* dan *longitude*-nya akan digunakan untuk mencari rute tercepat ke lokasi insiden, menghitung jarak antara kendaraan darurat ke lokasi insiden, dan *trigger* untuk merekayasa lampu lalu lintas.

2.9 Python

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, interaktif, *object oriented* dan dapat beroperasi di hampir semua *platform* seperti UNIX, Mac, Windows ataupun yang lain[12]. Sebagai bahasa pemrograman tingkat tinggi Python termasuk salah satu bahasa pemrograman yang mudah dipelajari karena sintaks yang jelas. Python menyediakan modul-modul yang siap dipakai dan juga Python memiliki struktur data tingkat tinggi yang efisien. Python menyediakan dua buah versi, yaitu versi 2 dan 3 yang setiap versinya memiliki kekurangan dan kelebihannya masing-masing.

Python digunakan karena dapat diintegrasikan dengan *server* yang menggunakan PHP dan database yang menggunakan MySQL. Selain itu, terdapat library-library yang banyak membantu dalam pembuatan *fuzzy control system*, seperti NumPy, SciPy, dan skfuzzy.

Pada Tugas Akhir ini, Python digunakan sebagai bahasa pemrograman utama untuk membangun *fuzzy control system* untuk penentuan prioritas kendaraan darurat. Tugas Akhir ini menggunakan Python versi 2.7. Gambar 2.6 merupakan logo Python.



Gambar 2.6 Logo Python[12]

2.10 NumPy

NumPy atau *numeric Python* merupakan pustaka perangkat lunak bahasa pemrograman Python bersifat *open-source* yang digunakan untuk pengolahan ilmiah matematika[13]. Selain digunakan untuk hal ilmiah, NumPy juga bisa digunakan untuk *container* multidimensi yang efisien untuk data generik. Tipe data *arbitrary* dapat didefinisikan, ini memungkinkan NumPy secara lancar dan cepat mengintegrasikan dengan banyak tipe basis data.

Pada Tugas Akhir ini, NumPy berperan penting untuk operasi data-data *array* yang digunakan dalam fungsi-fungsi dari *library* skfuzzy. Gambar 2.7 merupakan logo NumPy.



Gambar 2.7 Logo NumPy[13]

2.11 SciPy

SciPy merupakan singkatan dari *Scientific Python*. SciPy adalah pustaka perangkat lunak bahasa pemrograman Python yang bersifat *open-source*[13]. SciPy sering digunakan bersama dengan modul NumPy karena SciPy merupakan bagian besar dari NumPy dimana banyak fitur yang memiliki fungsi sama tetapi SciPy dianggap lebih lengkap dari pada NumPy.

Pada Tugas Akhir ini, Scipy digunakan untuk melakukan perhitungan matematis pada fungsi-fungsi yang disediakan *library* skfuzzy. Gambar 2.8 merupakan logo SciPy.



Gambar 2.8 Logo SciPy[13]

2.12 Skfuzzy

Skfuzzy (Scikit-Fuzzy) adalah *library* dari algoritma logika *fuzzy* yang dimaksudkan untuk digunakan pada SciPy Stack. Skfuzzy bersifat *open-source* dan dikembangkan oleh komunitas SciPy dengan bahasa pemrograman Python[14]. Skfuzzy menyediakan berbagai macam fungsi terkait *fuzzy* seperti *fuzzification*, *defuzzification*, menentukan *fuzzy rules*, membuat *membership function*, dan membangun *fuzzy control system*.

Pada Tugas Akhir ini, skfuzzy digunakan sebagai *library* pembantu dalam membangun *fuzzy control system* penentuan prioritas kendaraan darurat. Fungsi-fungsi yang digunakan antara lain, pembuatan *membership function*, penentuan *fuzzy rule*, dan pembuatan *control system* (*fuzzification* dan *defuzzification*). Versi skfuzzy yang digunakan adalah versi 0.2. Gambar 2.9 merupakan logo skfuzzy.



Gambar 2.9 Logo skfuzzy[14]

2.13 Android

Android merupakan sistem operasi *open-source* berbasis Linux yang dirancang perangkat bergerak layar sentuh seperti *smartphone* dan komputer tablet[15]. Dengan ketersediaannya yang *open-source*, memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dengan bahasa pemrograman Java. Faktor-faktor tersebut membuat Android menjadi sistem operasi *smartphone* paling banyak digunakan di dunia.

Pada Tugas Akhir ini, Android digunakan sebagai sistem operasi dalam pengembangan aplikasi kendaraan darurat. Versi Android yang digunakan saat pengembangan adalah versi 5.0

(Lollipop). Target *level API* minimal yang digunakan adalah level 16, karena dengan menggunakan *level API* 16 keatas maka aplikasi akan berjalan pada kurang lebih 99,2% perangkat. Aplikasi ini nantinya digunakan untuk pencarian rute tercepat ke lokasi insiden, pelaporan insiden yang terjadi, dan sebagai sumber data GPS (*latitude* dan *longitude*) dari kendaraan darurat. Gambar 2.10 merupakan logo Android.



Gambar 2.10 Logo Android[15]

2.14 Klasifikasi Bahaya Kebakaran

Menurut Keputusan Menteri Tenaga Kerja Republik Indonesia NO. KEP-186/MEN/1999 tentang Unit Penanggulangan Kebakaran di Tempat Kerja[16], klasifikasi tingkat potensi bahaya kebakaran dijelaskan pada Tabel 2.1.

Tabel 2.1 Tabel Klasifikasi Bahaya Kebakaran

Klasifikasi	Pengertian	Contoh Tempat
Bahaya Kebakaran Ringan	Merupakan kebakaran pada tempat yang mempunyai jumlah dan kemudahan terbakar rendah, dan apabila terjadi kebakaran melepaskan panas rendah, sehingga menjalarnya api lambat.	- Tempat ibadah - Gedung perkantoran - Gedung perumahan
Bahaya Kebakaran Sedang 1	Merupakan kebakaran pada tempat yang mempunyai jumlah dan kemudahan terbakar sedang, menimbun bahan dengan tinggi tidak lebih dari 2,5 meter dan apabila terjadi kebakaran	- Tempat parkir - Pabrik elektronik - Pabrik roti

Klasifikasi	Pengertian	Contoh Tempat
	melepaskan panas sedang, sehingga menjalanya api sedang.	
Bahaya Kebakaran Sedang 2	Merupakan kebakaran pada tempat yang mempunyai jumlah dan kemudahan terbakar sedang, menimbun bahan dengan tinggi lebih dari 4 meter dan apabila terjadi kebakaran melepaskan panas sedang, sehingga menjalarnya api sedang.	<ul style="list-style-type: none"> - Penggilingan padi - Percetakan dan penerbitan - Bengkel mesin - Perakitan kayu
Bahaya Kebakaran Sedang 3	Merupakan kebakaran pada tempat yang mempunyai jumlah dan kemudahan terbakar tinggi, dan apabila terjadi kebakaran melepaskan panas tinggi, sehingga menjalarnya api cepat.	<ul style="list-style-type: none"> - Pabrik pakaian - Pabrik barang plastik - Pergudangan
Bahaya Kebakaran Berat	Merupakan kebakaran pada tempat yang mempunyai jumlah dan kemudahan terbakar tinggi, menyimpan bahan cair, serta atau bahan lainnya dan apabila terjadi kebakaran apinya cepat membesar dengan melepaskan panas tinggi, sehingga menjalarnya api cepat.	<ul style="list-style-type: none"> - Pabrik kimia - Pabrik bahan peledak - Studio film dan televisi

Pada Tugas Akhir ini, klasifikasi ini akan dipadatkan menjadi tiga jenis saja, yaitu Bahaya Kebakaran Ringan, Bahaya Kebakaran Sedang (mencakup bahaya kebakaran sedang 1 dan 2),

dan Bahaya Kebakaran Berat (mencakup bahaya kebakaran sedang 3 dan berat).

2.15 Klasifikasi Ambulans

Menurut Keputusan Menteri Kesehatan dan Kesejahteraan Sosial Republik Indonesia No. 143/MENKES-KESOS/SK/II/2001 tentang Standarisasi Kendaraan Pelayanan Medik[17], klasifikasi jenis ambulans meliputi:

1. Ambulans Transportasi
2. Ambulans Gawat Darurat
3. Ambulans Rumah Sakit Lapangan
4. Ambulans Pelayanan Medik Bergerak
5. Ambulans Pengangkut Jenazah
6. Ambulans Udara

Dari beberapa jenis ambulans tersebut, yang akan digunakan dalam Tugas Akhir ini adalah Ambulans Transport untuk menangani jenis insiden transportasi pasien, Ambulans Gawat Darurat untuk menangani jenis insiden kondisi gawat darurat, dan Ambulans Pengangkut Jenazah untuk menangani jenis insiden penanganan jenazah.

2.16 Tugas Pokok Pemadam Kebakaran

Menurut Keputusan Walikota Surabaya No. 56 Tahun 2001 tentang Rincian Tugas dan Fungsi Dinas Pemadam Kebakaran pada Bagian Kelima (Sub Dinas Operasional) Pasal 13 menjelaskan bahwa Sub Dinas Operasional mempunyai tugas melaksanakan pencegahan, pemadaman dan penanggulangan kebakaran serta pertolongan akibat bencana lain[18].

Pada Tugas Akhir ini, tugas pemadam kebakaran akan digunakan sebagai kasus jenis insiden. Jenis insiden untuk pemadaman kebakaran akan dipadatkan menjadi pemadaman kebakaran serta penyelamatan jiwa.

BAB III

PERANCANGAN PERANGKAT LUNAK

Bab ini membahas mengenai perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada Tugas Akhir ini adalah pencarian rute tercepat untuk kendaraan darurat, pelaporan insiden, penentuan prioritas kendaraan darurat dengan *fuzzy control system*, dan perekayasa lampu lalu lintas.

3.1 Data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan dengan performa yang optimal.

3.1.1 Data Masukan

Data masukan adalah data yang digunakan sebagai masukan awal. Data yang digunakan dalam sistem ini terdiri dari dua jenis data masukan, yaitu data masukan untuk perangkat lunak Android dan data masukan untuk *server*. Data masukan pada perangkat lunak Android adalah data insiden yang dilaporkan oleh *user* dari aplikasi kendaraan darurat. Data insiden ini berupa data-data tentang jenis insiden yang terjadi, isu medis (untuk ambulans) atau bahaya kebakaran (untuk pemadam kebakaran), perkiraan jumlah korban, dan lokasi insiden. Data yang nantinya digunakan lebih lanjut oleh perangkat lunak Android adalah data lokasi insiden yang akan digunakan untuk mencari rute terdekat.

Data masukan untuk *server* dibagi menjadi dua jenis, yaitu data masukan untuk *fuzzy control system* dan data masukan untuk pengaturan rute kendaraan darurat. Data jenis insiden, isu medis atau bahaya kebakaran, perkiraan jumlah korban, dan lokasi insiden dari perangkat lunak Android nantinya akan dikirim ke *server*. Data-data tersebut akan digunakan sebagai data masukan

untuk *fuzzy control system* di *server* untuk menentukan prioritas dari kendaraan darurat tersebut. Untuk data masukan pengaturan rute, yang dibutuhkan adalah data lokasi kendaraan darurat (*latitude* dan *longitude*). Data lokasi ini nantinya akan digunakan untuk proses rekayasa lampu lalu lintas pada rute kendaraan darurat.

3.1.2 Data Keluaran

Data masukan untuk perangkat lunak Android yang berupa lokasi insiden nanti akan digunakan bersamaan dengan data lokasi saat ini dari perangkat Android untuk pencarian rute terdekat menggunakan Google Maps Directions API. Setelah diproses oleh Google Maps Directions API, data keluaran akan dihasilkan. Data keluaran ini berupa rute tercepat yang direkomendasikan oleh Google Maps Directions API dari lokasi perangkat Android menuju ke lokasi insiden.

Data keluaran dari *server* ada dua buah, yaitu data keluaran dari proses penghitungan prioritas kendaraan darurat oleh *fuzzy control system*, dan data berupa instruksi dari *server* untuk merekayasa lampu lalu lintas.

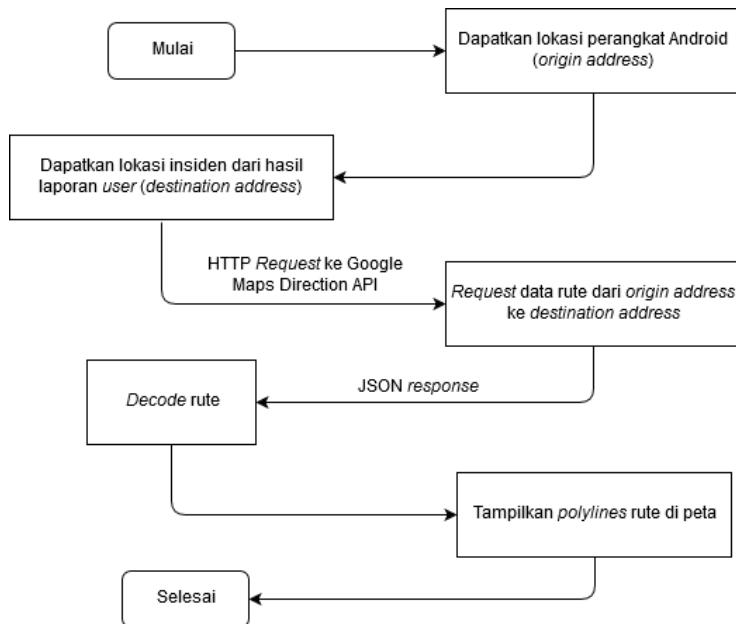
3.2 Dekripsi Umum Sistem

Rancangan sistem cerdas pengaturan rute kendaraan darurat menggunakan metode *fuzzy* dimulai dengan pelaporan insiden menggunakan aplikasi berbasis Android. Sesuai dengan jenis kendaraan dari *user*, *user* melaporkan jenis insiden, tingkat bahaya insiden (isu medis untuk jenis kendaraan ambulans dan bahaya kebakaran untuk jenis kendaraan pemadam kebakaran), perkiraan jumlah korban, dan lokasi insiden. Data-data ini akan dikirimkan ke *fuzzy control system* di *server* untuk selanjutnya dihitung prioritas dari kendaraan darurat yang menangani insiden tersebut. Sebelum data dimasukkan ke *fuzzy control system*, data-data ini diolah terlebih dahulu. Data jenis insiden dan data isu medis atau

bahaya kebakaran dihitung dulu bobotnya dan akan dijumlahkan untuk mendapatkan tingkat potensi bahaya insiden.

Setelah mendapatkan bobot masing-masing, tingkat bahaya insiden akan didapatkan dari penjumlahan bobot jenis insiden dengan isu medis untuk kendaraan ambulans, dan penjumlahan bobot jenis insiden dengan bahaya kebakaran untuk kendaraan darurat pemadam kebakaran. Selain itu, data lokasi insiden akan diolah menggunakan formula Haversine untuk mendapatkan lokasi rumah sakit atau pusat pemadam kebakaran terdekat dan mendapatkan jarak antara lokasi insiden dengan lokasi rumah sakit atau pusat pemadam kebakaran terdekat. Setelah itu, data tingkat bahaya insiden, data jarak, dan data perkiraan jumlah korban akan dimasukkan ke *fuzzy control system* untuk mendapatkan prioritas kendaraan darurat.

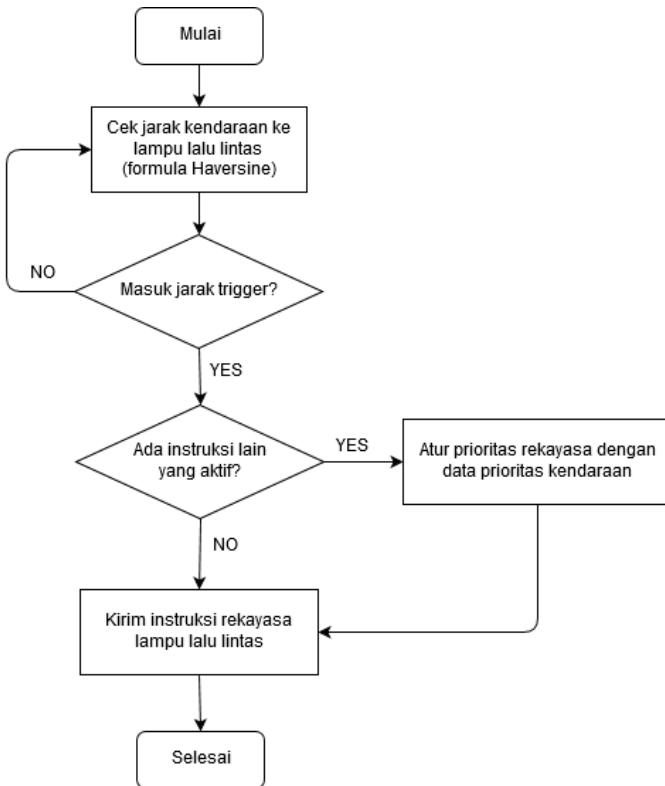
Tahap selanjutnya adalah pencarian rute pada aplikasi Android kendaraaan darurat. Pada tahap ini lokasi saat ini perangkat Android akan dideteksi (posisi *user* diasumsikan telah berada di lokasi servis terdekat) dan dengan lokasi insiden yang telah dilaporkan sebelumnya akan dikirimkan ke Google Maps Directions API untuk mencari rute tercepat dengan permintaan HTTP. Setelah mendapat rute tercepat dari Google Maps Directions API yang berupa data rute yang berformat JSON, aplikasi akan men-*decode* data JSON tersebut dan menampilkan *polylines* rute yang didapat di peta dengan Google Maps Android API. Diagram alir proses pencarian rute ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Proses Pencarian Rute

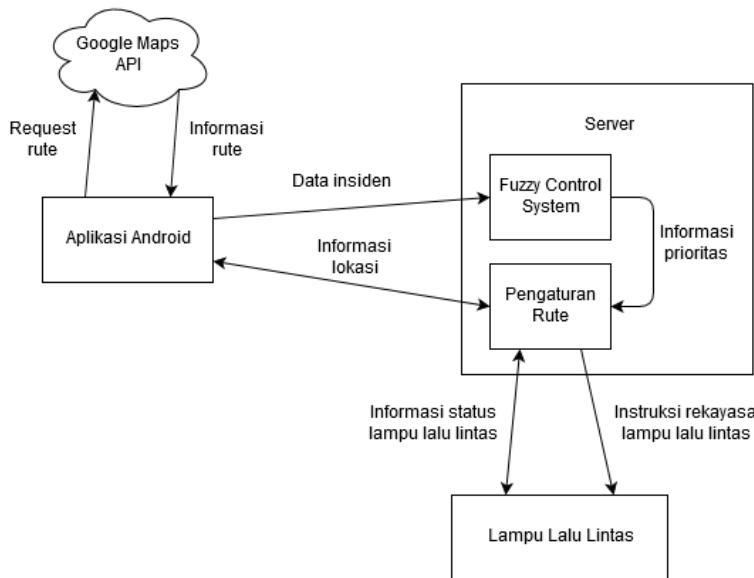
Saat kendaraan darurat melakukan perjalanan, *server* akan mengatur atau merekayasa lampu lalu lintas di jalur yang dilalui kendaraan darurat. Untuk merekayasa lampu lalu lintas, *server* membutuhkan data lokasi kendaraan darurat yang nantinya dihitung jaraknya dengan lampu lalu lintas. Sedangkan untuk lokasi lampu lalu lintas telah tersedia di *database server*. Selanjutnya, *server* akan menghitung jarak antara kendaraan darurat dengan lampu lalu lintas yang akan dilewati. Jika sudah berada dalam jarak jangkauan *trigger* lampu lalu lintas, *server* akan mengirimkan instruksi untuk merekayasa lampu lalu lintas yang akan dilalui kendaraan darurat tersebut untuk menyala hijau dan lampu lainnya di persimpangan tersebut menyala merah.

Jika pada saat proses perekayasaan lalu lintas ada *collision* antar kendaraan darurat di sebuah persimpangan, *server* akan memakai data prioritas yang telah dikalkulasi oleh *fuzzy control system* sebagai acuan prioritas perekayasaan lalu lintas kendaraan darurat yang mana yang akan didahulukan. Jika nilai prioritas sama, maka jarak akan dipakai sebagai parameter cadangan. Diagram alir proses pengaturan rute dan rekayasa lampu lalu lintas dijelaskan pada Gambar 3.2.



Gambar 3.2 Diagram Alir Proses Pengaturan Rute

Secara keseluruhan, arsitektur sistem cerdas pengaturan rute kendaraan darurat menggunakan metode *fuzzy* dijelaskan pada Gambar 3.3.



Gambar 3.3 Arsitektur Sistem

3.3 *Fuzzy Control System* untuk Penentuan Prioritas

Untuk menentukan prioritas dari kendaraan darurat, sistem ini menggunakan *fuzzy control system* pada *server*. *Fuzzy control system* ini menggunakan tiga parameter, yaitu tingkat bahaya insiden, jarak antara kendaraan darurat dan lokasi tujuan, dan perkiraan jumlah korban. Data-data ini didapatkan dari laporan *user* menggunakan aplikasi Android. Khusus untuk nilai tingkat bahaya insiden, perlu dilakukan pemrosesan lebih lanjut untuk mendapatkan nilainya.

Untuk mendapatkan nilai tingkat bahaya insiden, perlu dilakukan penjumlahan bobot jenis insiden dan isu medis untuk kendaraan darurat ambulans atau jenis insiden dan bahaya kebakaran untuk kendaraan darurat pemadam kebakaran. Bobot jenis insiden untuk ambulans ditunjukkan pada Tabel 3.1, bobot jenis insiden untuk pemadam kebakaran ditunjukkan pada Tabel 3.2, bobot isu medis ditunjukkan pada Tabel 3.3, dan bobot bahaya kebakaran ditunjukkan pada Tabel 3.4.

Tabel 3.1 Bobot Jenis Insiden untuk Ambulans

Jenis Insiden	Bobot
Penanganan jenazah	0
Transportasi pasien	1
Kondisi gawat darurat	2

Tabel 3.2 Bobot Jenis Insiden untuk Pemadam Kebakaran

Jenis Insiden	Bobot
Pemadaman kebakaran	2
Penyelamatan jiwa	2

Tabel 3.3 Bobot Isu Medis

Isu Medis	Bobot
Korban sudah meninggal	0
Kondisi luka ringan - sedang	2
Korban kritis	3

Tabel 3.4 Bobot Bahaya Kebakaran

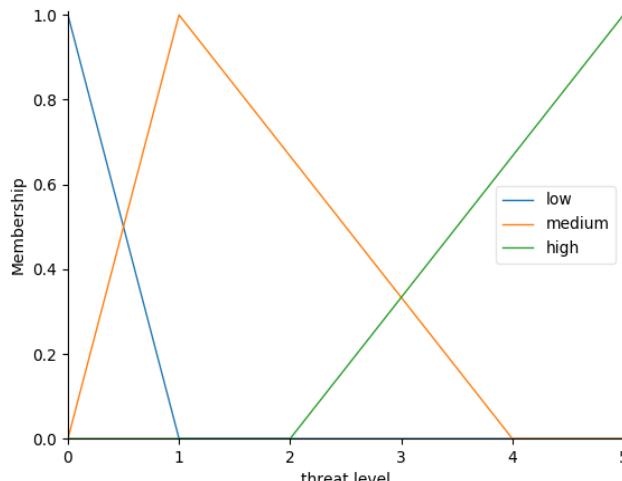
Bahaya Kebakaran	Bobot
Tidak ada bahaya kebakaran	0
Bahaya kebakaran ringan	1
Bahaya kebakaran sedang	2
Bahaya kebakaran berat	3

Setelah mendapatkan semua nilai untuk parameter *fuzzy control system*, selanjutnya akan dilakukan penghitungan prioritas oleh *fuzzy control system* tersebut.

Membership function yang dibuat pada *fuzzy control system* ini ada empat, yaitu *membership fuction* untuk tingkat bahaya insiden, *membership fuction* untuk jarak, *membership function* untuk jumlah korban, dan *membership function* untuk prioritas. Keempat *membership function* ini bertipe *triangular membership function*.

3.3.1 *Membership Function* Tingkat Bahaya Insiden

Membership function ini digunakan sebagai masukan (*antecedents*). Tingkat bahaya insiden digambarkan dengan tiga himpunan *fuzzy*, yaitu *low*, *medium*, dan *high* sesuai dengan hasil penjumlahan bobot dengan nilai maksimal 5. Gambar 3.4 menjelaskan tentang *range* dari masing-masing himpunan *fuzzy* dari *membership function* tingkat bahaya insiden.

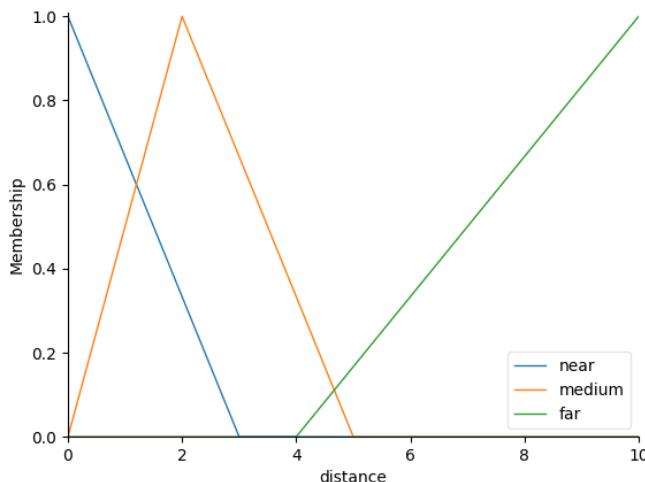


Gambar 3.4 *Membership Function* Tingkat Bahaya Insiden

Himpunan *fuzzy* tingkat bahaya insiden nilainya disesuaikan dengan bobot-bobot insiden yang telah ditetapkan. Contohnya, untuk himpunan *fuzzy* tingkat bahaya insiden *high*, nilai minimalnya adalah 2 yang merupakan bobot minimal dari insiden penyelamatan jiwa, pemadaman kebakaran, dan kondisi gawat darurat. Untuk himpunan *fuzzy medium*, titik puncaknya adalah 1 untuk membedakan tingkat bahaya dari penanganan jenazah dan transportasi pasien pada ambulans. Sedangkan untuk himpunan *fuzzy low*, hanya untuk insiden penanganan jenazah saja.

3.3.2 *Membership Function* Jarak

Membership function ini digunakan sebagai masukan (*antecedents*). Jarak digambarkan dengan tiga himpunan *fuzzy*, yaitu *near*, *medium*, dan *far*. Parameter jarak menggunakan satuan kilometer (km). Gambar 3.5 menjelaskan tentang *range* dari masing-masing himpunan *fuzzy* dari *membership function* jarak.

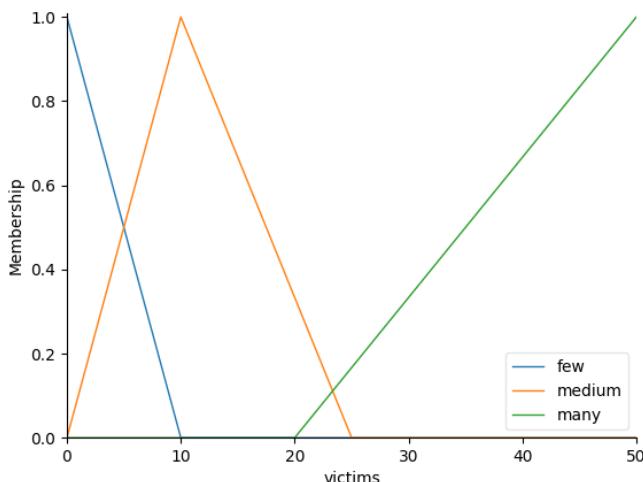


Gambar 3.5 *Membership Function* Jarak

Himpunan fuzzy jarak *far* didapatkan dari jarak maksimal yang didapat oleh rumah sakit atau pusat pemadam kebakaran yaitu ± 10 km. Dari jarak maksimal tersebut, ditentukan bahwa jarak *medium* merupakan setengah dari jarak maksimal sehingga didapat himpunan jarak *medium* mencakup jarak tempuh hingga 5 km. Dan yang terakhir untuk himpunan jarak *low* mencakup jarak tempuh hingga 3 km yang merupakan pembulatan keatas dari setengah jarak *medium*.

3.3.3 *Membership Function Jumlah Korban*

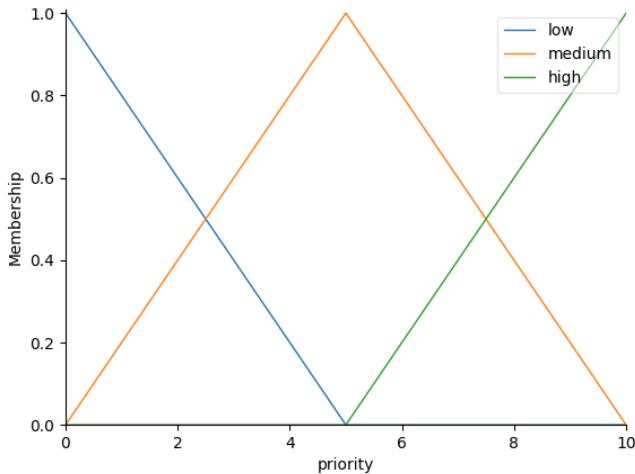
Membership function ini digunakan sebagai masukan (*antecedents*). Jumlah korban digambarkan dengan tiga himpunan *fuzzy*, yaitu *few*, *medium*, dan *many*. Gambar 3.6 menjelaskan tentang *range* dari masing-masing himpunan *fuzzy* dari *membership function* jumlah korban.



Gambar 3.6 *Membership Function Jumlah Korban*

3.3.4 Membership Function Prioritas

Membership function ini digunakan sebagai keluaran (*consequents*). Prioritas digambarkan dengan tiga himpunan *fuzzy*, yaitu *low*, *medium*, dan *high*. Keluaran dari *membership function* ini menentukan prioritas kendaraan darurat dengan skala 1-10. Gambar 3.7 menjelaskan tentang *range* dari masing-masing himpunan *fuzzy* dari *membership function* prioritas.



Gambar 3.7 Membership Function Prioritas

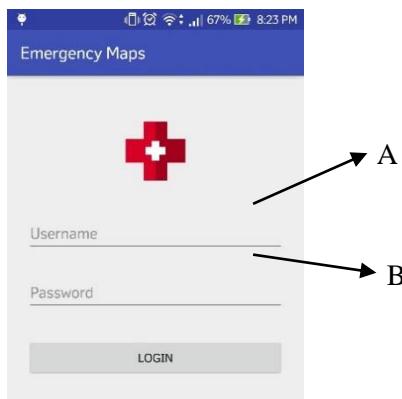
3.3.5 Fuzzy Rule

Fuzzy rule yang digunakan adalah:

1. IF tingkat bahaya insiden *low* OR jarak *low* THEN prioritas *low*
2. IF tingkat bahaya insiden *medium* THEN prioritas *medium*
3. IF tingkat bahaya insiden *high* OR jarak *far* THEN prioritas *high*
4. IF jumlah korban *many* THEN prioritas *high*

3.4 Perancangan Antarmuka

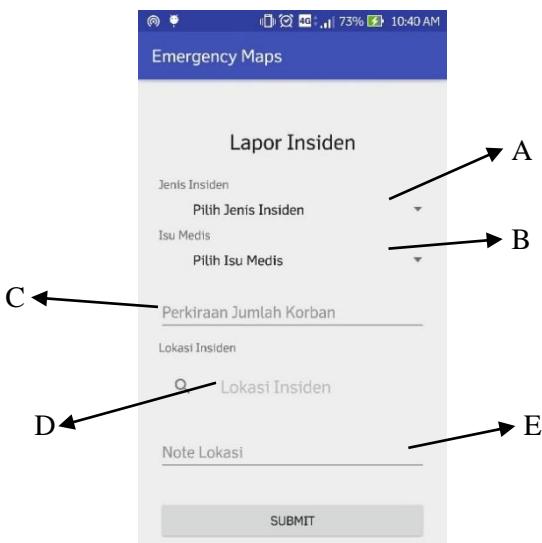
Aplikasi antarmuka digunakan untuk mempermudah pengguna dalam melakukan pelaporan insiden. Aplikasi antarmuka yang dibuat pada Tugas Akhir ini menggunakan perangkat bergerak bersistem operasi Android. Ada beberapa desain tampilan antarmuka yang akan digunakan, yaitu antarmuka *login*, antarmuka lapor insiden, dan antarmuka tampilkan rute pada Google Maps. Gambar 3.8, 3.9, dan 3.10 merupakan desain tampilan antarmuka yang dibuat pada Tugas Akhir ini.



Gambar 3.8 Desain Tampilan Antarmuka *Login*

Adapun komponen yang terdapat dalam Gambar 3.8 antara lain:

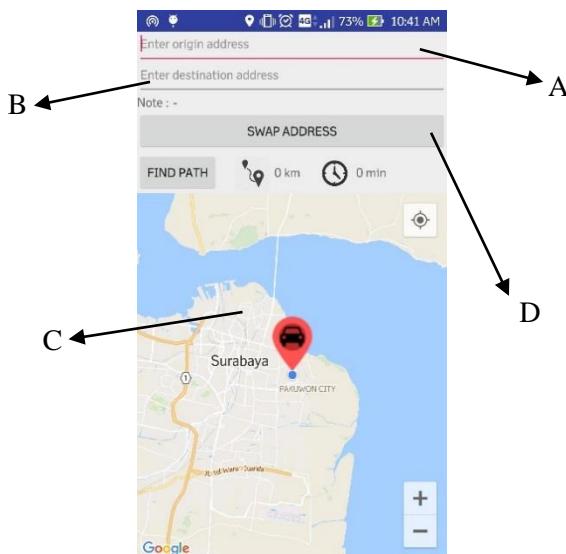
- A. Bagian A merupakan form untuk memasukan *username*.
- B. Bagian B merupakan form untuk memasukan *password*.



Gambar 3.9 Desain Tampilan Antarmuka Lapor Insiden

Adapun komponen yang terdapat dalam Gambar 3.9 antara lain:

- A. Bagian A merupakan form untuk memasukan jenis insiden.
- B. Bagian B merupakan form untuk memasukan isu medis untuk aplikasi ambulans atau bahaya kebakaran untuk aplikasi pemadam kebakaran.
- C. Bagian C merupakan form untuk memasukan perkiraan jumlah korban.
- D. Bagian D merupakan form untuk memasukkan lokasi insiden. Form ini merupakan form *autocomplete* yang disediakan oleh Google Maps Places API.
- E. Bagian E merupakan form untuk memasukkan *note* untuk menjelaskan detail lokasi jika diperlukan.



Gambar 3.10 Desain Tampilan Antarmuka Tampilkan Rute

Adapun komponen yang terdapat dalam Gambar 3.10 antara lain:

- Bagian A merupakan form untuk memasukan *origin address*. Form ini secara *default* menampilkan alamat dari lokasi perangkat Android.
- Bagian B merupakan form untuk memasukan *destination address*. Form ini secara *default* menampilkan alamat dari lokasi insiden yang telah dilaporkan sebelumnya.
- Bagian C merupakan tampilan peta dari Google Maps yang nantinya akan digunakan untuk menampilkan rute dari *origin address* ke *destination address* setelah *user* menekan tombol *Find Path*.
- Bagian D merupakan tombol untuk menukar *origin address* dengan *destination address* jika ingin mencari rute kembali ke lokasi awal.

BAB IV

IMPLEMENTASI

Bab ini berisi penjelasan mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa *Pseudocode* untuk membangun program.

4.1 Lingkungan Implementasi

Implementasi sistem cerdas pengaturan rute kendaraan darurat menggunakan metode *fuzzy* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Privilege	Perangkat	Jenis	Spesifikasi
Client	Perangkat Keras	Prosesor	Intel(R) Atom(TM) CPU Z3580 2,33 GHz
		Memori	4 GB DDR3
	Perangkat Lunak	Sistem Operasi	Android 5.0 Lollipop
Server	Perangkat Keras	Prosesor	Intel(R) Core(TM) i3- 2,40 GHz
		Memori	4 GB 1333 MHz DDR3
	Perangkat Lunak	Sistem Operasi	Windows 8

<i>Privilege</i>	Perangkat	Jenis	Spesifikasi
		Perangkat Pengembang	Android Studio IDE 3.0.1 dan PyCharm Community Edition IDE 2017.1

4.2 Implementasi

Pada sub bab implementasi ini menjelaskan mengenai pembangunan perangkat lunak secara detail dan menampilkan *Pseudocode* yang digunakan mulai tahap *login* di aplikasi kendaraan darurat hingga proses pengaturan rute kendaraan dan rekayasa lampu lalu lintas. Pada Tugas Akhir ini, implementasi aplikasi kendaraan darurat menggunakan bahasa pemrograman Java, implementasi *fuzzy control system* menggunakan bahasa pemrograman Python, *server side scripting* menggunakan PHP, dan implementasi database menggunakan MySQL.

4.2.1 Database

Database pada *server* menyimpan data-data terkait insiden baik yang dilaporkan dari ambulans maupun pemadam kebakaran, lokasi rumah sakit dan pusat pemadam kebakaran, lokasi lampu lalu lintas beserta statusnya, lokasi kendaraan darurat beserta statusnya, dan juga data *user*. Tabel-tabel yang ada pada *database* ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Tabel-tabel pada Database

Nama Tabel	Keterangan
am_incident_data	Merupakan tabel penyimpanan data tentang insiden dari ambulans
pb_incident_data	Merupakan tabel penyimpanan data tentang insiden dari pemadam kebakaran
emergency_services	Merupakan tabel penyimpanan data lokasi rumah sakit dan pusat pemadam kebakaran
vehicles	Merupakan tabel data lokasi kendaraan beserta detailnya
traffic_lights	Merupakan tabel penyimpanan data lokasi lampu lalu lintas beserta statusnya
users	Merupakan tabel penyimpanan data <i>user</i>

Atribut-atribut yang ada pada tabel am_incident_data dapat dilihat pada Tabel 4.3.

Tabel 4.3 Atribut Tabel am_incident_data

Nama Atribut	Tipe (Ukuran)	Keterangan
am_incident_id	int	Menyimpan <i>id</i> insiden
am_incident_type	varchar(50)	Menyimpan tipe insiden
am_incident_address	varchar(100)	Menyimpan alamat insiden
am_incident_lat	varchar(20)	Menyimpan <i>latitude</i> insiden
am_incident_long	varchar(20)	Menyimpan <i>longitude</i> insiden
am_medical_issue	varchar(50)	Menyimpan isu medis

Nama Atribut	Tipe (Ukuran)	Keterangan
am_nearest_service	int	Menyimpan <i>id</i> rumah sakit terdekat
am_priority	float	Menyimpan prioritas
am_distance	float	Menyimpan jarak
am_incident_victims	int	Menyimpan jumlah korban
am_note	varchar(100)	Menyimpan <i>note</i> lokasi tambahan

Atribut-atribut yang ada pada tabel pb_incident_data dapat dilihat pada Tabel 4.4.

Tabel 4.4 Atribut Tabel pb_incident_data

Nama Atribut	Tipe (Ukuran)	Keterangan
pb_incident_id	int	Menyimpan <i>id</i> insiden
pb_incident_type	varchar(50)	Menyimpan tipe insiden
pb_incident_address	varchar(100)	Menyimpan alamat insiden
pb_incident_lat	varchar(20)	Menyimpan <i>latitude</i> insiden
pb_incident_long	varchar(20)	Menyimpan <i>longitude</i> insiden
pb_fire_threat	varchar(50)	Menyimpan bahaya kebakaran
pb_nearest_service	int	Menyimpan <i>id</i> pusat pemadam terdekat
pb_priority	float	Menyimpan prioritas
pb_distance	float	Menyimpan jarak
pb_incident_victims	int	Menyimpan jumlah korban
pb_note	varchar(100)	Menyimpan <i>note</i> lokasi tambahan

Atribut-atribut yang ada pada tabel emergecy_services dapat dilihat pada Tabel 4.5.

Tabel 4.5 Atribut Tabel emergency_services

Nama Atribut	Tipe (Ukuran)	Keterangan
services_id	int	Menyimpan <i>id</i> servis
services_name	varchar(100)	Menyimpan nama servis
services_type	varchar(50)	Menyimpan tipe servis
services_address	varchar(100)	Menyimpan alamat insiden
services_lat	varchar(20)	Menyimpan <i>latitude</i> servis
services_long	varchar(20)	Menyimpan <i>longitude</i> servis

Atribut-atribut yang ada pada tabel vehicles dapat dilihat pada Tabel 4.6.

Tabel 4.6 Atribut Tabel vehicles

Nama Atribut	Tipe (Ukuran)	Keterangan
vehicle_id	int	Menyimpan <i>id</i> kendaraan
vehicle_lat	varchar(20)	Menyimpan <i>latitude</i> kendaraan
vehicle_long	varchar(20)	Menyimpan <i>longitude</i> kendaraan
vehicle_priority	float	Menyimpan prioritas
vehicle_distance	float	Menyimpan jarak
vehicle_active	int	Menyimpan penanda kendaraan aktif

Atribut-atribut yang ada pada tabel traffic_lights dapat dilihat pada Tabel 4.7.

Tabel 4.7 Atribut Tabel traffic_lights

Nama Atribut	Tipe (Ukuran)	Keterangan
tl_id	int	Menyimpan <i>id</i> lampu lalu lintas
tl_status	varchar(20)	Menyimpan status lampu lalu lintas
tl_lat	varchar(20)	Menyimpan <i>latitude</i> lampu lalu lintas
tl_long	varchar(20)	Menyimpan <i>longitude</i> lampu lalu lintas
tl_pos	varchar(10)	Menyimpan posisi lampu lalu lintas pada persimpangan

Atribut-atribut yang ada pada tabel users dapat dilihat pada Tabel 4.8.

Tabel 4.8 Atribut Tabel users

Nama Atribut	Tipe (Ukuran)	Keterangan
user_id	int	Menyimpan <i>id user</i>
username	varchar(20)	Menyimpan <i>username</i>
password	varchar(50)	Menyimpan <i>password</i>
type	varchar(20)	Menyimpan tipe kendaraan dari <i>user</i>

4.2.2 Proses *Login*

Proses *login* merupakan proses awal dari sistem ini. Pertama-tama *user* memasukkan *username* dan *password* pada halaman *login* di aplikasi kendaraan darurat. Lalu, data *username* dan *password* dikirim ke *server* untuk ditentukan jenis kendaraan dari *user* tersebut. Penentuan jenis kendaraan darurat tersebut nantinya digunakan sebagai informasi untuk aplikasi kendaraan darurat untuk menuju ke halaman lapor ambulans atau ke halaman lapor pemadam kebakaran. Pada sistem ini, jika *user* merupakan kendaraan darurat bertipe ambulans, *server* akan mengirimkan *response message* “Login 1”. Sedangkan jika *user* merupakan kendaraan darurat bertipe pemadam kebakaran, *server* akan mengirimkan *response message* “Login 2”. Proses *login* pada *server* dapat dilihat pada Pseudocode 4.1.

```

1  FUNCTION login():
2      in_username := $_POST['username']
3      in_password := $_POST['password']
4      sql := "SELECT * FROM users WHERE username =
in_username AND password = in_password"
5      result := mysqli_query(conn ,sql)
6      check := mysqli_fetch_array(result)
7      type := check['type']
8      IF isset(check) && type == "Ambulans":
9          echo "Login 1"
10     ENDIF
11     ELSE IF isset(check) && type == "Pemadam
Kebakaran":
12         echo "Login 2"
13     ENDIF
14     ELSE
15         echo "Invalid Username or Password"
16     ENDIF
17 ENDFUNCTION

```

Pseudocode 4.1 Kode Program *Login*

Setelah aplikasi kendaraan darurat menerima response message dari server, aplikasi akan beralih ke halaman pelaporan insiden sesuai dengan tipe kendaraan darurat. Proses penentuan jenis pelaporan kendaraan darurat dapat dilihat pada Pseudocode 4.2.

```

1   FUNCTION onPostExecute (String httpResponseMsg) :
2       super.onPostExecute (httpResponseMsg)
3           IF httpResponseMsg.equalsIgnoreCase ("Login
1") :
4               Intent intent := new
Intent (LoginActivity.this,
6 LaporAmbulansActivity.class)
7                   startActivity (intent)
8           ENDIF
9           ELSE IF
10      httpResponseMsg.equalsIgnoreCase ("Login 2") :
11          Intent intent := new
Intent (LoginActivity.this,
12 LaporDamkarActivity.class)
13             startActivity (intent)
14         ENDIF
15         ELSE
16             Toast.makeText (LoginActivity.this,
17 httpResponseMsg, Toast.LENGTH_LONG).show ()
18         ENDIF
19     ENDFUNCTION
```

Pseudocode 4.2 Kode Program Penentuan Jenis Pelaporan Insiden

4.2.3 Pelaporan Insiden

Setelah sistem membedakan tipe user dari proses login, proses selanjutnya adalah pelaporan insiden. Pada proses pelaporan insiden *user* akan diberi form pelaporan yang berbeda pada aplikasi kendaraan darurat sesuai dengan jenis kendaraan darurat. Dengan adanya sistem pembedaan jenis tersebut, *server* akan dapat dengan mudah menentukan bagaimana insiden tersebut

akan ditangani selanjutnya. *Field-field* yang ada pada form tersebut adalah jenis insiden, isu medis untuk kendaraan darurat bertipe ambulans atau bahaya kebakaran untuk kendaraan darurat bertipe pemadam kebakaran, perkiraan jumlah korban, lokasi insiden, dan *note* lokasi.

Selain itu, aplikasi juga mengirimkan lokasi saat ini dari perangkat Android sebagai informasi lokasi awal kendaraan darurat. Pada Tugas Akhir ini, lokasi awal dari kendaraan darurat berada di lokasi rumah sakit atau pusat pemadam kebakaran terdekat dari lokasi insiden. Proses pengambilan lokasi kendaraan darurat dapat dilihat pada Pseudocode 4.3.

```

1  FUNCTION getLatLong():
2      LocationManager locationManager := 
        (LocationManager)
3      getSystemService(Context.LOCATION_SERVICE)
4          Criteria criteria := new Criteria()
5          Location location := locationManager.
6          getLastKnownLocation(locationManager.
7          getBestProvider(criteria, false))
8          IF location != null:
9              double lat := location.getLatitude()
10             double lng := location.getLongitude()
11             ENDIF
12             RETURN lat, long
13 ENDFUNCTION
```

Pseudocode 4.3 Kode Program Pengambilan Lokasi Kendaraan Darurat

Setelah semua *field* yang dibutuhkan telah dimasukkan oleh *user*, aplikasi kendaraan darurat akan mengirimkan data-data laporan tersebut ke *server*. Proses pelaporan insiden pada aplikasi kendaraan darurat dapat dilihat pada Pseudocode 4.4.

```

1  FUNCTION laporFunction(incidentType, issue,
2    destination, victims, note, v_lat, v_long):
3      hashMap.put("incidentType", params[0])
4      hashMap.put("issue", params[1])
5      hashMap.put("destination", params[2])
6      hashMap.put("victims", params[3])
7      hashMap.put("note", params[4])
8      hashMap.put("v_lat", params[5])
9      hashMap.put("v_long", params[6])
10     finalResult :=
11       httpParse.postRequest(hashMap, HttpURL)
12
13       LaporFunctionClass laporFunctionClass := new
14       LaporFunctionClass()
15
16       laporFunctionClass.execute(incidentType,
17         issue, destination, victims, note, v_lat,
18         v_long)
19       RETURN finalResult
20   ENDFUNCTION
```

Pseudocode 4.4 Kode Program Pelaporan Insiden

Tahap selanjutnya adalah penyimpanan data insiden yang telah dilaporkan ke *database* oleh *server*. Penyimpanan data ini dibagi menjadi dua, yaitu penyimpanan data insiden untuk ambulans dan penyimpanan data insiden untuk pemadam kebakaran. Data-data yang disimpan adalah tipe insiden, isu medis atau bahaya kebakaran, perkiraan jumlah korban, note lokasi (jika ada), rumah sakit atau pusat pemadam kebakaran terdekat, prioritas, dan lokasi insiden. Untuk mencari rumah sakit atau pusat pemadam kebakaran terdekat, metode yang digunakan adalah formula Haversine. Dengan formula tersebut, dapat dihitung jarak lokasi insiden ke tiap rumah sakit atau pusat pemadam kebakaran lalu dicari yang mana yang mempunyai jarak paling minimum.

Selain itu, lokasi kendaraan darurat juga disimpan. Data ini nantinya akan digunakan untuk proses pengaturan rute dan rekayasa lampu lalu lintas. Proses penyimpanan data insiden untuk ambulans pada *server* dapat dilihat pada Pseudocode 7.1 pada lampiran. Sedangkan proses penyimpanan data insiden untuk

pemadam kebakaran pada *server* dapat dilihat pada Pseudocode 7.2 pada lampiran.

4.2.4 Pencarian Rute

Proses pencarian rute pada sistem ini menggunakan Google Maps Directions API. Untuk mencari rute, aplikasi kendaraan darurat akan mengirimkan HTTP *request* ke Google Maps Direction API dengan data *origin address* berupa alamat dari lokasi awal kendaraan darurat (di rumah sakit atau pusat pemadam kebakaran terdekat dari lokasi insiden) dan data *destination address* berupa alamat dari lokasi terjadinya insiden. Rute dengan indeks ke-0 akan diambil, karena diasumsikan rute dengan indeks ke-0 merupakan rute tercepat yang direkomendasikan oleh Google Maps Directions API.

4.2.4.1 Kelas *Route*

Kelas *route* digunakan sebagai objek atau *container* yang digunakan untuk menampung detail dari rute yang diberikan oleh Google Maps Directions API seperti jarak, perkiraan waktu tempuh, dan tentunya poin-poin dari rute itu sendiri. Untuk detail dari kelas *route*, dapat dilihat pada Pseudocode 4.5.

1	CLASS Route:
2	Distance distance
3	Duration duration
4	String endAddress
5	LatLng endLocation
6	String startAddress
7	LatLng startLocation
8	List<LatLng> points

Pseudocode 4.5 Kode Program Kelas *Route*

4.2.4.2 Proses Pencarian Rute

Proses pencarian rute dimulai dari men-*download* data mentah dari rute yang diminta dalam data yang berformat JSON. Lalu, data JSON tersebut diolah hingga didapatkan data-data rute yang diinginkan yaitu jarak, perkiraan waktu tempuh, dan poin-poin dari rute. Setelah itu, data poin-poin rute yang berupa *list decode* agar menjadi *polylines* yang bisa ditampilkan pada tampilan antarmuka peta di aplikasi kendaraan darurat. Proses pencarian rute dapat dilihat pada Pseudocode 7.3 pada lampiran.

Setelah data-data rute telah didapatkan dan berhasil *decode*, tahap selanjutnya adalah menampilkan rute yang didapat di aplikasi kendaraan darurat. Proses menampilkan rute dapat dilihat pada Pseudocode 4.6.

```

1  FUNCTION onDirectionFinderSuccess(routes):
2      polylinePaths := new ArrayList<>()
3      originMarkers := new ArrayList<>()
4      destinationMarkers := new ArrayList<>()
5
5      for route in routes:
6          originMarkers.add(mMap.addMarker(new
7              MarkerOptions()
8                  .title(route.startAddress)
9                  .position(route.startLocation())))
10
11          destinationMarkers.add(mMap.
12              addMarker(new MarkerOptions()
13                  .title(route.endAddress)
14                  .position(route.endLocation())))
15
16          for i in route.points.size():
17              polylineOptions.add(route.
points.get(i))
                  polylinePaths.add(mMap.
addPolyline())
ENDFOR
ENDFOR
ENDFUNCTION
```

Pseudocode 4.6 Kode Program Menampilkan Rute

4.2.5 Penghitungan Prioritas

Prioritas kendaraan darurat pada sistem ini dihitung menggunakan *fuzzy control system*. Data masukan yang berupa laporan insiden akan dimasukkan ke *fuzzy control system*, yang nantinya akan dihasilkan nilai prioritas dari kendaraan darurat dari data-data insiden yang diperoleh. Proses penghitungan prioritas dapat dilihat di Pseudocode 4.7. Untuk proses inisialisasi *membership function* dapat dilihat pada Pseudocode 7.4 pada lampiran.

```

1  FUNCTION fuzzyControlSystem(t_val, d_val,
2    v_val):
3    initMF()
4    rule1 := ctrl.Rule(threat['low'] |
5      distance['near'], priority['low'])
6    rule2 := ctrl.Rule(threat['medium'],
7      priority['medium'])
8    rule3 := ctrl.Rule(threat['high'] |
9      distance['far'], priority['high'])
10   rule4 := ctrl.Rule(victims['many'],
11     priority['high'])
12
13   prioritize_ctrl :=
14     ctrl.ControlSystem([rule1, rule2, rule3, rule4])
15
16   prioritize :=
17     ctrl.ControlSystemSimulation(prioritize_ctrl)
18
19   prioritize.input['threat level'] := t_val
20   prioritize.input['distance'] := d_val
21   prioritize.input['victims'] := v_val
22
23   prioritize.compute()
24
25   RETURN prioritize.output['priority']
26 ENDFUNCTION
```

Pseudocode 4.7 Kode Program Penghitungan Prioritas

4.2.6 Pengaturan Rute

Proses pengaturan rute membutuhkan data lokasi kendaraan darurat dan data lampu lalu lintas yang akan dituju (salah satu dari empat buah lampu lalu lintas di sebuah persimpangan) sesuai dengan rute dari kendaraan darurat. Proses ini dimulai dari menghitung jarak antara kendaraan darurat dengan lampu lalu lintas yang dituju. Metode pencarian jarak sama dengan proses pencarian rumah sakit atau pusat pemadam kebakaran terdekat, yaitu menggunakan formula Haversine.

Selanjutnya, jika kendaraan darurat sudah masuk jarak *trigger* (100 m), proses rekayasa lampu lalu lintas akan dilakukan. Lampu lalu lintas yang akan dituju oleh kendaraan darurat tersebut akan diberi instruksi untuk menyala hijau, dan lampu lalu lintas lain yang ada di persimpangan tersebut diberi instruksi untuk menyala merah sampai kendaraan darurat sudah melewati persimpangan tersebut.

Jika ada lebih dari satu kendaraan darurat yang aktif untuk proses rekayasa lampu lalu lintas, sistem akan mengurutkan prioritas kendaraan darurat. Kendaraan darurat dengan nilai prioritas paling tinggi akan didahulukan. Jika nilai prioritas sama, maka jarak yang paling jauh akan didahulukan. Proses pengaturan rute dapat dilihat pada Pseudocode 7.5 pada lampiran.

BAB V

HASIL UJI COBA DAN EVALUASI

Bab ini berisi penjelasan mengenai skenario uji coba dan evaluasi performa pada sistem cerdas pengaturan rute kendaraan darurat dengan metode *fuzzy*. Hasil uji coba didapatkan dari implementasi pada bab 4 dengan skenario yang berbeda. Bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

5.1 Lingkungan Pengujian

Lingkungan pengujian pada uji coba sistem cerdas pengaturan rute kendaraan darurat menggunakan metode *fuzzy* menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Pengujian

Privilege	Perangkat	Jenis	Spesifikasi
Client	Perangkat Keras	Prosesor	Intel(R) Atom(TM) CPU Z3580 2,33 GHz
		Memori	4 GB DDR3
	Perangkat Lunak	Sistem Operasi	Android 5.0 Lollipop
Server	Perangkat Keras	Prosesor	Intel(R) Core(TM) i3- 2,40 GHz
		Memori	4 GB 1333 MHz DDR3
	Perangkat Lunak	Sistem Operasi	Windows 8

Privilege	Perangkat	Jenis	Spesifikasi
		Perangkat Pengembang	Android Studio IDE 3.0.1 dan PyCharm Community Edition IDE 2017.1

5.2 Data Pengujian

Sub bab ini menjelaskan mengenai data yang digunakan untuk uji coba. Adapun data pengujian dalam Tugas Akhir ini adalah data insiden yang dilaporkan dari aplikasi Android kendaraan darurat. Seperti yang sudah dijelaskan sebelumnya, data insiden ini berupa data-data tentang jenis insiden yang terjadi, isu medis (untuk ambulans) atau bahaya kebakaran (untuk pemadam kebakaran), perkiraan jumlah korban, lokasi kendaraan darurat, dan lokasi insiden.

Dari data insiden tersebut, data jenis insiden, isu medis atau bahaya kebakaran, perkiraan jumlah korban, dan lokasi insiden dari aplikasi Android kendaraan darurat nantinya akan dikirim ke *server*. Data-data tersebut akan digunakan sebagai data masukan untuk *fuzzy control system* di *server* untuk menentukan prioritas dari kendaraan darurat tersebut. Sedangkan untuk data masukan pengaturan rute, yang dibutuhkan adalah data lokasi kendaraan darurat yang juga dikirim saat proses pelaporan insiden dilakukan. Data lokasi ini nantinya akan digunakan untuk proses rekayasa lampu lalu lintas pada rute kendaraan darurat.

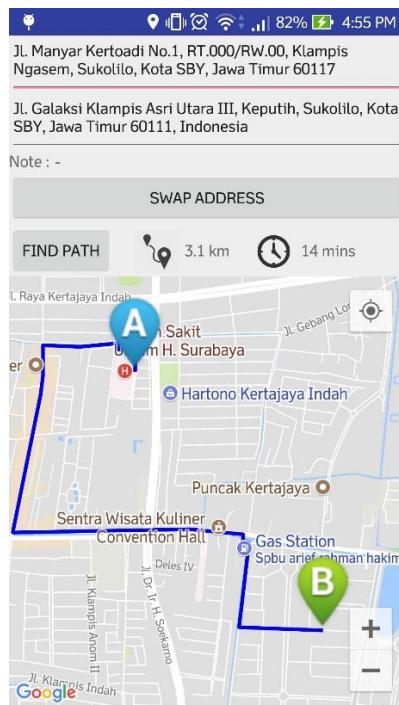
5.3 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, program akan diuji apakah sudah berjalan dengan benar dan bagaimana performa pada masing-masing skenario. Di dalam skenario uji coba, ada kondisi-kondisi tertentu yang digunakan untuk menguji performa sistem.

Skenario uji coba akan dibagi menjadi tiga bagian, yaitu skenario uji coba performa pencarian rute untuk menguji performa sistem dalam pencarian rute tercepat untuk kendaraan darurat, skenario uji coba performa penghitungan prioritas untuk menguji performa sistem dalam melakukan penghitungan prioritas kendaraan darurat sesuai dengan kondisi yang diberikan, dan yang terakhir skenario uji coba performa pengaturan rute untuk menguji performa sistem dalam mengatur dan merekayasa rute dan lampu lalu lintas yang akan dilalui kendaraan darurat.

5.3.1 Skenario Uji Coba Performa Pencarian Rute

Pada uji coba pencarian rute, sistem akan diuji performanya dalam mencari rute dari lokasi awal menuju ke lokasi sesuai dengan skenario uji coba yang akan dilakukan. Pengujian akan dilakukan mulai dari proses pelaporan insiden melalui aplikasi kendaraan darurat, proses pencarian lokasi rumah sakit atau pusat pemadam kebakaran terdekat menggunakan formula Haversine, sampai proses pencarian rute menggunakan Google Maps Directions API. *Screenshot* hasil dari pencarian rute pada aplikasi Android ditunjukkan pada Gambar 5.1.



Gambar 5.1 Contoh Hasil Pencarian Rute pada Aplikasi

Dalam proses pencarian lokasi rumah sakit atau pusat pemadam kebakaran terdekat, sistem akan mencari dari lokasi yang telah disediakan di *database*. Proses pencarian ini menggunakan formula Haversine yang mengkalkulasi jarak dari lokasi insiden dengan rumah sakit atau pusat pemadam kebakaran yang ada, lalu akan diambil yang jaraknya terpendek. Tabel lokasi rumah sakit dan pusat pemadam kebakaran yang disediakan dapat dilihat pada Tabel 5.2.

Tabel 5.2 Tabel Lokasi Rumah Sakit dan Pusat Pemadam Kebakaran

Nama Lokasi	Latitude	Longitude	Tipe Lokasi
Rumah Sakit Haji Surabaya	-7.283213	112.779637	Rumah Sakit
Rumah Sakit Universitas Airlangga	-7.269805	112.784943	Rumah Sakit
RSUD Dr. Soetomo	-7.267910	112.758470	Rumah Sakit
Dinas Kebakaran Pos Keputih Surabaya	-7.289785	112.803917	Pusat Pemadam Kebakaran
Pos Pemadam Sukolilo	-7.299667	112.770445	Pusat Pemadam Kebakaran
Pemadam Kebakaran Pos Pembantu 2 Cepu	-7.278351	112.762899	Pusat Pemadam Kebakaran

Dalam uji coba ini, terdapat 2 macam skenario uji coba, yaitu:

1. Pencarian rute dari lokasi rumah sakit terdekat ke lokasi insiden pada kendaraan darurat bertipe ambulans.
2. Pencarian rute dari lokasi pusat pemadam kebakaran terdekat ke lokasi insiden pada kendaraan darurat bertipe pemadam kebakaran.

5.3.1.1 Skenario Uji Coba Pencarian Rute 1

Skenario uji coba pencarian rute 1 adalah pencarian rute dari lokasi rumah sakit terdekat ke lokasi insiden pada kendaraan darurat bertipe ambulans. Uji coba akan dilakukan sebanyak lima

kali tiap alamat insiden. Lokasi insiden yang akan digunakan dalam uji coba ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Tabel Lokasi Insiden Pencarian Rute

No.	Alamat Insiden	Latitude	Longitude
1	Jl. Kertajaya Indah Tim. X, Manyar Sabrangan, Mulyorejo, Kota SBY, Jawa Timur 60116	-7.2782284	112.7822267
2	Jl. Sutorejo Tengah XI, Dukuh Sutorejo, Mulyorejo, Kota SBY, Jawa Timur 60113	-7.26536	112.7914951
3	Jl. Gubeng Kertajaya VII Raya, Airlangga, Gubeng, Kota SBY, Jawa Timur 60286	-7.2757908	112.7549644
4	Jl. Galaksi Klampis Asri Utara III, Keputih, Sukolilo, Kota SBY, Jawa Timur 60111	-7.2934059	112.7879553
5	Jl. Manyar Jaya XIV, Menur Pumpungan, Sukolilo, Kota SBY, Jawa Timur 60118	-7.2949397	112.7717143

Dari uji coba ini, akan didapatkan tingkat performa pencarian rute yang diukur dari kesuksesan pencarian rute, kesuksesan pencarian rumah sakit terdekat, dan waktu eksekusi pada saat pencarian rute. Hasil dari uji coba yang telah dilakukan dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Uji Coba Pencarian Rute 1

Uji coba ke-	Alamat Insiden	RS Terdekat yang Didapat	Pencarian Rute	Rata-rata Waktu (ms)
1	Jl. Kertajaya Indah Tim. X	Rumah Sakit Haji Surabaya	Sukses	19,00
2	Jl. Sutorejo Tengah XI	Rumah Sakit Universitas Airlangga	Sukses	18,00
3	Jl. Gubeng Kertajaya VII Raya	RSUD Dr. Soetomo	Sukses	20,80
4	Jl. Galaksi Klampis Asri Utara III	Rumah Sakit Haji Surabaya	Sukses	22,00
5	Jl. Manyar Jaya XIV	Rumah Sakit Haji Surabaya	Sukses	21,00
Total rata-rata waktu eksekusi (ms)				20,16

Dilihat dari Tabel 5.4, sistem berhasil mencari rumah sakit terdekat dan mencari rute dengan performa yang bagus, dan dengan rata-rata waktu eksekusi 20,16 ms juga menunjukkan bahwa sistem ini sudah bekerja dengan baik dalam mencari rute dari rumah sakit terdekat ke lokasi insiden.

5.3.1.2 Skenario Uji Coba Pencarian Rute 2

Skenario uji coba pencarian rute 2 adalah pencarian rute dari lokasi pusat pemadam kebakaran terdekat ke lokasi insiden pada kendaraan darurat bertipe ambulans. Uji coba juga akan dilakukan sebanyak lima kali tiap alamat insiden. Lokasi insiden yang akan digunakan dalam uji coba ini sama dengan uji coba performa pencarian rute 1 dan dapat dilihat pada Tabel 5.3 diatas.

Dari uji coba ini, akan didapatkan tingkat performa pencarian rute yang diukur dari kesuksesan pencarian rute, kesuksesan pencarian pusat pemadam kebakaran terdekat, dan waktu eksekusi pada saat pencarian rute. Hasil dari uji coba yang telah dilakukan dapat dilihat pada Tabel 5.5, dan *screenshot* contoh hasil pencarian rute pada aplikasi Android dapat dilihat pada Gambar 5.2.

Tabel 5.5 Hasil Uji Coba Pencarian Rute 2

Uji coba ke-	Alamat Insiden	Pusdamkar Terdekat yang Didapat	Pencarian Rute	Rata-rata Waktu (ms)
1	Jl. Kertajaya Indah Tim. X	Pemadam Kebakaran Pos Pembantu 2 Cepu	Sukses	16,40
2	Jl. Sutorejo Tengah XI	Dinas Kebakaran Pos Keputih Surabaya	Sukses	17,60
3	Jl. Gubeng Kertajaya VII Raya	Pemadam Kebakaran Pos Pembantu 2 Cepu	Sukses	17,40
4	Jl. Galaksi Klampis Asri Utara III	Dinas Kebakaran Pos Keputih Surabaya	Sukses	17,80
5	Jl. Manyar Jaya XIV	Pos Pemadam Sukolilo	Sukses	22,40
Total rata-rata waktu eksekusi (ms)				18,32

Dilihat dari Tabel 5.5, sistem berhasil mencari pusat pemadam kebakaran terdekat dan mencari rute dengan performa yang bagus, dan dengan rata-rata waktu eksekusi 18,32 ms juga menunjukkan bahwa sistem ini sudah bekerja dengan baik dalam mencari rute dari pusat pemadam kebakaran terdekat ke lokasi insiden.

5.3.2 Skenario Uji Coba Performa Penghitungan Prioritas

Pada uji coba penghitungan prioritas, sistem akan diuji performanya dalam menghitung prioritas kendaraan darurat dengan *fuzzy control system* dari setiap skenario uji coba yang akan dilakukan. Di setiap skenario uji coba, berbagai kondisi yang berbeda akan diterapkan. Dalam uji coba ini, terdapat 2 macam skenario uji coba, yaitu:

1. Penghitungan prioritas kendaraan darurat bertipe ambulans dengan berbagai kondisi insiden.
2. Penghitungan prioritas kendaraan darurat bertipe pemadam kebakaran dengan berbagai kondisi insiden.

Kondisi-kondisi insiden yang digunakan dalam uji coba ini nantinya akan diubah-ubah untuk tiap skenario. Kondisi insiden tersebut meliputi jenis insiden, isu medis atau bahaya kebakaran, perkiraan jumlah korban, dan jarak. Untuk parameter jarak, nilai yang digunakan untuk jarak *near* adalah 1 km, jarak *medium* adalah 3,5 km, dan jarak *far* adalah 5 km. Sedangkan untuk parameter jumlah korban, nilai yang digunakan untuk *few* adalah 5, *medium* adalah 15, dan *many* adalah 30. Uji coba akan dilakukan sebanyak lima kali untuk setiap kondisi. Kondisi insiden yang digunakan untuk ambulans dapat dilihat pada Tabel 5.6, dan kondisi insiden yang digunakan untuk pemadam kebakaran dapat dilihat pada Tabel 5.7.

Tabel 5.6 Tabel Kondisi Insiden untuk Ambulans

No. Kondisi	Jenis Insiden	Isu Medis	Jumlah Korban	Jarak
1	Penanganan Jenazah	Korban sudah meninggal	<i>Few</i> , <i>Medium</i> , <i>Many</i>	<i>Near</i> , <i>Medium</i> , <i>Far</i>
2	Transportasi Pasien	Korban luka ringan - sedang	<i>Few</i> , <i>Medium</i> , <i>Many</i>	<i>Near</i> , <i>Medium</i> , <i>Far</i>

No. Kondisi	Jenis Insiden	Isu Medis	Jumlah Korban	Jarak
3	Transportasi Pasien	Korban kritis	Few, Medium, Many	Near, Medium, Far
4	Kondisi gawat darurat	Korban kritis	Near, Medium, Far	Near, Medium, Far

Tabel 5.7 Tabel Kondisi Insiden untuk Pemadam Kebakaran

No. Kondisi	Jenis Insiden	Bahaya Kebakaran	Jumlah Korban	Jarak
1	Pemadaman Kebakaran	Bahaya kebakaran ringan	Few, Medium, Many	Near, Medium, Far
2	Pemadaman Kebakaran	Bahaya kebakaran sedang	Few, Medium, Many	Few, Medium, Many
3	Pemadaman Kebakaran	Bahaya kebakaran berat	Few, Medium, Many	Few, Medium, Many
4	Penyelamatan Jiwa	Tidak ada bahaya kebakaran	Few, Medium, Many	Near, Medium, Far

5.3.2.1 Skenario Uji Coba Penghitungan Prioritas 1

Skenario uji coba penghitungan prioritas 1 adalah penghitungan nilai prioritas untuk kendaraan darurat bertipe ambulans. Uji coba akan dilakukan sesuai dengan kondisi insiden untuk ambulans pada Tabel 5.6 diatas.

Dari uji coba ini, akan didapatkan tingkat performa penghitungan prioritas kendaraan darurat ambulans yang diukur dari waktu eksekusi dari *fuzzy control system*. Hasil dari uji coba yang telah dilakukan dapat dilihat pada Tabel 5.8.

Tabel 5.8 Hasil Uji Coba Penghitungan Prioritas 1

Kondisi			Nilai Prioritas	Rata-rata Waktu (ms)
No.	Korban	Jarak		
Kondisi 1	Few	Near	1,667	11,60
		Medium	1,667	13,40
		Far	4,405	13,60
	Medium	Near	1,667	16,00
		Medium	1,667	11,40
		Far	4,405	14,60
	Many	Near	3,889	14,20
		Medium	3,889	16,00
		Far	4,405	10,80
Kondisi 2	Few	Near	4,255	14,40
		Medium	5,404	13,20
		Far	5,838	13,40
	Medium	Near	4,255	13,00
		Medium	5,404	11,60
		Far	5,838	13,20
	Many	Near	4,255	13,00
		Medium	5,404	12,40
		Far	5,838	12,40
Kondisi 3	Few	Near	5,000	12,40
		Medium	8,194	12,60
		Far	8,194	10,80
	Medium	Near	5,000	15,60
		Medium	8,194	13,20
		Far	8,194	13,20
	Many	Near	5,000	15,40
		Medium	8,194	14,60

Kondisi			Nilai Prioritas	Rata-rata Waktu (ms)
No.	Korban	Jarak		
Kondisi 4	<i>Few</i>	<i>Far</i>	8,194	15,20
		<i>Near</i>	5,261	12,20
		<i>Medium</i>	8,333	13,20
		<i>Far</i>	8,333	15,20
	<i>Medium</i>	<i>Near</i>	5,261	15,40
		<i>Medium</i>	8,333	17,80
		<i>Far</i>	8,333	15,80
	<i>Many</i>	<i>Near</i>	5,261	12,00
		<i>Medium</i>	8,333	14,40
		<i>Far</i>	8,333	14,80
Total rata-rata waktu eksekusi (ms)				13,67

Dilihat dari Tabel 5.8, sistem berhasil menghitung prioritas kendaraan darurat ambulans dengan total rata-rata waktu eksekusi 13,7 ms dengan berbagai kondisi yang disediakan. Untuk hasil penghitungan prioritas, sistem kurang mendapatkan hasil yang optimal karena masih ada beberapa kondisi yang masih sama nilainya meski parameter jumlah korban dan jaraknya berbeda.

5.3.2.2 Skenario Uji Coba Penghitungan Prioritas 2

Skenario uji coba penghitungan prioritas 2 adalah penghitungan nilai prioritas untuk kendaraan darurat bertipe pemadam kebakaran. Uji coba akan dilakukan sesuai dengan kondisi insiden untuk pemadam kebakaran pada Tabel 5.7 diatas.

Dari uji coba ini, akan didapatkan tingkat performa penghitungan prioritas kendaraan darurat pemadam kebakaran yang diukur dari waktu eksekusi dari *fuzzy control system*. Hasil dari uji coba yang telah dilakukan dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Uji Coba Penghitungan Prioritas 2

Kondisi			Nilai Prioritas	Rata-rata Waktu (ms)
No.	Korban	Jarak		
Kondisi 1	Few	Near	4,255	10,6
		Medium	5,404	14,8
		Far	5,838	11,0
	Medium	Near	4,255	12,0
		Medium	5,404	14,2
		Far	5,838	11,4
	Many	Near	5,838	14,2
		Medium	5,404	15,6
		Far	5,838	12,4
Kondisi 2	Few	Near	5,000	13,4
		Medium	8,194	11,6
		Far	8,194	14,4
	Medium	Near	5,000	10,6
		Medium	8,194	11,8
		Far	8,194	17,0
	Many	Near	5,000	14,2
		Medium	8,194	13,4
		Far	8,194	12,8
Kondisi 3	Few	Near	5,261	15,2
		Medium	8,333	10,6
		Far	8,333	14,2
	Medium	Near	5,261	15,0
		Medium	8,333	13,0
		Far	8,333	14,4
	Many	Near	5,261	10,0
		Medium	8,333	15,6

Kondisi			Nilai Prioritas	Rata-rata Waktu (ms)
No.	Korban	Jarak		
Kondisi 4	<i>Few</i>	<i>Far</i>	8,333	11,6
		<i>Near</i>	4,250	11,8
		<i>Medium</i>	5,000	12,6
	<i>Medium</i>	<i>Far</i>	5,514	10,4
		<i>Near</i>	4,250	13,4
		<i>Medium</i>	5,000	11,8
		<i>Far</i>	5,514	14,2
	<i>Many</i>	<i>Near</i>	4,517	14,4
		<i>Medium</i>	5,276	15,8
		<i>Far</i>	5,514	12,2
Total rata-rata waktu eksekusi (ms)				13,1

Dilihat dari Tabel 5.9, sistem berhasil menghitung prioritas kendaraan darurat pemadam kebakaran dengan rata-rata waktu eksekusi 13,1 ms dengan berbagai kondisi yang disediakan. Untuk hasil penghitungan prioritas, sama dengan uji coba penghitungan prioritas 1, sistem kurang mendapatkan hasil yang optimal karena masih ada beberapa kondisi yang masih sama nilainya meski parameter jumlah korban dan jaraknya berbeda.

5.3.3 Skenario Uji Coba Performa Pengaturan Rute

Pada uji coba pengaturan rute, sistem akan diuji performanya dalam merekayasa lampu lalu lintas pada persimpangan dan mengatur prioritas kendaraan darurat yang akan didahulukan saat ada *collision* di persimpangan. Uji coba akan dilakukan dengan simulasi *database* lampu lalu lintas (tabel traffic_lights) dengan kendaraan darurat (tabel vehicles) pada *server*. Ada dua kondisi khusus yang digunakan dalam skenario uji coba pengaturan rute ini. Yang pertama adalah jumlah kendaraan darurat di sebuah persimpangan dengan jumlah maksimal empat

kendaraan (satu kendaraan di setiap lengan persimpangan). Yang kedua, untuk uji coba jika ada *collision* yang terjadi, rute yang dilalui oleh kendaraan darurat harus melalui persimpangan yang telah ditentukan. Persimpangan yang digunakan adalah persimpangan Jl. Raya Kertajaya Indah, Surabaya dengan Jl. Dr. Ir. H. Soekarno, Surabaya (-7.2805656, 112.7808589).



Gambar 5.2 Lokasi Persimpangan

Di persimpangan tersebut, akan ada empat buah lampu lalu lintas, yaitu di bagian utara, barat, timur, dan selatan persimpangan. Lampu lalu lintas inilah yang nanti akan direkayasa untuk kendaraan darurat yang akan melintas. Detail lokasi lampu lalu lintas dapat dilihat pada Tabel 5.10.

Tabel 5.10 Tabel Lokasi Lampu Lalu Lintas

No. ID	Lokasi	Latitude	Longitude
1	Utara	-7.280207	112.780852
2	Barat	-7.280579	112.780249
3	Selatan	-7.281005	112.780861
4	Timur	-7.280537	112.781344

Ada tiga buah respons *server* saat uji coba pengaturan rute ini. Yang pertama adalah pesan sukses ketika lampu lalu lintas berhasil direkayasa (lampu yang akan dilewati menjadi hijau dan yang lainnya menjadi merah) yang dapat dilihat pada Gambar 5.3.

The screenshot shows the 'Headers' tab selected in the top navigation bar. Below it, there are buttons for 'Pretty', 'Raw', 'Preview', 'HTML' (with a dropdown arrow), and a copy icon. A message box displays the text: "i 1 Sukses! Lampu lalu lintas berhasil direkayasa."

Gambar 5.3 Respons Server Sukses

Lalu yang kedua adalah pesan gagal saat kendaraan belum masuk jarak trigger untuk merekayasa lampu lalu lintas yang akan dilewati yang dapat dilihat pada Gambar 5.4.

The screenshot shows the 'Headers' tab selected in the top navigation bar. Below it, there are buttons for 'Pretty', 'Raw', 'Preview', 'HTML' (with a dropdown arrow), and a copy icon. A message box displays the text: "i 1 Kendaraan belum masuk jarak trigger."

Gambar 5.4 Respons Server Gagal 1

Dan yang terakhir adalah pesan gagal saat ada kendaraan darurat lain yang memiliki prioritas atau jarak yang lebih tinggi yang dapat dilihat pada Gambar 5.5.

The screenshot shows the 'Headers' tab selected in the top navigation bar. Below it, there are buttons for 'Pretty', 'Raw', 'Preview', 'HTML' (with a dropdown arrow), and a copy icon. A message box displays the text: "i 1 Ada kendaraan dengan prioritas lebih tinggi. Mohon tunggu."

Gambar 5.5 Respons Server Gagal 2

Dalam uji coba ini, terdapat 4 macam skenario uji coba, yaitu:

1. Rekayasa lampu lalu lintas saat ada satu kendaraan darurat yang akan melewati persimpangan (tidak ada *collision* yang terjadi).
2. Rekayasa lampu lalu lintas saat ada dua kendaraan darurat yang akan melewati persimpangan (ada *collision* yang terjadi).
3. Rekayasa lampu lalu lintas saat ada tiga kendaraan darurat yang akan melewati persimpangan (ada *collision* yang terjadi).
4. Rekayasa lampu lalu lintas saat ada empat kendaraan darurat yang akan melewati persimpangan (ada *collision* yang terjadi).

5.3.3.1 Skenario Uji Coba Pengaturan Rute 1

Skenario uji coba pengaturan rute 1 adalah pengaturan rute saat ada satu kendaraan darurat yang melewati persimpangan. Kondisi uji coba ada dua, yaitu untuk kendaraan darurat dengan jenis ambulans, dan kendaraan darurat dengan jenis pemadam kebakaran. Uji coba dilakukan sebanyak lima kali untuk tiap kondisi. Hasil dari uji coba yang telah dilakukan dapat dilihat pada Tabel 5.11.

Tabel 5.11 Hasil Uji Coba Pengaturan Rute 1

Kondisi	Jenis Kendaraan	Rekayasa Lampu Lalu Lintas	Rata-rata Waktu (ms)
1	Ambulans	Sukses	188,2
2	Pemadam Kebakaran	Sukses	230,2
Total rata-rata waktu eksekusi (ms)			209,2

Dilihat dari Tabel 5.9, sistem dapat merekayasa lampu lalu lintas yang akan dilalui oleh kendaraan darurat dengan baik. Sistem berhasil merekayasa lampu lalu lintas dengan kecepatan rata-rata eksekusi 209,2 ms. Jadi, dapat dikatakan bahwa performa sistem dalam mengatur rute saat ada satu kendaraan darurat yang melewati sebuah persimpangan sudah bekerja dengan baik untuk dua jenis kendaraan darurat yang diuji coba.

5.3.3.2 Skenario Uji Coba Pengaturan Rute 2

Skenario uji coba pengaturan rute 2 adalah pengaturan rute saat ada dua kendaraan darurat yang melewati persimpangan, sehingga ada *collision* yang harus ditangani. Uji coba dilakukan sebanyak lima kali untuk tiap kondisi. Kondisi pengujian dapat dilihat pada Tabel 5.12.

Tabel 5.12 Tabel Kondisi Pengaturan Rute 2

Kondisi	Keterangan
1	Dua kendaraan darurat dengan jenis sama dan nilai prioritas sama
2	Dua kendaraan darurat dengan jenis sama dan nilai prioritas berbeda
3	Dua kendaraan darurat dengan jenis berbeda dan nilai prioritas sama
4	Dua kendaraan darurat dengan jenis berbeda dan nilai prioritas berbeda

Detail dari kendaraan darurat yang digunakan saat uji coba dapat dilihat pada Tabel 5.13 (Kondisi 1), Tabel 5.14 (Kondisi 2), Tabel 5.15 (Kondisi 3), dan Tabel 5.16 (Kondisi 4).

Tabel 5.13 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 1

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3

Tabel 5.14 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 2

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Ambulans	6	3

Tabel 5.15 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 3

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Pemadam Kebakaran	6	3

Tabel 5.16 Detail Kendaraan Darurat Pengaturan Rute 2 Kondisi 4

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Pemadam Kebakaran	6	3

Dengan menggunakan kendaraan darurat dengan detail seperti diatas, hasil uji coba performa pengaturan rute 2 dapat dilihat pada Tabel 5.17.

Tabel 5.17 Hasil Uji Coba Pengaturan Rute 2

Kondisi	Rekayasa Lampu Lalu Lintas	Prioritas Rekayasa	Rata-rata Waktu (ms)
1	Sukses	1. Kendaraan 2	217,60
		2. Kendaraan 1	254,80
2	Sukses	1. Kendaraan 1	220,80
		2. Kendaraan 2	233,20
3	Sukses	1. Kendaraan 2	208,80
		2. Kendaraan 1	193,00
4	Sukses	1. Kendaraan 1	184,00
		2. Kendaraan 2	190,40
Rata-rata waktu eksekusi (ms)			212,89

Dilihat dari Tabel 5.17, sistem dapat merekayasa lampu lalu lintas yang akan dilalui oleh kendaraan darurat dengan baik. Sistem dapat merekayasa lampu lalu lintas dengan kecepatan rata-rata eksekusi 212,89 ms. Untuk pengaturan rute, sistem berhasil mengatur perekayasaan rute dua kendaraan darurat sesuai dengan nilai prioritas, atau jika nilai prioritas sama, jarak tempuh akan digunakan sebagai penentu. Jadi, dapat dikatakan bahwa performa sistem dalam mengatur rute saat ada dua kendaraan darurat yang melewati sebuah persimpangan sudah bekerja dengan baik.

5.3.3.3 Skenario Uji Coba Pengaturan Rute 3

Skenario uji coba pengaturan rute 3 adalah pengaturan rute saat ada tiga kendaraan darurat yang melewati persimpangan, sehingga ada *collision* yang harus ditangani. Uji coba dilakukan sebanyak lima kali untuk tiap kondisi. Kondisi pengujian dapat dilihat pada Tabel 5.18.

Tabel 5.18 Tabel Kondisi Pengaturan Rute 3

Kondisi	Keterangan
1	Tiga kendaraan darurat dengan jenis sama dengan nilai prioritas sama
2	Dua kendaraan darurat dengan jenis sama, satu kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas sama
3	Tiga kendaraan darurat dengan jenis sama dengan nilai prioritas berbeda
4	Dua kendaraan darurat dengan jenis sama, satu kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas berbeda

Detail dari kendaraan darurat yang digunakan saat uji coba dapat dilihat pada Tabel 5.19 (Kondisi 1), Tabel 5.20 (Kondisi 2), Tabel 5.21 (Kondisi 3), dan Tabel 5.22 (Kondisi 4).

**Tabel 5.19 Detail Kendaraan Darurat Pengaturan Rute 3
Kondisi 1**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3
3	Ambulans	6	5

**Tabel 5.20 Detail Kendaraan Darurat Pengaturan Rute 3
Kondisi 2**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3
3	Pemadam Kebakaran	6	5

Tabel 5.21 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 3

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Ambulans	6	3
3	Ambulans	4	5

Tabel 5.22 Detail Kendaraan Darurat Pengaturan Rute 3 Kondisi 4

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Pemadam Kebakaran	6	3
3	Pemadam Kebakaran	4	5

Dengan menggunakan kendaraan darurat dengan detail seperti diatas, hasil uji coba performa pengaturan rute 3 dapat dilihat pada Tabel 5.23.

Tabel 5.23 Hasil Uji Coba Pengaturan Rute 3

Kondisi	Rekayasa Lampu Lalu Lintas	Prioritas Rekayasa		Rata-rata Waktu (ms)
		1.	Kendaraan 3	
1	Sukses	2.	Kendaraan 2	200,40
		3.	Kendaraan 1	209,80
		1.	Kendaraan 3	206,00
2	Sukses	2.	Kendaraan 2	205,80
		3.	Kendaraan 1	209,80
		1.	Kendaraan 1	242,20
3	Sukses	2.	Kendaraan 2	228,40
		3.	Kendaraan 3	201,60

Kondisi	Rekayasa Lampu Lalu Lintas	Prioritas Rekayasa	Rata-rata Waktu (ms)
4	Sukses	1. Kendaraan 1	204,80
		2. Kendaraan 2	183,40
		3. Kendaraan 3	204,80
Rata-rata waktu eksekusi (ms)			207,08

Dilihat dari Tabel 5.23, sistem dapat merekayasa lampu lalu lintas yang akan dilalui oleh kendaraan darurat dengan baik. Sistem dapat merekayasa lampu lalu lintas dengan kecepatan rata-rata eksekusi 207,08 ms. Untuk pengaturan rute, sistem berhasil mengatur perekayasaan rute tiga kendaraan darurat sesuai dengan nilai prioritas, atau jika nilai prioritas sama, jarak tempuh akan digunakan sebagai penentu. Jadi, dapat dikatakan bahwa performa sistem dalam mengatur rute saat ada tiga kendaraan darurat yang melewati sebuah persimpangan sudah bekerja dengan baik.

5.3.3.4 Skenario Uji Coba Pengaturan Rute 4

Skenario uji coba pengaturan rute 4 adalah pengaturan rute saat ada empat kendaraan darurat yang melewati persimpangan, sehingga ada *collision* yang harus ditangani. Kondisi pengujian dapat dilihat pada Tabel 5.24.

Tabel 5.24 Tabel Kondisi Pengaturan Rute 4

Kondisi	Keterangan
1	Empat kendaraan darurat dengan jenis sama dengan nilai prioritas sama
2	Tiga kendaraan darurat dengan jenis sama, satu kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas sama
3	Dua kendaraan darurat dengan jenis sama, dua kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas sama

Kondisi	Keterangan
4	Empat kendaraan darurat dengan jenis sama dengan nilai prioritas berbeda
5	Tiga kendaraan darurat dengan jenis sama, satu kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas berbeda
6	Dua kendaraan darurat dengan jenis sama, dua kendaraan darurat dengan jenis berbeda, masing-masing dengan nilai prioritas sama

Detail dari kendaraan darurat yang digunakan saat uji coba dapat dilihat pada Tabel 5.25 (Kondisi 1), Tabel 5.26 (Kondisi 2), Tabel 5.27 (Kondisi 3), Tabel 5.28 (Kondisi 4), Tabel 5.29 (Kondisi 5), dan Tabel 5.30 (Kondisi 6).

Tabel 5.25 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 1

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3
3	Ambulans	6	5
4	Ambulans	6	7

Tabel 5.26 Detail Kendaraan Darurat Pengaturan Rute 4 Kondisi 2

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3
3	Ambulans	6	5
4	Pemadam Kebakaran	6	7

**Tabel 5.27 Detail Kendaraan Darurat Pengaturan Rute 4
Kondisi 3**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	6	1
2	Ambulans	6	3
3	Pemadam Kebakaran	6	5
4	Pemadam Kebakaran	6	7

**Tabel 5.28 Detail Kendaraan Darurat Pengaturan Rute 4
Kondisi 4**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Ambulans	6	3
3	Ambulans	4	5
4	Ambulans	8	7

**Tabel 5.29 Detail Kendaraan Darurat Pengaturan Rute 4
Kondisi 5**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Pemadam Kebakaran	6	3
3	Ambulans	4	5
4	Ambulans	8	7

**Tabel 5.30 Detail Kendaraan Darurat Pengaturan Rute 4
Kondisi 6**

No. Kendaraan	Jenis Kendaraan	Nilai Prioritas	Jarak ke Lokasi Insiden (km)
1	Ambulans	7	1
2	Pemadam Kebakaran	6	3
3	Pemadam Kebakaran	4	5
4	Ambulans	8	7

Dengan menggunakan kendaraan darurat dengan detail seperti diatas, hasil uji coba performa pengaturan rute 4 dapat dilihat pada Tabel 5.31.

Tabel 5.31 Hasil Uji Coba Pengaturan Rute 4

Kondisi	Rekayasa Lampu Lalu Lintas	Prioritas Rekayasa	Rata-rata Waktu (ms)
1	Sukses	1. Kendaraan 4	189,20
		2. Kendaraan 3	252,00
		3. Kendaraan 2	282,60
		4. Kendaraan 1	258,80
2	Sukses	1. Kendaraan 4	253,60
		2. Kendaraan 3	294,00
		3. Kendaraan 2	248,20
		4. Kendaraan 1	239,60
3	Sukses	1. Kendaraan 4	249,80
		2. Kendaraan 3	232,60
		3. Kendaraan 2	243,60
		4. Kendaraan 1	248,40
4	Sukses	1. Kendaraan 4	244,80
		2. Kendaraan 1	260,40
		3. Kendaraan 2	253,20
		4. Kendaraan 3	282,00

Kondisi	Rekayasa Lampu Lalu Lintas	Prioritas Rekayasa	Rata-rata Waktu (ms)
5	Sukses	1. Kendaraan 4	289,20
		2. Kendaraan 1	186,00
		3. Kendaraan 2	148,00
		4. Kendaraan 3	222,60
6	Sukses	1. Kendaraan 4	189,40
		2. Kendaraan 1	197,40
		3. Kendaraan 2	176,40
		4. Kendaraan 3	159,40
Rata-rata waktu eksekusi (ms)			233,38

Dilihat dari Tabel 5.31, sistem dapat merekayasa lampu lalu lintas yang akan dilalui oleh kendaraan darurat dengan baik. Sistem dapat merekayasa lampu lalu lintas dengan kecepatan rata-rata eksekusi 233,38 ms. Untuk pengaturan rute, sistem berhasil mengatur perekayasaan rute empat kendaraan darurat sesuai dengan nilai prioritas, atau jika nilai prioritas sama, jarak tempuh akan digunakan sebagai penentu. Jadi, dapat dikatakan bahwa performa sistem dalam mengatur rute saat ada empat kendaraan darurat yang melewati sebuah persimpangan sudah bekerja dengan baik.

5.4 Evaluasi Umum Skenario Uji Coba

Dari serangkaian uji coba yang telah dilakukan, didapatkan bahwa performa sistem secara keseluruhan sudah cukup baik. Untuk performa sistem dalam hal pencarian rute, mulai dari pencarian rumah sakit atau pusat pemadam kebakaran terdekat menggunakan formula Haversine hingga pencarian rute tercepat menggunakan Google Maps Direction API dapat berjalan dengan baik. Kecepatan eksekusi yang didapatkan dari uji coba pencarian rute menunjukkan bahwa sistem dapat mengeksekusi permintaan pencarian rute dengan cepat dengan kecepatan eksekusi rata-rata 19,24 ms dari dua skenario uji coba yang dilakukan.

Untuk performa sistem dalam penghitungan prioritas, sistem masih menunjukkan adanya kekurangan. Nilai prioritas yang didapatkan masih ada yang sama meski kondisi parameter yang dimasukkan sudah berbeda. Hal ini dikarenakan kurang spesifiknya *fuzzy rule* yang diimplementasikan pada *fuzzy control system* di sistem ini. Namun secara kecepatan, dengan kecepatan eksekusi rata-rata 13,385 ms dari dua skenario uji coba yang dilakukan, menunjukkan bahwa *fuzzy control system* dapat menangani penghitungan nilai prioritas dengan cepat.

Untuk performa sistem dalam pengaturan rute, dari empat skenario uji coba yang dilakukan, sistem dapat mengatur prioritas perekayasaan rute dan lampu lalu lintas untuk satu hingga empat kendaraan darurat di sebuah persimpangan dengan baik. Kecepatan eksekusi rata-rata sebesar 215,638 ms yang didapatkan dari uji coba pengaturan rute menunjukkan bahwa sistem dapat mengeksekusi permintaan pengaturan rute berdasarkan prioritas dan rekayasa lampu lalu lintas pada persimpangan dengan cepat.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

6.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil uji coba sistem cerdas pengaturan rute kendaraan darurat menggunakan metode *fuzzy* adalah sebagai berikut:

1. Sistem dapat mencari rute tercepat untuk kendaraan darurat menggunakan Google Maps Directions API dari informasi lokasi insiden yang dilaporkan oleh *user* dengan kecepatan waktu eksekusi rata-rata 19,24 ms.
2. Metode *fuzzy* dalam bentuk *fuzzy control system* sebagai metode pengaturan prioritas kendaraan darurat dapat menghitung nilai prioritas kendaraan darurat untuk mengatasi masalah *collision* yang terjadi di sebuah persimpangan dengan kecepatan waktu eksekusi rata-rata 13,385 ms.
3. Manajemen rekayasa lampu lalu lintas di persimpangan yang akan dilalui kendaraan darurat dapat diatur menggunakan nilai prioritas dan jarak tempuh kendaraan darurat dengan kecepatan waktu eksekusi rata-rata 215,638 ms.

6.2 Saran

Berikut meruapakan beberapa saran untuk pengembangan perangkat lunak di masa yang akan datang, saran diberikan terkait pengembangan pada Tugas Akhir ini adalah:

1. Meningkatkan ketepatan penghitungan nilai prioritas kendaraan darurat dengan *fuzzy rule* yang lebih spesifik jika menggunakan *fuzzy control system*.
2. Dapat mencoba menambahkan manajemen rekayasa lampu lalu lintas di lebih dari satu persimpangan.

DAFTAR PUSTAKA

- [1] K. Athavan, G. Balasubramanian, S. Jagadeeshwaran, and N. Dinesh, “Automatic Ambulance Rescue System,” in *2012 Second International Conference on Advanced Computing Communication Technologies*, 2012, pp. 190–195.
- [2] R. Tri Yulianto, *Implementasi Smart Traffic Light: Pengaturan Lalu Lintas Untuk Jalur Kendaraan Darurat Berbasis Android dan Mikrokontroler Arduino*. Jurusan Teknik Informatika-ITS, 2015.
- [3] A. Chowdhury, “Priority based and secured traffic management system for emergency vehicle using IoT,” in *2016 International Conference on Engineering MIS (ICEMIS)*, 2016, pp. 1–6.
- [4] “DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL,” Jul. 2010.
- [5] M. Godoy Simoes, “Introduction to Fuzzy Control.”
- [6] C. Veness, “Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript.” [Online]. Available: <https://www.movable-type.co.uk/scripts/latlong.html>. [Accessed: 10-Dec-2017].
- [7] “PHP: Hypertext Preprocessor.” [Online]. Available: <http://php.net/>. [Accessed: 10-Dec-2017].
- [8] “MySQL.” [Online]. Available: <https://www.mysql.com/>. [Accessed: 10-Dec-2017].
- [9] “JSON.” [Online]. Available: <http://www.json.org/>. [Accessed: 10-Dec-2017].
- [10] “Google Maps API,” *Google Developers*. [Online]. Available: <https://developers.google.com/maps/?hl=id>. [Accessed: 10-Dec-2017].
- [11] D. I. Heywood, S. Cornelius, and S. Carver, *An introduction to geographical information systems*, 2nd ed. Harlow, England ; New York: Prentice Hall, 2002.
- [12] “Python.org,” *Python.org*. [Online]. Available: <https://www.python.org/>. [Accessed: 10-Dec-2017].

- [13]“NumPy and SciPy Documentation — NumPy and SciPy documentation.” [Online]. Available: <https://docs.scipy.org/doc/>. [Accessed: 10-Dec-2017].
- [14]“skfuzzy 0.2 docs — skfuzzy v0.2 docs.” [Online]. Available: <http://pythonhosted.org/scikit-fuzzy/>. [Accessed: 10-Dec-2017].
- [15]“Android,” *Android*. [Online]. Available: https://www.android.com/intl/id_id/. [Accessed: 10-Dec-2017].
- [16]“Keputusan Menteri Tenaga Kerja R.I. No. KEP-186/MEN/1999 tentang Unit Penanggulangan Kebakaran di Tempat Kerja.” Menteri Tenaga Kerja Republik Indonesia, 1999.
- [17]“Keputusan Menteri Kesehatan dan Kesejahteraan Sosial Republik Indonesia Nomor 143/MENKES-KESOS/SK/II/2001 tentang Standarisasi Kendaraan Pelayanan Medik.” Menteri Kesehatan dan Kesejahteraan Sosial, 23-Feb-2001.
- [18]“Keputusan Walikota Surabaya Nomor 56 Tahun 2001 tentang Rincian Tugas dan Fungsi Dinas Pemadam Kebakaran Kota Surabaya.” Walikota Surabaya, 2001.

LAMPIRAN

Pseudocode 7.1 Kode Program Penyimpanan Data Insiden untuk Ambulans

```
1  FUNCTION getNearestService(incident_lat,
2    incident_long):
3      sql := "SELECT *, (6371 *
4          ACOS(COS(RADIANS("incident_lat")) *
5          COS(RADIANS(services_lat)) *
6          COS(RADIANS(services_long) -
7          RADIANS("incident_long")) +
8          SIN(RADIANS("incident_lat")) *
9          SIN(RADIANS(services_lat )))) AS distance FROM
10         emergency_services WHERE services_type =
11         'Pemadam Kebakaran' ORDER BY distance ASC LIMIT
12         1"
13
14      qry := mysqli_query(conn, sql)
15      res := mysqli_fetch_array(qry, MYSQLI_ASSOC)
16      nearest := res2['services_id']
17      dist := res2['distance']
18      RETURN nearest, dist
19
ENDFUNCTION
20
21
22 FUNCTION storeIncidentData():
23     IF $_SERVER['REQUEST_METHOD']=='POST':
24         incident_type := $_POST['incidentType']
25         issue := $_POST['issue']
26         destination := $_POST['destination']
27         victims := $_POST['victims']
28         note := $_POST['note']
29         v_lat := $_POST['v_lat']
30         v_long := $_POST['v_long']
31
32             formattedAddr := str_replace(' ', '+',
33 destination)
34             geocode :=
35             file_get_contents('https://maps.googleapis.com/m
36 aps/api/geocode/json?address='formattedAddr'&sen
37 sor=false&key=AIzaSyA4dGRJa069TEilJJh8TACUGMhKJK
38 wzEkU')
39             output := json_decode(geocode)
```

```

21           incident_lat := $output->results[0]-
>geometry->location->lat
22           incident_long := $output->results[0]-
>geometry->location->lng

23           getNearestService(incident_lat,
incident_long)

24           IF incident_type == "Penanganan
Jenazah":
25               IF issue == "Korban sudah
meninggal":
26                   threat := 0
ENDIF
ELSE IF issue == "Korban luka ringan
- sedang":
27                   threat := 2
ENDIF
ELSE IF issue == "Korban kritis":
28                   threat := 3
ENDIF
ENDIF
ELSE IF incident_type == "Transportasi
pasien":
29                   threat := 1
ENDIF
ELSE IF issue == "Korban luka ringan
- sedang":
30                   threat := 3
ENDIF
ELSE IF issue == "Korban kritis":
31                   threat := 4
ENDIF
ENDIF
ENDIF
IF incident_type == "Kondisi gawat
darurat":
32               IF issue == "Korban sudah
meninggal":
33                   threat := 2
ENDIF
ELSE IF issue == "Korban luka ringan
- sedang":
34                   threat := 2
ENDIF

```

```

49             threat := 4
50         ENDIF
51         ELSE IF issue == "Korban kritis":
52             threat := 5
53         ENDIF
54     ENDIF

55         priority = exec("python
priority_fuzzy.py threat dist victims")

56             //Store incident data to db
57             sql := ""INSERT INTO am_incident_data
(am_incident_id, am_incident_type,
am_medical_issue, am_incident_address,
am_incident_lat, am_incident_long,
am_nearest_service, am_incident_victims,
am_distance, am_priority, am_note) VALUES
(incident_id,
'incident_type','issue','destination',
'incident_lat', 'incident_long', nearest,
victims, dist, priority, 'note')")
58             qry := mysqli_query(conn, sql)

59             //Get nearest service address
60             sql2 := "SELECT * FROM
emergency_services WHERE services_id = nearest"
61                 qry2 := mysqli_query(conn, sql4)
62                 res2 := mysqli_fetch_array(qry2,
MYSQLI_ASSOC)
63                 service_address :=
res2['services_address']

64             //Store vehicle data to db
65             sql3 := "INSERT INTO vehicles
(vehicle_id, vehicle_lat, vehicle_long,
vehicle_priority, vehicle_active,
vehicle_distance) VALUES (v_id, 'v_lat',
've_long', priority, 1, dist)"
66             qry3 := mysqli_query(conn, sql3)

67             IF sql && sql3:
68                 message := "Success"
69                 result := array(message,
service_address);

```

70	echo implode("#", result)
71	ENDIF
72	ELSE
73	echo "Something went wrong. Please
try again."	
74	ENDIF
75	ENDIF
76	ELSE
77	echo "Connection Error.;"
78	ENDIF
79	ENDFUNCTION

Pseudocode 7.2 Kode Program Penyimpanan Data Insiden untuk Pemadam Kebakaran

```

1  FUNCTION getNearestService(incident_lat,
2    incident_long):
3      sql := "SELECT *, (6371 *
4        ACOS(COS(RADIANS("incident_lat")) *
5          COS(RADIANS(services_lat)) *
6          COS(RADIANS(services_long) -
7            RADIANS("incident_long")) +
8            SIN(RADIANS("incident_lat")) *
9            SIN(RADIANS(services_lat )))) AS distance FROM
10           emergency_services WHERE services_type =
11             'Pemadam Kebakaran' ORDER BY distance ASC LIMIT
12               1"
13
14      qry := mysqli_query(conn, sql)
15      res := mysqli_fetch_array(qry, MYSQLI_ASSOC)
16      nearest := res2['services_id']
17      dist := res2['distance']
18      RETURN nearest, dist
19
ENDFUNCTION
20
21 FUNCTION storeIncidentData():
22   IF $_SERVER['REQUEST_METHOD']=='POST':
23     incident_type := $_POST['incidentType']
24     issue := $_POST['issue']
25     destination := $_POST['destination']
26     victims := $_POST['victims']
27     note := $_POST['note']
28     v_lat := $_POST['v_lat']
29     v_long := $_POST['v_long']

30     formattedAddr := str_replace(' ', '+',
31 destination)
32     geocode :=
33     file_get_contents('https://maps.googleapis.com/m
34     aps/api/geocode/json?address='formattedAddr'&sen
35     sor=false&key=AIzaSyA4dGRJa069TEilJJh8TACUGMhKJK
36     wzEkU')
37     output := json_decode(geocode)

38     incident_lat := $output->results[0]-
39     >geometry->location->lat

```

```
22         incident_long := $output->results[0]->geometry->location->lng
23
24         getNearestService(incident_lat,
25             incident_long)
26
27         IF incident_type == "Pemadaman
28             kebakaran":
29                 IF issue == "Tidak ada bahaya
30                 kebakaran":
31                     threat := 2
32                     ENDIF
33                     ELSE IF issue == "Bahaya kebakaran
34             ringan":
35                     threat := 3
36                     ENDIF
37                     ELSE IF issue == "Bahaya kebakaran
38             sedang":
39                     threat := 4
40                     ENDIF
41                     ELSE IF issue == "Bahaya kebakaran
42             berat":
43                     threat := 5
44                     ENDIF
45                     ENDIF
46                     ELSE IF incident_type == "Penyelamatan
47             jiwa":
48                         IF issue == "Tidak ada bahaya
49                 kebakaran":
50                     threat := 2
51                     ENDIF
52                     ELSE IF issue == "Bahaya kebakaran
53             ringan":
54                     threat := 3
55                     ENDIF
56                     ELSE IF issue == "Bahaya kebakaran
57             sedang":
58                     threat := 4
59                     ENDIF
60                     ELSE IF issue == "Bahaya kebakaran
61             berat":
62                     threat := 5
63                     ENDIF
64                     ENDIF
```

```

51         priority := exec("python
priority_fuzzy.py threat dist victims")

52             //Store incident data to db
53             sql := "INSERT INTO pb_incident_data
(pb_incident_id, pb_incident_type,
pb_fire_threat, pb_incident_address,
pb_incident_lat, pb_incident_long,
pb_nearest_service, pb_incident_victims,
pb_distance, pb_priority, pb_note) VALUES
(incident_id,
'incident_type','issue','destination',
'incident_lat', 'incident_long', nearest,
victims, dist, priority, 'note')"
54             qry := mysqli_query(conn, sql)

55                 //Get nearest service address
56                 sql2 := "SELECT * FROM
emergency_services WHERE services_id = nearest"
57                     qry2 := mysqli_query(conn, sql4)
58                     res2 := mysqli_fetch_array(qry2,
MYSQLI_ASSOC)
59                         service_address :=
res2['services_address']

60                 //Store vehicle data to db
61                 sql3 := "INSERT INTO vehicles
(vehicle_id, vehicle_lat, vehicle_long,
vehicle_priority, vehicle_active,
vehicle_distance) VALUES (v_id, 'v_lat',
'venue', priority, 1, dist)"
62                     qry3 := mysqli_query(conn, sql3)

63             IF sql && sql3:
64                 message := "Success"
65                 result := array(message,
service_address);
66                     echo implode("#", result)
67             ENDIF
68             ELSE
69                 echo "Something went wrong. Please
try again."
70             ENDIF

```

71	ENDIF
72	ELSE
73	echo "Connection Error.";
74	ENDIF
75	ENDFUNCTION

Pseudocode 7.3 Kode Program Pencarian Rute

```

1   CLASS DirectionFinder:
2       CLASS DownloadRawData extends
3           AsyncTask<String, Void, String>:
4               FUNCTION doInBackground(params):
5                   link := params[0]
6                   try:
7                       url := new URL(link)
8                       is := url.openConnection().
9                           getInputStream()
10                      buffer := new StringBuffer()
11                      reader := new BufferedReader(new
12                          InputStreamReader(is))
13                      while (line := reader.readLine() != null):
14                          buffer.append(line + "\n")
15                      RETURN buffer.toString()
16                      catch MalformedURLException e:
17                          e.printStackTrace()
18                      catch IOException e:
19                          e.printStackTrace()
20                      RETURN null;
21                  ENDFUNCTION
22                  FUNCTION onPostExecute(res):
23                      try:
24                          parseJSON(res)
25                      catch JSONException e:
26                          e.printStackTrace()
27                  ENDFUNCTION
28
29                  FUNCTION parseJSON(data):
30                      IF data == null:
31                          RETURN
32                      routes := new ArrayList<Route>()
33                      jsonData := new JSONObject(data)
34                      jsonRoutes :=
35                      jsonData.getJSONArray("routes")
36                      for i in jsonRoutes.length():
37                          jsonRoute :=
38                          jsonRoutes.getJSONObject(i)
39                          route := new Route()

```

```

34             overview_polylineJson :=
35         jsonRoute.getJSONObject("overview_polyline")
36             jsonLegs :=
37             jsonRoute.getJSONArray("legs")
38                 jsonLeg := jsonLegs.getJSONObject(0)
39                     jsonDistance :=
40                     jsonLeg.getJSONObject("distance")
41                         jsonDuration :=
42                         jsonLeg.getJSONObject("duration")
43                             jsonEndLocation :=
44                             jsonLeg.getJSONObject("end_location")
45                             jsonStartLocation :=
46                             jsonLeg.getJSONObject("start_location")
47                                 route.distance := new
48                                     Distance(jsonDistance.getString("text"),
49                                         jsonDistance.getInt("value"))
50                                         route.duration := new
51                                             Duration(jsonDuration.getString("text"),
52                                                 jsonDuration.getInt("value"))
53                                                 route.endAddress :=
54                                                 jsonLeg.getString("end_address")
55                                                 route.startAddress :=
56                                                 jsonLeg.getString("start_address")
57                                                 route.startLocation := new
58                                                 LatLng(jsonStartLocation.getDouble("lat"),
59                                                     jsonStartLocation.getDouble("lng"))
60                                                 route.endLocation := new
61                                                 LatLng(jsonEndLocation.getDouble("lat"),
62                                                     jsonEndLocation.getDouble("lng"))
63                                                 route.points :=
64                                                 decodePolyLine(overview_polylineJson.getString("points"))
65                                         routes.add(route)
66             ENDFOR
67             RETURN routes
68         ENDFUNCTION

69
70         FUNCTION decodePolyLine(poly):
71             len := poly.length()
72             index := 0
73             decoded := new ArrayList<LatLng>()
74             lat := 0
75             lng := 0
76             while index < len:
77                 ...
78             ...
79         ENDFUNCTION

```

```
59         shift := 0
60         result := 0
61         do:
62             b := poly.charAt(index++) - 63
63             result |= (b & 0x1f) << shift
64             shift += 5
65             while b >= 0x20
66                 dlat := ((result & 1) != 0 ? ~(result >>
1) : (result >> 1))
67                 lat += dlat
68                 shift := 0
69                 result := 0
70                 do:
71                     b := poly.charAt(index++) - 63
72                     result |= (b & 0x1f) << shift
73                     shift += 5
74                     while b >= 0x20
75                         dlng := ((result & 1) != 0 ? ~(result >>
1) : (result >> 1))
76                         lng += dlng
77                         decoded.add(new LatLng(
78                             lat / 100000d, lng / 100000d
79                         ))
80             RETURN decoded
81         ENDFUNCTION
```

Pseudocode 7.4 Kode Program Inisialisasi *Membership Function*

```

1  FUNCTION initMF():
2      threat := ctrl.Antecedent(np.arange(0, 6,
3          1), 'threat level')
4      distance := ctrl.Antecedent(np.arange(0, 11,
5          1), 'distance')
6      victims := ctrl.Antecedent(np.arange(0, 51,
7          1), 'victims')
8      priority := ctrl.Consequent(np.arange(0, 11,
9          1), 'priority')
10
11         threat['low'] := fuzz.trimf(threat.universe,
12             [0, 0, 1])
13         threat['medium'] :=
14             fuzz.trimf(threat.universe, [0, 1, 4])
15         threat['high'] :=
16             fuzz.trimf(threat.universe, [2, 5, 5])
17
18         distance['near'] :=
19             fuzz.trimf(distance.universe, [0, 0, 3])
20         distance['medium'] :=
21             fuzz.trimf(distance.universe, [0, 2, 5])
22         distance['far'] :=
23             fuzz.trimf(distance.universe, [4, 10, 10])
24
25         victims['few'] :=
26             fuzz.trimf(victims.universe, [0, 0, 10])
27         victims['medium'] :=
28             fuzz.trimf(victims.universe, [0, 10, 25])
29         victims['many'] :=
30             fuzz.trimf(victims.universe, [20, 50, 50])
31
32         priority['low'] :=
33             fuzz.trimf(priority.universe, [0, 0, 5])
34         priority['medium'] :=
35             fuzz.trimf(priority.universe, [0, 5, 10])
36         priority['high'] :=
37             fuzz.trimf(priority.universe, [5, 10, 10])
38
39     RETURN
40 ENDFUNCTION

```

Pseudocode 7.5 Kode Program Pengaturan Rute

```

1   FUNCTION haversine(latitudeFrom, longitudeFrom,
2   latitudeTo, longitudeTo, earthRadius):
3       latFrom := deg2rad(latitudeFrom)
4       longFrom := deg2rad(longitudeFrom)
5       latTo := deg2rad(latitudeTo)
6       longTo := deg2rad(longitudeTo)
7       latDelta := latTo - latFrom
8       longDelta := longTo - longFrom
9       angle = 2 * asin(sqrt(pow(sin(latDelta/2),2)
+ cos(latFrom) * cos(latTo) *
pow(sin(longDelta/2),2)))
10      RETURN angle * earthRadius
11
ENDFUNCTION

12  FUNCTION updateTL(v_id, tl_id):
13      sql := "UPDATE traffic_lights SET
14      tl_status='green' WHERE tl_id=tl_id"
15      qry := mysqli_query(conn, sql)
16      sql2 := "UPDATE traffic_lights SET
17      tl_status='red' WHERE tl_id<>tl_id"
18      qry2 := mysqli_query(conn, sql2)

19      sql3 := "UPDATE vehicles SET
20      vehicle_active=0 WHERE vehicle_id=v_id"
21      qry3 := mysqli_query(conn, sql3)
22      RETURN
ENDFUNCTION

23  FUNCTION routeControl():
24      v_id := $_POST['v_id']
25      tl_id := $_POST['tl_id']

26      distance := haversine(v_lat, v_long, tl_lat,
27      tl_long, 6371000)
28      IF distance <= 100 && v_active == 1:
29          sql := "SELECT * FROM vehicles WHERE
30          vehicle_active=1"
31          qry := mysqli_query(conn, sql)

32          IF mysqli_num_rows(qry) > 1:

```

```

29          sql2 := "SELECT
MAX(vehicle_priority) AS max_priority FROM
vehicles WHERE vehicle_active=1";
30          qry2 := mysqli_query(conn, sql2)
31          res2 := mysqli_fetch_array(qry2,
MYSQLI_ASSOC)
32          max_priority = res2['max_priority']

33          sql3 := "SELECT * FROM vehicles
WHERE vehicle_priority=max_priority AND
vehicle_active=1"
34          qry3 := mysqli_query(conn, sql3)
35          IF (mysqli_num_rows(qry3) > 1:
36              sql4 := "SELECT * FROM vehicles
WHERE vehicle_active=1 ORDER BY vehicle_distance
DESC LIMIT 1"
37          qry4 := mysqli_query(conn,
sql4);
38          res4 := mysqli_fetch_array(qry4,
MYSQLI_ASSOC)
39          max_id := res4['vehicle_id']
40          IF max_id == v_id:
41              updateTL(v_id, tl_id)
42              echo "Sukses! Lampu lalu
lintas berhasil direkayasa."
43          ENDIF
44          ELSE
45              echo "Ada kendaraan dengan
prioritas lebih tinggi. Mohon tunggu."
46          RETURN
47      ENDIF
48      ENDIF
49      ELSE
50          res3 := mysqli_fetch_array(qry3,
MYSQLI_ASSOC)
51          max_id := res3['vehicle_id']
52          IF max_id == v_id:
53              updateTL(v_id, tl_id)
54              echo " Sukses! Lampu lalu
lintas berhasil direkayasa."
55          ENDIF
56          ELSE
57              echo " Ada kendaraan dengan
prioritas lebih tinggi. Mohon tunggu."

```

58	RETURN
59	ENDIF
60	ENDIF
61	ENDIF
62	ELSE
63	updateTL(v_id, tl_id)
64	echo " Sukses! Lampu lalu lintas
	berhasil direkayasa."
65	ENDIF
66	ENDIF
67	ELSE
68	echo "Kendaraan belum masuk jarak
	trigger."
69	RETURN
70	ENDIF
71	ENDFUNCTION

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Muhammad Faiez Ananda merupakan anak dari pasangan Bapak Amril Hasan dan Ibu Yustina. Lahir di Malang pada tanggal 30 April 1995. Penulis menempuh pendidikan formal dimulai dari SDN Purwantoro 8 Malang (2001-2007), SMPN 3 Malang (2007-2010), SMAN 3 Malang (2010-2013) dan S1 Teknik Informatika ITS (2013-2018). Penulis aktif dalam organisasi seperti Himpunan Mahasiswa Teknik Computer-Informatika (2014-2016). Penulis juga aktif dalam berbagai kegiatan kepanitiaan yaitu SCHEMATICS 2014 divisi REEVA, SCHEMATICS 2015 divisi NST, ITS EXPO 2014 sebagai staff Pagelaran Seni, dan ITS EXPO 2015 juga sebagai staff Pagelaran Seni. Penulis juga pernah kerja praktik di PT. Bahana Pembinaan Usaha Indonesia divisi *Information Technology* (IT) periode Juli – Agustus 2016. Penulis dapat dihubungi melalui email: faiez1995@gmail.com.