TUGAS AKHIR - KI141502

APLIKASI SCORING PERILAKU PENGEMUDI MENGGUNAKAN SENSOR FUSI

ANINDITA LARASATI
NRP 05111440000170

Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Dosen Pembimbing II
Sarwosri, S.Kom., M.T.

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

TUGAS AKHIR – KI141502

APLIKASI SCORING PERILAKU PENGEMUDI MENGGUNAKAN SENSOR FUSI

ANINDITA LARASATI
NRP 05111440000170

Dosen Pembimbing I
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Dosen Pembimbing II
Sarwosri, S.Kom., M.T.

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

(Halaman ini sengaja dikosongkan)

UNDERGRADUATE THESES – KI141502

DRIVER BEHAVIOR SCORING APPLICATION USING FUSION SENSOR

ANINDITA LARASATI
NRP 05111440000170

First Advisor
Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc.

Second Advisor
Sarwosri, S.Kom., M.T.

Department of Informatics
Faculty of Information Technology and Communication
Sepuluh Nopember Institute of Technology
Surabaya 2018

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

APLIKASI SCORING PERILAKU PENGEMUDI MENGGUNAKAN SENSOR FUSI
TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
ANINDITA LARASATI
NRP: 05111440000170

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.tech.Ir. Raden Venantius Hari Ginardi, M.Sc. .....
   (NIP. 19650518 199203 1 003) (Pembimbing 1)

2. Sarwosri, S.Kom., M.T. .....
   (NIP. 19760809 200112 2 001) (Pembimbing 2)

SURABAYA
JULI, 2018

(Halaman ini sengaja dikosongkan)

APLIKASI SCORING PERILAKU PENGEMUDI MENGGUNAKAN SENSOR FUSI

Nama Mahasiswa : Anindita Larasati
NRP : 05111440000170
Departemen : Informatika FTIK-ITS
Dosen Pembimbing 1: Dr.tech. Ir. R.V.Hari Ginardi, M.Sc.
Dosen Pembimbing 2: Sarwosri, S.Kom., M.T.

Abstrak

Perilaku pengemudi merupakan faktor tertinggi penyebab kecelakaan lalu lintas. Oleh karena itu, berbagai penelitian dilakukan untuk mengurangi tingkat kecelakaan yang terjadi akibat kelalaian pengemudi, salah satunya adalah aplikasi untuk mendeteksi perilaku pengemudi dengan menggunakan sensor. Akan tetapi, perilaku yang diamati belum dapat diketahui apakah perilaku pengemudi tersebut baik atau buruk.

Tugas akhir ini mengusulkan pengembangan aplikasi scoring perilaku pengemudi melalui pergerakan mobil. Sistem ini menggunakan metode fusion sensor dari tiga sensor yang terdapat pada smartphone android meliputi Gyroscope, Accelerometer, dan Magnetometer. Data yang didapat oleh sensor-sensor tersebut selanjutnya diolah sehingga mendapatkan penilaian terhadap perilaku pengemudi. Perilaku yang diamati meliputi tindakan belok mendadak, penambahan kecepatan mendadak, dan pengurangan kecepatan mendadak. Pengmudi juga dapat melihat apakah manuver yang dilakukan normal atau agresif. Setelah mendeteksi data perilaku pengemudi, aplikasi akan mengeluarkan hasil penilaian berupa angka nol hingga seratus.

Uji coba dilakukan dalam berbagai kondisi berbeda yaitu kondisi jalanan, mobil, dan mobile phone yang digunakan. Berdasarkan hasil uji coba yang dilakukan, diketahui bahwa aplikasi dapat melihat perilaku pengemudi dengan stabil.

Kata kunci: Accelerometer, Fusion Sensor, Gyroscope, Magnetometer, Scoring Pengemudi

DRIVER BEHAVIOR SCORING APPLICATION USING FUSION SENSOR

Student's Name	**:**	**Anindita Larasati**
Student's ID	**:**	**05111440000170**
Department	**:**	**Department of Informatics FTIK-ITS**
First Advisor	**:**	**Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.**
Second Advisor	**:**	**Sarwosri, S.Kom., M.T.**

Abstract

Driver Behavior is one of the highest factor that cause traffic accidents. Therefore, various studies have been conducted to reduce the level of accidents that occur due to driver negligence, one of which is the applications to detect the driver's behaviour by using sensor. However, the behaviour that has been obtained can not be known wether the driver behaviour is good or bad.

This Final Project propose the development of scoring applications for driver behaviour through car movement. This system uses the fusion sensor method of three sensors found in the android smartphone including Gyroscope, Accelerometer, and Magnetometer. The data obtained by the sensors are further process to get an assesment of the driver behaviour. Observed behaviour include sudden turnover, sudden speed acceleration, and sudden break. Driver can also see wether the manuver are normal or aggresive. After detecting the Driver's Behaviour data, the application will issu the assesment result in the form of zero to one hundred.

The experimental is conducted in variety of different conditions namely road conditions, cars, and mobile phones. Based on the experiments, it is known that the applications can see the driver's behaviour constantly stable.

Keyword: Gyroscope, Accelerometer, Magnetometer, Fusion Sensor, Driver Behavior

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yesus Kristus atas pimpinan, penyertaan, dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

"Aplikasi Scoring Perilaku Pengemudi Menggunakan Sensor Fusi"

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Dr.tech. Ir. R.V. Hari Ginardi, M.Sc. selaku dosen pembimbing yang telah memberikan masukan dan bimbingan kepada penulis selama pengerjaan Tugas Akhir ini.
2. Ibu Sarwosri, S.Kom., M.T. selaku dosen pembimbing yang telah memberikan masukan dan bimbingan kepada penulis selama pengerjaan Tugas Akhir ini.
3. Bapak Kombes Pol Hery Sasongko selaku Kakorlantas Mabes Polri yang telah membantu penulis dalam pemberian data untuk tugas akhir ini.
4. Keluarga penulis yang selalu memberikan dukungan, dan perhatian bagi penulis.
5. Seluruh tenaga pengajar dan karyawan Departemen Informatika ITS yang telah memberikan tenaga dan waktunya demi kelancaran proses belajar mengajar penulis selama perkuliahan.
6. Teman-teman administrator Laboratorium Pemrograman (Rina, Afiif, Sekbay, Hari, Nafiar, Nobby, Rizka,

Bonbon, Brian, Raca, Yuuta, John, Michael) yang memberikan semangat dan hiburan bagi penulis selama perkuliahan.

7. Prasetyo Nugrohadi yang telah menemani penulis dan sangat membantu penulis dalam pengerjaan Tugas Akhir ini.
8. Muhammad Hilman, Tiara Anggita, Rahmatin Nadia, Aviananda Dwirahma, Kania Amalia, Anne Annisa, dan Vinsensia Sipriana Zega yang selama ini memberikan semangat dalam hal studi maupun masalah lainnya.
9. Teman-teman angkatan 2014 Departemen Informatika ITS yang telah menemani perjuangan penulis selama masa perkuliahan.
10. Mas Daniel, mas Romen, mas uul, mas harrypotret, mbak rona, mas razi, mas wawan, mas dewangga, mas udin, mbak ine, mas wicak, dan kakak-kakak yang telah membantu penulis selama masa perkuliahan.
11. Seluruh anggota ITSTV yang memberikan perhatiannya kepada penulis.
12. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

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

Surabaya, Mei 2018

Anindita Larasati

DAFTAR ISI

LEMBAR PENGESAHAN ..... **v**
Abstrak ..... **vii**
Abstract ..... **ix**
KATA PENGANTAR ..... **xi**
DAFTAR ISI ..... **xiii**
DAFTAR GAMBAR ..... **xvii**
DAFTAR TABEL ..... **xix**
DAFTAR KODE PROGRAM ..... **xxi**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Batasan Permasalahan 4
1.4 Tujuan 5
1.5 Manfaat 5
1.6 Metodologi 5
1.6.1 Penyusunan Proposal Tugas Akhir 5
1.6.2 Studi Literatur 5
1.6.3 Implementasi Perangkat Lunak 6
1.6.4 Pengujian dan Evaluasi 6
1.6.5 Penyusunan Buku 6
1.7 Sistematika Penulisan Laporan 6
BAB II DASAR TEORI **7**
2.1 Deskripsi Umum Permasalahan 7
2.2 Pengemudi 7
2.3 Perilaku 7
2.4 Scoring 9
2.5 Riset Sejenis 10
2.5.1 Survei tentang Sensor Fusion 10
2.5.2 MIROAD 11
2.6 Android 12
2.7 Deskripsi Umum Sensor yang Digunakan 13
2.7.1 Gyro Sensor 13

2.7.2 Accelerometer 15
2.7.3 Magnetometer 18
2.8 Google Maps 19
2.9 Advance Driver Assistance System 21
2.10 Coefficient of Variation 25
BAB III ANALISIS DAN PERANCANGAN SISTEM 27
3.1 Analisis Sistem 27
3.1.1 Deskripsi Umum 27
3.1.2 Analisis Kebutuhan Sistem 28
3.1.3 Analisis Aktor 29
3.2 Perancangan Sistem 30
3.2.1 Kasus Penggunaan 30
3.2.1.1 Melihat Score Pengemudi 31
3.2.1.2 Melihat rekam jejak pengemudi 34
3.2.1.3 Mendeteksi Rotasi Perangkat 37
3.2.1.4 Mengukur percepatan 38
3.2.1.5 Mengukur magnetisasi dan arah medan magnet 40
3.2.2 Perancangan Antarmuka 42
3.2.2.1 Rancangan Antarmuka Halaman Utama 42
3.2.2.2 Rancangan Antarmuka Halaman Tujuan 43
3.2.2.3 Rancangan Antarmuka Halaman Proses 45
3.2.2.4 Rancangan Antarmuka Halaman Skor Pengemudi 46
BAB IV IMPLEMENTASI 49
4.1 Lingkungan Implementasi 49
4.1.1 Lingkungan Implementasi Perangkat Keras 49
4.1.2 Lingkungan Implementasi Perangkat Lunak 49
4.2 Implementasi Tampilan Antarmuka 49
4.2.1 Implementasi Halaman Utama 50
4.2.2 Implementasi Halaman Tujuan 50
4.2.3 Implementasi Halaman Status Pegemudi 53
4.2.4 Implementasi Halaman Skor Pengemudi 54

4.3 Implementasi Perangkat Lunak 55
4.3.1 Implementasi Proses Melihat Status Sensor 55
4.3.2 Implementasi Proses Memilih Tempat Tujuan 56
4.3.3 Implementasi Proses Menampilkan Informasi Perjalanan yang akan Ditempuh 57
4.3.4 Implementasi Proses Pengambilan data dari Sensor 59
4.3.5 Implementasi Proses Pengolahan Data 63
4.3.6 Implementasi Proses Menampilkan Skor Pengemudi 64
4.4 Langkah Penilaian Perilaku 65
BAB V UJI COBA DAN EVALUASI 69
5.1 Lingkungan Pengujian 69
5.2 Skenario Pengujian 69
5.2.1 Pengujian Fungsionalitas 70
5.2.2 Pengujian Perjalanan dengan Kondisi Lalu Lintas Padat 70
5.2.3 Pengujian Perjalanan dengan Kondisi Lalu Lintas Sepi 70
5.2.4 Pengujian dengan Mobil Berbeda 70
5.2.5 Pengujian dengan Handphone Berbeda 71
5.2.6 Pengujian dengan Mobil dan Handphone Berbeda 71
5.2.7 Pengujian dengan User Berbeda 71
5.3 Evaluasi Pengujian 71
5.3.1 Evaluasi Pengujian Fungsionalitas 72
5.3.2 Evaluasi Pengujian Perjalanan dengan Kondisi Lalu Lintas Padat 72
5.3.3 Evaluasi Pengujian Perjalanan dengan Kondisi Lalu Lintas Sepi 74
5.3.4 Evaluasi Pengujian dengan Mobil Berbeda 76
5.3.5 Evaluasi Pengujian dengan Handphone Berbeda ... 78
5.3.6 Evaluasi Pengujian Perjalanan dengan Mobil dan Handphone yang Berbeda 79

5.3.7 Evaluasi Pengujian Perjalanan User Berbeda 82
BAB VI KESIMPULAN DAN SARAN 85
6.1 Kesimpulan 85
6.2 Saran 86
DAFTAR PUSTAKA 87
LAMPIRAN 91
BIODATA PENULIS 93

DAFTAR GAMBAR

Gambar 1.1 Grafik fatalitas kecelakaan pada tahun 2016-2017 dari Korlantas Polri 1
Gambar 1.2 Grafik kecelakaan berdasarkan jenis kendaraan pada tahun 2016-2017 dari Korlantas Polri..... 2
Gambar 1.3 Faktor penyebab kecelakaan lalu lintas dan angkutan jalan tahun 2010-2016 dari Komisi Nasional Keselamatan Transportasi..... 3
Gambar 2.1 Hasil Foto Google 360 14
Gambar 2.2 Visualisasi Planetarium pada Google Sky View 15
Gambar 2.3 Visualisasi Implementasi Akselerometer pada Aplikasi Pokemon-Go 17
Gambar 2.4 Perbedaan GPS dan IPS 18
Gambar 2.5 Tampilan Google Maps pada Handphone 19
Gambar 2.6 Implementasi API Google pada Go-Jek..... 20
Gambar 2.7 Implementasi API Google pada Foursquare 21
Gambar 2.8 Tampilan Aplikasi Kruzr 24
Gambar 3.1 Diagram Alur Aplikasi 27
Gambar 3.2 Use Case pada Pengemudi 29
Gambar 3.3 Use Case pada Sensor Gyroscope 30
Gambar 3.4 Use Case pada Sensor Accelerometer 30
Gambar 3.5 Use Case pada Sensor Magnetometer 30
Gambar 3.6 Diagram Aktivitas Melihat Score Pengemudi 33
Gambar 3.7 Sequence diagram melihat score pengemudi 33
Gambar 3.8 Diagram Aktivitas Melihat Status Perjalanan Normal 35
Gambar 3.9 Sequence diagram melihat status perjalanan normal 35
Gambar 3.10 Diagram Aktivitas Melihat status perjalanan agresif 36
Gambar 3.11 Sequence Diagram Melihat Status Perjalanan Agresif 37
Gambar 3.12 Diagram Aktivitas Mendeteksi Rotasi Perangkat.. 38
Gambar 3.13 Sequence Diagram Sensor Gyro 38

Gambar 3.14 Diagram Aktivitas Mengukur Kecepatan 39
Gambar 3.15 Sequence Diagram Mengukur Percepatan 40
Gambar 3.16 Diagram Aktivitas Mengukur Magnetisasi dan Arah Medan Magnet 41
Gambar 3.17 Sequence Diagram Mengukur Magnetisasi dan Arah Medan Magnet 41
Gambar 3.18 Perancang UI Halaman Utama 42
Gambar 3.19 Perancangan UI Halaman Tujuan (1) 43
Gambar 3.20 Perancangan UI Halaman Tujuan (2) 44
Gambar 3.21 Perancangan UI Halaman Tujuan (3) 45
Gambar 3.22 Perancangan UI Halaman Proses 46
Gambar 3.23 Perancangan UI Halaman Skor Pengemudi 47
Gambar 4.1 UI Halaman Utama 50
Gambar 4.2 UI Halaman Tujuan (1) 51
Gambar 4.3 UI Halaman Tujuan (2) 51
Gambar 4.4 UI Halaman Tujuan (3) 52
Gambar 4.5 UI Halaman Tujuan (4) 52
Gambar 4.6 UI Halaman Tujuan (5) 53
Gambar 4.7 UI Halaman Status Pengemudi 54
Gambar 4.8 UI Halaman Skor Pengemudi 54
Gambar 4.9 Visualisasi Nilai pada masing-masing Sensor 65
Gambar 4.10 Contoh Hasil Tampilan Banyaknya Manuver pada Aplikasi 66
Gambar 4.11 Contoh Hasil Tampilan Skor pada Aplikasi 68
Gambar 5.1 Rute Pengujian Perjalanan dengan Kondisi Padat ... 73
Gambar 5.2 Rute Pengujian Perjalanan dengan Kondisi Sepi 75
Gambar 5.3 Rute Pengujian Perjalanan Global Seviila School - SMPN 99 Jakarta 77

DAFTAR TABEL

Tabel 1.1 Data jumlah kendaraan di Indonesia berdasarkan tipe kendaraan dari Badan Pusat Statistik tahun 2013-2016 3
Tabel 2.1 Data Laka Berdasarkan Perilaku Pengemudi Tahun 2013-2017 (1)..... 8
Tabel 2.2 Data Laka Berdasarkan Perilaku Pengemudi Tahun 2013-2017 (2)..... 9
Tabel 2.3 Bobot Faktor..... 9
Tabel 2.4 Variasi penggunaan sensor dan alat bantu lainnya untuk mendeteksi perilaku pengemudi 11
Tabel 2.5 Perbandingan SI Skoring Pengemudi dengan MIROAD 11
Tabel 2.6 Perubahan Nilai pada Gyroscope saat terjadi Rotasi Perangkat.....13
Tabel 2.7 Perubahan Nilai pada Sensor Akselerometer.....16
Tabel 2.8 Fitur Advance Driving Assistance System 22
Tabel 2.9 Penggunaan Perangkat pada Implementasi ADAS 23
Tabel 2.10 Perusahaan Otomotif yang Menggunakan ADAS 24
Tabel 3.1 Kebutuhan Fungsional.....28
Tabel 3.2 Kasus Penggunaan.....31
Tabel 3.3 Melihat Score Pengemudi.....31
Tabel 3.4 Melihat status perjalanan normal.....34
Tabel 3.5 Melihat status perjalanan agresif 36
Tabel 3.6 Mendeteksi rotasi perangkat.....37
Tabel 3.7 Melihat status perjalanan normal dan agresif 39
Tabel 3.8 Diagram Aktivitas Mengukur magnetisasi dan arah medan magnet 40
Tabel 4.1 Batasan Hasil Balikan Manuver 65
Tabel 4.2 Contoh Data Hasil Manuver pada Perjalanan 66
Tabel 5.1 Evaluasi Pengujian Fungsionalitas 72
Tabel 5.2 Hasil Uji Perjalanan dengan Kondisi Lalulintas Padat73
Tabel 5.3 Transformasi Hasil Uji Perjalanan dengan Kondisi Lalulintas Padat.....74
Tabel 5.4 Hasil Uji Perjalanan dengan Kondisi Lalulintas Sepi .75

Tabel 5.5 Transformasi Hasil Uji Perjalanan dengan Kondisi Lalulintas Sepi 76
Tabel 5.6 Hasil Uji Perjalanan dengan Mobil Berbeda 77
Tabel 5.7 Transformasi Hasil Uji Perjalanan dengan Mobil Berbeda..... 78
Tabel 5.8 Hasil Uji Perjalanan dengan Handphone Berbeda..... 78
Tabel 5.9 Transformasi Hasil Uji Perjalanan dengan Handphone Berbeda..... 79
Tabel 5.10 Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda 80
Tabel 5.11 Transformasi Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda (1) 80
Tabel 5.12 Transformasi Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda (2) 81
Tabel 5.13 Hasil Uji Perjalanan dengan User Berbeda 82

DAFTAR KODE PROGRAM

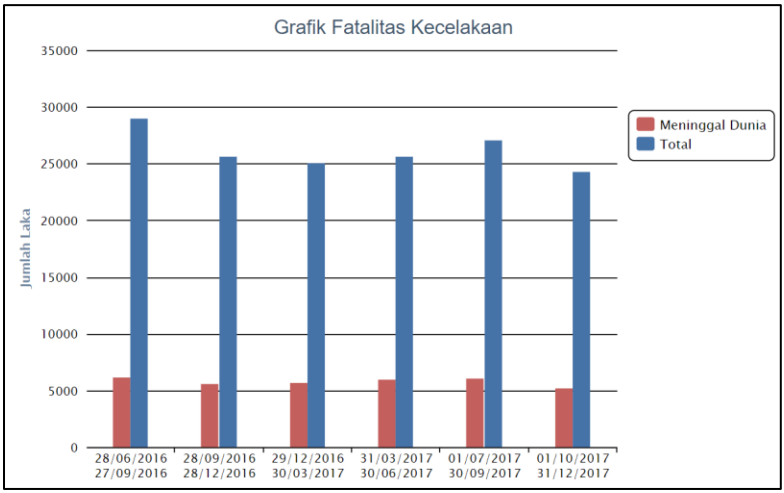
Kode Sumber 2.1 Penyaringan Gaya Gravitasi pada Hasil Pengukuran Sensor Akselerometer 16
Kode Sumber 4.1 Fungsi Melihat Status Sensor 56
Kode Sumber 4.2 Fungsi Proses Menampilkan Place Picker 57
Kode Sumber 4.3 Fungsi Proses Menerima Feedback dari Place Picker 57
Kode Sumber 4.4 Fungsi Mendeteksi Lokasi Saat Ini dan Mengambil Data Lokasi Tujuan dari API Google Maps 58
Kode Sumber 4.5 Fungsi Menampilkan Informasi Perjalanan yang akan Ditempuh..... 59
Kode Sumber 4.6 Fungsi Inisiasi Sensor 61
Kode Sumber 4.7 Fungsi Pengambilan Data Sensor 63
Kode Sumber 4.8 Fungsi Pengolahan Data 64
Kode Sumber 4.9 Fungsi Menampilan Skor Pengemudi 65

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

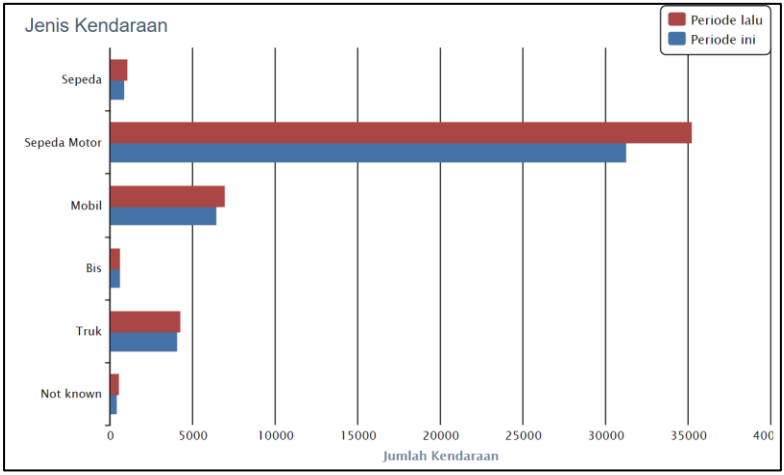
1.1 Latar Belakang

Alat transportasi sudah menjadi salah satu kebutuhan bagi manusia disamping sandang, pangan dan papan. Kendaraan merupakan kebutuhan yang penting karena hal ini akan sangat membantu manusia dalam beraktifitas. Tanpa alat transportasi, tentunya akan banyak sekali kegiatan penting yang terhambat bahkan tidak dapat kita lakukan. Ada berbagai macam jenis transportasi yang diciptakan oleh manusia, mulai dari transportasi laut, darat dan juga udara.



Gambar 1.1 Grafik fatalitas kecelakaan pada tahun 2016-2017 dari Korlantas Polri

Berdasarkan Gambar 1.1 dari Korlantas Polri, rata-kecelakaan yang terjadi di Indonesia per triwulan adalah 27.000 kejadian dengan 6000 kejadian mengakibatkan korban meninggal dunia.



Gambar 1.2 Grafik kecelakaan berdasarkan jenis kendaraan pada tahun 2016-2017 dari Korlantas Polri

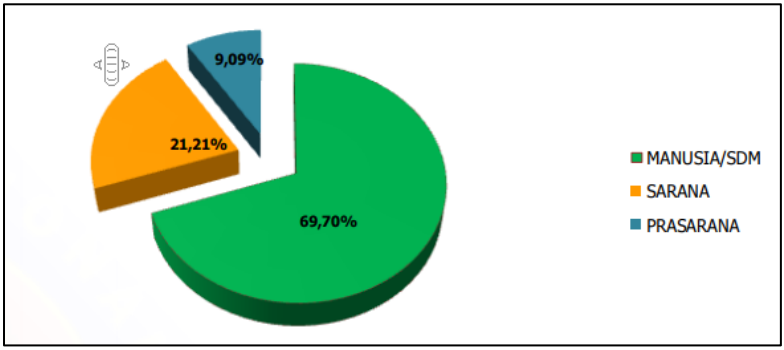
Gambar 1.2 dari Korlantas Polri menunjukkan bahwa kecelakaan tertinggi disebabkan oleh sepeda motor dengan rata-rata 37.000 kejadian, sedangkan roda empat 11.000 kejadian.

Berdasarkan Tabel 1.1 dari Badan Pusat Statistik, jumlah total kendaraan roda empat yang ada di Indonesia adalah 105.150.082, sedangkan roda empat 24.130.997. Dari data jumlah kendaraan dan jumlah kecelakaan, didapatkan bahwa rasio terjadinya kecelakaan roda empat lebih tinggi dengan 0.45% dan roda dua adalah 0.35%.

Tabel 1.1 Data jumlah kendaraan di Indonesia berdasarkan tipe kendaraan dari Badan Pusat Statistik tahun 2013-2016

Jenis Kendaraan Bermotor	2013	2014	2015	2016
Mobil Penumpang	11 484 514	12 599 038	13 480 973	14 580 666
Mobil Bis	2 286 309	2 398 846	2 420 917	2 486 898
Mobil Barang	5 615 494	6 235 136	6 611 028	7 063 433
Sepeda motor	84 732 652	92 976 240	98 881 267	105 150 082
Jumlah	104 118 969	114 209 260	121 394 185	129 281 079

Berdasarkan Gambar 1.3 dari Komisi Nasional Keselamatan Transportasi, faktor kecelakaan tertinggi disebabkan oleh kesalahan manusia dengan 69.7%, bukan sarana maupun prasana.



Gambar 1.3 Faktor penyebab kecelakaan lalu lintas dan angkutan jalan tahun 2010-2016 dari Komisi Nasional Keselamatan Transportasi

Saat ini sudah mulai dikembangan berbagai cara untuk mengantisipasi kecelakaan mobil yang disebabkan oleh perilaku pengemudi. Terdapat setidaknya 8 paper [1] yang membuat aplikasi untuk mengamati perilaku pengemudi lewat sensor dan berbagai perangkat lainnya. Salah satu aplikasi untuk mendeteksi perilaku pengemudi adalah dengan menggabungkan sensor *handphone* yaitu *MIROAD*. *MIROAD* menggunakan *fusion sensor* untuk mengetahui

banyaknya manuver yang mendadak maupun tidak ketika berkendara. Namun, *MIROAD* hanya memungkinkan pengguna untuk melihat manuver yang terjadi, tanpa bisa menilai apakah perilaku mengemudi tersebut baik atau buruk.

Untuk menilai perilaku pengemudi tersebut baik atau buruk, diperlukan aplikasi *scoring* perilaku yang menilai perilaku pengemudi lewat pergerakan mobil. Beberapa kegiatan yang dapat diamati adalah menambah kecepatan, mengurangi kecepatan, belok, serta ketepatan pengemudi untuk mencapai tujuan. Kegiatan tersebut dapat diamati dengan sensor Android [2]. Dalam tugas akhir ini, akan digunakan *fusion sensor* dari tiga sensor yang terdapat pada *smartphone* Android. Setelah kegiatan tersebut terdeteksi, dilakukan klasifikasi dengan dua kategori yaitu normal dan agresif. Setelah itu, penghitungan *score* pengemudi dapat dilakukan untuk melihat apakah perilaku pengemudi tersebut baik atau buruk.

1.2 Rumusan Masalah

Perumusan masalah yang terdapat pada tugas akhir ini, antara lain adalah :

1. Apa saja faktor yang mempengaruhi penilaian perilaku pengemudi yang dapat diamati?
2. Bagaimana mengklasifikasikan data perilaku pengemudi yang diolah dari sensor sebagai perilaku yang normal dan agresif?

1.3 Batasan Permasalahan

Permasalahan yang diangkat pada tugas akhir ini memiliki beberapa batasan :

1. Aplikasi hanya digunakan untuk memberikan hasil pengukuran berkendara dari satu kali perjalanan.
2. Aplikasi hanya dapat digunakan pada smartphone android.
3. Sensor yang digunakan hanya sensor yang tersedia pada smartphone yaitu sensor gyro, accelerometer, dan magnetometer.
4. Perilaku pengemudi yang diamati adalah belok secara normal dan agresif, menambah dan mengurangi kecepatan secara

normal dan agresif, serta ketepatan waktu yang dibandingkan dengan google maps.

1.4 Tujuan

Tujuan pembuatan tugas akhir ini yaitu melakukan analisis dan mendesain sistem pengukuran kualitas perilaku berkendara untuk menjawab permasalahan kecelakaan lalu lintas dan efisiensi waktu yang diakibatkan oleh perilaku pengemudi yang kurang baik.

1.5 Manfaat

Tugas akhir ini mampu memberikan informasi nilai seorang pengemudi dari perilakunya saat berkendara. Aplikasi ini diharapkan dapat meminimalisir angka kecelakaan lalu lintas oleh pengendara yang kurang baik. Selain itu, aplikasi ini juga dapat membantu penyedia jasa transportasi untuk menilai pengemudinya dengan lebih akurat.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahap pertama dalam proses pengerjaan tugas akhir ini adalah menyusun proposal tugas akhir. Pada proposal tugas akhir ini diajukan beberapa metode dan *tools* yang cocok untuk menyelesaikan masalah dari tugas akhir ini.

1.6.2 Studi Literatur

Pada tahapan ini dilakukan pengumpulan kebutuhan yang diperlukan guna mendukung pengerjaan tugas akhir. Referensi yang digunakan yaitu berupa hasil penelitian yang sudah pernah dilakukan, buku, artikel internet, atau sumber lain yang bisa dipertanggungjawabkan. Sejumlah referensi yang diperlukan dalam pembuatan sistem ini yaitu mengenai Android, Gyro sensor,

Accelerometer, magnetometer, Google maps API dan juga beberapa paper terkait tugas akhir ini.

1.6.3 Implementasi Perangkat Lunak

Pembuatan aplikasi ini menggunakan bahasa pemrograman Java dengan desain yang menggunakan tools Android Studio.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini akan dilakukan ujicoba langsung kepada pengemudi taksi online dan pengendara lain yang berbeda-beda. Pengujian dilakukan dari tempat asal ke tempat tujuan dengan kondisi yang bervariasi yaitu perilaku pengemudi dan kondisi jalan yang berbeda-beda. Media yang digunakan untuk testing adalah aplikasi android yaitu smartphone yang sudah diinstall aplikasi yang dibuat sebagai tugas akhir ini.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku yang menjelaskan seluruh konsep, teori dasar dari metode yang digunakan, implementasi, serta hasil yang telah dikerjakan sebagai dokumentasi dari pelaksanaan Tugas Akhir.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
2. Bab II. Tinjauan Pustaka
3. Bab III. Perancangan Perangkat Lunak
4. Bab IV. Implementasi
5. Bab V. Hasil Uji Coba dan Evaluasi
6. Bab VI. Kesimpulan dan Saran
7. Daftar Pustaka
8. Lampiran

BAB II
DASAR TEORI

Bab ini berisi pembahasan mengenai deskripsi umum permasalahan dan teori-teori dasar atau penjelasan dari metode, algoritma, *library* dan *tools* yang digunakan dalam Tugas Akhir.

2.1 Deskripsi Umum Permasalahan

Permasalahan yang diangkat pada Tugas Akhir ini adalah mengurangi terjadinya kecelakaan lalu lintas pada transportasi darat, khususnya penggunaan mobil. Seperti yang telah dijelaskan pada subbab 1.1, faktor terbesar penyebab kecelakaan adalah kesalahan pengemudi itu sendiri sehingga perlu dilakukan pencarian solusi untuk mengurangi kesalahan-kesalahan dalam mengemudi.

2.2 Pengemudi

Pengemudi atau dalam Bahasa Inggris *driver* adalah orang yang mengemudikan atau melakukan kegiatan berkendara. Pengemudi wajib mengikuti tata cara berlalu lintas untuk mengemudikan kendaraan. Saat ini untuk mengurangi potensi kecelakaan, pengemudi wajib mempunyai Surat Izin Mengemudi (SIM) sebagai bukti kelayakan untuk mengemudi. Dalam aplikasi ini, pengemudi bertindak sebagai aktor yang akan dinilai perilakunya saat berkendara.

2.3 Perilaku

Perilaku adalah tindakan atau aktivitas dari manusia itu sendiri yang mempunyai bentangan yang sangat luas. Menurut seorang ilmuan Amerika Robert Y. Kwick yang diambil dari Ensiklopedi Amerika, perilaku diartikan sebagai suatu aksi dan reaksi organisme terhadap lingkungannya, hal ini berarti bahwa perilaku baru akan terwujud bila ada sesuatu yang diperlukan untuk menimbulkan tanggapan yang disebut rangsangan, dengan demikian maka suatu rangsangan tertentu akan menghasilkan perilaku tertentu

pula. Dalam TA ini, perilaku yang dimaksud adalah perilaku pengemudi saat berkendara yang akan mempengaruhi laku lintas.

Menurut data yang diperoleh dari Korlantas POLRI pada tahun 2013-2017, terdapat 23 faktor perilaku pengemudi yang dapat menyebabkan kecelakaan lalu lintas. Faktor-faktor tersebut diuraikan pada Tabel 2.1.

Tabel 2.1 Data Laka Berdasarkan Perilaku Pengemudi Tahun 2013-2017 (1)

NO.	URAIAN	2013	2014	2015	2016	2017
12	PERILAKU PENGEMUDI					
	a. MELAKUKAN AKTIFITAS LAIN	1,245	1,732	1,983	4,041	4,749
	b. SALAH MEMBERI ISYARAT	318	347	353	631	816
	c. GAGAL MEMBERI TANDA	1,756	2,386	2,671	3,463	3,893
	d. TERTIDUR / KELELAHAN	1,259	1,197	1,081	1,353	1,239
	e. MENGGUNAKAN GADGET	80	64	58	76	86
	f. BERHENTI MENDADAK	517	378	439	558	513
	g. MEMDADAK MERUBAH KECEPATAN	1,571	1,215	1,058	1,669	1,388
	h. CEROBOH SAAT MENYALIP	13,214	11,871	11,518	12,867	11,159
	i. TIDAK WASPADA AKAN LALALINTAS DARI DEPAN	20,791	21,651	23,404	25,709	24,555
	j. CEROBOH SAAT BELOK	13,404	12,627	13,255	15,473	14,428
	k. LALAI SAAT MUNDUR	262	173	216	221	217
	l. MENYALIP DI TIKUNGAN	355	317	226	271	227
	m. SALAH POSISI PARKIR	726	661	623	764	726
	n. MENGABAIKAN APPIL	835	782	800	894	787
	o. MENGABAIKAN ATURAN LAJUR	2,949	3,196	3,226	3,614	3,708
	p. MELAMPAUI BATAS KECEPATAN	8,217	6,952	6,583	7,918	8,686
	q. CEROBOH TERHADAP ATURAN LAJUR	5,016	4,642	4,724	5,129	4,577
	r. MENGABAIKAN POLISI	50	56	55	45	60
	s. MEMOTONG SETELAH MENYALIP	212	201	210	212	223
	t. MENGABAIKAN RAMBU DAN MARKA	1,019	1,177	1,431	1,439	1,370
	u. GAGAL MENJAGA JARAK AMAN	10,741	13,552	16,370	19,729	18,303
	v. MENGABAIKAN HAK JALUR PEJALAN KAKI	6,707	6,177	6,881	7,603	6,726
	w. DATA TIDAK DIKETAHUI	7,519	12	40	118	165
	JUMLAH	98,763	91,366	97,205	113,797	108,601

*) UPDATE DATA LAKA IRSMS S.D 30 JANUARI 2018 JAM 01:00 WIB

Dari Tabel 2.1 yang didapat dari Korlantas POLRI, terdapat beberapa faktor perilaku pengemudi yang dapat diamati secara teknis pergerakan mobil dengan fusion sensor. Kelima perilaku tersebut dapat dilihat pada Tabel 2.2.

Tabel 2.2 Data Laka Berdasarkan Perilaku Pengemudi Tahun 2013-2017 (2)

PERILAKU PENGEMUDI	2013	2014	2015	2016	2017	JUMLAH
Berhenti Mendadak	517	378	439	558	513	2405
Mendadak Merubah Kecepatan	1571	1215	1058	1669	1388	6901
Ceroboh Saat Menyalip	13214	11871	11518	12867	11159	60629
Ceroboh Saat Belok	13404	12627	13225	15473	14428	69157
Memotong Setelah Menyalip	212	201	210	212	223	1058
					JUMLAH	140150

2.4 Scoring

Pengukuran menurut Nunnally & Berbstein (1994) dapat didefinisikan sebagai suatu proses pemberian angka atau label terhadap atribut dengan aturan-aturan yang terstandar atau yang telah disepakati untuk mempresentasikan atribut yang diukur.

Di dalam TA ini, *Scoring* dilakukan untuk mengetahui seberapa baik perilaku pengemudi dalam mengemudi. Berdasarkan data yang diambil dari Tabel 2.2 dari Korlantas POLRI, apabila dihitung maka dapat diamati bahwa bobot setiap faktor yang akan diamati adalah sebagai berikut :

Tabel 2.3 Bobot Faktor

PERILAKU PENGEMUDI	BOBOT
Berhenti Mendadak	1,716018552
Mendadak Merubah Kecepatan	5,678915448
Ceroboh Saat Menyalip	43,26007849
Ceroboh Saat Belok	49,34498751

Dengan demikian, maka sistem penilaian yang akan dilakukan dirumuskan sebagai berikut :

$$manuver_i = \frac{normal_manuver_i}{(normal_manuver_i + sudden_manuver_i)}$$

$$score = 100 * \sum_{i=0}^{n} (manuver_i * bobot_i)$$

Penilaian dapat dilakukan dengan menjumlahkan semua manuver yang dikalikan dengan bobotnya. Data yang dianalisis akan mengeluarkan hasil *score* dengan nilai antara angka 0-100.

2.5 Riset Sejenis

Sebelum merancang dan mengimplementasikan sistem, peninjauan riset serupa dilakukan guna menentukan standardisasi dan referensi dalam pembuatan aplikasi *Scoring* Perilaku Pengemudi.

2.5.1 Survei tentang Sensor Fusion

Pada [1], dilakukan survei pada beberapa paper dan jurnal mengenai penggunaan sensor perangkat bergerak untuk mendeteksi perilaku pengemudi. Didapatkan 8 publikasi yang berhasil mendeteksi perilaku pengemudi dengan cara yang berbeda. Menggabungkan beberapa sensor untuk mendapatkan hasil yang terbaik merupakan salah satu cara yang dilakukan dalam memperoleh hasil yang maksimal dalam mendeteksi manuver yang dilakukan oleh pengemudi. Selain menggunakan sensor, beberapa percobaan yang dilakukan juga menggunakan peralatan lainnya.

Perbedaan penggunaan sensor dan peralatan bantu lainnya dalam mendeteksi perilaku pengemudi dapat dilihat pada Tabel 2.4.

Tabel 2.4 Variasi penggunaan sensor dan alat bantu lainnya untuk mendeteksi perilaku pengemudi

Penulis	Penggunaan Sensor dan Alat Lainnya
P. Singh	Accelerometer, GPS, microphone
Fazeen	Accelerometer
Chigurupa	Accelerometer, GPS, video recorder
Johnson	Accelerometer, gyroscope, magnetometer, GPS, video recorder
Dai	Accelerometer, orientation sensor
Zhang	Vehicle simulator
Sathyanarayana	-
Gazali	Orientation sensor, GPS

2.5.2 MIROAD

Salah satu penerapan *sensor fusion* yang menghasilkan aplikasi adalah [3]. Aplikasi yang dibangun dengan menerapkan penggabungan accelerometer, gyroscope, magnetometer, GPS, video recorder ini memberikan hasil yang akurat. Aplikasi ini bernama *MIROAD : A Mobile-Sensor-Platform for Intelligent Recognition Of Aggressive Driving*. *MIROAD* dibuat dengan tujuan meminimalisir angka kecelakaan yang disebabkan oleh perilaku pengemudi yang agresif. *MIROAD* merupakan aplikasi perangkat bergerak iOS, sehingga sangat mudah mobilitasnya. Perbandingan Aplikasi *Scoring* Perilaku Pengemudi dengan *MIROAD* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Perbandingan SI Skoring Pengemudi dengan MIROAD

	SI Skoring Perilaku Pengemudi	**MIROAD**
Output	Skor pengemudi	Perilaku pengemudi (agresif / tidak agresif)
Basis	Android	iOS
Sensor	Accelerometer, Magnetometer, Gyroscope, GPS	Accelerometer, gyroscope, magnetometer, GPS,

	SI Skoring Perilaku Pengemudi	MIROAD
		video recorder
Fitur	Menghitung maneuver yang dilakukan pengemudi, Menghitung ketepatan waktu pengemudi, Menghitung skor pengemudi	Menghitung maneuver yang dilakukan pengemudi, Merekam maneuver berbahaya, merekam audio ketika pengemudi melakukan maneuver berbahaya
Deteksi manuver	1. belok kiri 2. belok kiri mendadak 3. belok kanan 4. belok kanan mendadak 5. menambah kecepatan 6. menambah kecepatan mendadak 7. mengurangi kecepatan 8. Mengurangi kecepatan mendadak	1. belok kiri 2. belok kiri mendadak 3. belok kanan 4. belok kanan mendadak 5. putar balik 6. putar balik mendadak 7. pindah jalur kanan mendadak 8. pindah jalur kiri mendadak

2.6 Android

Android adalah sistem operasi berbasis Linux. Sistem operasi yang bekerja pada layar sentuh seperti smartphone ini dikembangkan oleh Android, Inc. Google membeli android sejak tahun 2005 dan telah mengembangkannya hingga sekarang. Kini inovasi android oleh google banyak bermunculan seperti android TV, Android Auto, Android Wear dan masih banyak lagi. Sumber aplikasi inipun opensource dan dirilis dengan lisensi Apache. Sehingga Android memungkinkan untuk dimodifikasi dan didistribusikan secara bebas oleh pembuat aplikasi dan pengembang. Didalam smartphone android sendiri terdapat fitur-fitur yang dapat dimanfaatkan untuk *scoring* perilaku pengemudi.

2.7 Deskripsi Umum Sensor yang Digunakan

Aplikasi *Scoring* Perilaku Pengemudi menggunakan tiga sensor yang terdapat pada Android. Sensor yang digunakan adalah Gyro sensor, Accelerometer, dan Magnetometer. Ketiga sensor tersebut digunakan untuk mendapatkan data yang akan diolah. *Fusion sensor* dilakukan untuk meningkatkan kinerja aplikasi.

2.7.1 Gyro Sensor

Gyro sensor atau gyroscope adalah sensor yang bekerja untuk mengukur atau mempertahankan orientasi dengan ketetapan momentum sudut [4]. Gyro sensor dapat mendeteksi gerakan pengguna sesuai gerakan gravitasi. Sensor ini mempunyai output dari sumbu x, y dan z. yang masing-masing adalah pergerakan ke kanan dan kiri, atas dan bawah, serta ke depan dan ke belakang. Sensor ini juga terdapat pada hamper seluruh perangkat android. Pada android, sensor ini mengukur nilai rotasi pada 3 sumbu perangkat dalam satuan rad/s [5].

Gyroscope menggunakan standar koordinat sistem sensor, sehingga perubahan-perubahan nilai yang terukur ketika perangkat berotasi dari kedaan awal. Perubahan nilai pada rotasi tersebut merupakan standar definisi matematika pada rotasi positif dan tidak sama dengan definisi yang digunakan pada sensor orientasi. Perubahan nilai pada gyroscope saat terjadi rotasi pada perangkat dapat dilihat pada Tabel 2.6 dari Android Developers.

Tabel 2.6 Perubahan Nilai pada Gyroscope saat terjadi Rotasi Perangkat

Rotasi	Perubahan Nilai
Searah jarum jam	Nilai sumbu x,y,z negatif
Berlawanan jarum jam	Nilai sumbu x,y,z positif

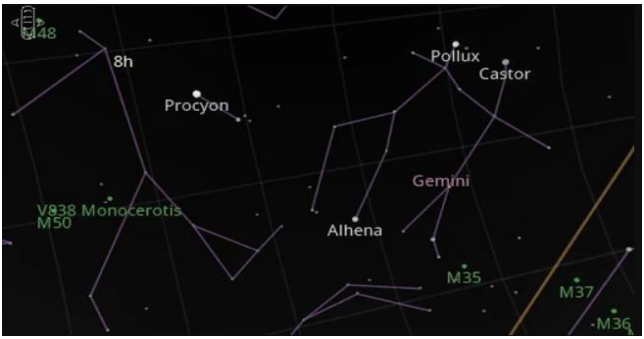
Pada umumnya, penggunaan gyroscope diproses dengan interval waktu tertentu untuk melihat nilai rotasi yang terjadi. Gyroscope pada android memberikan data mentah yang belum dilakukan penyaringan, sehingga biasanya akan ada sedikit

kesalahan yang harus dikompensasi. Akan tetapi, kekurangan dari gyroscope tersebut dapat dihilangkan dengan menggabungkan penggunaan gyroscope dengan sensor lain, seperti sensor gravitasi dan akselerometer.

Contoh penggunaan dari gyroscope pada masa lalu adalah untuk kontrol rudal, sedangkan pada saat ini sudah banyak aplikasi yang menggunakan gyroscope untuk menunjang fungsionalnya, salah satunya adalah Google 360° dan Google Sky Map.

Gambar 2.1 Hasil Foto Google 360

Implementasi gyroscope pada Google 360 (Gambar 2.1) adalah untuk mendeteksi pengambilan gambar dalam satu putaran penuh. Gyroscope akan mendeteksi berapa derajat perangkat sudah bergerak untuk mengambil data hingga perangkat berotasi 360 derajat. Implementasi gyroscope lainnya ada pada Google Sky Map. Aplikasi ini menerapkan gyroscope untuk menelusuri planetarium yang ditampilkan (Gambar 2.2).



Gambar 2.2 Visualisasi Planetarium pada Google Sky View

2.7.2 Accelerometer

Akselerometer adalah sensor yang dapat digunakan untuk mendeteksi orientasi suatu perangkat berdasarkan gerakan ke segala arah atau dengan menggoyangkan yang memungkinkan fitur untuk bertindak [4]. Sensor yang juga terdapat pada android ini mengukur percepatan suatu objek baik secara dinamik maupun statistik. Yang akan digunakan dalam tugas akhir ini adalah pengukuran dinamik.

Sensor akselerometer pada android digunakan untuk mengukur percepatan bahwa perangkat bergerak relatif dalam tiga sumbu, yaitu sumbu x, sumbu y dan sumbu z [5]. Dalam perangkat android terdapat dua macam akselerometer, yaitu linear akselerometer dan akselerometer. Perbedaan kedua sensor tersebut adalah pada pengukuran dan kalibrasi data. Data hasil pengukuran pada linear akselerometer tidak termasuk dengan gaya gravitasi dan belum dikalibrasi sedangkan pada akselerometer sebaliknya.

Sensor akselerometer mengukur percepatan yang dialami oleh perangkat dengan menghitung gaya yang dialami oleh sensor itu sendiri, seperti pada Persamaan 1 dan Persamaan 2 dari Android Developers.

$$A_D = -\left(\frac{1}{mass}\right)\sum F_S$$

Persamaan 1 Pengukuran Percepatan pada Sensor Akselerometer

$$A_D = -g - \left(\frac{1}{mass}\right)\sum F_S$$

Persamaan 2 Pengukuran Percepatan yang dipengaruhi Gaya Gravitasi pada Sensor Akselerometer

Persamaan 1 adalah persamaan untuk pengukuran percepatan secara umum, akan tetapi gaya gravitasi selalu memengaruhi hasil pengukuran sehingga persamaan yang terukur pada sensor di perangkat adalah Persamaan 2. Sebagai contoh, ketika perangkat diam pada suatu tempat maka percepatan yang diukur adalah $g = 9.81\, m/s^2$ sedangkan ketika perangkat jatuh maka percepatan yang terukur adalah $g = 0\, m/s^2$, sehingga untuk mengukur percepatan yang sebenarnya diperlukan penyaringan gaya gravitasi. Penyaringan gaya gravitasi pada sensor pada android dapat dilakukan dengan menerapkan Kode Sumber 2.1 dari Android Developers.

```
1  accelerometer[0] = event.values[0] - SensorManager.GRAVITY_EARTH;
2  accelerometer[1] = event.values[1] - SensorManager.GRAVITY_EARTH;
3  accelerometer[2] = event.values[2] - SensorManager.GRAVITY_EARTH;
```

Kode Sumber 2.1 Penyaringan Gaya Gravitasi pada Hasil Pengukuran Sensor Akselerometer

Akselerometer menggunakan standar koordinat sistem sensor, sehingga perubahan-perubahan nilai yang terukur ketika akselerometer bergerak dapat dilihat pada Tabel 2.7 dari Android Developers.

Tabel 2.7 Perubahan Nilai pada Sensor Akselerometer

Gerakan	Hasil Pengukuran
Perangkat bergerak ke kanan	Sumbu X bernilai positif
Perangkat bergerak ke kiri	Sumbu X bernilai negatif
Perangkat bergerak menjauh	Sumbu Y bernilai positif
Perangkat bergerak mendekat	Sumbu Y bernilai negative

Gerakan	Hasil Pengukuran
Perangkat bergerak ke atas	Sumbu Z bernilai A + gravitasi
Perangkat bergerak ke bawah	Sumbu Z bernilai 0

Akselerometer merupakan sensor yang sangat baik digunakan untuk mengukur pergerakan perangkat. Hampir seluruh perangkat android dilengkapi dengan sensor ini dan sensor ini juga membutuhkan tenaga 10 kali lebih ringan daripada sensor lainnya. Kekurangan dari akselerometer adalah perlunya penyaringan data.

Contoh penggunaan akselerometer adalah seperti fitur handshake untuk mengubah lagu, dan juga menyajikan pandangan landscape atau potret layar perangkat. Salah satu aplikasi yang terkenal dan mengimplementasikan penggunaan akselerometer adalah Pokemon Go. Aplikasi ini menggunakan akselerometer dalam membangun teknologi *Augmented Reality* digabungkan dengan sensor magnetometer dan GPS. Akselerometer digunakan untuk mendeteksi pergerakan secara *real-time* dari pengguna aplikasi seperti pada Gambar 2.3 Pokemon-Go.

Gambar 2.3 Visualisasi Implementasi Akselerometer pada Aplikasi Pokemon-Go

2.7.3 Magnetometer

Magnetometer umumnya digunakan untuk mengukur magnetisasi bahan magnetik dan arah medan magnet pada suatu titik [4]. Selain itu, magenetometer juga dapat digunakan untuk menentukan arah mata angin. Sensor ini terdapat pada hampir seluruh perangkat android [6]. Dalam hal penempatan, magnetometer dapat digunakan untuk menentukan arah ketika perangkat sedang bergerak menuju suatu tempat.

Magnetometer dapat digunakan dalam membuat *Indoor Positioning System* [7]. Dalam *IPS*, magnetometer menggunakan medan magnet bumi untuk menentukan posisi di dalam ruangan dan dapat mendeteksi pergerakan perangkat yang dilakukan dengan akurat. Penggunaan magnetometer sangat dibutuhkan dalam IPS karena mampu mendeteksi pergerakan dan posisi perangkat dengan lebih akurat daripada sensor posisi lainnya. Selain mendeteksi pergerakan dan posisi, sensor ini juga dapat digunakan untuk memetakan tempat di dalam ruangan. Dalam *positioning*, magnetometer di dalam penerapan pada IPS dapat memberikan hasil yang lebih akurat daripada penggunaan GPS. Perbedaan antara GPS dan IPS dapat dilihat pada Gambar 2.4 [6].

Gambar 2.4 Perbedaan GPS dan IPS

Selain keunggulannya dalam hal *positioning*¸ magnetometer memilik kekurangan yang harus diketahui oleh para penggunanya yaitu berubahnya medan magnet bumi seiring dengan perubahan kerak bumi.

Pada tugas akhir ini, magnetometer digunakan untuk meningkatkan akurasi gyroscope dalam mendeteksi belok / perubahan arah kendaraan dan menghilangkan kesalahan deteksi pada gyroscope ketika perpindahan lajur.

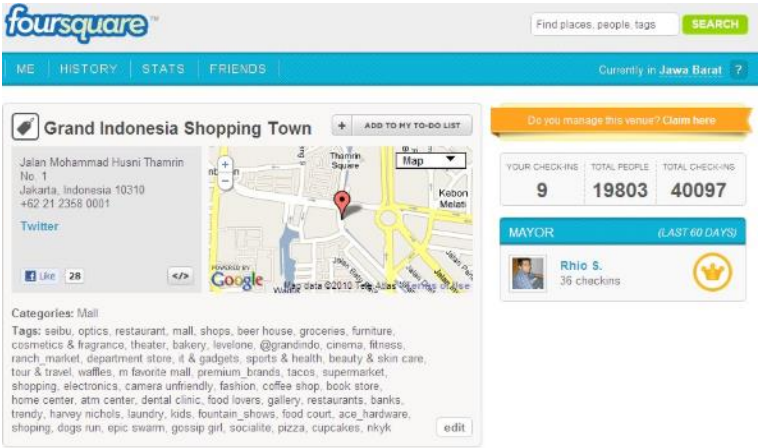
2.8 Google Maps

Google maps merupakan salah satu layanan yang dikembangkan oleh Google. Layanan pemetaan web ini memungkinkan kita untuk melihat citra satelit, peta jalan, panorama, kondisi lalu lintas, rute perjalanan dan masih banyak lagi. Layanan ini menggunakan Javascript, XML, dan AJAX. Biasanya google maps lebih sering digunakan untuk mencari lokasi dan rute perjalanan seperti pada Gambar 2.5.

Gambar 2.5 Tampilan Google Maps pada Handphone

Google maps juga menyediakan beberapa API guna memungkinkan pengguna untuk mengembangkan dan memperbaharui pemetaan layanan di dunia. Contoh penggunaan API Google yang terdapat pada beberapa aplikasi yang sudah kita kenal yaitu GOJEK dan *Foursquare* yang dapar dilihat pada Gambar 2.6 dan Gambar 2.7.

Gambar 2.6 Implementasi API Google pada Go-Jek



Gambar 2.7 Implementasi API Google pada Foursquare

Beberapa API Google juga digunakan pada pembuatan tugas akhir ini. API yang diperlukan diantaranya adalah Google Maps SDK for Android, Places API, Places SDK for Android, dan Directions API yang diperlukan dalam proses memilih tempat tujuan dan menampilkan informasi perjalanan ke tempat tujuan. Google Maps SDK for Android diperlukan untuk mengakses API dari Google Maps melalui perangkat Android. Aplikasi ini menggunakan Place Picker untuk memilih tempat tujuan yang kemudian detail dari tempat tujuan tersebut didapatkan dari Places API. Terakhir, info perjalanan didapatkan menggunakan Directions API.

2.9 Advance Driver Assistance System

Advance Driver Assistance System (ADAS) atau Sistem Asisten Mengemudi adalah sebuah sistem untuk membantu pengemudi ketika berkendara [8]. Sistem ini dibuat dengan tujuan untuk meningkatkan keamanan pada perjalanan dengan meminimalisir kesalahan yang dilakukan oleh pengemudi. Sistem ini adalah sistem yang dikembangkan untuk otomasi, adaptasi dan menambah sistem kendaraan agar mengemudi lebih baik dan aman. Fitur keamanan didesain untuk menghindari kecelakaan dengan

memberikan peringatan kepada pengemudi atau menambah keamanan dengan mengontrol faktor-faktor yang dapat menyebabkan kecelakaan.

GPS adalah sistem asisten mengemudi moderen pertama yang dikenalkan pada publik, yaitu pada tahun 1990 [9]. Sistem ini kemudian dianggap terbukti mampu meningkatkan pengalaman mengemudi dan keamanan berkendara sehingga terus diteliti dan kini berkembang dengan sangat pesat. Ada banyak fitur yang dimiliki oleh *ADAS*; beberapa fitur dirancang pada kendaraan dan beberapa fitur lainnya membutuhkan perangkat tambahan. Fitur-fitur dari *ADAS* yang telah dikembangkan maupun sedang dikembangkan saat ini dapat dilihat pada Tabel 2.8.

Tabel 2.8 Fitur Advance Driving Assistance System

NO	FITUR
1	Adaptive Cruise Control
2	Adaptive Light Control
3	Automatic Braking
4	Automatic Parking
5	Blind Spot Detection
6	Collision Avoidance Systems
7	Driver Drowsiness Detection
8	GPS Navigation
9	Hill Descent Control
10	Intelligent Speed Adaptation
11	Lane Departure Warning Systems
12	Night Vision
13	Tire Pressure Monitoring
14	Electronic Control Unit (ECU)
15	Pre-crash Warning
16	Human Machine Interface (HMI)
17	Smartphone Control Assistance

Terdapat setidaknya 17 fitur [9] [10] [11] dari *ADAS* yang sudah ada saat ini. Fitur-fitur yang ditanam pada kendaraan tersebut dirancang menggunakan *Electronic Control Unit* dan sensor. Detail penggunaan perangkat yang ditanam untuk masing-masing fitur dapat dilihat pada Tabel 2.9 [9].

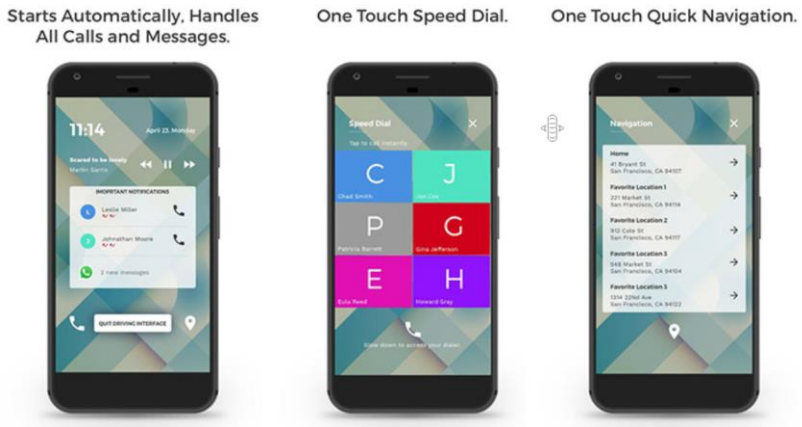
Tabel 2.9 Penggunaan Perangkat pada Implementasi ADAS

Fitur	Perangkat yang Digunakan
Cruise Control	ECU
Adaptive Cruise Control	ECU, Laser Sensor
Pre-crash Systems	Laser Sensor, Camera
Blind Spot Information System	Camera, Radar
Lane Departure Warning System	Camera
Autonomous Parking Assistance Systems	Radar, GPS
Drowsiness Detection System	Camera, Infrared LED

Selain dengan menggunakan perangkat yang ditanam pada kendaraan, beberapa industri teknologi juga mengembangkan perangkat *ADAS* berbasis perangkat bergerak. Fitur yang diberikan pada perangkat bergerak ini beragam, diantaranya adalah mendeteksi perilaku pengemudi menggunakan sensor pada perangkat bergerak yang telah dijelaskan pada subbab 2.5 dan mengontrol dan otomasi *handphone* pengemudi yang dikembangkan oleh Kruzr [11].

Kruzr menilai bahwa *handphone* adalah perangkat yang mampu meningkatkan konektifitas dan produktifitas penggunanya dan merupakan salah satu kebutuhan utama dalam beraktifitas. Akan tetapi, penggunaan *handphone* ketika berkendara justru dapat menyebabkan kecelakaan. Kurang lebih sekitar 25-30% kecelakaan dalam berkendara disebabkan oleh penggunaan *handphone* [12], terlebih 95% panggilan dan pesan yang diterima oleh pengguna *handphone* diterima ketika mengemudi. Berdasarkan permasalahan tersebut, maka Kruzr mengembangkan aplikasi yang mampu mengontrol *handphone* pengemudi dan mendeteksi ketika perangkat sedang bergerak dan menanyakan apakah pengguna sedang

mengemudi atau tidak. Aplikasi akan memulai dengan sendirinya apabila pengguna memberikan pesan bahwa ia sedang mengemudikan kendaraan. Salah satu fitur pada aplikasi ini adalah mengontrol masuknya pesan dan panggilan kepada *handphone* pengemudi dan memberikan pesan kepada pengemudi untuk selanjutnya pengemudi memberikan umpan balik apakah pesan atau panggilan tersebut penting. Selain itu, aplikasi ini juga memberikan kemudahan bagi pengemudi untuk menggunakan navigasi dan melakukan panggilan dengan memberikan fitur *one-touch speed dial* dan *one-touch navigation.* Tampilan aplikasi Kruzr dapat dilihat pada Gambar 2.8.



Gambar 2.8 Tampilan Aplikasi Kruzr

Penggunaan *ADAS* pada kendaraan juga telah banyak dilakukan oleh para perusahan otomotif akhir-akhir ini [13]. Beberapa perusahaan yang menanam sistem tersebut dapat dilihat pada Tabel 2.10 [12].

Tabel 2.10 Perusahaan Otomotif yang Menggunakan ADAS

Perusahaan	Produk	ADAS
Honda	2017 Honda Civic	Cruise control, forward-collision

Perusahaan	Produk	ADAS
		warning, collision-mitigation braking, lane-departure warning, lane-keeping assistance, road-departure mitigation, blind-spot system, rain-sensing wipers, dan standard multi-angle rearview camera
Subaru	2017 Subaru Impreza	Adaptive cruise control, blind spot monitor, pre-collision braking, dan lane-departure/lane-keeping system
Chevrolet	2017 Chevrolet Spark	Forward-collision alert, lane-departure warning, dan rear parking sensors
Toyota	2017 Toyota Yaris iA	Rear-view camera, Pre-Collision System, forward collision alert dan forward automatic-braking function

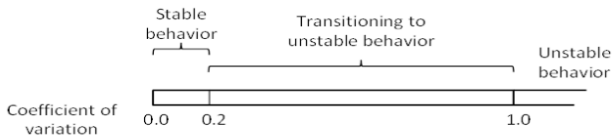
Teknologi *ADAS* diprediksi akan dikembangkan menjadi *Automated Driving System*¸ yaitu sistem yang mampu mengemudikan kendaraan secara otomatis. Dari subbab ini, dapat diketahui bahwa teknologi dapat meningkatkan keamanan dalam berkendara dan penerapan dalam perangkat bergerak memudahkan pengguna untuk menerapkankannya dimana saja sehingga pembuatan Sistem Infomasi Skoring Perilaku Pengemudi ini diharapkan mampu untuk membantu *ADAS* dan meningkatkan keamanan dalam mengemudi untuk meminimalisir terjadinya kecelakaan.

2.10 Coefficient of Variation

Koefisien variasi, juga dikenal sebagai *Coefficient of Variation* / *Relative Standard Deviation*, adalah ukuran standar penyebaran distribusi probabilitas atau distribusi frekuensi [14].

Koefisien variasi dapat diaplikasikan dalam beberapa hal, seperti analisis perubahan cuaca, analisis perubahan permintaan pelanggan, dan kimia analitik. Koefisien variasi diperoleh dengan membagi simpangan baku dengan rata-rata sampel.

$$KV = 100\% * \frac{standard\ deviation\ for\ data\ set\ (\sigma)}{average\ for\ the\ data\ set\ (X)}$$



Nilai koefisien variasi yang rendah (< 20%) menunjukkan bahwa data set tersebut stabil, nilai koefisien variasi 20-100 menunjukkan dat set tersebut transisi menuju tidak stabil, sedangkan nilai yang tinggi (> 100%) menunjukkan bahwa data tersebut sangat bervariasi [15].

Dalam contoh kasus menganalisis kebutuhan pelanggan, maka koefisien variasi diterapkan untuk menghitung stabilitas permintaan pelanggan terhadap suatu produk. Misalkan dalam 26 minggu, suatu produk memiliki rata-rata permintaan pelanggan 300 unit dengan standar deviasi 47 unit, maka koefisien variasi dari data tersebut adalah 16% dan kesimpulannya produk tersebut memiliki permintaan yang stabil.

$$KV = 100\% * \frac{47}{300} = 16\%$$

Pada tugas akhir ini, digunakan koefisien variasi untuk menghitung stabilitas aplikasi pada uji coba dengan beragam kondisi. Apabila data pada hasil uji tidak normal dan memiliki sebaran nilai yang kecil antara 0-10, maka dapat digunakan transformasi akar pada data tersebut [16] dengan mengubah setiap data menjadi akar dari data tersebut ditambah 0.5.

$$x' = \sqrt{(x + 0.5)}$$

BAB III
ANALISIS DAN PERANCANGAN SISTEM

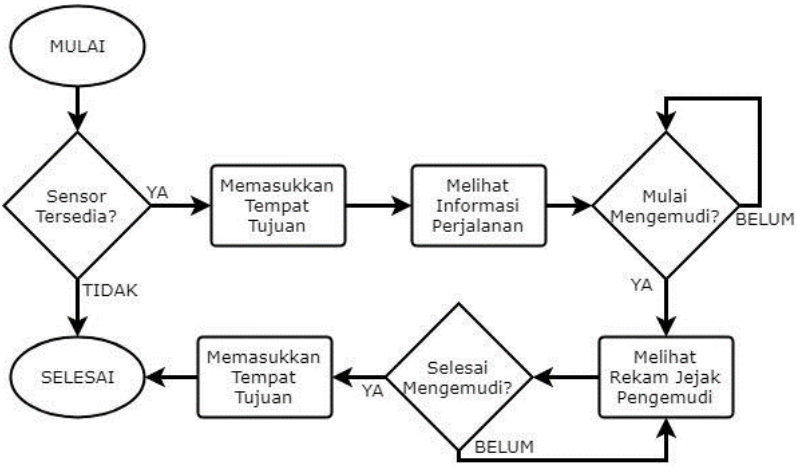
3.1 Analisis Sistem

Pada subbab ini akan dijelaskan deskripsi umum aplikasi, spesifikasi kebutuhan sistem pada aplikasi *Scoring* Perilaku Pengemudi, serta identifikasi aktor aplikasi *Scoring* Perilaku Pengemudi.

3.1.1 Deskripsi Umum

Aplikasi *Scoring* Perilaku Pengemudi adalah aplikasi berbasis perangkat bergerak yang digunakan untuk menilai perilaku pengemudi saat berkendara secara teknis menggunakan sensor yang terdapat pada android. Aplikasi ini dapat melihat status sensor, memasukkan tujuan perjalanan, melihat info perjalanan, melihat status perjalanan normal dan agresif, serta melihat *score* akhir pengemudi yang merupakan tujuan utama dari aplikasi ini.

Alur pada aplikasi ini terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alur Aplikasi

3.1.2 Analisis Kebutuhan Sistem

Analisis spesifikasi kebutuhan dalam aplikasi ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses yang dibutuhkan dan dijalankan dalam sistem. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.1.

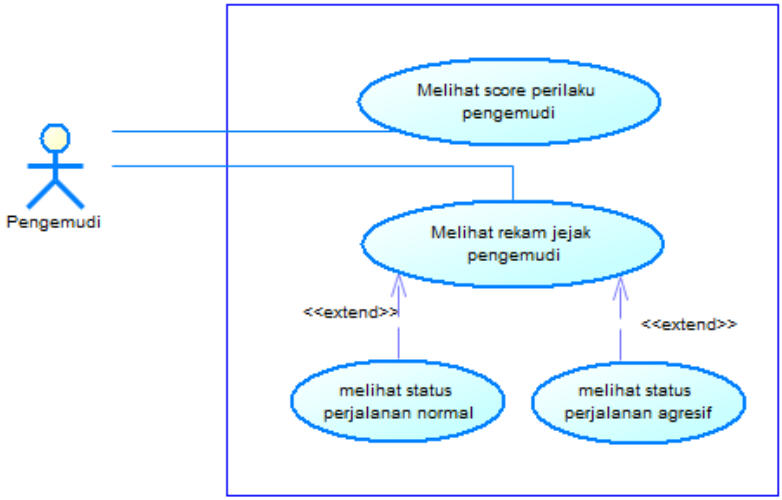
Tabel 3.1 Kebutuhan Fungsional

Kode Kebutuhan	**Kebutuhan Fungsional**	**Deskripsi**
F-1.1	Melihat *score* pengemudi	Pengemudi dapat melihat hasil score akhir pengemudi yang didapat dalam perjalanan dalam bentuk angka 0-100.
F-1.2	Melihat rekam jejak pengemudi	Pengemudi dapat melihat laporan kegiatan yang dilakukan secara *real time.*
F-1.2.1	Melihat status perjalanan normal	Pengemudi dapat melihat perilaku mengemudi yang normal
F-1.2.2	Melihat status perjalanan agresif	Pengemudi dapat melihat perilaku mengemudi yang agresif
F-2.1	Sensor menyiapkan sensor gyro	Sensir gyro menyiapkan sensor yang ada pada smartphone
F-2.2	Sensor mendeteksi rotasi perangkat	Sensor hyro mendeteksi rotasi perangkat
F-2.3	Sensor gyro mengirim data ke kontroller	Sensor gyro mengirim data yang telah diambil ke controller
F-3.1	Menyiapkan sensor accelerometer	Menyiapkan sensor accelerometer pada smartphone
F-3.2	Sensor mendeteksi percepatan	Sensor accelerometer mendeteksi percepatan kendaraan

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-3.3	Sensor mengirim data ke kontroller	Sensor accelerometer mengirim data yang telah diambil ke kontroller
F-4.1	Sensor menyiapkan sensor magnetometer	Sensor magnetometer menyiapkan sensor yang ada pada smartphone
F-4.2	Mengukur magnetisasi dan arah medan magnet	Sensor magnetometer mengukur magnetisasi dan arah medan magnet
F-4.3	Sensor mengirim data ke kontroller	Sensor magnetometer mengirim data ke kontroller

3.1.3 Analisis Aktor

Terdapat tiga aktor yang terlibat langsung dalam SI Scoring Perilaku Pengemudi yaitu pengemudi, sensor gyro, sensor accelerometer, dan sensor magnetometer. *Usecase* diagram dapat dilihat pada Gambar 3.2 hingga Gambar 3.5.



Gambar 3.2 Use Case pada Pengemudi

Gambar 3.3 Use Case pada Sensor Gyroscope

Gambar 3.4 Use Case pada Sensor Accelerometer

Gambar 3.5 Use Case pada Sensor Magnetometer

3.2 Perancangan Sistem

Tahap perancangan sistem dibagi menjadi beberapa bagian yaitu kasus penggunaan dan perancangan antarmuka aplikasi.

3.2.1 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan sebelumnya pada spesifikasi kebutuhan fungsional seperti pada Tabel 3.2.

Tabel 3.2 Kasus Penggunaan

Kode Kasus Penggunaan	**Kasus Penggunaan**
UC-001	Melihat *score* pengemudi
UC-002	Melihat rekam jejak pengemudi
UC-003	Melihat status perjalanan normal
UC-004	Melihat status perjalanan agresif
UC-005	Sensor mendeteksi rotasi perangkat
UC-006	Sensor mendeteksi percepatan
UC-007	Mengukur magnetisasi dan arah medan magnet

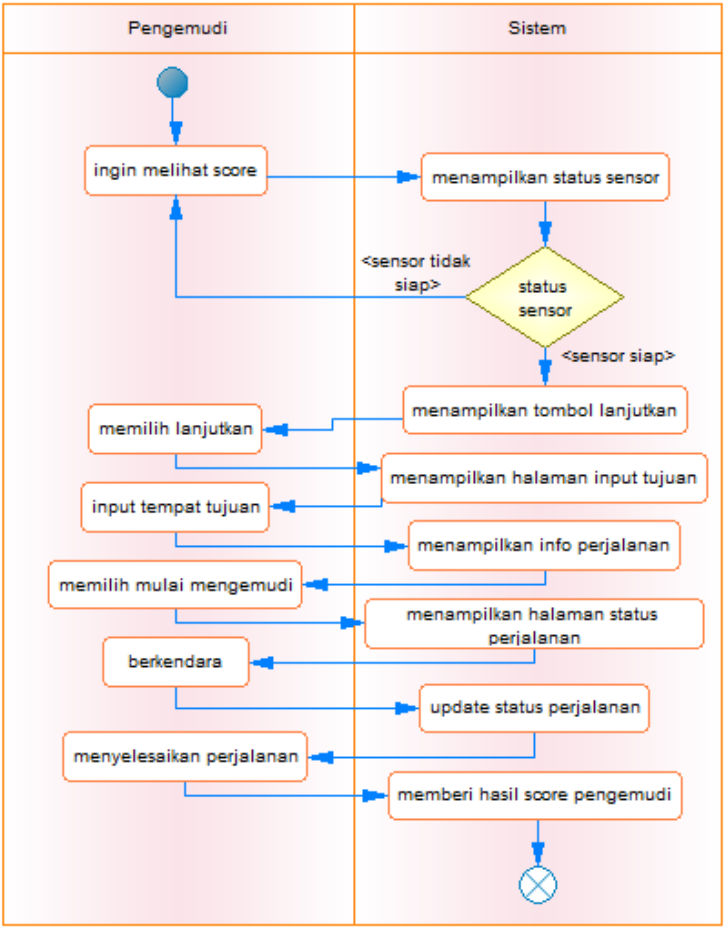
3.2.1.1 Melihat Score Pengemudi

Pada kasus penggunaan ini, pengemudi dapat mendapatkan hasil *scoring* perilaku pengemudi setelah selesai berkendara. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.3 dan aktivitas diagram pada Gambar 3.6.

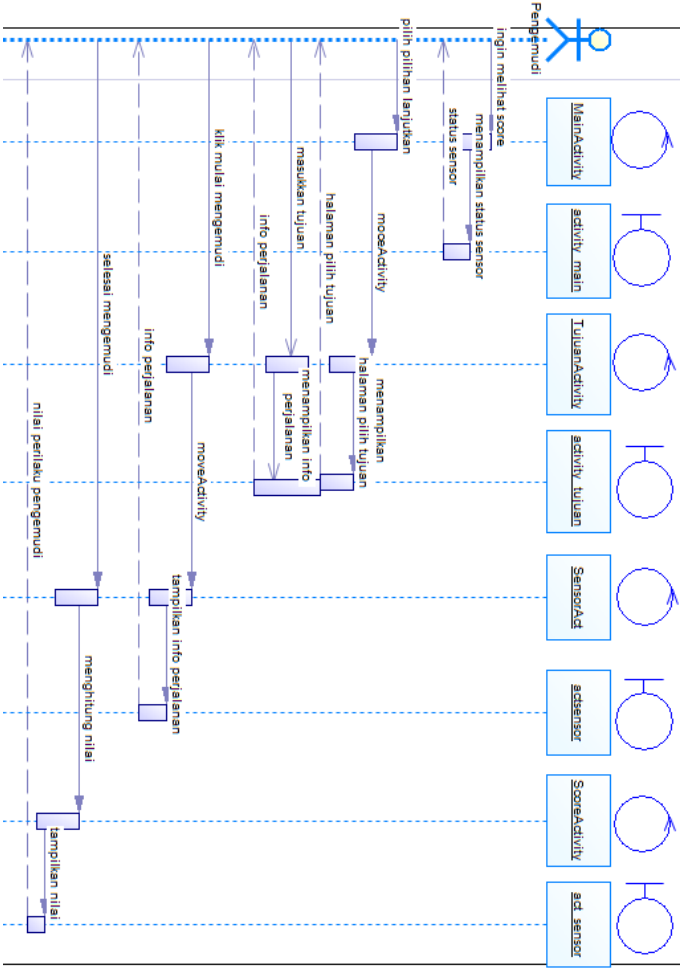
Tabel 3.3 Melihat Score Pengemudi

Kode	UC-001
Nama	Melihat score pengemudi
Aktor	Pengemudi
Deskripsi	Pengemudi dapat melihat hasil score pengemudi yang didapat setelah mengemudi dalam bentuk angka 0-100.
Kondisi Awal	Pengemudi membuka aplikasi
Kondisi akhir	Aplikasi memberikan hasil score
Alur Kejadian Normal	
1. Pengemudi ingin melihat *score* pengemudi 2. Sistem menampilkan status sensor 3. Sistem menampilkan pilihan lanjutkan 4. Pengemudi memilih lanjutkan 5. Sistem menampilkan halaman input tujuan 6. Pengemudi memasukkan tempat tujuan 7. Sistem menampilkan info perjalanan yang akan ditempuh 8. Pengemudi memilih mulai mengemudi	

9. Sistem menampilkan halaman status perjalanan 10. Pengemudi berkendara 11. Sistem mendeteksi dan *update* status perilaku pengemudi 12. Pengemudi menyelesaikan perjalanan 13. sistem memberi hasil score pengemudi
Alur Kejadian Alternatif
1. Sensor belum siap 2.1 kembali ke nomor 1



Gambar 3.6 Diagram Aktivitas Melihat Score Pengemudi



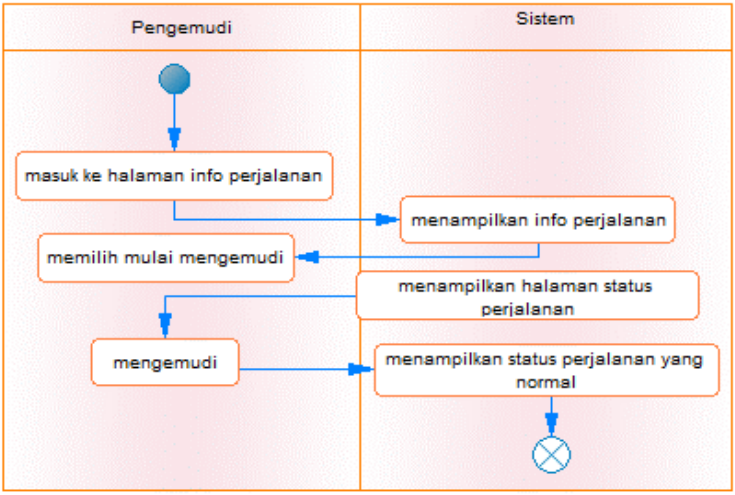
Gambar 3.7 Sequence diagram melihat score pengemudi

3.2.1.2 Melihat rekam jejak pengemudi

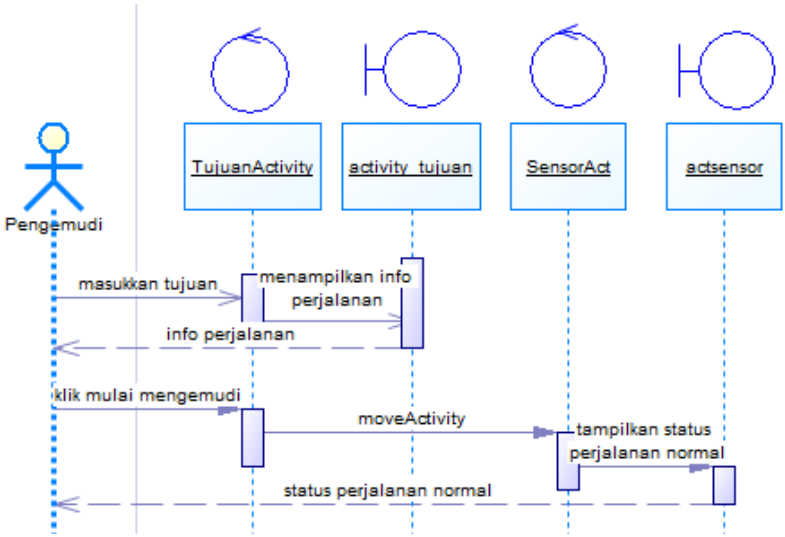
Pada kasus penggunaan ini, status kegiatan dapat ter-*update* setiap satu detik. Rekam jejak pengemudi diklasifikasikan dalam kategori normal dan agresif untuk setiap manuver. Spesifikasi kasus penggunaan status perjalanan normal dapat dilihat pada Tabel 3.4 dan diagram aktivitas pada Gambar 3.8. Spesifikasi kasus penggunaan status perjalanan agresif dapat dilihat pada Tabel 3.5 dan diagram aktivitas pada Gambar 3.10.

Tabel 3.4 Melihat status perjalanan normal

Kode	UC-003
Nama	Melihat status perjalanan normal
Aktor	Pengemudi
Deskripsi	Pengemudi dapat melihat rekam jejak perjalanan yang statusnya normal
Kondisi Awal	Pengemudi berada di halaman info perjalanan
Kondisi akhir	Pengemudi melihat rekam jejak perjalanan yang normal
Alur Kejadian Normal	
1. Pengemudi masuk ke halaman info perjalanan 2. Sistem menampilkan info perjalanan yang akan ditempuh 3. Pengemudi memilih mulai mengemudi 4. Sistem menampilkan halaman rekam jejak pengemudi yang normal	
Alur Kejadian Alternatif	
-	



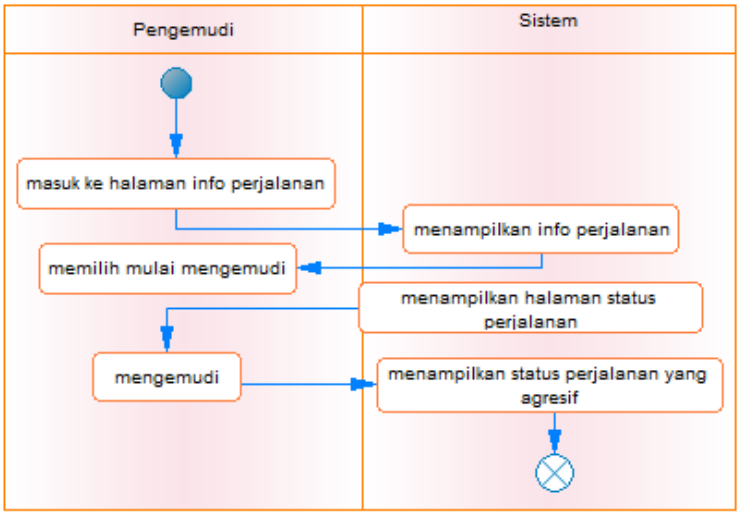
Gambar 3.8 Diagram Aktivitas Melihat Status Perjalanan Normal



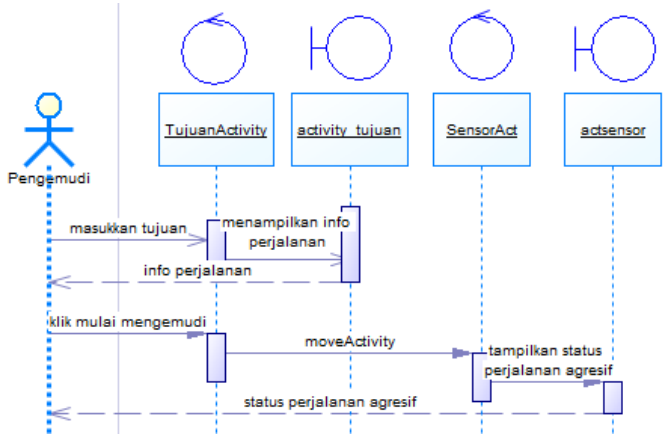
Gambar 3.9 Sequence diagram melihat status perjalanan normal

Tabel 3.5 Melihat status perjalanan agresif

Kode	UC-004
Nama	Melihat status perjalanan agresif
Aktor	Pengemudi
Deskripsi	Pengemudi dapat melihat rekam jejak perjalanan yang statusnya agresif
Kondisi Awal	Pengemudi berada di halaman info perjalanan
Kondisi akhir	Pengemudi melihat rekam jejak perjalanan yang agresif
Alur Kejadian Normal	
1. Pengemudi masuk ke halaman info perjalanan 2. Sistem menampilkan info perjalanan yang akan ditempuh 3. Pengemudi memilih mulai mengemudi 4. Sistem menampilkan halaman rekam jejak pengemudi yang agresif	
Alur Kejadian Alternatif	
-	



Gambar 3.10 Diagram Aktivitas Melihat status perjalanan agresif



Gambar 3.11 Sequence Diagram Melihat Status Perjalanan Agresif

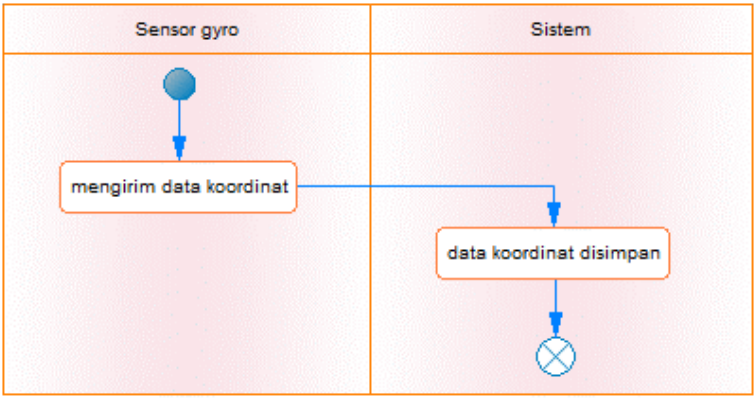
3.2.1.3 Mendeteksi Rotasi Perangkat

Pada kasus penggunaan ini, sensor gyro dapat mendeteksi rotasi perangkat yang dilakukan saat perjalanan. Data yang diambil dari sensor gyro adalah tiga dimensi. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.6 dan diagram aktivitas pada Gambar 3.12.

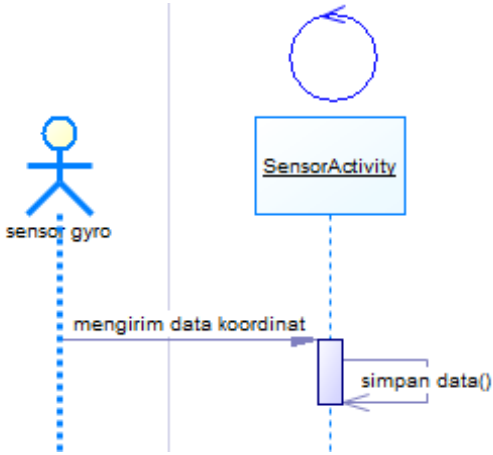
Tabel 3.6 Mendeteksi rotasi perangkat

Kode	UC-005
Nama	Mendeteksi rotasi perangkat
Aktor	Sensor Gyro
Deskripsi	Sensor gyro mendeteksi rotasi pergerakan mobil
Kondisi Awal	Sensor Gyro siap digunakan
Kondisi akhir	Sistem mengolah data dari sensor gyro
Alur Kejadian Normal	
1. Sensor mengirim data koordinat 2. Sistem menyimpan data koordinat	
Alur Kejadian Alternatif	

-



Gambar 3.12 Diagram Aktivitas Mendeteksi Rotasi Perangkat



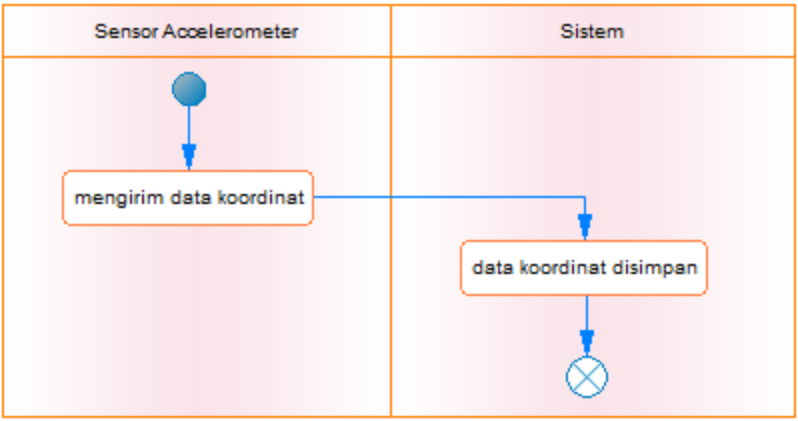
Gambar 3.13 Sequence Diagram Sensor Gyro

3.2.1.4 Mengukur percepatan

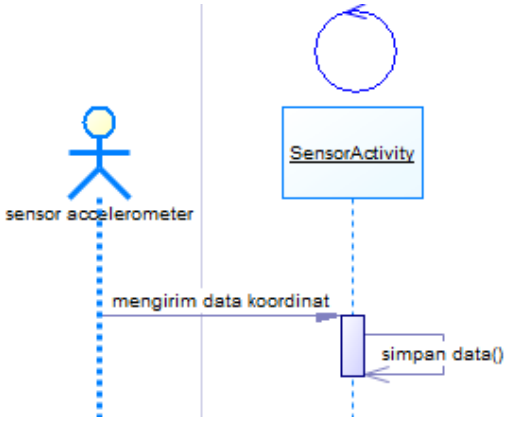
Pada kasus penggunaan ini, sensor *accelerometer* mengukur percepatan selama perjalanan.Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.7 dan diagram aktivitas pada Gambar 3.14.

Tabel 3.7 Melihat status perjalanan normal dan agresif

Kode	UC-006
Nama	Mengukur percepatan
Aktor	Sensor Accelerometer
Deskripsi	Sensor Accelerometer mengukur percepatan perjalanan
Kondisi Awal	Sensor Accelerometer siap digunakan
Kondisi akhir	Sistem mengolah data dari sensor Accelerometer
Alur Kejadian Normal	
1. Sensor mengirim data koordinat 2. Sistem menyimpan data koordinat	
Alur Kejadian Alternatif	
-	



Gambar 3.14 Diagram Aktivitas Mengukur Kecepatan



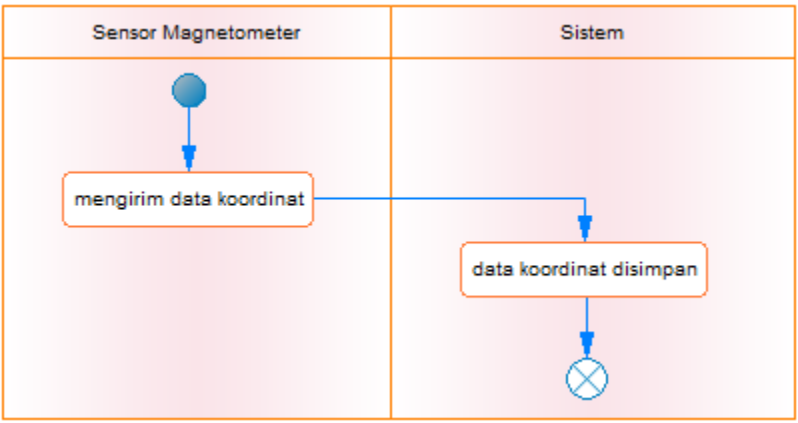
Gambar 3.15 Sequence Diagram Mengukur Percepatan

3.2.1.5 Mengukur magnetisasi dan arah medan magnet

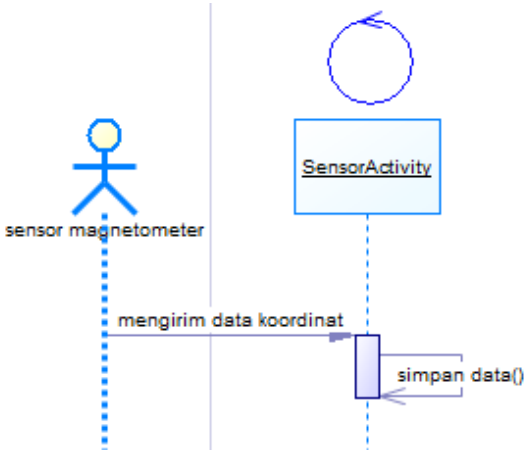
Pada kasus penggunaan ini, sensor magnetometer dibutuhkan untuk mengukur magnetisasi dan arah medan magnet pengemudi saat berkendara. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.8 dan aktivitas diagram pada Gambar 3.16.

Tabel 3.8 Diagram Aktivitas Mengukur magnetisasi dan arah medan magnet

Kode	UC-007
Nama	Mendeteksi rotasi perangkat
Aktor	Sensor Magnetometer
Deskripsi	Sensor magnetometer mengukur magnetisasi dan arah medan magnet
Kondisi Awal	Sensor Magnetometer siap digunakan
Kondisi akhir	Sistem mengolah data dari sensor magnetometer
Alur Kejadian Normal	
1. Sensor mengirim data koordinat 2. Sistem menyimpan data koordinat	
Alur Kejadian Alternatif	
-	



Gambar 3.16 Diagram Aktivitas Mengukur Magnetisasi dan Arah Medan Magnet



Gambar 3.17 Sequence Diagram Mengukur Magnetisasi dan Arah Medan Magnet

3.2.2 Perancangan Antarmuka

Subbab ini menjelaskan bagaimana rancangan antarmuka berinteraksi langsung dengan pengguna.

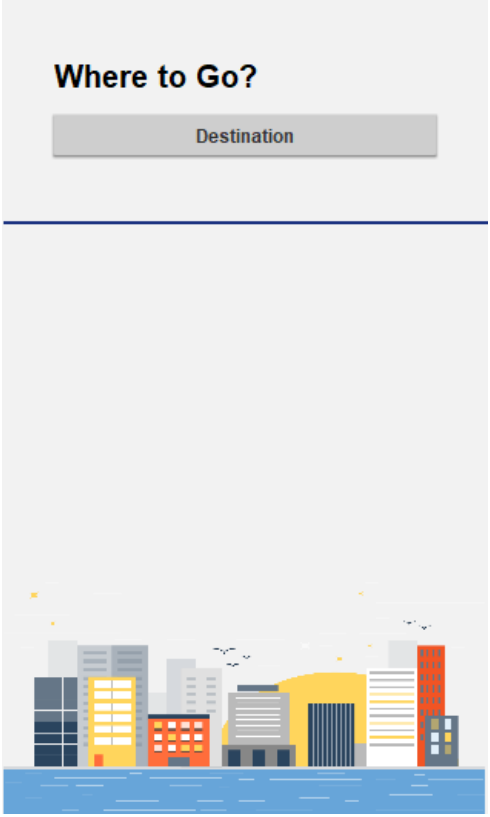
3.2.2.1 Rancangan Antarmuka Halaman Utama

Gambar 3.18 menunjukkan halaman utama aplikasi yang akan memberikan informasi kepada *user*. Informasi yang diberikan berupa status sensor yang akan digunakan pada *smartphone* Android. Terdapat tanda *checklist* yang menandakan bahwa sensor siap digunakan. Terdapat pilihan *next* untuk melanjutkan proses ke halaman selanjutnya.

Gambar 3.18 Perancang UI Halaman Utama

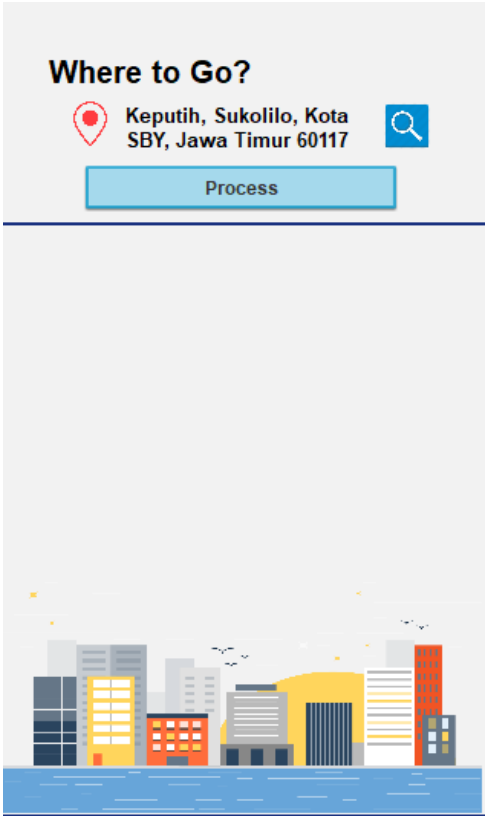
3.2.2.2 Rancangan Antarmuka Halaman Tujuan

Halaman tujuan berfungsi untuk memasukkan alamat tujuan perjalanan yang akan ditempuh. Pada Gambar 3.19 terdapat pilihan destination yang akan membuka Google Maps untuk memilih tempat.



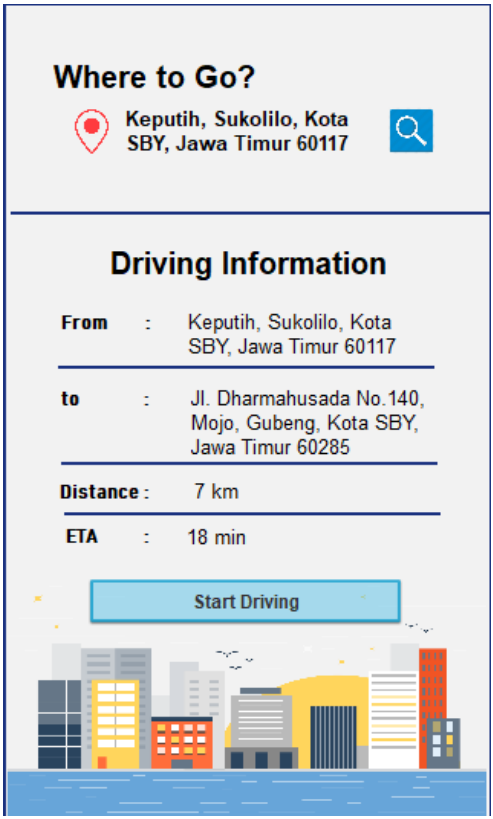
Gambar 3.19 Perancangan UI Halaman Tujuan (1)

Pada Gambar 3.20, terdapat *button process* untuk membuka info perjalanan yang akan ditempuh dan *icon loop* untuk mengulang pencarian tempat tujuan apabila tempat tujuan yang tertera tidak tersedia.



Gambar 3.20 Perancangan UI Halaman Tujuan (2)

Pada Gambar 3.21, terdapat informasi perjalanan yang akan dilakukan. Informasi tersebut meliputi tempat asal, tujuan, jarak dan juga perkiraan waktu sampai tujuan. Terdapat *button start driving* untuk memulai penilaian perilaku pengemudi.



Gambar 3.21 Perancangan UI Halaman Tujuan (3)

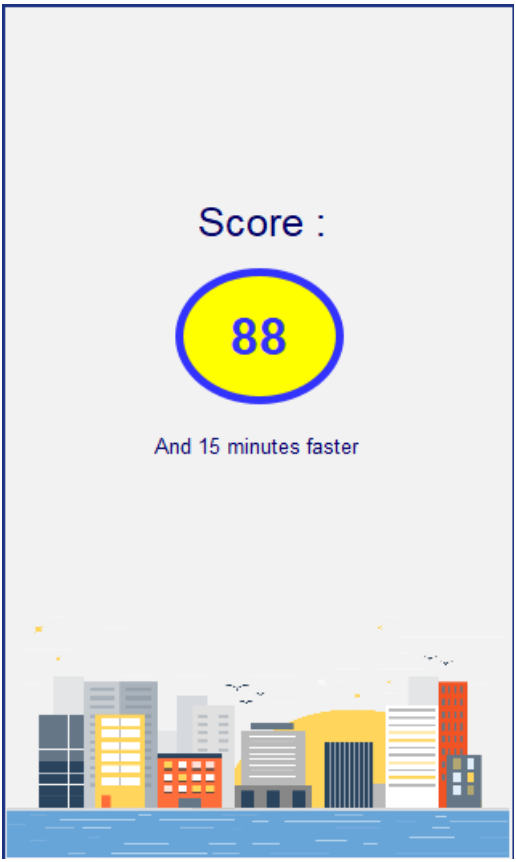
3.2.2.3 Rancangan Antarmuka Halaman Proses

Halaman Proses pada Gambar 3.22 menampilkan perilaku pengemudi selama perjalanan secara langsung, estimasi sampai ke tujuan, serta *button finish* untuk menyelesaikan perjalanan dan melihat nilai perilaku pengemudi.

Gambar 3.22 Perancangan UI Halaman Proses

3.2.2.4 Rancangan Antarmuka Halaman Skor Pengemudi

Halaman skor pengemudi pada Gambar 3.23 menampilkan nilai akhir pengemudi dan perbandingan durasi sampai dengan estimasinya.



Gambar 3.23 Perancangan UI Halaman Skor Pengemudi

(Halaman ini sengaja dikosongkan)

BAB IV
IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi adalah kode sumber untuk membangun program. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Dalam merancang aplikasi ini, digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

- Laptop
  Tipe : Macbook Pro
  Prosesor : Intel® Core™ i5-CPU (2.50 GHz)
  Memori/RAM : 8GB
- Handphone
  Tipe : Samsung A8
  Memori : 32GB
- Sensor
  1. Accelerometer
  2. Gyroscope
  3. Magnetometer

4.1.2 Lingkungan Implementasi Perangkat Lunak

Penjelasan perangkat lunak yang digunakan dalam implementasi aplikasi ini adalah sebagai berikut:

1. MacOS sebagai sistem operasi.
2. Android Studio sebagai *editor* kode dan *compiler*.

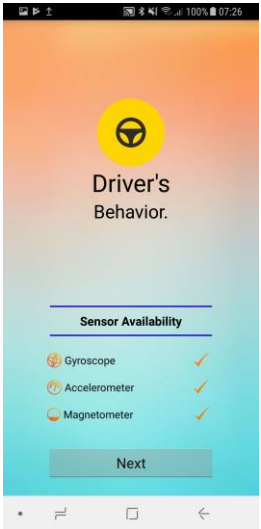
4.2 Implementasi Tampilan Antarmuka

Subbab ini membahas tentang implementasi tampilan antarmuka yang telah dirancang dan dibahas pada Bab III.

Selanjutnya akan dirinci berdasarkan halaman yang akan tampil dan dilihat oleh partisipan.

4.2.1 Implementasi Halaman Utama

Halaman pada Gambar 4.1 menunjukkan halaman utama yang berisi status sensor yang akan digunakan. *Icon* centang menunjukkan sensor siap digunakan. Terdapat pilihan next untuk melanjutkan proses *scoring* perilaku pengemudi yang mengarah ke halaman selanjutnya yaitu halaman tujuan. Kode Sumber dari proses yang terjadi pada halaman utama dapat dilihat pada Kode Sumber 4.1.



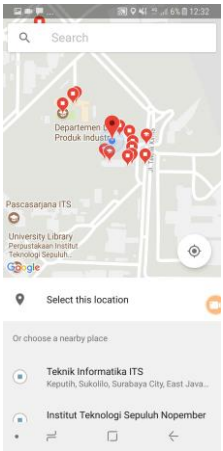
Gambar 4.1 UI Halaman Utama

4.2.2 Implementasi Halaman Tujuan

Halaman ini berisi dua bagian, yaitu tujuan perjalanan dan info perjalanan yang akan ditempuh. Pilihan di sebelah kanan pada bagian tujuan berfungsi untuk memilih tempat tujuan seperti pada Gambar 4.2. Kode Sumber dari proses yang terjadi pada halaman tujuan ketika tombol “destination” ditekan dan menampilkan *Place Picker* dapat dilihat pada Kode Sumber 4.2.

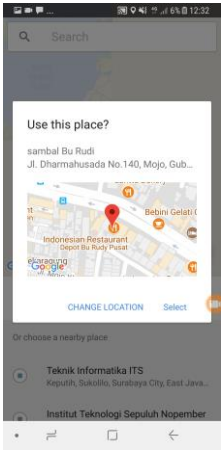
Gambar 4.2 UI Halaman Tujuan (1)

Kolom *search* berfungsi untuk memasukkan tempat tujuan seperti pada Gambar 4.3.



Gambar 4.3 UI Halaman Tujuan (2)

Pilihan select pada Gambar 4.4 berfungsi untuk mengkonfirmasi tempat tujuan. Pilihan *change location* dipilih apabila tempat tujuan belum valid.



Gambar 4.4 UI Halaman Tujuan (3)

Pilihan proses di bagian tujuan berfungsi untuk melihat info perjalanan yang akan ditempuh seperti pada Gambar 4.5. Kode sumber proses ini dapat dilihat pada Kode Sumber 4.3.

Gambar 4.5 UI Halaman Tujuan (4)

Pada bagian info perjalanan, terdapat pilihan mulai mengemudi seperti yang dijelaskan pada Gambar 4.6 untuk memulai penilaian perilaku pengemudi. Kode sumber proses untuk menampilkan info perjalanan dapat dilihat pada Kode Sumber 4.4 dan Kode Sumber 4.5.

Gambar 4.6 UI Halaman Tujuan (5)

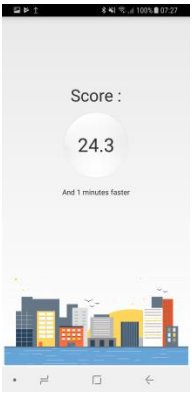
4.2.3 Implementasi Halaman Status Pegemudi

Halaman ini berisi status perilaku pengemudi yang sedang berjalan. Terdapat lima kotak yang dapat diamati di halman ini. Halaman yang dapat diamati adalah belok kiri, belok kiri mendadak, belok kanan, belok kanan mendadak, penambahan kecepatan, penambahan kecepatan mendadak, pengurangan kecepatan, dan pengurangan kecepatan mendadak, serta ETA yang berisi perkiraan waktu sampai tujuan seperti pada Gambar 4.7. Seluruh proses pada halaman status pengemudi dapat dilihat pada Kode Sumber 4.7.

Gambar 4.7 UI Halaman Status Pengemudi

4.2.4 Implementasi Halaman Skor Pengemudi

Halaman skor pengemudi berisi hasil akhir penilaian pengemudi dan perbedaan waktu tempuh sesungguhnya dengan estimasi waktu tempuh seperti pada Gambar 4.8. Kode sumber untuk perhitungan skor dapat dilihat pada Kode Sumber 4.8 sedangkan kode sumber menampilkan skor dapat dilihat Kode Sumber 4.9.



Gambar 4.8 UI Halaman Skor Pengemudi

4.3 Implementasi Perangkat Lunak

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari melihat status sensor hingga menampilkan skor perilaku pengemudi.

4.3.1 Implementasi Proses Melihat Status Sensor

Implementasi proses melihat status sensor memiliki beberapa proses yang dilakukan oleh sistem, yaitu memanggil sensor, mengecek apakah sensor siap, kemudian menampilkan status sensor pada antarmuka. Kode sumber dapat dilihat pada Kode Sumber 4.1.

Sistem memanggil masing-masing sensor dan menempatkan pada masing-masing variabel (*line 6-11)*. Setelah memanggil sensor, kemudian sistem akan melakukan pengecekan apakah sensor tersebut tersedia dan siap dan menampilkan tanda centang pada layout untuk magnetometer (*line 14-22*), magnetometer (*line 25-33*), dan akselerometer (*line 36-44*).

```
1   private boolean checkSensorAvailability()
2      {
3         boolean gyro, accelero, magneto;
4
5         // memanggil sensor, assign ke variabel
6         magnetometer =
7   sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
8         gyroscope =
9   sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
10        accelerometer =
11  sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
12
13        // cek magnetometer
14        if(magnetometer==null) {
15           magneto = false;
16           // jika tidak tersedia, tidak menampilkan centang
17           tvMagnetometer.setVisibility(View.INVISIBLE);
18        } else {
19           magneto = true;
20           // jika tersedia, tampilkan centang
21           tvMagnetometer.setVisibility(View.VISIBLE);
```

```
22        }
23
24        // cek gyroscope
25        if(gyroscope==null) {
26           gyro = false;
27           // jika tidak tersedia, tidak menampilkan centang
28           tvGyroscope.setVisibility(View.INVISIBLE)
29        } else {
30           gyro = true;
31           // jika tersedia, tampilkan centang
32           tvGyroscope.setText(View.VISIBLE);
33        }
34
35        // cek akselerometer
36        if(accelerometer==null) {
37           accelero = false;
38           // jika tidak tersedia, tidak menampilkan centang
39           tvAccelerometer. setVisibility (View.INVISIBLE);
40        } else {
41           accelero = true;
42           // jika tersedia, tampilkan centang
43           tvAccelerometer. setVisibility (View.VISIBLE);
44        }
45
46        if(gyro && accelero && magneto) return true;
47        else return false;
48     }
```

Kode Sumber 4.1 Fungsi Melihat Status Sensor

4.3.2 Implementasi Proses Memilih Tempat Tujuan

Dalam proses memilih tempat tujuan, sistem menunggu user untuk memasukkan tempat tujuan. Proses ini dimulai saat user menekan tombol lokasi yang kemudian sistem akan menampilkan *Place Picker* dari *API Google Maps*. Proses ini dapat dilihat pada Kode Sumber 4.2.

```
1   btnLocation.onClickListener(){
2      // inisialisasi placepicker
3      PlacePicker.IntentBuilder builder = newPlacePicker.IntentBuilder();
4
5      try {
```

```
6        // menampilkan place picker
7        startActivityForResult(builder.build(TujuanActivity.this), 1);
8     } catch (GooglePlayServicesRepairableException e) {
9        e.printStackTrace();
10    } catch (GooglePlayServicesNotAvailableException e) {
11       e.printStackTrace();
12    }
13    break;
14 }
```

Kode Sumber 4.2 Fungsi Proses Menampilkan Place Picker

Setelah user memilih tempat tujuan, selanjutnya hasil *feedback* dari Place Picker (alamat tujuan, latitude tujuan, longitude tujuan) tersebut akan diproses oleh sistem dan ditampilkan pada halaman tujuan (*line 7-13)*. Fungsi ini dapat dilihat pada Kode Sumber 4.3.

```
1  @Override
2  protected void onActivityResult(int requestCode, int resultCode, Intent
3  data) {
4        if (requestCode == 1) {
5           if (resultCode == RESULT_OK) {
6              // mengambil data dari tempat yang dipilih pada place picker
7              Place place = PlacePicker.getPlace(data, this);
8              finish = place.getAddress().toString();
9              finishLat = place.getLatLng().latitude;
10             finishLng = place.getLatLng().longitude;
11
12             // menampilkan data
13             etDestination.setText(finish);
14          }
15       }
16    }
```

Kode Sumber 4.3 Fungsi Proses Menerima Feedback dari Place Picker

4.3.3 Implementasi Proses Menampilkan Informasi Perjalanan yang akan Ditempuh

Sistem akan menampilkan informasi perjalanan yang akan ditempuh oleh pengemudi dari lokasi sekarang menuju ke lokasi

tujuan. Proses ini dimulai ketika user menekan tombol proses yang kemudian sistem akan mendeteksi lokasi sekarang (*line 3)*, mengambil alamat dari lokasi sekarang (*line 7-12*) dan mengambil data tujuan dari API Google Maps (*line 15)* yang dapat dilihat pada Kode Sumber 4.4. Setelah menerima data dari API Google Maps, sistem kemudian melakukan konversi dari JSON ke string (*line 6*) dan menampilkannya ke halaman tujuan (*line 10-13*) yang dapat dilihat pada Kode Sumber 4.5.

```
1   btnProcess.onClickListener(){
2      // get current loc
3      location = gpsTracker.getLocation();
4      geocoder = new Geocoder(TujuanActivity.this, Locale.getDefault());
5
6      // get current loc address
7      try {
8         start = geocoder.getFromLocation(location.getLatitude(),
9   location.getLongitude(), 1).get(0).getAddressLine(0);
10     } catch (IOException e) {
11        e.printStackTrace();
12     }
13
14     // get destination data from gmaps
15     new GetRouteDetails(TujuanActivity.this).execute();
16
17     break;
18  }
```

Kode Sumber 4.4 Fungsi Mendeteksi Lokasi Saat Ini dan Mengambil Data Lokasi Tujuan dari API Google Maps

```
1   private void setInformation(String s) {
2      JSONObject jsonObject = null;
3
4      try {
5         jsonObject = new JSONObject(s);
6         parse(jsonObject);
7      } catch (JSONException e) {
8         e.printStackTrace();
9      } finally {
10        tvStart.setText(start);
```

```
11        tvDestination.setText(finish);
12        tvDistance.setText(distance);
13        tvTime.setText(duration+" ("+time+")");
14    }
15 }
```

Kode Sumber 4.5 Fungsi Menampilkan Informasi Perjalanan yang akan Ditempuh

4.3.4 Implementasi Proses Pengambilan data dari Sensor

Dalam proses pengambilan data dari sensor, sistem terlebih dahulu melakukan pengaturan untuk memanggil sensor, yaitu interval pengambilan data dan sumbu apa saja yang akan diambil dari masing-masing sensor. Sistem akan mengambil sumbu x dari sensor magnetometer (*line 3-23*), sumbu x, y, dan z dari sensor gyroscope (*line 26-52*), dan sumbu x,y, dan z dari sensor accelerometer (*line 55-92*). Pengambilan nilai dari ketiga sensor dilakukan setiap 1 detik. Proses ini dapat dilihat pada Kode Sumber 4.6.

```
1   private void init() {
2     // magnetometer
3     sensorManager1.registerListener(new SensorEventListener() {
4     @Override
5     public void onSensorChanged(SensorEvent sensorEvent) {
6             if(timestampMag ==-1) {
7                 befmagz = sensorEvent.values[0];
8                 magz = befmagz;
9                 timestampMag = sensorEvent.timestamp;
10            }
11            if((sensorEvent.timestamp-
12  timestampMag)*NS2S>=(delay/1000)) {
13                befmagz = magz;
14                magz = sensorEvent.values[0];
15                timestampMag = sensorEvent.timestamp;
16            }
17        }
18
19        @Override
20        public void onAccuracyChanged(Sensor sensor, int i) {
21
```

```
22          }
23      }, magnetometer,delay*1000);
24
25      // gyroscope
26      sensorManager2.registerListener(new SensorEventListener() {
27          @Override
28          public void onSensorChanged(SensorEvent sensorEvent) {
29              if(timestampGyro ==-1) {
30                  gyrox = sensorEvent.values[0];
31                  gyroy = sensorEvent.values[1];
32                  gyroz = sensorEvent.values[2];
33                  x = sensorEvent.values[0];
34                  z = sensorEvent.values[2];
35                  timestampGyro = sensorEvent.timestamp;
36              }
37              if((sensorEvent.timestamp-
38  timestampGyro)*NS2S>=(delay/1000)) {
39                  gyrox = sensorEvent.values[0];
40                  gyroy = sensorEvent.values[1];
41                  gyroz = sensorEvent.values[2];
42                  x = sensorEvent.values[0];
43                  z = sensorEvent.values[2];
44                  timestampGyro = sensorEvent.timestamp;
45              }
46          }
47
48          @Override
49          public void onAccuracyChanged(Sensor sensor, int i) {
50
51          }
52      }, gyroscope, delay*1000);
53
54      // akselerometer
55      sensorManager3.registerListener(new SensorEventListener() {
56          @Override
57          public void onSensorChanged(SensorEvent sensorEvent) {
58              if(timestampAcc ==-1) {
59                  accx = sensorEvent.values[0]-
60  SensorManager.GRAVITY_EARTH;
61                  accy = sensorEvent.values[1]-
62  SensorManager.GRAVITY_EARTH;
63                  accz = sensorEvent.values[2]-
64  SensorManager.GRAVITY_EARTH;
```

```
65                  y = sensorEvent.values[1]-
66  SensorManager.GRAVITY_EARTH;
67                  gforce = Math.sqrt(accx*accx+accy*accy+accz*accz)-
68  SensorManager.GRAVITY_EARTH;
69                  timestampAcc = sensorEvent.timestamp;
70              }
71              if((sensorEvent.timestamp-
72  timestampAcc)*NS2S>=delay/(1000)) {
73                  accx = sensorEvent.values[0]-
74  SensorManager.GRAVITY_EARTH;
75                  accy = sensorEvent.values[1]-
76  SensorManager.GRAVITY_EARTH;
77                  accz = sensorEvent.values[2]-
78  SensorManager.GRAVITY_EARTH;
79                  y = sensorEvent.values[1]-
80  SensorManager.GRAVITY_EARTH;
81                  gforce = Math.sqrt(accx*accx+accy*accy+accz*accz)-
82  SensorManager.GRAVITY_EARTH;
83                  timestampAcc = sensorEvent.timestamp;
84              }
85          }
86
87          @Override
88          public void onAccuracyChanged(Sensor sensor, int i) {
89
90          }
91      }, accelerometer, delay+1000);
92  }
```

Kode Sumber 4.6 Fungsi Inisiasi Sensor

Setelah sensor dipersiapkan dan sudah dilakukan pengaturan, maka proses selanjutnya adalah mengolah data yang telah didapatkan dari sensor untuk menentukan manuver yang terjadi saat berkendara. Kode sumber proses ini dapat dilihat pada Kode Sumber 4.7.

Pertama, sistem akan melakukan pengecekan apakah posisi pengemudi sekarang sudah berada pada alamat tujuan (*line 12-26*). Jika pengemudi belum sampai pada tujuan, maka sistem akan mengambil nilai dari masing-masing sensor (*line 28-69*). Sistem mendeteksi 4 manuver, yaitu belok kiri (*line 29-35*), belok kanan

(*line 36-42*), penambahan kecepatan (*line 44-51*), dan pengurangan kecepatan (*line 52-59*).

```
1   private void drive() {
2      final GPSTracker gpsTracker = new GPSTracker(this);
3      final Handler handler = new Handler();
4      final Runnable runnable = new Runnable() {
5         @SuppressLint("MissingPermission")
6         @Override
7         public void run() {
8            location = gpsTracker.getLocation();
9            boolean turn = false;
10
11           // check if finished
12           if(checkLocation(location.getLatitude(),
13  location.getLongitude())){
14              handler.removeCallbacks(this);
15
16              rea = mdformat.format(calendar.getTime());
17
18              try {
19                 date1 = mdformat.parse(eta);
20                 date2 = mdformat.parse(rea);
21              } catch (ParseException e) {
22                 e.printStackTrace();
23              }
24
25              long dif = (date2.getTime()-date1.getTime())/1000;
26              }
27              // not yet finished
28              else {
29                 if (z>0.02f &&  abs(befmagz-magz)>1.3f) {
30                 turn = true;
31                 if(gforce>7.0f) sudleft++;
32                 left++;
33                 tvLeft.setText(String.valueOf(left));
34                 tvSuddenLeft.setText(String.valueOf(sudleft));
35              }
36              else if (z<- 0.02f && abs(befmagz-magz) >1.3f) {
37                 turn = true;
38                 if(gforce>7.0f) sudright++;
39                 right++;
40                    tvRight.setText(String.valueOf(right));
41                    tvSuddenRight.setText(String.valueOf(sudright));
```

```
42                }
43                if(!turn){
44                   if(accy>0.7f){
45                      if(gforce>6.0f){
46                         sudacc++;acc++;
47                      }
48                      else {
49                         acc++;
50                      }
51                   }
52                else if(accy<-1.7){
53                   if(gforce<1.2f){
54                      suddecc++;
55                      decc++;
56                   }
57                   else {
58                      decc++;
59                   }
60                }
61                   tvAcc.setText(String.valueOf(acc));
62                   tvDecc.setText(String.valueOf(decc));
63                   tvSuddenAcc.setText(String.valueOf(sudacc));
64                   tvSuddenDecc.setText(String.valueOf(suddecc));
65             }
66             handler.postDelayed(this, delay);
67          }
68       }
69    };
      runnable.run();
   }
```

Kode Sumber 4.7 Fungsi Pengambilan Data Sensor

4.3.5 Implementasi Proses Pengolahan Data

Sistem menghitung skor perilaku pengemudi berdasarkan jumlah manuver-manuver yang telah dideteksi oleh sensor berdasarkan bobot yang telah didefiniskan pada Subbab 2.4. Kode sumber dapat dilihat pada Kode Sumber 4.8.

```
1   private Double countScore()
2      {
3         Double score = 0;
```

```
4        Double rights=0, lefts=0, accs=0, deccs=0;
5
6        if(right!=0) rights = (double)(right-sudright)/(double)right;
7        if(left!=0) lefts = (double)(left-sudleft)/(double)left;
8        if(acc!=0) accs = (double)(acc-sudacc)/(double)acc;
9        if(decc!=0) deccs = (double)(decc-suddecc)/(double)decc;
10
11       score = 0.4625*rights+0.4625*lefts+0.058*accs+0.017*deccs;
12       score = score *100;
13
14       Log.d(TAG, "Score: " + score);
15
16       return Double.valueOf(score);
17     }
```

Kode Sumber 4.8 Fungsi Pengolahan Data

4.3.6 Implementasi Proses Menampilkan Skor Pengemudi

Pada Kode Sumber 4.8 nilai dari perilaku pengemudi telah dihitung yang kemudian akan ditampilkan pada halaman skor pengemudi dengan kode yang dapat dilihat pada Kode Sumber 4.9.

```
1   protected void onCreate(Bundle savedInstanceState)
2     {
3        super.onCreate(savedInstanceState);
4        setContentView(R.layout.activity_score);
5
6        Double score = getIntent().getDoubleExtra("score", 0);
7        long dif = getIntent().getLongExtra("dif", 0);
8
9        tvScore = findViewById(R.id.tvScore);
10       tvTime = findViewById(R.id.tvTime);
11
12       tvScore.setText(String.format("%.1f",score));
13
14       if(dif>-60000 && dif<60000)
15          tvTime.setText("You are on time!");
16       else if(dif<=-60000)
17          tvTime.setText("And: "+abs(dif/60000)+" minutes faster");
18       else if(dif>=60000)
19          tvTime.setText("And: "+abs(dif/60000)+" minutes slower");
```

```
}
```

Kode Sumber 4.9 Fungsi Menampilan Skor Pengemudi

4.4 Langkah Penilaian Perilaku

Implementasi langkah penilaian perilaku dilakukan dalam 2 tahap, yaitu menentukan manuver yang terjadi hasil balikan dari sensor lalu melakukan perhitungan berdasarkan rumus pada 2.4.

Contoh hasil balikan sensor dengan visualisasi nilai tiap sensor yang didapat dapat dilihat pada Gambar 4.9 yang kemudian diolah berdasarkan batasan pada Tabel 4.1.

Gambar 4.9 Visualisasi Nilai pada masing-masing Sensor

Tabel 4.1 Batasan Hasil Balikan Manuver

Manuver	Gyrometer	Accelerometer	Magnetometer	G-Force
Belok kiri	z > 0.02f	-	z[1]-z[0] > 1.3f	-
Belok kiri mendadak	z > 0.02f	-	z[1]-z[0] > 1.3f	>7.0f
Belok kanan	z < -0.02f	-	z[1]-z[0] > 1.3f	-
Belok kanan mendadak	z < -0.02f	-	z[1]-z[0] > 1.3f	>7.0f

Manuver	Gyrometer	Accelerometer	Magnetometer	G-Force
Akselerasi	-	y > 0.7	-	-
Akselerasi mendadak	-	y > 0.7	-	>6.0f
Deselerasi	-	y < -1.7	-	-
Deselerasi mendadak	-	y < -1.7	-	<1.2f

Setelah mendapatkan banyaknya nilai masing-manuver pada perjalanan, maka akan dilakukan perhitungan nilai dari pengemudi berdasarkan dengan hasil pada aplikasi dapat dilihat pada Gambar 4.10.

Gambar 4.10 Contoh Hasil Tampilan Banyaknya Manuver pada Aplikasi

Dari Gambar 4.10, diketahui manuver yang dilakukan pada perjalanan seperti pada Tabel 4.2.

Tabel 4.2 Contoh Data Hasil Manuver pada Perjalanan

Manuver	Mendadak	Normal
Belok Kiri	1	0
Belok Kanan	1	1

Akselerasi	0	0
Deselerasi	1	2

Berdasarkan rumus yang terdapat pada 2.4, maka didapatkan nilai tiap manuver adalah:

$$belok_kiri = \frac{0}{(0+1)} = 0 \qquad akselerasi = \frac{0}{(0+0)} = 0$$

$$belok_kanan = \frac{1}{(1+1)} = 0.5 \qquad deselerasi = \frac{2}{(2+1)} = 0.67$$

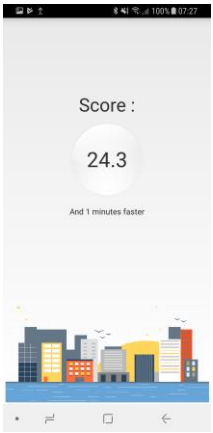
Setelah didapatkan nilai masing-masing manuver, maka akan dihitung nilai total dengan mengalikan nilai masing-masing manuver dengan bobot masing-masing.

Nilai dari pengemudi dihitung sesuai dengan rumus *score* yang terdapat pada bab 2.4, sehingga didapatkan nilai dari pengemudi adalah:

$$score = 100 * ((0 * 0.4625) + (0.5 * 0.4625) + (0 * 0.058) + (0.67 * 0.017))$$

$$score = 100 * (0 + 0.23125 + 0 + 0.1139) = 24.264 = 24.3$$

Sesuai dengan perhitungan, maka nilai akhir yang diperoleh dari pengemudi adalah 24.3 sesuai dengan tampilan pada Gambar 4.11. Perhitungan perbedaan waktu sampai dan perkiraan waktu dapat dihitung dengan mencari selisih antara kedua waktu. Sesuai Gambar 4.10, perkiraan waktu sampai adalah pukul 07:28 sedangkan waktu ketika selesai mengemudi adalah 07:27 sehingga didapatkan bahwa pengemudi menyelesaikan perjalanan dengan waktu tempuh satu menit lebih cepat.



Gambar 4.11 Contoh Hasil Tampilan Skor pada Aplikasi

BAB V
UJI COBA DAN EVALUASI

Pada bab ini akan dilakukan tahap uji coba dan evaluasi Aplikasi *Scoring* Perilaku Pengemudi. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas yang telah dijabarkan pada Bab III dan berkendara pada tiga macam kondisi lalulintas dan jalan. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1 Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

- Kendaraan :
  1. Toyota Yaris
  2. Toyota Corolla Altis
  3. Nissan Teana
  4. Toyota Fortuner
- Handphone :
  1. Samsung Galaxy A7
  2. Samsung Galaxy C9 Pro
- Lokasi :
  1. Perjalanan dengan kondisi lalu lintas ramai
  2. Perjalanan dengan kondisi lalu lintas sepi

5.2 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan pada aplikasi *scoring* perilaku pengemudi. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas serta pengujian langsung pada kendaraan dengan dua kondisi jalan, yaitu lalulintas padat dan lalulintas sepi. Pengujian juga dilakukan dengan empat mobil, dua handphone, dan empat user yang berbeda.

5.2.1 Pengujian Fungsionalitas

Pengujian kebutuhan fungsional dilakukan oleh dua orang dengan menggunakan metode *black box*. Metode *black box* adalah metode di mana pengujian ditekankan pada pola masukan dan keluaran yang sesuai dengan skenario pada Tabel 3.3 hingga Tabel 3.8. Pengujian yang dilakukan mengacu pada kasus penggunaan yang dijelaskan pada Bab III.

5.2.2 Pengujian Perjalanan dengan Kondisi Lalu Lintas Padat

Pengujian perjalanan dengan kondisi lalulintas padat dilakukan oleh seorang pengemudi dengan lingkungan pengujian sesuai pada subbab 5.1. Lokasi perjalanan dilakukan dari Jl. Dharmahusada Indah I no 42 menuju ke Depot Bu Rudy. Perjalanan dilakukan dengan jarak tempuh 1.3 kilometer dan waktu tempuh 4 menit. Perjalanan dilakukan selama 10 kali dengan rute, pengemudi, dan kondisi lingkungan pengujian yang sama untuk melihat stabilitas nilai yang dihasilkan oleh aplikasi.

5.2.3 Pengujian Perjalanan dengan Kondisi Lalu Lintas Sepi

Pengujian perjalanan dengan kondisi lalulintas padat dilakukan oleh seorang pengemudi dengan lingkungan pengujian sesuai pada subbab 5.1. Lokasi perjalanan dilakukan dengan melakukan satu putaran pada Jalan Laguna Raya Kejawen Putih. Perjalanan dilakukan dengan jarak tempuh 1.5 kilometer dan waktu tempuh 3 menit. Perjalanan dilakukan selama 10 kali dengan rute, pengemudi, dan kondisi lingkungan pengujian yang sama untuk melihat stabilitas nilai yang dihasilkan oleh aplikasi.

5.2.4 Pengujian dengan Mobil Berbeda

Pengujian perjalanan dengan mobil berbeda dilakukan ole seorang pengemudi pada rute yang sama dan dalam kondisi lalulintas normal sesuai pada subbab 5.1. Pengujian dilakukan dengan mengemudi dari depan Global Sevilla School ke SMPN 99

Jakarta. Perjalanan dilakukan sebanyak satu kali pada masing-masing tipe mobil yang diuji.

5.2.5 Pengujian dengan Handphone Berbeda

Pengujian perjalanan dengan handphone berbeda dilakukan oleh seorang pengemudi pada rute yang sama dan dalam kondisi lalulintas normal sesuai pada subbab 5.1. Pengujian dilakukan dengan mengemudi dari depan Global Sevilla School ke SMPN 99 Jakarta. Perjalanan dilakukan sebanyak satu kali dengan dua *handphone*.

5.2.6 Pengujian dengan Mobil dan Handphone Berbeda

Pengujian perjalanan dengan mobil dan handphone berbeda dilakukan pada rute yang sama dalam kondisi lalulintas normal dilakukan oleh seorang pengemudi sesuai pada subbab 5.1. Pengujian dilakukan dengan mengemudi dari depan Global Sevilla School ke SMPN 99 Jakarta. Perjalanan dilakukan sebanyak 3 kali dengan mobil yang berbeda dan dua *handphone* pada tiap perjalanan.

5.2.7 Pengujian dengan User Berbeda

Pengujian perjalanan dengan user dilakukan oleh empat user berbeda. Pengujian dilakukan di sekitar Jalan Kertajaya, Jalan Raya ITS, dan Manyar dengan kondisi lalu lintas normal-ramai. Perjalanan dilakukan sebanyak satu kali pada masing-masing user.

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional pengujian langsung pada kendaraan dengan dua kondisi jalan yang berbeda, yaitu lalulintas padat dan lalulintas sepi, dua hadphone yang berbeda, empat mobil yang berbeda, dan mobil dan handphone yang berbeda.

5.3.1 Evaluasi Pengujian Fungsionalitas

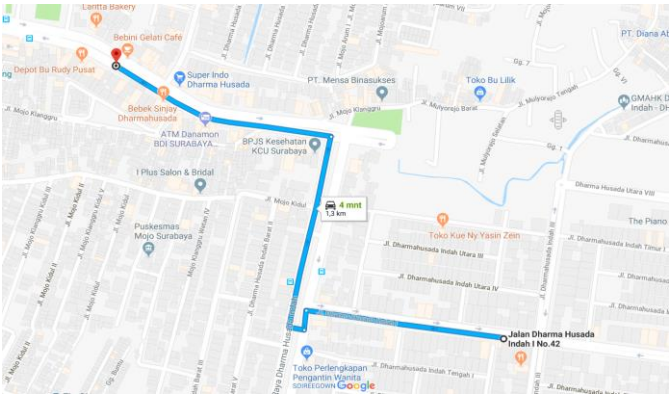
Hasil pengujian fungsionalitas secara keseluruhan dapat dilihat pada Tabel 5.1. Data berdasarkan tabel tersebut menunjukkan bahwa semua skenario pengujian berhasil dan program berjalan dengan baik sehingga pada Tabel 5.1 dapat ditarik kesimpulan bahwa fungsionalitas dari aplikasi bekerja sesuai dengan yang diharapkan.

Tabel 5.1 Evaluasi Pengujian Fungsionalitas

Kode Kasus Penggunaan	Kasus Penggunaan	Hasil
SP -001	Melihat *score* pengemudi	Berhasil
SP -002	Melihat rekam jejak pengemudi	Berhasil
SP -003	Melihat status perjalanan normal	Berhasil
SP -004	Melihat status perjalanan agresif	Berhasil
SP -005	Sensor mendeteksi rotasi perangkat	Berhasil
SP -006	Sensor mendeteksi percepatan	Berhasil
SP -007	Mengukur magnetisasi dan arah medan magnet	Berhasil

5.3.2 Evaluasi Pengujian Perjalanan dengan Kondisi Lalu Lintas Padat

Pengujian pada perjalanan dengan kondisi lalulintas padat dilakukan pada rute pada Gambar 5.1. Jarak tempuh adalah 1,3 kilometer dengan perkiraan waktu tempuh 4 menit.



Gambar 5.1 Rute Pengujian Perjalanan dengan Kondisi Padat

Pengujian dilakukan dengan melakukan 10 kali perjalanan dengan pengemudi yang sama untuk melihat stabilitas aplikasi. Kelemahan dari percobaan ini adalah gaya pengemudi yang berbeda-beda pada setiap percobaan. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi dengan hasil seperti pada Tabel 5.2.

Tabel 5.2 Hasil Uji Perjalanan dengan Kondisi Lalulintas Padat

Uji 1: Tempat Ramai									
Uji ke-	**Left**	**sudden left**	**Right**	**sudden right**	**acc**	**sudden acc**	**decc**	**sudden decc**	**score**
1	2	0	3	1	82	12	9	0	**83.73**
2	2	0	7	1	94	20	11	0	**92.16**
3	3	0	2	1	57	6	12	3	**75.84**
4	3	0	4	0	99	24	25	5	**98.25**
5	6	0	2	1	105	21	27	1	**75.65**
6	3	0	5	0	102	21	18	2	**98.62**
7	1	1	1	0	76	10	10	1	**52.82**
8	3	1	3	0	99	19	27	5	**83.16**
9	3	0	4	0	88	19	22	2	**98.59**
10	5	1	3	1	104	17	15	4	**73.93**

Dari data tersebut, didapatkan bahwa data pada belok kiri, belok kiri mendadak, belok kanan, belok kanan mendadak, dan pengurangan kecepatan mendadak memiliki sebaran nilai 0-10, sehingga perlu dilakukan transformasi data untuk melakukan

perhitungan koefisien variasi. Hasil dari transformasi data dapat dilihat pada Tabel 5.3.

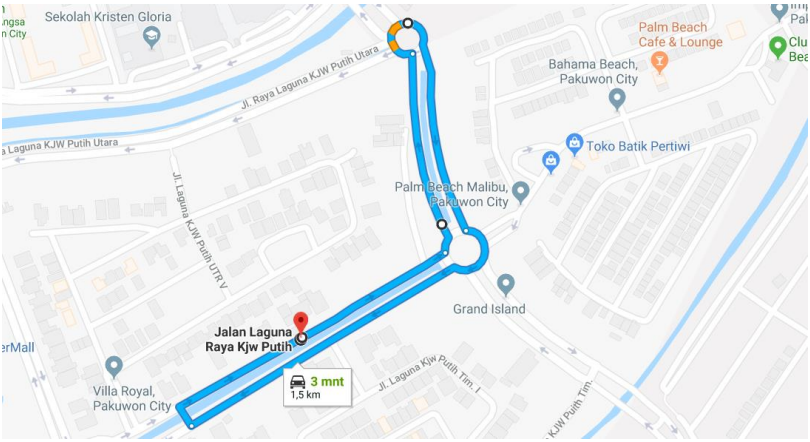
Tabel 5.3 Transformasi Hasil Uji Perjalanan dengan Kondisi Lalulintas Padat

Uji 1: Tempat Ramai (Transformed)									
Uji ke-	**Left**	**sudden left**	**Right**	**sudden right**	**acc**	**sudden acc**	**decc**	**sudden decc**	**score**
1	2	0.71	1.87	1.22	82	12	9	0.71	**83.73**
2	2	0.71	2.74	1.22	94	20	11	0.71	**92.16**
3	3	0.71	1.58	1.22	57	6	12	1.87	**75.84**
4	3	0.71	2.12	0.71	99	24	25	2.35	**98.25**
5	6	0.71	1.58	1.22	105	21	27	1.22	**75.65**
6	3	0.71	2.35	0.71	102	21	18	1.58	**98.62**
7	1	1.22	1.22	0.71	76	10	10	1.22	**52.82**
8	3	1.22	1.87	0.71	99	19	27	2.35	**83.16**
9	3	0.71	2.12	0.71	88	19	22	1.58	**98.59**
10	5	1.22	1.87	1.22	104	17	15	2.12	**73.93**
Mean	3.1	0.86	1.93	0.97	90.60	16.90	17.60	1.57	83.28
Standar Deviasi	1.45	0.25	0.43	0.27	15.25	5.70	7.18	0.61	14.54
Koefisien Variasi	46.75	28.99	22.16	28.24	16.83	33.76	40.81	38.69	17.46

Dari data hasil transformasi tersebut, didapatkan bahwa nilai koefisien variasi dari tiap manuver kurang dari 100% dan nilai rata-rata koefisien variasi dari seluruh manuver adalah 30,41% sehingga data dapat dianggap belum stabil. Dari percobaan ini ditemukan gaya pengemudi yang berbeda-beda pada setiap percobaan sehingga data yang didapat kurang maksimal.

5.3.3 Evaluasi Pengujian Perjalanan dengan Kondisi Lalu Lintas Sepi

Pengujian pada perjalanan dengan kondisi lalulintas sepi dilakukan pada rute pada Gambar 5.2. Jarak tempuh adalah 1,5 kilometer dengan perkiraan waktu tempuh 3 menit.



Gambar 5.2 Rute Pengujian Perjalanan dengan Kondisi Sepi

Pengujian dilakukan dengan melakukan 10 kali perjalanan dengan pengemudi yang sama untuk melihat stabilitas aplikasi. Kelemahan dari percobaan ini adalah kondisi jalan dan gaya pengemudi yang berbeda-beda pada setiap percobaan. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi dengan hasil seperti pada Tabel 5.4.

Tabel 5.4 Hasil Uji Perjalanan dengan Kondisi Lalulintas Sepi

Uji 2: Tempat Sepi									
Uji ke-	**Left**	**sudden left**	**Right**	**sudden right**	**acc**	**sudden acc**	**decc**	**sudden decc**	**score**
1	6	1	11	1	78	15	14	1	**86.85**
2	3	1	12	0	57	7	11	1	**83.72**
3	6	2	8	1	72	18	6	0	**77.35**
4	3	0	8	0	75	15	7	0	**98.84**
5	5	1	16	1	81	15	8	1	**86.57**
6	10	3	9	0	109	89	16	3	**81.07**
7	7	1	14	1	71	11	9	1	**89.00**
8	3	0	10	1	85	20	21	3	**93.77**
9	5	1	8	0	98	22	18	1	**89.35**
10	1	0	16	2	99	18	16	2	**92.95**

Dari data tersebut, didapatkan bahwa data pada belok kiri, belok kiri mendadak, belok kanan mendadak, dan pengurangan kecepatan mendadak memiliki sebaran nilai 0-10, sehingga perlu

dilakukan transformasi data untuk melakukan perhitungan koefisien variasi. Hasil dari transformasi data dapat dilihat pada Tabel 5.5.

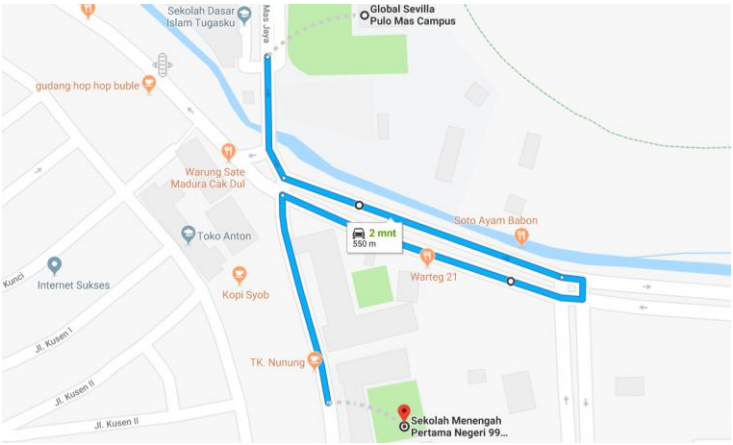
Tabel 5.5 Transformasi Hasil Uji Perjalanan dengan Kondisi Lalulintas Sepi

Uji 2: Tempat Sepi (Transformed)									
Uji ke-	**Left**	**sudden left**	**Right**	**sudden right**	**acc**	**sudden acc**	**decc**	**sudden decc**	**score**
1	2.55	1.22	11	1.22	78	15	14	1.22	**86.85**
2	1.87	1.22	12	0.71	57	7	11	1.22	**83.72**
3	2.55	1.58	8	1.22	72	18	6	0.71	**77.35**
4	1.87	0.71	8	0.71	75	15	7	0.71	**98.84**
5	2.35	1.22	16	1.22	81	15	8	1.22	**86.57**
6	3.24	1.87	9	0.71	109	89	16	1.87	**81.07**
7	2.74	1.22	14	1.22	71	11	9	1.22	**89.00**
8	1.87	0.71	10	1.22	85	20	21	1.87	**93.77**
9	2.35	1.22	8	0.71	98	22	18	1.22	**89.35**
10	1.22	0.71	16	1.58	99	18	16	1.58	**92.95**
Mean	2.26	1.17	11.2	1.05	82.5	23	12.6	1.28607	87.95
Standar Deviasi	0.57	0.38	3.19	0.32	15.61	23.59	5.13	0.40	6.46
Koefisien Variasi	25.10	32.72	28.48	30.10	18.92	**102.56**	40.68	31.32	7.34

Dari data hasil transformasi tersebut, didapatkan bahwa nilai koefisien variasi dari tiap manuver kurang dari 100%, kecuali pada manuver penambahan kecepatan mendadak. Nilai rata-rata koefisien variasi dari seluruh manuver adalah 35,25% sehingga data dapat dianggap belum stabil. Dari percobaan ini ditemukan pengemudi yang berbeda-beda pada setiap percobaan sehingga data yang didapat kurang maksimal.

5.3.4 Evaluasi Pengujian dengan Mobil Berbeda

Pengujian pada perjalanan dengan kondisi lalulintas sepi dilakukan pada rute pada Gambar 5.3. Jarak tempuh adalah 550 meter dengan perkiraan waktu tempuh 2 menit.



Gambar 5.3 Rute Pengujian Perjalanan Global Seviila School - SMPN 99 Jakarta

Pengujian dilakukan dengan menggunakan tiga mobil berbeda dengan masing-masing satu kali perjalanan dengan pengemudi yang sama untuk melihat apakah mobil yang berbeda memberikan hasil pengukuran yang sama. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi dengan hasil seperti pada Tabel 5.6.

Tabel 5.6 Hasil Uji Perjalanan dengan Mobil Berbeda

UJI 3: Mobil									
UJI	Left	sudden left	Right	sudden right	acc	sudden acc	decc	sudden decc	score
Nissan Teana	5	0	7	0	21	0	1	0	100.00
Toyota Fortuner	6	1	7	0	26	3	2	0	91.62
Toyota Altis	4	1	4	0	32	1	1	0	88.26

Dari data tersebut, didapatkan bahwa selain data percepatan seluruh data lainnya memiliki sebaran nilai 0-10, sehingga perlu dilakukan transformasi data untuk melakukan perhitungan koefisien variasi. Hasil dari transformasi data dapat dilihat pada Tabel 5.7.

Tabel 5.7 Transformasi Hasil Uji Perjalanan dengan Mobil Berbeda

UJI 3: Mobil (Transformed)									
UJI	Left	sudden left	Right	sudden right	acc	sudden acc	decc	sudden decc	score
Nissan Teana	2.35	0.71	2.74	0.71	21	0.71	1.22	0.71	100.00
Toyota Fortuner	2.55	1.22	2.74	0.71	26	1.87	1.58	0.71	91.62
Toyota Altis	2.12	1.22	2.12	0.71	32	1.22	1.22	0.71	88.26
Mean	2.34	1.05	2.53	0.71	26.3333	1.27	1.34	0.71	93.29
Standar Deviasi	0.21	0.30	0.36	0.00	5.51	0.58	0.21	0.00	5.92
Koefisien Variasi	9.16	28.40	14.07	0.00	20.91	46.00	15.32	0.00	6.35

Dari percobaan ini, didapatkan bahwa nilai koefisien variasi dari seluruh manuver kurang dari 100% dan nilai rata-rata koefisien variasi dari seluruh manuver adalah 15,58% sehingga data dapat dianggap stabil pada percobaan dengan mobil berbeda.

5.3.5 Evaluasi Pengujian dengan Handphone Berbeda

Pengujian pada perjalanan dengan handphone berbeda dilakukan pada rute pada Gambar 5.3 satu kali dengan mobil yang sama dan dua handphone dengan aplikasi yang dijalankan secara bersamaan. Kedua handphone diletakkan berdampingan pada dua buah *phone handle* yang direkatkan pada kaca depan mobil. Pengujian dilakukan untuk melihat apakah penggunaan handphone yang berbeda akan memberikan hasil pengukuran yang sama. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi dengan hasil seperti pada Tabel 5.8.

Tabel 5.8 Hasil Uji Perjalanan dengan Handphone Berbeda

Uji 4: HP									
Uji ke-	Left	sudden left	Right	sudden right	acc	sudden acc	decc	sudden decc	score
1	4	1	4	0	32	1	1	0	88.26
2	4	1	5	0	34	1	4	0	88.27

Dari data tersebut, didapatkan bahwa selain data percepatan seluruh data lainnya memiliki sebaran nilai 0-10, sehingga perlu dilakukan transformasi data untuk melakukan perhitungan koefisien variasi. Hasil dari transformasi data dapat dilihat pada Tabel 5.9.

Tabel 5.9 Transformasi Hasil Uji Perjalanan dengan Handphone Berbeda

Uji 4: HP (Transformed)									
Uji ke-	Left	sudden left	Right	sudden right	acc	sudden acc	decc	sudden decc	score
1	2.12	1.22	2.12	0.71	32	1.22	1.22	0.71	88.26
2	2.12	1.22	2.35	0.71	34	1.22	2.12	0.71	88.27
Mean	2.12	1.22	2.23	0.71	33	1.22	1.67	0.71	88.26
Standar Deviasi	0.00	0.00	0.16	0.00	1.41	0.00	0.63	0.00	0.24
Koefisien Variasi	0.00	0.00	7.09	0.00	4.29	0.00	37.89	0.00	0.27

Dari data tersebut, didapatkan bahwa nilai koefisien variasi dari seluruh manuver kurang dari 100% dan nilai rata-rata koefisien variasi dari seluruh manuver adalah 5,50% sehingga data dapat dianggap stabil pada percobaan dengan *handphone* berbeda.

5.3.6 Evaluasi Pengujian Perjalanan dengan Mobil dan Handphone yang Berbeda

Pengujian pada perjalanan dengan handphone berbeda dilakukan pada rute pada Gambar 5.3. Pengujian dilakukan dengan menggunakan tiga mobil yang berbeda dan dua handphone dengan aplikasi yang dijalankan secara bersamaan. Pengujian dilakukan untuk melihat apakah penggunaan handphone dan mobil yang berbeda akan memberikan hasil pengukuran yang sama. Handphone diletakkan pada holder dan ditempel ke kaca depan mobil. P1 adalah percobaan dengan menggunakan Samsung Galaxy A7, P2 adalah percobaan dengan menggunakan Samsung Galaxy C9 Pro. Uji coba dilakukan satu kali pada masing-masing mobil. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi dengan hasil seperti pada Tabel 5.10.

Tabel 5.10 Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda

Manuver / Mobil	TOYOTA FORTUNER		NISSAN TEANA		TOYOTA COROLLA ALTIS	
	P1	P2	P1	P2	P1	P2
left	6	5	5	4	4	4
sudden left	1	0	0	0	1	1
right	7	4	7	3	4	5
sudden right	0	0	0	0	0	0
acceleration	26	31	21	17	32	34
sudden acc	3	2	0	1	1	1
decceleration	2	6	1	1	1	4
sudden dec	0	0	0	0	0	0
SCORE	91.62	99.63	100.00	99.66	88.26	88.27

Dari data tersebut, didapatkan bahwa selain data percepatan seluruh data lainnya memiliki sebaran nilai 0-10, sehingga perlu dilakukan transformasi data untuk melakukan perhitungan koefisien variasi. Hasil dari transformasi data dapat dilihat pada Tabel 5.11 sedangkan hasil pengolahan data dari hasil transformasi data tersebut dapat dilihat pada Tabel 5.12.

Tabel 5.11 Transformasi Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda (1)

Manuver / Mobil	TOYOTA FORTUNER		NISSAN TEANA		TOYOTA COROLLA ALTIS	
	P1	P2	P1	P2	P1	P2
left	2.55	2.35	2.35	2.12	2.12	2.12
sudden left	1.22	0.71	0.71	0.71	1.22	1.22
right	2.74	2.12	2.74	1.87	2.12	2.35
sudden right	0.71	0.71	0.71	0.71	0.71	0.71
acceleration	26	31	21	17	32	34
sudden acc	1.87	1.58	0.71	1.22	1.22	1.22
decceleration	1.58	2.55	1.22	1.22	1.22	2.12
sudden dec	0.71	0.71	0.71	0.71	0.71	0.71
SCORE	88.26	88.27	0.00	0.00	56.68	58.58

Tabel 5.12 Transformasi Hasil Uji Perjalanan dengan Mobil dan Handphone yang Berbeda (2)

Manuver / Mobil	TOYOTA FORTUNER					NISSAN TEANA					TOYOTA COROLLA ALTIS					AVG CV
	P1	P2	AVG	SD	CV	P1	P2	AVG	SD	CV	P1	P2	AVG	SD	CV	
left	2.55	2.35	2.45	0.14	5.90	2.35	2.12	2.23	0.16	7.09	2.12	2.12	2.12	0.00	0.00	4.33
sudden left	1.22	0.71	0.97	0.36603	37.89	0.71	0.71	0.71	0	0	1.22	1.22	1.22	0	0.00	12.63
right	2.74	2.12	2.43	0.44	17.96	2.74	1.87	2.30	0.61	26.62	2.12	2.35	2.23	0.16	7.09	17.23
sudden right	0.71	0.71	0.71	0	0.00	0.71	0.71	0.71	0	0	0.71	0.71	0.71	0	0.00	0.00
acceleration	26.00	31.00	28.50	3.54	12.41	21.00	17.00	19.00	2.83	14.89	32.00	34.00	33.00	1.41	4.29	10.53
sudden acc	1.87	1.58	1.73	0.20	11.87	0.71	1.22	0.97	0.37	37.89	1.22	1.22	1.22	0.00	0.00	16.59
decceleration	1.58	2.55	2.07	0.68474	33.15	1.22	1.22	1.22	0	0	1.22	2.12	1.67	0.63	37.89	23.68
sudden dec	0.71	0.71	0.71	0	0.00	0.71	0.71	0.71	0	0	0.71	0.71	0.71	0	0.00	0.00
SCORE	91.62	99.63	95.62	5.66	5.92	100.00	99.66	99.83	0.24	0.24	88.26	88.27	88.26	0.01	0.01	2.06

Dari data tersebut, didapatkan bahwa nilai koefisien variasi dari tiap manuver kurang dari 100% dan nilai rata-rata koefisien variasi dari seluruh manuver adalah 9,67% sehingga data dapat dianggap stabil.

5.3.7 Evaluasi Pengujian Perjalanan User Berbeda

Pengujian pada perjalanan dengan kondisi lalulintas normal – ramai. Pengujian dilakukan dengan menggunakan empat pengemudi berbeda dengan masing-masing satu kali perjalanan dengan mobil yang sama untuk melihat apakah skor yang diberikan oleh aplikasi sesuai dengan perilaku pengemudi. Data yang dievaluasi adalah keluaran manuver dan skor pengemudi serta kecenderungan perilaku mengemudi seperti pada Tabel 5.13.

Tabel 5.13 Hasil Uji Perjalanan dengan User Berbeda

UJI COBA DENGAN USER				
Nama	Fourir. A	Faisal. I	Rizky. A	Aldo. R
cara mengemudi	agresif	normal	normal	normal
left	5	1	2	6
sudden left	3	0	1	1
right	3	6	3	6
sudden right	0	0	0	1
acceleration	161	85	74	187
sudden acc	54	11	6	46
decceleration	51	6	2	24
sudden dec	4	0	1	1
SCORE	70.2	99.2	76.4	86.2

Dari data tersebut, didapatkan bahwa aplikasi mampu memberikan nilai yang cukup sesuai dengan cara mengemudi dari tiap pengemudi kecuali pada pengemudi Rizky. A karena nilai yang

diberikan oleh aplikasi terlalu kecil sehingga akurasi dari aplikasi pada uji coba dengan pengemudi yang berbeda adalah 75%.

(Halaman ini sengaja dikosongkan)

BAB VI
KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

Setelah melakukan perancangan, implementasi, uji coba, dan evaluasi, dapat diambil kesimpulan sebagai berikut:

1. Faktor-faktor perilaku pengemudi yang dapat diamati oleh sensor gyroscope, accelerometer, dan magnetometer adalah belok secara normal dan agresif, akselerasi secara normal maupun agresif, serta pengurangan kecepatan secara normal dan agresif.
2. Dari hasil pengujian yang sudah dilakukan, manuver dikelompokkan menjadi normal dan agresif dengan metode *thresholding*. Manuver belok kanan secara normal tercatat ketika nilai Gyro z>0.02, belok kiri secara normal tercatat ketika nilai Gyro z<-0.02, akselerasi secara normal tercatat ketika nilai akselerometer y>0.07, deselerasi secara normal tercatat ketika nilai akselerometer y<-1.7. Manuver agresif dideteksi dengan menggunakan nilai G-force. Belok kanan secara mendadak tercatat ketika nilai Gyro z>0.02 dan G-force>7.0, belok kiri secara agresif tercatat ketika nilai Gyro z<-0.02 dan G-force>7.0, akselerasi secara agresif tercatat ketika nilai akselerometer y>0.07 dan G-force>0.06, deselerasi secara agresif tercatat ketika nilai akselerometer y<-1.7 dan G-force<1.2.
3. Uji coba pada lalu lintas sepi memiliki stabilitas paling rendah dengan rata-rata koefisien variasi 35,25% dan memiliki satu manuver yang tidak stabil, yaitu penambahan kecepatan mendadak dengan koefisien

variasi 102,56% (Tabel 5.5). Hal ini dipengaruhi oleh gaya pengemudi yang berbeda pada setiap percobaan.

4. Menurut evaluasi yang dilakukan, didapatkan bahwa nilai rata-rata koefisien variasi dari seluruh manuver adalah 19,28 % sehingga data stabil.

6.2 Saran

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

1. Melakukan uji coba dengan meminimalisir faktor yang mempengaruhi ketidakstabilan data dan meningkatkan stabilitas aplikasi pada percobaan aplikasi dengan kondisi jalan berbeda.
2. Visualisasi dimana pengemudi melakukan manuver.
3. Menambah kamera untuk mendeteksi jarak aman dengan kendaraan di depan.
4. Menambah kamera untuk mendeteksi ekspresi wajah saat mengemudi.

DAFTAR PUSTAKA

[1] D. Bansal dan N. Kalra, “Analyzing Driver Behavior using Smartphone Sensors: A,” *International Journal of Electronic and Electrical Engineering,* vol. 7, no. 7, pp. 697-702, 2014.

[2] H. S. Martha Putra, D. Purwitasari dan R. V. H. Ginardi, “Sistem Akuisisi Data untuk Membantu Memetakan Kualitas Angkutan Umum dengan Perangkat Android,” *Jurnal Teknik ITS,* vol. 4, 2015.

[3] M. M. Trivedi dan D. A. Johnson, “Driving Style Recognition Using a Smartphone as a Sensor Platform,” *International IEEE Conference onIntelligent Transportation Systems,* pp. 1609-1615, 2011.

[4] Gunawan, “Pengertian Accelerometer, Gyroscope, Proximity, Ambient Light, Digital Compass - haiwiki.info,” [Online]. Available: https://haiwiki.info/teknologi/sensor-smartphone-android/.

[5] A. Developers, “Motion Sensors | Android Developers,” [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion.

[6] A. Developers, “Position Sensors | Android Developers,” [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_position.

[7] S. Anthony, “Indoor navigation on your smartphone, using the Earth’s magnetic field — just like a pigeon - ExtremeTech,” [Online]. Available: http://www.extremetech.com/computing/132484-indoor-navigation-on-your-smartphone-using-the-earths-magnetic-field-just-like-a-homing-pigeon.

[8] Wikipedia, “Advanced driver-assistance systems - Wikipedia,”

[Online]. Available: https://en.wikipedia.org/wiki/Advanced_driver-assistance_systems.

[9] J. Laukkonen, “13 Advanced Driver Assistance Systems,” [Online]. Available: https://www.lifewire.com/advanced-driver-assistance-systems-534859.

[10] A. Shaout, D. Colella dan S. A. , “Advanced Driver Assistance Systems - Past, Present and Future,” 2015.

[11] Kruzr, “Kruzr - The Future of Driving,” Kruzr, [Online]. Available: http://www.kruzr.co/#/.

[12] S. Chowdhary, “App review: Kruzr Android app is a smart driving assistant - Financial Express,” Financial Express, [Online]. Available: https://www.financialexpress.com/industry/technology/app-review-kruzr-android-app-is-a-smart-driving-assistant/878493/.

[13] Autobytel.com, “10 Top Cars with Driver Assist Features | Autobytel.com,” [Online]. Available: https://www.autobytel.com/car-buying-guides/features/10-top-cars-with-driver-assist-features-132049/.

[14] Wikipedia, “Coefficient of Variation - Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Coefficient_of_variation.

[15] M. O’Connor, “Coefficient of Variation - Lean Math,” [Online]. Available: http://www.leanmath.com/blog-entry/coefficient-variation.

[16] A. Hidayat, “Pengertian Dan Jenis Transformasi Data - Uji Statistik,” [Online]. Available: https://www.statistikian.com/2013/01/transformasi-data.html.

[17] “SURVEY STATISTICAL CONFIDENCE: HOW MANY IS ENOUGH?,” GreatBrook, [Online]. Available:

https://greatbrook.com/survey-statistical-confidence-how-many-is-enough. [Diakses Januari 2018].

[18] "Survey Random Sample Calculator," custom insight, [Online]. Available: https://www.custominsight.com/articles/random-sample-calculator.asp. [Diakses 6 Januari 2018].

[19] B. P. Statistik, "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949-2016," [Online]. Available: https://www.bps.go.id/linkTableDinamis/view/id/1133.

[20] K. N. K. Transportasi, "Data Investigasi Kecelakaan LLAJ tahun 2010-2016," 31 November 2016. [Online]. Available: http://knkt.dephub.go.id/knkt/ntsc_home/Media_Release/Media%20Release%20KNKT%202016/Media%20Release%202016%20-%20IK%20LLAJ%2020161130.pdf.

[21] K. L. K. N. R. Indonesia, "KORLANTAS POLRI - Vehicle Type," [Online]. Available: http://www.korlantas-irsms.info/graph/vehicleTypeData.

[22] K. L. K. N. R. Indonesia, "KORLANTAS POLRI - Accident Count," [Online]. Available: http://www.korlantas-irsms.info/graph/accidentData.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Penulis, **Anindita Larasati**, lahir di Jakarta, 19 Juni 1996. Penulis merupakan seorang mahasiswi yang sedang menempuh studi di Departemen Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS).

Selama menempuh pendidikan di ITS, penulis aktif menjadi administrator Laboratorium Pemrograman (LP) Teknik Informatika ITS. Selain menjadi administrator, penulis juga aktif di beberapa organisasi institut seperti anggota ITSTV pada tahun pertama, wakil ketua OPREC ITSTV pada tahun kedua, dan ketua divisi HRM ITSTV pada tahun ketiga dan keempat. Organisasi institut selain ITSTV adalah sebagai sekertaris di BK-MWA-WM (Badan Koordinasi Majelis Wali Amanat Wakil Mahasiswa), staff pada beberapa event LMB (Lembaga Minat Bakat), anggota dan staff UKTK (Unit Kegiatan Tari dan Karawitan) ITS. Penulis juga aktif dalam beberapa organisasi yang ada di jurusan, seperti staff media informasi HMTC (Himpunan Mahasiswa Teknik Computer-Informatika) dan Schematics ITS, serta staff ahli pada tahun selanjutnya di Media Informasi HMTC.

Di luar organisasi, penulis juga aktif menjadi MC (Master of Ceremony) baik di dalam maupun luar kampus, pemateri di beberapa pelatihan media kampus, presenter di ITSTV, serta aktif sebagai presenter TV lokal Surabaya SBOTV yang merupakan salah satu dari grup Jawapos.

Penulis dapat dihubungi melalui email: aninditagltm@gmail.com.