



**TUGAS AKHIR - EC184801**

**PENGEMBANGAN SIMULATOR MENGEMUDI UNTUK  
RISET DETEKSI PENGEMUDI MENGANTUK**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Dosen Pembimbing  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniarno, ST., MT.

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

*Halaman ini sengaja dikosongkan.*



**TUGAS AKHIR - EC184801**

**PENGEMBANGAN SIMULATOR MENGEMUDI UNTUK  
RISET DETEKSI PENGEMUDI MENGANTUK**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Dosen Pembimbing  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniarno, ST., MT.

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

*Halaman ini sengaja dikosongkan.*



**FINAL PROJECT - EC184801**

**DRIVING SIMULATOR DEVELOPMENT FOR DROWSY  
DRIVER DETECTION RESEARCH**

Muhammad Nur Ady Maulana  
NRP 0721 16 4000 0035

Advisors  
Ahmad Zaini, ST., M.Sc.  
Dr. Eko Mulyanto Yuniaro, ST., MT.

Department of Computer Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2020

*Halaman ini sengaja dikosongkan.*

## **LEMBAR PENGESAHAN**

### **Pengembangan Simulator Mengemudi Untuk Riset Deteksi Pengemudi Mengantuk**

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Muhammad Nur Ady Maulana (NRP: 07211640000035)

Tanggal Ujian : 09 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh :

Ahmad Zaini, ST., MSc. (Pembimbing I)  
NIP: 1197504192002121003

Dr. Eko Mulyanto Yuniarno, ST., MT. (Pembimbing II)  
NIP: 196806011995121009

Diah Puspito Wulandari, ST., M.Sc. (Penguji I)  
NIP. 198012192005012001

Susi Juniastuti, S.T.,M.Eng (Penguji II)  
NIP. 196506181999032001

Eko Pramunanto, S.T., M.T. (Penguji III)  
NIP. 196612031994121001

Mengetahui  
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.  
NIP. 197003131995121001

## **ABSTRAK**

Nama Mahasiswa	:	Hafizh Fauzan
Judul Tugas Akhir	:	Deteksi Jatuh pada Manusia Lanjut Usia Berbasis 3D - CNN pada Sistem Tertanam
Pembimbing	:	1. Dr. Reza Fuad Rachmadi, S.T., M.T. 2. Arief Kurniawan, ST., MT.

Jatuh adalah penanda kelemahan, imobilitas, dan gangguan kesehatan akut dan kronis pada orang tua. Bahkan ketika cederanya tidak begitu serius, lansia seringkali kesulitan untuk bangkit tanpa bantuan ,terkadang mengarah ke 'long-lie' di mana lansia tetap terjebak di lantai untuk periode waktu yang lama. 'Long-lie' dapat menyebabkan dehidrasi, *Ulkus dekubitus*, pneumonia, hipotermia dan kematian. Pada Tugas Akhir ini akan dikembangkan sebuah sistem yang dapat mendeteksi jatuh pada manusia lanjut usia menggunakan algoritma *3D Convolutional Neural Network* berbasis sistem tertanam. Adapun training data yang digunakan berasal dari beberapa dataset publik dan akan dibuat dataset pribadi untuk testing dan revisi sistem. Hasil yang diharapkan melalui Tugas Akhir ini adalah terciptanya sebuah sistem pendekripsi jatuh yang dapat melakukan kontak dengan keluarga dan rumah sakit jika diperlukan apabila terdeteksi jatuh sehingga penderita dapat mengalami penanganan medis dengan cepat.

Kata Kunci : Jatuh, Deteksi, Sistem

# KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Pengembangan Simulator Mengemudi Untuk Riset Deteksi Pengemudi Mengantuk (*Driving Simulator Development for Drowsy Driver Detection Research*)**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi *Game Technology*, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak, ibu, adik, dan keluarga saya, atas semangat dan dukungan untuk tetap berkuliah
2. Bapak Dr. Reza Fuad Rachmadi, S.T., M.T.
3. Bapak Arief Kurniawan, ST., MT.
4. Bapak - Ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Teman - teman Asisten Lab B201 Telematika yang selalu membantu dan meneman
6. Teman - teman UKM Kendo ITS yang selalu memberi semangat
7. Serta teman - teman angkatan 2017 yang telah bersama - sama melalui kehidupan perkuliahan bersama penulis

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juni 2021

Penulis

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>Abstrak</b>	viii
<b>KATA PENGANTAR</b>	ix
<b>DAFTAR ISI</b>	xi
<b>DAFTAR GAMBAR</b>	xiii
<b>DAFTAR TABEL</b>	xv
<b>NOMENKLATUR</b>	xvii
<b>1 PENDAHULUAN</b>	1
1.1 Latar belakang . . . . .	1
1.2 Permasalahan . . . . .	2
1.3 Tujuan . . . . .	3
1.4 Batasan masalah . . . . .	3
1.5 Sistematika Penulisan . . . . .	3
1.6 Relevansi . . . . .	4
<b>2 TINJAUAN PUSTAKA</b>	7
2.1 Simulator Berkendara / <i>Driving Simulator</i> . . . . .	7
2.2 <i>Perception Time</i> (PR) . . . . .	7
2.3 Tingkat Kantuk Subjektif dan Objektif ( <i>Subjective and Objective Sleepiness</i> ) . . . . .	7
2.4 <i>Bézier curve</i> . . . . .	8
2.5 <i>6 Degree of Freedom</i> . . . . .	8
2.6 Kecepatan / <i>Velocity</i> . . . . .	9
2.7 <i>Colission</i> . . . . .	10
<b>3 DESAIN DAN IMPLEMENTASI SISTEM</b>	11
3.1 Cakupan Tugas Akhir . . . . .	11
3.1.1 Cakupan <i>Hardware</i> . . . . .	11
3.1.2 Cakupan <i>Software</i> . . . . .	12
3.1.3 <i>Output yang diharapkan</i> . . . . .	12
3.2 Desain Sistem . . . . .	12

3.2.1	Data - Data Internal Simulator . . . . .	12
3.2.2	Data - Data Eksternal Simulator . . . . .	18
3.3	Desain Lajur Simulator . . . . .	22
3.4	Desain <i>Behaviour</i> dari Kendaraan Lain . . . . .	24
3.5	Alur Kerja . . . . .	26
3.6	Pembuatan Simulasi Menggunakan <i>Unity Game Engine</i> . . . . .	26
3.6.1	Pembuatan Sirkuit Jalan . . . . .	26
3.6.2	Pembuatan <i>Terrain</i> . . . . .	31
3.6.3	<i>Menu</i> dan <i>User Interface</i> . . . . .	31
3.6.4	<i>Log File System</i> dan <i>Kalkulasi Data</i> . . . . .	33
3.7	Pengaturan dan Konfigurasi <i>Steering Wheel Controller</i> . . . . .	33
3.8	Pembuatan Modul Pengambilan Data dengan <i>Microcontroller</i> . . . . .	36
3.9	Penggabungan Seluruh Sistem Menjadi Satu Modul	37
<b>4</b>	<b>PENGUJIAN DAN ANALISA</b>	<b>41</b>
4.1	Pengujian <i>User Interface</i> - Pemilihan Jumlah Lajur	41
4.2	Pengujian Pengambilan Data - Kecepatan . . . . .	41
4.3	Pengujian Pengambilan Data - Informasi Spasial . . . . .	42
4.4	Pengujian Pengambilan Data - <i>Response Time</i> dan <i>Input</i> Pengemudi . . . . .	43
4.5	Pengujian Pengambilan Data - <i>Citra Webcam</i> . . . . .	44
4.6	Pengujian Pengambilan Data - <i>Serial Data USB</i> . . . . .	44
4.7	Pengujian Respon Sinyal dari <i>Steering Wheel Controller</i> terhadap simulator . . . . .	44
4.8	Pengujian <i>User Experience / UX</i> dari Pengguna . . . . .	45
<b>5</b>	<b>PENUTUP</b>	<b>59</b>
5.1	Kesimpulan . . . . .	59
5.2	Saran . . . . .	59
<b>DAFTAR PUSTAKA</b>		<b>61</b>
<b>Biografi Penulis</b>		<b>63</b>

## DAFTAR GAMBAR

2.1	Contoh kurva bezier pada bidang 3 dimensi . . . . .	8
2.2	<i>6 Degrees of Freedom - Pitch, Roll, dan Yaw[1]</i> . . . . .	9
2.3	Kecepatan . . . . .	9
3.1	Blok Diagram Cakupan Disiplin Ilmu Tugas Akhir . .	11
3.2	Desain umum modul Simulator . . . . .	13
3.3	<i>Relative Distance</i> Oranye : Jarak dari <i>Center of Mass</i> Mobil ke Batas Pinggir Kanan Jalan Merah : Jarak dari <i>Center of Mass</i> Mobil ke Batas Pinggir Kiri Jalan	14
3.4	Response Time - Ilustrasi Jalur yang diharapkan - Kasus Tidak ada Mobil Lain . . . . .	15
3.5	Response Time - Ilustrasi Jalur yang diharapkan - Kasus terdapat mobil lain . . . . .	16
3.6	2 Macam <i>Colliders</i> Mobil - <i>Box Collider</i> dan <i>Mesh Collider</i> . . . . .	18
3.7	<i>Boundary</i> dalam sirkuit . . . . .	19
3.8	<i>Boundary</i> luar sirkuit . . . . .	20
3.9	Kubus yang memiliki <i>texture information</i> dari <i>Webcam / Imaging Device</i> . . . . .	22
3.10	Desain sirkuit 2 lajur dengan jarak yang pendek dan kompleksitas yang rendah . . . . .	23
3.11	Desain sirkuit 2 lajur dengan jarak yang panjang dan kompleksitas yang tinggi . . . . .	23
3.12	Desain sirkuit 3 lajur . . . . .	24
3.13	Desain sirkuit 4 lajur . . . . .	25
3.14	<i>Shanghai International Circuit[2]</i> . . . . .	26
3.15	5 Macam <i>Bézier curve</i> sirkuit (dari kiri ke kanan) : batas kiri jalan, <i>Vehicle AI Path - Counterclock Wise</i> , tengah lajur / marka jalan, <i>Vehicle AI Path - Clock Wise</i> , serta batas kanan jalan . . . . .	27
3.16	<i>Bézier curve</i> tampak atas, merah : <i>anchor points</i> , biru : <i>control points</i> . . . . .	28
3.17	<i>Bézier curve</i> pada <i>3d plane</i> . . . . .	29
3.18	<i>Normal Force</i> Jalan ( <i>Yaw 0 Derajat</i> ) . . . . .	30
3.19	<i>Normal Force</i> Jalan ( <i>Yaw 60 Derajat</i> ) . . . . .	30

3.20	<i>Terrain</i> setelah ditambahkan bukit, dan pepohonan	32
3.21	<i>User Interface</i> dari Menu . . . . .	32
3.22	Diagram <i>Log File System</i> , proses ekstraksi data, dan proses <i>append data</i> . . . . .	34
3.23	Diagram Steering Wheel yang digunakan, <i>Sumber : Buku Manual PXN Steering Wheel</i> . . . . .	35
3.24	Grafik hasil <i>plotting</i> data <i>input</i> pengemudi . . . . .	36
3.25	Diagram Sinkronisasi <i>capture rate</i> kamera dengan <i>sampling rate</i> dari sensor arduino . . . . .	37
3.26	Diagram <i>Flow Chart</i> Komunikasi Serial Melalui <i>port COM</i> . . . . .	39
4.1	Korelasi <i>user interface</i> pemilihan lajur dengan <i>scene</i> yang dimuat oleh <i>simulator</i> . . . . .	50
4.2	<i>Frame webcam</i> yang tersimpan di dalam <i>harddrive</i> . .	51
4.3	Diagram Sistem Komunikasi Arduino Dengan PC Simulator . . . . .	51
4.4	Contoh data serial yang diterima oleh arduino . . . . .	52
4.5	<i>Input</i> dari pengemudi - <i>Horizontal Axis</i> adalah Data sudut <i>steering wheel</i> , <i>Vertical Axis</i> adalah data nilai tekanan pedal gas / rem . . . . .	52
4.6	Perbandingan sinyal dari <i>steering wheel controller</i> , sebelum dan sesudah grafik unity - <i>Horizontal Axis</i> / Sudut <i>steering wheel</i> . . . . .	53
4.7	Perbandingan sinyal dari <i>steering wheel controller</i> , sebelum dan sesudah grafik unity - <i>Vertical Axis</i> / Nilai tekanan pedal gas/rem . . . . .	53
4.8	Grafik analisa kuantitatif nilai error dari <i>steering wheel controller</i> . . . . .	54

## DAFTAR TABEL

3.1	Tabel Informasi Sirkuit Jalan . . . . .	24
3.2	Tabel Informasi AI Tiap - Tiap Tipe Lajur . . . . .	25
4.1	Skenario kuesioner untuk pengujian kepuasan pengguna . . . . .	46
4.2	Hasil pengujian kepuasan pengguna. . . . .	47
4.3	Tabel <i>Input Pengemudi Selama Kurang Lebih 1 Detik</i>	48
4.4	Data Informasi Kecepatan . . . . .	49
4.5	Data Informasi Spasial . . . . .	55
4.6	Tabel <i>Response Time</i> . . . . .	56
4.7	Data Deteksi <i>Colission</i> . . . . .	57

*Halaman ini sengaja dikosongkan*

## NOMENKLATUR

- fps* : *Frame Per Second / Jumlah Citra Perdetik*  
*unit* : unit pengukuran *Unity Game Engine*

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

Penelitian ini di latar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

### 1.1 Latar belakang

Indonesia mulai memasuki periode aging population, dimana terjadi peningkatan umur harapan hidup yang diikuti dengan peningkatan jumlah lansia, dan Indonesia mengalami peningkatan jumlah penduduk lansia dari 18 juta jiwa (7,56%) pada tahun 2010, menjadi 25,9 juta jiwa (9,7%) pada tahun 2019, dan diperkirakan akan terus meningkat dimana tahun 2035 menjadi 48,2 juta jiwa (15,77%) [3]. Berdasarkan data dari Badan Pusat Statistika (BPS) Indonesia keberadaan lansia yang tinggal sendiri, di mana persentasenya mencapai 9,38 %. Jika dilihat berdasarkan tipe daerah, persentase lansia di perdesaan yang tinggal sendiri lebih tinggi dibandingkan lansia di perkotaan (10,10 % berbanding 8,74 %). Bahkan, terdapat kesenjangan yang cukup tinggi pada lansia yang tinggal sendiri antara lansia perempuan dengan laki-laki (13,39 % berbanding 4,98 %) [4]. Lansia yang tinggal sendiri digambarkan sebagai kelompok yang berisiko dan membutuhkan perhatian khusus [5].

Menurut data World Health Organization (WHO) pada tahun 2018, jatuh adalah penyebab utama kedua kematian akibat cedera yang tidak disengaja atau tidak disengaja di seluruh dunia [6]. Berdasarkan data dari Centers for Disease Control and Prevention tingkat kematian akibat jatuh yang disesuaikan dengan usia adalah 64 kematian per 100.000 orang dewasa yang lebih tua, tingkat kematian akibat jatuh di antara orang dewasa berusia 65 tahun ke atas meningkat sekitar 30% dari 2009 hingga 2018. [7]. Jatuh adalah penanda kelemahan, imobilitas, dan gangguan kesehatan akut dan kronis pada orang tua. Jatuh pada gilirannya mengurangi fungsinya dengan menyebabkan cedera, keterbatasan aktivitas, takut jatuh, dan kehilangan mobilitas. Kebanyakan cedera pada lansia adalah akibat jatuh; patah tulang pinggul, lengan bawah, humerus, dan

panggul biasanya diakibatkan oleh efek gabungan dari jatuh dan osteoporosis [8]. Bahkan ketika cederanya tidak begitu serius, lansia sering kali kesulitan untuk bangkit tanpa bantuan [9], terkadang mengarah ke 'long-lie' di mana lansia tetap terjebak di lantai untuk waktu periode waktu yang lama. 'long-lie' dapat menyebabkan dehidrasi, *Ulkus dekubitus*, pneumonia, hipotermia dan kematian [10].

Teknologi untuk melakukan deteksi jatuh sudah ada, dan umumnya dapat dikategorikan menjadi 2 jenis, yaitu secara visual dan secara *wearable*. Tetapi, lansia memiliki kecenderungan akan membawa hal penting sehingga deteksi secara *wearable* tidak disarankan dikarenakan lansia harus selalu menggunakan alat tersebut yang juga membuat lansia tidak nyaman. Mayoritas lansia juga memiliki kondisi keuangan yang tidak kuat dikarenakan bergantung pada uang pensiun dan tidak memiliki pemasukan.

Oleh karena itu, pada tugas akhir ini akan dikembangkan sebuah sistem untuk melakukan deteksi jatuh dan pelaporan berbasis visual dan juga sistem tertanam dengan harga terjangkau. Metode yang digunakan, yaitu 3D-CNN, berfungsi untuk melakukan deteksi jatuh yang melakukan deteksi secara *frame sequence*, sehingga meningkatkan akurasi dan mengurangi *false alarm*. Diharapkan dengan pengembangan tugas akhir ini, sistem dapat melakukan deteksi jatuh pada lansia sehingga dapat menghubungi keluarga terdekat jika terjadi kejadian jatuh.

## 1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, dapat dirumuskan beberapa rumusan masalah sebagai berikut yaitu meningkatnya angka manusia lanjut usia serta jumlah manusia lanjut usia tinggal sendiri yang relatif banyak, dimana sulit mendapatkan bantuan. Lalu risiko jatuh pada manusia lanjut usia semakin meningkat dengan bertambahnya umur, dikarenakan penurunan fisik. Kemudian manusia lanjut usia mengalami penurunan fisik, sehingga jika mengalami kecelakaan, luka yang diderita dapat menyebabkan kematian jika tidak cepat ditangani.

### **1.3 Tujuan**

Penelitian ini bertujuan untuk merancang sebuah perangkat yang mampu untuk mendeteksi orang yang jatuh secara otomatis. Perangkat tersebut akan menggunakan input video dari *IP Camera* yang diakses oleh sebuah *Single Board Computer* yang didalamnya telah ditanamkan sebuah program pengolahan citra digital. Jika terdeteksi orang jatuh maka gambar akan dikirimkan kepada rumah sakit terdekat dan meminta bantuan. Alarm juga akan menyala untuk mencari orang terdekat dan memberitahukan keluarga baik secara SMS ataupun media sosial.

### **1.4 Batasan masalah**

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah:

1. Pengujian dilakukan dalam ruangan [5]
2. Orang yang akan jatuh hanya 1 orang
3. Kegiatan Uji adalah kegiatan pengambilan data berupa :
  - (a) Korelasi User Interface dengan Lajur yang dimuat.
  - (b) Kecepatan Mobil
  - (c) Informasi Spasial Mobil
  - (d) Respon Waktu Pengendara (*Response Time*)
  - (e) Citra Wajah Pengendara
  - (f) *Serial Data* dari *Microcontroller*
  - (g) Respon Sinyal dari *Steering Wheel Controller* terhadap simulator
  - (h) Kuesioner *User Experience / UX* Pengguna
4. Kegiatan Uji Menggunakan dataset pribadi

### **1.5 Sistematika Penulisan**

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

## 1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

## 2. BAB II Dasar Teori

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu sistem simulator dan pengambilan data variabel - variabel uji.

## 3. BAB III Perancangan Sistem dan Impementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah pengolahan data hingga menghasilkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar penjelasan sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

## 4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisanya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji

## 5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

## 1.6 Relevansi

### 1. Validasi Skala Kantuk Karolinska (*Karolinska Sleepiness Scale*) / KSS dengan Variabel – Variabel EEG

Peneliti pada riset tersebut bertujuan untuk melakukan validasi terhadap skala tingkat kantuk yang di proposisikan oleh Institut Karolinska Swedia, dengan variabel – variabel EEG.

Peneliti memiliki landasan riset data korelasi antara variabel – variabel EEG yang sudah valid dan terbukti. Maka selanjutnya peneliti ingin melakukan analisa korelasi antara variabel – variabel EEG dengan tingkat kantuk yang dimiliki oleh subjek riset. Kesimpulan dari riset tersebut menyatakan, skala tingkat kantuk karolinska memiliki korelasi yang kuat dengan variabel – variabel EEG [7]

2. Tingkat Kantuk Subjektif, Simulasi kinerja Mengemudi menggunakan durasi kedipan mata.

Riset ini memiliki tujuan untuk memproposisikan Tingkat Kantuk secara subjektif, data yang digunakan pada paper riset ini berupa lama durasi tingkat kedipan mata, dari banyak data yang diambil pada riset ini didapatkan korelasi antara durasi kedip mata dengan tingkat kantuk, namun kemudian peneliti menyimpulkan bahwa untuk dapat menentukan pengukuran Tingkat Kantuk secara Subjektif, harus dilakukan replikasi riset ini beberapa kali lagi [8]

3. Persepsi – Waktu Respon terhadap Bahaya tak terduga dijalanan

*Research Question* yang diangkat oleh peneliti pada riset ini adalah “Seberapa lama toleransi waktu yang diperbolehkan untuk pengemudi bereaksi terhadap bahaya tak terduga dijalan?”, atau pada riset ini disebut dengan Perception Time (PR). Peneliti juga menyebutkan bahwa terlalu tinggi menilai angka PR dapat meningkatkan biaya konstruksi jalan. Sedangkan terlalu rendah menilai angka PR dapat menyebabkan meningkatnya bahaya dijalan bagi pengendara kendaraan bermotor di jalan. Setelah melakukan penelitian, peneliti pada riset ini menyimpulkan bahwa waktu response yang ideal bagi 95% populasi yang mengemudi adalah 1,6 detik. Peneliti menambahkan, kesimpulan tersebut hanya berlaku pada situasi yang sama dengan kondisi pengujian. Apabila terdapat bahaya yang lebih mengintimidasi, hal tersebut dapat menghasilkan kemungkinan nilai PR yang berbeda. [9] 11

*Halaman ini sengaja dikosongkan*

## BAB 2

# TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

### 2.1 Simulator Berkendara / *Driving Simulator*

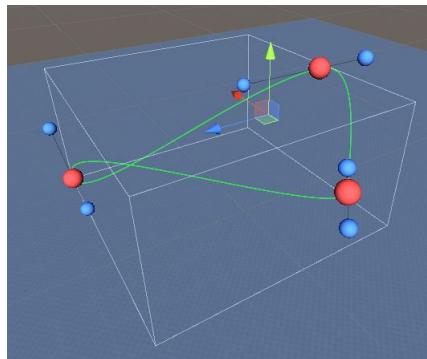
*Driving Simulator* atau *Simulator Berkendara* adalah penggambaran suatu sistem atau proses berkendara dengan peragaan menyerupai proses atau kegiatan berkendara sesungguhnya. *Driving simulator* dapat digunakan untuk kegiatan hiburan, serta juga dapat digunakan untuk kegiatan riset yang mana dapat digunakan untuk memonitor perilaku pengemudi, kinerja, serta perhatian dari pengemudi yang sedang diuji pada simulator tersebut. [11]

### 2.2 *Perception Time (PR)*

*Perception Time (PR)* merupakan istilah yang diciptakan untuk mengukur tingkat kewaspadaan pengemudi. Perhitungan PR, adalah waktu yang dibutuhkan dari terlihatnya halangan tidak terduga hingga waktu yang dibutuhkan oleh pengemudi untuk menekan pedal gas untuk berhenti. PR adalah faktor penting yang telah digunakan oleh para arsitek untuk mendesain serta membangun jalan raya. [9]

### 2.3 Tingkat Kantuk Subjektif dan Objektif (*Subjective and Objective Sleepiness*)

*Subjective sleepiness* atau tingkat kantuk subjektif perlu dibedakan dengan *Objective Sleepiness* atau tingkat kantuk objektif, hal ini dikarenakan untuk mendeteksi pengemudi yang sedang kelelahan diperlukan fitur - fitur objektif oleh suatu sistem sehingga dapat digunakan sebagai dataset untuk sistem deteksi tersebut. Paper ini mengajukan bagaimana cara untuk mendapatkan fitur - fitur *objective sleepiness* tersebut[6]



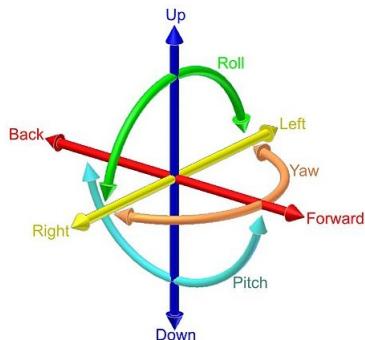
Gambar 2.1: Contoh kurva bezier pada bidang 3 dimensi

## 2.4 *Bézier curve*

*Bézier curve* atau kurva bezier adalah kurva parametrik yang banyak digunakan pada bidang grafika komputer serta bidang - bidang lain yang berhubungan. Sifat dari kurva bezier yang parametrik dapat digunakan untuk memodelkan kurva pada 2 atau bahkan 3 dimensi grafika komputer. Selain itu sifat parametrik ini juga membawa sifat lain yang mana menyebabkan kurva bezier memiliki sifat dapat diskalakan sesuai dengan kebutuhan. Kurva bezier memiliki 2 komponen utama, yaitu *control points* atau titik kontrol, serta *anchor points* atau titik jangkar Dengan menentukan lokasi serta orientasi (apabila pada 3 dimensi) dari titik - titik kontrol dan titik jangkar yang digunakan pada kurva bezier, sehingga dapat di modelkan seluruh jenis kurva dengan mudah, serta pada penerapan di grafika komputer, cukup mudah dalam melakukan operasi transformasi pada titik - titik kontrol untuk mendapatkan jenis kurva yang diinginkan.[10]

## 2.5 6 *Degree of Freedom*

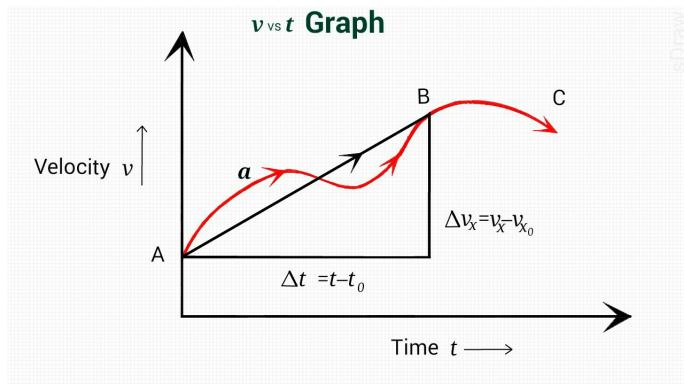
6 *Degree of Freedom* / *DoF*, adalah istilah atau penjelasan dari pergerakan suatu objek pada dimensi spasial ketiga. *DoF* memiliki 3 macam istilah yang perlu di ketahui, Yang pertama *Pitch* ialah merepresentasikan sudut putaran terhadap sumbu *x*, yang kedua *Yaw*, yang merepresentasikan sudut putaran terhadap sumbu *y*, yang ketiga *Roll*, yang merepresentasikan sudut putaran terhadap



**Gambar 2.2:** 6 Degrees of Freedom - Pitch, Roll, dan Yaw[1]

sumbu  $z$ .

## 2.6 Kecepatan / Velocity



**Gambar 2.3:** Kecepatan

Kecepatan / *velocity* pada bidang grafika komputer, terdapat 2 jenis yaitu: kecepatan vektor serta besaran kecepatan, kecepatan vektor ialah seberapa unit jauh objek berpindah dalam suatu hitungan waktu terhadap masing - masing sumbu x, y, serta z.

## 2.7 Collision

Konsep *Collision* pada *Unity Game Engine*, ialah terdapat 2 macam *collider*. *Collider* itu sendiri ialah entitas yang dapat berinteraksi dengan *collider lain*. 2 Macam *collider* adalah *trigger collider* serta *physics collider*. *Physics collider* ialah entitas *collider* yang mensimulasikan hukum fisika dengan akurat, seperti contohnya interaksi roda dengan jalan/tanah, mobil dengan mobil lain, dsb. Sedangkan *Trigger Collider* tidak mensimulasikan hal tersebut, fungsi dari *trigger collider* ialah *unity* dapat mengetahui bahwa suatu entitas *collider* lain telah berinteraksi dengan *trigger collider* tersebut, sehingga dapat melakukan perintah terprogram menggunakan *event system*

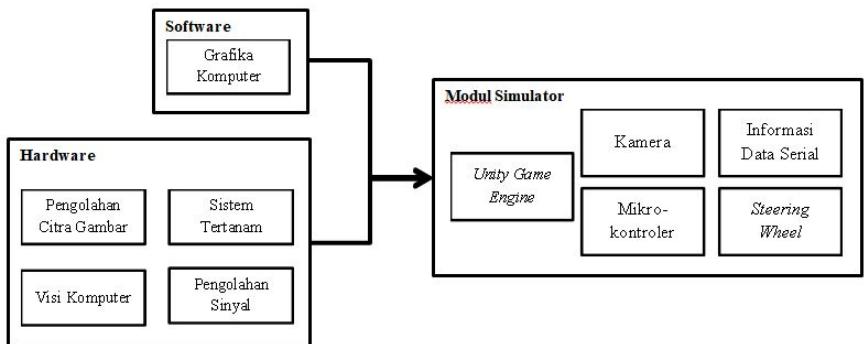
# BAB 3

## DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

### 3.1 Cakupan Tugas Akhir

Tugas akhir ini merupakan salah satu bentuk implementasi grafika komputer untuk mensimulasikan pengalaman berkendara yang digabungkan dengan sistem *Microcontroller* untuk pengambilan data, berikut pada Gambar 3.1 adalah cakupan Tugas Akhir dari Desain Sistem.



**Gambar 3.1:** Blok Diagram Cakupan Disiplin Ilmu Tugas Akhir

#### 3.1.1 Cakupan *Hardware*

Desain sistem secara umum pada gambar 3.1, yang mencakup disiplin ilmu perangkat keras atau *hardware*, ialah pengolahan citra gambar, visi komputer, sistem tertanam, serta pengolahan sinyal.

Disiplin pengolahan citra gambar didapatkan dari pengambilan citra pengemudi menggunakan kamera, visi komputer didapatkan dari proses *recognition* wajah pengemudi, sistem tertanam atau *embedded system* didapatkan dari komunikasi data - data serial menggunakan Arduino atau Mikroprosesor yang tersambung dengan simulator, kemudian yang terakhir Pengolahan sinyal didapatkan dari pengolahan data - data analog seperti data *Electroencephalography (EEG)* / detak jantung pengemudi, atau data *Electrooculography (EOG)* / data kedipan mata dari pengemudi.

### **3.1.2 Cakupan Software**

Cakupan Tugas Akhir ini pada bagian perangkat lunak atau *software*, lebih ditekankan dalam upaya pembuatan *Simulation Environment* atau Lingkungan Simulasi menggunakan *Unity Game Engine*. *Simulation Environment* dibuat dengan menggunakan disiplin ilmu grafika komputer 3d (3d *Computer Graphics*), serta juga menggunakan proses - proses pengembangan dari *game engine* dan *physics engine* yang lain.

### **3.1.3 Output yang diharapkan**

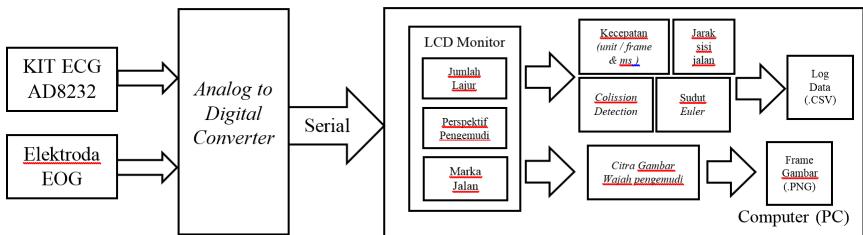
*Output* atau keluaran yang diharapkan dari tugas akhir ini ialah, dihasilkannya suatu modul simulasi yang terintegrasi lengkap dengan *tools - tools* dan *peripheral* yang dapat mensimulasikan suatu pengalaman mengemudi menggunakan suatu *simulator*, serta dapat melakukan proses pengambilan data - data primer yang valid, sehingga dapat diolah untuk proses riset selanjutnya.

## **3.2 Desain Sistem**

Pada Tugas Akhir ini, dilakukan penggabungan perangkat lunak berupa *game engine* untuk mensimulasikan proses berkendara, dengan perangkat keras berupa *controller* dari *simulator* dan *Micro-controller* untuk proses pengambilan data. Proses kerja dari sistem ini akan dijelaskan melalui diagram alur pada gambar 3.2. Selain itu, simulator ini memiliki 2 jenis data yang dapat diukur. Berikut penjelasan 2 macam jenis data tersebut.

### **3.2.1 Data - Data Internal Simulator**

Jenis data yang pertama adalah data yang berasal dari dalam modul simulator, yaitu data - data seperti kecepatan mobil, informasi spasial seperti, sudut *euler* (*pitch,yaw,roll*) dan jarak relatif



**Gambar 3.2:** Desain umum modul Simulator

terhadap pinggir jalan, informasi tabrakan / *colission*, serta informasi waktu respon / *response time*. Data - data ini, disebut sebagai data internal, dikarenakan data - data tersebut bisa di ekstraksi langsung dari *game engine*.

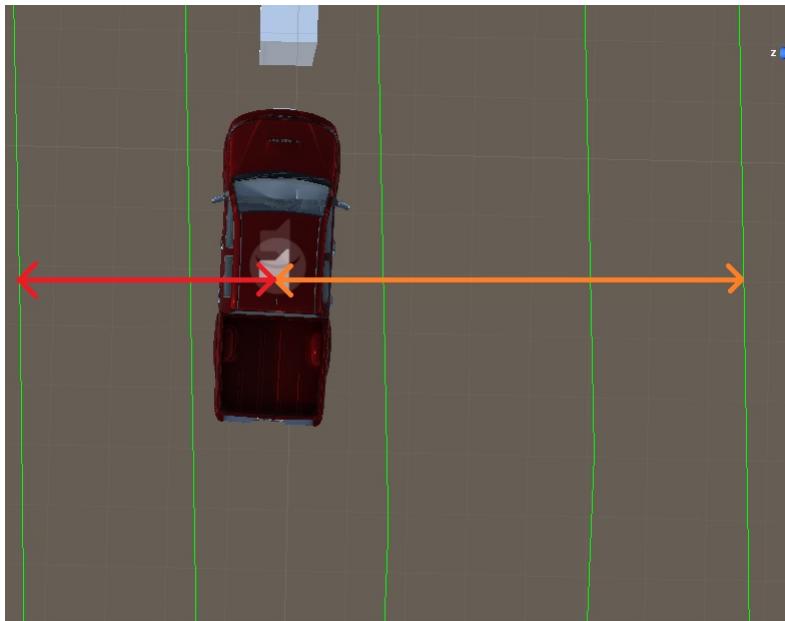
### Kecepatan Mobil / *Velocity*

Pada unity game engine, menggunakan library Unity. Bisa didapatkan secara langsung variabel - variabel yang berhubungan dengan kecepatan. Pada tugas akhir ini, dibutuhkan data kecepatan mobil relatif terhadap dunia atau biasa disebut *global velocity*. Namun selanjutnya, data kecepatan global pun bisa dibagi menjadi 4 macam, yaitu : *velocity* atau kecepatan kearah sumbu *x*, *velocity* atau kecepatan kearah sumbu *y*, serta *velocity* atau kecepatan kearah sumbu *z*, ketiga hal tersebut bisa juga disebut kecepatan vektor *x,y,z*. Dan yang terakhir, adalah *velocity magnitude*, atau tingkat kebesaran suatu kecepatan berupa skalar. Contoh data ini bisa dilihat pada tabel pengujian 4.4

### Informasi Spasial

Selain kecepatan, bisa didapatkan pula informasi spasial yang ada pada simulator. Pada tugas akhir ini, informasi spasial simulator adalah sudut euler yang merepresentasikan 6 derajat kebebasan atau biasa disebut *6 Degree of Freedom (DoF)*), yaitu *pitch*, *yaw*, dan *roll* (gambar 2.2). Pada Tugas akhir ini, ketiga macam rotasi atau derajat kebebasan seluruhnya dimanfaatkan pada proses pembuatan desain jalan.

Selanjutnya data spasial yang bisa didapatkan ialah posisi relatif kendaraan terhadap pinggir jalan (gambar 3.3). Dengan mengu-



**Gambar 3.3:** *Relative Distance*

Oranye : Jarak dari *Center of Mass* Mobil ke Batas Pinggir Kanan Jalan

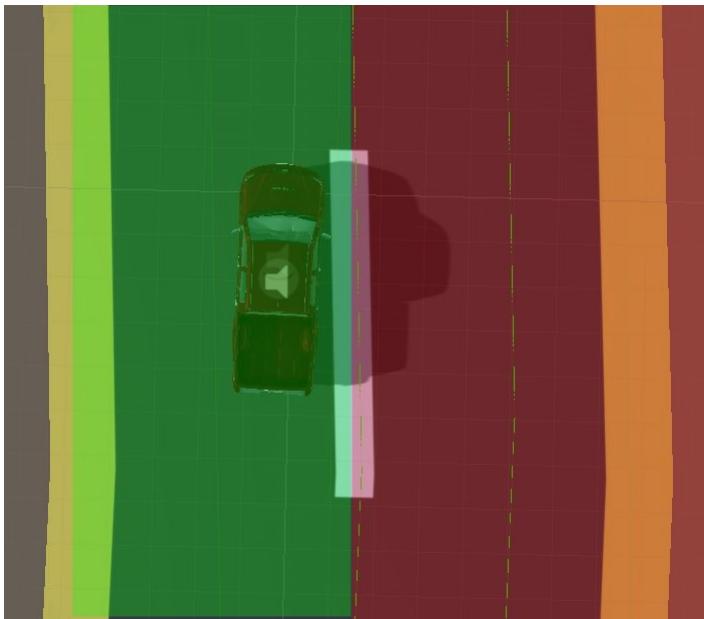
Merah : Jarak dari *Center of Mass* Mobil ke Batas Pinggir Kiri Jalan

kur jarak terdekat dari titik pusat massa atau *Center of Mass* mobil, ke batas pinggir kanan dan batas pinggir kiri jalan, dapat diketahui dimana posisi mobil di suatu lajur tersebut.

Dengan menggabungkan dua macam jenis data tersebut, bisa didapatkan informasi yang sangat jelas tentang posisi dan orientasi dari kendaraan pengujian simulator ini. Yang nanti kedepannya sangat dibutuhkan untuk proses riset selanjutnya, yang akan memanfaatkan data - data tersebut.

## Response Time

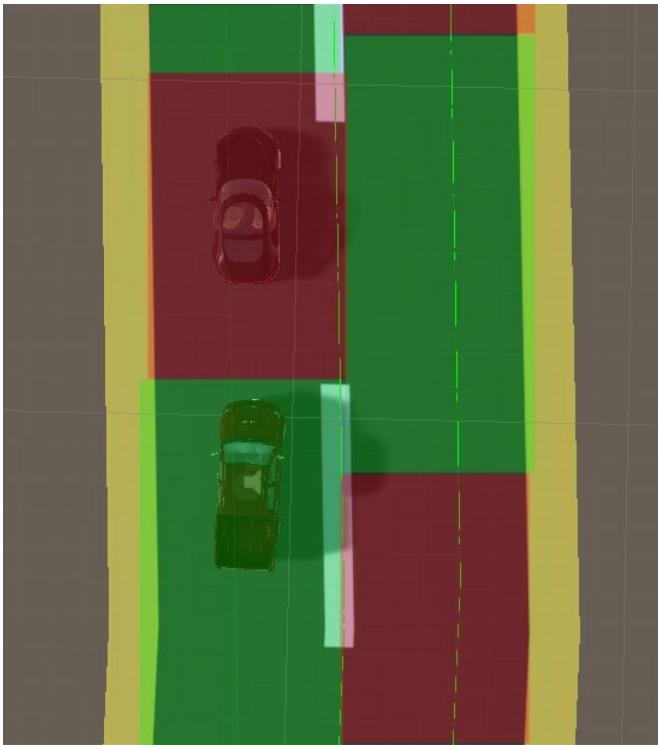
*Response time* adalah metode pengukuran tingkat kewaspadaan pengemudi, dengan cara mendekripsi ketika pengemudi keluar dari lajur yang diharapkan. Lajur yang diharapkan disini, yang di-



**Gambar 3.4:** Response Time - IlustrasiJalur yang diharapkan - Kasus Tidak ada Mobil Lain

maksud ialah lajur sebelah kiri, dikarenakan desain simulator menggunakan model jalan yang ada di Indonesia. Berikut penjelasan tentang data *Response time* pada gambar 3.4 dan 3.5.

Pada gambar 3.4, adalah gambar ilustrasi jalur yang diharapkan, pada kasus dimana tidak ada mobil lain didekat mobil pengujian. Warna hijau menandakan area yang di tandai oleh sistem simulator sebagai jalur yang diharapkan oleh sistem, artinya sistem mendekksi bahwa mobil pada jalur yang telah sesuai serta pengemudi masih dalam keadaan waspada serta memiliki kontrol terhadap mobil. Apabila sistem mendekksi mobil memasuki area yang berwarna merah, maka sistem akan melakukan pencatatan data yaitu kapan mobil mulai keluar dari jalur (tanggal dan waktu). Selanjutnya, sistem juga akan melacak mobil apabila mobil telah kembali ke area hijau, yang mana pada saat ini, sistem akan melakukan pencatatan



**Gambar 3.5:** Response Time - Ilustrasi Jalur yang diharapkan - Kasus terdapat mobil lain

data seberapa lama mobil telah keluar dari area hijau dengan *unit* sekon, yang selanjutnya bisa didapatkan pula, seberapa lama mobil telah keluar dari area hijau dalam *unit* hitungan *frame*, dengan cara mengalikan waktu dalam sekon, dengan *average framerate* dari simulator saat itu.

Selanjutnya, pada gambar 3.5, adalah ilustrasi kasus dimana terdapat mobil didekat mobil uji, yang menyebabkan berubahnya jalur yang diharapkan. Apabila sistem mendeteksi terdapat mobil lain didekat mobil uji, sistem akan melakukan perubahan jalur yang diharapkan oleh sistem, metode penerapan pendektsian seperti ini

dapat diterapkan dengan banyak cara, salah satunya bisa dilakukan dengan membuat *trigger box collision* disekitar mobil uji, yang mana ukurannya lebih besar dari *bounding box* / ukuran mobil uji. Apabila terdapat suatu objek (seperti mobil lain), yang memasuki *trigger box collision* dari mobil uji, maka bisa dilakukan perubahan pendektsian jalur. Pada Tugas Akhir ini, digunakan metode ini untuk mendeteksi adanya mobil lain disekitar mobil uji, serta mendeteksi perlu terjadinya perubahan jalur yang diharapkan.

Cara lain yang lebih mudah, namun tentunya dengan mengorbankan tingkat keakurasi pendektsian yaitu adalah, dengan mengukur jarak mobil uji dengan mobil lain. Apabila jarak mobil uji dengan mobil lain ini dibawah nilai threshold, maka bisa dilakukan perubahan jalur yang diharapkan.

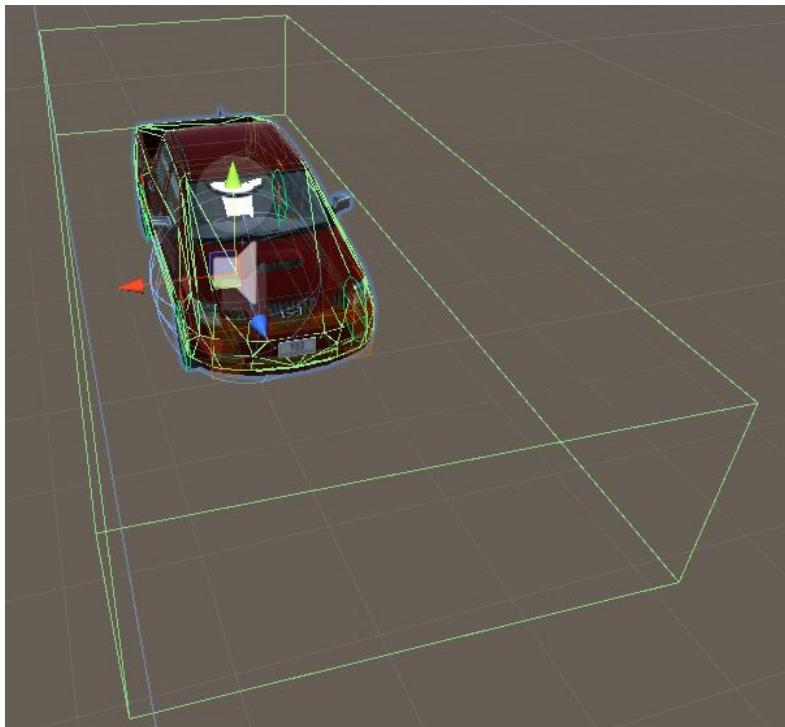
Selanjutnya perlu digaris bawahi, bahwa sistem deteksi perubahan jalur ini juga sangat berkaitan dengan sistem deteksi terjadinya tabrakan. Sistem pendektsian dan pencatatan data atau *logging* dari *collision event* ini akan dijelaskan pada subbab selanjutnya.

### ***Colission Event***

Pada subbab sebelumnya telah dijelaskan metode untuk mendeteksi keberadaan mobil lain di dekat mobil uji pada tugas akhir ini, yaitu dengan cara menggunakan *trigger box event*. Pada subbab ini, *collision event* yang dimaksud terdapat 2 macam. Yaitu yang pertama adalah, kejadian tabrakan dengan mobil lain, yang kedua kejadian tabrakan dengan pinggir jalan / *boundary*.

Untuk proses pendektsian tabrakan dengan mobil lain, dapat menggunakan *mesh collider* yang sudah tersedia dengan mobil, untuk mempermudah proses deteksi, serta menyederhanakan *physics interaction* antar mobil. Selain mobil uji, seluruh *mesh collider* dari mobil lain ditentukan sebagai *trigger*. Hal ini untuk menghindari terjadinya interaksi - interaksi yang tidak diinginkan ketika terjadi tabrakan antar mobil.

Dengan menggunakan fungsi `void onCollisionEnter()` dan fungsi `void onTriggerEnter()` pada unity, dapat dilakukan pengecekan apabila terjadi overlap antar 2 *GameObject* yang ada pada unity. Dengan menentukan *GameObject* mana yang perlu dideteksi, sehingga langkah selanjutnya ialah mencatat data colis-



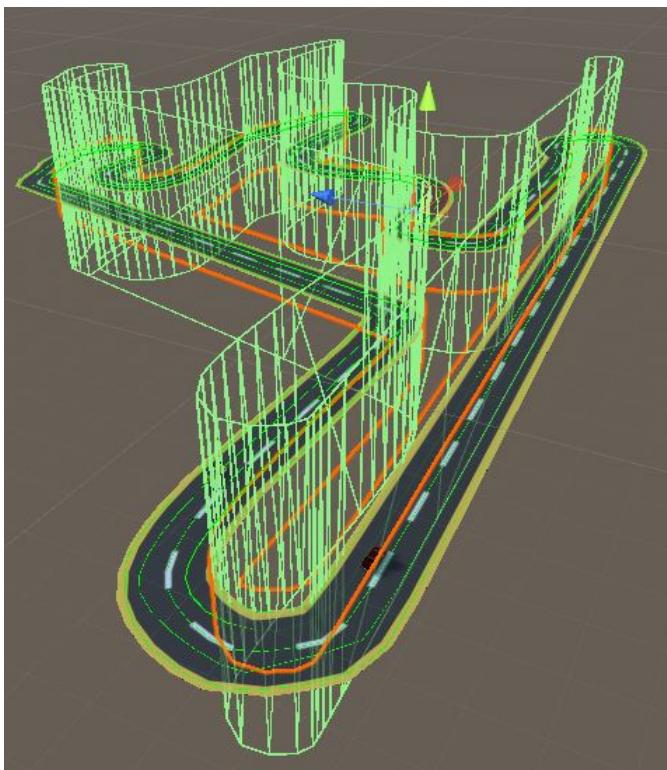
**Gambar 3.6:** 2 Macam *Colliders* Mobil - *Box Collider* dan *Mesh Collider*

sion kedalam *log file system*, 3 *GameObject* yang perlu dideteksi adalah : *Boundary Luar* / Batas Pinggir Kanan Jalan, *Boundary Dalam* / Batas Pinggir Kiri Jalan, serta Mobil lain.

Contoh data yang diambil pada *colission event* bisa dilihat pada tabel 4.7.

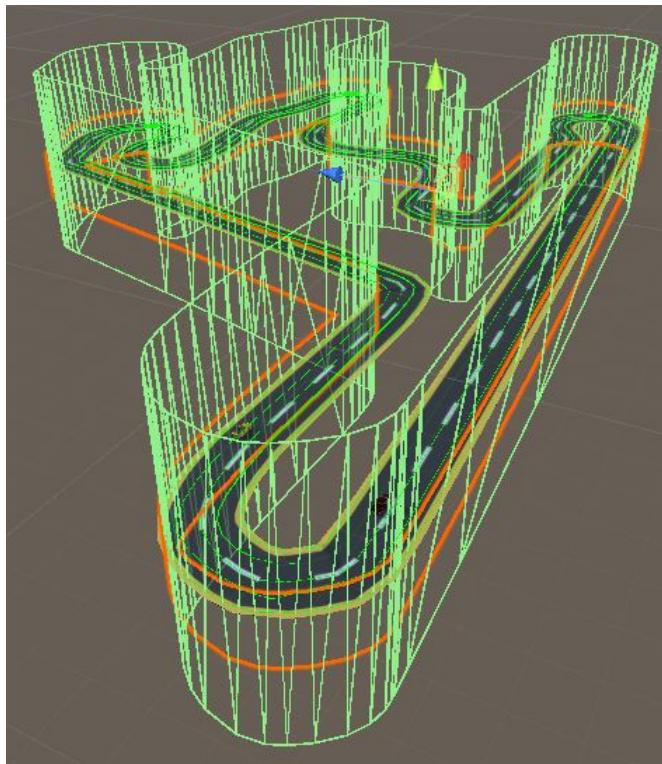
### 3.2.2 Data - Data Eksternal Simulator

Selanjutnya Jenis data yang kedua adalah data yang berasal dari luar modul simulator, yaitu data - data seperti sinyal *Electroencephalography (EEG)* / detak jantung pengemudi, dan atau data sinyal *Electrooculography (EOG)* / data kedipan mata dari pengemu-



**Gambar 3.7:** *Boundary* dalam sirkuit

di, serta data citra wajah pengemudi menggunakan kamera. Data - data tersebut diatas, dikatakan sebagai data eksternal dikarenakan data - data tersebut didapatkan dari peralatan *peripheral* yang dipasang pada modul simulator. Berhubung data EEG dan EOG merupakan data sinyal analog, maka diperlukannya suatu *Analog to Digital Converter* atau ADC, supaya data bisa di rekam dalam *log file*. Tentunya, perlu diperhatikan juga sampling rate dari ADC ini, sehingga bisa relevan dan dapat disesuaikan dengan data - data yang lain.



**Gambar 3.8:** *Boundary* luar sirkuit

### ***Serial Communication* Melalui port COM**

Pada tugas akhir ini, *serial communication* melalui *port COM*, dilakukan menggunakan *Python*. Diagram *flow chart* dari proses komunikasi data serial arduino ke PC, bisa di lihat pada gambar 3.26.

Proses dari modul pembacaan data serial dimulai dengan pembuatan *file* dengan nama tanggal dijalankannya modul tersebut. Hal ini dilakukan untuk mengetahui kapan waktu dan tanggal modul pembacaan dijalankan. Selain itu, hal ini memastikan bahwasanya file - file dapat dipisahkan atau disortir berdasarkan waktu dan tanggal, yang artinya data - data yang dicatat, tidak akan bercampur

aduk dengan data - data percobaan yang sebelumnya.

Dengan mengasumsikan bahwa arduino akan melakukan pengiriman data, *python* akan terus membaca data dari arduino, dengan maksimum *baudrate* yang telah ditentukan. Kemudian, setelah proses pembacaan dilakukan, akan dilakukan proses *decode*. Proses *Decode* ialah proses dimana *python* akan melakukan konversi data biner, menjadi data - data diskrit *integer* / bilangan bulat, yang kemudian hasil dari proses *decode*, akan dilakukan proses append pada *file* yang telah dibuat sebelumnya.

Informasi lebih jelas tentang proses *append file* dan *log file system* akan dijelaskan lebih lanjut pada sub-bab 3.6.4

### **Citra Wajah Pengemudi / Webcam**

Pada *Unity Game Engine*, sistem mensupport pembacaan informasi kamera webcam sebagai salah satu bagian dari *library mobile camera information*, yang artinya adalah, apabila platform dari simulator menggunakan PC dengan OS *Windows*, dan terdapat *peripheral Webcam* yang terpasang, Unity dapat mendeteksi *webcam* tersebut sebagai *Imaging Device*.

Sebagai *Imaging Device*, data yang diperoleh unity dari *webcam* ditangkap sebagai *texture information*, atau informasi texture dari suatu *GameObject*.

Dikarenakan informasi yang ditangkap oleh unity merupakan suatu *texture*, maka *texture* tersebut bisa di pasangkan ke suatu *GameObject*, agar memiliki tampilan dari *webcam*. Bisa dilihat, pada gambar 3.9, ketika informasi *webcam* di pasangkan dengan kubus.

Namun pada tugas akhir ini, yang dibutuhkan bukanlah informasi *texture* yang bisa digunakan oleh game ini. Yang dibutuhkan adalah *frame - frame* / gambar wajah dengan *timestamp* yang sesuai. Maka langkah selanjutnya adalah mengekstrak informasi *texture* tersebut keluar dari unity. Hal ini mudah dilakukan dengan melakukan fungsi *encoding* ke tipe *file* yang berekstensi *PNG*. Setelah encoding dilakukan, *file stream* akan menuliskan data hasil *encoding* menjadi suatu gambar lengkap dengan *timestamp* nya.

Hasil pengujian pengambilan data citra wajah pengemudi menjadi frame - frame yang telah di *encode* menjadi *PNG* bisa di lihat pada gambar 4.2



**Gambar 3.9:** Kubus yang memiliki *texture information* dari *Webcam / Imaging Device*

### 3.3 Desain Lajur Simulator

Pada tugas akhir ini, terdapat 4 macam lajur yang dapat dipilih oleh pengemudi. 4 Macam lajur tersebut berfungsi sebagai basis riset untuk mengeliminasi terjadinya bias atau pengaruh yang dihasilkan oleh perbedaan jumlah lajur yang digunakan.

Berikut pada Gambar 3.10, 3.11, 3.12, dan 3.13, adalah tampak atas dari desain - desain lajur yang digunakan pada simulator ini, seluruh lajur didesain dengan bentuk *closed-loop circuit*. Hal ini bertujuan agar mengurangi ukuran aset jalan yang akan digu-



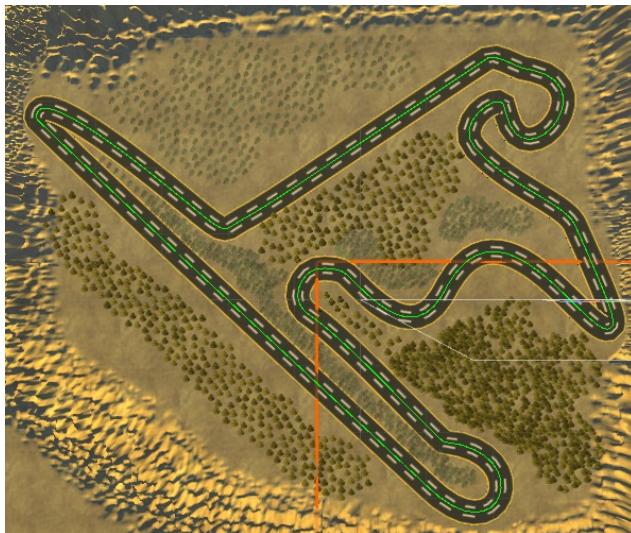
**Gambar 3.10:** Desain sirkuit 2 lajur dengan jarak yang pendek dan kompleksitas yang rendah



**Gambar 3.11:** Desain sirkuit 2 lajur dengan jarak yang panjang dan kompleksitas yang tinggi

nakan. Selain itu desain seperti ini dapat berdampak pada durasi pengambilan data, desain sirkuit yang *closed-loop* memungkinkan kegiatan proses pengambilan data supaya tidak bergantung pada panjang jalan yang digunakan.

Tabel 3.1 adalah detil informasi dari desain jalan / sirkuit yang dibuat pada simulator ini.



**Gambar 3.12:** Desain sirkuit 3 lajur

**Tabel 3.1:** Tabel Informasi Sirkuit Jalan

No.	Jumlah Lajur (Unit)	Jarak (Unit)	Lebar (Unit)	Kompleksitas (Tampak Atas)	
				CW Curve	CCW Curve
1	2	43,145	12	4	0
2	2	260.226	12	6	10
3	3	260.226	16	6	10
4	4	411.346	20	11	12

### 3.4 Desain *Behaviour* dari Kendaraan Lain

Pada Tugas akhir ini, Desain dari *AI Behavior* kendaraan lain cukup mengikuti lajur sirkuit sesuai dengan jalur yang telah didefinisikan oleh *Bézier curve*, kemudian pada script *follower* yang ada di unity, dikarenakan desain lajur yang *closed-loop*, perlu di definisikan arah dari kendaraan tersebut, dilihat dari tampak atas, apakah kendaraan memiliki *clockwise path* atau *counter-clockwise path*.

Selain arah dari kendaraan, diperlukannya pendefinisian kece-



**Gambar 3.13:** Desain sirkuit 4 lajur

**Tabel 3.2:** Tabel Informasi AI Tiap - Tiap Tipe Lajur

No.	Lajur	Jumlah Mobil	AI Rotation		Velocity (Unit/Frame)	
			CW	CCW	Min	Max
1	2 ( <i>Short</i> )	1	X	✓	5.0f	15.0f
2	2 ( <i>Long</i> )	2	✓	✓	0.1f	0.9f
3	3	3	✓	✓	0.1f	0.9f
4	4	4	✓	✓	0.01f	0.1f

patan kendaraan tersebut, pada Tugas Akhir ini kecepatan kendaraan ditentukan dengan randomisasi dalam suatu rentang tiap kali simulator dijalankan.

Informasi detil dari desain AI kendaraan lain di *unity*, dapat dilihat pada tabel 3.2

### 3.5 Alur Kerja

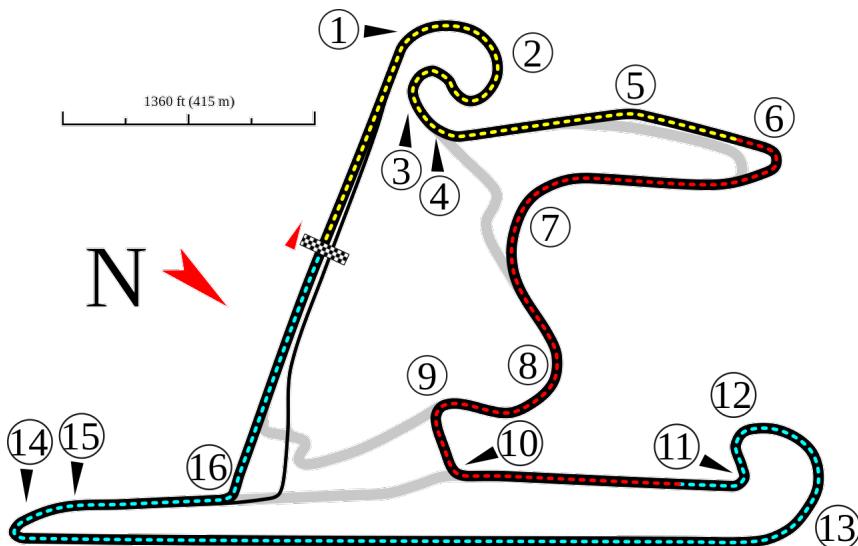
Pembuatan tugas akhir ini dibagi menjadi beberapa tahapan, yaitu:

1. Pembuatan Simulasi Menggunakan *Unity Game Engine*
2. Pengaturan dan Konfigurasi *Steering Wheel Controller*
3. Pembuatan Modul Pengambilan Data dengan *Microcontroller*
4. Penggabungan Seluruh Sistem Menjadi Satu Modul

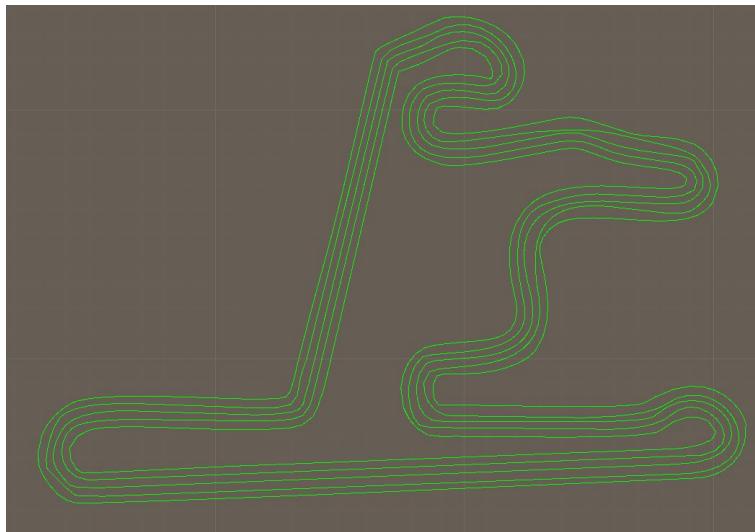
### 3.6 Pembuatan Simulasi Menggunakan *Unity Game Engine*

Pada Tugas Akhir ini, proses pembuatan simulasi menggunakan *Unity Game Engine*. Namun tidak hanya menggunakan *Unity Game Engine* saja, tentunya selain memanfaatkan *unity editor*, juga diperlukan *source code editor*, pada tugas akhir ini menggunakan *Visual Studio 2017*. Selain itu terdapat *tools - tools* lain di luar *unity* yang dapat mempermudah proses pembuatan Simulator ini.

#### 3.6.1 Pembuatan Sirkuit Jalan



Gambar 3.14: *Shanghai International Circuit*[2]



**Gambar 3.15:** 5 Macam *Bézier curve* sirkuit (dari kiri ke kanan) : batas kiri jalan, *Vehicle AI Path - Counterclock Wise*, tengah lajur / marka jalan, *Vehicle AI Path - Clock Wise*, serta batas kanan jalan

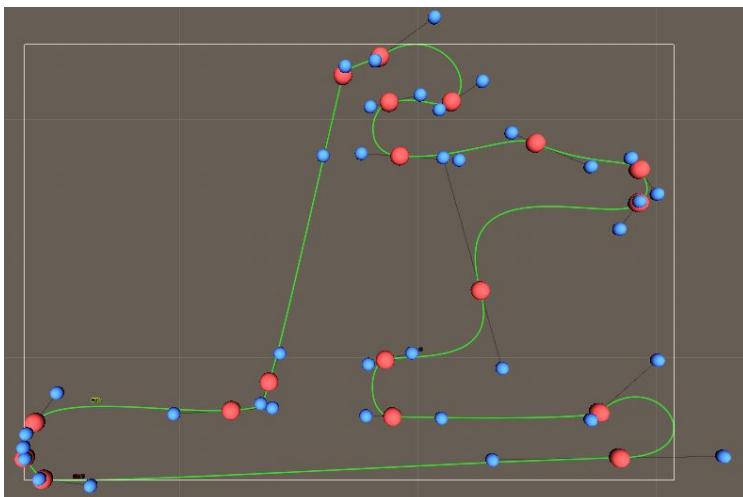
Proses pembuatan sirkuit jalan adalah proses yang paling memakan waktu pada tugas akhir ini. Yang pertama perlu dilakukan adalah mendefinisikan *path* jalan menggunakan *Bézier curve*, atau jalur yang akan digunakan sebagai *base* dari *3d mesh* jalan atau lajur itu sendiri, salah satu contohnya adalah pada gambar 3.15, bisa dilihat terdapat 5 macam kurva bezier yang dapat ditemukan. Semua 5 kurva tersebut dibutuhkan untuk proses kalkulasi informasi jarak serta informasi *colission* yang nantinya akan disimpan kedalam *log file*.

Menggunakan *Shanghai International Circuit* dari *Formula 1*, sebagai referensi bentuk dari sirkuit, diperlukannya penyesuaian tingkat kelengkungan dari belokan yang ada pada sirkuit referensi. Untuk menghindari *mesh overlap*, maka tiap - tiap tikungan yang tajam, perlu diperhalus sedemikian hingga *mesh overlap* agar tidak terjadi. Namun tidak terlalu dikurangi sehingga pengemudi tetap waspada terhadap tikungan tersebut, serta supaya pengemudi

mengurangi kecepatan sebelum melakukan belokan tersebut.

Hal ini penting, dikarenakan pada saat proses pengujian berjalan, mobil tidak diharapkan untuk keluar dari lajur pengujian.

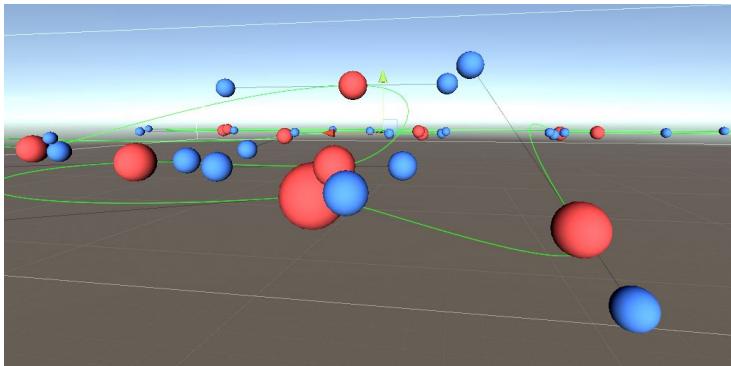
### ***Editing Tools : Path Creator***



**Gambar 3.16:** *Bézier curve* tampak atas, merah : *anchor points*, biru : *control points*

Untuk memudahkan proses pembuatan *Bézier curve*, pada unity. Penulis menggunakan *tools* yang tersedia di unity yang berfungsi untuk mempermudah kalkulasi matematis serta proses *scripting* yang ada pada unity untuk menghasilkan kurva tersebut. Tools tersebut ialah *Path Creator* oleh Sebastian Lague, Path Creator ini dipilih dikarenakan, memudahkan dalam proses manipulasi kurva *Bézier*. Dikarenakan pada *unity* tidak mensupport kurva *Bézier* secara natif, maka biasanya para *game developer*, membuat *script* mereka masing - masing untuk membuat kurva *Bézier*, dengan melakukan kalkulasi persamaan matematis pada script mereka tersebut, kemudian diperlukannya manipulasi titik - titik kontrol sesuai dengan *script* tersebut.

Namun *Path Creator* memungkinkan penulis untuk melewati proses kalkulasi matematis serta pendefinisian titik - titik kontrol,



Gambar 3.17: *Bézier curve* pada *3d plane*

langsung ke proses manipulasi *drag and drop* titik - titik kontrol, pada *Unity Editor*.

Pada gambar 3.16 dan 3.17, bisa dilihat titik - titik yang dapat dimanipulasi menggunakan *Path Creator*. Warna merah menunjukkan *anchor points*, yang artinya titik tersebut merupakan titik - titik statis pada suatu persamaan kurva, dengan memanipulasi lokasi dari *anchor points* pada 3 dimensi, dapat didapatkan bentuk *approximation* dari target sirkuit diinginkan. Tentunya, semakin banyak *anchor points* yang digunakan, semakin akurat pula *approximation* jalan yang dibuat. Namun tentunya, kurva jalan menjadi lebih kompleks serta lebih sulit untuk dimanipulasi.

Kemudian, warna biru menunjukkan *control points*, *control points* ini menentukan lengkungan yang diharapkan diantara tiap - tiap *anchor points*, tentunya agar didapatkannya lengkungan yang sesuai dengan belokan yang ada pada sirkuit referensi, titik - titik inilah yang akan dapat dimanipulasi.

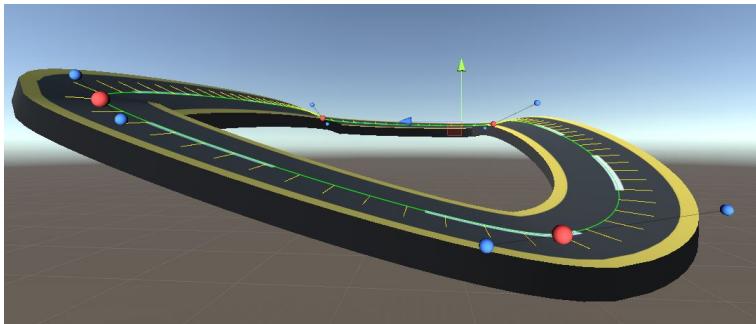
### ***Plane Projection* serta *Normal Vector***

Pada Tugas Akhir ini, informasi tentang *plane projection* yang ada pada kurva *Bézier* sangat penting. Hal ini dapat menunjukkan informasi tentang kekasaran medan atau *level* ketinggian dari suatu daerah pada *terrain* yang ada pada jalan.

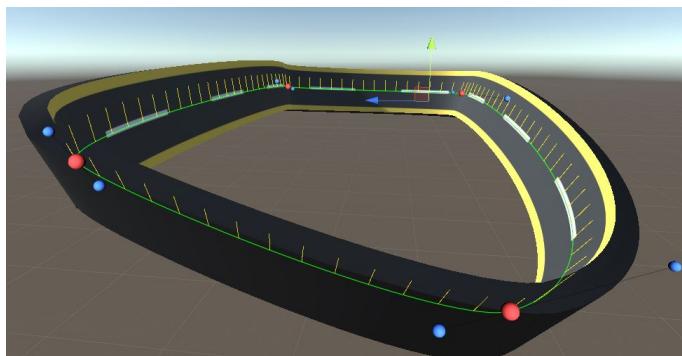
Pada Tugas akhir ini, jalan yang memiliki 2 dan 3 lajur, memiliki *plane projection* terhadap *XZ Plane*, artinya medan yang ada

pada kedua macam jumlah lajur tersebut memiliki kedataran yang seragam, tidak memiliki tanjakan ataupun turunan.

Sedangkan, pada jalan yang memiliki 4 lajur, dengan memanfaatkan semua sumbu *XYZ* maka dapat dihasilkan jalur yang memiliki tanjakan dan turunan.



Gambar 3.18: *Normal Force Jalan (Yaw 0 Derajat)*



Gambar 3.19: *Normal Force Jalan (Yaw 60 Derajat)*

Kemudian selanjutnya adalah, penentuan *normal force*. Pada *path creator*, parameter yang dapat dikontrol adalah *global angle*, dari normal force tersebut. Hal ini menentukan sudut rotasi / *yaw*, dari jalan yang dibuat pada path yang ditentukan.

Berikut pada gambar 3.18 dan gambar 3.19, perbandingan efek

dari *normal force* yang ada di jalan. Hal ini penting diperhatikan untuk proses desain serta pembuatan jalan, yang memanfaatkan seluruh 6 derajat kebebasan (*Degree of Freedom*), *pitch*, *yaw*, dan *roll*

### 3.6.2 Pembuatan *Terrain*

Medan atau *terrain*, merupakan kanvas kosong dari sebuah *scene*, yang dapat diisi dengan berbagai macam objek - objek estetis untuk meningkatkan realisme dan imersivitas pada proses simulasi. Pada tugas akhir ini medan yang dibuat pada tiap lajur tidak terlalu berpengaruh terhadap proses pengambilan data itu tersendiri, namun merupakan salah satu *design choice* atau proses pemilihan desain agar meningkatkan kualitas dari *user experience*.

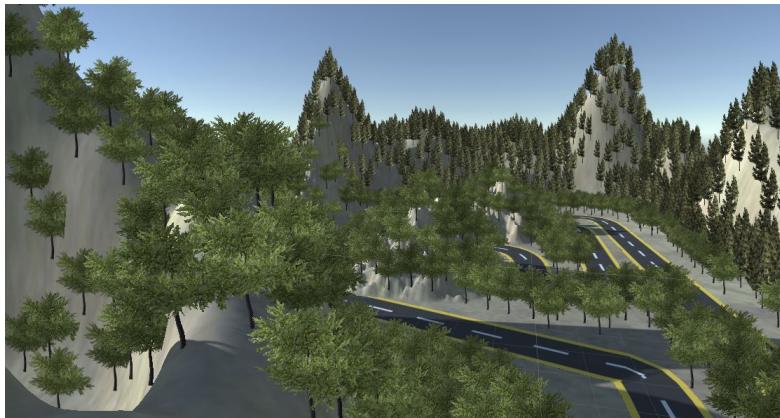
Setelah proses pembuatan jalan selesai, untuk meningkatkan estetika, diperlukannya *terrain* yang mengelilingi jalan. Pada *Unity*, objek - objek *non-collision* dapat ditaruh untuk meningkatkan estetika tersebut. Seperti contohnya, pohon - pohon dan semak - semak atau rerumputan. Selain itu, dapat diterapkan perubahan kontur dari *terrain*. Seperti contohnya dapat diterapkan proses peninggian sebagian dari *terrain* untuk menghasilkan bukit, atau bahkan sebaliknya, proses penurunan dari sebagian *terrain* dapat menghasilkan lembah.

Selain itu, pada *unity* dapat ditambahkannya tekstur pada *terrain*. Tekstur disini yang dimaksud ialah, warna atau gambar yang dapat diterapkan pada *terrain* agar menghasilkan corak pada *terrain*. Seperti contohnya tekstur dari tanah coklat, atau tekstur warna pasir kuning, dan sebagainya.

Selanjutnya, perlu di garis bawahi juga, pada Tugas Akhir ini, salah satu *design choice* merupakan jalan berbentuk sirkuit, maka dari itu *terrain*, untuk mengakomodasi hal tersebut di terapkan bukit yang mengelilingi sirkuit jalan, untuk memberikan kesan bahwasanya mobil tidak dapat keluar dari lajur.

### 3.6.3 *Menu* dan *User Interface*

Proses pembuatan *Menu* dibutuhkan untuk memudahkan pemilihan jumlah lajur nanti setelah simulator menjalani proses *build*. *Menu* disini dibutuhkan agar tidak terlalu kompleks, namun cukup agar menu dapat menjelaskan serta merepresentasikan tampilan atau tombol yang ditekan. Pemilihan yang intuitif serta fungsional-



Gambar 3.20: Terrain setelah ditambahkan bukit, dan pepohonan

litas yang berjalan sudah cukup untuk proses ketika di menu.

Perlu diperhatikan juga, dapat diterapkan pula *asynchronous loading*, pada menu ini. Artinya, *scene* atau jenis lajur yang dipilih, akan di muat sebelum tombol ditekan. Supaya proses *loading* akan menjadi lebih cepat.



Gambar 3.21: User Interface dari Menu

Gambar ?? adalah, gambar yang menunjukkan hasil menu yang dibuat. Terdapat 4 tombol yang bisa ditekan, tiap - tiap tombol memiliki penjelasan *text* yang jelas, yaitu merepresentasikan tombol yang akan membawa *user* menuju *scene* atau lajur yang sesuai dengan text tersebut, seperti 2 lajur pendek, 2 lajur panjang, 3 lajur, serta 4 lajur. Selain itu, dipasang pula *background canvas* serta logo ITS pada *User Interface Menu*.

### **3.6.4 Log File System dan Kalkulasi Data**

*Log File System* adalah salah satu komponen penting dari Tugas Akhir ini, agar didapatkan suatu *Log File System*, beberapa macam komponen dibutuhkan dari sistem ini (gambar 3.21).

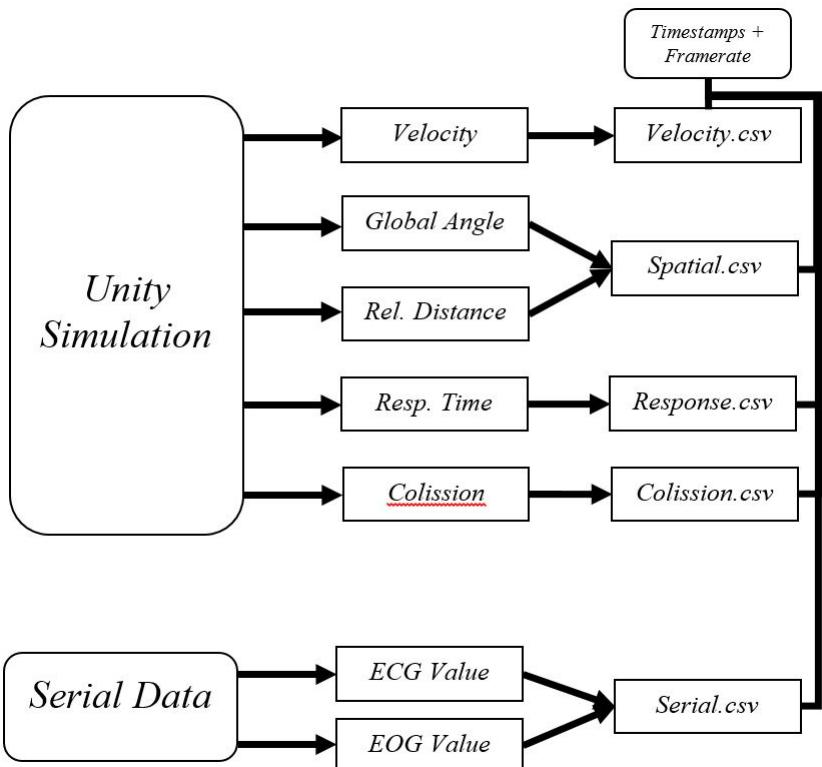
Yang pertama adalah, proses ekstraksi data. Proses ini sangat mudah dikarenakan informasi - informasi yang dibutuhkan sudah tersedia dan tertulis pada variabel - variabel yang dipakai pada simulasi serta komunikasi *serial data*.

Setelah proses ekstraksi data yang dibutuhkan, selanjutnya dibuatlah komunikasi *file stream* dari *unity* ke *OS file system*, pada Tugas Akhir ini, OS yang digunakan adalah Windows 8.1. Komunikasi file stream yang dibuat bertugas untuk mengecek apakah *path* dari direktori yang dituju sudah ada, apabila belum tersedia folder yang diharapkan, maka *file stream*, akan membuat folder pada direktori yang diinginkan dan membuat *file - file* sesuai dengan *output data* yang ada.

Langkah selanjutnya adalah *file stream* akan membuka *file* yang telah dibuat tersebut, kemudian melakukan *append*, atau penambahan data kedalam *file* tersebut. Jenis *Append* yang pertama kali dilakukan adalah melakukan *append header*, atau informasi - informasi pada ujung tabel (label), serta informasi *separator* dari kolom. Pada Tugas Akhir ini, seluruh *output file*, menggunakan *separator* kolom berupa *tab* atau simbol \t sedangkan separator baris berupa *line break* atau simbol \n

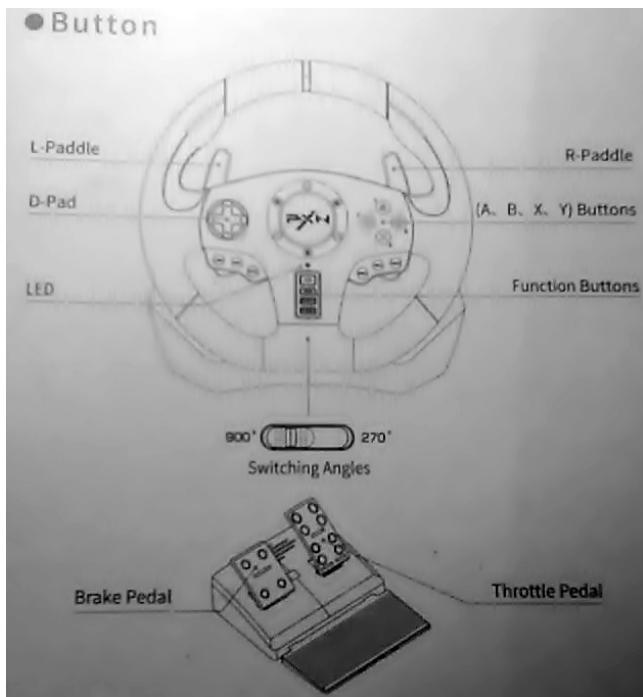
## **3.7 Pengaturan dan Konfigurasi *Steering Wheel Controller***

Konfigurasi kontroller yang perlu dilakukan adalah salah satunya melakukan *mapping* tombol - tombol dari keyboard ke tombol - tombol serta perangkat analog dari *steering wheel controller* yang digunakan, seperti contohnya perangkat analog yang perlu dilaku-



**Gambar 3.22:** Diagram *Log File System*, proses ekstraksi data, dan proses *append data*

kan *sampling* adalah pedal, *sampling* yang dimaksud adalah melakukan pembagian data voltase analog yang ada di pedal menjadi nilai - nilai diskrit yang kemudian dapat dikonversikan ke kecepatan mobil, sesuai nilai - nilai diskrit yang didapatkan. Selain itu juga perlu dilakukan kalibrasi mode sudut dari steering wheel yang memiliki 2 macam mode, yaitu mode 270 derajat putaran, serta 900 derajat putaran. Pada Tugas akhir ini, *steering wheel* akan menggunakan mode 270 derajat, serta proses pergantian gigi (*gear shift*) menggunakan proses manual tanpa kopling. Hal ini dikarenakan

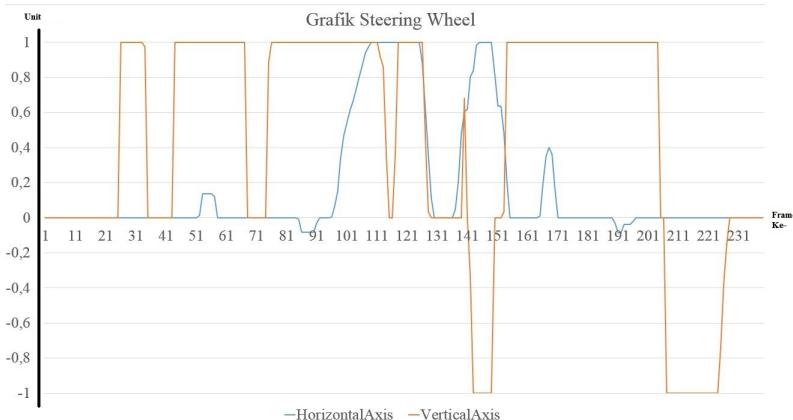


**Gambar 3.23:** Diagram Steering Wheel yang digunakan, *Sumber : Buku Manual PXN Steering Wheel*

keterbatasan perangkat keras yang digunakan.

Berikut pada gambar 3.23 adalah *key map* dari sistem simulator tugas akhir ini. proses kalibrasi pedal gas dan rem disambungkan ke *vertical axis / axis y* dari *unity controller*, maka bisa didapatkan nilai dari pedal atau rem tersebut, berupa data *float* yang memiliki rentang 0 hingga 1. Sedangkan kalibrasi *steering wheel* disambungkan dengan *horizontal axis / axis x* dari *unity controller*, yang memiliki data berupa *float* dengan rentang -1 hingga 1.

Kemudian, data *float* tersebut, dapat dilakukan proses kalkulasi untuk mendapatkan sudut dari pedal gas dan rem, serta sudut dari *steering wheel*, sehingga dapat dilakukan proses *plotting* seperti berikut pada gambar 3.24



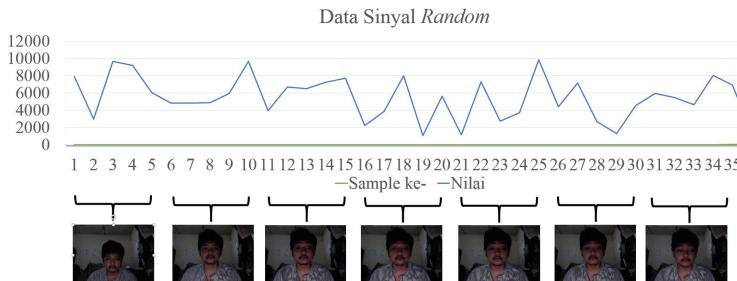
**Gambar 3.24:** Grafik hasil *plotting* data *input* pengemudi

Selain *vertical axis* dan *horizontal axis*, terdapat pula tombol - tombol lain dari *steering wheel controller* yang digunakan. Yaitu tombol **X** untuk menyalakan mesin dari mobil (*starter*), selain itu ada tombol *R-Paddle* dan *L-Paddle* untuk menaikkan dan menurunkan gigi mesin secara berurutan. Proses ini, tidak memerlukan kopling, cukup menekan tombol *L-R-Paddle* untuk menaikkan atau menurunkan gigi mobil.

Di tengah *steering wheel* juga terdapat *switch* untuk mengganti mode *steering* yaitu mode 270 derajat atau 900 derajat.

### 3.8 Pembuatan Modul Pengambilan Data dengan *Microcontroller*

Pembuatan modul pengambilan data serial seperti EEG dan ECG, dapat dilakukan menggunakan Arduino, sehingga perlu di siapkan port USB. Sedangkan pengambilan data berupa gambar wajah pengemudi menggunakan kamera, dapat disambungkan langsung ke *Unity Game Engine* dan dapat langsung disimpan ke *harddrive* komputer simulator. Hal ini dilakukan agar data yang masuk berupa gambar wajah pengemudi serta sinyal - sinyal yang di peroleh oleh *microcontroller* dapat di lakukan pengolahan dan proses analisa. Perlu di perhatikan juga *sampling rate* dari sistem pengambilan sinyal arduino, dengan *capture rate* dari *webcam*, kedua



**Gambar 3.25:** Diagram Sinkronisasi *capture rate* kamera dengan *sampling rate* dari sensor arduino

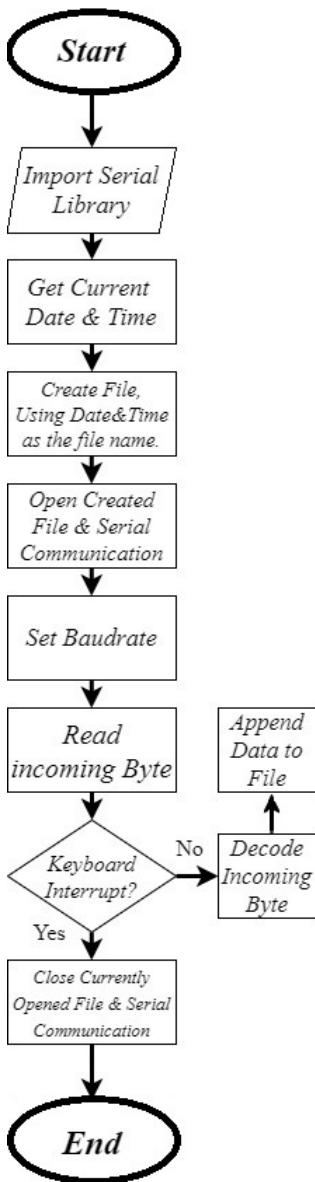
data tersebut perlu di sinkronkan dengan memperhatikan kedua nilai tersebut. *Sampling rate* dari arduino disini bisa didapatkan dari *datasheet* sensor yang digunakan, kemudian dilakukan kalkulasi dengan *margin of error* yang disebabkan oleh kegagalan transmisi data dari arduino ke komputer simulator melalui kabel USB, agar didapatkan hasil yang dapat diterima. Sedangkan *capture rate* dari kamera merupakan seberapa banyak gambar atau *frame* yang diambil oleh kamera tiap *frame* yang di proses oleh simulator. Pada Tugas Akhir ini, target capture rate adalah 1:1. Yang artinya, kamera akan mengambil gambar wajah pengemudi tiap frame game tersebut dijalankan. Berikut pada gambar 3.25 adalah diagram dari penjelasan sinkronisasi kedua data tersebut, menjadi suatu modul analisis sinyal dengan gambar wajah.

### 3.9 Penggabungan Seluruh Sistem Menjadi Satu Modul

Langkah terakhir adalah mengintegrasikan seluruh sistem menjadi satu modul yang utuh, supaya dapat melakukan pengolahan data dengan irisan waktu ( $t$ ) yang bersamaan. Keluaran yang diharapkan adalah, simulator dapat melakukan pengambilan data analog, data citra gambar, serta data - data internal simulasi secara bersamaan dan integrasi. Seperti yang dijelaskan pada bab 3.6.4 hingga bab 3.7, Penggabungan Sub - sub modul, menjadi satu modul utuh adalah seperti berikut

1. Sub-modul pengambilan data internal, mencakup :
  - (a) Sub-Modul Kecepatan Mobil
    - i. Kecepatan Vektor Sumbu  $x,y,z$
    - ii. *Magnitude* Kecepatan Vektor
  - (b) Sub-Modul Informasi Spasial
    - i. Posisi Relatif Mobil terhadap jalan
    - ii. Sudut Euler dan *6 Degree of Freedom* - Pitch, Yaw, Roll
  - (c) Sub-Modul Respon Pengemudi
    - i. Informasi Respon Pengemudi Ketika kendaraan keluar jalur
    - ii. Informasi *Input* dari Steering Wheel Controller (Sudut *Steering Wheel* dan Nilai Tekanan Pedal Gas dan Rem)
2. Sub-modul pengambilan data eksternal, mencakup :
  - (a) Sub-Modul Sinyal dan Citra Wajah
    - i. Sinyal dari sensor, bisa berupa ECG, EEG, atau EOG
    - ii. Citra Wajah tiap frame dari kamera

Pada tugas akhir ini, sub - sub modul dibentuk berdasarkan kebutuhan proses pengujian, namun untuk riset kedepannya, proses pembuatan serta sinkronisasi data - data sub-modul bisa dapat dilakukan penyesuaian sesuai dengan kebutuhan riset tersebut yang menggunakan data dari simulator ini.



Gambar 3.26: Diagram Flow Chart Komunikasi Serial Melalui port COM

*Halaman ini sengaja dikosongkan*

## BAB 4

# PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian dari Tugas Akhir serta analisa dari desain sistem simulator dan implementasinya. Pengujian dibagi menjadi lima bagian antara lain:

1. Pengujian *User Interface* - Pemilihan Jumlah Lajur
2. Pengujian Pengambilan Data - Kecepatan
3. Pengujian Pengambilan Data - Informasi Spasial
4. Pengujian Pengambilan Data - *Response Time* dan *Input* Pengemudi
5. Pengujian Pengambilan Data - Citra *Webcam*
6. Pengujian Pengambilan Data - *Serial Data USB*
7. Pengujian Respon Sinyal dari *Steering Wheel Controller* terhadap simulator
8. Pengujian *User Experience / UX* Pengguna

Dengan dilaksanakannya beberapa pengujian tersebut, sehingga dapat ditarik kesimpulan dari pelaksanaan tugas akhir ini.

### 4.1 Pengujian *User Interface* - Pemilihan Jumlah Lajur

Pada pengujian *user interface*, *user* dapat memilih jumlah lajur yang akan digunakan. Terdapat 4 *scene* yang dapat dipilih dari *user interface*, yaitu 2 lajur (pendek), 2 lajur (panjang) , 3 lajur, serta 4 lajur.

Kesimpulan pada pengujian *user interface* (Gambar 4.1), tombol menu lajur yang ditampilkan dan yang dipilih oleh *user* sudah berkorelasi dengan *scene* yang dimuat oleh game.

### 4.2 Pengujian Pengambilan Data - Kecepatan

Pada pengujian pengambilan data kecepatan, didapatkan data seperti pada tabel 4.4, data yang didapatkan merupakan data mentah (*raw data*) tanpa pengolahan atau kalkulasi tambahan, dari *Unity Game Engine* dengan cara mengakses komponen *RigidBody Game Object* mobil, dengan fungsi `GetComponent<Rigidbody>()`, sehingga didapatkan *object variable* yaitu *velocity* untuk menda-

patkan kecepatan vektor  $x,y,z$ , serta `velocity.magnitude` untuk mendapatkan besaran dari kecepatan vektor tersebut. Pengambilan / *sampling* data dilakukan tiap frame, sehingga tabel 4.4 hanya menampilkan data kurang lebih  $\frac{1}{2}$  detik.

Dapat disimpulkan pengujian ini (tabel 4.4) dapat menghasilkan data dengan tingkat akurasi yang tinggi, dikarenakan data dapat diambil dalam rentang waktu yang cukup kecil tiap framenya (Unit tiap Frame).

### 4.3 Pengujian Pengambilan Data - Informasi Spasial

Informasi spasial pada pengujian ini terdapat 2 macam data, yaitu posisi relatif kendaraan terhadap jalan, serta sudut orientasi kendaraan. Data posisi relatif terhadap jalan didapat dari mengukur jarak terdekat dari titik pusat masa kendaraan ke garis batas lajur kanan dan garis batas lajur kiri, dengan mengakses salah satu *object* dari *Bézier Path Creator*[12] `PathCreator.path` sehingga didapatkan *object method* `GetClosestPointOnPath()` untuk mendapatkan koordinat terdekat dari mobil dari class tersebut, lalu akses class *transform* dari *game object* mobil, untuk mendapatkan koordinat dari *center of mass* mobil simulator menggunakan `transform.position`, kemudian setelah didapatkan 2 koordinat yang ingin diukur jaraknya, sehingga kemudian dapat menggunakan fungsi dari *unity* yaitu `Vector3.Distance()` untuk mendapatkan jarak dari kedua koordinat tersebut.

Kemudian Sudut Orientasi kendaraan pada pengujian ini menggunakan Sudut spasial *Euler* untuk mengetahui orientasi kendaraan pada 3 dimensi, untuk mendapatkan nilai tersebut pada *unity, script* dapat mengakses *class variable* `transform.eulerAngles`.

Maka dari itu, untuk posisi relatif, terdapat 2 macam data yaitu, Jarak dari batas kanan dan Jarak dari batas kiri dengan cara mengukur jarak dari 2 koordinat yaitu, *center of mass* mobil dan titik terdekat dari *center of mass* tersebut, sedangkan untuk Sudut Orientasi kendaraan, didapatkan 3 macam data yaitu, sudut orientasi relatif terhadap sumbu  $x,y,z$  atau *6 Degree of Freedom - Pitch, Yaw, and Roll*. Berikut datanya pada tabel 4.5

Kesimpulan yang dapat ditarik pada pengujian pengambilan data informasi spasial adalah, pada tabel 4.5, dapat diketahui bahwa

lebar dari jalan pada simulator adalah kurang lebih 12 unit, dengan menggunakan 2 data tersebut, yaitu jarak mobil dari batas kiri dan kanan jalan, sehingga dapat diketahui lokasi tepatnya dari mobil. Selain itu, dapat disimpulkan juga bahwa mobil berada di jalan yang datar, dikarenakan nilai sudut *euler* dari sumbu *x* dan sumbu *z* mendekati 0.

#### 4.4 Pengujian Pengambilan Data - *Response Time* dan *Input* Pengemudi

*Response Time* adalah waktu yang menunjukkan seberapa secepat pengemudi kembali ke lajur semula apabila simulator telah mendeteksi mobil keluar dari lajur sebelah kiri. Durasi yang dihitung adalah semenjak mobil keluar dari lajur, hingga mobil kembali ke lajur. Data yang didapat dari simulator berupa waktu dalam sekon, serta durasi frame semenjak mobil keluar dari lajur.

Selain itu diperlukannya perubahan definisi dari 'keluar jalur' apabila sistem mendeteksi bahwa pengemudi akan menyalip kendaraan yang lain. Pada kasus tersebut, pendektsian keluar jalur akan diberikan *flag* bahwasanya terdapat mobil didepan kendaraan pengemudi, sehingga pendektsian sistem keluar jalur sementara dimatikan. Apabila sistem sudah mendeteksi tidak ada kendaraan didepan pengemudi, sistem pendektsian keluar jalur akan berjalan kembali.

Kemudian, pada pengujian ini erat kaitannya dengan mendekksi respon *input* dari pengemudi. Maka dari itu, informasi *input* dari pengemudi juga perlu di lakukan pengambilan datanya agar diketahui sudut dari *steering wheel* serta tekanan pedal gas dan rem. Data - data *input* ini penting dikarenakan berkaitan dengan data respon pengemudi ketika pengemudi mengetahui adanya rintangan atau objek tidak terduga ketika di jalan. Dengan menganalisa grafik sudut *steering wheel* dan tekanan pedal gas dan rem, akan didapatkan redundansi data selain citra wajah pengemudi serta data - data biometrik pengemudi seperti EEG dan ECG, untuk analisa proses kapan terjadinya *microsleep* pada pengemudi. Grafik data *input* dari pengemudi bisa di lihat pada gambar 4.5

## **4.5 Pengujian Pengambilan Data - *Citra Webcam***

Pada gambar 4.2, bisa dilihat hasil dari *capture webcam* yang telah di *encode* menjadi format *.png*.

Kesimpulan dari pengujian pengambilan data *webcam* ini menunjukkan bahwa, harus dipastikan bahwa *source* dari *webcam* telah tepat terdeteksi sebagai suatu alat pengambil citra pada unity. Kemudian dipastikan apakah hasil *encode* telah menghasilkan keluaran gambar yang diinginkan. Terjadinya kesalahan pendekripsi *camera source* atau terjadinya kesalahan pada proses *encode*, akan menyebabkan hasil keluaran citra *webcam* tidak sesuai dengan yang diharapkan.

## **4.6 Pengujian Pengambilan Data - *Serial Data USB***

*Serial Data* disini ialah, data yang didapatkan dari komunikasi via *serial USB*. Desain sistem komunikasi serial menggunakan port COM, bisa dilihat pada gambar 4.5.

Pada tugas akhir ini, sistem komunikasi serial arduino dengan - PC simulator adalah menggunakan script python untuk mengakses *Port COM* yang menghubungkan Arduino dengan PC. Setelah *Port COM* Diakses oleh script python, script python akan menerima *byte* data dari Arduino, kemudian script tersebut akan merubah data menjadi angka bilangan bulat atau *integer* tiap 4 *byte* (*32 bit*) data yang masuk.

Kemudian setelah data dirubah menjadi data angka bilangan bulat, script python akan mengakses *file system* dari komputer, dan akan membuat file dengan ekstensi *.csv*, lalu menambahkan angka tersebut kedalam file yang baru dibuat tersebut, serta memformat dengan *column separator* '*\t*' dan *row separator* '*\n*'.

## **4.7 Pengujian Respon Sinyal dari *Steering Wheel Controller* terhadap simulator**

Pada pengujian ini, dikarenakan sistem memiliki mekanisme masukan dari pengguna berupa *steering wheel controller*, maka diperlukan pengujian respon dari *steering wheel controller* terhadap proses - proses berjalan di *unity*. Kegiatan pengujian adalah dengan membandingkan sinyal dari *steering wheel controller* yang berjalan sebelum *script - script* lain pada *unity*, dengan sinyal yang setelah melalui proses - proses / *script* lain pada *unity*. Hal ini dapat dilakukan dengan cara membandingkan sinyal yang dihasilkan oleh *script* yang berjalan sebelumnya dengan sinyal yang dihasilkan oleh *script* yang berjalan setelahnya.

kukan dengan menerapkan fitur *Script Execution Order* pada unity. Grafik data sinyal tersebut bisa dilihat pada gambar 4.6 dan 4.7, sedangkan grafik analisa *error* kuantitatif nilai error bisa dilihat pada gambar 4.8

Dari analisa kuantitatif terhadap nilai error tersebut, dapat dilihat bahwa grafik nilai error selalu mendekati 0 persen ketika *run-time*, sehingga dapat disimpulkan bahwa *input* dari *steering wheel* tidak terlalu terpengaruh proses yang berjalan pada unity, sehingga grafik terlihat saling menindih.

#### **4.8 Pengujian *User Experience / UX* dari Pengguna**

Pengujian kepuasan pengguna dilakukan dengan melakukan pengujian terhadap subjek pengendara mobil kemudian dilakukan proses pengisian kuesioner setelah pengguna mencoba alat simulator. Jumlah responden dalam pengujian kepuasan pengguna ini sebanyak 3 responden, terdiri dari keluarga dekat penulis. Hal ini dikarenakan keterbatasan situasi dan kondisi dimana ketika pengujian dilakukan sedang adanya pandemi *COVID-19*, sehingga hal tersebut membatasi jumlah responden yang bersukarela mencoba alat simulator ini. Responden menerima 17 buah pernyataan sesuai pada tabel 4.1 Opsi tingkat persetujuan yang disediakan adalah sebagai berikut :

1. Sangat Tidak Setuju (STS)
2. Tidak Setuju (TS)
3. Netral (N)
4. Setuju (S)
5. Sangat Setuju (SS)

Terdapat 3 macam pertanyaan yang diajukan, yaitu pertanyaan dengan tujuan untuk mengetahui bagaimana pengetahuan pengguna tentang teknologi simulasi khususnya teknologi simulasi berkenaan, pertanyaan dengan tujuan untuk mengukur bagaimana kepuasan pengguna terhadap tampilan serta realitas simulasi, serta yang terakhir pertanyaan dengan tujuan untuk mengetahui bagaimana pendapat pengguna tentang teknologi simulator yang diterapkan untuk proses pengambilan data di lapangan.

Pengujian kepuasan pengguna dihitung dari banyaknya persentase total poin yang didapatkan dari total maksimal poin untuk

**Tabel 4.1:** Skenario kuesioner untuk pengujian kepuasan pengguna

No	Pertanyaan
1	Saya mengetahui tentang teknologi simulasi
2	Saya mengetahui tentang game yang melibatkan mengendarai mobil
3	Saya pernah mengemudikan mobil virtual (game, simulator sim, dll)
4	Saya merasa menu User Interface Pemilihan Jumlah lajur aplikasi simulator sudah jelas
5	Saya merasa menu User Interface Pemilihan Jumlah lajur aplikasi simulator sudah menarik
6	Saya merasa user interface dashboard mobil sudah jelas
7	Saya merasa user interface dashboard mobil sudah menarik
8	Saya merasa user interface tutorial cara mengemudikan mobil sudah jelas
9	Saya merasa penggunaan perangkat keras steering wheel controller sudah intuitif
10	Saya merasa jumlah lajur yang yang dapat dipilih sudah mewakili berbagai macam kondisi jalan yang sesungguhnya
11	Saya merasa simulasi yang disajikan sudah cukup realistik
12	Saya merasa pemandangan yang disediakan pada tiap - tiap lajur sudah cukup menarik
13	Saya merasa tertarik akan teknologi simulasi berkendara untuk riset
14	Saya merasa teknologi simulasi dapat mengantikan proses pengambilan data di lapangan
15	Saya menjadi tertarik untuk menggunakan simulator ini
16	Saya setuju perangkat ini untuk diterapkan di institusi penelitian di Indonesia
17	Saya setuju perangkat ini dapat mendukung perkembangan riset deteksi pengemudi mengantuk

**Tabel 4.2:** Hasil pengujian kepuasan pengguna.

Pernyataan	Jawaban				
	STS	TS	N	S	SS
Pernyataan 1	0,00%	0,00%	0,00%	66,67%	33,33%
Pernyataan 2	33,33%	0,00%	0,00%	33,33%	33,33%
Pernyataan 3	33,33%	0,00%	0,00%	33,33%	33,33%
Pernyataan 4	0,00%	0,00%	33,33%	33,33%	33,33%
Pernyataan 5	0,00%	0,00%	0,00%	100,00%	0,00%
Pernyataan 6	0,00%	0,00%	33,33%	33,33%	33,33%
Pernyataan 7	0,00%	0,00%	33,33%	66,67%	0,00%
Pernyataan 8	0,00%	0,00%	0,00%	33,33%	66,67%
Pernyataan 9	0,00%	0,00%	100,00%	0,00%	0,00%
Pernyataan 10	0,00%	0,00%	33,33%	33,33%	33,33%
Pernyataan 11	0,00%	0,00%	0,00%	33,33%	66,67%
Pernyataan 12	0,00%	0,00%	33,33%	33,33%	33,33%
Pernyataan 13	0,00%	0,00%	0,00%	33,33%	66,67%
Pernyataan 14	0,00%	0,00%	33,33%	33,33%	33,33%
Pernyataan 15	0,00%	0,00%	66,67%	0,00%	33,33%
Pernyataan 16	0,00%	0,00%	0,00%	33,33%	66,67%
Pernyataan 17	0,00%	0,00%	0,00%	33,33%	66,67%

setiap pernyataan yang disediakan (likert scale). Berdasarkan parameter tersebut, setiap opsi memiliki poin yang berbeda yaitu SS=5 poin, S=4 poin, N=3 poin, TS=2 poin, dan STS=1 poin. Semakin besar presentasenya, maka semakin *valid* pernyataan tersebut.

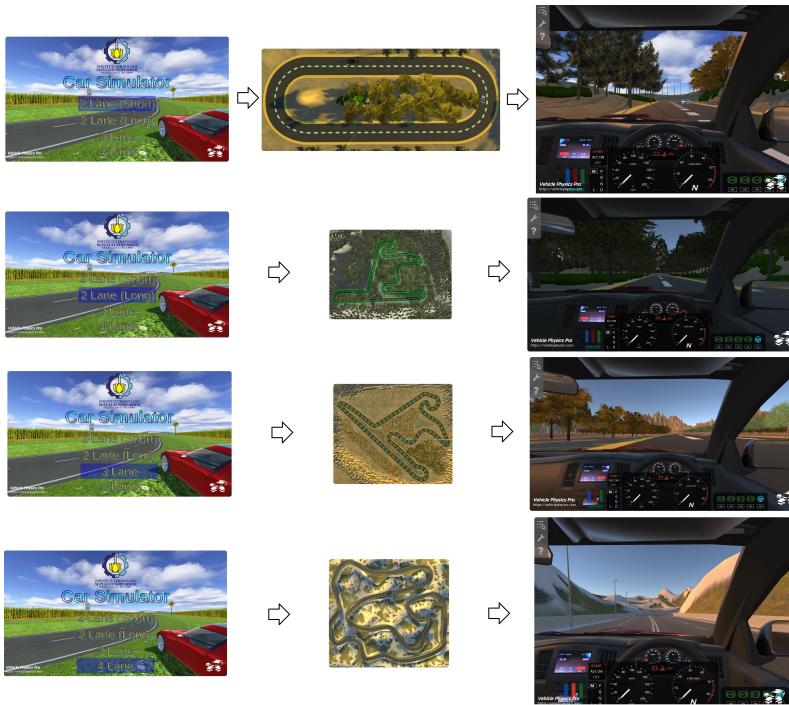
Berikut hasil dari proses pengujian tingkat kepuasan pengguna (tabel 4.2)

**Tabel 4.3:** Tabel *Input* Pengemudi Selama Kurang Lebih 1 Detik

<i>HorizontalAxis</i>	<i>VerticalAxis</i>	<i>Frame</i>	<i>Date Time</i>
0	0	74	01.22.25.590387
0	0,8822352	75	01.22.25.748759
0	1	76	01.22.25.923766
0	1	77	01.22.26.234632
0	1	78	01.22.26.413535
0	1	79	01.22.26.570931
0	1	80	01.22.26.736147
0	1	81	01.22.26.892566
0	1	82	01.22.27.071471
0	1	83	01.22.27.225934
0	1	84	01.22.27.390172
-0,007369441	1	85	01.22.27.574942
-0,08203134	1	86	01.22.27.756776
-0,08203134	1	87	01.22.27.921995
-0,08203134	1	88	01.22.28.186928
-0,08203134	1	89	01.22.28.361919
-0,08203134	1	90	01.22.28.542779
-0,03229748	1	91	01.22.28.699196
-0,00187061	1	92	01.22.28.865390
0	1	93	01.22.29.040383
0	1	94	01.22.29.211465
0	1	95	01.22.29.367885
0,001748414	1	96	01.22.29.528212
0,06535155	1	97	01.22.29.686586
0,1510722	1	98	01.22.30.072742
0,3418816	1	99	01.22.30.228184
0,4662774	1	100	01.22.30.431527
0,5492486	1	101	01.22.30.600656
0,6156012	1	102	01.22.30.783469
0,6681455	1	103	01.22.30.942821
0,7344981	1	104	01.22.31.153007
0,8036612	1	105	01.22.31.317244
0,8700138	1	106	01.22.31.499081
0,9363663	1	107	01.22.31.668210
0,972292	1	108	01.22.31.833426

**Tabel 4.4:** Data Informasi Kecepatan

No.	VelX	VelY	VelZ	VelMagn	AvgFrameRate
	(Unit/Frame)			(Frame/Sec)	
1	1,0,E-05	-1,1,E-04	4,4,E-06	1,1,E-04	52,2394
2	1,1,E-05	-9,9,E-05	3,7,E-06	9,9,E-05	52,2469
3	1,2,E-05	-8,3,E-05	3,3,E-06	8,4,E-05	52,2751
4	1,2,E-05	-6,5,E-05	2,8,E-06	6,6,E-05	52,2905
5	1,2,E-05	-4,5,E-05	2,3,E-06	4,7,E-05	52,3272
6	1,1,E-05	-2,6,E-05	1,9,E-06	2,8,E-05	52,3012
7	1,0,E-05	-5,8,E-06	1,6,E-06	1,2,E-05	52,3211
8	8,6,E-06	1,2,E-05	1,3,E-06	1,5,E-05	52,3379
9	7,1,E-06	2,8,E-05	1,1,E-06	2,9,E-05	52,3672
10	5,4,E-06	4,2,E-05	9,2,E-07	4,2,E-05	52,3583
11	3,7,E-06	5,3,E-05	5,0,E-07	5,3,E-05	52,3787
12	3,7,E-06	5,3,E-05	5,0,E-07	5,3,E-05	52,4137
13	2,1,E-06	6,0,E-05	1,6,E-07	6,1,E-05	52,4484
14	6,2,E-07	6,5,E-05	-4,1,E-07	6,5,E-05	52,4828
15	-4,7,E-07	6,6,E-05	-9,2,E-07	6,6,E-05	52,4952
16	-1,3,E-06	6,5,E-05	-1,3,E-06	6,5,E-05	52,5117
17	-1,9,E-06	6,1,E-05	-1,8,E-06	6,1,E-05	52,5453
18	-2,1,E-06	5,5,E-05	-1,8,E-06	5,5,E-05	52,5479
19	-2,1,E-06	5,5,E-05	-1,8,E-06	5,5,E-05	52,5684
20	-2,3,E-06	4,7,E-05	-2,4,E-06	4,8,E-05	52,6013
21	-2,3,E-06	3,9,E-05	-2,5,E-06	3,9,E-05	52,6340
22	-2,2,E-06	2,8,E-05	-2,2,E-06	2,9,E-05	52,6664
23	-2,1,E-06	1,7,E-05	-1,8,E-06	1,8,E-05	52,6986
24	-2,1,E-06	7,0,E-06	-1,7,E-06	7,5,E-06	52,7100
25	-2,0,E-06	-2,4,E-06	-1,2,E-06	3,4,E-06	52,6537
26	-2,0,E-06	-1,1,E-05	-8,4,E-07	1,1,E-05	52,6855
27	-1,9,E-06	-1,8,E-05	-1,1,E-06	1,9,E-05	52,7169
28	-1,9,E-06	-1,8,E-05	-1,1,E-06	1,9,E-05	52,7482
29	-1,9,E-06	-2,5,E-05	-5,4,E-07	2,5,E-05	52,7733
30	-1,9,E-06	-2,5,E-05	-5,4,E-07	2,5,E-05	52,7733

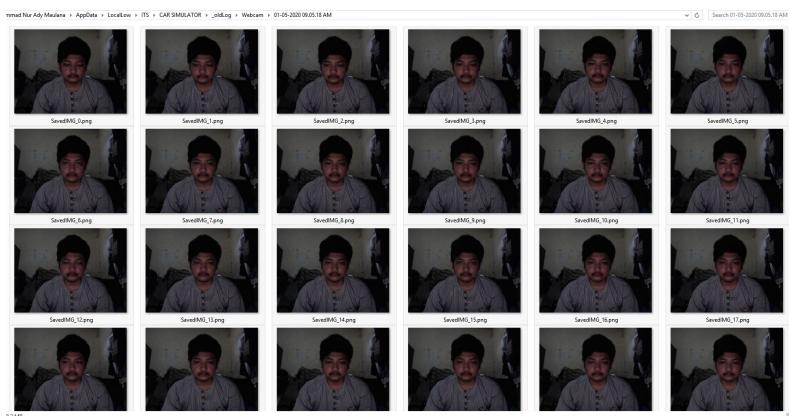


a. Tombol yang dipilih

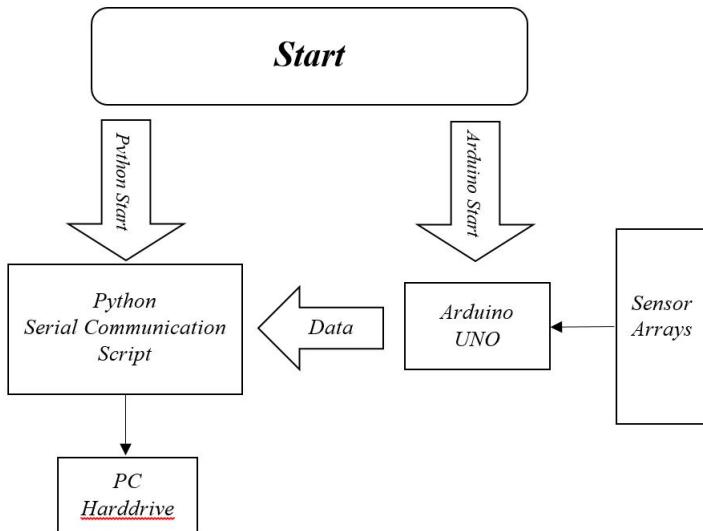
b. Lajur yang harapkan

c. Lajur yang dimuat

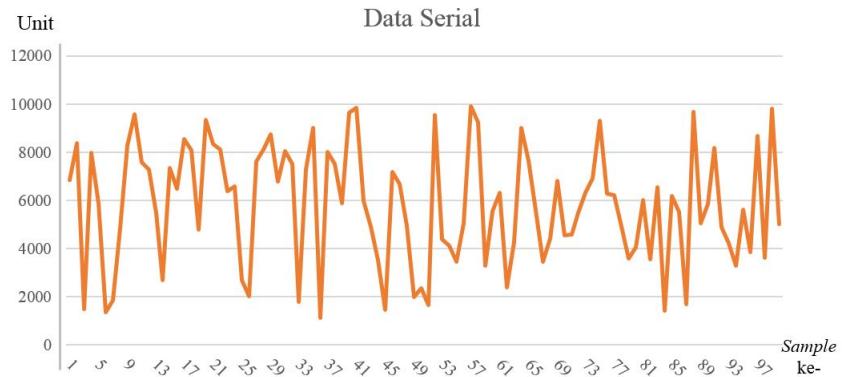
**Gambar 4.1:** Korelasi *user interface* pemilihan lajur dengan *scene* yang dimuat oleh simulator



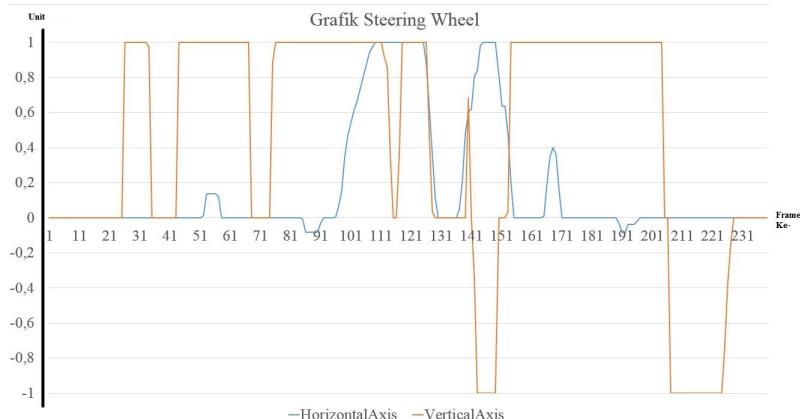
**Gambar 4.2:** Frame webcam yang tersimpan di dalam harddrive



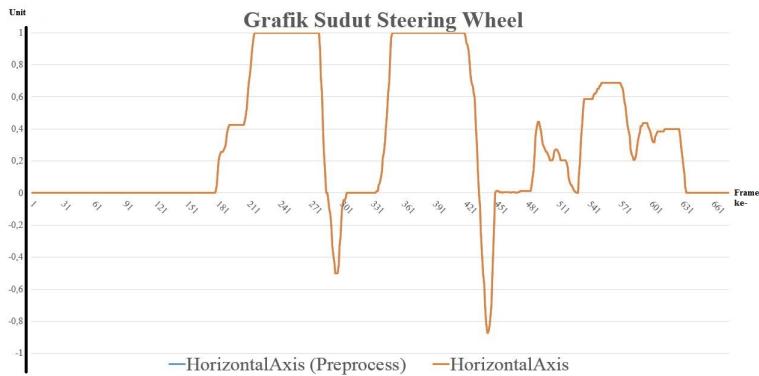
**Gambar 4.3:** Diagram Sistem Komunikasi Arduino Dengan PC Simulator



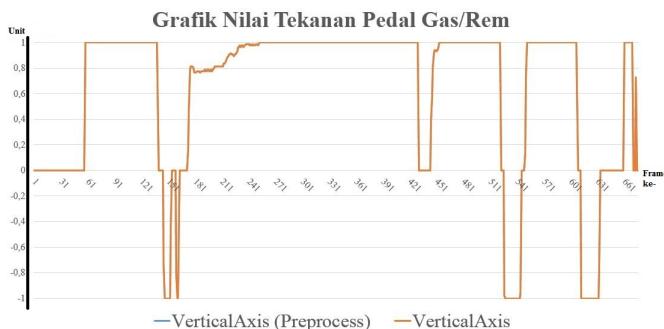
**Gambar 4.4:** Contoh data serial yang diterima oleh arduino



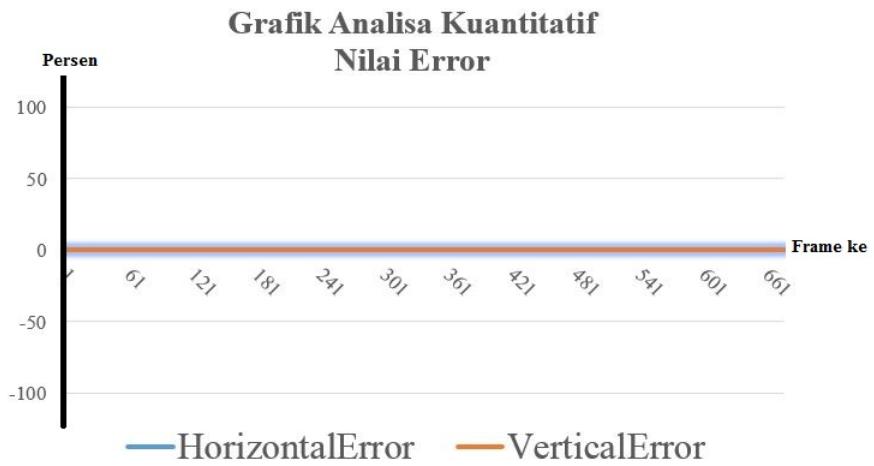
**Gambar 4.5:** Input dari pengemudi - *Horizontal Axis* adalah Data sudut steering wheel, *Vertical Axis* adalah data nilai tekanan pedal gas / rem



**Gambar 4.6:** Perbandingan sinyal dari *steering wheel controller*, sebelum dan sesudah grafik unity - *Horizontal Axis / Sudut steering wheel*



**Gambar 4.7:** Perbandingan sinyal dari *steering wheel controller*, sebelum dan sesudah grafik unity - *Vertical Axis / Nilai tekanan pedal gas/rem*



**Gambar 4.8:** Grafik analisa kuantitatif nilai error dari *steering wheel controller*

**Tabel 4.5:** Data Informasi Spasial

No.	Dist. Left (Unit)	Dist. Right (Unit)	Euler Angles (Degrees)		
			pitch	yaw	roll
1	4,290802	7,87863	(0,0,	91,7,	0,0)
2	4,290906	7,878543	(0,1,	91,7,	0,0)
3	4,290405	7,879219	(0,5,	91,7,	0,0)
4	4,290458	7,879376	(0,7,	91,7,	0,0)
5	4,290488	7,879422	(0,7,	91,7,	0,0)
6	4,29053	7,879436	(0,7,	91,7,	0,0)
7	4,29057	7,879446	(0,6,	91,7,	0,0)
8	4,290622	7,879434	(0,6,	91,7,	0,0)
9	4,290658	7,879415	(0,5,	91,7,	0,0)
10	4,290717	7,879354	(0,4,	91,7,	0,0)
11	4,290747	7,87931	(0,4,	91,7,	0,0)
12	4,290762	7,879263	(0,3,	91,7,	0,0)
13	4,290767	7,879231	(0,3,	91,7,	0,0)
14	4,290767	7,879195	(0,3,	91,7,	0,0)
15	4,29076	7,879167	(0,2,	91,7,	0,0)
16	4,290745	7,879142	(0,2,	91,7,	0,0)
17	4,290729	7,879126	(0,2,	91,7,	0,0)
18	4,290688	7,879111	(0,2,	91,7,	0,0)
19	4,290656	7,879109	(0,3,	91,7,	0,0)
20	4,290623	7,879117	(0,3,	91,7,	0,0)
21	4,290589	7,879128	(0,3,	91,7,	0,0)
22	4,290563	7,879138	(0,3,	91,7,	0,0)
23	4,290539	7,879152	(0,4,	91,7,	0,0)
24	4,290514	7,87917	(0,4,	91,7,	0,0)
25	4,290488	7,879189	(0,4,	91,7,	0,0)
26	4,290459	7,879222	(0,5,	91,7,	0,0)
27	4,290454	7,879234	(0,5,	91,7,	0,0)
28	4,290452	7,879245	(0,5,	91,7,	0,0)
29	4,290462	7,87925	(0,5,	91,7,	0,0)

**Tabel 4.6:** Tabel *Response Time*

No.	Start	Return	Duration	
			Seconds	Frame
1	1:00:37	1:00:43	6,707	366,538
2	1:03:15	1:03:16	1,542	84,517
3	1:06:21	1:06:30	9,048	488,140
4	1:08:05	1:08:08	3,141	177,906
5	1:09:58	1:10:07	9,863	594,838
6	1:10:19	1:10:29	0,878	50,819
7	1:12:47	1:12:48	1,936	111,184
8	1:13:00	1:13:00	0,947	54,926
9	1:14:27	1:14:31	4,245	235,470
10	1:16:25	1:16:34	9,917	579,450
11	1:19:26	1:19:27	1,972	116,880
12	1:20:05	1:20:05	0,266	14,058
13	1:21:40	1:21:50	0,911	56,291
14	1:27:39	1:27:49	0,734	42,888
15	1:36:56	1:36:56	0,972	53,664
16	1:37:04	1:37:12	8,662	475,197
17	1:38:35	1:38:40	5,015	280,840
18	1:39:44	1:39:49	5,885	307,903
19	1:40:17	1:40:22	5,361	320,641
20	1:43:05	1:43:09	4,009	243,587
21	1:45:51	1:45:53	2,657	150,546
22	1:48:19	1:48:23	4,068	220,445
23	1:48:27	1:48:30	3,068	168,034
24	1:51:09	1:51:17	8,246	447,923
25	1:55:19	1:55:25	6,886	367,437
26	1:59:21	1:59:31	0,553	33,473

**Tabel 4.7:** Data Deteksi *Colission*

No.	ColissionStart	GameObjectTag	GameObjectName
1	01.49.16 PM	Boundary	LeftLane
2	01.49.18 PM	Boundary	LeftLane
3	01.49.59 PM	Boundary	LeftLane
4	01.50.13 PM	Boundary	RightLane
5	01.50.39 PM	Boundary	RightLane
6	01.51.39 PM	OtherVehicle	SportsVehicleYellow
7	01.53.21 PM	OtherVehicle	SportsVehicleYellow
8	01.59.11 PM	OtherVehicle	SportsVehicleYellow
9	02.30.15 PM	Boundary	RightLane
10	02.30.48 PM	Boundary	RightLane

*Halaman ini sengaja dikosongkan*

## BAB 5

# PENUTUP

### 5.1 Kesimpulan

Dari hasil pengujian yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Tombol - tombol jumlah lajur pada *Interface - Main Menu* telah berkorelasi dengan benar terhadap *scene* yang dimuat
2. Pengujian pengambilan data kecepatan menghasilkan data berdasarkan kalkulasi vektor global, diperlukan pengujian untuk memverifikasi keakuratan data tersebut.
3. Pengujian pengambilan data spasial menghasilkan data relatif posisi mobil terhadap garis pinggir jalan, dapat disimpulkan data tersebut dapat digunakan untuk suplemen data pengujian pengambilan data *response time*, diperlukan pengujian untuk memverifikasi keakuratan data tersebut
4. Pengujian citra webcam memiliki *performance cost* yang sangat tinggi, yaitu *execution time* tiap framenya mencapai 250-550 milisekon, hal ini disebabkan oleh proses unity dalam melakukan *encoding* data berupa *texture* menjadi suatu citra. Permasalahan ini ada pada level perangkat keras (GPU dan CPU). Diperlukannya suatu kompromi antara *performance* dan akurasi
5. Proses kalkulasi data serta berjalannya *script* utama pada *unity* tidak terlalu berpengaruh terhadap respon *steering wheel*, nilai error mendekati 0 persen atau akurat hingga 5 angka dibelakang koma (0.000001%)(gambar 4.8) hal ini disebabkan oleh kecilnya *performance cost* dari *script* tersebut (20-100 milisekon).
6. Pengujian UX tidak konklusif, yang disebabkan oleh situasi dan kondisi pandemi *COVID-19*. Diperlukannya pengujian UX dengan jumlah responden yang lebih banyak sehingga dapat mewakili target demografi pengguna yang dituju.

### 5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

1. Melakukan *refactor* / penataan ulang terhadap struktur source code.
2. Mengurangi *performance cost* dari source code.
3. Meningkatkan estetik dari simulator mulai dari UI, kualitas objek 3D, serta animasi - animasi atau detail - detail lain yang dapat meningkatkan imersifitas dari simulator.
4. Menambah kapabilitas dari simulator dengan menambah jenis data yang bisa diambil oleh simulator.
5. Melakukan survey terhadap pengguna untuk fitur yang perlu ditambahkan pada simulator ini
6. Melakukan survei kuesioner dengan jumlah responden yang lebih banyak agar mewakili target demografi pengguna yang dituju

## DAFTAR PUSTAKA

- [1] Wikipedia contributors, “6 degrees of freedom,” 2020. [Online; accessed 6-May-2020]. (Dikutip pada halaman xiii, 9).
- [2] Wikipedia contributors, “Shanghai international circuit — Wikipedia, the free encyclopedia,” 2020. [Online; accessed 6-May-2020]. (Dikutip pada halaman xiii, 26).
- [3] “Lalu lintas dan angkutan jalan nomor 22 tahun 2009.” Terakhir diakses pada tanggal 30 Oktober 2019. (Dikutip pada halaman 1).
- [4] G. Kecklund and T. Åkerstedt, “Sleepiness in long distance truck driving: an ambulatory eeg study of night driving,” *Ergonomics*, vol. 36, no. 9, pp. 1007–1017, 1993. (Dikutip pada halaman 1).
- [5] H. De Rosario, J. S. Solaz, N. Rodriguez, and L. M. Bergasa, “Controlled inducement and measurement of drowsiness in a driving simulator,” *IET intelligent transport systems*, vol. 4, no. 4, pp. 280–288, 2010. (Dikutip pada halaman 1, 3).
- [6] T. Åkerstedt and M. Gillberg, “Subjective and objective sleepiness in the active individual,” *International Journal of Neuroscience*, vol. 52, no. 1-2, pp. 29–37, 1990. (Dikutip pada halaman 1, 7).
- [7] K. Kaida, M. Takahashi, T. Åkerstedt, A. Nakata, Y. Otsuka, T. Haratani, and K. Fukasawa, “Validation of the karolinska sleepiness scale against performance and eeg variables,” *Clinical Neurophysiology*, vol. 117, no. 7, pp. 1574–1581, 2006. (Dikutip pada halaman 1, 5).
- [8] M. Ingre, T. Åkerstedt, B. Peters, A. Anund, and G. Kecklund, “Subjective sleepiness, simulated driving performance and blink duration: examining individual differences,” *Journal of sleep research*, vol. 15, no. 1, pp. 47–53, 2006. (Dikutip pada halaman 2, 5).

- [9] P. L. Olson and M. Sivak, “Perception-response time to unexpected roadway hazards,” Human factors, vol. 28, no. 1, pp. 91–96, 1986. (Dikutip pada halaman 2, 5, 7).
- [10] Wikipedia contributors, “Bézier curve — Wikipedia, the free encyclopedia,” 2020. [Online; accessed 29-April-2020 ]. (Dikutip pada halaman 2, 8).
- [11] Wikipedia contributors, “Driving simulator — Wikipedia, the free encyclopedia,” 2020. [Online; accessed 29-April-2020 ]. (Dikutip pada halaman 7).
- [12] Sebastian Lague, “Bézier path creator,” 2020. [Online; accessed 11-July-2020]. (Dikutip pada halaman 42).

## BIOGRAFI PENULIS



Penulis adalah salah satu mahasiswa S1 Departemen Teknik Komputer Fakultas Teknologi Elektro Institut Teknologi Sepuluh Nopember ITS. Penulis sangat tertarik dengan riset - riset yang berhubungan dengan sistem tertanam (*embedded system*), grafika komputer (*computer graphics*), dan visi komputer (*Computer Vision*).

*Halaman ini sengaja dikosongkan*