



**RANCANG BANGUN DETEKSI KECELAKAAN PADA MOBIL LISTRIK
DENGAN *EXPONENTIAL SMOOTHING***

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

RIDLO ALFATORIQ PUTRO

18410200033

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

**RANCANG BANGUN DETEKSI KECELAKAAN PADA MOBIL LISTRIK
DENGAN *EXPONENTIAL SMOOTHING***

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama : Ridlo Alfatoriq Putro
NIM : 18410200033
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2022

Tugas Akhir
RANCANG BANGUN DETEKSI KECELAKAAN PADA MOBIL LISTRIK
DENGAN *EXPONENTIAL SMOOTHING*

Dipersiapkan dan disusun oleh:

Ridlo Alfatoriq Putro

NIM : 18410200033

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 1 Juli 2022

Susunan Dewan Pembahas

Pembimbing:

I. Harianto, S.Kom., M.Eng

NIDN: 0722087701

II. Yosefine Triwidyastuti, M.T.

NIDN: 0729038504

Pembahas:

Pauladie Susanto, S.Kom., M.T.


NIDN: 0729047501



DN: cn=Harianto, o=ID, o=Universitas
dinamika, ou=Fakultas Teknologi dan
Informatika, email=hari@idna.mik.aac,
id
Date: 2022.07.19 09:01:34 +07'00'



Digitally signed by Universitas
Dinamika
Date: 2022.07.19 10:05:42 +07'00'



cn=Pauladie Susanto, o=FTI
Undika, ou=Prodi S1 TK,
email=pauladie@dinamika.ac.id,
c=ID
2022.07.15 13:43:20 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



Digitally signed by
Universitas Dinamika
Date: 2022.07.19
16:07:00 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



UNIVERSITAS
Jangan takut dengan bayang-bayang yang belum terjadi”
~ Ridlo Alfatoriq Putro ~
Dinamika



UNIVERSITAS

Dipersembahkan kepada Bapak, Ibu, serta Keluarga Besar. Tidak bisa dipungkiri kedua Orang Tua menjadi salah satu motivasi terbesar untuk menyelesaikan Tugas Akhir ini. Beserta semua orang yang selalu membantu, mendukung, memberi masukan, dan memberi motivasi sehingga tetap berusaha dan belajar agar bermanfaat bagi orang lain.

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Ridlo Alfatoriq Putro

NIM : 18410200033

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Tugas Akhir

Judul Karya : **RANCANG BANGUN DETEKSI KECELAKAAN PADA
MOBIL LISTRIK DENGAN *EXPONENTIAL
SMOOTHING***

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

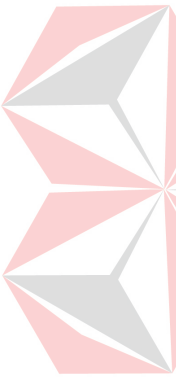
Surabaya, 15 Juni 2022

Yang menyatakan



Ridlo Alfatoriq Putro
NIM : 18410200033

ABSTRAK



Transportasi merupakan kendaraan yang penting untuk menunjang pembangunan, ekonomi dan keberhasilan daerah. Salah satu transportasi yang diatur oleh pemerintah adalah percepatan program mobil atau kendaraan listrik bertenaga baterai. Mobil listrik memiliki keunggulan diantaranya tidak menimbulkan polusi udara, tidak membutuhkan bahan bakar fosil, serta fitur yang menguntungkan pengguna. Kecelakaan Mobil tidak terlepas dari kemacetan yang dapat memperlambat penanganan korban jika tidak dilakukan pertolongan dengan cepat. Dengan demikian, diciptakan alat untuk mendeteksi kecelakaan dengan menggunakan metode *Exponential Smoothing*. Beberapa komponen pada alat adalah sensor MPU6050, *flexible sensor*, dan GPS. Adapun metode *Exponential Smoothing* dapat mengurangi *noise* sinyal *output* dari *flexible sensor* dengan cara melakukan *trial error* untuk menentukan nilai permulusan alpha. Hasil pengujian pada *flexibel sensor* menggunakan metode *Exponential Smoothing* mampu mengurangi *noise* tanpa menghilangkan data asli dengan nilai permulusan atau alpha 0.3 serta berkurang sebesar 10.4853% dari data asli. Sensor MPU6050 memiliki nilai error sebesar 0.8164% pada sudut pitch dan 1.6140% pada sudut roll dibandingkan dengan busur. Sedangkan pengujian GPS Neo6M mampu mendeteksi lokasi dibandingkan dengan *handphone* dengan nilai selisih 168.8000 meter, serta dibandingkan dengan Garmin GPS nilai selisih sebesar 8.8150 meter. Selanjutnya uji coba pengiriman data GPS ke Telegram Bot menghasilkan nilai rata-rata pengiriman sebesar 4.1490 detik.

Kata Kunci: *Exponential Smoothing, flexibel sensor, Sensor MPU6050, GPS Neo 6M, Telegram Bot.*

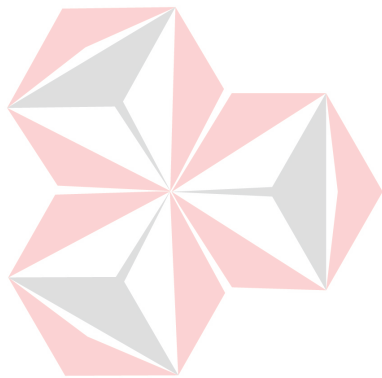
KATA PENGANTAR

Segala puji atas kehadiran Tuhan Yang Maha Esa, karena dengan rahmat dan karunia-Nya, Sholawat serta salam tidak lupa penulis curahkan kepada Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “ Rancang Bangun Deteksi Kecelakaan Pada Mobil Listrik Dengan *Exponential Smoothing*”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Teknik pada program studi S1 Teknik Komputer Universitas Dinamika. Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

1. Orang Tua penulis, yang telah memberikan kontribusi besar atas apa yang telah penulis lakukan hingga dapat menyelesaikan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika, dan selaku Penguji dalam Tugas Akhir ini.
4. Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing I yang selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Ibu Yosefine Triwidyastuti, M.T., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Seluruh rekan seperjuangan Teknik Komputer angkatan 2018 serta rekan-rekan Himpunan Mahasiswa S1 Teknik Komputer Universitas Dinamika.

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun bahasa yang digunakan. Penulis berharap laporan Tugas Akhir ini dapat bermanfaat bagi semua pihak dan dapat menjadi bahan acuan untuk penelitian selanjutnya. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun

Surabaya, 13 Juni 2022



UNIVERSITAS
Dinamika

Penulis

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II	4
LANDASAN TEORI.....	4
2.1 Flexible Sensor	4
2.2 NodeMCU ESP32	5
2.3 Global Position System Neo 6M (GPS)	6
2.4 Sensor MPU 6050.....	7
2.5 Chat Telegram Bot	7
2.6 <i>Exponential Smoothing</i>	7
BAB III.....	9
METODOLOGI PENELITIAN	9
3.1 Blok Diagram	9

3.2	Flowchart Sistem	10
3.3	Perancangan Elektronika dan perangkat lunak.....	12
3.3.1	Perancangan Flexible Sensor	12
3.3.2	Perancangan sensor MPU 6050	13
3.3.3	Perancangan GPS Neo 6M.....	14
3.3.4	Perancangan Keseluruhan Sistem	15
3.4	Rancangan Mekanik	15
BAB IV	17
HASIL DAN PEMBAHASAN	17
4.1	Hasil pengujian Flexible Sensor	17
4.2	Hasil pengujian <i>Flexible Sensor</i> menggunakan <i>Exponential Smoothing</i>	20
4.3	Hasil pengujian perbandingan nilai Alpha pada Metode <i>Exponential Smoothing</i>	22
4.4	Hasil pengujian GPS Neo 6M	24
4.5	Hasil pengiriman Data GPS ke Telegram	27
4.6	Hasil pengujian Sensor MPU 6050	29
4.7	Hasil pengujian Keseluruhan Sistem.....	33
BAB V	37
PENUTUPAN	37
5.1	Kesimpulan.....	37
5.2	Saran	37
DAFTAR PUSTAKA	39
LAMPIRAN	41
BIODATA PENULIS	77

DAFTAR GAMBAR

Gambar 2.1 Rangkaian Flexible Sensor	4
Gambar 2.2 Flexible Sensor	5
Gambar 2.3 NodeMCU ESP32	5
Gambar 2.4 GPS Neo 6M	6
Gambar 2.5 Sensor MPU 6050	7
Gambar 3.1 Blok Diagram	9
Gambar 3.2 Flowchart Sistem.....	11
Gambar 3.3 Rangkaian Skematik <i>Flexible Sensor</i>	13
Gambar 3.4 Rangkaian Skematik sensor MPU 6050.....	14
Gambar 3.5 Rangkaian Skematik GPS Neo 6M	14
Gambar 3.6 Rangkaian Skematik keseluruhan	15
Gambar 3.7 Tempat komponen.....	16
Gambar 4.1 Rangkaian <i>flexible sensor</i>	18
Gambar 4.2 Pemasangan alat <i>waterpass</i> dan busur	18
Gambar 4.3 Uji Coba <i>flexible sensor</i> 4 sudut 20 derajat.....	22
Gambar 4.4 Uji Coba <i>flexible sensor</i> 4 sudut 30 derajat.....	23
Gambar 4.5 Uji Coba <i>flexible sensor</i> 4 sudut 40 derajat.....	23
Gambar 4.6 Data GPS Neo 6M dan GPS <i>Handphone</i>	26
Gambar 4.7 Lama waktu pengiriman ke telegram	29

DAFTAR TABEL

Tabel 4.1 Konfigurasi Pin <i>flexible sensor</i>	18
Tabel 4.2 Hasil Pengujian <i>flexibel sensor</i> tanpa metode	19
Tabel 4.3 Hasil pengujian <i>flexibel sensor</i> menggunakan metode	21
Tabel 4.4 Hasil Pengujian GPS Neo 6M.....	25
Tabel 4.5 Hasil Pengujian lama waktu pengiriman ke telegram.....	28
Tabel 4.6 Hasil Pengujian sudut pitch depan dan belakang.....	30
Tabel 4.7 Hasil Pengujian sudut <i>pitch</i> depan dan belakang.....	30
Tabel 4.8 Hasil Pengujian sudut <i>roll</i> kanan dan kiri.....	31
Tabel 4.9 Hasil Pengujian keseluruhan sistem.....	34



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1 Tabel perbandingan penggunaan resistor	41
Lampiran 2 Flowchart <i>Flexible sensor</i> dengan <i>Exponential smoothing</i>	41
Lampiran 3 Flowchart sensor MPU6050	42
Lampiran 4 Flowchart GPS Neo 6M	43
Lampiran 5 Flowchar Telegram.....	44
Lampiran 6 Program <i>flexible sensor</i>	44
Lampiran 7 Pengambilan data <i>flexible sensor</i>	45
Lampiran 8 Program <i>flexible sensor</i> menggunakan <i>Exponential Smoothing</i>	45
Lampiran 9 Grafik Pengambilan data <i>flexible sensor</i> menggunakan <i>Exponential Smoothing</i>	47
Lampiran 10 Program GPS Neo 6M.....	52
Lampiran 11 Program sensor MPU6050	53
Lampiran 12 Program keseluruhan sistem.....	53
Lampiran 13 Pengambilan data keseluruhan lokasi kecelakaan	56
Lampiran 14 Tabel perbandingan nilai rata-rata permulusan pada Metode <i>Exponential Smoothing</i>	71
Lampiran 15 Hasil Turnitin.....	73

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Transportasi merupakan kendaraan yang sangat penting untuk menunjang pembangunan, ekonomi dan keberhasilan daerah. Pentingnya kiprah transportasi tercermin pada aplikasi dan fungsinya, tidak hanya terkait aspek kehidupan berbangsa dan bernegara, namun pula terhadap pelayanan transportasi bagi pergerakan orang dan barang di dalam negeri (Syawani, 2018). Kebutuhan Peran angkutan ditetapkan dalam Peraturan Menteri Nomor 15 Tahun 2019 (Menteri Perhubungan Republik Indonesia, 2019) tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Umum Pada Jalur. Selain kendaraan bermotor, pemerintah Indonesia juga mengatur pengangkutan kendaraan listrik dalam Peraturan Presiden Republik Indonesia Nomor 55 Tahun 2019 tentang percepatan program kendaraan listrik bertenaga baterai.

Dengan dukungan Peraturan Presiden Nomor 55 tersebut, penulis tergagas ikut serta dalam pengembangan kendaraan beroda empat bertenaga listrik yang dikembangkan oleh kampus. Mobil listrik ini didukung oleh motor listrik DC (*Direct Current*) dengan memakai tenaga listrik yang tersimpan pada baterai (*accu*). Keunggulan utama pada mobil listrik adalah tidak menimbulkan polusi udara, serta tidak membutuhkan bahan bakar fosil sebagai energi penggerak utamanya, juga mengurangi efek rumah kaca. Selain keunggulan pada bahan bakarnya, mobil listrik dapat dilengkapi dengan fitur yang menguntungkan pengguna. Keunggulan lainnya yaitu dapat meminimalkan kelalaian pengemudi (*Human Error*) yang berakibat fatal apabila terjadi kecelakaan. Kurangnya kecepatan penanganan saat terjadi kecelakaan juga dapat menimbulkan kesalahan fatal yang dapat mengakibatkan korban menjadi meninggal. Selain itu kecelakaan juga menimbulkan kemacetan apabila tidak segera ditangani. Hingga tahun 2017, laporan data kecelakaan mobil Surabaya tercatat 1.338. Jumlahnya didominasi oleh mobil pribadi seperti sepeda motor dan mobil. Jumlah tersangka kecelakaan lalu lintas terbanyak adalah luka

ringan yaitu 473 (70%), sedangkan terendah 83 (12%). Dari 1.338 kecelakaan mobil, 461 tewas dan 47 luka berat (Agustin et al., 2020).

Berdasarkan penelitian sebelumnya oleh (Asman et al., 2021) dengan judul Rancang Bangun Sistem Pendeteksi Kecelakaan Dan Tracking Lokasi Mobil Berbasis Android tersebut hanya menggunakan Sensor accelerometer MMA7361 untuk mengukur kecepatan linier dan mendeteksi getaran. Selain itu menggunakan GPS sebagai tracking kecelakaan kemudian dikirimkan melalui GSM/GPRS module atau via SMS. Kelemahan dari penelitian ini adalah hanya terdapat satu jenis sensor, sehingga data kecelakaan hanya didapatkan dari sensor tersebut. Pengiriman data GPS melalui GSM atau SMS juga mulai tertinggal karena fitur internet yang lebih mudah seperti WhatsApp, Line dan Telegram.

Berdasarkan uraian masalah dan hasil penelitian terdahulu yang tertulis diatas, maka penulis mengembangkan sistem deteksi kecelakaan yang diterapkan pada mobil listrik serta menerapkan metode *Exponential Smoothing* untuk mengurangi *noise* pada *flexible sensor*. Selain itu menggunakan sensor MPU 6050 untuk mengukur kemiringan kendaraan mobil listrik ketika terjadi kecelakaan. Alat tersebut mendeteksi bengkokkan ketika terjadi tabrakan dan mendeteksi kemiringan untuk mempermudah mobil listrik ketika mengalami kecelakaan serta dapat mengirimkan informasi berupa koordinat GPS melalui aplikasi telegram.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana kerja pengiriman data GPS (*Global Positioning System*) ke Chat Telegram Bot pada mobil listrik?
2. Bagaimana menerapkan *Exponential Smoothing* pada *Flexible Sensor* untuk mengurangi *noise* agar *output* sensor menjadi stabil?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Letak *Flexible Sensor* hanya terdapat di depan mobil.
2. Penggunaan sudut pada sensor MPU6050 hanya *Pitch* dan *Roll*.
3. Pengiriman data GPS (*Global Positioning System*) melalui koneksi internet.
4. Lokasi kecelakaan terdapat sinyal untuk mengirimkan data (*Global Positioning System*).
5. Pengujian kendaraan dilakukan pada jalan yang datar.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada Tugas Akhir ini sebagai berikut:

1. Mengirimkan data GPS (*Global Positioning System*) ke Chat Telegram Bot pada mobil listrik.
2. Menerapkan *Exponential Smoothing* pada *Flexible Sensor* untuk mengurangi *noise* agar *output* sensor menjadi stabil.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

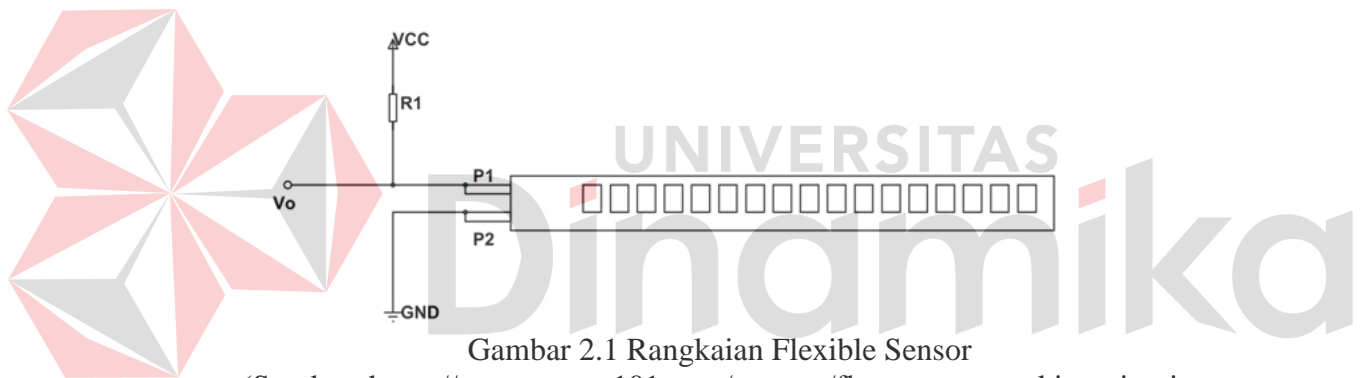
1. Dapat mengetahui lokasi kecelakaan melalui pesan yang berisi data GPS (*Global Positioning System*).
2. Menghindari keterlambatan penanganan korban kecelakaan, agar menghindari korban meninggal dunia.
3. Untuk menghindari kemacetan yang diakibatkan oleh keterlambatan penanganan baik dari korban maupun kendaraan kecelakaan.

BAB II

LANDASAN TEORI

2.1 Flexible Sensor

Fleksibel Sensor adalah sensor yang dapat mendeteksi pembengkokan akibat perubahan resistansi. Sensor beroperasi pada tegangan 5volt dengan daya puncak 1 watt. Rentang resistansi *fleksibel sensor* adalah 45K hingga 125K ohm, dan toleransi kesalahan sekitar 30% (Componen101, 2018). Resistansi memiliki nilai yang berbeda untuk setiap derajat dan bertambah seiring dengan pembengkokan yang terjadi. Selain itu, nilai keluaran sensor tergantung pada resistansi konstan yang digunakan. Berikut Gambar 2.1 rangkaian pada *Flexible Sensor* sehingga menghasilkan nilai keluaran.



Gambar 2.1 Rangkaian Flexible Sensor

(Sumber: <https://components101.com/sensors/flex-sensor-working-circuit-datasheet>)

Gambar 2.1 menunjukkan rangkaian pembagi tegangan dari *fleksibel sensor* dimana R1 sebagai resistansi konstan. Vo adalah tegangan titik tengah dan merupakan tegangan keluaran. Tegangan pada Vo berubah seiring dengan resistansi konstan R1. Rumus pembagi tegangan pada rangkaian *fleksibel sensor* adalah:

$$V_o = V_{cc} \left(\frac{RX}{R1 + RX} \right) \quad (2.1)$$

RX : Resistansi *Flexible Sensor*

R1 : Resistor konstan

Vo : Tegangan keluaran (*Volt Output*)



Gambar 2.2 Flexible Sensor

2.2 NodeMCU ESP32

NodeMCU ESP32 adalah salah satu keluarga *mikrokontroler* yang paling populer. Diperkenalkan dan dikembangkan oleh *Espressive Systems*. Sudah ada modul WiFi pada chip yang mendukung pembangunan sistem aplikasi untuk *Internet of Things* ESP32. NodeMCU ESP32 memiliki dual core 32-bit yang lebih unggul dari seri ESP8266 sebelumnya dan dilengkapi dengan *Bluetooth Low Energy (BLE)* (Ramadhana & Putra, 2020).

Berikut spesifikasi NodeMCU ESP32

- CPU: Xtensa dual core LX6 - 160MHz
- Arsitektur: 32bit
- Flash Memory: 16MB
- SRAM: 512kB
- GPIO Pin (ADC/DAC): 36 (18/2)
- Bluetooth dan WiFi: ada
- SPI/I2C/UART: 4/2/2



Gambar 2.3 NodeMCU ESP32

Dari *datasheet* di atas terlihat bahwa NodeMCU ESP32 memiliki kelebihan yang cukup jika sistem membutuhkan banyak pin analog. NodeMCU ESP32 juga dapat menggunakan *Bluetooth Low Energy* (BLE) ketika koneksi WiFi tidak memungkinkan. Tidak seperti *Bluetooth* yang selalu aktif, (BLE) selalu dalam mode tidur kecuali saat memulai koneksi. Akibatnya, NodeMCU ESP32 mengonsumsi daya yang sangat kecil. Sekitar 1/100 konsumsi daya *Bluetooth* (BLE) (tergantung pada kasus penggunaan).

2.3 Global Position System Neo 6M (GPS)

GPS yang berarti *Global Positioning System* adalah sistem navigasi yang menyediakan informasi dari perangkat yang terhubung dengan satelit. Perangkat ini disebut penerima GPS. Data informasi yang diperoleh berupa posisi lintang dan bujur. Informasi lintang dan bujur ini memungkinkan untuk menemukan objek dari satelit (Desnanjaya et al., 2021).



UNIVERSITAS
Dikomika

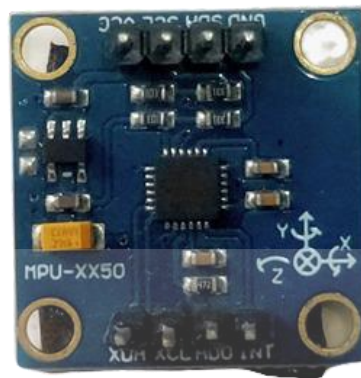


Gambar 2.4 GPS Neo 6M

Gambar 2.4 menunjukkan bentuk fisik modul GPS Neo 6M. Modul ini sesuai dengan APM2 dan APM2.5 dengan *EEPROM on board* (memori read-only yang dapat dihapus secara elektrik) dapat digunakan untuk menyimpan data konfigurasi. Antarmuka komunikasi serial TTL (*Transistor-Transistor Logic*) (RX / TX) dapat diakses dari *mikrokontroler* dengan fungsi UART (*Universal Asynchronous Receiver Transmitter*) atau emulasi TTL serial.

2.4 Sensor MPU 6050

Sensor MPU 6050 adalah modul sensor yang memiliki dua fungsi. dua fungsi tersebut adalah *accelerometer* yang dilengkapi dengan sistem mikro elektromekanis (MEMS) dan *Gyroscope* dalam sebuah otak *chip* (Suprayogi et al., 2019). Dengan 16-bit yang dilakukan konversi untuk menentukan sumbu agar sensor ini dapat beroperasi secara optimal. Nilai sumbu x, sumbu y, dan sumbu z dari sensor ini dapat diambil secara bersamaan.



Gambar 2.5 Sensor MPU 6050

2.5 Chat Telegram Bot

Telegram Bot API (*Application Programming Interface*) adalah teknologi *open source* yang dikembangkan dan disediakan oleh Telegram untuk membangun aplikasi Telegram Bot untuk pengembang. Telegram menyediakan kode sumber yang saat ini digunakan. Jenis API kedua adalah API Telegram Bot memungkinkan siapa saja untuk membuat bot yang merespon semua pengguna ketika mengirimkan pesan berupa perintah yang dapat diterima oleh bot tersebut (Cokrojoyo et al., 2017).

2.6 Exponential Smoothing

Metode *Exponential Smoothing* adalah metode prediksi yang umum digunakan untuk analisis *time series*, karena prosedur prediksi otomatisnya sederhana dan mudah (Maricar, 2019). Metode ini mengarah pada perbaikan dengan meredamkan data masa lalu dan mendapatkan prediksi tentang data masa

depan. Data yang diredam biasanya berisi data yang tidak stabil atau acak. *Exponential Smoothing* adalah metode prediksi yang paling banyak digunakan. Hal ini karena metode ini membutuhkan komputasi yang lebih sedikit. Prosedur metode ini adalah terus mengulang perhitungan dengan data terbaru. Metode *Exponential Smoothing* dapat dituliskan secara sistematis sebagai berikut (Firdaus & Sani, 2021).

$$F_t = \alpha X_t + (1 - \alpha)F_{t-1} \quad (2.2)$$

Keterangan:

F_t : nilai *output* filter sekarang

α : nilai konstanta permulusan

X_t : nilai aktual sensor

F_{t-1} : nilai *output* filter sebelumnya

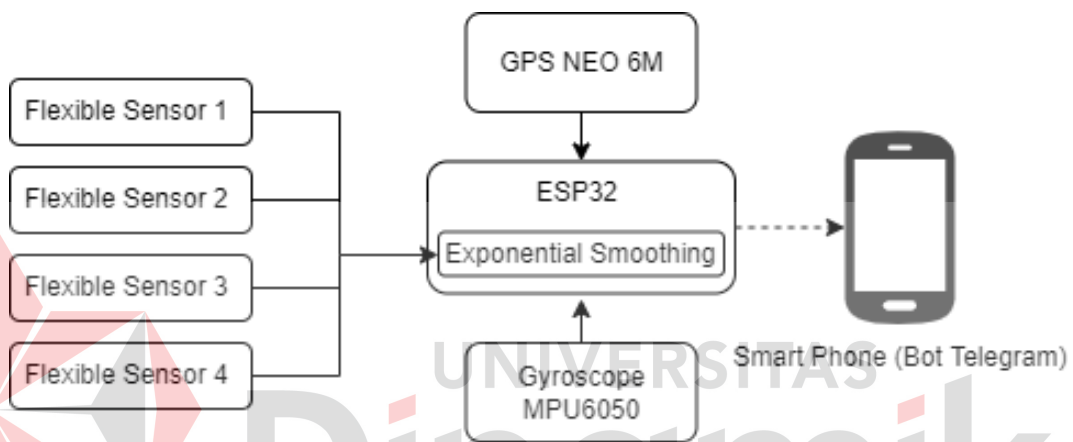
Bobot disimbolkan dengan α (alpha) yang dapat disebutkan dengan nilai permulusan. Nilai konstanta ini dapat dipilih antara nilai 0 dan 1, karena berlaku $0 < \alpha < 1$ (Ostertagová & Ostertag, 2011). Nilai α (alpha) menentukan berapa banyak data sinyal yang harus diperhitungkan. Nilai α (alpha) ditentukan berdasarkan pengguna. Pembobotan α (alpha) berpengaruh terhadap sinyal jika nilai α (alpha) rendah, maka menerima banyak data sinyal yang diperhitungkan dan semakin lambat respon perubahan sinyal. Sedangkan jika nilai α (alpha) tinggi, maka menerima sedikit data sinyal yang diperhitungkan dan semakin cepat respon perubahan sinyal (Firdaus & Sani, 2021). Penelitian tersebut menggunakan nilai konstantan α (alpha) 0.1, 0.2, dan 0.3 dengan menggunakan 2 sensor sebagai input. Hasil pada tabel yang dijabarkan menunjukkan nilai α (alpha) 0.3 mempunyai keberhasilan yang tinggi.

BAB III

METODOLOGI PENELITIAN

3.1 Blok Diagram

Diagram skema sistem dari deteksi kecelakaan mobil listrik dapat dilihat pada Gambar 3.1 Blok Diagram berikut. Diagram tersebut terdiri dari beberapa komponen *input*, proses dan *output* yang memiliki fungsi masing-masing.



Gambar 3.1 Blok Diagram

Pada Gambar Blok Diagram diatas, masing-masing komponen memiliki peran masing diantaranya ialah sebagai berikut:

1. *input*

- a. *Flexible Sensor*: Sensor ini berfungsi untuk mengambil data yang berasal dari bengkokkan yang diakibatkan tabrakan bodi depan mobil listrik.
- b. *Sensor MPU 6050*: Sensor ini berfungsi mengambil data sudut *pitch* dan *roll* ketika mobil mengalami kecelakaan terguling.
- c. *GPS Neo 6M*: komponen ini berfungsi mengambil data koordinat *latitude* dan *longitude* untuk mengetahui lokasi mobil.

2. *Proses*

- a. *NodeMCU ESP32*: Alat ini merupakan komponen utama yang berfungsi untuk mengontrol *input* dan *output*.
- b. *Exponential Smoothing*: Digunakan untuk meredam nilai *output* dari *Flexible Sensor* agar data yang diperoleh menjadi tidak *noise*. Dengan

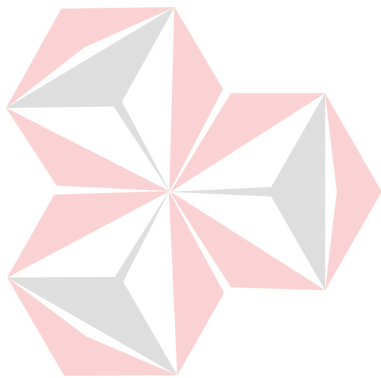
menggunakan α (alpha) yang ditentukan dengan cara *Trial and Error* agar mendapatkan nilai yang baik. Setiap *Flexible Sensor* tersebut dilakukan dengan metode *Exponential Smoothing* yang sama.

3. Output

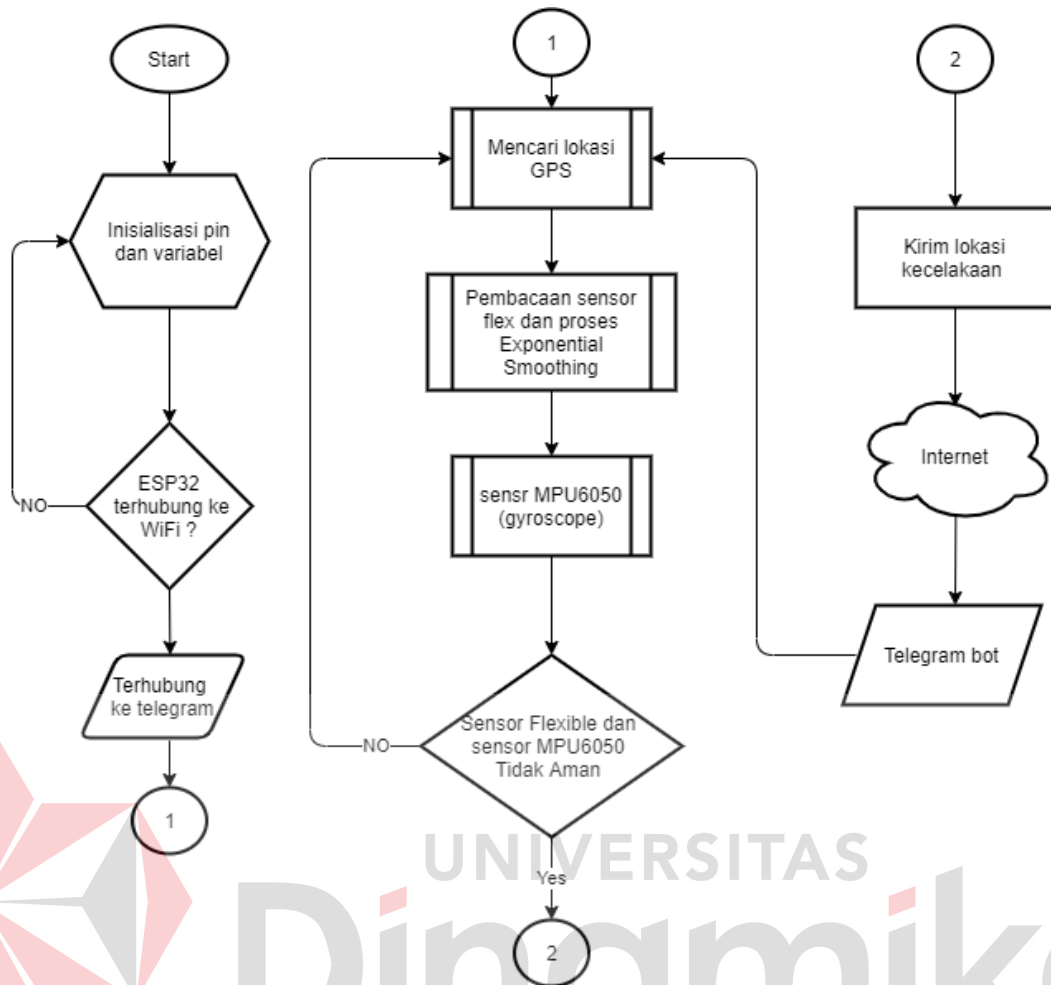
Telegram Bot: Merupakan aplikasi untuk menerima data GPS yang disebabkan oleh kecelakaan.

3.2 Flowchart Sistem

Gambar 3.6 adalah flowchart seluruh sistem dimana dikontrol oleh NodeMCU ESP32 sebagai komponen proses. Komponen *input* terdiri dari *Flexible sensor*, sensor MPU6050, dan GPS. Selain itu komponen *output* menggunakan Bot Telegram sebagai penerima pesan yang dikirimkan oleh NodeMCU ESP32.



UNIVERSITAS
Dinamika



Gambar 3.2 Flowchart Sistem

Algoritma diawali dengan inisialisasi masing-masing sensor dan variabel untuk menentukan port, SSID dan Password WiFi. Kemudian melakukan koneksi ke WiFi dengan pengecekan apakah NodeMCU ESP32 sudah terhubung ke WiFi jika belum, maka mengulang proses inisialisasi SSID dan *Password* WiFi. Berikutnya proses koneksi telegram dengan NodeMCU ESP32 dengan melakukan *test connection*. Jika sudah terkoneksi dengan telegram, proses selanjutnya GPS Neo 6M mencari lokasi *latitude* dan *longitude* mobil secara *real time* kemudian disimpan dalam variabel *string*. Pembacaan data dilakukan oleh *Flexible Sensor* dan sensor MPU 6050 untuk mengetahui nilai dari masing-masing sensor. Proses perhitungan pada *flexible sensor* menggunakan metode *Exponential Smoothing* agar dapat mengurangi *noise*. Sedangkan pada sensor MPU 6050 untuk mendeteksi kemiringan pada saat mobil terguling dengan sudut *roll* miring kanan dan miring kiri, sedangkan sudut *pitch* ke arah depan dan ke arah belakang. Apabila salah satu

sensor melebihi *threshold* yang sudah ditemukan sehingga kondisi tidak aman (Yes), maka data informasi lokasi pada GPS Neo 6M yang disimpan pada variabel *string* akan dikirimkan ke telegram dengan ID telegram yang sudah dibuat. Jika tidak melebihi *threshold* atau kondisi aman (No), maka kembali melakukan pembacaan GPS Neo 6M.

3.3 Perancangan Elektronika dan perangkat lunak

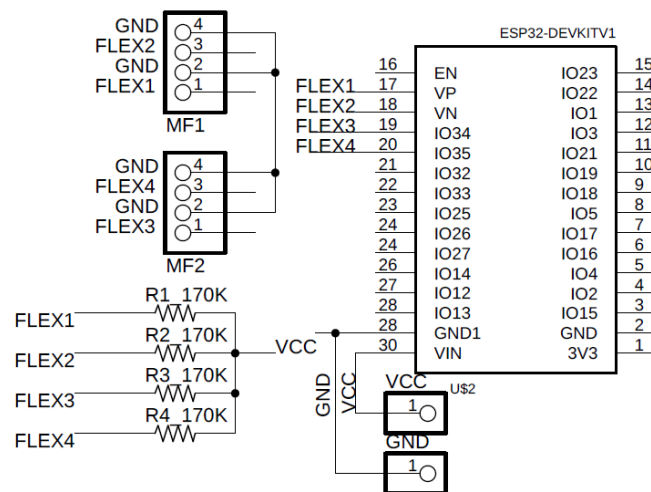
Bagian ini menjelaskan desain elektronik serta flowchart sistem yang dibuat dengan tujuan merancang sistem sebelum implementasi langsung.

3.3.1 Perancangan Flexible Sensor

Pada Gambar 3.2 Merupakan rangkaian skematik *flexible sensor* yang berfungsi sebagai deteksi bengkokkan sensor pada body depan mobil akibat kecelakaan. Rangkaian tersebut terdiri 4 buah *flexibel sensor* masing-masing menggunakan rangkaian pembagi tegangan dengan resistor 170K ohm. Masing-masing tegangan keluaran dari *flexible sensor* dihubungkan dengan NodeMCU ESP32 melalui pin analog. Pada penelitian ini menggunakan resistor 170K dikarenakan memiliki nilai yang mudah dibaca apabila mengalami perubahan bengkokkan 30 derajat. Tabel serta gambar perbandingan penggunaan resistor dapat dilihat pada lampiran 1. Nilai resistor 1K dan 10K seharusnya memiliki nilai yang besar, namun pada tabel tersebut nilai perubahan kecil dibandingkan dengan resistor 170K.

Algoritma pada *flexible sensor* dimulai dengan inisialisasi beberapa variabel untuk menyimpan data pembacaan masing-masing sensor. Selain itu dilakukan proses *Exponential Smoothing* pada setiap sensor untuk mendapatkan nilai yang lebih stabil. Adapun flowchart dapat dilihat pada lampiran 2. Penentuan *threshold* pada sensor ini dilakukan dengan cara membaca nilai ADC yang keluar dari sensor dengan berbagai sudut yang telah ditentukan diantaranya 20, 30, dan 40 derajat. Penggunaan sudut ini bertujuan untuk mengetahui seberapa besar nilai ADC

sehingga dapat memperoleh nilai yang cocok untuk disimpulkan sebagai kecelakaan.

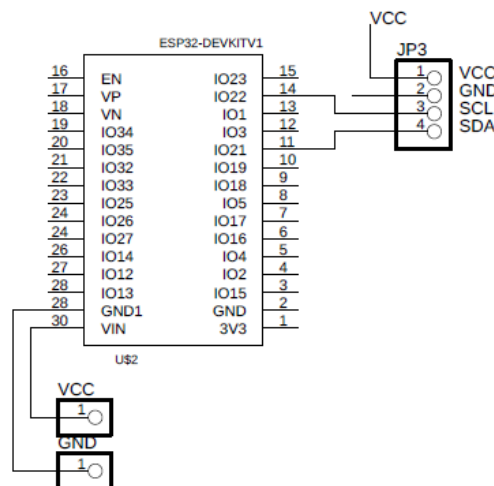


Gambar 3.3 Rangkaian Skematik *Flexible Sensor*

3.3.2 Perancangan sensor MPU 6050

Pada Gambar 3.3 adalah rangkaian skematik sensor MPU 6050 yang berfungsi mendeteksi kemiringan mobil ketika mengalami kecelakaan dengan mengandalkan sudut *pitch* dan *roll*. Sensor ini dihubungkan pada pin SDA SCL dengan NodeMCU ESP32. Skematik perancangan dari sensor MPU 6050 bisa dilihat pada gambar 3.3.

Algoritma sensor MPU 6050 diawali dengan inisialisasi variabel kemudian pembacaan sensor untuk mendapatkan nilai yang akan di konversikan ke sudut. Pada flowchart ini dapat dilihat pada lampiran 3. Pada sensor MPU6050 ini memiliki parameter derajat pengujian yaitu 330, 300, dan 270. Penentuan sudut ini untuk mengukur kemiringan kendaraan agar dapat disimpulkan sebagai kecelakaan.

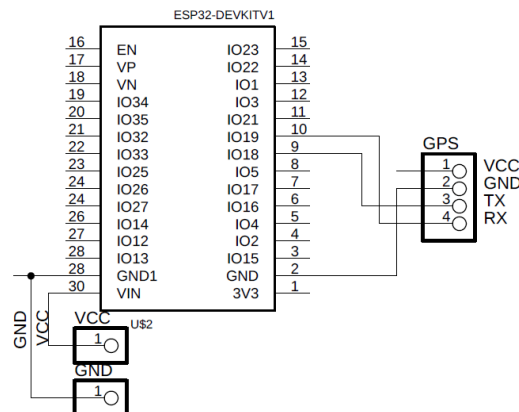


Gambar 3.4 Rangkaian Skematik sensor MPU 6050

3.3.3 Perancangan GPS Neo 6M

Pada Gambar 3.4, GPS digunakan untuk mendapatkan koordinat letak lokasi atau letak modul GPS. Modul GPS yang digunakan adalah GPS Shield DuinoPeak Neo 6M yang dapat mendeteksi garis lintang dan garis bujur. Modul GPS Neo 6M berkomunikasi dengan NodeMCU ESP32 melalui komunikasi serial menggunakan pin TX dan RX.

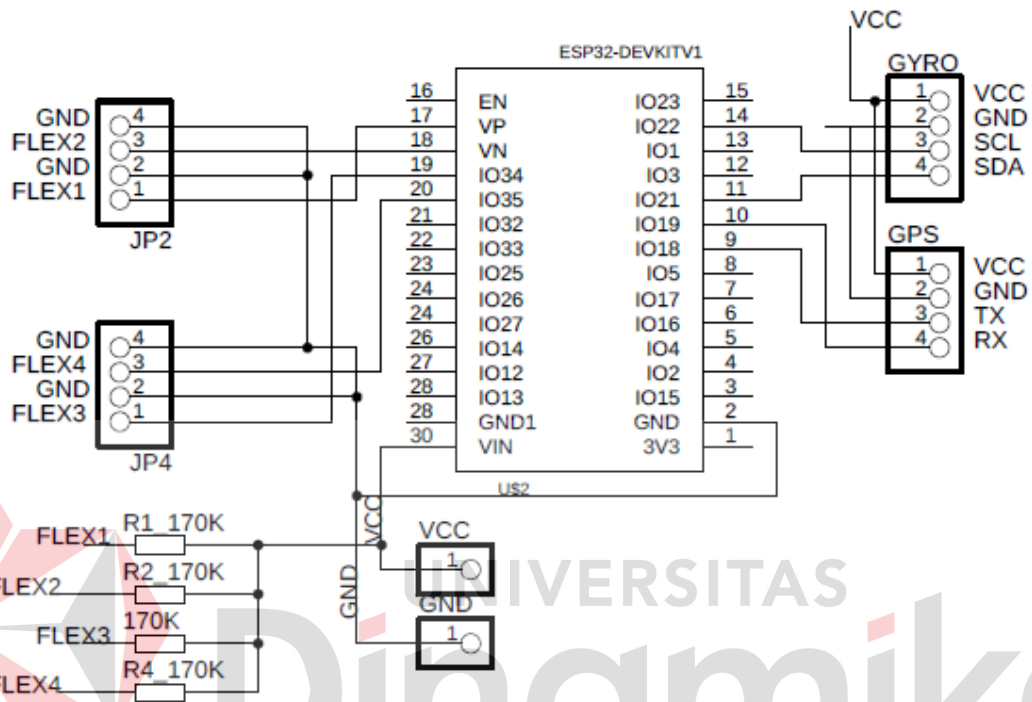
Alur algoritma pada GPS Neo 6M ini dimulai dengan melakukan inisialisasi variabel kemudian memeriksa port komunikasi serial agar GPS dapat dibaca oleh NodeMCU ESP32. Proses selanjutnya pembacaan data GPS yang telah terkirim yaitu data *latitude* dan *longitude* kemudian dilakukan konversi ke variabel string agar dapat dikirimkan ke Telegram Bot. Flowchar pada GPS Neo 6M dapat dilihat pada lampiran 4.



Gambar 3.5 Rangkaian Skematik GPS Neo 6M

3.3.4 Perancangan Keseluruhan Sistem

Pada Gambar 3.5 merupakan rangkaian skematik keseluruhan yang dibuat pada *software Eagle*. Skematik tersebut dicetak dalam *PCB* sebagai *shield* pendukung NodeMCU ESP32 dan sensor lainnya. Penggunaan pin juga dapat dilihat untuk mengetahui kebutuhan sistem.



Gambar 3.6 Rangkaian Skematik keseluruhan

3.4 Rancangan Mekanik

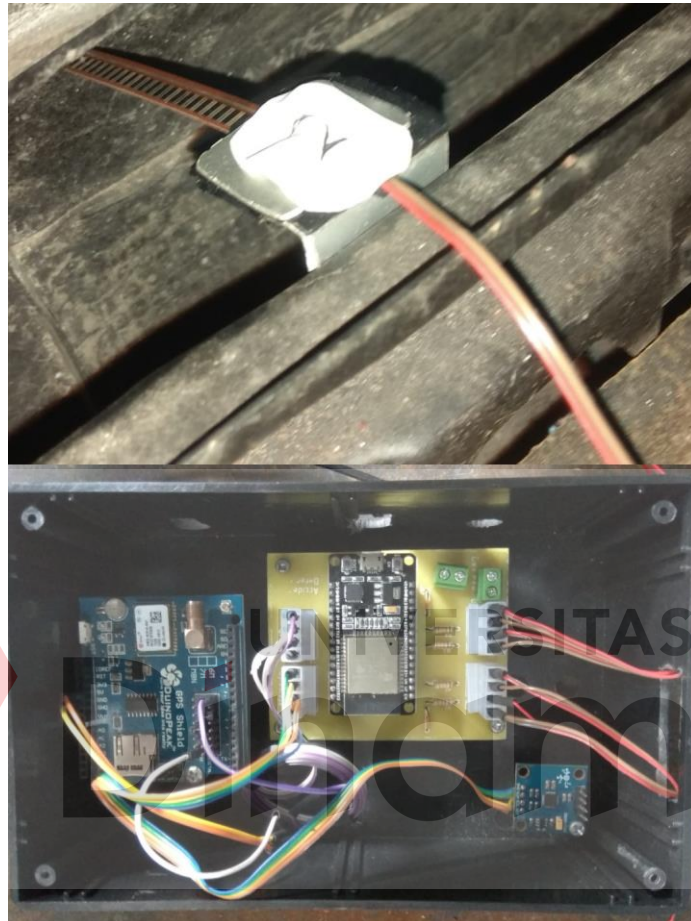
Bagian ini dijelaskan tentang rancangan pemasangan komponen yang terdiri dari *flexible sensor*, serta penempatan *mikrokontroler* sensor MPU6050, GPS Neo 6M agar dapat berjalan sesuai yang diinginkan. Rancangan pemasangan komponen dapat dilihat bawah ini:

a. Tempat *flexible sensor*

Pada gambar 3.7 merupakan tempat untuk pemasangan *flexible sensor* dengan dibuatkan *mounting*, kemudian dipasangkan ke lubang baut yang berada pada *chasis* depan mobil.

b. Tempat sensor MPU6050, kontroler dan GPS Neo 6M

Komponen tersebut dijadikan satu pada wadah *blackbox* untuk memudahkan ketika pemeliharaan komponen dan *blackbox* diletakkan di bawah *dashboard* mobil listrik.



Gambar 3.7 Tempat komponen

BAB IV

HASIL DAN PEMBAHASAN

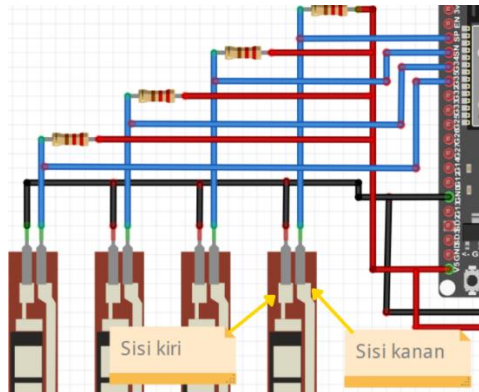
4.1 Hasil pengujian Flexible Sensor

Pengujian *Flexible Sensor* ini merupakan pengambilan data dengan hasil nilai ADC. Pengujian ini dilakukan pada 4 buah *flexible sensor* dimana masing-masing sensor ditabrakkan pada sebuah benda dengan derajat yang telah di tentukan. Menggunakan resistor yaitu 170k ohm pada setiap sensornya dengan rangkaian pembagi tegangan untuk menghasilkan nilai ADC yang mudah dibaca. Tabel 4.1 merupakan hasil pengujian *flexible sensor* tanpa metode. Berikut komponen yang dibutuhkan dalam proses pengujian, diantaranya:

- a. PC/Laptop
- b. Resistor 170k ohm
- c. *Flexible sensor* 4 buah
- d. NodeMCU ESP32
- e. Kabel USB Micro
- f. *Software* Arduino IDE
- g. *Waterpass*
- h. Busur dan mistar

Berikut adalah prosedur dalam pengujian *Flexible sensor*

- a. Hidupkan PC/Laptop.
- b. Buka *Software* Arduino IDE, dan masukkan *Source Code* ke dalam *sketch*.
- c. Hubungkan kabel USB Micro ke PC/Laptop.
- d. Pasang sensor dengan menggabungkan rangkaian pembagi tegangan seperti gambar 4.1 dan konfigurasi pin pada 4.1.

Gambar 4.1 Rangkaian *flexible sensor*Gambar 4.2 Pemasangan alat *waterpass* dan busurTabel 4.1 Konfigurasi Pin *flexible sensor*

Sensor ke	Flexible Sensor (kanan)	Flexible Sensor (kiri)	ESP32
1	Resistor	GND	GPIO36
2	Resistor	GND	GPIO39
3	Resistor	GND	GPIO34
4	Resistor	GND	GPIO35

- e. Kemudian upload program pada lampiran 6.
- f. Pasang *waterpass* pada busur seperti pada 4.2.
- g. Bengkokkan *flexibel sensor* dengan mistar pada sudut 20, 30, dan 40 derajat sesuai dengan busur. Setelah itu lihat pada Arduino IDE dan buka pada menu

tools kemudian serial plotter untuk melihat hasilnya. Dokumentasi pengujian dapat dilihat pada lampiran 7.

Tabel 4.2 Hasil Pengujian *flexibel sensor* tanpa metode

Data	Busur (derajat)	Pengujian Tanpa Metode (Kelengkungan)			
		Flex1	Flex2	Flex3	Flex4
1	20	554	757	882	940
2		989	1,104	897	951
3		561	823	866	1,021
4		1,102	875	837	1,012
5		592	1,013	1,150	1,110
6		1,115	1,115	837	1,079
7		612	714	1,214	1,180
8		1,138	1,196	907	1,088
9		623	715	1,125	1,134
10		1,182	1,116	1,150	1,142
1	30	590	897	860	896
2		623	827	829	897
3		695	1,241	1,638	942
4		1,472	813	810	1,311
5		699	1,285	1,491	1,017
6		1,373	880	799	1,604
7		697	1,551	1,285	1,198
8		1,674	815	816	1,747
9		711	1,880	1,479	1,238
10		1,605	1,088	923	1,556
1	40	1,424	802	885	869
2		1,095	944	858	918
3		1,210	1,479	842	880
4		1,274	1,187	1,190	1,363
5		1,444	1,838	790	1,040
6		1,600	1,427	1,215	2,614
7		1,382	1,197	1,006	1,425
8		1,640	944	1,351	2,983
9		1,744	2,052	1,109	1,479
10		1,378	991	1,776	2,941

Tabel diatas adalah simulasi tabrakan dengan benda di depan sehingga dapat menghasilkan nilai keluaran berupa ADC. Dapat dilihat bahwa nilai tersebut tidak konsisten pada setiap derajatnya sehingga dapat dikatakan banyak *noise* saat pengambilan data tanpa metode. Pengujian dilakukan menabrakkan sensor pada sebuah benda dengan sudut 20, 30, dan 40 derajat. Dari tabel diatas pada sudut 20 diambil dari 0-20 derajat sehingga memiliki nilai perubahan maksimal 1,182 ADC. Penggunaan nilai derajat ini dikarenakan apabila mobil melewati jalan berlubang,

maka sensor dapat bergetar antara sudut 0-20 derajat sehingga mengantisipasi adanya kesalahan dalam menyatakan kondisi kecelakaan. Pada saat sudut 30 derajat nilai maksimal diatas 1200 sehingga dapat dikatakan kecelakaan karena sudut diatas 30 derajat bodi kendaraan mengalami bengkokkan, oleh karena itu nilai *threshold* 1200 ADC.

4.2 Hasil pengujian *Flexible Sensor* menggunakan *Exponential Smoothing*

Pengujian *Flexible Sensor* ini merupakan pengambilan data dengan hasil nilai ADC. Pengujian ini dilakukan pada 4 buah *flexible sensor* dimana masing-masing sensor ditabrakkan pada sebuah benda dengan derajat yang telah di tentukan. Menggunakan resistor yang sama yaitu 170k ohm dengan rangkaian pembagi tegangan untuk menghasilkan nilai ADC yang mudah dibaca. Pengujian ini menggunakan metode *Exponential Smoothing* yang menentukan nilai dengan cara *trial error*. Tabel 4.3 merupakan hasil pengujian *flexible sensor* menggunakan metode dan berikut komponen yang dibutuhkan dalam proses pengujian, diantaranya:

- a. PC/Laptop
- b. Resistor 170k ohm
- c. *Flexible sensor* 4 buah
- d. NodeMCU ESP32
- e. Kabel USB Micro
- f. Software Arduino IDE
- g. *Waterpass*
- h. Busur dan mistar

Berikut adalah langkah-langkah dalam pengujian *Flexible sensor* menggunakan metode:

- a. Hidupkan PC/Laptop.
- b. Buka *Software Arduino IDE*, dan masukkan *Source Code* ke dalam *sketch*.
- c. Siapkan rumus dari metode *Exponential Smoothing* dan nilai permulasan (α) 0.3, 0.4, dan 0.7.
- d. Hubungkan kabel USB Micro ke PC/Laptop.

- e. Pasang sensor dengan menggabungkan rangkaian pembagi tegangan seperti gambar 4.1. dan konfigurasi pin pada tabel 4.1.
- f. Kemudian upload program pada lampiran 8.
- g. Pasang *waterpass* pada busur seperti pada 4.2.
- h. Bengkokkan *flexibel sensor* dengan mistar pada sudut 20, 30, dan 40 derajat sesuai dengan busur. Setelah itu lihat pada Arduino IDE dan buka pada menu tools kemudian *serial plotter* untuk melihat hasilnya. Dokumentasi pengujian dapat dilihat pada lampiran 7.

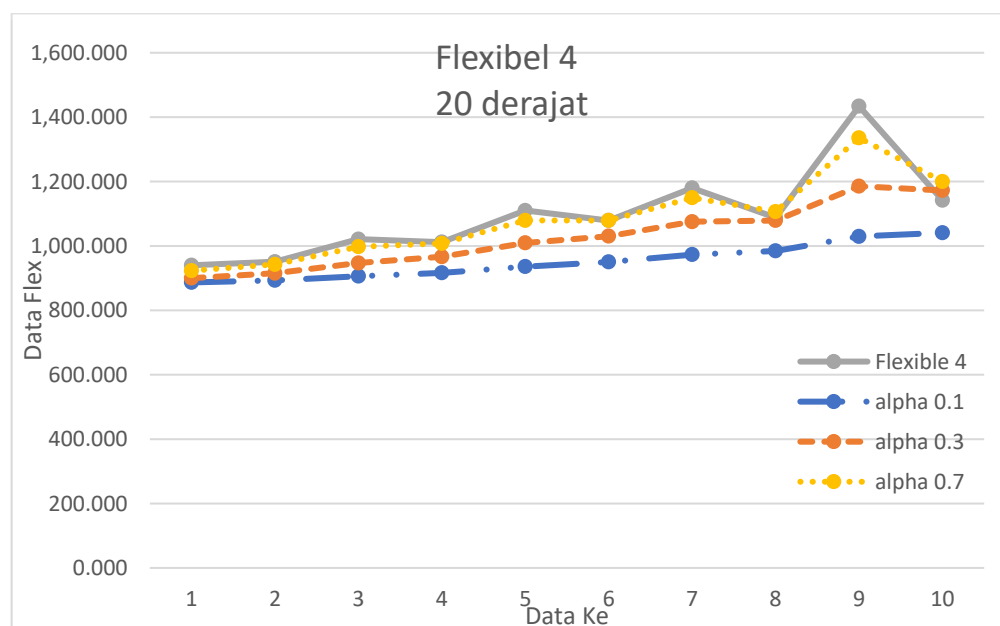
Tabel 4.3 Hasil pengujian *flexibel sensor* menggunakan metode

Data	Busur (derajat)	Pengujian Menggunakan Metode (Kelengkungan)			
		Flex1	Flex2	Flex3	Flex4
1	20	657.1310	837.5780	880.9240	900.0200
2		789.8790	917.5050	885.7470	915.3140
3		698.3270	889.1530	879.8230	947.0200
4		859.7960	884.9070	866.9760	966.5140
5		752.6780	923.3350	951.8830	1,009.5600
6		977.6070	1,010.8350	917.4180	1,030.3920
7		831.3640	921.7840	1,006.3930	1,075.2740
8		1,034.0180	1,034.0490	976.5750	1,079.0920
9		869.6110	938.3340	1,021.1020	1,185.5640
10		1,074.5670	1,051.6340	1,089.7720	1,172.4950
1	30	650.0090	786.9600	859.1940	894.7750
2		792.3980	798.9720	850.1360	895.4430
3		838.5430	931.5800	1,086.4950	909.4100
4		1,091.9260	896.0060	1,003.5470	1,029.8870
5		934.7560	1,012.7040	1,149.7830	1,026.0210
6		1,110.0530	972.8930	1,044.5480	1,199.4150
7		944.8320	1,146.3250	1,116.6830	1,198.9900
8		1,236.4990	1,046.9280	1,026.4780	1,363.3930
9		1,026.3000	1,296.8490	1,262.2350	1,325.7750
10		1,257.7800	1,234.1950	1,290.4640	1,394.8430
1	40	1,223.4720	759.2780	859.2270	887.3000
2		1,172.0830	814.6950	858.8590	896.5100
3		1,187.2500	1,013.9860	853.8010	891.5570
4		1,221.9500	1,065.8900	954.6610	1,032.9900
5		1,310.7700	1,297.5230	905.2630	1,035.0930
6		1,315.6370	1,336.3660	998.1840	1,508.7650
7		1,241.2290	1,294.5560	1,000.5290	1,483.6360
8		1,311.8830	1,189.3890	1,105.6700	1,933.4450
9		1,327.0780	1,448.1730	1,106.6690	1,797.1120
10		1,347.4470	1,311.0210	1,307.4680	2,140.2780

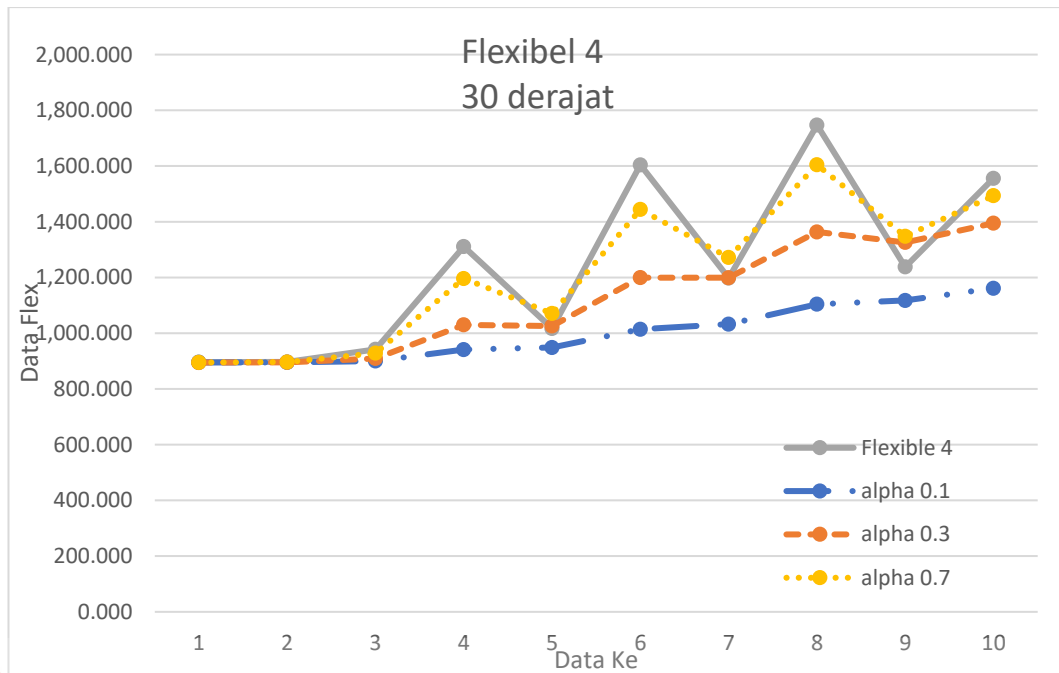
Pada tabel 4.3 tersebut dapat dilihat bahwa nilai data *ADC flexible sensor* mengalami peredaman *noise*. Meskipun peredaman tidak terlalu mulus, tetapi nilai tersebut mendekati nilai dari keluaran *flexible sensor*. Nilai alpha tersebut adalah 0.3 pada masing-masing *flexible sensor* dengan melakukan simulasi tabrakan ke benda dengan sudut 20, 30, 40 derajat. Dapat dilihat bahwa nilai tersebut konsisten pada 20 dan 30 derajat jika dibandingkan dengan pengambilan data tanpa metode. Nilai *flexible sensor* meningkat mengikuti bengkokkan pada setiap derajat pada sensor. Bengkokkan ini mencapai 1,234.195 jika dibulatkan yaitu 1,200 pada sudut 30 derajat dimana nilai tersebut dapat dijadikan *threshold* untuk menentukan nilai ketika terjadi tabrakan.

4.3 Hasil pengujian perbandingan nilai Alpha pada Metode *Exponential Smoothing*

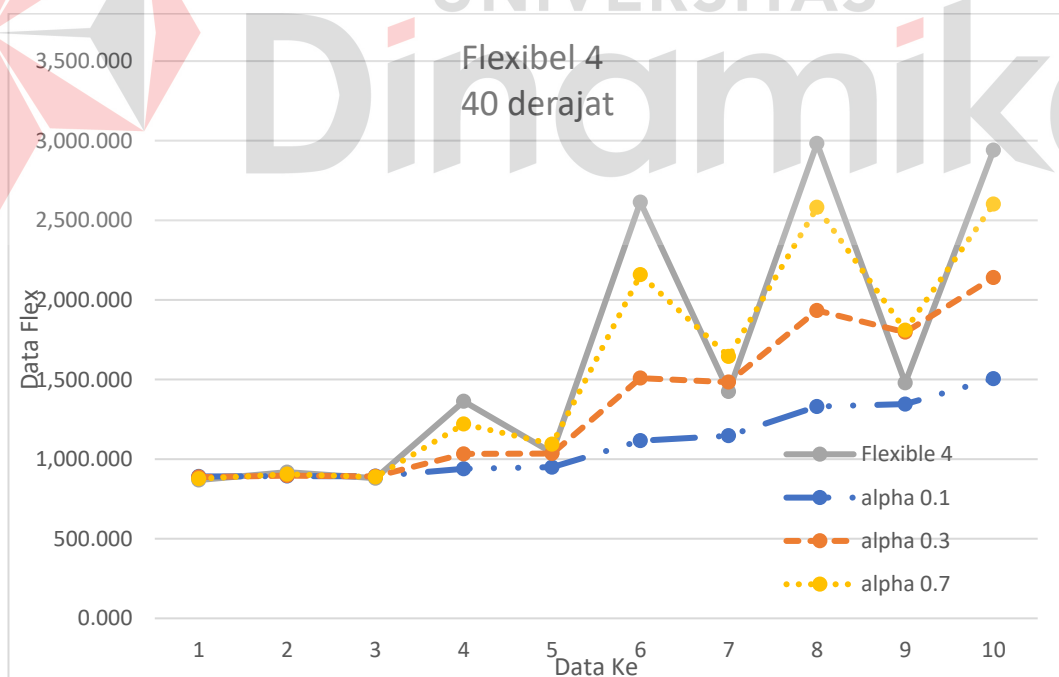
Pada pengujian ini bertujuan untuk mengetahui berapa nilai permulusan (alpha) yang cocok pada *flexible sensor*. Pengambilan data sensor ini dilakukan secara bersamaan dengan tiga permulusan yaitu 0.1, 0.3, dan 0.7. Masing-masing dilakukan pengujian dengan menabrakkan suatu benda ke sensor. Berikut merupakan pengambilan data *flexible sensor* 4 dapat dilihat pada grafik pengujian:



Gambar 4.3 Uji Coba *flexible sensor* 4 sudut 20 derajat



Gambar 4.4 Uji Coba *flexible sensor 4* sudut 30 derajat



Gambar 4.5 Uji Coba *flexible sensor 4* sudut 40 derajat

Pengujian ini dilakukan dengan nilai alpha yang dijalankan secara bersamaan dengan menggunakan metode dan tidak menggunakan metode. Hasil grafik menunjukkan nilai mengalami peningkatan pada setiap derajat, namun juga

mengalami penurunan seiring dengan bengkokkan yang terjadi. Peningkatan tersebut mencapai nilai 1200 pada bengkokkan 30 derajat. Pengujian tersebut dapat disimpulkan bahwa grafik yang menggunakan metode lebih teredam daripada tanpa metode dengan nilai *noise* berkurang 5.8755% pada alpha 0.7, 10.4853% pada alpha 0.3, dan 17.4258% pada alpha 0.1 dihitung dengan rumus sebagai berikut:

$$\text{Total \%} = \left(\frac{|\text{Data asli} - \text{data setelah metode}|}{\text{Data asli}} \right) * 100\% \quad (4.1)$$

Perhitungan diatas dapat dilihat pada tabel yang berada di lampiran 14. Penggunaan nilai alpha juga mempengaruhi tingkat *noise* dari sensor tersebut. Nilai permulusan tersebut juga berkurang sebesar 10.4853% pada alpha 0.3. Dari grafik maupun perhitungan diatas dapat disimpulkan pergerakan nilai alpha 0.3 mampu mengurangi *noise* tanpa menghilangkan data asli dengan nilai permulusan atau alpha 0.3. Selain itu gambar pengujian dapat dilihat pada lampiran 9.

4.4 Hasil pengujian GPS Neo 6M

Pengujian ini untuk membandingkan antara GPS *handphone* dan GPS sensor, sehingga dapat mengetahui jarak masing-masing. Berikut merupakan komponen yang diperlukan dalam melakukan pengujian GPS diantaranya:

- a. PC/Laptop
- b. GPS Neo 6M
- c. GPS *handphone*
- d. Garmin GPS
- e. Baterai power bank
- f. *Software* Arduino IDE
- g. Aplikasi telegram
- h. NodeMCU ESP32
- i. Kabel USB Micro
- j. *Flexible sensor*
- k. Hotspot internet

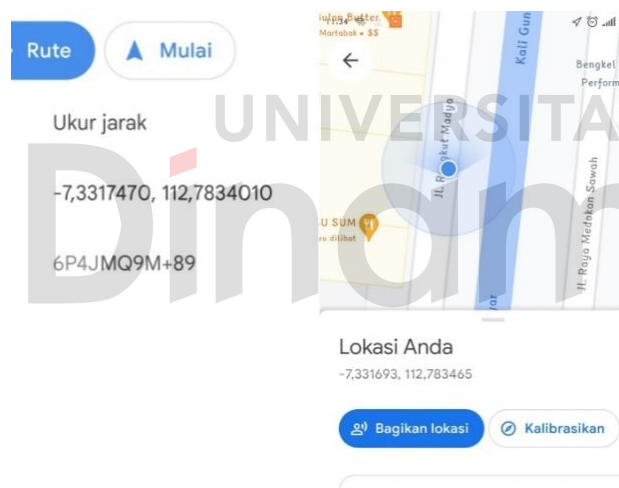
Berikut adalah langkah-langkah dalam Pengujian GPS Neo 6M:

- Hidupkan PC/Laptop, buka Arduino IDE dan Hotspot internet.
- Upload program di Arduino IDE dengan memakai kabel USB Micro pada NodeMCU ESP32, kemudian pasang kabel GPS Neo 6M TX ke GPIO18 dan RX ke GPIO19. Program dapat dilihat pada lampiran 10.
- Sambungkan koneksi VCC GND ke NodeMCU ESP32. Kemudian pasang antenna pada port yang telah tersedia.
- Pasangkan *flexible sensor* untuk menjadi *trigger* agar dapat mengirimkan GPS. Pemasangan rangkaian dapat dilihat pada 4.1.
- Pasangkan power bank sebagai sumber tegangan.
- Cari lokasi atau jalan untuk mengetahui apakah GPS berfungsi dengan baik. Agar lokasi dapat muncul, maka bengkokkan *Flexible sensor* kemudian lihat lokasi yang terkirim di telegram.
- Selain itu lihat juga pada GPS garmin dan *Handphone* untuk memastikan lokasi apakah sudah tepat.

Tabel 4.4 Hasil Pengujian GPS Neo 6M

No	Nama Lokasi	Data GPS			Selisih <i>handphone</i> sensor (KM)	Selisih Garmin sensor (KM)
		Garmin	Handphone	Sensor		
1	Rungkut madya v2	-7.33167, 112.78342	-7.331693, 112.783465	-7.3317470, 112.7834010	0.009267	0.008815
2	Ir Soekarno	-7.32946, 112.78116	-7.329483, 112.78119	-7.3294250, 112.7810440	0.01735	0.01337
3	Rungkut madya	-7.33153, 112.77936	-7.331536, 112.779345	-7.3317470, 112.7791210	0.03407	0.03573
4	Rungkut asri tengah	-7.32801, 112.77566	-7.32806, 112.775644	-7.3280580, 112.7756960	0.005739	0.006652
5	Rungkut harapan	-7.32275, 112.77823	-7.322704, 112.778313	-7.3227110, 112.7782750	0.004263	0.006591
6	Kertajaya	-7.27946, 112.76388	-7.279449, 112.763869	-7.2794550, 112.7638090	0.006651	0.007851
7	Pucang Anom timur	-7.28501, 112.75574	-7.285002, 112.755799	-7.2847800, 112.7559970	0.03296	0.03818
8	Ngagel jaya Selatan	-7.29340, 112.75378	-7.293303, 112.753728	-7.2932880, 112.7537540	0.003317	0.01278

9	Dinoyo	-7.28427, 112.7460 5	-7.284282, 112.74604 3	-7.2842660. 112.746025 0	0.002666	0.002793
10	Soetomo	-7.28311, 112.7359 1	-7.28313. 112.73592 9	-7.2832100. 112.735954 0	0.009313	0.01213
11	Ciliwung	-7.29371, 112.7350 7	-7.293744. 112.73505 8	-7.2937220. 112.735016 0	0.005239	0.006104
12	Diponegoro	-7.28780, 112.7348 6	-7.287799. 112.73485 2	-7.2876320. 112.735069 0	0.03029	0.02967
13	Adityawarm an	-7.29358, 112.7305 8	-7.293518. 112.73056 7	-7.2940280. 112.730354 0	0.06138	0.05570
14	Jl Nusa indah	-7.33206, 112.7918 9	-7.33054. 112.79187 9	-7.3320570. 112.791924 0	0.168800	0.003765
15	dukuh pakis	-7.28859, 112.7138 2	-7.288606. 112.71385 2	-7.2886970. 112.713966 0	0.01614	0.02002
Rata-rata					0.16880000	0.00881500



Gambar 4.6 Data GPS Neo 6M dan GPS *Handphone*

Dari pada gambar 4.6 merupakan salah satu sampel pengambilan data pada GPS *handphone* (kanan) dan GPS Neo 6M (kiri). Lokasi tersebut berupa data *longitude* dan *latitude* dan dibuka pada Google Maps. Pengambilan data pada GPS Neo 6M dapat dilihat pada telegram bot kemudian dibuka pada Google Maps, setelah itu data internet dimatikan kemudian dapat dilihat hasil dari data koordinatnya.

Tabel 4.4 menunjukkan bahwa setiap lokasi mempunyai jarak yang berbeda-beda antara GPS *handphone* dan GPS Neo 6M. Pengambilan data setiap titik memakan waktu sekitar 5-7 menit agar mendapatkan data yang maksimal. Dari

tabel tersebut dapat disimpulkan bahwa rata-rata dari handphone dan GPS yaitu sebesar 0.168800 Km atau 168,800 meter. Kemudian rata-rata dari Garmin dan GPS 0.008815 Km atau 8.8150 meter.

4.5 Hasil pengiriman Data GPS ke Telegram

Pada pengujian ini untuk mengetahui berapa waktu yang diperlukan pengiriman data GPS dari NodeMCU ESP32 ke aplikasi telegram bot. Pengambilan data ini diambil pada saat pengujian GPS Neo 6M menentukan titik lokasi. Berikut merupakan komponen yang diperlukan dalam melakukan pengujian GPS diantaranya:

- a. PC/Laptop
- b. GPS Neo 6M
- c. Stopwatch
- d. Resistor 170k ohm
- e. *Flexible sensor* 4 buah
- f. Kabel jumper
- g. Baterai power bank
- h. *Software* Arduino IDE
- i. Aplikasi telegram
- j. NodeMCU ESP32
- k. Kabel USB Micro
- l. Hotspot internet

Berikut adalah langkah-langkah dalam pengiriman data GPS ke Telegram:

- a. Hidupkan PC/Laptop, buka Arduino IDE dan Hotspot internet
- b. Upload program di Arduino IDE dengan memakai kabel USB Micro pada NodeMCU ESP32 kemudian pasang kabel GPS Neo 6M TX ke GPIO18 dan RX ke GPIO19. Program dapat dilihat pada lampiran 12.
- c. Setup GPS Neo 6M dengan memasang kabel TXRX ke masing-masing pin dan sambungkan koneksi VCC GND ke NodeMCU ESP32. Kemudian pasang antenna pada port yang telah tersedia.

- d. Pasangkan *Flexible sensor* untuk menjadi *trigger* agar dapat mengirimkan GPS. Pemasangan rangkaian dapat dilihat pada 4.1.
- e. Pasangkan power bank sebagai sumber tegangan.
- f. Setelah semua tersambung, lakukan pengujian dengan cara *trigger flexible sensor* secara bersamaan klik start pada stopwatch. Apabila data diterima dengan tanda notifikasi, maka klik stop pada stopwatch kemudian lihat hasil data stopwatch tersebut.

Tabel 4.5 Hasil Pengujian lama waktu pengiriman ke telegram

Data	Data GPS	Terkirim di telegram (detik)
1	-7,3317470, 112,7834010	4.5570
2	-7,3294250, 112,7810440	4.1790
3	-7,3317470, 112,7791210	4.3830
4	-7,3280580, 112,7756960	4.3220
5	-7,3227110, 112,7782750	4.0120
6	-7,2794550, 112,7638090	4.1200
7	-7,2847800, 112,7559970	4.0710
8	-7,2932880, 112,7537540	3.9620
9	-7,2842660, 112,7460250	3.7210
10	-7,2832100, 112,7359540	4.1970
11	-7,2937220, 112,7350160	4.0070
12	-7,2876320, 112,7350690	3.4630
13	-7,2940280, 112,7303540	4.5900
14	-7,3320570, 112,7919240	4.2240
15	-7.288697, 112.713966	4.4300
Rata Rata		4.1492



Gambar 4.7 Lama waktu pengiriman ke telegram

Tabel 4.5 adalah pengambilan data GPS yang dikirimkan dari NodeMCU ESP32 ke telegram dengan menggunakan aplikasi stopwatch. Pengambilan ini dilakukan dengan cara ketika melakukan *trigger flexible sensor* secara bersamaan klik tombol start pada stopwatch. Setelah muncul notifikasi pada *handphone*, maka klik tombol stop pada stopwatch kemudian lihat hasilnya. Pengujian ini dilakukan selama 15 kali data untuk melihat rata-rata *delay* pada saat pengiriman data.

4.6 Hasil pengujian Sensor MPU 6050

Pengujian sensor MPU 6050 berfungsi untuk mengetahui berapa besar *error* yang didapatkan pada sensor MPU 6050. Pengujian dilakukan dengan beberapa derajat kemudian dibandingkan dengan busur. Pengambilan sudut sensor MPU 6050 adalah sudut *pitch* (y) dan *roll* (x). Berikut komponen yang dibutuhkan dalam proses pengujian, diantaranya:

- PC/Laptop
- Sensor MPU 6050
- NodeMCU ESP32
- Kabel USB Micro

- e. *Software* Arduino IDE
- f. Busur
- g. Mistar
- h. Waterpass

Berikut adalah langkah-langkah dalam pengujian sensor MPU 6050.

- a. Hidupkan PC/Laptop.
- b. Buka *Software* Arduino IDE dan masukkan *Source Code* ke dalam *sketch*. Upload program di Arduino IDE dengan memakai kabel USB Micro. Program dapat dilihat lampiran 11.
- c. Hubungkan kabel USB Micro ke PC/Laptop.
- d. Pasang sensor dengan menyambungkan pin SDA SCL ke NodeMCU ESP32.
- e. Pasang *waterpass* pada busur seperti pada 4.2.
- f. Kemudian lakukan dengan cara memiringkan Sensor MPU6050 ke kanan dan ke kiri serta ke depan dan ke belakang sesuai dengan sudut yang telah ditentukan dapat dilihat pada tabel berikut:

Tabel 4.6 Hasil Pengujian sudut pitch depan dan belakang

Miring kanan (derajat)	Miring kanan (derajat)	ke depan (derajat)	Ke belakang (derajat)
30	330	30	330
60	300	60	300
90	270	90	270

Tabel 4.7 Hasil Pengujian sudut *pitch* depan dan belakang

Pengujian sudut <i>pitch</i> depan					Pengujian sudut <i>pitch</i> belakang			
No	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)	Busur	Hasil Pengujian (°)	Selisih (°)	Error (%)
1	30	29.3080	0.6920	2.3067	330	329.5200	0.4800	0.1455
2		29.6540	0.3460	1.1533		329.2080	0.7920	0.2400
3		29.0390	0.9610	3.2033		328.8810	1.1190	0.3391
4		29.0770	0.9230	3.0767		329.0990	0.9010	0.2730
5		28.8090	1.1910	3.9700		329.5960	0.4040	0.1224
6		29.3030	0.6970	2.3233		328.9800	1.0200	0.3091
7		29.0910	0.9090	3.0300		329.4080	0.5920	0.1794
8		29.7330	0.2670	0.8900		329.6380	0.3620	0.1097
9		28.7040	1.2960	4.3200		328.8470	1.1530	0.3494
10		28.6330	1.3670	4.5567		329.2950	0.7050	0.2136

Pengujian sudut <i>pitch</i> depan					Pengujian sudut <i>pitch</i> belakang			
No	Busur	Hasil Pengujian ()	Selisih ()	Error (%)	Busur	Hasil Pengujian ()	Selisih ()	Error (%)
		Rata-rata	0.8649	2.8830		Rata-rata	0.7528	0.2281
11	60	59.7710	0.2290	0.3817	300	298.1810	1.8190	0.6063
12		59.9130	0.0870	0.1450		298.8890	1.1110	0.3703
13		59.4170	0.5830	0.9717		298.9110	1.0890	0.3630
14		59.8970	0.1030	0.1717		299.4340	0.5660	0.1887
15		59.7370	0.2630	0.4383		298.7390	1.2610	0.4203
16		59.6800	0.3200	0.5333		298.9700	1.0300	0.3433
17		59.9620	0.0380	0.0633		298.3470	1.6530	0.5510
18		59.8170	0.1830	0.3050		297.9530	2.0470	0.6823
19		59.3140	0.6860	1.1433		298.8710	1.1290	0.3763
20		59.9720	0.0280	0.0467		298.6830	1.3170	0.4390
		Rata-rata	0.2520	0.4200		Rata-rata	1.3022	0.4341
21	90	89.6810	0.3190	0.3544	270	269.2720	0.7280	0.2696
22		89.5310	0.4690	0.5211		269.3630	0.6370	0.2359
23		89.4320	0.5680	0.6311		268.9880	1.0120	0.3748
24		89.7510	0.2490	0.2767		269.3110	0.6890	0.2552
25		89.4940	0.5060	0.5622		268.9010	1.0990	0.4070
26		89.7110	0.2890	0.3211		268.9780	1.0220	0.3785
27		89.2580	0.7420	0.8244		268.9070	1.0930	0.4048
28		89.8050	0.1950	0.2167		268.7450	1.2550	0.4648
29		89.1430	0.8570	0.9522		268.1950	1.8050	0.6685
30		89.2250	0.7750	0.8611		269.0500	0.9500	0.3519
		Rata-rata	0.4969	0.5521		Rata-rata	1.0290	0.3811
Rata-rata error keseluruhan								0.8164

Pengujian sudut *roll* (x) atau kemiringan ke kanan dan kiri dengan beberapa sudut dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Hasil Pengujian sudut *roll* kanan dan kiri

Pengujian sudut <i>Roll</i> kanan					Pengujian sudut <i>Roll</i> kiri			
No	Busur	Hasil Pengujian ()	Selisi h ()	Error (%)	Busur	Hasil Pengujian ()	Selisi h ()	Error (%)
1	30	28.2470	1.7530	5.8433	330	326.5130	3.4870	1.0567
2		28.6520	1.3480	4.4933		326.1650	3.8350	1.1621
3		28.5620	1.4380	4.7933		326.1660	3.8340	1.1618
4		28.0990	1.9010	6.3367		326.4910	3.5090	1.0633

Pengujian sudut <i>Roll</i> kanan					Pengujian sudut <i>Roll</i> kiri				
N o	Busu r	Hasil Pengujia n ()	Selisi h ()	Error (%)	Busu r	Hasil Pengujia n ()	Selisi h ()	Error (%)	
5	60	28.6260	1.3740	4.5800	300	326.2650	3.7350	1.1318	
6		28.5570	1.4430	4.8100		326.2620	3.7380	1.1327	
7		28.3090	1.6910	5.6367		326.0550	3.9450	1.1955	
8		28.6690	1.3310	4.4367		326.4640	3.5360	1.0715	
9		27.9590	2.0410	6.8033		326.3520	3.6480	1.1055	
10		28.6480	1.3520	4.5067		326.9700	3.0300	0.9182	
Rata-rata		1.5672	5.2240	Rata-rata		3.6297	1.0999		
11		58.4150	1.5850	2.6417		298.0390	1.9610	0.6537	
12		58.2940	1.7060	2.8433		297.6080	2.3920	0.7973	
13		58.9390	1.0610	1.7683		298.0180	1.9820	0.6607	
14	59.0890	0.9110	1.5183	297.4940	2.5060	0.8353			
15	58.9950	1.0050	1.6750	297.9310	2.0690	0.6897			
16	59.6790	0.3210	0.5350	297.6770	2.3230	0.7743			
17	59.1350	0.8650	1.4417	297.4030	2.5970	0.8657			
18	58.5850	1.4150	2.3583	297.6630	2.3370	0.7790			
19	58.5930	1.4070	2.3450	298.3110	1.6890	0.5630			
20	59.8200	0.1800	0.3000	297.5260	2.4740	0.8247			
Rata-rata		1.0456	1.7427	Rata-rata		2.2330	0.7443		
21	90	89.5350	0.4650	0.5167	270	268.5920	1.4080	0.5215	
22		89.7760	0.2240	0.2489		268.5940	1.4060	0.5207	
23		89.5490	0.4510	0.5011		268.8330	1.1670	0.4322	
24		89.3470	0.6530	0.7256		268.5120	1.4880	0.5511	
25		89.5770	0.4230	0.4700		268.9250	1.0750	0.3981	
26		89.9760	0.0240	0.0267		268.8860	1.1140	0.4126	
27		89.2130	0.7870	0.8744		269.5110	0.4890	0.1811	
28		89.6840	0.3160	0.3511		268.9980	1.0020	0.3711	
29		89.7070	0.2930	0.3256		268.6720	1.3280	0.4919	
30		89.5900	0.4100	0.4556		269.0370	0.9630	0.3567	
Rata-rata		0.4046	0.4496	Rata-rata		1.1440	0.4237		
Rata-rata error keseluruhan								1.6140	

Pengujian sensor MPU 6050 pada sudut *pitch* (y) ke depan dan belakang pada sudut yang telah ditentukan menggunakan busur. Pada sudut *pitch* (y) ke depan dapat dilihat nilai rata-rata *error* pada masing-masing sudut yaitu 2.8830%, 0.4200%, dan 0.5521%. Kemudian sudut *pitch* (y) ke belakang mempunyai nilai rata-rata *error* sebesar 0.2281%, 0.4341%, 0.3811% dan total *error* keseluruhan

sudut *pitch* (y) adalah 0.8164 %. Hasil tersebut menunjukkan nilai pada sudut *pitch* (y) memiliki *error* yang kecil. Pengujian selanjutnya yaitu pada sudut *roll* (x) ke kanan menunjukkan nilai rata-rata *error* 5.2240%, 1.7427%, dan 0.4496%. Sudut *roll* (x) ke kiri nilai rata-rata *error* sebesar 1.0999%, 0.7443%, 0.4237%. Rata-rata *error* keseluruhan pada sudut *roll* (x) adalah 1.6140%. Dari pengujian sensor MPU6050 diatas, maka ditentukan *threshold* pada setiap sudut *pitch* dan *roll*. Sudut *pitch* saat ke depan yaitu 90 derajat dan ke belakang sebesar 270 derajat. Kemudian pada sudut *roll* saat miring ke kanan yaitu 90 derajat, sedangkan ke kiri sebesar 270 derajat. Penentuan *threshold* ini berdasarkan informasi yang penulis dapatkan apabila mobil terperosok ke dalam got air atau selokan. Selain itu diakibatkan dari berbagai jenis kecelakaan seperti mobil menabrak pembatas jalan atau mobil berjalan *zig-zag* dalam kecepatan tinggi (Muhammad Fathan; Azwar, 2022).

4.7 Hasil pengujian Keseluruhan Sistem

Pengujian kecelakaan mobil ini dilakukan pada sepeda motor, dikarenakan mobil listrik belum bisa digunakan di jalan umum karena mekanisme pengereman dan kemudi yang kurang memadai. Pengujian pada sepeda motor dilakukan pada setiap lokasi jalan raya umum yang berbeda-beda. Pengujian ini dilakukan pada kedua sensor agar mendapatkan lokasi kecelakaan. *Flexible sensor* dibengkokkan sebagai simulasi kecelakaan untuk mendapatkan lokasi GPS. Kemudian sensor MPU6050 dimiringkan ke kanan dan kiri, serta diarahkan ke depan dan belakang agar mendapatkan titik lokasi kecelakaan. Berikut komponen yang dibutuhkan dalam pengujian keseluruhan sistem diantaranya:

- a. GPS Neo 6M
- b. Sensor MPU 6050
- c. *Handphone*
- d. Garmin GPS
- e. Stopwatch
- f. Resistor 170k ohm
- g. Flexible sensor 4 buah
- h. Kabel jumper

- i. Baterai power bank
- j. Aplikasi telegram
- k. NodeMCU ESP32
- l. Kabel USB Micro
- m. Hotspot internet

Berikut adalah langkah-langkah dalam melakukan Pengujian Keseluruhan Sistem:

- a. Hidupkan PC/Laptop dan buka Arduino IDE.
- b. Upload program di Arduino IDE dengan memakai kabel USB Micro pada NodeMCU ESP32 kemudian pasang pin molex sesuai dengan skematik. Program dapat dilihat pada lampiran 12.
- c. Pasangkan power bank sebagai sumber tegangan.
- d. Siapkan hotspot sebagai sumber internet agar tersambung ke NodeMCU ESP32.
- e. Simulasikan kecelakaan menggunakan sepeda motor dengan cara menabrakkan *flexible sensor* 1 dan 2, 3 dan 4, dan semua *flexible sensor* dengan sudut 30 derajat agar dapat mengirimkan data GPS ke telegram.
- f. Simulasikan kecelakaan menggunakan sepeda motor dengan posisi sensor dimiringkan ke kanan dan kiri. Kemudian posisi jatuh ke depan dan belakang.
- g. Cari lokasi jalan raya untuk mengetahui apakah GPS berfungsi dengan baik.
- h. Gunakan stopwatch untuk melihat lama waktu pengiriman telegram dengan cara *trigger flexible sensor* bersamaan dengan klik tombol start stopwatch kemudian lihat layar stopwatch klik stop apabila terdapat notifikasi pada telegram berupa koordinat GPS. Gambar pengujian dapat dilihat pada lampiran 13.

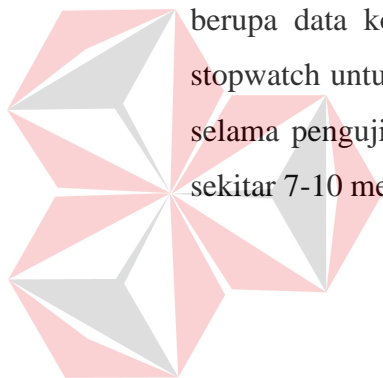
Tabel 4.9 Hasil Pengujian keseluruhan sistem

No	Nama Lokasi	Sensor GPS Neo6M	Status Kecelakaan	Status Keberhasilan	Terkirim di telegram (detik)
1	Jl. Rungkut Madya 149-167, Rungkut Kidul, Kec. Rungkut, Kota SBY, Jawa Timur 60293	-7.3317470. 112.7834010	Kecelakaan	Berhasil	4.5570

No	Nama Lokasi	Sensor GPS Neo6M	Status Kecelakaan	Status Keberhasilan	Terkirim di telegram (detik)
2	Jl. Dr. Ir. H. Soekarno, No. J51, Rungkut Kidul, Rungkut, Surabaya City, East Java 60293	-7.3294250. 112.7810440	Kecelakaan	Berhasil	4.1790
3	Rungkut Asri Tengah I No.2, Rungkut Kidul, Kec. Rungkut, Kota SBY, Jawa Timur 60293	-7.3280580. 112.7756960	Kecelakaan	Berhasil	4.3220
4	Jl. Manyar Kertoarjo 31, Manyar Sabrangan, Kec. Mulyorejo, Kota SBY, Jawa Timur 60116	-7.2794550. 112.7638090	Kecelakaan	Berhasil	4.1200
5	Jl. Pucang Anom Tim., Kertajaya, Kec. Gubeng, Kota SBY, Jawa Timur 60282	-7.2847800. 112.7559970	Kecelakaan	Berhasil	4.0710
6	PQ43+MFH, Jl. Krukah Ps., Ngagelrejo, Kec. Wonokromo, Kota SBY, Jawa Timur 60245	-7.2932880. 112.7537540	Kecelakaan	Berhasil	3.9620
7	Jl. Dr. Soetomo 94-84, DR. Soetomo, Kec. Tegalsari, Kota SBY, Jawa Timur 60264	-7.2832100. 112.7359540	Kecelakaan	Berhasil	4.1970
8	Jl. Diponegoro 64- 56, DR. Soetomo, Kec. Tegalsari, Kota SBY, Jawa Timur 60241	-7.2876320. 112.7350690	Kecelakaan	Berhasil	3.4630
9	Surabaya Town Square Level 1 No 75, Jl. Adityawarman No.55, Sawunggaling, Wonokromo, Surabaya City, East Java 60242	-7.2940280. 112.7303540	Kecelakaan	Berhasil	4.1970
10	Jl. Raya Dukuh Kupang, Pakis, Kec. Sawahan,	-7.2886970. 112.7139660	Kecelakaan	Berhasil	4.4300

No	Nama Lokasi	Sensor GPS Neo6M	Status Kecelakaan	Status Keberhasilan	Terkirim di telegram (detik)
	Kota SBY, Jawa Timur				
Rata-rata					4.1498

Dari tabel pengujian diatas merupakan seluruh rangkaian pengujian kecelakaan. Setiap sensor sudah dimasukkan *threshold* masing-masing, diantaranya 4 *flexible sensor* adalah 1200 ADC, dan sensor MPU6050 masing-masing sudut diatas 90 dan dibawah 270 derajat. *Flexible sensor* menggunakan metode permulasan (alpha) 0.3 agar *noise* teredam dan tidak menghilangkan nilai aslinya. Pada saat pengujian, beberapa sensor diuji dengan simulasi tabrakan kecelakaan kemudian pada saat bersamaan klik tombol start pada stopwatch. Saat notifikasi berupa data koordinat muncul pada *handphone*, maka klik tombol stop pada stopwatch untuk melihat berapa lama pengiriman data koordinat ke telegram bot. selama pengujian untuk mengambil data satu titik koordinat memerlukan waktu sekitar 7-10 menit agar koordinat GPS sesuai dengan lokasinya.



UNIVERSITAS
Dinamika

BAB V

PENUTUPAN

Pada bab ini, penulis memaparkan kesimpulan serta saran dari hasil pengujian yang telah dilakukan dalam penelitian tugas akhir ini. Sehingga hasil kesimpulan dan saran dapat dikembangkan pada penelitian selanjutnya.

5.1 Kesimpulan

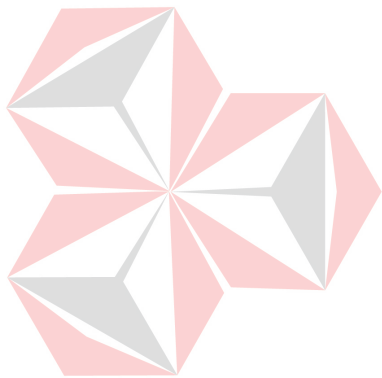
Pada penelitian ini, terdapat beberapa tahapan dimulai dari perancangan, hasil dan pembahasan. Dari tahapan tersebut tentang penelitian Rancang Bangun Deteksi Kecelakaan Pada Mobil Listrik Dengan *Exponential Smoothing* dapat disimpulkan sebagai berikut:

1. Berdasarkan pengujian *flexibel sensor* menggunakan metode *Exponential Smoothing* yang dapat dilihat pada grafik, mampu mengurangi *noise* tanpa menghilangkan data asli dengan nilai permulusan atau *alpha* 0.3, dan nilai berkurang sebesar 10.4853% dibandingkan dengan data asli.
2. Sensor MPU6050 memiliki nilai *error* sebesar 0.8164% pada sudut *pitch* dan 1.6140% pada sudut *roll* dibandingkan dengan busur.
3. GPS Neo 6M mampu mendeteksi lokasi dibandingkan dengan *handphone* dengan nilai selisih 168.8000 meter, serta dibandingkan dengan Garmin GPS nilai selisih sebesar 8.8150 meter.
4. Uji coba pengiriman data GPS ke Telegram Bot menghasilkan nilai rata-rata pengiriman sebesar 4.14900 detik.

5.2 Saran

Pada perancangan hasil dan pembahasan yang telah dilakukan oleh penulis, maka terdapat poin-poin yang memungkinkan melakukan pengembangan untuk lebih meningkatkan hasil yang baik, diantaranya:

1. Pada tahap pengujian sebaiknya dilakukan pada mobil listrik untuk mendapatkan nilai yang aktual.
2. Apabila diterapkan pada mobil listrik sebaiknya memakai *accu* mobil sebagai sumber tegangan. Dikarenakan saat pada penelitian ini belum ada *stepdown* yang dapat menurunkan tegangan sampai 5 volt 2 ampere

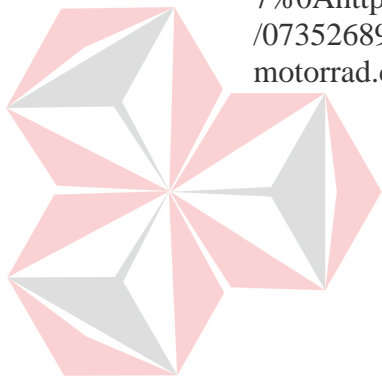


UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Agustin, I. W., Meidiana, C., & Muljaningsih, S. 2020. Studi Simulasi Model Kecelakaan Pengendara Mobil untuk Meningkatkan Keselamatan Lalu Lintas di Daerah Perkotaan. *Warta Penelitian Perhubungan* 32(2), 93–102. <https://doi.org/10.25104/warlit.v32i2.1513>
- Asman, S., Werman Kasoep, & Nefy Puteri Novani. 2021. Rancang Bangun Sistem Pendeteksi Kecelakaan Dan Tracking Lokasi Mobil Berbasis Android. *Chipset* 2(02), 7–14. <https://doi.org/10.25077/chipset.2.02.7-14.2021>
- Cokrojoyo, A., Andjarwirawan, J., & Noertjahyana, A. 2017. Pembuatan Bot Telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP. *Jurnal Infra*, 5(1), 224–227, Program Studi Teknik Informatika Fakultas. <http://studentjournal.petra.ac.id/index.php/teknik-informatika/article/view/5163>
- Componen101. 2018. *Fitur Sensor Fleksibel, Bekerja, Sirkuit & Lembar Data*. <https://components101.com/sensors/flex-sensor-working-circuit-datasheet>, diakses tanggal 31 Mei 2022
- Desnanjaya, I. G. M. N., Nugraha, I. M. A., & Hadi, S. 2021. Sistem Pendeteksi Keberadaan Nelayan Menggunakan GPS Berbasis Arduino. *Jurnal Sumberdaya Akuatik Indopasifik*, 5(2), 157–168. <https://ejournalfpikunipa.ac.id/index.php/JSAT/article/view/143>
- Firdaus, F., & Sani, A. 2021. Stasiun Pemantau Cuaca Berbasis IoT (Internet of Things) dengan Metode Exponential Smoothing. *Journal of Applied Electrical Engineering*, 5(2), 50–54. <https://doi.org/10.30871/jaee.v5i2.3172>
- Maricar, M. A. 2019. Analisa Perbandingan Nilai Akurasi Moving Average Dan Exponential Smoothing Untuk Sistem Peramalan Pendapatan Pada Perusahaan XYZ. *Jurnal Sistem Dan Informatika*, 13(2), 36–45.
- Menteri Perhubungan Republik Indonesia. 2019. *Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 15 Tahun 2019 Tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Bermotor Umum Dalam Trayek*. In PERMEN Perhubungan Republik Indonesia. http://jdih.dephub.go.id/assets/uudocs/permen/2015/PM_180_Tahun_2015.pdf
- Muhammad Fathan; Azwar. 2022. *Tiga Faktor Penyebab Mobil Kecelakaan dan Terbalik di Jalanan*. <https://otomotif.kompas.com/read/2020/08/02/084100515/tiga-faktor-penyebab-mobil-kecelakaan-dan-terbalik-di-jalanan>, diakses tanggal 1 Juni 2022

- Ostertagová, E., & Ostertag, O. 2011. The Simple Exponential Smoothing Model. *Faculty of Mechanical Engineering*, 380–384.
- Ramadhana, M., & Putra, S. 2020. Tesis. *Sistem Kontrol Suhu Dan Kelembapan Pada Budidaya Jamur Tiram Berbasis Iot*. Politeknik Menufaktur Negeri Bangka Belitung.
- Suprayogi, A., Fitriyah, H., & Tibyani. 2019. Sistem Pendeteksi Kecelakaan Pada Sepeda Motor Berdasarkan Kemiringan Menggunakan Sensor Gyroscope Berbasis Arduino. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(3), 3079–3085.
- Syanwani. 2018. Tesis. *Peran Transportasi Air Pada Perekonomian Masyarakat Desa Teluk Rendah Ilir Kecamatan Teboilir Kabupaten Tebo*. Universitas Islam Negeri Sulthan Thaha Saifuddin Jambi. (Vol. 11, Issue 1).
<http://link.springer.com/10.1007/978-3-319-59379-1>
<http://dx.doi.org/10.1016/B978-0-12-420070-8.00002-7>
<http://dx.doi.org/10.1016/j.ab.2015.03.024>
<https://doi.org/10.1080/07352689.2018.1441103>
<http://www.chile.bmw-motorrad.cl/sync/showroom/lam/es/>



UNIVERSITAS
Dinamika