

UNIVERSITAS  
**Dinamika**

**SISTEM DETEKSI JARAK AMAN KENDARAAN SEPEDA MOTOR  
BERBASIS WIFI**

**TUGAS AKHIR**



**Program Studi  
S1 TEKNIK KOMPUTER**

UNIVERSITAS  
**Dinamika**

**Oleh:**

**Aviva Salsabilah Harun**

**19410200038**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2023**

**SISTEM DETEKSI JARAK AMAN KENDARAAN SEPEDA MOTOR  
BERBASIS WIFI**

**TUGAS AKHIR**

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



UNIVERSITAS  
**Dinamika**

Oleh:

**Nama : Aviva Salsabilah Harun**  
**NIM : 19410200038**  
**Program Studi : S1 Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA**

**2023**

## Tugas Akhir

### SISTEM DETEKSI JARAK AMAN KENDARAAN SEPEDA MOTOR BERBASIS WIFI

Dipersiapkan dan disusun oleh

**Aviva Salsabilah Harun**  
NIM: 19410200038

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 1 Agustus 2023

#### Susunan Dewan Pembahas

Pembimbing:

I. **Hariato, S.Kom., M.Eng.**  
NIDN: 0722087701

II. **Pauladie Susanto, S.Kom., M.T.**  
NIDN: 0729047501

Pembahas:

I. **Musayyanah, S.ST., M.T.**  
NIDN: 0730069102

cn=Harianto Harianto,  
o=Universitas Dinamika,  
ou=Prodi S1 Teknik Komputer,  
email=hari@dinamika.ac.id, c=ID  
2023.08.02 11:09:46 +07'00'

cn=Pauladie Susanto, o=FTI  
Undika, ou=Prodi S1 TK,  
email=pauladie@dinamika.ac.id,  
c=ID  
2023.08.02 11:01:53 +07'00'

Digitally signed by  
Musayyanah  
DN: cn=Musayyanah,  
o=Universitas Dinamika,  
ou=S1 Teknik Komputer,  
email=musayyanah@dyna  
mika.ac.id, c=ID  
Date: 2023.08.02 13:48:25  
+07'00'  
Adobe Acrobat Reader  
version: 2023.003.20244

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

Digitally signed by  
Universitas Dinamika  
Date: 2023.08.03  
09:31:12 +07'00'

**Tri Sagirani, S.Kom., M.MT.**  
NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika  
UNIVERSITAS DINAMIKA



**Amsal 23:18**

**“Karena masa depan sungguh ada, dan harapanmu tidak akan hilang”**

UNIVERSITAS  
**Dinamika**



“Dipersembahkan kepada Papa, Mama, Kerabat, serta seluruh orang yang menyayangi saya, mendukung, memotivasi dan memberikan doa terbaiknya kepada saya.”

UNIVERSITAS  
**Dinamika**

**PERNYATAAN  
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Aviva Salsabilah Harun**  
NIM : **19410200038**  
Program Studi : **S1 Teknik Komputer**  
Fakultas : **Teknologi dan Informatika**  
Jenis Karya : **Laporan Tugas Akhir**  
Judul Karya : **SISTEM DETEKSI JARAK AMAN KENDARAAN  
SEPEDA MOTOR BERBASIS WIFI**

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 Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan di kelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 16 Juni 2023



Aviva Salsabilah Harun  
NIM: 19410200038



## ABSTRAK

Dalam berkendara, fokus merupakan salah satu faktor yang harus diperhatikan oleh pengemudi. Jika pengemudi kehilangan fokus, maka saat terjadi kecelakaan tepat di depan atau di belakang pengemudi lain, besar kemungkinan pengemudi tersebut akan terkejut sehingga sulit untuk mengendalikan kendaraannya. Dalam hal ini teknologi sangatlah penting dalam membantu meningkatkan keamanan serta keselamatan dalam berkendara. Pada penelitian ini dibuat alat sebagai deteksi jarak aman kendaraan dengan menggunakan WiFi, dimana nilai RSSI yang di hasilkan dari WiFi akan dikonversikan ke satuan jarak. Hasil jarak yang didapatkan akan digunakan sebagai acuan untuk perbandingan jarak aman. Sistem ini juga dibuat komunikasi antar *node* yang menghasilkan keluaran berupa LED sebagai indikator cahaya, LCD untuk menampilkan nilai jarak dan kecepatan serta Buzzer sebagai indikator suara. Hasil dari penelitian ini, konversi jarak menggunakan *log-normal shadowing model* dengan memanfaatkan RSSI Wi-Fi sebagai parameter pengukuran pada area *outdoor* mendapatkan rata-rata *error* terkecil yaitu 13,17% pada jarak 20 meter. Konversi perubahan dari sensor MPU-6050 dari akselerasi ke kecepatan menggunakan *library* mendapatkan nilai hasil uji coba saat kendaraan berada pada kecepatan *real* 21-30 km/jam menghasilkan nilai *error* 6,73%, pada kecepatan *real* 31-40 km/jam menghasilkan nilai *error* 8,41%, sedangkan untuk kecepatan *real* 41-50 km/jam menghasilkan nilai *error* 7,13%. Dalam pengujian alat keseluruhan, pada jarak RSSI 11-15 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED merah menyala, buzzer berbunyi menandakan bahwa jarak melebihi batas aman, kemudian pada jarak RSSI 16-30 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED kuning menyala, buzzer berbunyi dengan jeda menandakan bahwa jarak masuk peringatan batas aman, sedangkan pada Jarak RSSI diatas 30 meter dengan kecepatan 21- 30 km/jam mendapatkan hasil keluaran LCD menyala, LED hijau menyala, dan buzzer mati menandakan bahwa jarak sangat aman.

**Kata Kunci:** *Distance, RSSI, Log Normal Shadowing*

## KATA PENGANTAR

Puji Syukur dipanjatkan kehadirat Tuhan yang Maha Esa, atas berkat, rahmat, kasih, serta karunia-Nya yang telah diberikan kepada penulis, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “SISTEM DETEKSI JARAK AMAN KENDARAAN SEPEDA MOTOR BERBASIS WIFI”. Penyusunan Laporan Tugas Akhir ini disusun dalam rangka memenuhi salah satu syarat dalam menyelesaikan Program Sarjana Teknik Komputer di Universitas Dinamika.

Melalui kesempatan ini, penulis akan mengucapkan rasa terima kasih terhadap semua yang sudah memberikan dukungan, motivasi, dan juga bimbingan dalam upaya untuk menyelesaikan laporan Tugas Akhir ini. Oleh karena itu ucapan terima kasih ini di persembahkan kepada :

1. Orang Tua yang telah memberikan kontribusi besar berupa dukungan penuh dalam apa yang akan dikerjakan penulis sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT. Selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T. Selaku Ketua Program Studi S1 Teknik Komputer serta menjadi Dosen Pembimbing II yang selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir.
4. Bapak Harianto, S.Kom., M.Eng. Selaku Dosen Pembimbing I yang selalu memberikan waktu, bimbingan, serta ilmu dalam menyelesaikan Tugas Akhir.
5. Ibu Musayyanah, S.ST., M.T. Selaku Dosen Pembahas yang juga memberikan saran dan kritik dalam menyelesaikan Tugas Akhir.
6. Seluruh Dosen dan Laboran Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu, dan juga bimbingan yang sangat berguna.
7. Ilham Fahmi Amri Rosada sebagai orang terdekat yang sudah membantu dan memberikan dukungan penuh pada penulis sehingga dapat menyelesaikan Tugas Akhir.



8. Seluruh teman dan sahabat S1 Teknik Komputer angkatan 2019 yang telah memberikan dukungan dan semangat untuk membantu penulis untuk menyelesaikan seluruh rangkaian perkuliahan.

Penulis menyadari dalam Laporan Tugas Akhir ini masih jauh dari kata sempurna dan juga masih banyak terdapat kekurangan dalam penyusunan Laporan ini. Maka dari itu, ijinkan penulis memohon maaf apabila dan Laporan Tugas Akhir ini masih terdapat kesalahan, baik dalam penulisan maupun bahasa yang digunakan. Penulis juga selalu terbuka dalam kritik dan saran yang diberikan oleh para pembaca yang bersifat membangun untuk kesempurnaan Laporan ini.

Surabaya, 15 Juni 2023

Penulis



UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

|   |             |
|---|-------------|
| <b>SURAT PERNYATAAN .....</b>               | <b>vi</b>   |
| <b>ABSTRAK .....</b>                        | <b>vii</b>  |
| <b>KATA PENGANTAR.....</b>                  | <b>viii</b> |
| <b>DAFTAR ISI.....</b>                      | <b>x</b>    |
| <b>DAFTAR GAMBAR.....</b>                   | <b>xiii</b> |
| <b>DAFTAR TABEL .....</b>                   | <b>xiv</b>  |
| <b>DAFTAR LAMPIRAN .....</b>                | <b>xv</b>   |
| <b>BAB I PENDAHULUAN.....</b>               | <b>1</b>    |
| 1.1 Latar Belakang .....                    | 1           |
| 1.2 Rumusan Masalah .....                   | 2           |
| 1.3 Batasan Masalah.....                    | 2           |
| 1.4 Tujuan.....                             | 2           |
| 1.5 Manfaat.....                            | 3           |
| <b>BAB II LANDASAN TEORI .....</b>          | <b>4</b>    |
| 2.1 Jarak Aman Kendaraan.....               | 4           |
| 2.2 Wi-Fi .....                             | 5           |
| 2.3 ESP32 NodeMCU-32S.....                  | 6           |
| 2.4 GY-521 MPU-6050 .....                   | 7           |
| 2.5 LCD OLED 0.91 .....                     | 8           |
| 2.6 LED .....                               | 8           |
| 2.7 Buzzer.....                             | 9           |
| 2.8 <i>Log Normal Shadowing Model</i> ..... | 10          |
| <b>BAB III METODOLOGI PENELITIAN .....</b>  | <b>12</b>   |
| 3.1 Blok Diagram <i>Node</i> .....          | 12          |
| 3.2 Rangkaian Skematik Perangkat.....       | 13          |

|   |    |
|---|----|
| 3.3 Perancangan <i>Hardware</i> .....                           | 14 |
| 3.4 Implementasi <i>Log-Normal Shadowing Model</i> .....        | 15 |
| 3.5 Protokol Komunikasi antar <i>Node</i> .....                 | 16 |
| 3.5.1 Komunikasi Wi-Fi .....                                    | 16 |
| 3.6 Skenario Pengambilan Data .....                             | 17 |
| 3.7 Algoritma Antar <i>Node</i> .....                           | 18 |
| 3.7.1 Fungsi <i>ModeAP</i> .....                                | 19 |
| 3.7.2 Fungsi <i>ModeStation</i> .....                           | 19 |
| 3.7.3 Fungsi LogDistrib ( <i>Log-Normal Distribution</i> )..... | 20 |
| 3.7.4 Fungsi LNSM( <i>Log-Normal Shadowing Model</i> ).....     | 21 |
| 3.7.5 Fungsi runMPU .....                                       | 22 |
| 3.7.6 Fungsi <i>DisplayOled</i> .....                           | 23 |
| 3.7.7 Fungsi <i>LedNotif</i> .....                              | 23 |
| 3.7.8 Fungsi CekKecepatan 60 km/jam.....                        | 24 |
| 3.7.9 Fungsi CekKecepatan 50 km/jam.....                        | 25 |
| 3.7.10 Fungsi CekKecepatan 40 km/jam.....                       | 26 |
| 3.7.11 Fungsi CekKecepatan 30 km/jam.....                       | 27 |

**BAB IV HASIL DAN PEMBAHASAN ..... 28**

|  |    |
|--|----|
| 4.1 Perbandingan RSSI terhadap Jarak .....                             | 28 |
| 4.1.1 Tujuan Pengujian Perbandingan RSSI terhadap Jarak .....          | 28 |
| 4.1.2 Prosedur Pengujian Perbandingan RSSI terhadap Jarak .....        | 28 |
| 4.1.3 Hasil Pengujian Perbandingan RSSI terhadap Jarak .....           | 28 |
| 4.2 Pengujian Perubahan RSSI ke Jarak menggunakan LNSM.....            | 29 |
| 4.2.1 Tujuan Pengujian Perubahan RSSI ke Jarak menggunakan LNSM.....   | 29 |
| 4.2.2 Prosedur Pengujian Perubahan RSSI ke Jarak menggunakan LNSM..... | 29 |
| 4.2.3 Hasil Pengujian Perubahan RSSI ke Jarak menggunakan LNSM.....    | 29 |

|  |           |
|--|-----------|
| 4.3 Pengujian Kecepatan .....                          | 31        |
| 4.3.1 Tujuan Pengujian Kecepatan .....                 | 31        |
| 4.3.2 Prosedur Pengujian Kecepatan .....               | 32        |
| 4.3.3 Hasil Pengujian Kecepatan .....                  | 32        |
| 4.4 Pengujian Keseluruhan <i>Node</i> .....            | 33        |
| 4.4.1 Tujuan Pengujian Keseluruhan <i>Node</i> .....   | 33        |
| 4.4.2 Prosedur Pengujian Keseluruhan <i>Node</i> ..... | 33        |
| 4.4.3 Hasil Pengujian Keseluruhan <i>Node</i> .....    | 33        |
| <b>BAB V PENUTUP .....</b>                             | <b>35</b> |
| 5.1 Kesimpulan .....                                   | 35        |
| 5.2 Saran .....  | 36        |
| <b>DAFTAR PUSTAKA .....</b>                            | <b>37</b> |
| <b>LAMPIRAN .....</b>                                  | <b>39</b> |



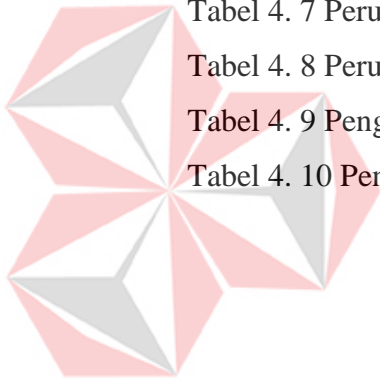
UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2. 1 Jarak Aman Kendaraan .....                                     | 4  |
| Gambar 2. 2 NodeMCU-32S .....  | 6  |
| Gambar 2. 3 Development Board NodeMCU-32S .....                            | 7  |
| Gambar 2. 4 Sensor GY-521 MPU-6050.....                                    | 7  |
| Gambar 2. 5 LCD OLED 0.91 .....  | 8  |
| Gambar 2. 6 LED .....  | 9  |
| Gambar 2. 7 Buzzer.....  | 9  |
| Gambar 3. 1 Komunikasi Antar <i>Node</i> .....                             | 12 |
| Gambar 3. 2 Rangkaian Skematik <i>Node</i> .....                           | 13 |
| Gambar 3. 3 Tampak Depan <i>Node</i> .....                                 | 14 |
| Gambar 3. 4 Tampak Dalam <i>Node</i> .....                                 | 14 |
| Gambar 3. 5 Tampak Sensor <i>Accelerometer</i> .....                       | 14 |
| Gambar 3. 6 <i>Flowchart</i> perhitungan <i>log-normal shadowing</i> ..... | 15 |
| Gambar 3. 7 Program <i>Log-Normal Shadowing</i> .....                      | 16 |
| Gambar 3. 8 Komunikasi Antar <i>Node</i> .....                             | 16 |
| Gambar 3. 9 Posisi <i>Node</i> .....                                       | 17 |
| Gambar 3. 10 <i>Flowchart Global</i> .....                                 | 18 |
| Gambar 3. 11 <i>Flowchart ModeAP</i> .....                                 | 19 |
| Gambar 3. 12 <i>Flowchart ModeStation</i> .....                            | 19 |
| Gambar 3. 13 <i>Flowchart LogNormalDistrib</i> .....                       | 20 |
| Gambar 3. 14 <i>Flowchart</i> Fungsi <i>LSNM</i> .....                     | 21 |
| Gambar 3. 15 <i>Flowchart</i> Fungsi <i>runmpu()</i> .....                 | 22 |
| Gambar 3. 16 <i>Flowchart</i> Fungsi <i>DisplayOled</i> .....              | 23 |
| Gambar 3. 17 <i>Flowchart</i> Fungsi <i>LedNotif()</i> .....               | 23 |
| Gambar 3. 18 <i>Flowchart</i> Fungsi <i>CekKecepatan 60 km/jam</i> .....   | 24 |
| Gambar 3. 19 <i>Flowchart</i> Fungsi <i>CekKecepatan 50 km/jam</i> .....   | 25 |
| Gambar 3. 20 <i>Flowchart</i> Fungsi <i>CekKecepatan 40 Km/jam</i> .....   | 26 |
| Gambar 3. 21 <i>Flowchart</i> Fungsi <i>CekKecepatan 30 Km/jam</i> .....   | 27 |

## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 2. 1 Spesifikasi NodeMCU-32S.....  | 6  |
| Tabel 2. 2 Spesifikasi OLED 0.91 .....   | 8  |
| Tabel 2. 3 Konfigurasi Pin OLED 0.91 .....   | 8  |
| Tabel 2. 4 Nilai Pathloss Exponent.....  | 11 |
| Tabel 3. 1 Rangkaian Pin dari Perangkat.....   | 13 |
| Tabel 4. 1 Pengujian Perubahan RSSI terhadap Jarak.....                              | 28 |
| Tabel 4. 2 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 15 meter ..... | 29 |
| Tabel 4. 3 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 20 meter ..... | 30 |
| Tabel 4. 4 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 25 meter ..... | 30 |
| Tabel 4. 5 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 30 meter ..... | 30 |
| Tabel 4. 6 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 40 meter ..... | 30 |
| Tabel 4. 7 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 50 meter ..... | 31 |
| Tabel 4. 8 Perubahan Jarak <i>Real</i> ke Jarak Hasil RSSI pada jarak 60 meter ..... | 31 |
| Tabel 4. 9 Pengujian Prediksi Kecepatan Menggunakan Sensor.....                      | 32 |
| Tabel 4. 10 Pengujian Keseluruhan Alat Pada Kecepatan 21-30 Km/jam .....             | 33 |



UNIVERSITAS  
Dinamika

## DAFTAR LAMPIRAN

|   |     |
|---|-----|
| Lampiran 1. Pengambilan Data RSSI ke Jarak 15 Meter.....                | 39  |
| Lampiran 2. Pengambilan Data RSSI ke Jarak 20 Meter.....                | 44  |
| Lampiran 3. Pengambilan Data RSSI ke Jarak 25 Meter.....                | 50  |
| Lampiran 4. Pengambilan Data RSSI ke Jarak 30 Meter.....                | 55  |
| Lampiran 5. Pengambilan Data RSSI ke Jarak 40 Meter.....                | 61  |
| Lampiran 6. Pengambilan Data RSSI ke Jarak 50 Meter.....                | 66  |
| Lampiran 7. Pengambilan Data RSSI ke Jarak 60 Meter.....                | 71  |
| Lampiran 8. Pengambilan Data Kecepatan.....                             | 78  |
| Lampiran 9. Pengambilan Data Keseluruhan.....                           | 80  |
| Lampiran 10. Program <i>Node 1</i> .....                                | 83  |
| Lampiran 11. Program <i>Node 2</i> .....                                | 88  |
| Lampiran 12. Program <i>Node 3</i> .....                                | 94  |
| Lampiran 13. <i>Basic Program Selected Wifi Multi Mode Access</i> ..... | 100 |
| Lampiran 14. Kartu Bimbingan .....                                      | 101 |
| Lampiran 15. Hasil Plagiasi .....                                       | 102 |



UNIVERSITAS  
Dinamika



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Untuk mengendarai kendaraan bermotor, pengendara harus memenuhi syarat untuk mengemudikan kendaraan tersebut di jalan raya. Faktor yang sangat perlu di perhatikan pengemudi adalah tetap fokus dalam berkendara. Jika pengendara kurang fokus dalam berkendara, maka saat terjadi kecelakaan tepat di belakang atau di depan pengemudi lain, besar kemungkinan pengemudi tersebut akan terkejut sehingga berakibat sulit untuk mengendalikan kendaraannya. Salah satu penyebab dari banyaknya kecelakaan yang terjadi yaitu kurangnya kesadaran pengemudi dalam hal menjaga jarak aman berkendara (Suartawan et al., 2021).

Jumlah kecelakaan lalu lintas terus bertambah dan meningkat setiap tahunnya, hal ini beriringan juga dengan meningkatnya pengguna kendaraan bermotor (Syahriza, 2019). Dilansir dari Polresta Sidoarjo mencatat selama bulan januari 2023 telah terjadi sebanyak 233 kecelakaan lalu lintas. Bila dirinci berdasarkan jenis kendaraan yaitu 194 sepeda motor, 46 mobil, 1 bus, 2 kendaraan roda tiga, dan 5 kendaraan tidak bermotor. Kemudian dari faktor penyebab yaitu faktor manusia diantaranya lelah 68 kasus, mengantuk 21 kasus, melanggar alat pemberi isyarat lalu lintas 12 kasus, menyalip kendaraan tidak hati-hati 43 kasus, menyebrang tidak hati-hati 33 kasus, dan melawan arus 33 kasus. Iptu Ony Purnomo Kanit Laka Lantas Polresta Sidoarjo mengatakan, faktor kelalaian manusia mendominasi sejumlah kecelakaan di Kabupaten Sidoarjo. Alasan terjadinya kecelakaan lalu lintas paling banyak dikarenakan kelalaian, tidak bisa menjaga jarak, tidak memperhatikan jarak yang cukup, dan berkendara tanpa menggunakan kelengkapan mengemudi (Suarasurabaya.net, 2023).

Berdasarkan permasalahan tersebut teknologi memiliki peranan penting dalam membantu meningkatkan keamanan serta keselamatan berkendara. Pada penelitian sebelumnya (Safiro, 2022) membuat rancang bangun simulasi alat pendeteksi jarak aman antar kendaraan menggunakan sensor ultrasonik berbasis *Internet of Things*. Dalam penelitian tersebut merancang indera pendeteksi jarak aman berkendara yang memanfaatkan sensor ultrasonik untuk mengukur jeda

kondusif yang terdapat di depan dan di belakang kendaraan. Namun batas pengujian alat dengan jarak sensor hanya 1 meter yang diindikasikan dengan LED hijau, 0,7 meter diindikasikan dengan LED kuning dan 0.50 meter diindikasikan dengan LED merah.

Maka dari itu, dalam penelitian ini diusulkan alat sebagai deteksi jarak aman kendaraan sepeda motor dengan menggunakan Wi-Fi RSSI, dimana data RSSI yang didapat akan dikonversi ke satuan jarak sehingga hasilnya digunakan sebagai acuan untuk perbandingan jarak aman. Selain itu, perancangan sistem ini juga dibuat komunikasi antar *node* yang kemudian hasil keluarannya berupa LED, LCD, dan *Buzzer*.

## 1.2 Rumusan Masalah

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

1. Bagaimana mendapatkan nilai RSSI melalui jaringan Wi-Fi pada ESP32?
2. Bagaimana mendapatkan nilai kecepatan menggunakan sensor MPU-6050?
3. Bagaimana menerapkan jarak aman sepeda motor menggunakan nilai RSSI Wi-Fi?

## 1.3 Batasan Masalah

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

1. Proyek masih dalam bentuk *prototype*.
2. Sensor MPU-6050 hanya bisa memprediksi nilai kecepatan dengan perubahan dari medan area serta pergerakan motor.

## 1.4 Tujuan

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

1. Mengetahui nilai RSSI Wi-Fi menggunakan mikrokontroler ESP32.
2. Mengetahui kecepatan kendaraan dengan sensor MPU-6050.

3. Dapat menerapkan sistem pendeteksi jarak aman sepeda motor menggunakan *Log-Normal Shadowing Model*.

### **1.5 Manfaat**

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Dapat menambahkan pengetahuan dan penerapan tentang keamanan berkendara.
2. Mengantisipasi terjadinya kecelakaan akibat tidak menjaga jarak aman dalam berkendara.
3. Mengurangi tingkat kecelakaan sepeda motor.



UNIVERSITAS  
**Dinamika**

## BAB II

### LANDASAN TEORI

#### 2.1 Jarak Aman Kendaraan

Mengendarai sepeda motor masih merupakan kendaraan favorit masyarakat dibandingkan dengan mobil pribadi, angkutan umum, ataupun kendaraan lain. Selain efisiensi waktu, motor juga lebih hemat dalam pengeluaran biaya bahan bakar. Dalam hal ini tentunya untuk mengendarai sepeda motor banyak hal yang perlu di perhatikan untuk keselamatan berkendara terutama menjaga jarak aman kendaraan. Dilansir dari laman resmi Kompas.com, untuk mengendarai motor juga butuh keahlian, saat tidak bisa menjaga keseimbangan maka akan sangat mudah terjatuh. Salah satu cara agar aman ketika mengendarai motor yaitu dengan menjaga jarak dengan kendaraan didepannya (Kompas.com, 2020).

Menjaga jarak aman saat berkendara merupakan kewajiban untuk setiap pengendara yang telah diatur dalam UU Pasal 62 PP No.43 tahun 1993 tentang Jarak Antar Kendaraan (IndonesiaPemerintah, 1993). Untuk menjaga jarak aman, dibutuhkan sekitar 0,5 hingga 1 detik untuk melakukan rem mendadak dan ini merupakan waktu reflek seseorang. Setelah melakukan pengereman, motor membutuhkan 0,5 sampai 1 detik untuk dapat berhenti dengan sempurna. Berdasarkan hal tersebut, maka dipertimbangkan 3 detik ditentukan sebagai jarak yang paling aman untuk kendaraan (Kompas.com, 2020).



| Kecepatan    | Jarak Minimal | Jarak Aman |
|--------------|---------------|------------|
| 30 km / jam  | 15 meter      | 30 meter   |
| 40 km / jam  | 20 meter      | 40 meter   |
| 50 km / jam  | 25 meter      | 50 meter   |
| 60 km / jam  | 40 meter      | 60 meter   |
| 70 km / jam  | 50 meter      | 70 meter   |
| 80 km / jam  | 60 meter      | 80 meter   |
| 90 km / jam  | 70 meter      | 90 meter   |
| 100 km / jam | 80 meter      | 100 meter  |

Gambar 2. 1 Jarak Aman Kendaraan  
(Sumber: Kompas.com, 2020)

Untuk dapat menentukan jarak aman berkendara, maka harus mengetahui berapa jarak tempuh kendaraan dalam waktu 1 detik berdasarkan kecepatan. Berikut cara menentukan jarak aman yang ideal dapat dilihat pada persamaan (1).

Misal,

$$\text{Kecepatan} \frac{30 \text{ km}}{3.600 \text{ detik}}$$

maka, jarak tempuh 1 km menggunakan waktu 120 detik

$$v = \frac{1 \text{ km}}{120 \text{ detik}}$$

$$t = 3 \text{ detik} \quad (1)$$

$$s = v \cdot t$$

$$\frac{1 \text{ km}}{120 \text{ detik}} \cdot 3 \text{ detik}$$

$$\frac{1}{40} \text{ km} = \frac{1}{40} \text{ km} \cdot 1000 \text{ m} = 25 \text{ m}$$

## 2.2 Wi-Fi

*Wireless Fidelity* atau biasa disingkat Wi-Fi merupakan teknologi yang banyak digunakan saat ini karena dapat memancarkan koneksi internet tanpa menggunakan kabel. Wi-Fi biasanya digunakan dilingkungan tempat tinggal, lingkungan kerja, kampus, dan tempat umum lainnya (Pusvita & Huda, 2019).

Wi-Fi bekerja menggunakan gelombang radio. Kelebihan jaringan Wi-Fi yaitu *user* tidak dibatasi dengan ruang gerak namun dibatasi dengan jarak jangkauan pemancar Wi-Fi (Musdalifa & Panu, 2019). Frekuensi Wi-Fi berdasarkan IEEE terbagi menjadi 4 bagian, yaitu:

1. 802.11b 11 Mb/s 2.4 GHz B
2. 802.11g 54 Mb/s 2.4 GHz B, G
3. 802.11a 54 Mb/s 5 GHz A
4. 802.11n 100 Mb/s 2.4 GHz B, G, N

### 2.3 ESP32 NodeMCU-32S



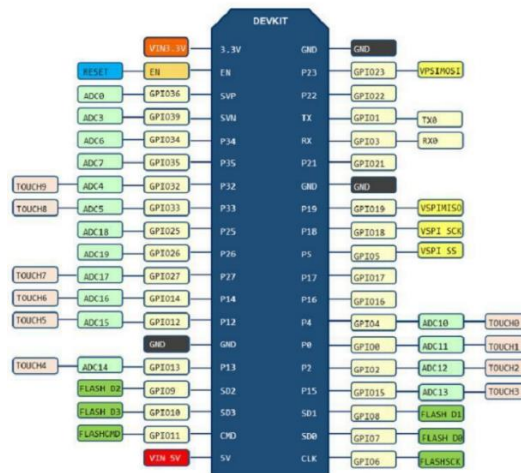
Gambar 2. 2 NodeMCU-32S

NodeMCU-32S merupakan sebuah modul yang dikembangkan oleh Axinke Technology. Inti dari modul ini ada *chip* ESP32 yang menyediakan fitur adaptif yang luas. Modul ini mengintegrasikan *Traditional Bluetooth*, *Low Power Bluetooth*, dan Wi-Fi. *Chip* WROOM-32U yang tertanam di *board* NodeMCU-32S ini tidak memiliki *antenna internal* dan harus menggunakan *antenna external*. Berdasarkan IEEE, frekuensi Wi-Fi pada NodeMCU-32S termasuk pada bagian 4, yaitu 802.11n 100 Mb/s 2.4 Ghz B,G,N. Berikut spesifikasi dari NodeMCU-32S ESP32:

Tabel 2. 1 Spesifikasi NodeMCU-32S

| Spesifikasi            | Keterangan                                       |
|------------------------|--|
| Kecepatan              | 150 Mbps   |
| Frekuensi Wi-Fi        | 2.400 ~ 2.483,5 MHz                              |
| Frekuensi Clock        | 80 MHz – 240 MHz                                 |
| ADC                    | 18 channels, resolution 12 bit                   |
| DAC                    | 2 channels, resolution 8 bit                     |
| Antena                 | PCB/antena eksternal                             |
| Interface              | GPIO, Capacitive sensor, ADC, DAC, PWM, I2C, I2S |
| SPI Flash              | standar 32Mb                                     |
| Operating Current      | 80mA   |
| Operating Voltage      | 2,2V – 3,6VDC                                    |
| Logic Level            | 3,3V   |
| Support                | Wi-Fi, Bluetooth                                 |
| Chip Serial            | CH340  |
| 11n mode, Output Power | 12 dBm (MIN) 14 dBm (MAX)                        |

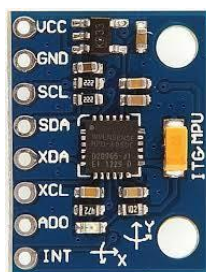
NodeMCU-32 V1.3 memiliki *interface* sebanyak 38 pin yang terhubung. Untuk *development board* dan fungsi pin dapat dilihat pada Gambar 2.3, berikut:



Gambar 2. 3 Development Board NodeMCU-32S

## 2.4 GY-521 MPU-6050

Sensor ini merupakan sensor yang dapat membaca kemiringan seperti sudut pada suatu benda atau objek berdasarkan data yang dihasilkan dari *gyroscope* dan *accelerometer* (Natakusuma, 2018). Modul GY-521 MPU-6050 adalah modul inti dari MPU-6050 yang merupakan 6 axis *Motion Processing Unit* yang lengkap sehingga membuat modul ini siap dipakai dengan tegangan *supply* sebesar 3-5 VDC (Mangkusasmito et al., 2020). Sensor ini dapat dikategorikan sebagai sensor *Inertial Measurement Unit* (IMU) yang memanfaatkan sistem pengukuran *Gyroscopes* dan *Accelerometers* untuk memperkirakan posisi yang relatif, kecepatan dan percepatan suatu objek.



Gambar 2. 4 Sensor GY-521 MPU-6050  
(Sumber: (Natakusuma, 2018))



## 2.5 LCD OLED 0.91

OLED (*Organic Lamp Emitting Diode*) memiliki kontras layar yang tinggi sehingga tidak membutuhkan cahaya latar. OLED 0.91 merupakan modul dengan tampilan grafis monokrom yang berukuran 0.91 inci dan resolusi 128x32 pixel yang menggunakan komunikasi I2C. OLED 0.91 tidak hanya menampilkan karakter, tapi juga dapat menampilkan gambar dan animasi (Yogie & Habi, 2019).



Gambar 2. 5 LCD OLED 0.91

Untuk mengetahui spesifikasi OLED 0.91 dapat dilihat pada Tabel 3 sebagai berikut:

Tabel 2. 2 Spesifikasi OLED 0.91

| Spesifikasi   | Keterangan |
|---------------|------------|
| Tegangan      | 3,3 – 5V   |
| Driver IC     | SSD1306    |
| Komunikasi    | I2C        |
| Resolusi      | 128x32     |
| Display Color | White/Blue |

Untuk mengetahui konfigurasi pin OLED 0.91 dapat dilihat pada Tabel 4 sebagai berikut:

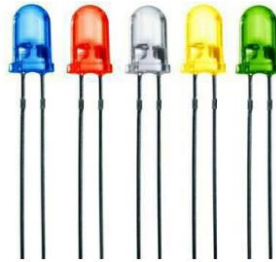
Tabel 2. 3 Konfigurasi Pin OLED 0.91

| Jenis Pin | Keterangan  |
|-----------|---|
| GND       | GND   |
| VCC       | 3,3 – 5V  |
| SCL       | Di hubungkan dengan pin <i>serial clock</i> (SCL) |
| SDA       | Dihubungkan dengan pin <i>serial data</i> (SDA)   |

## 2.6 LED

*Light Emitting Diode* atau sering disingkat LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang digunakan. LED juga dapat memancarkan sinar

inframerah yang tidak tampak oleh mata seperti pada *Remote Control* TV ataupun *Remote Control* perangkat elektronik lainnya.



Gambar 2. 6 LED

Bentuk LED mirip dengan sebuah bola lampu yang kecil dan dapat dipasangkan dengan mudah kedalam berbagai perangkat elektronika. Berbeda dengan lampu pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. Maka dari itu saat ini LED yang bentuknya kecil telah banyak digunakan sebagai lampu penerang dalam LCD TV untuk menggantikan lampu *tube* (Kho, Dickson, 2014).

## 2.7 Buzzer

Buzzer merupakan sebuah komponen elektronika yang dapat menghasilkan getaran suara berupa gelombang bunyi. Buzzer akan menghasilkan getaran suara saat jumlah tegangan yang diberikan sesuai dengan spesifikasi bentuk dan ukuran buzzer. Bentuk dari buzzer seperti tabung silinder dengan sebuah lubang kecil dibagian atas dan dua buah pin di bagian bawah.

Bentuk Buzzer      Simbol Buzzer



Gambar 2. 7 Buzzer

Buzzer biasanya digunakan sebagai *alarm*, komponen rangkaian anti maling, sebagai *timer*, indikator suara sebagai tanda peringatan bahaya dan lain-lain. Prinsip kerja buzzer terdiri dari kumparan yang terpasang secara diafragma. Ketika kumparan tersebut dialiri listrik, maka akan menjadi elektromagnet. Karena kumparan dipasang secara diafragma, maka setiap kumparan akan menggerakkan

diafragma tersebut secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara (Prastyo, 2022).

## 2.8 Log Normal Shadowing Model

*Receive signal strength indicator* atau biasa disingkat RSSI merupakan sebuah teknologi yang digunakan untuk mengukur kekuatan sinyal yang diterima oleh perangkat *wireless*. Untuk mengubah sinyal dalam bentuk jarak adalah menggunakan *Log-normal shadowing model* yang cocok dalam lingkungan *outdoor*.

*Log-normal shadowing model* adalah model yang banyak dikembangkan dan memiliki parameter yang dapat dikonfigurasi dan cukup stabil bila digunakan pada lingkungan *indoor* dan *outdoor*. Perhitungan ini diperoleh dengan mempertimbangkan nilai *path loss* yang diambil pada Model Okumura-Hatta, yang dimana untuk mendapatkan nilai  $PL(d_0)$  dan  $PL(d)$  dengan melihat kondisi lingkungan dan nilai *path loss exponent* sebagai estimasi jarak (Yanziah et al., 2020).

$$PL(d_0) = RSSI - 10n \log(d_0) \quad (2)$$

$$\overline{PL}(d) = \overline{PL}(d_0) + 10n \log\left(\frac{d}{d_0}\right) \quad (3)$$

Dimana,  $PL(d)$  adalah *path loss* pada jarak  $d$  dalam meter untuk *outdoor*,  $PL_0$  adalah *path loss* pada jarak referensi  $d_0$  (1 meter).  $PL_0$  dapat diperoleh dari pengukuran dilapangan,  $n$  adalah *path loss exponent* yang disesuaikan dengan kondisi lingkungan,  $d$  adalah jarak antara *access point* dan *node* dalam meter,  $X\sigma$  adalah variabel *random* yang terdistribusi *gaussian* dengan *zero mean* dan standar deviasi  $\sigma$  (dalam dB). Variabel  $n$  dan standar deviasi  $X\sigma$  akan disesuaikan dengan kondisi lingkungan pengujian.  $P_T$  adalah daya pancar *node* dalam dBm (Yanziah et al., 2020). Sehingga perhitungan RSSI dapat dilihat pada persamaan (4).

$$RSSI = P_T - (PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X\sigma) \quad (4)$$

Sedangkan untuk mendapatkan nilai  $X\sigma$  menggunakan rumus *Log-Normal Distribution* yang dimana untuk mendapatkan nilai *Mean* dan *Standard Deviation* serta *Gaussian Random Distribution* berguna untuk mendapatkan nilai *Z table* yang nantinya sebagai parameter pada *Log-Normal Shadowing Model*. Nilai *Mean* sendiri didapatkan dari nilai rata – rata dari data RSSI yang didapat pada jarak yang ditentukan (Pavlovic, 2022). Untuk rumusnya dapat dilihat pada persamaan (5).

$$\hat{\mu} = \frac{\sum_k \ln RSSI_k}{total} \quad (5)$$

Untuk mendapatkan nilai *Standard Deviation*, maka total dari rata-rata perdata RSSI yang diakarkan menjadi nilai sigma. Rumusnya bisa dilihat pada persamaan (6) dan persamaan (7) adalah pecahan dari persamaan (5).

$$\hat{\sigma}^2 = \frac{\sum_k (\ln RSSI_k - \hat{\mu})^2}{total} \quad (6)$$

$$\hat{\sigma} = \sqrt{\frac{\sum_k (\ln RSSI_k - \hat{\mu})^2}{total}} \quad (7)$$

Selanjutnya untuk mendapatkan nilai jarak setiap meternya dalam *Log-Normal Shadowing Model*, bisa didapatkan dengan parameter yang telah dihitung sebelumnya. Untuk rumus bisa dilihat pada persamaan (8).

$$d = d_0 10^{-((RSSI - P_T + PL_0 - X\sigma)/10n)} \quad (8)$$

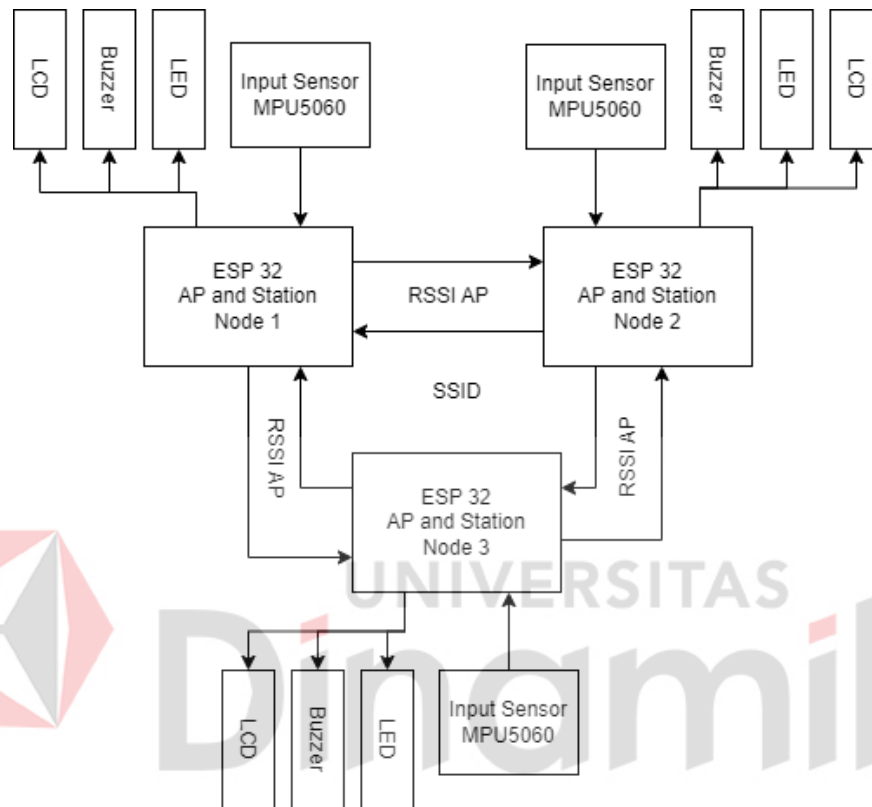
Tabel 2. 4 Nilai Pathloss Exponent

| <i>Environment</i>                   | <i>Path loss exponent</i> |
|--------------------------------------|---------------------------|
| <i>Free Space</i>                    | 2                         |
| <i>Urban area cellular radio</i>     | 2.7                       |
| <i>Shadowed Urban Cellular Radio</i> | 3 to 5                    |
| <i>In Building line of site</i>      | 1.6 to 8                  |
| <i>Obstructed in Building</i>        | 4 to 6                    |
| <i>Obstructed in Factory</i>         | 2 to 3                    |

(Sumber:(Isni et al., 2020))

## BAB III METODOLOGI PENELITIAN

### 3.1 Blok Diagram *Node*



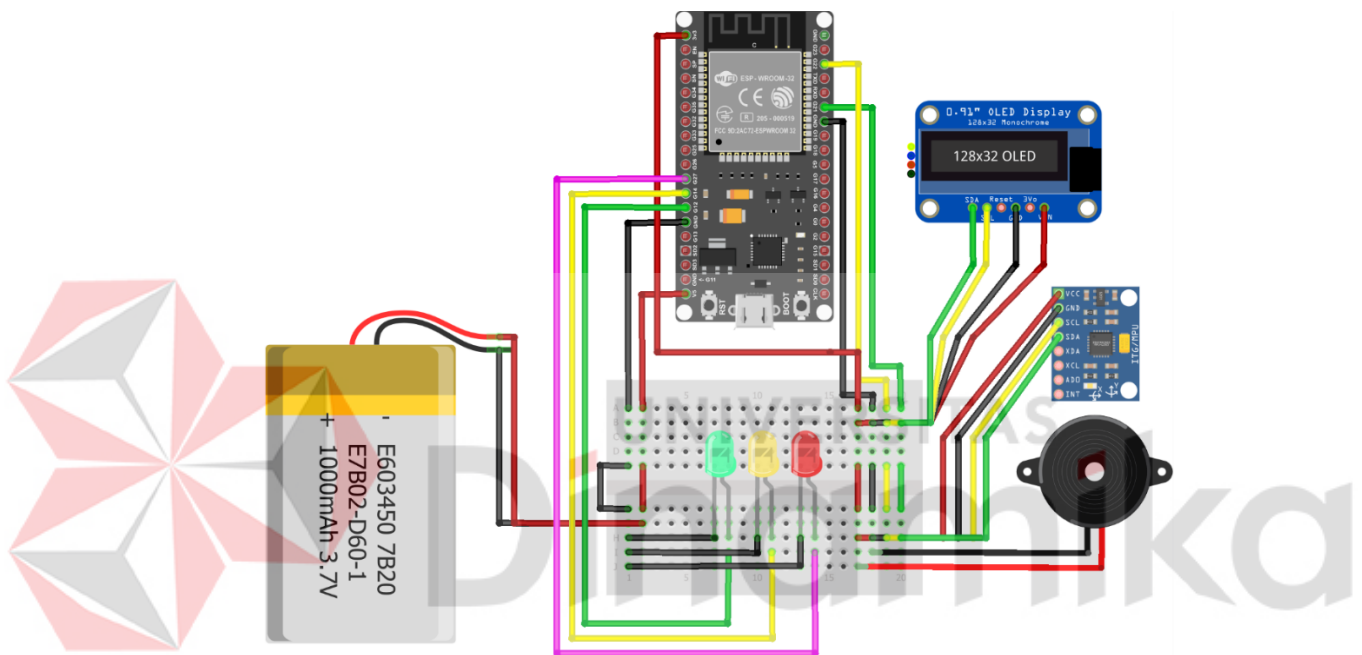
Gambar 3. 1 Komunikasi Antar *Node*

Pada Gambar 3.1, menunjukkan perancangan blok diagram komunikasi antar *node*. Perancangan ini terdapat inputan sensor MPU-6050 sebagai pengukur kecepatan motor dan terdapat 3 output, yaitu LED, LCD, dan Buzzer. Mikrokontroler yang digunakan adalah 3 buah ESP32S. Cara komunikasi *node* ini yaitu :

1. Input sensor MPU-6050 untuk mengukur kecepatan motor.
2. Masing-masing *node* akan bekerja sebagai *access point* dan *station* secara bersamaan, sehingga masing-masing *node* juga dapat memancarkan dan menerima SSID secara bersamaan.
3. Setelah menerima SSID, selanjutnya akan didapatkan data RSSI yang kemudian dikonversi ke satuan jarak.

4. *Node* melakukan proses mendapatkan nilai kecepatan dari sensor MPU-6050.
5. Setelah mendapatkan nilai kecepatan dan jarak, *node* akan mengeluarkan output LCD untuk menampilkan nilai jarak dan kecepatan.
6. Kemudian *node* mengeluarkan output LED dan Buzzer sebagai indikator aman dan bahaya.

### 3.2 Rangkaian Skematik Perangkat



Gambar 3. 2 Rangkaian Skematik *Node*

Keterangan pin dari Gambar 3.2 dapat dilihat pada tabel 3.1:

Tabel 3. 1 Rangkaian Pin dari Perangkat

| Komponen        | Pin Out            |
|-----------------|--------------------|
| Sensor MPU-6050 | 21 (SDA), 22 (SCL) |
| LCD Oled 0.91"  | 21 (SDA), 22 (SCL) |
| LED Red         | 27                 |
| LED Yellow      | 14                 |
| LED Green       | 12                 |

### 3.3 Perancangan *Hardware*

Perancangan perangkat keras terdiri dari ESP32S, LCD, LED, Buzzer dan sensor *accelerometer*.



Gambar 3. 3 Tampak Depan *Node*



Gambar 3. 4 Tampak Dalam *Node*

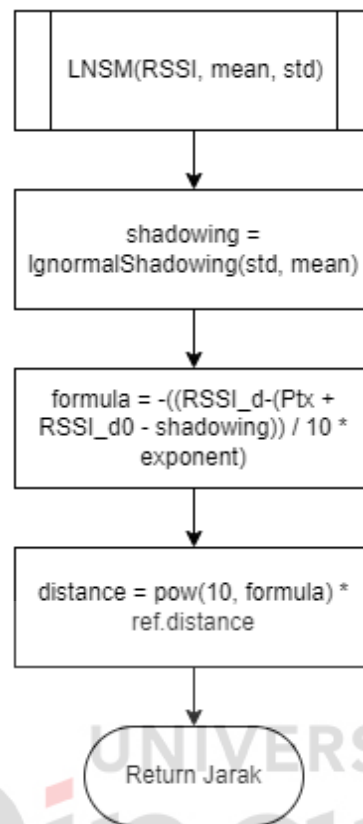
Perancangan sensor *accelerometer* dibuat wadah terpisah untuk memaksimalkan cara kerja dari sensor *accelerometer*. Sensor *accelerometer* hanya bisa memprediksi kecepatan melalui getaran dan kemiringan sepeda motor.



Gambar 3. 5 Tampak Sensor *Accelerometer*



### 3.4 Implementasi *Log-Normal Shadowing Model*



Gambar 3. 6 Flowchart perhitungan *log-normal shadowing*

Pada Gambar 3.6, dijelaskan alur dari perhitungan *log-normal shadowing*, dimana adanya inialisasi referensi jarak, referensi RSSI, nilai *path loss exponent*, standard deviasi (Sigma) dan *transmitter power* sebagai acuan untuk perhitungan dari *log-normal shadowing model*. Dari nilai inialisasi tadi akan dihitung untuk mencari *shadowing value* dengan acuan *value* dari standard deviasi (sigma). Nilai sigma ini bisa berbeda-beda tiap area dan lingkungan pengambilan, dikarenakan tingkat perubahan pada RSSI akan berubah-ubah. Untuk perhitungan Pengambilan data *shadowing* ini sudah ada dalam *library* mencakup perhitungan *gaussian* dan Z indeks 'mu'. Setelah mendapatkan nilai *shadowing* maka akan masuk perhitungan *log-normal shadowing model*, dengan beberapa referensi data tambahan seperti *transmitter power* yang dimana berguna untuk memberikan konstant *power* pada Wi-Fi.

```

// Log-normal shadowing model parameters
float referenceDistance = 1.0; // Referensi jarak dalam meters
float maxDistance = 60.0; // maximal jarak dalam meters
float RSSI_d0 = -20.0; // path loss RSSI dengan referensi jarak (d0) (PL(d0) = RSSI - 10 *
n * log10(d0))
// path loss RSSI pada jarak tertentu (PL(d) = PL(d0) + 10 * n * log10(d) + X)
float pathLossExponent = 2.7; // Path loss exponent ( n = (RSSI_1 - RSSI_2) / (10 *
log10(d_1 / d_2)) )
// Standard deviation di log-normal shadowing
float* stdev_2;
float* stdev_3;
float ptx_2 = 0, ptx_3 = 0; // Transmitter power
//float RSSI_d = -66.0;

float LNSM(float RSSI_d, float stdDev, float mean, float Ptx) {
  // Calculate distance using log-normal shadowing model
  float shadowing = lognormalShadowing(stdDev, mean);
  float estimateRSSI = (Ptx - (RSSI_d0 + 10 * pathLossExponent * log10(maxDistance /
referenceDistance) + shadowing));
  float formula = -((RSSI_d - (Ptx + RSSI_d0 - shadowing)) / (10 * pathLossExponent));
  Serial.print("Sigma Shadowing: ");
  Serial.print(shadowing);
  Serial.print(", Estimate: ");
  Serial.println(estimateRSSI);
  float distance = pow(10, formula) * referenceDistance;
  distance = (distance < 65) ? distance : 60;
  return distance;
}

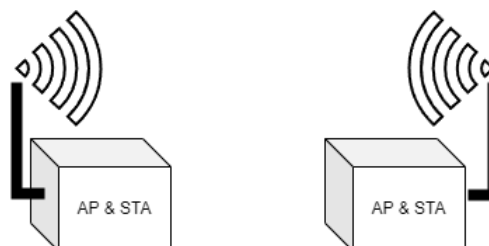
```

Gambar 3. 7 Program *Log-Normal Shadowing*

## 3.5 Protokol Komunikasi antar *Node*

### 3.5.1 Komunikasi Wi-Fi

Wi-Fi digunakan untuk komunikasi antar *node to node*. Dimana dalam fitur NodeMCU ESP32S terdapat *Multi Mode Access*. *Multi Mode Access* adalah mengaktifkan *station mode* dan *access point* secara bersamaan. Sehingga *Multi Mode Access* pada perangkat ESP32S bisa memancarkan jaringan dan mencari jaringan dalam 1 perangkat pada area disekitarnya, dengan cara melakukan *scan* atau pindai data dalam pancaran yaitu Channel & RSSI. Jika pancaran diterima dan saling mengenal, maka dengan mudah mengetahui informasi dari wifi pemancar tersebut.



Gambar 3. 8 Komunikasi Antar *Node*

Setiap *device* atau perangkat akan melakukan *scanning* secara bersamaan secara berkala. Dengan cara tersebut bisa menjadikan tiap *Node* saling mengenali satu sama lain. Untuk program *basic Multi Mode Access* dalam Wi-Fi bisa dilihat pada Lampiran 13. Untuk implementasi pada *Node* bisa dilihat pada Lampiran 10.

### 3.6 Skenario Pengambilan Data



Gambar 3. 9 Posisi *Node*

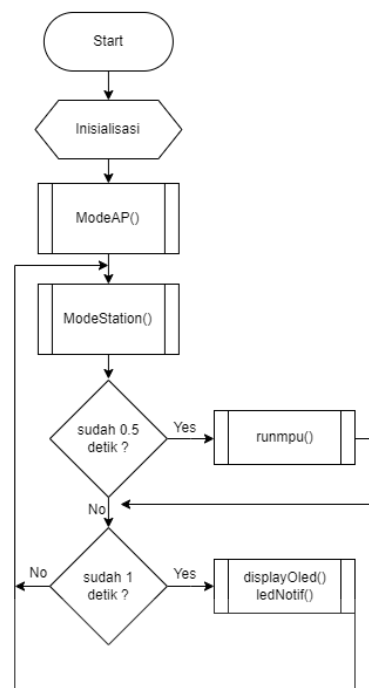
Untuk mendapatkan koefisien *path loss* ( $n$ ), maka sampel data diambil dengan jarak 1 meter sebagai referensi. Persamaan koefisien *path loss* ditunjukkan pada (8).

$$n = \frac{RSSI_{d0} - RSSI}{10 \log_{10}(d)} \quad (8)$$

Berdasarkan pengambilan data di depan bengkel robot Universitas Dinamika, terdapat rata-rata RSSI dengan jarak 1 meter sebesar -20 dBm. Sedangkan untuk RSSI jarak 60 meter sebesar -64 dBm. Selanjutnya dengan nilai RSSI yang telah didapat, maka nilai  $n$  didapatkan 2.59.

Pemindaian akan dilakukan secara berkala selama *Node* menyala untuk mendapatkan nilai RSSI secara *Real-time*. Setelah data RSSI didapat akan dikirim atau diolah dengan Rumus *Log-Normal Shadowing Distance* untuk merubah nilai RSSI ke jarak yang ada tiap *Node*-nya.

### 3.7 Algoritma Antar Node

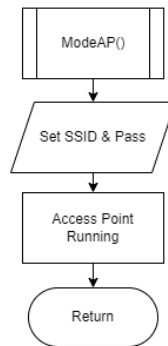


Gambar 3. 10 *Flowchart Global*

Pada gambar 3.10 adalah *flowchart global*. Saat mulainya program dilakukan inisialisasi variabel yang dibutuhkan. Setelah inisialisasi, dilanjutkan dengan menjalankan program *Mode Access Point* untuk memancarkan jaringan SSID. Kemudian dilanjutkan dengan menjalankan *Mode Station* sebagai penerima SSID. Setelah itu program akan melakukan pengecekan apakah waktu sudah berjalan selama 0.5 detik, jika ya maka program akan menjalankan program function *runmpu()* yang dimana pada *function* ini memanggil sebuah *library* perhitungan pengambilan data *accelerometer X* yang diubah menjadi *Horizontal Velocity* dengan nilai perubahan satuan km/jam dan hasilnya akan disimpan ke *variabel global getData*, kemudian melanjutkan ke pengecekan berikutnya apakah waktu sudah berjalan selama 1 detik. Jika tidak program akan melanjutkan untuk melakukan pengecekan selama 1 detik. Selanjutnya setelah program melakukan pengecekan apakah waktu sudah berjalan selama 1 detik, maka jika ya akan masuk program function *displayOled()* yang dimana digunakan untuk menampilkan data kecepatan dan jarak setiap 1 detik, Sedangkan *function ledNotif()* digunakan sebagai pengecekan atau aturan jarak aman yang akan menghasilkan *output led* dan

*buzzer* sebagai penanda . Jika tidak, maka program akan kembali mengulang sebuah program.

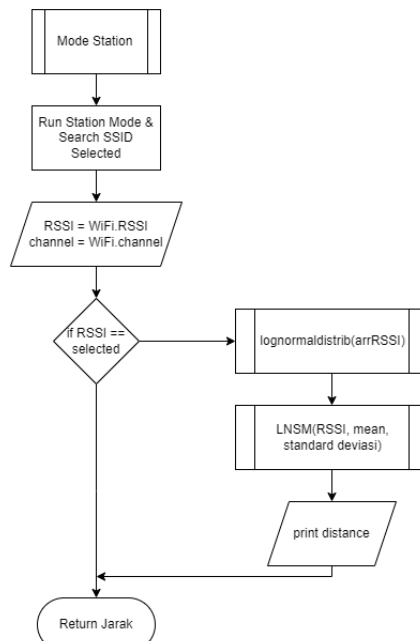
### 3.7.1 Fungsi *ModeAP*



Gambar 3. 11 *Flowchart ModeAP*

Pada Gambar 3.11, menjalankan pemanggilan *function ModeAP()* untuk mengaktifkan mode *Access Point* dan juga membuat *SSID* dan *PASS*. Kemudian untuk memancarkan Wi-Fi, maka dibuat proses berjalan selama alat dalam keadaan menyala. Setelah itu program akan kembali ke perintah utama.

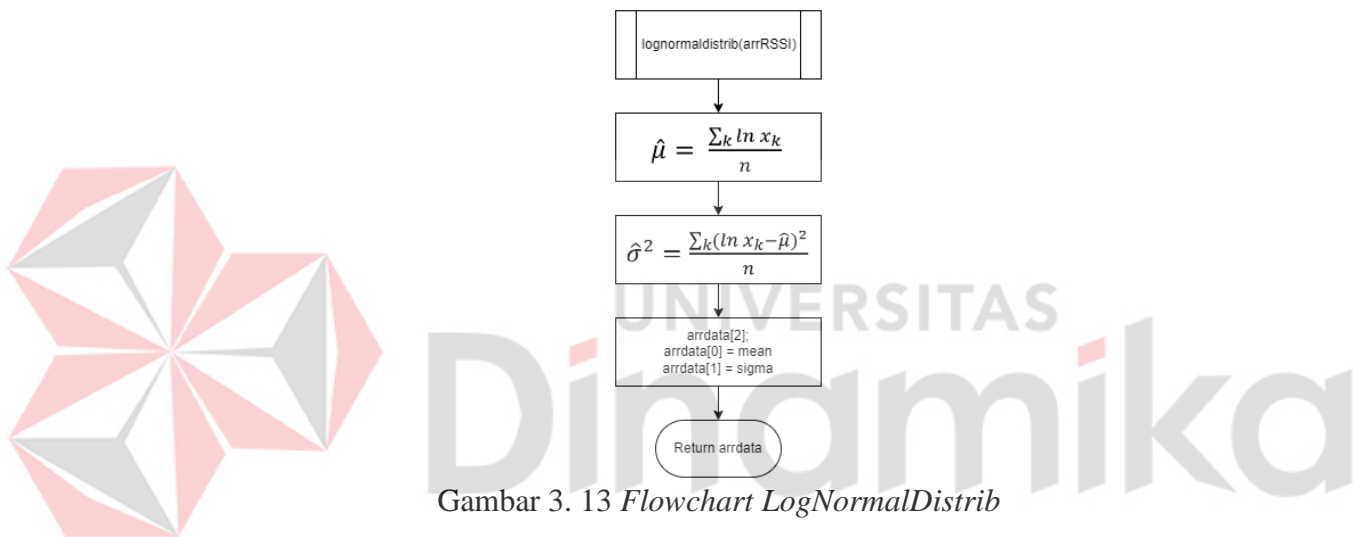
### 3.7.2 Fungsi *ModeStation*



Gambar 3. 12 *Flowchart ModeStation*

Pada Gambar 3.12, menjalankan sebuah *function ModeStation()* melakukan proses *scanning* Wi-Fi untuk mendapatkan SSID dari setiap *node*, selanjutnya masuk pada kondisi fungsi LogDistrib yang mendapatkan nilai *mean* dan *standard deviasi* dari sebuah data *RSSI*. Setelah selesai menjalankan fungsi LogDistrib maka akan menjalankan fungsi LNSM dengan parameter data mean, standard deviasi, dan *RSSI* untuk mendapatkan nilai shadowing pada jarak yang telah ditentukan serta prediksi jarak yang telah didapat.

### 3.7.3 Fungsi LogDistrib (*Log-Normal Distribution*)

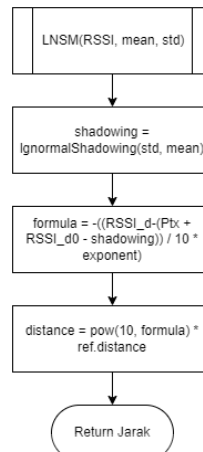


Gambar 3. 13 Flowchart LogNormalDistrib

Pada Gambar 3.13, merupakan fungsi *lognormaldistrib* yang berjalan pada sebuah *library LgNormShadow.h* yang dimana didalamnya menjalankan proses perhitungan untuk mendapatkan nilai “*mean*” dan “*standard deviasi*” yang juga diperlukan parameter data sebuah *RSSI* untuk mendapatkan hasilnya. Dari data *RSSI* dengan beberapa banyaknya data yang telah disimpan pada sebuah *array* maka akan melakukan perhitungan untuk mendapatkan “*mean*” dengan menjumlahkan seluruh data pada *array* dan dibagi dengan banyaknya data *array*, maka mendapatkan hasil *mean*, sedangkan untuk mendapatkan *nilai standard deviasi* maka menjalankan perhitungan dari tiap nilai data *RSSI* dikurangkan dengan *mean*, kemudian dipangkat kan 2 serta dibagi dengan banyak-nya data dari *RSSI*. Hasil tersebut yang kemudian akan diakar-kan menjadi sebuah nilai “*standard deviasi*”. Hasil dari nilai mean dan standard deviasi akan disimpan ke

sebuah *array* kembali yang nantinya akan dijadikan parameter pada *Log-Normal Shadowing Model*.

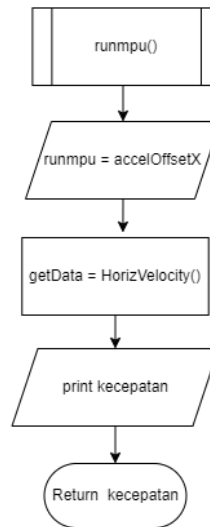
### 3.7.4 Fungsi LNSM(*Log-Normal Shadowing Model*)



Gambar 3. 14 *Flowchart* Fungsi LNSM

Pada Gambar 3.14, dijelaskan sebuah *function* LNSM yaitu untuk mendapatkan nilai *shadowing* yang dimana nantinya akan digunakan sebagai parameter perhitungan RSSI ke Jarak. Pertama menjalankan *function Header* program yang dimana *header* program ini adalah sebuah *library* yang diperoleh untuk mencari nilai *log-normal shadowing* yang fungsinya peredam gangguan, *noise* atau halangan yang didapatkan selama mengambil data RSSI yang akan dijadikan perubahan jarak dengan parameter tambahan yaitu *Standard Deviasi* (*sigma*) dan *mean*. Parameter tersebut didapatkan dari perhitungan sebelumnya yaitu masukkan data RSSI dengan jarak tertentu. Setelah mendapatkan nilai *shadowing*, masuk ke formula *log-normal distance* yang dimana rumusnya dengan parameter *transmitterpower* Wi-Fi 2 dBm. *Transmitterpower* ini didapat dengan setting program agar keluaran dari transmisi Wi-Fi ESP berjalan di 2 dBm menggunakan *source code library* pada ESP32. RSSI\_d0 yaitu -20.00 dari hasil *path loss* RSSI referensi dengan jarak 1 meter, RSSI\_d yaitu keluaran hasil *path loss* dari RSSI pada jarak tertentu, dan nilai *path loss exponent* (*n*) dengan nilai 2.7 yang didapat dari persamaan (8).

### 3.7.5 Fungsi runMPU



Gambar 3. 15 Flowchart Fungsi *runmpu()*

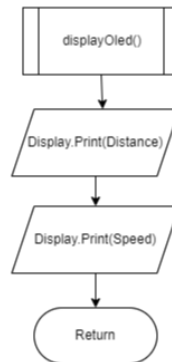
Pada Gambar 3.15, yaitu memanggil *function runmpu()* untuk menjalankan sensor MPU6050 agar mendapatkan nilai *accelerometer* dan *gyrometer*. Fungsi sensor MPU6050 sebagai prediksi sebuah kecepatan motor dengan memanfaatkan sebuah *library* luar bernama *HorizontalVelocity* dengan fungsi *HorizVelocity()* pada *header* program mengubah akselerasi menjadi kecepatan menggunakan data input nilai *acceleration x* dan *gyro x, y, serta z* sebagai parameternya. Kemudian hasil tersebut akan didapatkan nilai kecepatan dengan satuan m/s. untuk mendapatkan satuan km/jam pada persamaan (9).

$$km/j = \frac{1/1000}{1/3600} = 3.6 \quad (9)$$

Setelah itu untuk mengubah ke kecepatan maka *velocity \* 3.6*. Setelah mendapatkan hasil dari kecepatan, selanjutnya nilai kecepatan akan disimpan ke variabel global dan kembali lagi ke program utama.



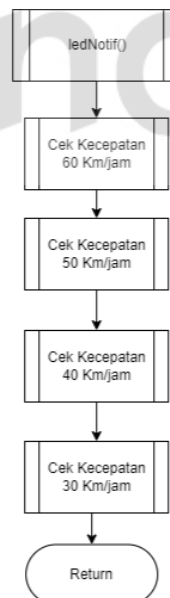
### 3.7.6 Fungsi *DisplayOled*



Gambar 3. 16 *Flowchart* Fungsi *DisplayOled*

Pada Gambar 3.16 menjelaskan sebuah *function displayOled()* yang gunanya sebagai menampilkan data yang didapatkan dari jarak dan kecepatan tiap node kemudian akan ditampilkan kesebuah *Display Oled*.

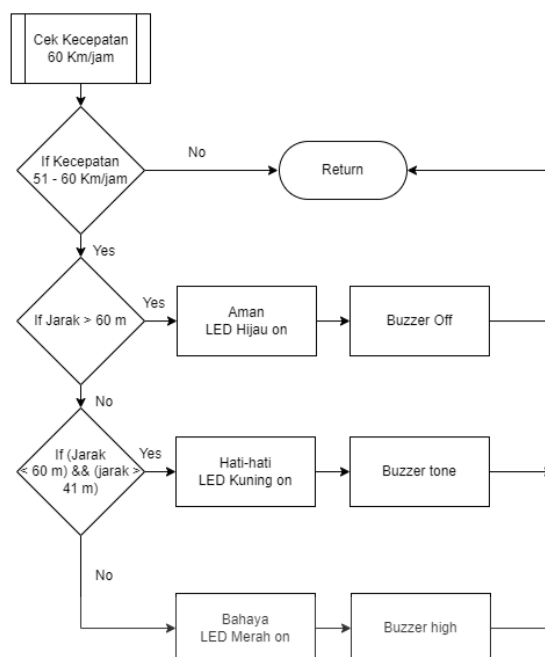
### 3.7.7 Fungsi *LedNotif*



Gambar 3. 17 *Flowchart* Fungsi *LedNotif()*

Pada Gambar 3.17 menjelaskan sebuah fungsi yang guna memanggil fungsi lain untuk mengecek jarak aman tiap nodenya serta mengeluarkan sebuah output atau notifikasi sebuah LED dan Buzzer sebagai peringatan atau pemberitahuan.

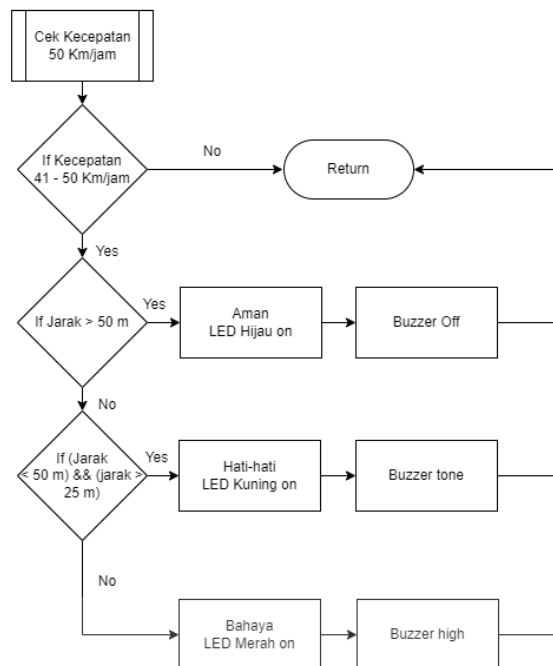
### 3.7.8 Fungsi CekKecepatan 60 km/jam



Gambar 3. 18 Flowchart Fungsi CekKecepatan 60 km/jam

Pada Gambar 3.18, merupakan *flowchart* untuk melakukan proses pengecekan kecepatan di 60 km/jam. Jika kecepatan diantara 51 sampai 60 km/jam terpenuhi, maka hasilnya ya, yang berarti program lanjut ke pengecekan berikutnya yaitu jarak. Jika tidak, maka program cek kecepatan 60 km/jam akan berakhir dan kembali ke program utama. Selanjutnya, jika dengan kecepatan 51 sampai 60 km/jam dengan jarak lebih dari 60 meter, maka LED hijau akan menyala yang menandakan kendaraan dalam keadaan jarak aman. Jika tidak, maka program akan lanjut ke cek kondisi apakah jarak kurang dari 60 meter dan lebih dari 41 meter. Jika ya, maka LED kuning akan menyala yang menandakan hati-hati, kemudian buzzer *tone* juga akan ikut menyala. Jika tidak atau jarak kurang dari 41 meter, maka LED merah akan menyala yang menandakan kecepatan kendaraan dalam keadaan bahaya dan buzzer *high* juga akan ikut menyala. Jika kondisi tersebut ada yang terpenuhi, secara otomatis program akan berakhir dan kembali ke program utama.

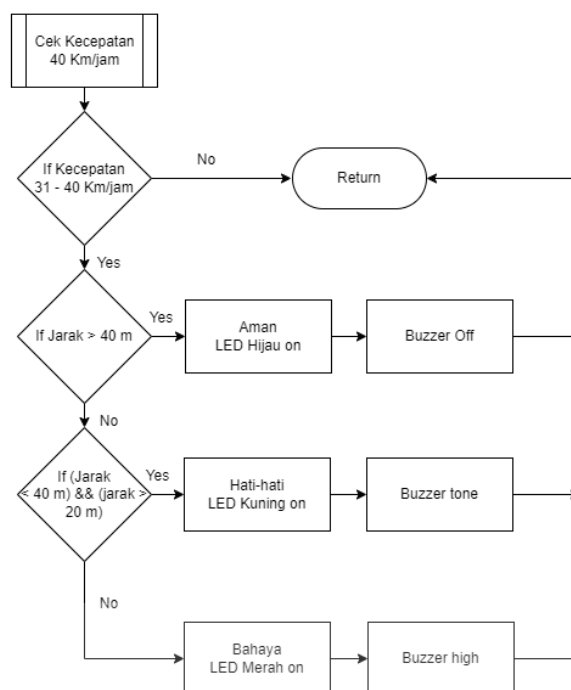
### 3.7.9 Fungsi CekKecepatan 50 km/jam



Gambar 3. 19 Flowchart Fungsi CekKecepatan 50 km/jam

Pada Gambar 3.19, merupakan *flowchart* untuk melakukan proses pengecekan kecepatan di 50 km/jam. Jika kecepatan diantara 41 sampai 50 km/jam terpenuhi, maka hasilnya ya, yang berarti program lanjut ke pengecekan berikutnya yaitu jarak. Jika tidak, maka program cek kecepatan 50 km/jam akan berakhir dan kembali ke program utama. Selanjutnya, jika dengan kecepatan 41 sampai 50 km/jam dengan jarak lebih dari 50 meter, maka LED hijau akan menyala yang menandakan kendaraan dalam keadaan jarak aman. Jika tidak, maka program akan lanjut ke cek kondisi apakah jarak kurang dari 50 meter dan lebih dari 25 meter. Jika ya, maka LED kuning akan menyala yang menandakan hati-hati, kemudian buzzer *tone* juga akan ikut menyala. Jika tidak atau jarak kurang dari 25 meter, maka LED merah akan menyala yang menandakan kecepatan kendaraan dalam keadaan bahaya dan buzzer *high* juga akan ikut menyala. Jika kondisi tersebut ada yang terpenuhi, secara otomatis program akan berakhir dan kembali ke program utama.

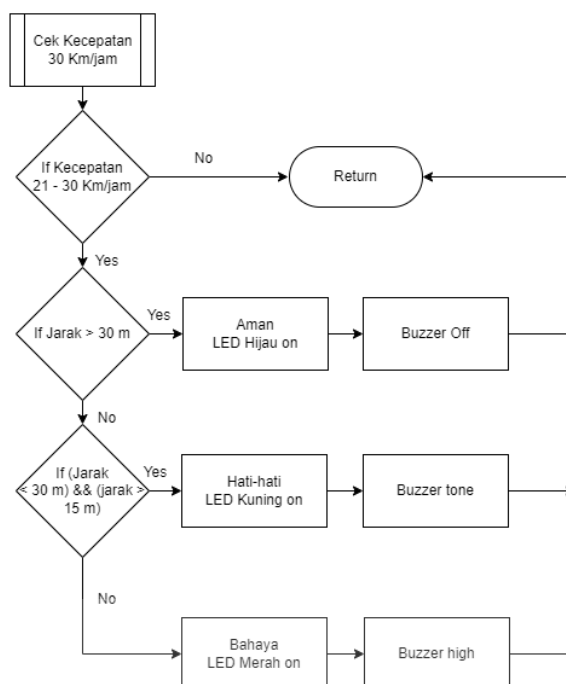
### 3.7.10 Fungsi CekKecepatan 40 km/jam



Gambar 3. 20 Flowchart Fungsi CekKecepatan 40 Km/jam

Pada Gambar 3.20, merupakan *flowchart* untuk melakukan proses pengecekan kecepatan di 40 km/jam. Jika kecepatan diantara 31 sampai 40 km/jam terpenuhi, maka hasilnya ya, yang berarti program lanjut ke pengecekan berikutnya yaitu jarak. Jika tidak, maka program cek kecepatan 30 km/jam akan berakhir dan kembali ke program utama. Selanjutnya, jika dengan kecepatan 31 sampai 40 km/jam dengan jarak lebih dari 40 meter, maka LED hijau akan menyala yang menandakan kendaraan dalam keadaan jarak aman. Jika tidak, maka program akan lanjut ke cek kondisi apakah jarak kurang dari 40 meter dan lebih dari 20 meter. Jika ya, maka LED kuning akan menyala yang menandakan hati-hati, kemudian buzzer *tone* juga akan ikut menyala. Jika tidak atau jarak kurang dari 20 meter, maka LED merah akan menyala yang menandakan kecepatan kendaraan dalam keadaan bahaya dan buzzer *high* juga akan ikut menyala. Jika kondisi tersebut ada yang terpenuhi, secara otomatis program akan berakhir dan kembali ke program utama.

### 3.7.11 Fungsi CekKecepatan 30 km/jam



Gambar 3. 21 Flowchart Fungsi CekKecepatan 30 Km/jam

Pada Gambar 3.21, merupakan *flowchart* untuk melakukan proses pengecekan kecepatan di 30 km/jam. Jika kecepatan diantara 21 sampai 30 km/jam terpenuhi, maka hasilnya ya, yang berarti program lanjut ke pengecekan berikutnya yaitu jarak. Jika tidak, maka program cek kecepatan 30 km/jam akan berakhir dan kembali ke program utama. Selanjutnya, jika dengan kecepatan 21 sampai 30 km/jam dengan jarak lebih dari 30 meter, maka LED hijau akan menyala yang menandakan kendaraan dalam keadaan jarak aman. Jika tidak, maka program akan lanjut ke cek kondisi apakah jarak kurang dari 30 meter dan lebih dari 15 meter. Jika ya, maka LED kuning akan menyala yang menandakan hati-hati, kemudian buzzer *tone* juga akan ikut menyala. Jika tidak atau jarak kurang dari 15 meter, maka LED merah akan menyala yang menandakan kecepatan kendaraan dalam keadaan bahaya dan buzzer *high* juga akan ikut menyala. Jika kondisi tersebut ada yang terpenuhi, secara otomatis program akan berakhir dan kembali ke program utama.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Perbandingan RSSI terhadap Jarak

#### 4.1.1 Tujuan Pengujian Perbandingan RSSI terhadap Jarak

Pada pengujian ini dilakukan untuk mendapatkan nilai RSSI sebagai perbandingan hasil nilai RSSI yang di dapat terhadap jarak yang sudah ditentukan selama 15 menit pada area sekitar Universitas Dinamika.

#### 4.1.2 Prosedur Pengujian Perbandingan RSSI terhadap Jarak

1. Pengujian ini dilakukan menggunakan 3 buah *node* dan 3 sepeda motor dalam keadaan diam.
2. Dilakukan pengujian sebanyak 7 kali berdasarkan jarak *real* 15 meter, 20 meter, 25 meter, 30 meter, 40 meter, 50 meter, dan 60 meter.
3. Setiap pengujian menggunakan waktu selama 15 menit.

#### 4.1.3 Hasil Pengujian Perbandingan RSSI terhadap Jarak

Berdasarkan hasil pengujian pada tiap jarak, didapatkan jumlah data sebanyak 250 data yang sudah diolah di dalam *excel* yang kemudian dilakukan rata-rata tiap RSSI-nya, data keseluruhan dapat dilihat pada Lampiran 1.

Tabel 4. 1 Pengujian Perubahan RSSI terhadap Jarak

| Jarak (meter) | Rata-rata RSSI (dBm) |
|---------------|----------------------|
| 15 meter      | -49,00               |
| 20 meter      | -52,02               |
| 25 meter      | -57,52               |
| 30 meter      | -58,52               |
| 40 meter      | -59,96               |
| 50 meter      | -62,43               |
| 60 meter      | -65,39               |

Dari hasil pengolahan data pada Tabel 4.1, diketahui pada jarak 15 meter mendapatkan hasil rata-rata RSSI -49,00 dBm, pada jarak 20 meter mendapatkan hasil rata-rata RSSI -52,02 dBm, pada jarak 25 meter mendapatkan hasil rata-rata

RSSI -57,52 dBm, pada jarak 30 meter mendapatkan hasil rata-rata RSSI -58,52 dBm, pada jarak 40 meter mendapatkan hasil rata-rata RSSI -59,96 dBm, pada jarak 50 meter mendapatkan hasil rata-rata RSSI -62,43 dBm, dan pada jarak 60 meter mendapatkan hasil rata-rata RSSI -65,39 dBm.

## 4.2 Pengujian Perubahan RSSI ke Jarak menggunakan LNSM

### 4.2.1 Tujuan Pengujian Perubahan RSSI ke Jarak menggunakan LNSM

Pada pengujian ini dilakukan perubahan pada hasil nilai RSSI yang didapat ke dalam rumus *Log-normal shadowing* untuk diubah ke satuan jarak. Tujuannya untuk mendapatkan nilai jarak sesuai dengan yang sudah ditentukan.

### 4.2.2 Prosedur Pengujian Perubahan RSSI ke Jarak menggunakan LNSM

1. Pengujian ini dilakukan menggunakan 3 buah *node* dan 3 sepeda motor dalam keadaan diam.
2. Dilakukan pengujian sebanyak 7 kali berdasarkan jarak *real* 15 meter, 20 meter, 25 meter, 30 meter, 40 meter, 50 meter, dan 60 meter.
3. Pengujian ini dilakukan selama 15 menit dan menghasilkan 250 data.
4. Pengujian ini menggunakan rumus *Log-normal shadowing* untuk mengubah nilai RSSI ke dalam satuan Jarak.

### 4.2.3 Hasil Pengujian Perubahan RSSI ke Jarak menggunakan LNSM

Jarak rata-rata merupakan jarak yang sudah diolah dalam *excel*. Data keseluruhan bisa dilihat pada Lampiran 1. Untuk mendapatkan nilai *error* dari jarak *real* dengan jarak rata-rata dari RSSI, maka digunakan persamaan (10).

$$Error = \left| \frac{\text{jarak real} - \text{jarak RSSI}}{\text{Jarak real}} \right| \times 100\% \quad (10)$$

Tabel 4. 2 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 15 meter

|                            | N1-N2 | N1-N3 | N3-N1 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 14,20 | 13,98 | 15,05 |
| Rata-rata <i>error</i> (%) | 19,26 | 19,16 | 22,84 |

Hasil pengujian pada jarak 15 meter didapatkan rata-rata *error* sebesar 20,42%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* besar

dikarenakan sinyal antenna pada tiap *node* pada saat pengambilan terdapat banyak gangguan seperti *noise*, halangan dan interferensi yang membuat perubahan RSSI cukup besar.

Tabel 4. 3 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 20 meter

|                            | N1-N2 | N1-N3 | N3-N1 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 18,11 | 18,08 | 17,74 |
| Rata-rata <i>error</i> (%) | 12,48 | 12,78 | 14,24 |

Hasil pengujian pada jarak 20 meter didapatkan rata-rata *error* sebesar 13,17%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* lebih rendah dari pengukuran jarak 15 meter dikarenakan sinyal antenna pada tiap *node* yang menjangkau area lingkungan terdapat minim *noise*, halangan dan interferensi.

Tabel 4. 4 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 25 meter

|                            | N2-N1 | N2-N3 | N3-N2 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 28,59 | 28,43 | 28,90 |
| Rata-rata <i>error</i> (%) | 22,67 | 19,11 | 21,19 |

Hasil pengujian pada jarak 25 meter didapatkan rata-rata *error* sebesar 20,99%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* besar dikarenakan kondisi lingkungan yang terdapat *noise*, halangan, dan interferensi yang besar.

Tabel 4. 5 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 30 meter

|                            | N3-N1 | N3-N2 | N2-N3 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 31,01 | 31,30 | 30,92 |
| Rata-rata <i>error</i> (%) | 16,06 | 13,90 | 15,68 |

Hasil pengujian pada jarak 30 meter didapatkan rata-rata *error* sebesar 15,21%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* rendah karena sinyal antenna pada tiap *node* yang menjangkau area lingkungan terdapat minim *noise*, halangan dan interferensi.

Tabel 4. 6 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 40 meter

|                            | N2-N1 | N2-N3 | N3-N2 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 34,93 | 35,35 | 38,47 |
| Rata-rata <i>error</i> (%) | 14,98 | 14,32 | 15,46 |



Hasil pengujian pada jarak 40 meter didapatkan rata-rata *error* sebesar 14,92%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* rendah karena sinyal antenna pada tiap node yang menjangkau area lingkungan terdapat minim noise, halangan dan interferensi.

Tabel 4. 7 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 50 meter

|                            | N1-N2 | N1-N3 | N3-N1 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 43,09 | 43,61 | 44,86 |
| Rata-rata <i>error</i> (%) | 20,25 | 19,81 | 17,86 |

Hasil pengujian pada jarak 50 meter didapatkan rata-rata *error* sebesar 19,31%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* lebih besar dari yang sebelumnya dikarenakan sinyal antenna pada tiap *node* terdapat banyak gangguan seperti noise, halangan dan interferensi yang membuat perubahan RSSI.

Tabel 4. 8 Perubahan Jarak *Real* ke Jarak Hasil RSSI pada jarak 60 meter

|                            | N1-N2 | N1-N3 | N3-N1 |
|----------------------------|-------|-------|-------|
| Rata-rata Jarak RSSI       | 50,85 | 51,76 | 51,72 |
| Rata-rata <i>error</i> (%) | 15,25 | 13,73 | 13,88 |

Hasil pengujian pada jarak 60 meter didapatkan rata-rata *error* sebesar 14,29%. Berdasarkan hasil tersebut, disimpulkan bahwa nilai rata-rata *error* menurun karena sinyal antenna pada tiap *node* diberikan batasan hingga 60 meter yang membuat nilai *error* menjadi mengecil.

Untuk mengubah nilai RSSI ke bentuk jarak, menggunakan rumus *Log-Normal Shadowing* pada persamaan (11).

$$d = d_0 10^{-((RSSI - P_T + PL_0 - X\sigma)/10n)} \quad (11)$$

### 4.3 Pengujian Kecepatan

#### 4.3.1 Tujuan Pengujian Kecepatan

Tujuan pengujian ini untuk menguji keakuratan sensor MPU-6050. Pengujian ini dilakukan untuk mengetahui perbandingan kecepatan pada *accelerometer* yang

telah di konversi menjadi kecepatan menggunakan *HorizontalVelocity* dan kecepatan *real*.

### 4.3.2 Prosedur Pengujian Kecepatan

1. Pengujian menggunakan 3 buah *node* dan 1 sepeda motor, pengujian dilakukan selama 10 menit.
2. Pengujian dilakukan di area Universitas Dinamika.
3. Melakukan konversi kecepatan pada *accelerometer* menjadi kecepatan menggunakan *HorizontalVelocity*.
4. Pengujian kecepatan dilakukan hanya pada kecepatan 21 km/jam hingga 42km/jam.

### 4.3.3 Hasil Pengujian Kecepatan

Berdasarkan hasil uji kecepatan *accelerometer*, pada kecepatan dengan *range* 21 km/jam sampai 30 km/jam diperoleh rata-rata 24,45 km/jam yang artinya mendekati kecepatan *real*. Pada kecepatan dengan *range* 31 km/jam sampai 40 km/jam diperoleh rata-rata 30,48 km/jam yang berarti mendekati kecepatan *real*. Pada kecepatan dengan *range* 41 km/jam sampai 50 km/jam diperoleh rata-rata 41,15 km/jam yang berarti juga mendekati jarak *real*. Hasil pengujian kecepatan dapat dilihat pada Lampiran 8.

Tabel 4. 9 Pengujian Prediksi Kecepatan Menggunakan Sensor

| Kecepatan <i>real</i><br>(km/jam) | Kecepatan MPU-6050<br>(km/jam) |    |    | <i>Error</i> (%) |      |      |
|-----------------------------------|--------------------------------|----|----|------------------|------|------|
|                                   | N1                             | N2 | N3 | N1               | N2   | N3   |
| 21-30                             | 26                             | 27 | 23 | 4,0              | 8,0  | 8,0  |
|                                   | 23                             | 24 | 23 | 8,0              | 4,0  | 8,0  |
| 31-40                             | 27                             | 31 | 31 | 10,0             | 3,3  | 3,0  |
|                                   | 33                             | 35 | 34 | 3,3              | 16,7 | 13,3 |
| 41-50                             | 41                             | 45 | 44 | 2,5              | 7,5  | 12,5 |
|                                   | 42                             | 37 | 49 | 5,0              | 5,0  | 7,5  |

Untuk mendapatkan nilai *error* dari hasil kecepatan dengan perbandingan kecepatan *real* dengan kecepatan *accelerometer*, maka didapatkan rumus pada persamaan (8).

$$Error = \left| \frac{Kec.Real - Kec.Accelero}{Kec.Real} \right| \times 100\% \quad (12)$$

#### 4.4 Pengujian Keseluruhan *Node*

##### 4.4.1 Tujuan Pengujian Keseluruhan *Node*

Tujuan pengujian ini untuk menguji keseluruhan alat apakah sudah bekerja sesuai ketentuan. Pengujian ini juga dilakukan untuk mengetahui jumlah *error* yang dihasilkan alat.

##### 4.4.2 Prosedur Pengujian Keseluruhan *Node*

1. Pengujian dilakukan di area Universitas Dinamika, dan dilakukan selama 10 menit.
2. Pengujian dilakukan menggunakan 3 buah *node* dan 2 buah motor.
3. Pengujian *node* dilakukan secara bergantian yaitu N1-N2, N1-N3, dan N2-N3.

##### 4.4.3 Hasil Pengujian Keseluruhan *Node*

Pengujian ini hanya dilakukan pada kecepatan 21 Km/jam hingga 30 Km/jam.

Hasil uji keseluruhan dapat dilihat pada Lampiran 9.

Tabel 4. 10 Pengujian Keseluruhan Alat Pada Kecepatan 21-30 Km/jam

| Jarak RSSI |       |       | Kecepatan MPU-6050 (km/jam) |       |       | Output |     |        |       |     |        |       |     |        |
|------------|-------|-------|-----------------------------|-------|-------|--------|-----|--------|-------|-----|--------|-------|-----|--------|
|            |       |       |                             |       |       | N1-N2  |     |        | N1-N3 |     |        | N2-N3 |     |        |
| N1-N2      | N1-N3 | N2-N3 | N1-N2                       | N1-N3 | N2-N3 | LCD    | LED | Buzzer | LCD   | LED | Buzzer | LCD   | LED | Buzzer |
| 25         | 13    | 35    | 25                          | 23    | 24    | T      | K   | L      | T     | M   | H      | T     | H   | O      |
| 33         | 12    | 35    | 24                          | 25    | 24    | T      | K   | L      | T     | M   | H      | T     | H   | O      |
| 13         | 26    | 30    | 23                          | 27    | 27    | T      | M   | H      | T     | K   | L      | T     | K   | L      |
| 20         | 23    | 12    | 27                          | 22    | 24    | T      | K   | L      | T     | K   | L      | T     | M   | H      |
| 22         | 20    | 11    | 27                          | 25    | 26    | T      | K   | L      | T     | K   | L      | T     | M   | H      |
| 19         | 15    | 29    | 25                          | 23    | 25    | T      | K   | L      | T     | M   | H      | T     | K   | L      |

Ket :

LCD : T = Tampil

LED : K = Kuning, M = Merah, H = Hijau

Buzzer : H = High, L = Low, O = Off

Hasil pengujian menunjukkan bahwa pada jarak RSSI 11-15 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED merah menyala, buzzer berbunyi menandakan bahwa jarak melebihi batas aman, kemudian pada jarak RSSI 16-30 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED kuning menyala, buzzer berbunyi dengan jeda menandakan bahwa jarak masuk peringatan batas aman, sedangkan

pada Jarak RSSI diatas 30 meter dengan kecepatan 21- 30 km/jam mendapatkan hasil keluaran LCD menyala, LED hijau menyala, dan buzzer mati menandakan bahwa jarak sangat aman. Maka dari hasil keseluruhan pengujian disimpulkan alur kerja pada alat berjalan dengan baik.



UNIVERSITAS  
**Dinamika**

## BAB V PENUTUP

### 5.1 Kesimpulan

Dari hasil beberapa pengambilan data selama pengujian, didapatkan beberapa kesimpulan diantaranya:

1. Konversi jarak menggunakan *log-normal shadowing model* dengan memanfaatkan RSSI Wi-Fi sebagai parameter pengukuran pada area *outdoor* mendapatkan rata-rata *error* terkecil sebesar 13,17% pada jarak 20 meter. Hasil tersebut didapat karena ketika pengambilan data pada area lingkungan *outdoor* dalam kondisi minim halangan dan gangguan seperti angin. Sedangkan ketika kondisi area lingkungan *outdoor* dalam kondisi banyak halangan dan gangguan, maka nilai RSSI yang didapat akan berubah.
2. Konversi perubahan dari sensor MPU-6050 dari akselerasi ke kecepatan menggunakan *library* mendapatkan nilai hasil uji coba saat kendaraan dalam kecepatan *real* 21-30 km/jam menghasilkan nilai *error* 6,73%, dan dalam kecepatan *real* 31-40 km/jam menghasilkan nilai *error* 8,41%, sedangkan untuk kecepatan *real* 41-50 km/jam menghasilkan nilai *error* 7,13%.
3. Dalam pengujian alat keseluruhan, pada jarak RSSI 11-15 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED merah menyala, buzzer berbunyi menandakan bahwa jarak melebihi batas aman, kemudian pada jarak RSSI 16-30 meter dengan kecepatan 21-30 km/jam mendapatkan hasil keluaran LCD menyala, LED kuning menyala, buzzer berbunyi dengan jeda menandakan bahwa jarak masuk peringatan batas aman, sedangkan pada Jarak RSSI diatas 30 meter dengan kecepatan 21- 30 km/jam mendapatkan hasil keluaran LCD menyala, LED hijau menyala, dan buzzer mati menandakan bahwa jarak sangat aman.

## 5.2 Saran

Adapun saran untuk mengembangkan penelitian ini agar lebih baik, berikut beberapa saran yaitu:

1. Mengganti sensor MPU-6050 dengan sensor *rotary encoder* ke motor, karena sensor MPU-6050 kurang cocok untuk mendeteksi dan prediksi kecepatan.



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- IndonesiaPemerintah. (1993). Peraturan Pemerintah Republik Indonesia Nomor 43 Tahun 1993 Tentang Prasarana dan Lalu Lintas Jalan. *Peraturan Pemerintah Republik Indonesia*, 78.
- Isni, R. N., Santosa, B., & Simanjuntak, O. S. (2020). E-Toll Collection Pada Sistem Transaksi Tol Tertutup Dengan Metode Layanan Berbasis Lokasi Studi Kasus Pt Jasa Marga (Persero), Tbk. *Telematika*, 17(2), 131. <https://doi.org/10.31315/telematika.v1i1.3382>
- Kho, Dickson. *Teknik Elektronika*. 2014. <https://teknikelektronika.com/pengertian-led-light-emitting-diode-cara-kerja/>. Diakses tanggal 9 Maret, 2023.
- Kompas.com. 2020. *Internet*. Paling Sering Dilupakan, Menjaga Jarak Aman Pengendara Motor. pada <https://otomotif.kompas.com/read/2020/11/04/102200315/paling-seringdilupakan-menjaga-jarak-aman-pengendara-motor>. Diakses tanggal 9 maret 2023.
- Mangkusasmito, F., Tadeus, D. Y., Winarno, H., & Winarno, E. (2020). Peningkatan Akurasi Sensor GY-521 MPU-6050 dengan Metode Koreksi Faktor Drift. *Ultima Computing : Jurnal Sistem Komputer*, 12(2), 91–95. <https://doi.org/10.31937/sk.v12i2.1791>
- Musdalifa, M., & Panu, S. A. (2019). Perancangan Jaringan Wifi Dengan Menggunakan Mikrotik Pada SMP Negeri 3 Mallusetasi Kabupaten Barru. *Publikasi Pendidikan*, 9(1), 28. <https://doi.org/10.26858/publikan.v9i1.7505>
- Natakusuma, Bagoes Prawira. (2018). *Inertia Measurement Unit (IMU) Sensor Application To Improve Walking Straight Motion On Quadruped Robot*. Thesis, Surabaya: Institut Teknologi Sepuluh Nopember Surabaya.
- Pavlovic, Maja. *Log-normal Distribution-A simple explanation*. February 16, 2022. <https://towardsdatascience.com/log-normal-distribution-a-simple-explanation-7605864fb67c>. Diakses tanggal 16 Juli, 2023.
- Prastyo, Elga Aris. *Edukasi Elektronika*. November 2022. <https://www.edukasielektronika.com/2022/10/pengertian-dan-penjelasan-tentang.html>. Diakses tanggal 9 Maret, 2023.
- Pusvita, W. Y., & Huda, Y. (2019). ANALISIS KUALITAS LAYANAN JARINGAN INTERNET WIFI.ID MENGGUNAKAN PARAMETER QOS (Quality Of Service). *Voteteknika (Vocational Teknik Elektronika Dan Informatika)*, 7(1), 54. <https://doi.org/10.24036/voteteknika.v7i1.103643>

Safiro, G. A. (2022). *TA: Rancang Bangun Simulasi Alat Pendeteksi Jarak Aman Antar Kendaraan Menggunakan Sensor Ultrasonik Berbasis Internet of Things*. <https://repository.dinamika.ac.id/id/eprint/6219/>

Suartawan, P. E., Fitriani, D., Aryuni, M., Rifai, R., & Sasue, O. (2021). *Pengembangan Sistem Lidar Pendeteksi Jarak Aman*.

Suarasurabaya.net. 2023. *Internet*. Faktor Manusia Dominasi Penyebab Kecelakaan di Sidoarjo. <https://www.suarasurabaya.net/kelanakota/2023/faktor-manusia-dominasipenyebab-kecelakaan-di-sidoarjo-polisi-minta-patuhi-aturan-lalu-lintas/>. Diakses tanggal 2 maret 2023.

Syahriza, M. (2019). Kecelakaan Lalulintas : Perlukah Mendapatkan Perhatian Khusus? *AVERROUS: Jurnal Kedokteran Dan Kesehatan Malikussaleh*, 5(2), 89. <https://doi.org/10.29103/averrous.v5i2.2083>

Yanziah, A., Soim, S., & Rose, M. M. (2020). Analisis Jarak Jangkauan Lora Dengan Parameter Rssi Dan Packet Loss Pada Area Urban. *Jurnal Teknologi Technosciantia*, 13(1), 27–34.

Yogie, S., & Habi, A. (2019). *Kontrol Dan Monitoring Fiting Lampu Berbasis Android*. <http://repository.polman-babel.ac.id/id/eprint/162/1/HABI & YOGIE.pdf>



UNIVERSITAS  
**Dinamika**