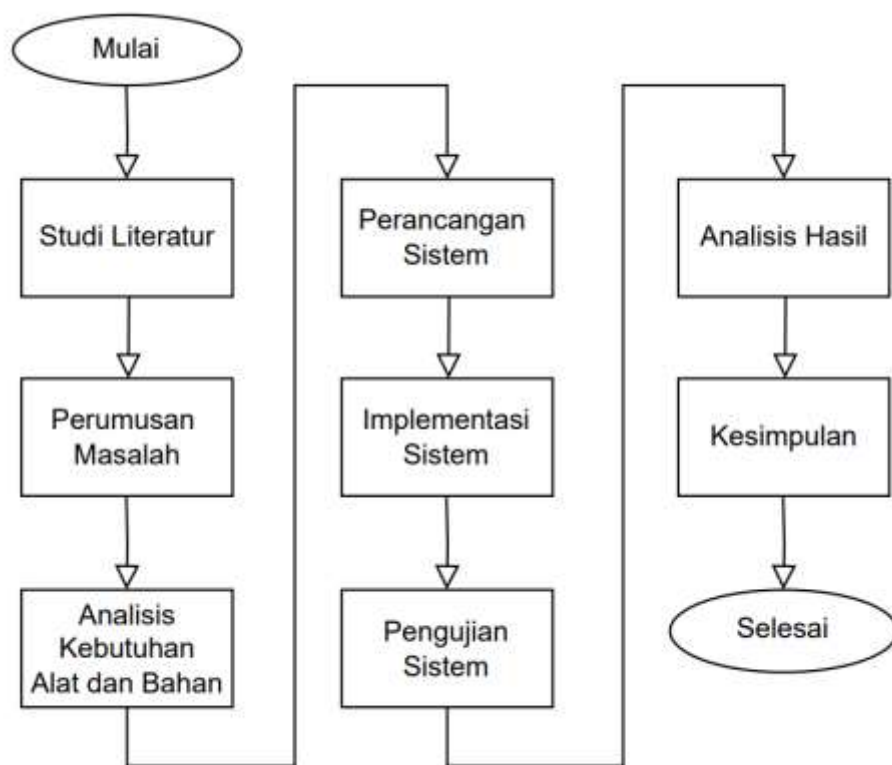


BAB III

METODE PENELITIAN

3.1 Flowchart Penelitian

Dalam melaksanakan perancangan sistem ini dilakukan dalam beberapa tahap, diantaranya seperti yang tertera pada flowchat dibawah ini.



Gambar 3. 1 Flowchart Penelitian

Berdasarkan Gambar 3.1 pada penelitian ini terdapat beberapa tahap, diantaranya Studi Literatur, Perumusan Masalah, Analisis Kebutuhan Alat dan Bahan, Perancangan Sistem, Implementasi Sistem, Pengujian Sistem, Analisis Hasil dan Kesimpulan.

3.1.1 Studi literatur

Pada tahapan ini, dilakukan pengumpulan referensi dari berbagai sumber yang relevan, seperti buku, jurnal ilmiah, dan artikel dari internet. Referensi ini digunakan sebagai dasar teori yang mendukung penelitian serta untuk memahami konsep-konsep terkait, seperti sistem OBD-II, sensor akselerometer, GPS, dan perilaku pengemudi. Informasi yang diperoleh dari studi literatur ini akan menjadi landasan untuk merancang sistem monitoring kendaraan dan menganalisis data yang dikumpulkan selama penelitian.

3.1.2 Perumusan Masalah

Pertama yang dilakukan adalah melakukan identifikasi masalah yang ada pada karakter pengemudi terutama yang berkaitan dengan kecepatan berdasarkan aturan yang ditetapkan pemerintah dan produsen kendaraan. Penelitian ini akan mengumpulkan data dari OBD-II untuk menganalisis perilaku pengemudi, dan menganalisis perilaku pengemudi dan mengidentifikasi ketika terjadi kecelakaan dengan sensor GPS dan MPU6050.

3.2 Alat dan Bahan

Pada proses penelitian "Implementasi Sistem Monitoring Mobil Berbasis OBD–II Untuk Deteksi Kecelakaan Dan Perilaku Pengemudi” menggunakan beberapa alat dan bahan untuk merancang perangkat keras dan perangkat lunak. Berikut merupakan penjabaran fungsi alat dan bahan yang digunakan berdasarkan jenis perangkatnya:

3.2.1 Perangkat Keras

Dalam analisis kebutuhan perangkat keras merupakan suatu hal yang penting sebelum melakukan perakitan sistem, karena berpengaruh terhadap proses perancangan sebuah alat. Berikut merupakan penjelasan perangkat keras yang digunakan:

1. Laptop

Perangkat yang digunakan untuk pembuatan sistem dengan spesifikasi menggunakan prosesor Intel Core I3 dengan memori penyimpanan sementara 12 GB.

2. Raspberry Pi 3 Model B

Sebagai *Hardware* atau alat untuk mengontrol semua sistem yang berjalan.

3. On-Board Diagnostics (OBD-II)

Modul yang berfungsi untuk memantau kinerja sensor-sensor pada kendaraan.

4. MPU6050

Modul yang terdapat 2 gabungan *Accelerometer*, *gyroscope*, dan *magnetometer*.

5. GPS Ublox Neo-6M

Modul untuk mendeteksi lokasi pengguna di permukaan bumi secara global berbasis satelit.

6. Kabel Jumper

Kabel jumper digunakan untuk menyambungkan koneksi antar

komponen pada sistem.

7. Bluetooth Dongle

Berfungsi sebagai penerima data dari OBD-II yang dikirim ke Raspberry Pi 3B.

8. Modem Wifi

Modem wifi berfungsi sebagai penghubung antara sistem monitoring kendaraan dengan jaringan internet.

3.2.2 Perangkat Lunak

Pada penelitian ini perangkat lunak yang digunakan adalah software Thonny Python IDE. Thonny Python IDE merupakan software editor teks pada Raspberry pi yang bisa menulis dan menjalankan kode python yang digunakan untuk mengembangkan logika sistem dan mengolah data dari sensor. Perangkat lunak ini menggunakan protokol HTTP untuk komunikasi antar perangkat dan server dan penambahan beberapa library python. Selain itu, beberapa perangkat lunak pendukung juga dimanfaatkan dalam pengembangan sistem, di antaranya Laragon sebagai server lokal untuk pengembangan *backend*, Postman untuk pengujian antarmuka API, serta Visual Studio Code (VS Code) untuk pembuatan dan pengelolaan tampilan website.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap paling awal dalam pembuatan sistem. Gambaran pelayanan sistem, batasan sistem, *input* dan output sistem. Pada sistem ini terdapat dua bagian yaitu bagian yang

dilakukan oleh pengguna (user), dan bagian yang dilakukan secara otomatis oleh sistem yang dirancang. Dari hasil studi literatur yang telah dilakukan oleh penulis, maka fungsi yang dapat dilakukan oleh pengguna antara lain:

1. Menghubungkan OBD-II pada mobil.
2. Menghubungkan Raspberry pi 3 ke Laptop.
3. Membangun program Python pada Raspberry pi 3 Model B.
4. Mengaktifkan sistem dengan mengkoneksikan OBD-II ke Raspberry pi 3 Model B dengan Bluetooth dan menjalankan program Python pada terminal Raspberry pi 3 Model B.
5. Memonitoring data Kecepatan, putaran mesin dan beban mesin pada terminal Raspberry pi 3 Model B.
6. Memonitoring data posisi dari GPS, Roll dan Pitch dari MPU6050 pada terminal Raspberry pi.

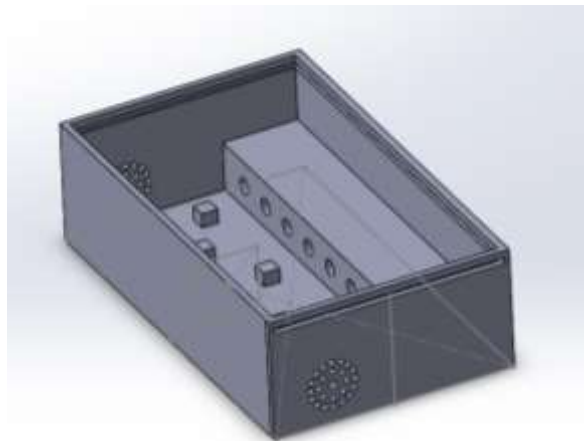
Adapun fungsi yang dilakukan oleh sistem diantaranya sebagai berikut:

1. Membaca data kecepatan dalam satuan km/h, putaran mesin dalam satuan RPM, serta beban mesin (*engine load*) dalam satuan persen (%) dari modul OBD-II yang terhubung dengan ECU kendaraan.
2. Membaca data Raw dari sensor pada modul MPU6050 yang kemudian dikonversi dalam bentuk derajat atau kemiringan terhadap sumbu x, sumbu y dan sumbu z. Selain itu pembacaan menjadi nilai akselerometer dalam satuan m/s^2 atau dengan nilai G.

3. Membaca data lokasi dari sensor GPS Neoblock 6M yang memberikan informasi mengenai posisi geografis kendaraan dalam bentuk koordinat lintang dan bujur (*latitude* dan *longitude*).
4. Mengirimkan paket data ke *cloud* pada interval tertentu dengan menggunakan protokol HTTP.

3.4 Perancangan Sistem

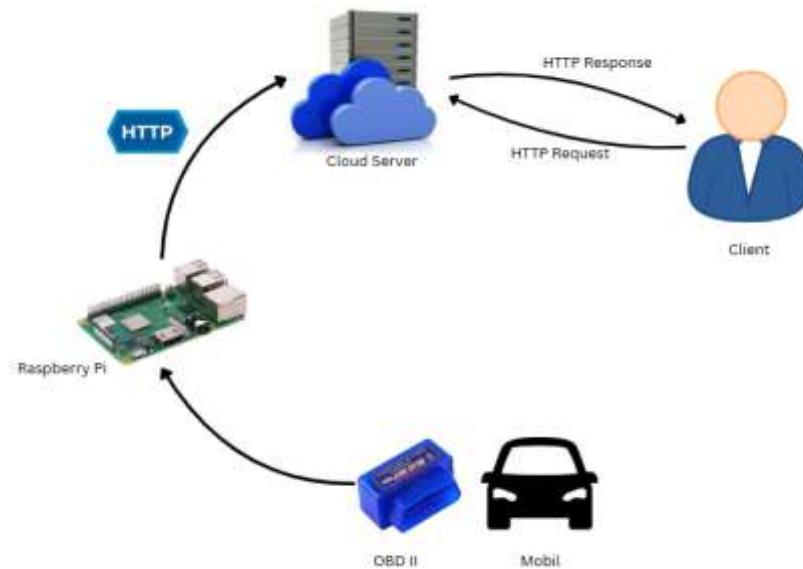
3.4.1 Desain Mekanik



Gambar 3.2 Desain Mekanik

Pada tahap ini, perancangan mekanik dilakukan untuk membuat desain kotak (box) penyimpanan yang mampu menampung dan melindungi beberapa komponen utama sistem, yaitu Raspberry Pi 3B, modem Wi-Fi, sensor MPU6050, sensor GPS, serta kipas pendingin. Desain kotak ini dirancang menggunakan perangkat lunak SolidWorks.

3.4.2 Desain Sistem

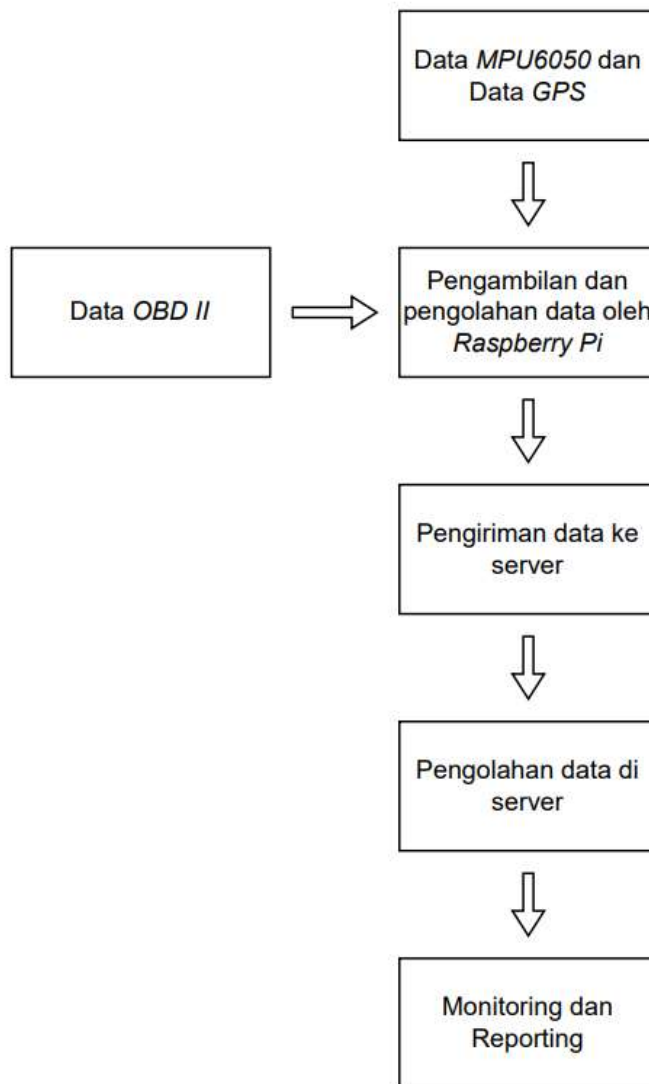


Gambar 3.3 Desain Sistem

Modul OBD-II terhubung langsung dengan ECU kendaraan, OBD-II ini dilengkapi Bluetooth untuk terkoneksi dengan Raspberry Pi 3 Model B menggunakan protokol sesuai dengan tipe kendaraan. Sensor *Accelerometer* dan gyroskop pada modul MPU6050 terhubung langsung dengan Raspberry Pi 3. Modul GPS Ublox Neo-6M Terhubung langsung dengan Raspberry Pi 3 untuk memberikan informasi titik koordinat posisi kendaraan. Modul OBD-II untuk mengambil data kecepatan, putaran mesin dan beban mesin dari kendaraan. Modul sensor MPU6050 untuk mendeteksi nilai kemiringan atau data *Accelerometer* dan *Gyroscope*. Data ECU dan posisi secara langsung dikirim ke cloud sedangkan data untuk deteksi kecelakaan dari MPU6050 diolah di Raspberry Pi 3 sebelum di kirim ke cloud. Dengan demikian sistem dapat memonitoring kondisi kendaraan dan

menyimpulkan karakteristik pengemudi berdasarkan tingkat kesalahan pengemudi terhadap batasan kecepatan, putaran mesin dan beban mesin.

3.4.3 Blok Diagram Sistem Keseluruhan

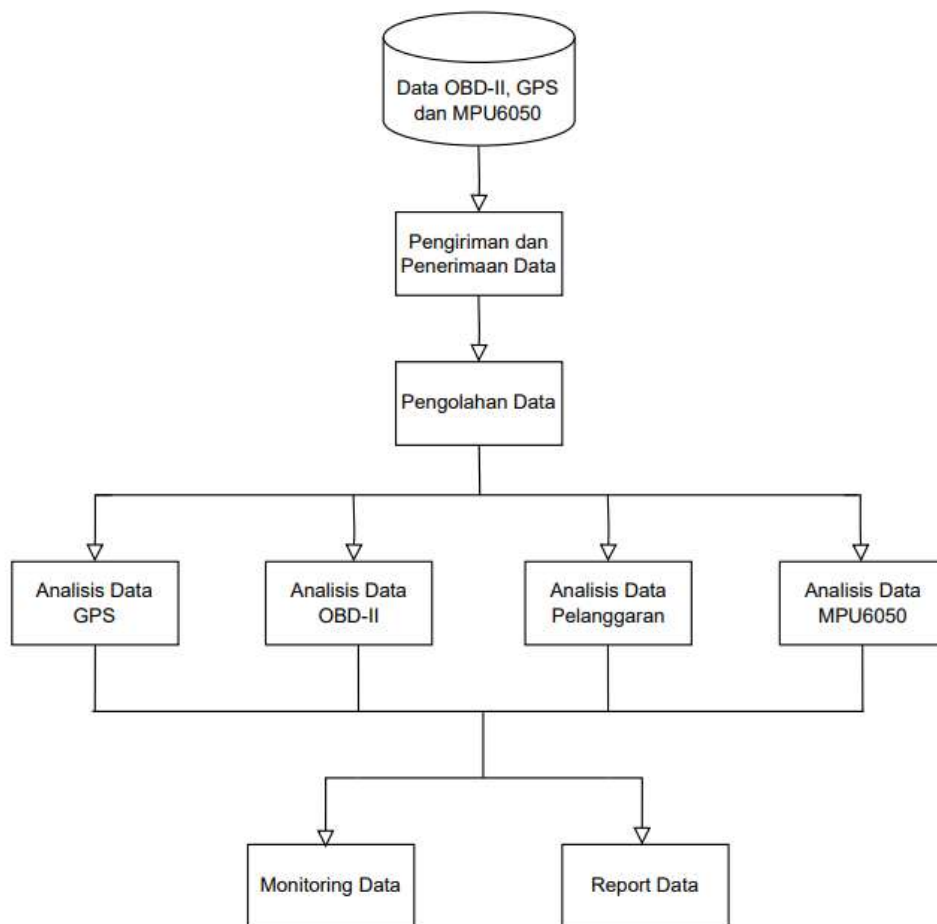


Gambar 3.4 Blok Diagram Keseluruhan

Pada Gambar 3.4 terdapat blok diagram keseluruhan sistem yang memungkinkan monitoring kendaraan berdasarkan data yang dikumpulkan dari OBD-II, MPU6050, dan sensor GPS melalui website. Sistem ini juga

dapat menampilkan riwayat data seperti kecepatan, putaran mesin, dan beban mesin dari OBD-II. Selain itu, sensor MPU6050 memberikan informasi tentang kemiringan maksimal kendaraan untuk mendeteksi ketika terjadi kecelakaan, sementara GPS berfungsi sebagai penentu lokasi yang mendeteksi perubahan posisi kendaraan.

3.4.3.1 Cara Kerja Sistem



Gambar 3.5 Cara Kerja Sistem

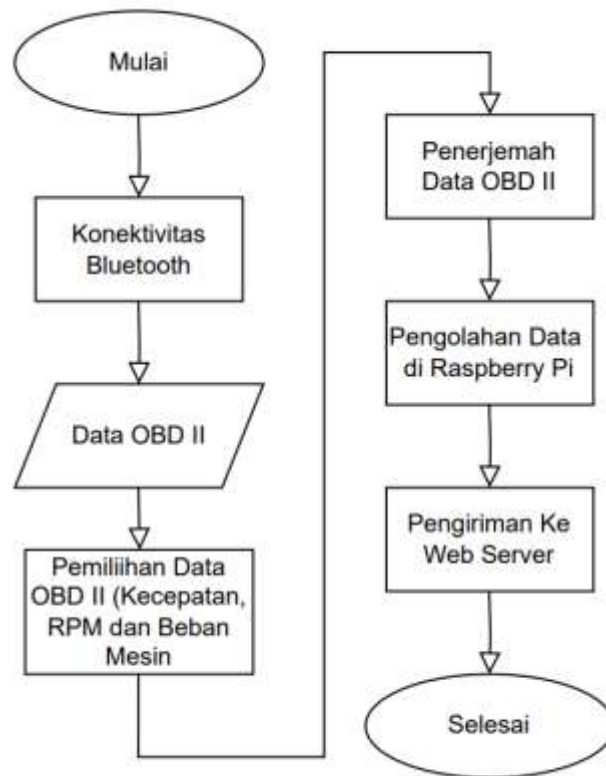
Sistem berkerja dengan melakukan pengolahan data dari OBD-II, MPU6050 dan GPS untuk pengiriman ke database. Data yang sudah ada di

database akan diolah untuk kemudian dianalisis dan disajikan di website. Analisis data yang dilakukan adalah analisis deteksi kecelakaan dan analisis data pelanggaran terhadap kemandirian berkendara berdasarkan kecepatan dan pelanggaran terhadap kendaraan yang dilakukan oleh pengemudi.

Pada sistem ini terdapat 3 data pada OBD-II yang diperlukan yaitu kecepatan, putaran mesin dan beban mesin. Untuk mengambil ke tiga data tersebut dibutuhkan 3 program sebagai berikut:

1. Program yang berfungsi untuk melakukan pemindaian port *bluetooth* dan melakukan bind ke port serial yang tersedia dan menentukan protokol kendaraan.
2. Program yang berfungsi untuk memilih PID yang diterima dari OBD-II yang diubah menjadi desimal.
3. Program yang berfungsi untuk mengatur alur komunikasi dengan perangkat OBD-II. Selain itu, program tersebut juga menginisialisasi port dengan cara mengatur kembali koneksi dengan perangkat OBD-II.

3.4.3.2 Flowchart Pengambilan dan Pengiriman Data OBD-II



Gambar 3.6 Flowchart Pengambilan Data OBD-II

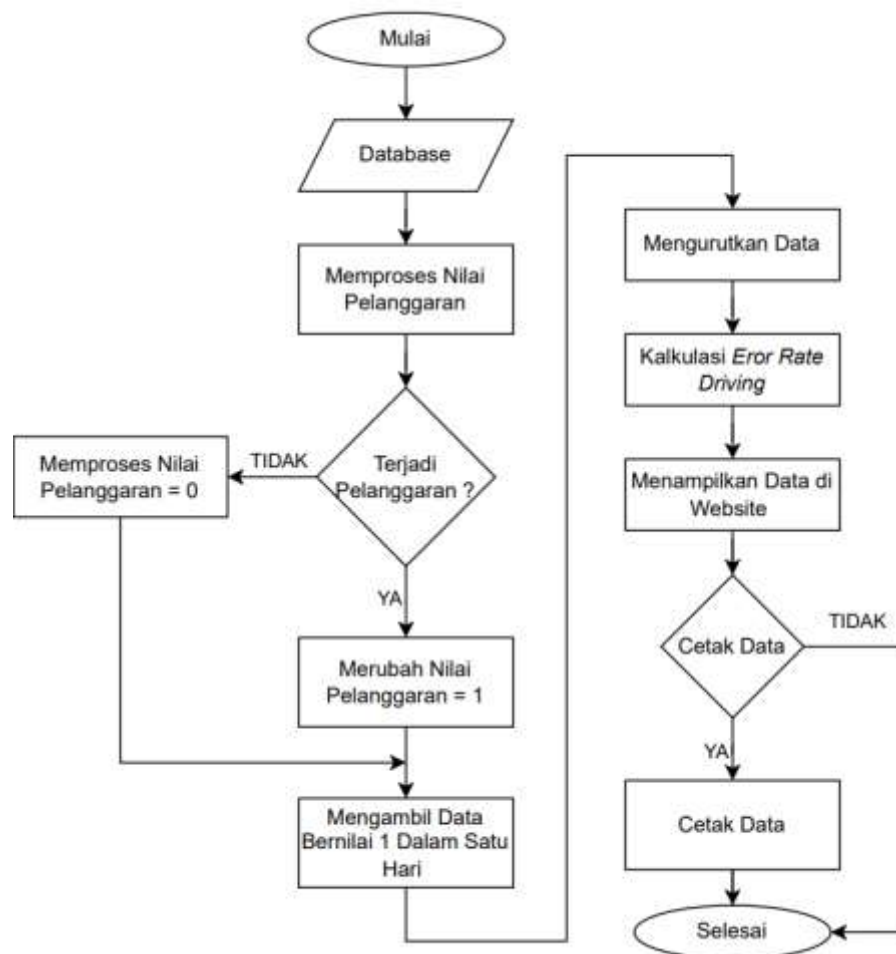
Pada proses pengambilan data OBD-II terdapat beberapa tahapan diantaranya yaitu:

1. OBD-II yang dilengkapi Bluetooth terhubung dengan kendaraan dihubungkan dengan Raspberry Pi dengan protokol ISO 15031-5.
2. Pemilihan data dari OBD-II untuk dikirim ke Raspberry Pi meliputi kecepatan, putaran mesin dan beban mesin.
3. Data PID yang diperoleh dari OBD-II disimpan terlebih dahulu dalam array sebelum diproses untuk dikirimkan.
4. Pengolahan data ini meliputi menentukan interval pengiriman data.

5. Data yang telah di proses kemudian dilakukan pengiriman ke database.

Setelah data OBD-II dikirim ke database, data yang diterima kemudian diolah oleh beberapa program. Program utama adalah program monitoring, program pengolahan data pelanggaran dan terakhir program deteksi kecelakaan.

3.4.3.3 Flowchart Pengolahan Data Pelanggaran Keseluruhan



Gambar 3.7 Flowchart Pengolahan Data Pelanggaran

Pada Gambar 3.7 merupakan proses monitoring dan pencetakan data perilaku pengemudi dengan beberapa tahapan proses sebagai berikut:

1. Pengambilan data dari database berupa data kecepatan, putaran mesin dan beban mesin.
2. Tahap kedua pemrosesan nilai awal pelanggaran bernilai 0.
3. Pemrosesan berdasarkan terjadi pelanggaran atau tidak terjadi pelanggaran. Pelanggaran ini meliputi 3 faktor yaitu kecepatan, putaran mesin dan beban mesin. Untuk menentukan posisi kendaraan, menggunakan *Open Street Maps API* untuk mendapatkan informasi berbagai jalan dengan menentukan batasan kecepatan jalan tol paling tinggi 100 kilometer per jam dan paling rendah 60 kilometer per jam, antar kota paling tinggi 80 kilometer per jam, jalan kawasan perkotaan paling tinggi 50 kilometer per jam dan jalan kawasan permukiman dengan batasan kecepatan paling tinggi 30 kilometer per jam. Putaran mesin lebih dari 4000 RPM dan beban mesin lebih dari 80%. Masing – masing pelanggaran memproses nilai 1 ketika ada satu pelanggaran.
4. Penyetelan pengambilan data dalam waktu satu sesi sewa berapa dengan menjumlahkan total pengemudi melakukan pelanggaran.
5. Mengurutkan data jumlah pelanggaran dari kategori kecepatan, beban mesin dan putaran mesin dalam satu sesi sewa.
6. Kalkulasi eror rate driving untuk mengkonversi nilai kesalahan ke dalam bentuk persentase dari total data yang terkirim.
7. Menampilkan data monitoring perhari di halaman web.

1. Flowchart Proses Pengolahan Data Pelanggaran Kecepatan



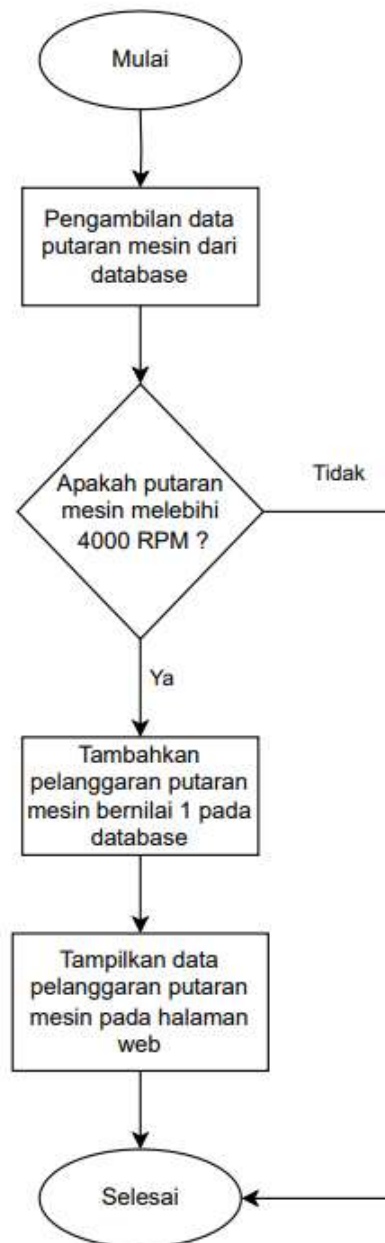
Gambar 3. 8 *Flowchart* pengolahan Pelanggaran Kecepatan

Pada Gambar 3.8 merupakan proses pengolahan data pelanggaran beban mesin dengan beberapa tahapan proses sebagai berikut:

1. Pengambilan data putaran mesin dan jenis jalan dari database.

2. Pengecekan terhadap jenis jalan untuk mengetahui apakah termasuk dalam kategori jalan tol, jalan antar kota, jalan perkotaan, atau kawasan permukiman. Setiap jenis jalan memiliki batas kecepatan yang telah ditentukan, yaitu jalan tol memiliki batas kecepatan minimum 60 km/jam dan maksimum 100 km/jam, jalan antar kota memiliki batas kecepatan maksimum 80 km/jam, jalan perkotaan maksimum 50 km/jam, dan kawasan permukiman maksimum 30 km/jam.
3. Jika kecepatan berapa diluar aturan kecepatan, maka catat pelanggaran bernilai 1 dan tampilkan pada halaman web, dan jika tidak ada pelanggaran, maka proses dihentikan (selesai).

2. Flowchart Proses Pengolahan Data Pelanggaran Putaran Mesin

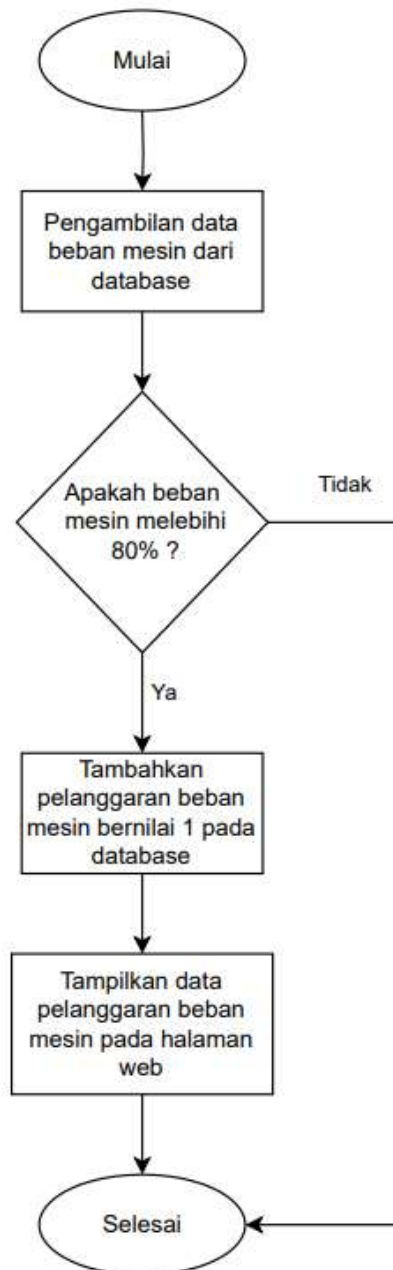


Gambar 3. 9 *Flowchart* pengolahan Pelanggaran Putaran Mesin

Pada Gambar 3.9 merupakan proses pengolahan data pelanggaran beban mesin dengan beberapa tahapan proses sebagai berikut:

1. Pengambilan data putaran mesin dari database.
2. Pengecekan apakah Jika sudut kemiringan jalan kurang dari 25° , maka batas RPM yang diperbolehkan adalah 4000. Sebaliknya, jika kemiringan lebih dari 25° , batas RPM yang diizinkan naik menjadi 6000. Jika ada data melebihi batas catat pelanggaran putaran mesin 1 pada database dan menampilkan ke halaman web.
3. Jika melebihi batas maka catat pelanggaran bernilai 1, dan jika tidak ada pelanggaran maka selesai.

3. Flowchart Proses Pengolahan Data Pelanggaran Beban Mesin

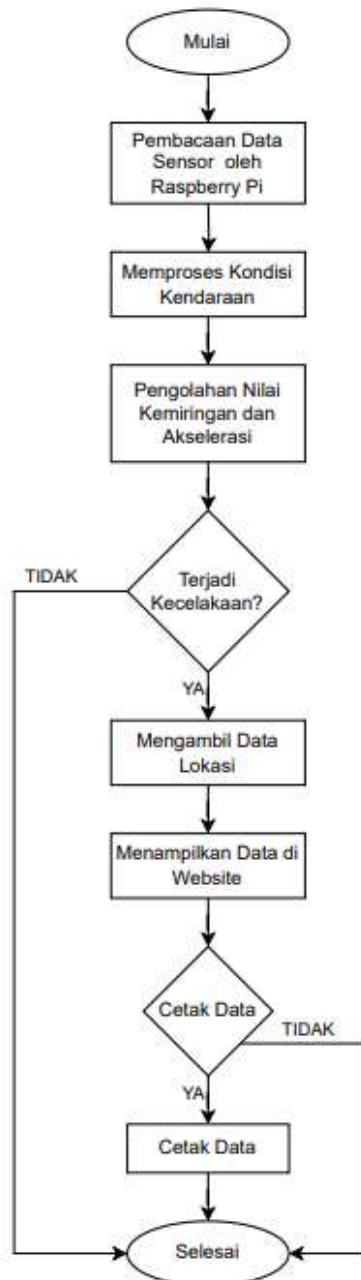


Gambar 3. 10 *Flowchart* pengolahan Pelanggaran Beban Mesin

Pada Gambar 3.10 merupakan proses pengolahan data pelanggaran beban mesin dengan beberapa tahapan proses sebagai berikut:

1. Pengambilan data beban mesin dari database.
2. Memproses jumlah pelanggaran beban mesin bernilai 0.
3. Cek apakah ada data beban mesin yang melebihi 80% pada database.
4. Jika tidak ada pelanggaran maka selesai.
5. Jika ada pelanggaran beban mesin, maka pelanggaran ditampilkan pada halaman web.

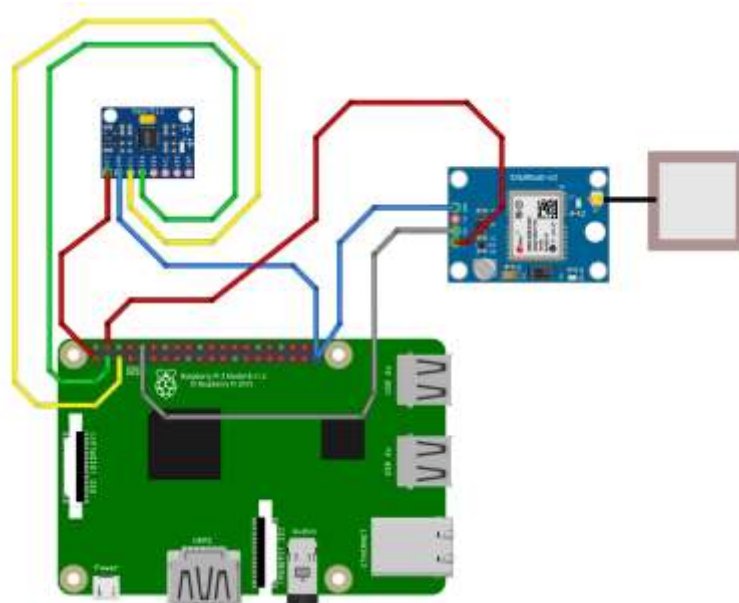
3.4.3.4 Flowchart Pengolahan Deteksi Kecelakaan



Gambar 3.11 *Flowchart* Pengolahan Detekeksi Kecelakaan

Pada Gambar 3.11 merupakan proses monitoring dan pencetakan data perilaku pengemudi dengan beberapa tahapan proses sebagai berikut:

1. Pengambilan data dari sensor MPU6050 oleh Raspberry Pi berupa data kemiringan dan akselerasi kendaraan.
 2. Memproses kondisi kendaraan bernilai 0 untuk mendeteksi ketika tidak terindikasi terjadi kecelakaan.
 3. Pemrosesan dilakukan dengan pembacaan kemiringan kendaraan dan akselerasi kendaraan menggunakan data sensor *Accelerometer* dan *Gyroscope* MPU6050 untuk menindikasikan kecelakaan yaitu mengklasifikasikan tingkat kecelakaan yaitu benturan ringan, potensi kecelakaan, kecelakaan dan kecelakaan parah berdasarkan data akselerasi negatif lebih dari 4G pada sumbu x dan sumbu y, serta kemiringan kurang dari 45 derajat dari tanah.
 4. Jika tidak terjadi kecelakaan maka selesai.
 5. Ketika terjadi kecelakaan sistem akan mengambil data lokasi terakhir saat terjadi kecelakaan menggunakan *Open Street Maps API*.
 6. Data disajikan pada website.
- a. Perancangan Perangkat Keras



Gambar 3.12 Rangkaian Hardware

Konfigurasi pin dari rangkaian *Hardware* pada Gambar 3.12 dijelaskan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Konfigurasi Pin MPU6050 dengan Raspberry Pi 3

MPU6050	Raspberry pi 3 Model B
SDA	SDA
SCI	SCL
VCC	5V
GND	GND

Tabel 3.2 Konfigurasi Pin GPS dengan Raspberry Pi 3

GPS Neoblock 6M	Raspberry pi 3 Model B
VCC	5V
GND	GND
RX	TX

b. Perancangan Perangkat Lunak

Perangkat lunak pada sistem monitoring ini merupakan perancangan program yang dijalankan pada Raspberry Pi 3 Model B untuk mengontrol seluruh sensor. Dalam sistem ini, bahasa pemrograman yang digunakan adalah Python di software Thonny Python IDE. Program ini akan mengatur seluruh aktivitas pengiriman dan pembacaan data pada terminal, serta mengontrol kinerja seluruh komponen yang terhubung ke Raspberry Pi 3 Model B.

Program ini terdiri dari beberapa bagian, di antaranya, pemanggilan pustaka (library) dan inisialisasi pin GPIO sensor. Program ini juga mencakup penerimaan data mesin dari OBD-II, pembacaan nilai dari sensor MPU6050, konversi nilai dari data mentah (raw) menjadi Roll, Pitch, dan Yaw. Proses pengkodean dimulai setelah menghubungkan Raspberry Pi 3 Model B ke komputer. Adapun tahapan sistem kerja dalam pemrograman Python adalah sebagai berikut:

1. Raspberry pi 3 *Model B* melakukan inisialisasi terhadap seluruh komponen yang digunakan dalam sistem yaitu OBD-II, sensor *Accelerometer* dan *Gyroscope* pada modul MPU6050 dan informasi Lokasi pada sensor GPS Neoblock 6M.
2. Raspberry pi 3 *Model B* mendapatkan data yang terbaca dari sensor. Data mesin dari OBD-II, data mentah yang diperoleh dari sensor *Accelerometer* dan *Gyroscope* kemudian dikonversi menjadi nilai *Roll*

dan *Pitch*. Sementara itu sensor GPS Neoblock 6M mengirim informasi lokasi ke Raspberry pi.

3. Data mesin, nilai *Roll* dan *Pitch*, informasi Lokasi yang telah diterima dari sensor kemudian di olah oleh oleh Raspberry pi 3 untuk dikirimkan ke *cloud* untuk di analisis dan *reporting*.

3.5 Metode Pengambilan Data

1. Sumber Data

Data kecepatan, putaran mesin dan beban mesin bersumber dari ECU kendaraan menggunakan OBD-II. Data kemiringan (*Roll dan Pitch*) dan akselerasi diambil dari sensor *Accelerometer* dan *Gyroscope* MPU6050.

2. Pembacaan Sensor

Sensor MPU6050 berfungsi untuk mengukur kemiringan kendaraan dan akselerasi. OBD-II berfungsi untuk membaca data kecepatan, putaran mesin dan beban mesin.

3. Penyimpanan Data

Seluruh data kecepatan, beban mesin, dan putaran. akan disimpan dalam *database MySQL*.

4. Pengolahan Data dari *Database*

Data yang telah dikumpulkan dan disimpan di *database* akan diambil untuk dianalisis dan disajikan di *website*.

5. Deteksi Kecelakaan

Deteksi kecelakaan dilakukan dengan menganalisis data akselerasi yang diperoleh dari sensor. Jika kemiringan kendaraan pada sumbu x dan

sumbu y lebih besar dari 45 derajat dari tanah dan akselerasi negatif lebih dari 4G, sistem akan menandai kejadian tersebut sebagai teridentifikasi kecelakaan.

6. Pengambilan Lokasi Kecelakaan

Sistem akan mengambil data lokasi terakhir kendaraan menggunakan *Open Street Maps API*.

7. Pencetakan data

Jika tidak terjadi kecelakaan, data dapat dicetak dalam format PDF berdasarkan pengaturan yang telah ditentukan sebelumnya baik.

3.6 Pengujian Sistem

Pengujian sistem merupakan tahap untuk memastikan bahwa semua komponen dan fungsi sistem berfungsi dengan baik setelah implementasi. Pengujian ini dilakukan untuk menyimpulkan kinerja sistem secara keseluruhan dengan fokus pada pengambilan data. Proses pengujian beberapa langkah berikut:

1. Menyiapkan Perangkat Keras

Menyiapkan perangkat keras termasuk Raspberry pi 3 *Model B*, modul OBD-II, sensor MPU6050, dan GPS Neoblock 6M untuk memastikan semua komponen terhubung dan berfungsi dengan baik.

2. Pengujian Pengambilan Data Jenis Jalan

Melakukan pengujian pada berbagai jenis jalan, mulai dari jalan bebas hambatan, jalan antar kota, jalan kawasan perkotaan, hingga jalan

kawasan permukiman untuk mengamati bagaimana sistem beradaptasi dengan kondisi jalan yang berbeda.

3. Rekam Data

Merekam data perilaku selama perjalanan dengan mengumpulkan informasi meliputi kecepatan kendaraan (*km/h*), putaran mesin (*RPM*), dan beban mesin.

4. Rekam Data Deteksi Kecelakaan

Merekam data deteksi kecelakaan dengan mencatat data percepatan *Roll* dan *Pitch* dari sensor MPU6050. Data ini digunakan untuk mengindikasikan terjadinya kecelakaan, dilengkapi dengan rekam sudut kemiringan dan nilai *G-force* yang dibandingkan dengan nilai *threshold* yang ditentukan untuk menandai potensi benturan

5. Pengujian *Packet loss*

Pengujian *packet loss* merupakan salah satu parameter penting dalam pengujian Quality of Service (QoS) yang bertujuan untuk mengetahui jumlah paket data yang hilang (tidak sampai tujuan) selama proses pengiriman data dari node menuju master atau server. Pengujian QoS ini mengukur kinerja sistem dalam hal keandalan transmisi data, terutama pada pengiriman data hasil deteksi kecelakaan.

Parameter QoS yang diuji dalam pengujian ini adalah *packet loss*, yaitu persentase kehilangan data selama komunikasi antara node (mikrokontroler dan sensor) dengan server (database). Adapun rumus perhitungan *packet loss* (%) sebagai berikut:

$$\text{Packet Loss} \quad (2.3)$$

$$= \frac{\text{Incoming message} - \text{Output message}}{\text{Incoming message}} \times 100 (\%)$$

Keterangan:

Incoming message = Data dikirim

Output message = Data diterima

Untuk memudahkan interpretasi hasil *packet loss*, nilai *packet loss* dikelompokkan ke dalam beberapa kategori atau indeks seperti yang ditunjukkan pada Tabel 3.3.

Tabel 3. 3 Nilai Indeks *Packet loss*

Kategori <i>Packet loss</i>	<i>Packet loss</i> (%)	Indeks
Sangat Baik	0	4
Baik	3	3
Sedang	15	2
Tidak Baik	25	1

3.7 Lokasi Penelitian

Lokasi perancangan alat ini dilakukan di Kecamatan Salopa, Kabupaten Tasikmalaya, Jawa Barat 46192. Sementara itu, pengujian operasional alat dilakukan melalui perjalanan lapangan yang dimulai dari Kecamatan Cisayong, Kabupaten Tasikmalaya menuju Kota Bandung, kemudian kembali ke Kecamatan Cisayong, Kabupaten Tasikmalaya.