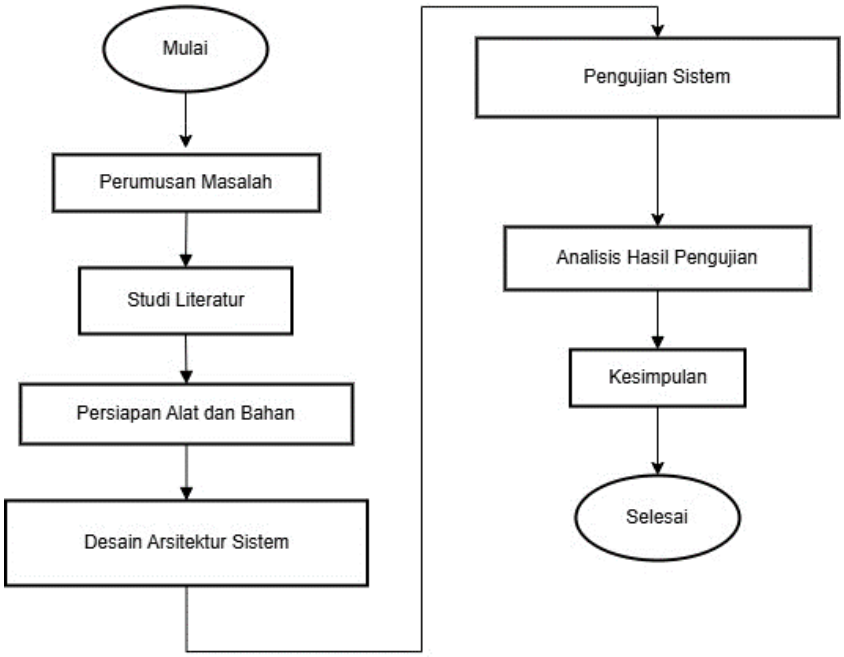
BAB III

METODOLOGI PENELITIAN

3.1 Flowchart penelitian



Gambar 3.1 Flowchart Penelitian

Berdasarkan Gambar 3.1 Flowchart penelitian terdiri dari tahapan Perumusan Masalah, Studi Literatur, Persiapan Alat dan Bahan, Desain Arsitektur Sistem, Pengujian Sistem, serta Analisis Hasil dan Kesimpulan.

3.1.1 Perumusan Masalah

Langkah awal dalam penelitian ini dimulai dengan mengidentifikasi permasalahan yang berkaitan dengan kondisi pengemudi, khususnya saat mengalami kantuk atau distraksi akibat penggunaan ponsel. Kedua kondisi tersebut diketahui sebagai faktor dominan penyebab kecelakaan lalu lintas.

Berdasarkan permasalahan tersebut, penelitian ini merancang sistem yang memanfaatkan kamera dan algoritma YOLO untuk mendeteksi objek terkait gejala kantuk dan distraksi. Hasil deteksi kemudian diproses menggunakan logika berbasis durasi untuk mengklasifikasikan kondisi pengemudi ke dalam kategori mengantuk, atau terdistraksi. Klasifikasi tersebut dikirimkan ke server berbasis web untuk disimpan dan dianalisis lebih lanjut.

3.1.2 Studi Literatur

Studi literatur dalam penelitian ini dilakukan untuk meninjau berbagai referensi terkait implementasi algoritma YOLO dalam mendeteksi objek visual pada pengemudi. Deteksi objek-objek tersebut menjadi dasar bagi sistem dalam melakukan klasifikasi kondisi pengemudi berdasarkan aturan waktu tertentu.

Kajian juga mencakup aspek teknis penggunaan kamera sebagai alat akuisisi citra secara real-time. Informasi dari studi literatur ini dijadikan landasan dalam pemilihan algoritma deteksi, desain arsitektur sistem, serta perangkat pendukung lainnya guna mengembangkan sistem klasifikasi kondisi pengemudi yang efektif dan efisien secara *real-time*.

3.2 Persiapan Alat dan Bahan

Dalam penelitian “Analisis Penyebab Kecelakaan Kendaraan Berdasarkan Deteksi Kantuk Dan Distraksi Pengemudi Menggunakan YOLO”, terdapat dua kategori utama alat dan bahan yang digunakan, yaitu perangkat keras dan perangkat lunak.

3.2.1 Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini mencakup komponen utama yang berperan dalam akuisisi data, pemrosesan informasi, serta komunikasi dengan server. Berikut merupakan penjelasan perangkat keras yang digunakan:

1. Laptop

   Perangkat ini berfungsi dalam tahap pengembangan sistem, implementasi kode program, serta pengujian fungsionalitas perangkat lunak.

2. Raspberry Pi 5

   Berperan sebagai unit pemrosesan utama yang digunakan untuk menjalankan algoritma visi komputer dalam proses deteksi objek visual pengemudi, serta mengirimkan data hasil deteksi ke server melalui protokol komunikasi HTTP.

3. Webcam

   Webcam berfungsi untuk menangkap citra wajah pengemudi secara real-time sebagai input dalam proses deteksi kondisi mengantuk atau terdistraksi akibat penggunaan ponsel saat berkendara.

3.2.2 Perangkat Lunak

Selain perangkat keras, penelitian ini juga menggunakan berbagai perangkat lunak untuk mendukung proses pelabelan data, pelatihan model, serta analisis performa deteksi. Beberapa perangkat lunak utama yang digunakan antara lain:

1. Roboflow

   Roboflow digunakan sebagai alat bantu dalam proses pelabelan dataset secara manual serta melakukan augmentasi data untuk meningkatkan variasi citra.

2. Google *Colaboratory* (*Colab*)

   Platform cloud untuk pelatihan YOLOv8 dengan dukungan GPU Tesla T4 dan RAM 15 GB.

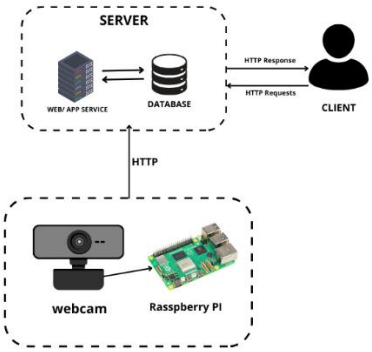
3. Ultralytics YOLOv8

   Library resmi YOLO yang digunakan untuk melatih, memvalidasi, memprediksi, dan menampilkan hasil deteksi secara efisien.

3.3 Desain Arsitektur Sistem

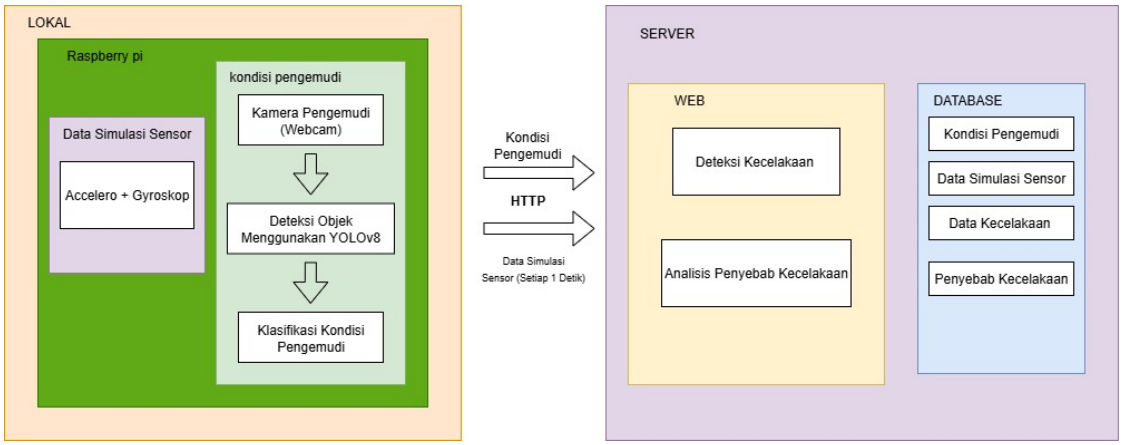
Sistem yang dirancang dalam penelitian ini terdiri atas dua komponen utama, yaitu perangkat lokal yang dipasang pada kendaraan dan server berbasis web. Perangkat lokal mencakup kamera dan Raspberry Pi 5 yang berfungsi sebagai unit pemrosesan utama. Kamera terhubung langsung dengan Raspberry Pi untuk melakukan akuisisi citra wajah pengemudi secara real-time. Selain itu, Raspberry Pi juga menghasilkan data simulatif dari sensor akselerometer dan gyroskop.

Kedua jenis data, yaitu data hasil klasifikasi kondisi pengemudi dan data sensor simulatif, dikirimkan secara paralel ke server melalui jaringan internet. Server web berperan sebagai penerima, penyimpan, dan pengelola data yang dikirimkan dari perangkat Raspberry Pi. Desain keseluruhan arsitektur sistem disajikan pada Gambar 3.2 sebagai acuan visual dari integrasi komponen yang digunakan.



Gambar 3.2 Desain Arsitektur Sistem

3.4 Alur Kerja Sistem



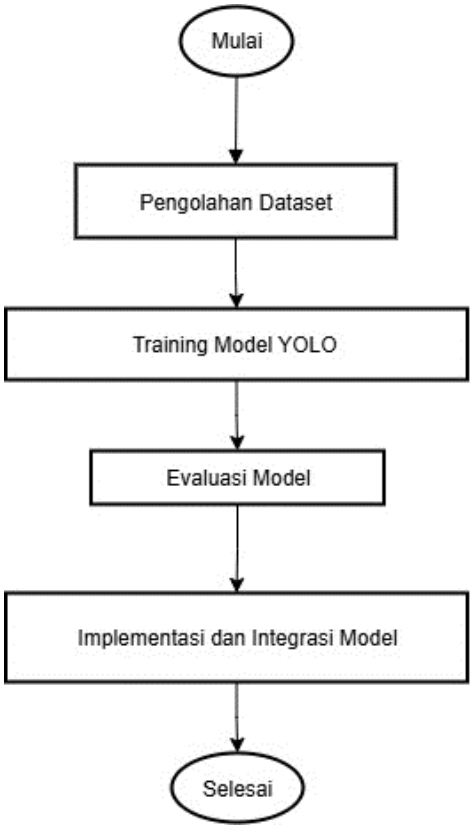
Gambar 3.3 Alur Kerja Sistem

Gambar 3.3 menyajikan alur kerja sistem yang terdiri atas dua komponen utama, yaitu perangkat lokal yang dipasang pada kendaraan dan server berbasis web. Sistem ini berfungsi untuk melakukan proses deteksi objek visual pada pengemudi secara real-time, mengklasifikasikan kondisi pengemudi berdasarkan hasil deteksi tersebut, serta melakukan analisis terhadap kemungkinan penyebab terjadinya kecelakaan berdasarkan data yang diterima.

Proses diawali dengan akuisisi citra wajah pengemudi melalui kamera, yang selanjutnya diproses oleh model YOLOv8 pada perangkat Raspberry Pi untuk mendeteksi objek visual yang berkaitan dengan kondisi pengemudi. Berdasarkan hasil deteksi tersebut, Raspberry Pi menerapkan logika klasifikasi berbasis aturan guna menentukan kondisi pengemudi ke dalam dua kategori, yaitu mengantuk atau distraksi.

Secara bersamaan, Raspberry Pi juga mengirimkan data simulatif dari sensor akselerometer dan gyroskop ke server setiap satu detik. Data tersebut kemudian dianalisis oleh server untuk mendeteksi kemungkinan terjadinya kecelakaan berdasarkan ambang batas yang telah ditetapkan. Jika sistem mendeteksi adanya kecelakaan, maka dilakukan analisis terhadap data klasifikasi kondisi pengemudi dalam rentang waktu sebelum kejadian untuk menentukan faktor penyebab kecelakaan tersebut.

3.5 Flowchart Pembuatan Model YOLO



Gambar 3.4 Flowchart Pembuatan Model YOLO

Pada proses pelatihan dan implementasi model YOLOv8, terdapat beberapa tahapan di antaranya yaitu:

1. Pengolahan Dataset

   Dataset pada penelitian ini dikumpulkan secara mandiri dalam berbagai kondisi pencahayaan dan sudut pandang. Citra diberi anotasi manual menggunakan Roboflow untuk menandai objek visual sesuai kelas, lalu melalui proses augmentasi guna meningkatkan variasi data. Dataset yang telah diolah kemudian dibagi ke dalam tiga bagian yaitu *train*, *valid*, dan *test*.

2. Pelatihan Model YOLO

Model YOLO dilatih menggunakan platform Google Colaboratory dengan memanfaatkan GPU berbasis cloud. Dataset yang telah dipersiapkan digunakan dalam proses pelatihan untuk melatih model dalam mengenali objek visual. Tujuan dari proses ini adalah agar model mampu melakukan deteksi kondisi visual secara cepat dan tepat ketika diimplementasikan pada perangkat edge seperti Raspberry Pi.

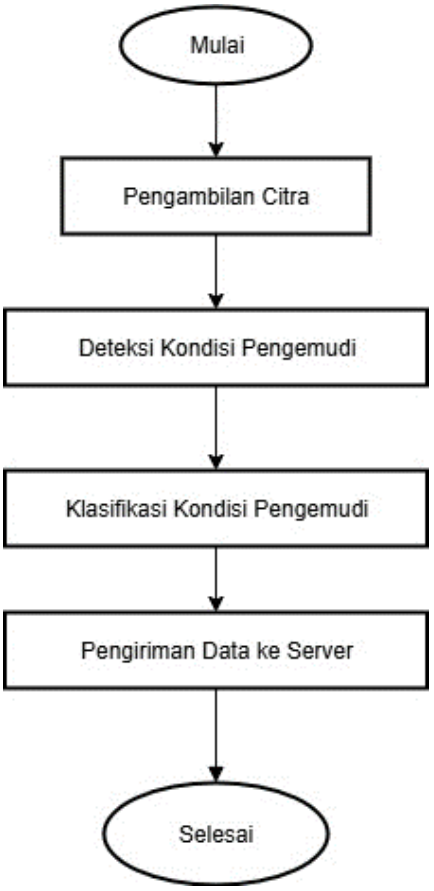
3. Evaluasi Performa Model

Model yang telah dilatih kemudian dievaluasi menggunakan data validasi dengan mengacu pada metrik evaluasi seperti precision, recall, *dan* mean average precision (mAP), untuk mengukur kemampuan model dalam mengenali objek yang belum pernah dilihat sebelumnya.

4. Implementasi dan Integrasi Model

Model deteksi yang telah melewati tahapan pelatihan dan evaluasi selanjutnya diimplementasikan pada perangkat Raspberry Pi untuk dijalankan secara lokal. Hasil deteksi objek dari model tersebut digunakan sebagai input dalam sistem klasifikasi kondisi pengemudi**.**

3.6 Flowchart Klasifikasi Kondisi Pengemudi



Gambar 3.5 Flowchart Deteksi Kondisi Pengemudi

Pada proses klasifikasi kondisi pengemudi, terdapat beberapa tahapan diantaranya yaitu:

1. Kamera Menangkap Citra Pengemudi

   Kamera eksternal yang terpasang pada bagian dashboard kendaraan digunakan untuk mengambil citra pengemudi secara *real-time*.

2. Deteksi Kondisi Pengemudi

Citra yang diperoleh diproses oleh Raspberry Pi menggunakan model YOLOv8 untuk mendeteksi empat objek visual yaitu mata kanan tertutup, mata kiri tertutup, menguap, dan ponsel.

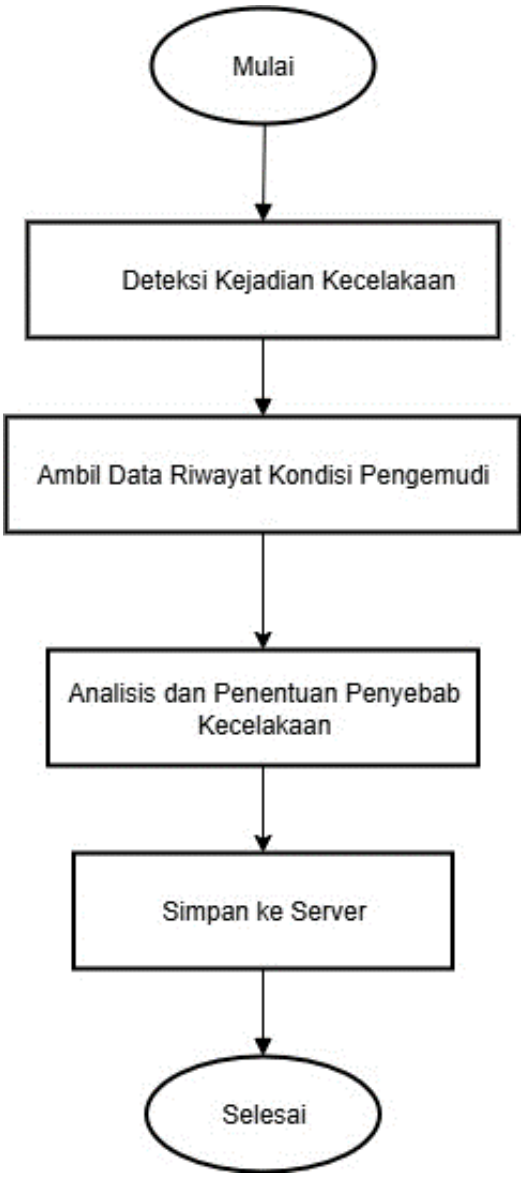
3. Klasifikasi Kondisi Pengemudi

Berdasarkan hasil deteksi tersebut, Raspberry Pi menerapkan logika berbasis aturan untuk mengklasifikasikan kondisi pengemudi ke dalam dua kategori, yaitu mengantuk, dan distraksi.

4. Kirim Data ke Server

Sistem mengirim hasil klasifikasi hanya jika objek terdeteksi minimal dua detik. Kondisi mengantuk dikategorikan saat kedua mata tertutup atau aktivitas menguap terdeteksi, sedangkan distraksi ditetapkan jika ponsel terdeteksi dalam durasi sama. Jika deteksi berlanjut, data dikirim ulang tiap dua detik. Namun jika terputus, perhitungan waktu dimulai ulang. Apabila koneksi internet tidak tersedia, data disimpan sementara di perangkat lokal dan dikirim otomatis saat koneksi pulih.

3.7 Flowchart Analisis Penyebab Kecelakaan



Gambar 3.6 Flowchart Analisis Penyebab Kecelakaan

Pada proses analisis penyebab kecelakaan terdapat beberapa tahapan di antaranya yaitu :

1. Deteksi Kejadian Kecelakaan

   Server menerima data simulasi sensor akselerometer dan gyroskop setiap satu detik. Jika nilai percepatan melebihi 4 G atau jika kemiringan sudut lebih dari 45 derajat sistem mengidentifikasi kejadian tersebut sebagai kecelakaan.

2. Pengambilan Riwayat Kondisi Pengemudi

   Setelah kecelakaan terdeteksi, server mengakses data klasifikasi kondisi pengemudi dalam rentang waktu 10 detik sebelum kejadian. Data ini merupakan hasil klasifikasi yang telah dikirimkan oleh Raspberry Pi sebelumnya dan telah tersimpan di server web yaitu mengantuk, atau distraksi.

3. Analisis dan Penentuan Penyebab

   Data yang terkumpul dianalisis untuk menentukan kondisi yang paling dominan. Jika kondisi mengantuk lebih dominan, maka penyebab kecelakaan ditetapkan sebagai akibat dari kantuk. Sebaliknya, jika distraksi lebih dominan, maka penyebab diklasifikasikan sebagai akibat dari distraksi. Apabila proporsi keduanya seimbang, maka penyebab dikategorikan sebagai gabungan antara kantuk dan distraksi. Namun, jika tidak terdapat data kondisi apa pun sebelum kecelakaan, maka penyebabnya tidak dapat diketahui.

4. Hasil analisis disimpan ke server

Hasil analisis disimpan ke server dan ditampilkan melalui antarmuka web.

3.8 Metode Pengambilan Data

1. Sumber Data

Penelitian ini menggunakan dua jenis data, yaitu visual dan simulatif. Data visual diperoleh dari kamera yang mengarah ke pengemudi dan diproses secara real-time oleh Raspberry Pi 5 menggunakan model YOLOv8 untuk mendeteksi kondisi seperti mata tertutup, menguap, dan penggunaan ponsel. Berdasarkan hasil deteksi tersebut, Raspberry Pi mengklasifikasikan kondisi pengemudi secara lokal ke dalam dua kategori yaitu mengantuk, dan distraksi.

Sementara itu, data simulatif berupa nilai akselerometer dan giroskop dihasilkan secara acak untuk mensimulasikan kejadian kecelakaan. Suatu kejadian dianggap sebagai kecelakaan apabila percepatan melebihi 4 G atau kemiringan sudut lebih dari 45 derajat. Data ini berfungsi sebagai pemicu logika sistem dan tidak merepresentasikan kecelakaan nyata.

2. Pengiriman dan Penyimpanan Data Klasifikasi dan Sensor

Raspberry Pi mengirimkan dua jenis data ke server web secara paralel. Data dari sensor akselerometer dan giroskop dikirim setiap satu detik untuk mendeteksi kemungkinan kecelakaan. Sementara itu, hasil klasifikasi kondisi pengemudi dikirim hanya jika terdeteksi kondisi mengantuk atau distraksi. Jika

tidak ada objek yang memenuhi logika klasifikasi, maka tidak ada data yang dikirim ke server.

Setiap data klasifikasi mencakup label kondisi pengemudi, waktu klasifikasi, dan citra hasil deteksi. Jika koneksi internet terputus, seluruh data disimpan sementara di perangkat dan akan dikirim ulang secara otomatis setelah koneksi tersedia kembali.

3. Analisis Penyebab Kecelakaan

Setelah kejadian kecelakaan terdeteksi, server mengakses data klasifikasi kondisi pengemudi yang telah dikirim oleh Raspberry Pi dalam rentang waktu sepuluh detik sebelum kejadian. Data tersebut tersimpan dalam dua kategori kondisi, yaitu mengantuk atau distraksi. Jika kondisi mengantuk lebih dominan, maka penyebab dikategorikan sebagai akibat kantuk. Jika distraksi lebih sering muncul, maka dikategorikan sebagai akibat distraksi. Apabila keduanya muncul dengan frekuensi yang sama, maka penyebab dianggap sebagai gabungan antara kantuk dan distraksi.

4. Akses dan Visualisasi

Seluruh data yang terkumpul dan hasil analisis disajikan melalui antarmuka web dalam bentuk tabel. Riwayat kondisi pengemudi, waktu kejadian kecelakaan, dan penyebab yang teridentifikasi dapat ditampilkan secara terstruktur. Sistem juga menyediakan fitur ekspor laporan dalam format PDF yang mencakup informasi sesi penyewaan serta hasil analisis penyebab kecelakaan.

3.9 Metode Pengujian Sistem

Tahapan pengujian yang dilakukan untuk memverifikasi fungsionalitas utama sistem, mulai dari deteksi objek oleh model YOLOv8 hingga mekanisme pengiriman dan penyimpanan data. Pengujian dibagi menjadi tiga tahap sebagai berikut:

1. Menyiapkan Perangkat Keras

   Menyiapkan perangkat keras termasuk Raspberry Pi 5 dan kamera untuk memastikan semua komponen berfungsi dengan baik.

2. Evaluasi Model YOLO

Model YOLOv8 dievaluasi untuk mendeteksi enam objek visual, yaitu mata kanan terbuka, mata kanan tertutup, mata kiri terbuka, mata kiri tertutup, aktivitas menguap, dan ponsel. Evaluasi dilakukan menggunakan metrik *precision, recall, F1-score, dan mean Average Precision* (mAP). Tujuan pengujian ini adalah untuk mengukur akurasi model sebelum diterapkan pada perangkat Raspberry Pi.

3. Pengujian Deteksi

Pengujian dilakukan di dalam kendaraan untuk mengevaluasi keandalan sistem dalam mendeteksi empat objek visual secara real-time, yaitu mata kanan tertutup, mata kiri tertutup, menguap, dan ponsel. Proses pengujian melibatkan dua orang pengemudi yang berbeda guna memastikan keakuratan sistem pada berbagai karakteristik wajah. Variasi kondisi yang diuji meliputi pencahayaan siang dan malam, serta penggunaan kacamata transparan, semi-transparan, dan hitam.

4. Pengujian Mekanisme Pengiriman Data

Melakukan pemutusan koneksi internet secara terkontrol untuk mengamati respons sistem dalam menyimpan data secara lokal. Setelah koneksi kembali tersedia, sistem diuji untuk memastikan data yang tertunda dapat terkirim ulang secara otomatis tanpa kehilangan maupun duplikasi.