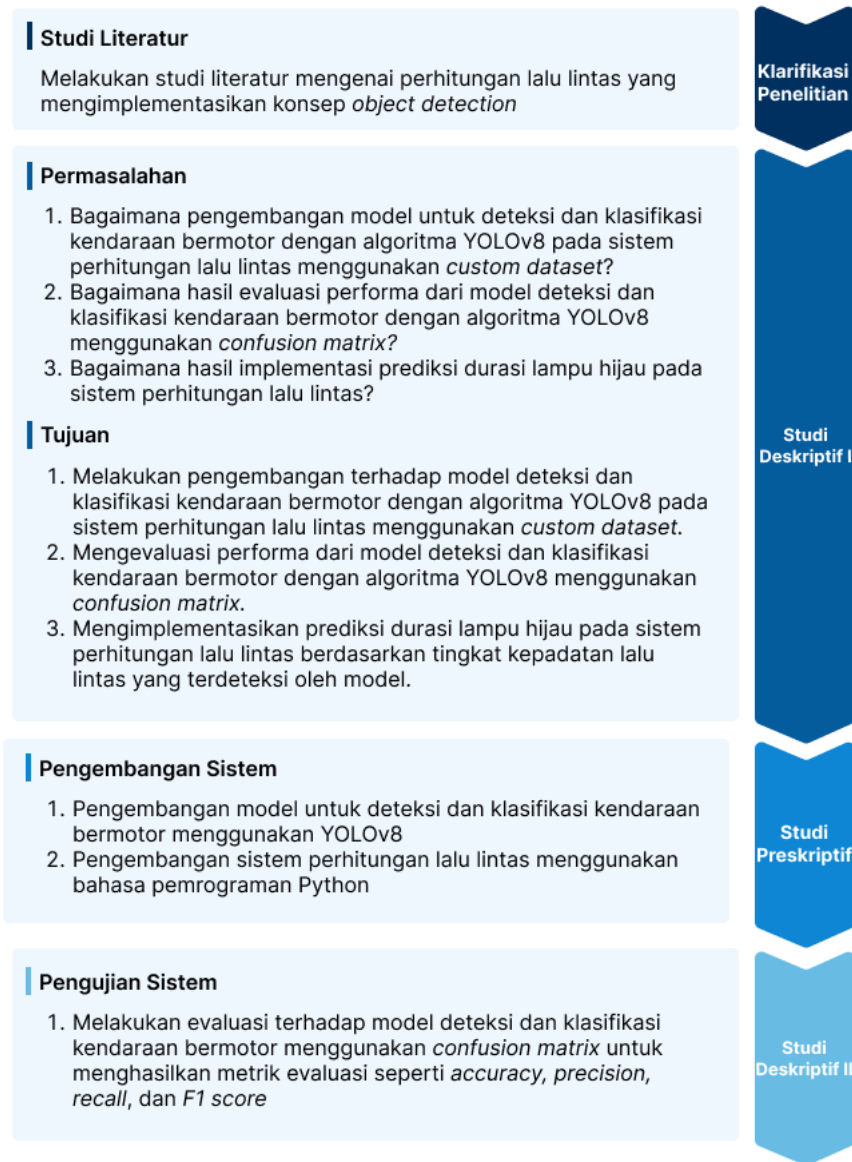


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian adalah suatu kerangka kerja yang berfungsi sebagai panduan dalam melaksanakan penelitian. *Design Research Methodology* (DRM) dipilih dalam penelitian ini dikarenakan DRM cocok diterapkan pada penelitian yang berfokus pada solusi permasalahan dan pengembangan perangkat lunak (Orisa, Faisol, & Ashari, 2023). Skema penelitian dirancang oleh penulis sesuai dengan metode DRM, sebagaimana tergambar pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

3.1.1 Klarifikasi Penelitian

Dalam tahap klarifikasi penelitian ini, fokus utama berada pada studi literatur mengenai perhitungan lalu lintas yang memanfaatkan konsep *object detection*. Studi literatur ini mencakup pemahaman terhadap konsep *object detection* beserta algoritma-algoritma yang terdapat pada konsep *object detection* itu sendiri, terutama algoritma YOLO dengan penekanan pada penggunaannya dalam konteks perhitungan lalu lintas untuk deteksi dan pengawasan objek kendaraan secara *realtime*.

Tahap klarifikasi ini bertujuan untuk menyusun landasan teoritis yang kuat dan pemahaman yang jelas terkait implementasi algoritma YOLO pada sistem perhitungan lalu lintas. Dengan demikian, tahap ini menjadi langkah kunci dalam memahami konsep-konsep dasar yang akan diimplementasikan dalam penelitian lebih lanjut terkait pengembangan sistem yang diusulkan.

3.1.2 Studi Deskriptif I

Tahap berikutnya melibatkan penguraian permasalahan dan tujuan penelitian berdasarkan hasil temuan dari studi literatur yang telah dilaksanakan pada tahap sebelumnya. Analisis literatur menjadi dasar untuk merinci aspek-aspek krusial yang perlu diatasi dan dicapai dalam penelitian ini. Informasi yang ditemukan akan menjadi pondasi yang kuat dalam mengembangkan solusi bagi permasalahan penelitian yang diidentifikasi.

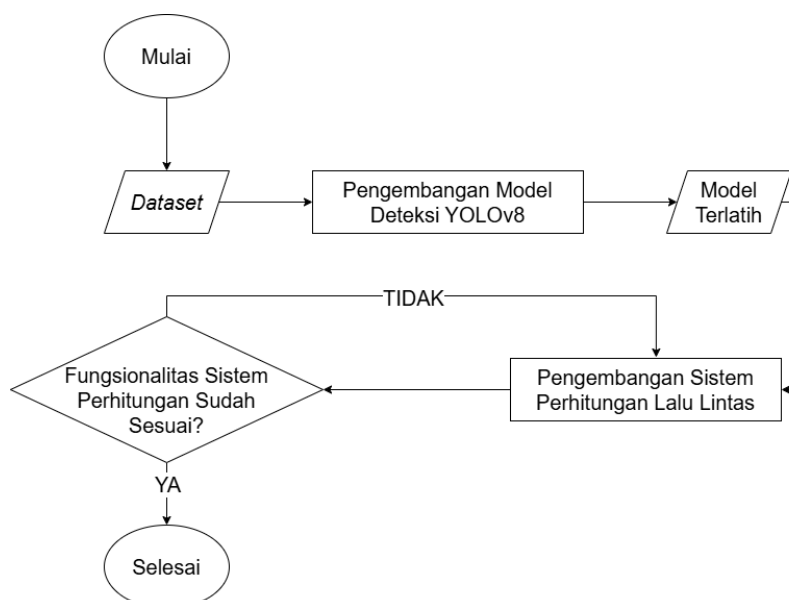
3.1.3 Studi Preskriptif

Pada tahap Studi Preskriptif ini, penelitian berfokus terhadap pengembangan sistem pada dua sub-sistem. Pertama, pengembangan model deteksi dan klasifikasi kendaraan bermotor menggunakan algoritma YOLOv8 akan ditempuh. Pelatihan pada model dilakukan berdasarkan *custom dataset* yang terdiri dari 3 kelas, yakni *car*, *motorcycle*, dan *large vehicle*. Sehingga keluaran dari proses ini adalah model deteksi dan klasifikasi kendaraan bermotor terlatih berbasis YOLOv8 yang memiliki performa terbaik.

Selanjutnya, penelitian ini akan melibatkan perancangan sistem perhitungan lalu lintas menggunakan bahasa pemrograman Python. Pada tahap ini, integrasi model terlatih dilakukan terhadap sistem perhitungan lalu lintas. Selain itu,

dilakukan proses konfigurasi mekanisme sistem perhitungan lalu lintas, mulai dari tahap *input*, proses, hingga *output*. Proses *input* akan bersumber dari rekaman video lalu lintas, yang kemudian diproses oleh sistem perhitungan lalu lintas untuk dideteksi dan dihitung jumlah kendaraan dari masing-masing kelas. *Output* dari sistem ini akan berisi informasi mengenai jumlah kendaraan per jenis, jumlah total kendaraan, dan prediksi durasi lampu hijau untuk lalu lintas adaptif.

Keseluruhan tahapan ini bertujuan untuk menghasilkan sistem komprehensif yang mencakup model deteksi dan klasifikasi kendaraan bermotor yang diimplementasikan pada sistem perhitungan lalu lintas. Alur dari studi preskriptif digambarkan dalam bentuk *flowchart* yang tercantum pada Gambar 3.2.



Gambar 3.2 *Flowchart* Studi Preskriptif

3.1.4 Studi Deskriptif II

Dalam tahap Studi Deskriptif II, penelitian ini akan mengimplementasikan evaluasi terhadap model yang telah dikembangkan. Pengujian ini akan memanfaatkan *confusion matrix*, nantinya dari *confusion matrix* dapat digunakan berbagai rumus yang menghasilkan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1 score* untuk menyediakan gambaran mendalam tentang kinerja model. Metode ini digunakan peneliti untuk mengukur sejauh mana model dapat mendeteksi kendaraan dengan benar.

3.2 Alat Penelitian

Dibawah ini tercantum alat yang digunakan untuk mendukung jalannya penelitian. Peralatan penelitian akan dibagi menjadi dua kelompok, yakni perangkat keras dan perangkat lunak.

1. Perangkat keras
 - Processor : Intel Core i7-12700H 20 CPU @2.3GHz
 - Kartu grafis : NVIDIA GeForce RTX 3050
 - Kartu memori : 16GB RAM
 - Penyimpanan data : SSD 512GB
2. Perangkat lunak

Tabel 3.1
Perangkat Lunak Penelitian

No.	Perangkat Lunak	Deskripsi
1	<i>Operating System</i> Windows 11	Sistem operasi untuk menjalankan penelitian
2	<i>Web Browser</i> Microsoft Edge	Peramban untuk menjalankan penelitian
3	<i>Integrated Development Environment</i> (Visual Studio Code)	Mewadahi seluruh proses pengembangan sistem perhitungan lalu lintas dengan menggabungkan berbagai <i>tools</i> ke dalam satu <i>interface</i> yang terpadu
4	NVIDIA CUDA Toolkit v11.8	Platform komputasi paralel yang digunakan untuk mempercepat proses komputasi pada <i>machine learning</i> untuk sistem deteksi
5	NVIDIA cuDNN v9.1.1	<i>Library</i> pendukung pelatihan <i>machine learning</i> untuk sistem deteksi yang memanfaatkan GPU NVIDIA
6	PyTorch v2.3	<i>Framework machine learning</i> untuk mendukung efisiensi penggunaan CUDA pada GPU dalam proses pengembangan sistem deteksi

No.	Perangkat Lunak	Deskripsi
7	Anaconda v2023.09	<i>Package manager open source</i> untuk kemudahan <i>environment configuration</i> pada sistem deteksi
8	Python v3.11	Bahasa pemrograman yang digunakan untuk pengembangan sistem deteksi
9	Jupyter Notebook v2024.1.1	<i>Development environment</i> yang digunakan untuk menjalankan kode secara interaktif pada sistem deteksi
10	OpenCV v4.9.0.80	<i>Library</i> untuk pengolahan dan pemrosesan video pada sistem deteksi
11	Supervision v.0.18.0	<i>Tools</i> untuk pengolahan dan pemrosesan video pada sistem deteksi
12	Roboflow v1.1.19	<i>Framework</i> untuk pengolahan <i>dataset (annotate)</i> pada sistem deteksi
13	Ultralytics v8.1.11	<i>Framework</i> untuk <i>visualisasi, training, dan deploy</i> model pada sistem deteksi
14	NVIDIA GeForce Experience 3.26	Digunakan untuk mengumpulkan <i>dataset</i> penelitian yakni video CCTV PELINDUNG dalam bentuk rekaman

3.3 Instrumen Penelitian

Instrumen yang dimanfaatkan dalam penelitian ini adalah rumus-rumus terkait *confusion matrix* untuk mengevaluasi tingkat keberhasilan model deteksi kendaraan. *Confusion matrix* merupakan alat evaluasi kinerja yang umum digunakan dalam pemodelan statistik untuk memberikan gambaran mengenai performa model yang akan dihasilkan. Sederhananya, *confusion matrix* menggambarkan hasil performa model pada suatu *dataset* melalui perbandingan prediksi model dengan nilai yang sebenarnya. *Confusion matrix* terdiri dari empat *region* yang mewakili kombinasi atas hasil prediksi dan nilai aktual, seperti yang terdapat pada Tabel 3.2.

Tabel 3.2
Confusion Matrix

Jenis Kelas	Aktual	
Prediksi	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Pengukuran kinerja dengan menggunakan *confusion matrix* melibatkan empat kategori yang mencerminkan hasil dari proses klasifikasi sebagai berikut:

1. *True Positive (TP)*: Jumlah data yang benar-benar positif dan berhasil diklasifikasikan dengan benar oleh sistem.
2. *True Negative (TN)*: Jumlah data yang benar-benar negatif dan berhasil diklasifikasikan dengan benar oleh sistem.
3. *False Positive (FP)*: Jumlah data yang sebenarnya negatif, namun sistem salah mengklasifikasikannya sebagai positif.
4. *False Negative (FN)*: Jumlah data yang sebenarnya positif, namun sistem salah mengklasifikasikannya sebagai negatif.

Dengan menggunakan kombinasi yang terdapat pada *confusion matrix*, terdapat beberapa metrik evaluasi yang dapat dihasilkan seperti *accuracy*, *precision*, *recall*, dan *f1 score*.

Accuracy merupakan ukuran seberapa sering model membuat prediksi yang benar dari seluruh prediksi yang dibuat. Semakin tinggi tingkat akurasi, semakin baik kinerja model dalam mengklasifikasikan semua kelas. Dengan kata lain, dalam penelitian ini *accuracy* merujuk pada seberapa sering model dapat mengidentifikasi kendaraan dengan benar dan tanpa kesalahan. Nilai *accuracy* dapat diperoleh melalui persamaan berikut.

$$\frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (3.1)$$

Precision merupakan ukuran seberapa sering prediksi positif model benar dari seluruh prediksi positif yang dibuat. Dengan kata lain, dalam penelitian ini *precision* merujuk pada seberapa sering prediksi positif model benar, yaitu kendaraan yang terdeteksi oleh model sebagai kendaraan. Nilai *precision* dapat diperoleh melalui persamaan berikut.

$$\frac{TP}{TP + FP} \dots\dots\dots (3.2)$$

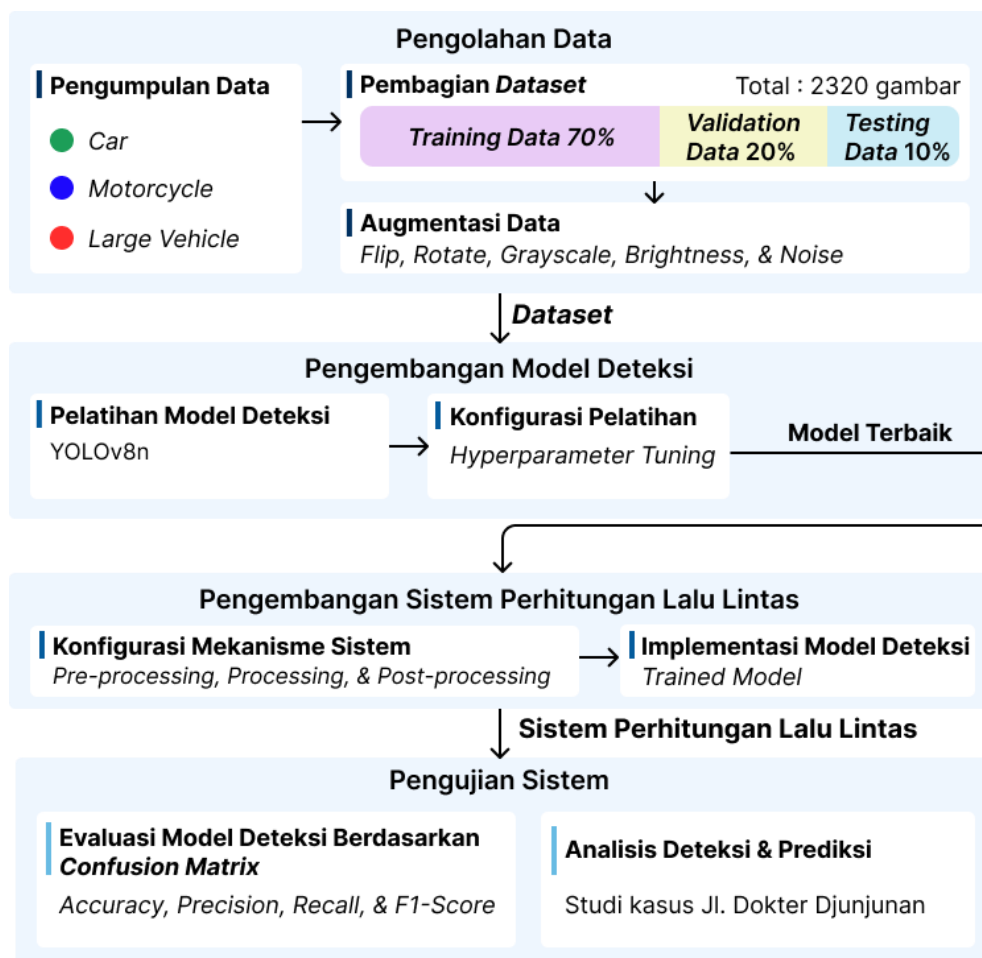
Recall merupakan ukuran seberapa sering model mampu mendeteksi semua kasus positif dari seluruh kasus positif yang ada. Dengan demikian, dalam penelitian ini *recall* merujuk pada seberapa sering model dapat mengidentifikasi kendaraan yang sebenarnya sebagai kendaraan. Nilai *recall* dapat diperoleh melalui persamaan berikut.

$$\frac{TP}{TP + FN} \dots\dots\dots (3.3)$$

F1 score menggabungkan *precision* dan *recall* untuk memberikan nilai komprehensif tentang kinerja model untuk mendeteksi sebanyak mungkin dengan tepat. Dengan kata lain, *f1 score* merupakan rata-rata dari *precision* dan *recall*, yang memberikan penilaian seimbang antara kedua metrik tersebut. Nilai *f1 score* dapat diperoleh melalui persamaan berikut.

$$\frac{2 \times Precision \times Recall}{Precision + Recall} \dots\dots\dots (3.4)$$

3.4 Prosedur Penelitian



Gambar 3.3 Prosedur Penelitian

3.4.1 Pengolahan Data

Pada tahap pengolahan data, pengumpulan data akan dilakukan dengan cara mengumpulkan *dataset* yang berisikan kendaraan di jalan-jalan Kota Bandung yang terdiri dari 3 kelas, yakni *car*, *motorcycle*, dan *large vehicle*. Tujuan dari pelabelan adalah untuk membantu mengidentifikasi dan menjelaskan konten dari setiap data. *Output* dari proses pelabelan adalah file teks berformat .txt berisikan angka yang merepresentasikan label kelas hingga titik koordinat dari masing-masing *bounding box*. Kemudian, pembagian *dataset* dilakukan dengan rasio 70% untuk data latih, 20% untuk data validasi, dan 10% untuk data uji. Terakhir, augmentasi dilakukan terhadap *dataset* untuk meningkatkan variasi data. Keluaran dari tahap pengolahan data adalah *dataset* kendaraan yang telah diproses dan siap digunakan untuk tahap pelatihan model. *Dataset* ini menggunakan anotasi teks dan konfigurasi YAML

Riyandi Firman Pratama, 2024

DETEKSI KENDARAAN BERMOTOR DAN PREDIKSI DURASI LAMPU HIJAU DENGAN ALGORITMA YOLOv8 PADA SISTEM PERHITUNGAN LALU LINTAS: STUDI KASUS JALAN DOKTER DJUNJUNAN KOTA BANDUNG

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

(YAML *Ain't Markup Language*) berbasis format YOLOv8 dan memiliki format file .zip.

3.4.2 Pengembangan Model Deteksi

Setelah *dataset* kendaraan terbuat, tahap pengembangan model deteksi akan ditempuh. Pelatihan model deteksi menjadi langkah awal pada tahap ini dengan menggunakan model *pre-trained* YOLOv8n (*nano*) dengan memanfaatkan *platform* Ultralytics HUB. Konfigurasi *hyperparameter tuning* dilakukan pada pelatihan model yang meliputi beberapa parameter penting, seperti *epochs*, *image size*, *device*, dan *batch size*. Keluaran dari tahap pengembangan model deteksi ini adalah model dengan performa terbaik.

3.4.3 Pengembangan Sistem Perhitungan Lalu Lintas

Pengembangan sistem perhitungan lalu lintas dilakukan pada IDE Visual Studio Code dengan menggunakan bahasa pemrograman Python dan memanfaatkan Anaconda sebagai *package manager* untuk mengonfigurasi *environment* pada sistem perhitungan lalu lintas. Secara garis besar, struktur sistem perhitungan lalu lintas dapat terbagi menjadi 3 bagian, yakni *pre-processing*, *processing*, dan *post-processing*. Pada tahap pengembangan juga dilakukan *load* terhadap model yang telah terlatih. Sehingga keluaran dari tahapan ini adalah sistem perhitungan lalu lintas yang telah mengimplementasikan model deteksi terlatih. Keluaran dari tahapan ini adalah sistem perhitungan lalu lintas yang mampu melakukan deteksi, klasifikasi, dan perhitungan kendaraan, dengan keluaran berupa informasi mengenai jumlah kendaraan per jenis, jumlah total kendaraan, dan prediksi durasi lampu hijau untuk lalu lintas adaptif. Rumus yang digunakan dalam prediksi durasi lampu hijau tersebut diperoleh melalui persamaan yang telah teruji meningkatkan intensitas kelancaran lalu lintas sebesar 23% yang dilakukan pada penelitian (Gandhi, Solanki, Daptardar, & Baloorkar, 2020). Persamaan tersebut adalah sebagai berikut.

$$\frac{\text{Jumlah Kendaraan} \times \text{Rata-Rata Waktu Kendaraan Melintas}}{\text{Jumlah Jalur} + 1} \dots\dots\dots (3.5)$$

3.4.4 Pengujian Sistem

Pada tahapan ini dilakukan pengujian terhadap dua aspek. Pertama, evaluasi terhadap model deteksi dilakukan dengan menggunakan 10% data uji, kemudian dari evaluasi tersebut diperoleh *confusion matrix* yang dapat digunakan untuk mengukur performa dan kualitas model deteksi, metrik evaluasi tersebut meliputi *accuracy*, *precision*, *recall*, dan *f1 score*. Selain itu, analisis deteksi dan prediksi akan dilakukan terhadap sistem menggunakan studi kasus Jalan Dokter Djunjunan Kota Bandung.