

BAB III

METODOLOGI PENELITIAN

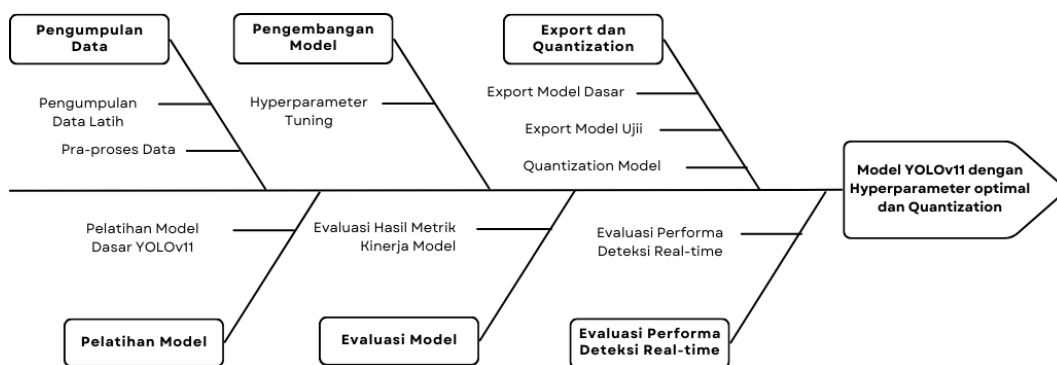
3.1 Roadmap Penelitian

Secara keseluruhan, rencana penelitian ini sejalan dengan *roadmap* di Universitas Siliwangi pada bagian *artificial Intelligence*. Topik penelitian yang dipilih yaitu *Computer Vision and Robotics* pada ranah *basic application of intelligent informatics*. *Roadmap* penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Roadmap Penelitian AIS (AIS, 2024)

Gambar 3.1 merupakan *Road Map* penelitian dari *AI Research Group* Universitas Siliwangi 2020-2024. Pemilihan *basic Application* pada penelitian ini yaitu *Computer Vision and Robotics* dimaksudkan agar terciptanya model yang lebih akurat tetapi tidak mengorbankan waktu *interface* aplikasi dalam mendeteksi objek secara langsung. *knowledge based system* penelitian akan digambarkan sebagai fishbone diagram pada Gambar 3.2.



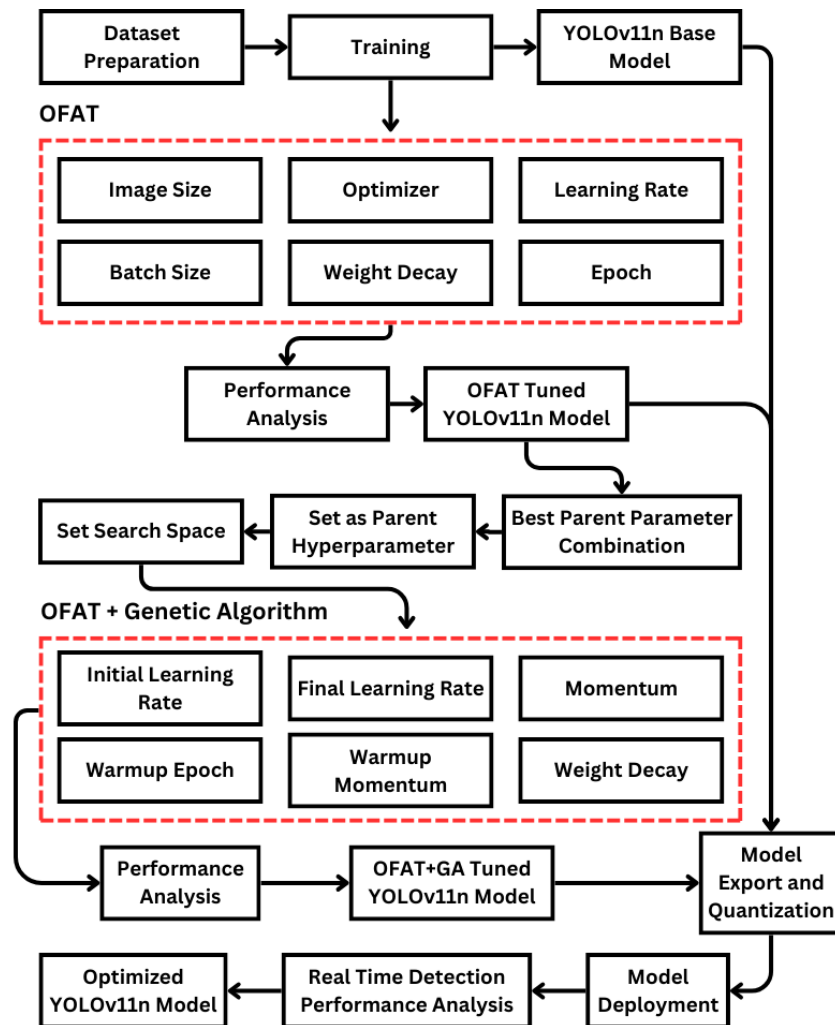
Gambar 3.2 *Fishbone* Penelitian

Gambar 3.2 merupakan *fishbone* merepresentasikan *knowledge based system* penelitian. Penelitian dirancang untuk mendapatkan model YOLOv11 dengan *Hyperparameter* yang optimal dan *Quantization* agar didapat model yang dapat berjalan dengan baik di perangkat *low-end* tetapi dengan tingkat akurasi yang tetap tinggi. Pengambilan data akan berupa data latih, data validasi dan data uji. Kemudian data akan di praproses. *Dataset* yang sudah dipraproses akan digunakan untuk pelatihan model YOLOv11 dasar. Setelah model dasar didapat, model itu akan dijadikan *pre-trained* model untuk pelatihan model baru dengan *hyperparameter tuning*. Akan didapat model dasar dan model-model yang sudah melalui *hyperparameter tuning*. Model-model ini akan dievaluasi kinerja menggunakan metrik kinerja yang dihasilkan model setelah proses pelatihan model.

Setelah model yang teroptimisasi didapat, model akan di-*export* ke format dasar dan format yang sudah terkuantisasi. Hasil model tadi akan diuji dan dievaluasi performa deteksi objek real-time dengan men-*deploy* model ke aplikasi yang sudah disiapkan.

3.2 Metode Penelitian

Metode penelitian mengadopsi metode-metode oleh (Casas dkk., 2023; Ramos dkk., 2024b, 2024a) untuk *Hyperparameter tuning* menerapkan metode One Factor at A Time (OFAT) untuk memahami pengaruh masing-masing hyperparameter terhadap performa deteksi berdasarkan karakteristik model dan dataset yang digunakan. Setelah diperoleh nilai awal yang optimal dari OFAT, proses dilanjutkan dengan penerapan Algoritma Genetika (GA) untuk mengeksplorasi ruang pencarian hyperparameter secara lebih luas dan menyeluruh, mengikuti pendekatan yang diusulkan oleh (Naik & Rudra, 2024; Suhail & Brindha, 2024). Metode oleh (Dalal dkk., 2024; Javed dkk., 2021; Kluska & Zięba, 2020; Liberatori dkk., 2022) digunakan sebagai metode *quantization* dengan menggabungkan kedua metode agar mendapat model yang dapat berjalan dengan baik di perangkat *low-end* tetapi dengan tingkat akurasi yang tetap tinggi pada model YOLOv11. Metode ini melibatkan beberapa langkah seperti yang ditunjukkan dalam Gambar 3.3 dengan penerapan *hyperparameter tuning* dan *quantization* untuk membuat model YOLOv11 dapat berjalan pada perangkat low-end dengan akurasi dan waktu inferensi yang optimal.



Gambar 3.3 Tahapan Penelitian

3.2.1 Studi Literatur

Studi Literatur dilakukan sebagai pedoman dalam penelitian dengan mempelajari dan mengkaji teori-teori melalui buku dan *survey paper* yang berkaitan dengan penelitian yang dilakukan. Kemudian mencari *technical paper*, laporan penelitian sebelumnya, dan karya ilmiah lainnya. Literatur-literatur tersebut dikaji secara menyeluruh dan membandingkan literatur dengan penelitian sebelumnya mencari perbedaan serta kekurangan dari hasil tersebut agar dapat dikaji kembali serta memaksimalkan penemuan yang ada.

3.2.2 Pengumpulan Data

Penggunaan data berasal dari pengambilan Gambar yang dilakukan secara mandiri berupa video kendaraan yang sedang berlalu-lalang di jalan. Kemudian data dari video dikonversi menjadi data Gambar dengan jeda pengambilan Gambar 5 Gambar per detik video. Pemilihan 5 Gambar per-detik ini didasari oleh penelitian (Lee & Hwang, 2022; Lu & Yuan, 2020; Qi dkk., 2023) yang menyatakan bahwa untuk kebutuhan deteksi objek, rentang 3-5 Gambar per detik video sudah mencukupi. Pemilihan rentang yang lebih tinggi akan mengakibatkan *delay* di saat model melakukan deteksi objek.

3.2.3 Praproses Data

Terdapat tiga proses di dalam praproses data, yaitu pelabelan dan anotasi, pembagian data, dan augmentasi Gambar.

1. Pelabelan dan anotasi dengan memberikan kelas pada setiap data Gambar. Kemudian, membuat koordinat *bounding box* pada objek yang berada pada Gambar dan memasukkannya ke dalam kelas dari objek tersebut.
2. Pembagian data menggunakan 70% data sebagai data latih, 20% data validasi, dan 10% data uji. Penggunaan 70% data untuk melatih model memastikan model memiliki data yang cukup untuk mempelajari pola, kelas gambar, dan fitur spasial yang ada (Tan dkk., 2021). Validasi 20% membantu menilai performa model selama proses pelatihan, memungkinkan pengoptimalan parameter model sebelum evaluasi akhir pada data uji (Ghasemzadeh dkk., 2023; Tan dkk., 2021). Data validasi membantu dalam penyesuaian parameter model sehingga model tidak overfitting pada data latih (Ghasemzadeh dkk.,

2023; Singh dkk., 2021). Hal ini membuat model dapat belajar fitur-fitur kelas tertentu lebih umum tanpa hanya mengikuti pola spesifik yang ada pada data latih.

3. Augmentasi Gambar dilakukan untuk meningkatkan keragaman data tanpa menambah jumlah Gambar. Gambar akan diproses dengan menyesuaikan *contrast* Gambar, lalu membalikkan gambar secara horizontal 180 derajat, perubahan perspektif, perubahan pencahayaan, perubahan *exposure*, dan *noise* Gambar. Hal ini berguna untuk meningkatkan akurasi dan pemahaman model pada objek yang akan dideteksi.

Ketiga proses ini dilakukan dengan bantuan *software* Roboflow.

3.2.4 Pelatihan Model YOLOv11 Dasar

Proses pelatihan model dasar menggunakan parameter *default* dari YOLOv11 tanpa perubahan. Proses ini perlu dilakukan untuk mendapatkan model dasar sebagai acuan perbandingan dengan model yang sudah dioptimalisasi nantinya. Selain itu, model ini juga yang akan jadi model dasar atau *pre-trained* model untuk proses *hyperparameter tuning*.

Model dasar YOLOv11 secara *default* memiliki *hyperparameter* pelatihan seperti pada Tabel 3.1.

Tabel 3.1 *Default Hyperparameter YOLOv11* (Derrenger dkk., 2024)

Model	YOLOv11n
<i>Image Size</i> (imgz)	640 (640x640 pixel)
<i>Optimizer</i>	Auto
<i>Learning Rate</i> (lr0)	0,01

<i>Batch Size (batch)</i>	16
<i>Epoch</i>	50
<i>Weight Decay</i>	0,0005

Pada Tabel 3.1, *Optimizer* akan diset berdasarkan jumlah *dataset* dan *batch size*. Ketika jumlah total iterasi masih dibawah 10 ribu iterasi maka secara otomatis *optimizer* yang dipilih *optimizer* Adamw. Ketika jumlah iterasi lebih besar dari itu, maka *optimizer* SGD yang akan terpilih (Derrenger dkk., 2024).

3.2.5 Hyperparameter Tuning

Proses *hyperparameter tuning* akan dilakukan menggunakan kombinasi dua metode yaitu *One Factor at A Time* (OFAT) dan juga Algoritma Genetika. Metode OFAT dipilih untuk menguji dampak langsung dari setiap *hyperparameter* terhadap hasil deteksi model YOLOv11 dasar yang sudah dilatih sebelumnya. Metode OFAT akan menguji satu persatu dampak dari perubahan *hyperparameter learning rate, batch size, image size dan optimizer*. Kombinasi nilai *hyperparameter* yang akan diuji ditunjukkan pada Tabel 3.2

Tabel 3.2 *Hyperparameter* dan *value* yang digunakan selama tahap *tuning*.

Model	YOLOv11n
<i>Image Size (imgz)</i>	640 (640x640 px), 320 (320x320 px)
<i>Optimizer</i>	SGD, RMSprop, Adam, Adamw, Nadam dan RAdam.
<i>Intial Learning Rate (lr0)</i>	0.0001, 0.001, 0.01
<i>Batch Size (batch)</i>	8, 16, 64, 95% GPU

<i>Weight Decay</i>	0.00005, 0.0005, 0.005
<i>Epoch</i>	25, 50, 75, 100

Setelah didapatkan kombinasi *hyperparameter* optimal metode OFAT, nilai tersebut akan digunakan sebagai *hyperparameter parent* bagi metode Algoritma Genetika serta mendefinisikan ruang pencarian hanya di sekitar nilai *hyperparameter parent* yang sudah didapatkan. *Hyperparameter* yang akan dioptimalkan adalah *learning rate*, *batch size*, *image size* dan *optimizer*. Dengan menggabungkan metode OFAT dan Algoritma Genetika maka proses optimasi *hyperparameter* dapat dilakukan secara lebih terarah dan efisien, karena ruang pencarian telah dibatasi pada area nilai yang terbukti menjanjikan oleh metode OFAT (Casas dkk., 2023; Ramos dkk., 2024b, 2024a). Hal ini memungkinkan Algoritma Genetika untuk mengeksplorasi kombinasi yang lebih optimal tanpa melakukan pencarian secara acak di seluruh ruang parameter.

Evaluasi terhadap setiap generasi model dalam proses evolusi akan menggunakan nilai *fitness* dengan menggunakan Persamaan 6. Penggunaan nilai *fitness* bertujuan untuk mendapatkan sistem deteksi objek yang tidak hanya memiliki akurasi tinggi dalam mendeteksi dan melokalisasi objek (melalui mAP), tetapi juga memiliki keseimbangan antara *precision* dan *recall*, yang direpresentasikan oleh F1-Score. Pendekatan ini relevan untuk aplikasi real-time, terutama dalam mendeteksi objek bergerak cepat, di mana baik ketepatan (*precision*) maupun keberhasilan pendeteksian (*recall*) sangat krusial untuk keandalan sistem (Alif, 2024; Benjumea dkk., 2021; Chen & Lin, 2019).

Jumlah iterasi percobaan yang akan dilakukan untuk Algoritma Genetika akan disesuaikan dengan kapasitas dari GPU yang digunakan serta batas waktu *runtime* Jupyter Notebook pada platform Google Colab atau Kaggle. Jumlah iterasi yang akan diuji ditunjukkan pada Tabel 3.3

Tabel 3.3 Iterasi tuning menggunakan Algoritma Genetika

Model	YOLOv11n
Iterasi Algoritma Genetika	200
<i>Optimizer</i>	<i>Best from OFAT</i>

3.2.6 Evaluasi Model

Evaluasi yang pertama dilakukan merupakan evaluasi model berdasarkan metrik kinerja yang dihasilkan setelah proses pelatihan model. Evaluasi akan dilakukan baik untuk model dasar maupun mode-model yang sudah dilakukan *hyperparameter tuning*. Nilai-nilai metrik kinerja yang dievaluasi merupakan nilai *precision*, *recall*, *F1-score*, mAP50, mAP50-95, waktu training dan waktu inferensi model dalam *millisecond* (ms).

3.2.7 Export Model dan Quantization

Setelah didapatkan model YOLOv11 baik model dasar dan model hasil *hyperparameter tuning*, model akan di-*export* ke dalam format Tflite yang bisa dibaca oleh perangkat Android. Proses ini akan menggunakan modul *export* yang sudah tersedia pada *library* Ultralytics dengan *argument* *half* untuk *export* model menggunakan *quantization Half Precision* dan *int8* untuk *export* model

menggunakan *quantization integer* (Derrenger dkk., 2024). Hasil akhirnya akan berupa tiga model dengan format tensor parameter dalam 32-bit dan 8-bit.

3.2.8 Deployment Model pada Aplikasi

Model-model yang sudah dalam format tflite akan di-*deploy* ke dalam aplikasi android pada perangkat yang sudah di tentukan. Aplikasi akan dibuat menggunakan *framework* Flutter dan bahasa pemrograman Dart.

3.2.9 Evaluasi Performa Deteksi Objek Real-time

Evaluasi akhir dari semua model berupa perbandingan antara model dasar dan model yang sudah dilakukan *hyperparameter tuning* serta perbandingan antara model dengan kuantisasi 32-bit, dan 8-bit. Proses evaluasi ini akan menguji model yang sudah di-*deploy* pada perangkat *low-end* dengan spesifikasi yang sudah ditentukan untuk mendeteksi objek secara *real-time*. Nilai-nilai yang akan dievaluasi pada model berupa F1-Score, mAP50, mAP50-95, nilai *fitness*, *Inference*, dan FPS deteksi.