

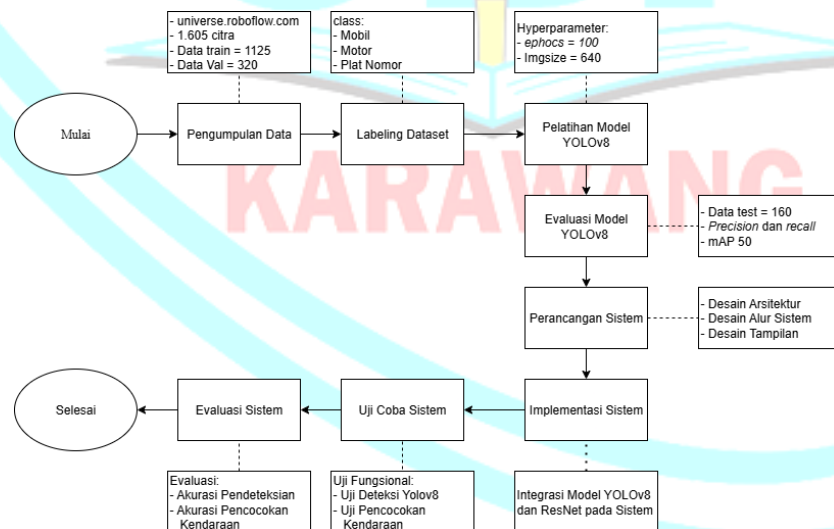
BAB III METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah data citra mobil dan motor dengan plat nomor kendaraan yang berada di Indonesia, yang digunakan untuk membangun sebuah sistem parkir cerdas yang memanfaatkan teknologi *computer vision* untuk mendeteksi dan verifikasi kendaraan yang akan masuk dan keluar area parkir untuk otomatisasi pemeriksaan kendaraan pada sistem parkir berbasis QR code. Penelitian ini berlangsung selama 3 bulan pada bulan Maret sampai Mei 2025.

3.2 Prosedur Penelitian

Penelitian ini dilaksanakan melalui serangkaian langkah sistematis yang meliputi pengumpulan data, labeling *dataset*, pelatihan model YOLOv8, evaluasi model YOLOv8, perancangan sistem, implementasi sistem, uji coba sistem, dan evaluasi sistem. Berikut adalah prosedur langkah-langkah penelitian yang dilakukan.



Gambar 3.1 *Flowchart* Prosedur Penelitian

3.2.1 Pengumpulan Data

Pengumpulan data merupakan langkah pertama dalam mengembangkan model *computer vision*. *Dataset* yang dikumpulkan berupa gambar-gambar yang diperlukan yaitu kendaraan mobil, dan motor dengan plat nomor



kendaraan yang berada di Indonesia. *Dataset* yang telah dikumpulkan yaitu sebanyak 1605 gambar, yang mencakup berbagai jenis kendaraan, dan warna untuk memastikan variasi yang cukup dalam pelatihan model. Gambar-gambar tersebut merupakan *dataset open source* yang didapatkan dari situs *roboflow* dengan format jpg dan dengan ukuran gambar yang sama yaitu 640x640 piksel.

Tabel 3.1 Contoh Gambar Kendaraan

No	Gambar	Keterangan
1		Mobil dengan Plat Nomor
2		Motor dengan Plat Nomor

3.2.2 Labeling Dataset

Dataset yang diperoleh dari tahap pengumpulan data, selanjutnya dilakukan proses labeling atau di anotasi per gambar dengan menggunakan situs *makesense.ai*. Situs ini adalah tools gratis yang digunakan untuk melabeli *dataset* gambar untuk keperluan *computer vision*, seperti *object detection* dan *image recognition*. Pada penelitian ini, dibuat tiga *class* yang dianotasi yaitu motor, mobil, dan plat nomor dengan menggunakan *bounding box rectangle* atau persegi panjang. Setelah proses anotasi selesai, hasil anotasi di export dengan format zip yang berisikan file-file YOLO untuk proses pelatihan model.

Tabel 3.2 Contoh Pelabelan *Class* Pada Citra

No	Gambar	<i>Class</i>
1		Mobil & Plat Nomor
2		Motor & Plat Nomor

3.2.3 Pelatihan Model YOLOv8

Model YOLOv8 *pre-trained* (telah dilatih sebelumnya) dilatih menggunakan *dataset* yang sudah dianotasi. Proses pelatihan bertujuan menghasilkan model YOLOv8 *fine tuning* yang mampu mendeteksi kendaraan dengan akurasi tinggi. *Fine-tuning* adalah proses di mana model YOLOv8 yang telah dilatih sebelumnya pada *dataset* umum seperti *yolov8n.pt* disesuaikan kembali atau di latih ulang dengan data pribadi agar bisa mengenali objek-objek yang lebih spesifik atau berbeda konteks sesuai dengan *dataset*. Pelatihan ini menggunakan *epoch* yang melibatkan iterasi berulang untuk meminimalkan *loss function* dan meningkatkan akurasi deteksi dengan ukuran input gambar yang disesuaikan. Berikut tahapan dalam pelatihan model dengan YOLOv8 dengan pseudo code:

Tabel 3.3 Algoritma Modeling YOLOv8

Algoritma: Modeling YOLOv8

Input: *Dataset* Kendaraan Indonesia

Output: Model YOLOv8 *fine-tuned*

1. Persiapkan folder *dataset* berisi gambar, dan label dalam format YOLO (.txt)
 2. Buat file konfigurasi (.yaml) yang berisi *path* folder dan daftar *class* yang dikenali
 3. Pilih arsitektur model YOLOv8 *pre-trained*
 4. Tentukan parameter pelatihan seperti *epochs*, *batch*, dan *imgsz*.
 5. Jalankan pelatihan model
 6. Selama pelatihan, sistem akan otomatis memproses data, menghitung *loss*, dan menyimpan model terbaik
 7. Evaluasi performa model menggunakan metrik mAP, *precision*, dan *recall*
 8. Simpan model hasil pelatihan (*best.pt*) ke lokasi yang telah ditentukan digunakan dalam pengujian atau *deployment*
-

3.2.4 Evaluasi Model YOLOv8

Setelah model dilatih, dilakukan evaluasi terhadap kinerja model menggunakan data uji. Untuk mengevaluasi kinerja model YOLO yang telah dilatih, dilakukan analisis terhadap metrik-metrik utama seperti *precision*, *recall*, dan mAP (*mean Average Precision*). Hasil evaluasi ini dianalisis untuk menentukan seberapa baik model mampu mengidentifikasi jenis kendaraan.

Tabel 3.4 *Confusion Metrics*

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

Rumus *Precision*, *Recall* & *mAP*:

1. *Precision*

Metrik ini mengukur seberapa akurat deteksi model terhadap objek yang diklasifikasikan sebagai benar. *Precision* dihitung sebagai rasio antara jumlah deteksi yang benar (*True Positives*) dan jumlah semua deteksi yang dihasilkan model (*True Positives + False Positives*). *Precision* memberikan gambaran tentang tingkat kesalahan deteksi objek yang tidak ada.

$$Precision = \frac{TP}{TP+FP}$$

(2)

Keterangan:

TP = *True Positive*

FP = *False Positive*

2. Recall

Recall mengukur sejauh mana model dapat mendeteksi semua objek yang relevan dalam gambar. *Recall* dihitung sebagai rasio antara jumlah deteksi yang benar (*True Positives*) dan jumlah objek yang sebenarnya ada dalam gambar (*True Positives + False Negatives*). *Recall* memberikan gambaran tentang kemampuan model dalam menangkap semua objek yang ada.

$$Recall = \frac{TP}{TP+FN}$$

(3)

Keterangan:

TP = *True positive*

FP = *False positive*

FN = *False negative*

3. mAP (*Mean Average Precision*)

mAP adalah metrik utama yang digunakan dalam *object detection* untuk mengukur performa model secara keseluruhan. mAP dihitung dengan menghitung rata-rata *precision* pada berbagai tingkat *recall*, kemudian mengambil nilai rata-rata dari seluruh kelas objek yang ada. mAP

memberikan gambaran tentang seberapa baik model secara keseluruhan dalam mendeteksi objek-objek di berbagai kelas.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Keterangan:

N = Jumlah Kelas

AP_i = *Average precision* untuk kelas i

3.2.5 Perancangan Sistem

Perancangan sistem mencakup desain arsitektur perangkat lunak, desain alur kerja sistem, dan desain antarmuka pengguna. Perancangan sistem ini dilakukan untuk memastikan bahwa seluruh alur sistem dapat berjalan dengan baik. Berikut penjelasan lebih rinci mengenai perancangan sistem dalam penelitian ini:

1. Desain Arsitektur Perangkat Lunak

Proses ini menentukan struktur dari suatu sistem perangkat lunak. Ini mencakup pemilihan prinsip dan pola desain yang digunakan untuk membangun dan mengembangkan sistem tersebut. Arsitektur perangkat lunak sistem dirancang dengan mempertimbangkan kebutuhan skalabilitas, dan kemudahan pemeliharaan. Sistem ini menggunakan pendekatan client-server, di mana komponen utama terdiri dari dua bagian utama yaitu frontend dan backend.

2. Desain Alur Kerja Sistem

Desain alur kerja sistem menggambarkan urutan langkah-langkah atau proses yang terjadi di dalam suatu sistem untuk menyelesaikan tugas tertentu. Alur kerja sistem berfokus pada proses pemeriksaan kendaraan yang terotomatisasi dari saat kendaraan masuk hingga keluar dari area parkir, dengan menggunakan kode QR untuk identifikasi.

3. Desain Antarmuka Pengguna

Pada proses ini merancang tampilan visual dan interaksi antara pengguna dengan sistem. Antarmuka pengguna dirancang agar mudah digunakan, memudahkan pengguna dalam memonitor status kendaraan.

3.2.6 Implementasi Sistem

Pada tahap ini, dilakukan pengerjaan untuk server yang dikembangkan menggunakan *framework* flask sebagai server yang diintegrasikan dengan *database* PostgreSQL sebagai penyimpanan data. Model *computer vision* yang sudah dibuat juga diintegrasikan dengan server untuk mendeteksi jenis kendaraan dari citra kamera. Selain itu, dikembangkan program antarmuka pengguna menggunakan GUI Tkinter yang berfungsi untuk menampilkan hasil olahan data dan mendukung proses pemeriksaan kendaraan secara interaktif.

3.2.7 Uji Coba Sistem

Sistem diuji untuk menilai kinerja dan tingkat akurasi secara menyeluruh. Pengujian dilakukan dengan mencakup dua aspek utama, yaitu kemampuan deteksi objek kendaraan dan keakuratan proses pencocokan kendaraan. Uji deteksi objek bertujuan untuk memastikan bahwa sistem dapat mengenali kendaraan secara tepat pada berbagai kondisi, sedangkan uji pencocokan citra kendaraan digunakan untuk mengevaluasi kemampuan sistem dalam mencocokkan data kendaraan yang terdeteksi dengan data referensi yang telah tersimpan.

3.2.8 Evaluasi Sistem

Data yang di peroleh dari uji coba sistem dianalisis untuk mengidentifikasi kelebihan dan kekurangan sistem. Analisis ini mencakup evaluasi terhadap tingkat keakuratan deteksi kendaraan, keandalan proses pencocokan gambar, serta faktor-faktor yang dapat mempengaruhi kinerja sistem secara keseluruhan. Dengan demikian, hasil analisis ini diharapkan dapat menjadi dasar untuk perbaikan dan pengembangan sistem di masa mendatang, agar dapat beroperasi secara optimal dan memenuhi kebutuhan implementasi pada lingkungan nyata.