Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah

Tesis

Farrel Muhammad Raihan Akhdan
24917011

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Sains Data
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2026

Lembar Pengesahan Pembimbing

Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah

Farrel Muhammad Raihan Akhdan
24917011

Yogyakarta, 5 Agustus 2026

Pembimbing I	Pembimbing II
Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D	Dr. Syarif Hidayat, S.Kom., M.I.T

Lembar Pengesahan Penguji

Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah

Farrel Muhammad Raihan Akhdan

24917011

Yogyakarta, 5 Agustus 2026

Tim Penguji

Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D Ketua	_____
Dr. Syarif Hidayat, S.Kom., M.I.T Anggota I	_____
Ir. Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D Anggota II	_____

Mengetahui

Ketua Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

Irving Vitra Paputungan, S.T., Ms.C., Ph.D

Pernyataan Keaslian

Saya yang bertanda tangan di bawah ini menyatakan bahwa tesis berjudul *Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah* adalah hasil karya saya sendiri, dan sepanjang pengetahuan saya tidak memuat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti sebaliknya, saya bersedia menerima sanksi.

Yogyakarta, 11 Agustus 2026



Farrel Muhammad Raihan Akhdan

24917011

Persembahan

Alhamdulillahirabbil'alamin, puji syukur ke hadirat Allah SWT atas segala rahmat, hidayah, dan kekuatan-Nya sehingga tesis ini dapat diselesaikan. Shalawat serta salam senantiasa tercurah kepada Rasulullah Muhammad SAW.

Karya ini saya persembahkan sebagai bentuk bakti dan rasa syukur yang tak terhingga kepada:

- Papa dan Mama tercinta, Edy El Khomsin dan Fitri Marisa. Terima kasih atas doa yang tak pernah putus di sepertiga malam, keringat, dan kasih sayang yang menjadi alasan utama saya bisa menyelesaikan studi Magister Informatika ini.
- Keluarga besar yang selalu memberikan dukungan moral dan doa dalam setiap langkah.
- Seluruh guru dan dosen yang telah ikhlas berbagi ilmu, semoga menjadi amal jariyah yang tak terputus.
- Sahabat-sahabat terdekat, yang senantiasa mendengarkan keluh kesah, memberikan semangat, dan menemani di saat suka maupun duka.
- Teman-teman seperjuangan, atas setiap diskusi, bantuan, dan kebersamaan yang membuat perjalanan panjang penyusunan tesis ini menjadi lebih ringan dan bermakna.

(Yogyakarta, 2026)

Farrel Muhammad Raihan Akhdan

Kata Pengantar

Assalamu'alaikum warahmatullahi wabarakatuh

Puji syukur kehadirat Allah SWT atas segala limpahan rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan tesis ini. Shalawat serta salam senantiasa tercurah kepada junjungan kita, Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman. Tesis berjudul Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah disusun sebagai salah satu syarat memperoleh gelar Magister pada Program Studi Magister Informatika, Universitas Islam Indonesia.

Penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU., ASEAN Eng. selaku Rektor Universitas Islam Indonesia, yang telah memberikan kesempatan untuk menimba ilmu di kampus ini.
2. Bapak Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku dosen pembimbing pertama, atas kesabaran dalam memberikan arahan, masukan, serta bimbingan hingga tesis ini selesai.
3. Bapak Dr. Syarif Hidayat, S.Kom., M.I.T. selaku dosen pembimbing kedua, atas ketabahannya dalam membimbing tesis ini hingga selesai.
4. Teman-teman di Magister Informatika yang senantiasa memberikan dukungan dan semangat kepada penulis untuk menyelesaikan tesis ini.

Penulis menyadari tesis ini masih memiliki kekurangan, baik dari segi penulisan maupun kedalaman materi. Oleh karena itu, kritik dan saran yang membangun dari semua pihak sangat penulis harapkan demi penyempurnaan penelitian di masa mendatang. Akhir kata, semoga tesis ini dapat memberikan manfaat, inspirasi, dan kontribusi positif bagi perkembangan ilmu pengetahuan, khususnya di bidang visi komputer dan infrastruktur jalan.

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 12 Agustus 2026



Farrel Muhammad Raihan Akhdan

Abstrak

Perbandingan YOLO11n dan YOLOv8n untuk Deteksi Kerusakan Jalan Multi-Kelas pada Dataset Multi-Wilayah Indonesia: Studi Efisiensi Data dan Potensi Penerapan Berbiaya Rendah

Kerusakan jalan seperti lubang (pothole) dan retak mengancam keselamatan lalu lintas dan mempercepat kerusakan struktur perkerasan, sementara inspeksi manual masih lambat, mahal, dan sulit diterapkan pada jaringan jalan yang luas. Keberhasilan solusi deteksi otomatis pada konteks ini dinilai dari dua parameter yang saling melengkapi, yaitu akurasi deteksi (mAP) dan efisiensi komputasi (parameter, FLOPs, kecepatan inferensi), dengan pemantauan jalan berbiaya rendah secara berkala—bukan deteksi waktu-nyata pada kendaraan bergerak—sebagai use case utama. Penelitian ini membandingkan YOLOv8n, baseline satu-tahap yang matang dan banyak diadopsi, dengan YOLO11n, generasi lebih baru yang menggunakan backbone lebih efisien dan modul atensi tambahan, untuk menilai manfaatnya pada deteksi kerusakan jalan multi-kelas (pothole, longitudinal_crack, dan alligator_crack) pada kondisi jalan Indonesia. Dataset yang digunakan berupa gabungan 4.088 citra jalan Indonesia yang memadukan 767 citra primer lapangan dari lima provinsi (Kementerian PUPR) dengan 3.321 citra sekunder publik berdomain Indonesia. Evaluasi dilakukan melalui rancangan terkendali 2 x 2 yang menyilangkan dua arsitektur dengan dua kondisi data, yaitu data primer saja serta data gabungan primer dan sekunder. Seluruh model dilatih dengan konfigurasi yang identik dan diuji pada subset pengujian primer terkunci yang sama sebanyak 140 citra, disertai pencegahan kebocoran data melalui deduplikasi berbasis hash konten, sehingga perbandingan bersifat adil. Setiap konfigurasi dilatih dengan tiga seed acak dan hasilnya dilaporkan sebagai rata-rata beserta simpangan baku. Hasil menunjukkan keunggulan YOLO11n atas baseline paling menonjol pada rezim data terbatas, yaitu mAP50 0,354 ± 0,006 dibanding 0,314 ± 0,015, sekaligus lebih stabil antar-seed. Keunggulan tersebut menyempit ketika data pelatihan diperbanyak dengan data sekunder karena YOLOv8n memperoleh peningkatan yang lebih besar, sehingga mengindikasikan YOLO11n lebih efisien-data. Konfigurasi terbaik, yaitu YOLO11n dengan data gabungan, mencapai mAP50 0,369 ± 0,017 dan mAP50-95 0,151 ± 0,013. YOLO11n juga lebih hemat parameter (14,0%), FLOPs (21,2%), dan ukuran model (12,7%) meskipun waktu inferensinya sedikit lebih lama, dengan kedua model beroperasi jauh di atas ambang waktu-

nyata. Analisis kesalahan menunjukkan kegagalan didominasi objek yang tidak terdeteksi, bukan kekeliruan antar-kelas, dengan pothole sebagai kelas paling menantang yang tidak terbantu penambahan data sekunder sehingga mengindikasikan kesenjangan domain. Meskipun demikian, efisiensi komputasi dan kestabilan performa yang ditunjukkan YOLO11n tetap mendukung potensinya sebagai solusi pemantauan kerusakan jalan yang lebih murah dan dapat diskalakan dibandingkan inspeksi manual, sekalipun akurasi mutlak masih perlu ditingkatkan pada penelitian lanjutan.

Kata kunci: deteksi kerusakan jalan, YOLO11n, YOLOv8n, pembelajaran mendalam, efisiensi data, pemantauan infrastruktur berbiaya rendah

Abstract

Comparison of YOLO11n and YOLOv8n for Multi-Class Road Damage Detection on a Multi-Region Indonesian Dataset: A Study of Data Efficiency and Low-Cost Deployment Potential

Road damage such as potholes and cracks threatens traffic safety and accelerates the deterioration of pavement structure, while manual inspection remains slow, costly, and difficult to apply across extensive road networks. Success of an automated detection solution in this context is assessed through two complementary parameters, namely detection accuracy (mAP) and computational efficiency (parameters, FLOPs, inference speed), with periodic low-cost road monitoring—rather than real-time detection on a moving vehicle—as the primary use case. This study compares YOLOv8n, a mature and widely adopted one-stage baseline, with YOLO11n, a newer generation that uses a more efficient backbone and an additional attention module, to assess its benefit for multi-class road damage detection (pothole, longitudinal_crack, and alligator_crack) under Indonesian road conditions. The dataset used is a combined set of 4,088 Indonesian road images, merging 767 primary field images from five provinces (Ministry of Public Works and Housing) with 3,321 secondary public images of Indonesian domain. Evaluation was conducted through a controlled 2×2 design crossing two architectures with two data conditions, namely primary data only and combined primary-and-secondary data. All models were trained with identical configurations and evaluated on the same locked primary test subset of 140 images, with data leakage prevention through content-hash deduplication, ensuring a fair comparison. Each configuration was trained with three random seeds, and results are reported as mean and standard deviation. Results show that YOLO11n's advantage over the baseline is most pronounced in the limited-data regime, with mAP50 of 0.354 ± 0.006 compared to 0.314 ± 0.015, along with greater stability across seeds. This advantage narrows when training data is expanded with secondary data, as YOLOv8n obtains a larger improvement, indicating that YOLO11n tends to be more data-efficient. The best configuration, YOLO11n with combined data, achieves mAP50 of 0.369 ± 0.017 and mAP50-95 of 0.151 ± 0.013. YOLO11n is also more efficient in parameters (14.0%), FLOPs (21.2%), and model size (12.7%), although its inference time is slightly longer, with both models operating well above the real-time threshold. Error analysis shows that failures are dominated by undetected objects rather than

inter-class confusion, with pothole being the most challenging class, one that does not benefit from the addition of secondary data, indicating a possible domain gap. Nonetheless, the computational efficiency and stability demonstrated by YOLO11n still support its potential as a cheaper and more scalable alternative to manual inspection for road damage monitoring, even though absolute accuracy warrants further improvement in future work.
Keywords: road damage detection, YOLO11n, YOLOv8n, deep learning, data efficiency, low-cost infrastructure monitoring

Daftar Isi

Lembar Pengesahan Pembimbing i
Lembar Pengesahan Penguji ii
Pernyataan Keaslian iii
Persembahan iv
Kata Pengantar v
Abstrak vi
Abstract viii
Daftar Isi x
Daftar Tabel xiii
Daftar Gambar xiv
Glosarium xv
BAB I Pendahuluan 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 4
1.4 Manfaat Penelitian 4
1.5 Batasan Masalah 5
1.6 Metode Penelitian 5
1.7 Sistematika Penulisan 7
BAB II Tinjauan Pustaka 8
2.1 Deteksi Kerusakan Jalan Berbasis YOLO pada Benchmark Publik 8
2.2 Penyempurnaan Arsitektur Berbasis Mekanisme Atensi 8
2.3 Studi Perbandingan Antar-Versi YOLO 9
2.4 Deteksi Lubang 9
2.5 Penelitian pada Konteks Indonesia 9

2.6 Sintesis Keterbatasan dan Posisi Penelitian 10
2.7 Landasan Teori 11
2.7.1 Deteksi Objek Satu-Tahap 11
2.7.2 Arsitektur Keluarga YOLO 12
2.7.3 YOLOv8n dan YOLO11n 12
2.7.4 Karakteristik Kerusakan Jalan 14
2.7.5 Metrik Evaluasi 15
2.7.6 Kompleksitas dan Kecepatan Model 16
2.7.7 Prapemrosesan dan Augmentasi Data 16
2.7.8 Pembagian Data dan Pencegahan Kebocoran 17
2.7.9 Istilah dan Proses Pelatihan 17
2.7.10 Variabilitas Antar-Seed dan Simpangan Baku 18
2.7.11 Confusion Matrix, Kurva Evaluasi, dan Overfitting 18
BAB III Metodologi Penelitian 20
3.1 Penyiapan Data 21
3.1.1 Sumber dan Komposisi Data 22
3.1.2 Definisi Kelas dan Anotasi 23
3.1.3 Pembagian Data 24
3.1.4 Prapemrosesan dan Augmentasi 25
3.2 Model Deteksi 26
3.3 Rancangan Eksperimen dan Justifikasinya 28
3.4 Konfigurasi Pelatihan 30
3.5 Metrik Evaluasi 31
BAB IV Hasil dan Pembahasan 34
4.1 Perbandingan Performa Keseluruhan 34
4.2 Hasil Per-Seed 35
4.3 Analisis Per-Kelas 39

4.4 Efisiensi Model 40
4.5 Analisis Kesalahan 41
4.6 Efisiensi Data sebagai Keunggulan Utama YOLO11n 46
4.7 Pengaruh Penambahan Data Sekunder 47
4.8 Implikasi untuk Pemantauan Berbiaya Rendah 48
4.9 Nilai Absolut dan Konteks Dataset 49
4.10 Keterbatasan 50
4.11 Jawaban atas Pertanyaan Penelitian 51
4.12 Evaluasi Tambahan: Tracking Objek pada Video 52
BAB V Kesimpulan dan Saran 55
5.1 Kesimpulan 55
5.2 Saran 57
Daftar Pustaka 59

Daftar Tabel

Tabel 2.1 Sintesis Penelitian Terdahulu dan Keterbatasannya 10
Tabel 3.1 Komposisi Dataset Menurut Sumber dan Kelas..... 23
Tabel 3.2 Ringkasan Karakteristik Arsitektur YOLOv8n dan YOLO11n 27
Tabel 3.3 Empat Konfigurasi Hasil Persilangan Rancangan 2×2..... 28
Tabel 3.4 Pemetaan Eksperimen terhadap Pertanyaan Penelitian 29
Tabel 3.5 Hyperparameter Pelatihan 30
Tabel 3.6 Kompleksitas dan Kecepatan Model pada GPU dan CPU (imgsz 960, batch 1) 32
Tabel 4.1 Performa Deteksi Keseluruhan (rata-rata ± simpangan baku, n = 3 seed) 34
Tabel 4.2 Performa Keseluruhan per Seed (12 Run)..... 35
Tabel 4.3 Performa Kelas Pothole per Seed (12 Run)..... 36
Tabel 4.4 Performa Kelas Longitudinal Crack per Seed (12 Run)..... 37
Tabel 4.5 Performa Kelas Alligator Crack per Seed (12 Run)..... 38
Tabel 4.6 Performa Per Kelas (rata-rata ± simpangan baku, n = 3 seed) 39
Tabel 4.7 Perbandingan Kecepatan Inferensi pada GPU dan CPU 40
Tabel 4.8 Perbandingan Jumlah Objek Unik Hasil Tracking (ByteTrack) terhadap Ground Truth 53
Tabel 4.9 Perbandingan Jumlah Objek Unik Hasil Tracking (BoT-SORT) terhadap Ground Truth 53

Daftar Gambar

Gambar 2.1 Ilustrasi Perbedaan Arsitektur YOLOv8n dan YOLO11n 13
Gambar 3.1 Alur Metodologi Penelitian 20
Gambar 3.2 Alur Tahapan Penyiapan Data 21
Gambar 3.3 Contoh Citra Tiap Kelas Kerusakan 23
Gambar 4.1 Confusion matrix ternormalisasi, rata-rata 3 seed, keempat konfigurasi 41
Gambar 4.2 Kurva Precision-Recall YOLO11n pada subset pengujian primer, dari seed representatif 43
Gambar 4.3 Kurva pelatihan (loss & metrik per epoch), dari seed representatif 43
Gambar 4.4 Contoh Kasus False Negative pada Kelas Pothole 44
Gambar 4.5 Contoh Kasus Tiga Pothole Terlewat Sekaligus 45
Gambar 4.6 Contoh Kasus Dua Pothole dengan Ukuran Berbeda Terlewat 45

Glosarium

Daftar berikut merangkum istilah teknis yang dipakai berulang kali pada naskah ini beserta definisi singkatnya.

Istilah	**Definisi**
YOLO (You Only Look Once)	Keluarga arsitektur deteksi objek satu-tahap yang memprediksi kotak pembatas dan kelas objek secara langsung dari citra dalam sekali proses inferensi.
mAP (mean Average Precision)	Rata-rata nilai Average Precision (AP) di seluruh kelas objek; metrik utama akurasi deteksi yang dipakai pada penelitian ini.
mAP50	Nilai mAP yang dihitung pada ambang IoU tunggal sebesar 0,5.
mAP50-95	Nilai mAP yang dirata-ratakan pada rentang ambang IoU 0,5 hingga 0,95 dengan kelipatan 0,05; metrik yang lebih ketat daripada mAP50.
IoU (Intersection over Union)	Rasio luas irisan terhadap luas gabungan antara kotak pembatas prediksi dan kotak pembatas acuan (ground truth), dipakai untuk menilai ketepatan lokalisasi.
FLOPs (Floating Point Operations)	Jumlah operasi hitung titik-mengambang yang dibutuhkan model untuk satu kali inferensi; indikator kompleksitas komputasi.
GFLOPs	FLOPs dalam satuan miliar (giga).
FPS (Frames per Second)	Jumlah citra yang dapat diproses model per detik; indikator kecepatan inferensi.
CNN (Convolutional Neural Network)	Jaringan saraf tiruan berbasis operasi konvolusi yang menjadi fondasi sebagian besar arsitektur deteksi objek modern, termasuk YOLO.
GPU (Graphics Processing Unit)	Unit pemroses grafis yang dipakai untuk mempercepat pelatihan dan inferensi model pembelajaran mendalam.
CPU (Central Processing Unit)	Unit pemroses utama komputer; pada penelitian ini dipakai untuk mengukur kecepatan inferensi pada perangkat tanpa akselerasi GPU khusus.
RTX	Seri GPU konsumen NVIDIA; pada penelitian ini dipakai RTX 3050 untuk pelatihan dan pengujian model.

PUPR	Kementerian Pekerjaan Umum dan Perumahan Rakyat; sumber sebagian citra jalan primer pada dataset penelitian ini.
GNSS (Global Navigation Satellite System)	Sistem penentuan posisi berbasis satelit, disebut dalam konteks metadata lokasi pengambilan citra.
RMSE (Root Mean Square Error)	Akar dari rata-rata kuadrat galat; salah satu metrik yang disebut pada tinjauan pustaka terkait pengukuran kekasaran/kerusakan permukaan jalan.
SPPF (Spatial Pyramid Pooling - Fast)	Modul pada backbone YOLO yang menggabungkan fitur pada berbagai skala reseptif secara efisien.
C2f	Modul blok fitur pada backbone/neck YOLOv8 (Cross Stage Partial dengan 2 konvolusi, fast) yang menggabungkan beberapa cabang konvolusi ringan.
C3k2	Modul blok fitur pada backbone/neck YOLO11 sebagai varian dari C2f dengan kernel konvolusi yang dapat dikonfigurasi.
C2PSA	Modul atensi pada YOLO11 (Cross Stage Partial with Position-Sensitive Attention) yang memperkuat representasi fitur pada area penting citra.
HSV (Hue, Saturation, Value)	Ruang warna yang dipakai pada augmentasi data untuk mengubah rona, saturasi, dan kecerahan citra pelatihan.
seed	Nilai awal pembangkit bilangan acak yang menentukan inisialisasi bobot dan urutan data pelatihan; penelitian ini memakai tiga seed berbeda per konfigurasi untuk menilai variabilitas hasil.
best.pt	Berkas checkpoint bobot model dengan performa validasi terbaik selama pelatihan; dipakai untuk pengujian akhir pada penelitian ini.

BAB I
Pendahuluan

1.1 Latar Belakang

Jaringan jalan merupakan salah satu aset infrastruktur yang paling menentukan produktivitas ekonomi, kelancaran logistik, dan mobilitas publik. Kerusakan permukaan jalan seperti lubang (pothole) dan retak tidak hanya mempercepat degradasi struktur perkerasan, tetapi juga meningkatkan risiko kecelakaan lalu lintas dan kerusakan kendaraan, sehingga deteksi dan pemeliharaan yang tepat waktu menjadi prioritas bagi pengelola jalan (Yu et al., 2024a; Zhang et al., 2024). Kerusakan jalan pun cenderung bersifat progresif: retak halus yang tidak segera ditangani dapat menjadi jalan masuk air ke lapisan bawah perkerasan, mempercepat pelemahan struktur, dan pada akhirnya berkembang menjadi lubang yang biaya perbaikannya jauh lebih besar. Karena itu, deteksi dini bukan sekadar persoalan teknis, melainkan berkaitan langsung dengan efisiensi anggaran pemeliharaan. Di Indonesia, dengan total panjang jaringan jalan mencapai sekitar 546.116 km dan sekitar 31% di antaranya berada dalam kondisi rusak hingga rusak berat (Badan Pusat Statistik, 2022), kebutuhan akan pemantauan kondisi jalan yang efisien menjadi semakin mendesak. Skala jaringan yang sedemikian luas menyiratkan bahwa pendekatan pemantauan apa pun harus dapat diterapkan secara masif dengan biaya yang terjangkau, karena metode yang mahal atau lambat tidak akan mampu menjangkau keseluruhan jaringan dalam waktu yang wajar.

Inspeksi jalan konvensional masih sangat bergantung pada survei manual dan penilaian visual oleh tenaga ahli. Prosedur semacam ini bersifat padat karya, memakan waktu, subjektif, dan sulit diterapkan pada jaringan jalan yang luas atau tersebar secara geografis (Yu et al., 2024a). Penilaian yang bergantung pada persepsi manusia juga rentan terhadap perbedaan antar-surveyor, sehingga hasil yang sama belum tentu diperoleh untuk ruas jalan yang sama pada waktu yang berbeda. Keterbatasan tersebut mendorong pergeseran ke arah inspeksi berbasis visi komputer, yaitu model pembelajaran mendalam mendeteksi dan mengklasifikasikan kerusakan jalan langsung dari citra, sehingga menghasilkan penilaian yang lebih cepat dan konsisten dengan biaya operasional yang lebih rendah (Zhang et al., 2024). Pendekatan otomatis ini menjanjikan penilaian yang dapat diulang dan diskalakan, dua sifat yang justru sulit dipenuhi oleh inspeksi manual pada jaringan jalan yang luas.

Di antara pendekatan pembelajaran mendalam, keluarga You Only Look Once (YOLO) banyak digunakan untuk deteksi kerusakan jalan karena keseimbangannya antara akurasi dan kecepatan inferensi pada deteksi satu tahap secara waktu-nyata (Ali & Zhang, 2024; Sapkota et al., 2025). Selain itu, YOLO merupakan salah satu dari sedikit keluarga arsitektur deteksi objek yang masih dikembangkan secara aktif dengan perilisan generasi baru secara berkelanjutan, termasuk pada rezim model ringan (nano), sehingga menjadi kandidat yang tepat untuk menguji apakah peningkatan arsitektur antargenerasi memberikan manfaat nyata pada konteks deteksi kerusakan jalan. Berbagai generasi YOLO telah diadaptasi secara luas untuk tugas deteksi kerusakan perkerasan; sejumlah studi berbasis YOLOv5 dan YOLOv8 melaporkan performa yang kuat untuk deteksi retak dan lubang (Ma & Chen, 2025; Wang et al., 2024b; Zeng & Zhong, 2024). Rilis YOLO11 pada tahun 2024 memperkenalkan penyempurnaan arsitektur yang ditujukan untuk meningkatkan ekstraksi fitur dan efisiensi komputasi sekaligus menurunkan jumlah parameter, karakteristik yang menarik untuk penerapan pada perangkat dengan sumber daya terbatas (Khanam & Hussain, 2024; Ultralytics, 2024). Namun, sebagaimana ditunjukkan oleh studi perbandingan lintas-versi, generasi yang lebih baru tidak selalu unggul pada semua jenis tantangan, sehingga manfaat sebenarnya dari peningkatan arsitektur perlu diuji pada domain yang menjadi sasaran, bukan diandaikan (Jegham et al., 2024).

Meskipun demikian, terdapat tiga kesenjangan yang masih terbuka. Pertama, dari sisi cakupan data, benchmark publik yang paling banyak digunakan, yaitu RDD2022, memuat 47.420 citra jalan dari enam negara dan tidak mencakup Indonesia maupun kawasan Asia Tenggara tropis lainnya (Arya et al., 2021, 2024). Perbedaan iklim tropis, praktik konstruksi, dan pola beban lalu lintas berpotensi menghasilkan karakteristik kerusakan dan kondisi akuisisi citra yang berbeda, sehingga performa model yang dilatih pada distribusi lain belum tentu bertahan pada kondisi jalan Indonesia. Kedua, penelitian deteksi kerusakan jalan pada konteks Indonesia memang telah dilakukan (Denaro & Lim, 2025; Ismail & Rahmadewi, 2025; Jonathan et al., 2026; Sasmito et al., 2023; Takyudin et al., 2025), tetapi umumnya terbatas pada satu kota atau satu wilayah administratif, sebagian besar memanfaatkan dataset publik daring alih-alih citra lapangan instansi pengelola jalan, dan mengevaluasi satu model tunggal tanpa pembanding arsitektur yang setara. Ketiga, dari sisi metodologi, klaim keunggulan varian YOLO yang lebih baru kerap tercampur (confounded) karena studi yang berbeda menggunakan dataset, prapemrosesan, augmentasi, dan hyperparameter yang berbeda. Sebagian studi bahkan menetapkan pilihan arsitektur berdasarkan asumsi kualitatif,

misalnya anggapan bahwa strategi pemangkasan dinamis pada YOLO11 lebih memprioritaskan objek berukuran besar sehingga kurang spesifik untuk ekstraksi fitur retak (Xu et al., 2025), tanpa disertai pengujian terkendali yang menyetarakan seluruh kondisi eksperimen. Ketiga kesenjangan ini saling berkaitan: ketiadaan data lapangan Indonesia yang representatif mempersulit pengujian yang relevan secara kontekstual, sementara ketiadaan perbandingan yang terkendali membuat klaim keunggulan antararsitektur sulit dinilai secara meyakinkan.

Untuk menjawab kesenjangan tersebut, penelitian ini mengevaluasi arsitektur YOLO11n generasi lebih baru terhadap baseline YOLOv8n untuk deteksi kerusakan jalan multi-kelas pada kondisi jalan Indonesia, menggunakan dataset gabungan 4.088 citra yang memadukan data primer lapangan dari lima provinsi dengan data sekunder publik berdomain Indonesia (Basily, 2020). Kedua arsitektur dilatih dalam konfigurasi eksperimen yang identik dan dievaluasi pada subset pengujian primer yang dikunci, sehingga perbedaan performa dapat diatribusikan pada arsitektur atau pada data pelatihan, bukan pada pengaturan eksperimen. Dengan rancangan demikian, penelitian ini sekaligus menyediakan perbandingan antar-versi yang terkendali pada domain deteksi kerusakan jalan lapangan Indonesia, sebuah kombinasi yang belum dijumpai pada penelitian terdahulu.

1.2 Rumusan Masalah

Masalah yang melatari penelitian ini dapat diuraikan ke dalam tiga lapis yang saling berkaitan. Pada lapis praktis, inspeksi kerusakan jalan di Indonesia masih bertumpu pada survei manual yang lambat, mahal, dan subjektif, sementara solusi otomatis yang hendak diterapkan harus cukup ringan untuk dijalankan pada perangkat berbiaya rendah. Pada lapis kontekstual, model deteksi yang tersedia umumnya dikembangkan dan diuji pada benchmark internasional yang tidak memuat kondisi jalan Indonesia, sedangkan penelitian domestik yang telah ada terbatas pada cakupan satu wilayah dan sebagian besar tidak menggunakan citra lapangan instansi pengelola jalan. Pada lapis metodologis, klaim keunggulan arsitektur YOLO generasi yang lebih baru sulit dinilai karena perbandingan antarstudi tercampur dengan perbedaan dataset dan pengaturan eksperimen.

Ketiga lapis masalah tersebut menuntut jawaban yang berbeda. Lapis metodologis menuntut perbandingan arsitektur pada kondisi yang benar-benar disetarakan. Lapis kontekstual menuntut penilaian atas seberapa besar penambahan data berdomain Indonesia memperbaiki performa pada kondisi lapangan. Lapis praktis menuntut pengukuran biaya

komputasi model dan pemahaman atas jenis kegagalan yang dominan. Berdasarkan uraian tersebut, penelitian ini dirumuskan ke dalam empat pertanyaan berikut.

1. Bagaimana perbandingan performa arsitektur YOLO11n generasi lebih baru terhadap baseline YOLOv8n dalam mendeteksi kerusakan jalan multi-kelas (pothole, longitudinal_crack, dan alligator_crack) pada kondisi jalan Indonesia, ketika keduanya dilatih dengan konfigurasi yang identik?
2. Sejauh mana penambahan data sekunder berdomain Indonesia memengaruhi performa deteksi kedua arsitektur pada data uji lapangan yang sama?
3. Bagaimana perbandingan efisiensi kedua arsitektur, yaitu jumlah parameter, FLOPs, ukuran model, dan kecepatan inferensi, dalam konteks penerapan pemantauan jalan berbiaya rendah?
4. Pada kondisi apa kedua model paling sering gagal, dan jenis kesalahan apa yang dominan, yaitu objek terlewat atau kesalahan antar-kelas?

1.3 Tujuan Penelitian

Sejalan dengan rumusan masalah, penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa deteksi YOLO11n dan YOLOv8n untuk kerusakan jalan multi-kelas pada kondisi jalan Indonesia dalam konfigurasi pelatihan yang identik; menganalisis pengaruh penambahan data sekunder berdomain Indonesia terhadap performa kedua arsitektur; membandingkan efisiensi kedua arsitektur dari sisi kompleksitas model dan kecepatan inferensi untuk mendukung potensi penerapan berbiaya rendah; serta menganalisis pola dan jenis kesalahan dominan kedua model melalui metrik deteksi, confusion matrix, dan kurva Precision-Recall.

1.4 Manfaat Penelitian

Penelitian ini memberikan empat manfaat. Pertama, tersedianya perbandingan yang adil dan terkendali antara YOLO11n dan YOLOv8n dalam kondisi pelatihan yang sepenuhnya identik, sehingga perbedaan yang teramati mencerminkan arsitektur dan bukan pengaturan eksperimen. Kedua, tersusunnya dataset gabungan multi-wilayah untuk konteks Indonesia yang memadukan data primer lapangan dari lima provinsi dengan dataset publik berdomain Indonesia, sehingga evaluasi mencerminkan kondisi jalan Indonesia yang belum terwakili pada benchmark yang ada. Ketiga, diperolehnya bukti empiris mengenai pengaruh penambahan data sekunder melalui rancangan ablasi yang diukur pada target pengujian primer yang sama. Keempat, tersedianya indikator berorientasi penerapan berupa ukuran

model, jumlah parameter, FLOPs, dan kecepatan inferensi sebagai dasar penilaian potensi pemantauan kondisi jalan berbiaya rendah berbasis perangkat tepi.

1.5 Batasan Masalah

Ruang lingkup penelitian ini dibatasi sebagai berikut. Deteksi dibatasi pada tiga kelas kerusakan, yaitu pothole, longitudinal_crack, dan alligator_crack, sehingga jenis kerusakan lain seperti retak lateral tidak termasuk dalam tugas deteksi. Data primer berasal dari kantor wilayah Kementerian Pekerjaan Umum dan Perumahan Rakyat di lima provinsi, yaitu Bali, Banten, Daerah Istimewa Yogyakarta, Jawa Tengah, dan Jawa Timur, sehingga generalisasi ke wilayah lain berada di luar cakupan pengujian. Arsitektur yang dibandingkan dibatasi pada varian nano dari YOLOv8 dan YOLO11 tanpa modifikasi arsitektur, karena penelitian ini menargetkan penerapan berbiaya rendah. Seluruh pelatihan dan pengukuran efisiensi dilakukan pada satu perangkat keras yang sama, yaitu GPU NVIDIA RTX 3050, sehingga nilai kecepatan inferensi bersifat relatif terhadap perangkat tersebut. Perbandingan arsitektur dibatasi pada dua generasi dari satu keluarga YOLO, yaitu YOLOv8n dan YOLO11n, agar perbandingan dapat dilakukan secara terkendali dan setara dalam satu kerangka kerja; perbandingan lintas-keluarga dengan arsitektur ringan lain seperti MobileNet-SSD atau EfficientDet berada di luar cakupan penelitian ini dan menjadi arah pengembangan lanjutan. Penelitian ini juga berfokus pada perbandingan model dan tidak mencakup pengembangan aplikasi atau penerapan langsung pada perangkat tepi.

1.6 Metode Penelitian

Penelitian ini disusun melalui tahapan berikut:

1. Penyiapan Data

   Tahap ini mencakup pengumpulan 767 citra primer lapangan dari lima provinsi (Kementerian PUPR) dan 3.321 citra sekunder publik berdomain Indonesia, sehingga total dataset gabungan berjumlah 4.088 citra. Ketiga kelas kerusakan (pothole, longitudinal_crack, dan alligator_crack) didefinisikan mengikuti taksonomi distres perkerasan, dengan anotasi data sekunder yang diselaraskan melalui harmonisasi label dua tahap. Data kemudian dibagi secara terstratifikasi menjadi subset pelatihan, validasi, dan pengujian, disertai deduplikasi berbasis hash konten untuk mencegah kebocoran data. Seluruh citra selanjutnya melewati prapemrosesan letterbox 960×960 piksel dan augmentasi standar sebelum digunakan untuk pelatihan.
2. Pemodelan

Dua arsitektur detektor satu tahap disiapkan untuk dibandingkan, yaitu YOLOv8n sebagai baseline dan YOLO11n sebagai generasi yang lebih baru. Kedua model menggunakan varian nano karena penelitian ini menyasar penerapan berbiaya rendah pada perangkat bersumber daya terbatas. Perbedaan utama keduanya terletak pada komponen backbone (C2f pada YOLOv8n digantikan C3k2 pada YOLO11n) serta penambahan modul atensi C2PSA pada YOLO11n, sementara neck dan head kedua model tetap sama agar perbandingan terkendali di tingkat komponen.

3. Rancangan Eksperimen

   Evaluasi dirancang melalui skema terkendali 2×2 yang menyilangkan faktor arsitektur (YOLOv8n dan YOLO11n) dengan faktor kondisi data (data primer saja dan data gabungan primer-sekunder), menghasilkan empat konfigurasi yang diuji. Rancangan ini memungkinkan pengaruh arsitektur dan pengaruh penambahan data dipisahkan secara eksplisit, alih-alih tercampur seperti pada sebagian studi terdahulu. Seluruh konfigurasi dievaluasi pada subset pengujian primer yang identik dan terkunci, sehingga perbandingan antarkonfigurasi tetap adil.

4. Pelatihan

   Setiap konfigurasi dilatih sebanyak tiga kali dengan seed acak yang berbeda (seed 0, 1, dan 2), menghasilkan dua belas proses pelatihan secara keseluruhan. Pengulangan ini dilakukan untuk menilai kestabilan hasil terhadap variasi inisialisasi bobot dan urutan data, mengingat proses pelatihan jaringan saraf bersifat stokastik. Seluruh proses pelatihan menggunakan konfigurasi hyperparameter yang identik pada keempat konfigurasi, sehingga selisih performa yang teramati dapat diatribusikan pada faktor arsitektur atau data, bukan pada perbedaan pengaturan pelatihan.

5. Evaluasi

   Performa deteksi dievaluasi menggunakan metrik precision, recall, mAP50, dan mAP50-95 pada subset pengujian primer yang terkunci dan identik bagi seluruh konfigurasi. Efisiensi model dinilai dari jumlah parameter, FLOPs, ukuran berkas, serta kecepatan inferensi pada GPU dan CPU. Analisis kesalahan dilakukan melalui confusion matrix dan kurva Precision-Recall untuk mengidentifikasi pola dan penyebab kegagalan deteksi, dilengkapi contoh kualitatif yang dipilih secara terprogram.

1.7 Sistematika Penulisan

Sistematika penulisan tesis ini disusun dengan struktur sebagai berikut:

BAB I PENDAHULUAN

Berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, garis besar metode penelitian, dan sistematika penulisan. Bab ini memberikan gambaran umum mengenai arah dan ruang lingkup penelitian.

BAB II TINJAUAN PUSTAKA

Memaparkan penelitian terdahulu yang relevan beserta keterbatasannya, posisi penelitian ini, serta landasan teori mengenai deteksi objek, arsitektur YOLO, dan metrik evaluasi yang digunakan. Bab ini menjadi dasar teoretis dan menunjukkan celah penelitian yang diisi oleh penelitian ini.

BAB III METODOLOGI PENELITIAN

Menjelaskan secara rinci tahapan penyiapan data, arsitektur model, rancangan eksperimen 2×2, konfigurasi pelatihan, dan metrik evaluasi. Bab ini menguraikan justifikasi metodologis di balik setiap keputusan eksperimen.

BAB IV HASIL DAN PEMBAHASAN

Menyajikan dan membahas performa deteksi keseluruhan dan per kelas, efisiensi model, analisis kesalahan, serta jawaban atas keempat pertanyaan penelitian. Pembahasan dikaitkan dengan temuan pada Bab II untuk memperkuat interpretasi hasil.

BAB V KESIMPULAN DAN SARAN

Merangkum temuan utama, keterbatasan penelitian, dan saran untuk penelitian lanjutan. Bab ini menjadi penutup yang menegaskan kontribusi serta arah pengembangan penelitian berikutnya.

BAB II
Tinjauan Pustaka

2.1 Deteksi Kerusakan Jalan Berbasis YOLO pada Benchmark Publik

Arya dkk. (2021, 2024) menyusun RDD2022, benchmark 47.420 citra jalan dari enam negara dengan lebih dari 55.000 anotasi dan empat kelas kerusakan (retak memanjang, melintang, buaya, dan lubang), yang menjadi rujukan skema kelas pada banyak studi berikutnya. Zeng dan Zhong (2024) mengembangkan YOLOv8-PD, yaitu YOLOv8n yang dimodifikasi dengan modul transformer (BOT), atensi LSKA, blok C2fGhost, dan kepala deteksi LSCD-Head; dilatih pada 10.000 citra RDD2022 (bagi 8:2, 300 epoch, masukan 640, SGD, batch 16), model mencapai mAP 70,6% pada 111,9 FPS dengan 2,3 juta parameter, tetapi akurasi lubang hanya 53,1% karena ukuran objek lubang yang menempati sedikit piksel dan jumlah sampelnya terbatas. Studi lain memodifikasi YOLOv5/YOLOv8, seperti DenseSPH-YOLOv5 dengan DenseNet dan kepala Swin-transformer untuk objek multi-skala (Roy & Bhaduri, 2023), serta varian lain (Ma & Chen, 2025; J. Wang et al., 2024). Kesamaan kelompok ini: peningkatan dicapai melalui modifikasi arsitektur, sehingga kemampuan arsitektur dasar antargenerasi—tanpa modifikasi—belum banyak diuji.

2.2 Penyempurnaan Arsitektur Berbasis Mekanisme Atensi

Mekanisme atensi memberi bobot berbeda pada bagian citra yang lebih informatif agar model memusatkan perhatian pada wilayah retak. Xu dkk. (2025) menambahkan CBAM pada C2f YOLOv8n dan mengganti rugi CIOU dengan WIOU, memperoleh peningkatan 1,8% (mAP50), 1,7% (mAP50-95), dan 1,8% (recall) atas YOLOv8n. Penting bagi penelitian ini, Xu dkk. memilih YOLOv8n dengan alasan bahwa pemangkasan dinamis YOLO11 dinilai kurang spesifik untuk fitur retak—namun alasan tersebut merupakan asumsi pemilihan model, bukan hasil pengujian terkendali. Pendekatan atensi serupa diterapkan pada YOLOv8-GAM-Wise-IoU (Xiong et al., 2024), EVA-YOLOv8 (Y. Liu et al., 2026), dan perbaikan berbasis CBAM (Yu et al., 2024), serta studi berorientasi perangkat tepi (Geng et al., 2024; S. Wang et al., 2024). Seperti kelompok sebelumnya, yang diukur adalah performa arsitektur termodifikasi.

2.3 Studi Perbandingan Antar-Versi YOLO

Jegham dkk. (2024) melakukan benchmark komprehensif lintas-versi YOLO menggunakan precision, recall, mAP, waktu proses, GFLOPs, dan ukuran model pada beberapa dataset, dan menemukan bahwa setiap versi memiliki keunggulan serta keterbatasan yang khas—versi lebih baru tidak selalu unggul pada semua jenis tantangan. Studi semacam ini menegaskan bahwa perbandingan antar-versi adalah pertanyaan yang sah dan menuntut metrik serta kondisi evaluasi yang seragam; namun umumnya dilakukan pada dataset umum lintas-domain, belum pada deteksi kerusakan jalan lapangan Indonesia—celah yang dijawab penelitian ini.

2.4 Deteksi Lubang

Safyari dkk. (2024) meninjau metode deteksi lubang berbasis visi komputer, dan Khan dkk. (2024) mengembangkan deteksi lubang untuk kendaraan otonom; studi lain memakai YOLOv5 waktu-nyata (Klčo et al., 2023; Bhavan Kumar et al., 2023) serta analisis penapis fitur (Adebiyi et al., 2024). Pola yang berulang: lubang konsisten dilaporkan sebagai kelas dengan akurasi terendah dibanding retak—misalnya 53,1% pada Zeng dan Zhong (2024)—terutama ketika objeknya menempati sedikit piksel dan sampelnya terbatas, sehingga kesulitan ini tampak bersumber dari karakteristik objek dan data, bukan pilihan arsitektur.

2.5 Penelitian pada Konteks Indonesia

Sasmito dkk. (2023) memadukan YOLO dengan GNSS di Semarang (overall accuracy 88%, kappa 86%, RMSE posisi ±5,6 m); Denaro dan Lim (2025) memakai pembelajaran transfer pada jalan nasional Palembang; Jonathan dkk. (2026) mengintegrasikan YOLOv8 ke aplikasi mobile di Palangkaraya (2.296 + 2.245 citra; evaluasi 420 citra: precision 62,4%, recall 62,0%, mAP 62,9%, dengan retak buaya terendah); dua studi lain memakai YOLOv11 untuk lubang (Takyudin dkk., 2025, estimasi luas berbasis web; Ismail & Rahmadewi, 2025, studi kasus Karawang). Cakupannya umumnya satu kota/wilayah, mayoritas memakai dataset daring alih-alih citra lapangan instansi, dan mengevaluasi satu model tanpa pembanding arsitektur setara maupun pelaporan multi-seed/uji terkunci.

2.6 Sintesis Keterbatasan dan Posisi Penelitian

Tabel 2.1 merangkum penelitian terdahulu beserta keterbatasannya. Tabel tersebut memuat lima kolom—studi, metode, dataset, hasil utama, dan keterbatasan yang relevan—sehingga metode dan capaian tiap studi dapat disandingkan langsung dengan keterbatasan yang menjadi dasar pemosisian penelitian ini.

Tabel 2.1 Sintesis Penelitian Terdahulu dan Keterbatasannya

Studi	Metode	Dataset	Hasil utama	Keterbatasan yang relevan
Arya dkk. (2021, 2024)	Benchmark multi-negara	RDD2022; 47.420 citra, 6 negara	Standar kelas dan benchmark rujukan	Tidak memuat citra Indonesia maupun Asia Tenggara tropis
Zeng & Zhong (2024)	YOLOv8-PD (BOT, LSKA, C2fGhost, LSCD-Head)	10.000 citra RDD2022; bagi 8:2	mAP 70,6%; 111,9 FPS; 2,3 jt parameter	Evaluasi pada subset validasi tanpa subset pengujian terpisah; lubang hanya 53,1%; tanpa pengulangan seed
Xu dkk. (2025)	YOLOv8n + CBAM-C2f, rugi WIOU	Dataset retak permukaan jalan	mAP50 +1,8%; mAP50-95 +1,7%; recall +1,8%	Pemilihan YOLOv8n atas YOLO11 didasarkan asumsi kualitatif, tanpa pengujian terkendali
Xiong dkk. (2024); EVA-YOLOv8 (Liu dkk., 2026); Yu dkk. (2024b)	Penyempurnaan berbasis atensi	Dataset retak (jembatan/jalan)	Peningkatan akurasi atas model dasar	Arsitektur dimodifikasi sehingga tidak menguji arsitektur dasar secara setara
Roy & Bhaduri (2023); Ma & Chen (2025); Wang dkk. (2024b)	Varian YOLOv5/YOLO v8 termodifikasi	Benchmark publik	Peningkatan pada objek multi-skala	Perbandingan lintas studi tercampur perbedaan pengaturan
Safyari dkk. (2024); Khan dkk. (2024); Adebiyi dkk. (2024); Bhavan Kumar dkk. (n.d.); Klčo dkk. (2023)	Deteksi/klasifikasi lubang	Dataset lubang	Lubang konsisten menjadi kelas tersulit	Fokus kelas tunggal; tidak menguji deteksi multi-kelas bersamaan
Sasmito dkk. (2023)	YOLO + GNSS	Citra jalan Kota Semarang	Overall accuracy 88%; kappa 86%; RMSE 5,6 m	Satu kota; model tunggal; metrik klasifikasi, bukan mAP deteksi
Denaro & Lim (2025)	YOLO, pembelajaran transfer	Jalan nasional Palembang	Deteksi kerusakan akurat dan cepat	Satu wilayah; tanpa pembanding arsitektur setara
Jonathan dkk. (2026)	YOLOv8 + aplikasi mobile	2.296 + 2.245 citra; 7 jenis kerusakan	Precision 62,4%; recall 62,0%; mAP 62,9%	Satu kota; model tunggal; tanpa multi-seed
Takyudin dkk. (2025); Ismail & Rahmadewi (2025)	YOLOv11 + aplikasi web	Dataset daring (mis. Roboflow)	Deteksi lubang waktu-nyata	Kelas tunggal; data daring, bukan citra lapangan instansi; tanpa pembanding terkendali

Dari sepuluh baris tabel, tiga keterbatasan muncul berulang.Ketiga keterbatasan berulang tersebut adalah: (1) ketiadaan representasi kondisi jalan Indonesia pada benchmark internasional, sedangkan studi domestik terbatas satu wilayah dan jarang memakai citra lapangan instansi—persoalan kesesuaian domain, bukan sekadar asal geografis; (2) perbandingan antararsitektur yang tercampur perbedaan pengaturan (confounding), sehingga selisih sulit diatribusikan; dan (3) integritas evaluasi yang jarang dinyatakan eksplisit (subset uji terkunci, pencegahan kebocoran, pengulangan seed). Penelitian ini menjawab ketiganya melalui data lapangan lima provinsi (PUPR) yang dipadu data sekunder berdomain Indonesia, perbandingan terkendali yang menyetarakan seluruh pengaturan, serta evaluasi ber-integritas (uji primer terkunci, deduplikasi hash, tiga seed), sehingga menjadi perbandingan antar-versi YOLO terkendali pertama pada domain deteksi kerusakan jalan lapangan Indonesia.

2.7 Landasan Teori

2.7.1 Deteksi Objek Satu-Tahap

Deteksi objek menggabungkan dua persoalan: menentukan kelas objek dan menentukan letaknya melalui kotak pembatas. Pendekatan dua tahap (mis. rumpun R-CNN) menghasilkan usulan wilayah lalu mengklasifikasikannya—akurat pada lokalisasi tetapi menuntut komputasi lebih besar karena tiap usulan diproses terpisah. Pendekatan satu tahap (mis. YOLO, SSD) memprediksi kelas dan kotak dalam satu proses jaringan, sehingga waktu inferensinya lebih singkat pada beban komputasi setara (Sapkota et al., 2025; Ali & Zhang, 2024). Karena pemantauan jalan menuntut pemrosesan cepat dan berbiaya rendah, pendekatan satu tahap menjadi pilihan yang lazim.

Pada pendekatan satu tahap terdapat pula arsitektur yang dirancang khusus untuk lingkungan mobile, yaitu MobileNet-SSD dan EfficientDet. EfficientDet dirilis Google Research pada November 2019 dan tidak memperoleh generasi baru sejak itu (Tan et al., 2020). MobileNet-SSD memadukan kerangka deteksi SSD (2016) dengan backbone MobileNet; kombinasi ini mapan pada era MobileNet V1–V3 (2017–2019). Backbone MobileNet sendiri masih berkembang—MobileNetV4 (2024) dan MobileNetV5 (2025)—tetapi versi tersebut diposisikan sebagai backbone klasifikasi umum, bukan sebagai detektor nano siap-pakai yang dirawat dan dirilis lintas-generasi seperti keluarga YOLO. Oleh karena itu, untuk pertanyaan penelitian mengenai manfaat peningkatan arsitektur *antargenerasi*

pada detektor nano, keluarga YOLO menyediakan rangkaian generasi yang paling langsung sebanding dalam satu kerangka kerja.

2.7.2 Arsitektur Keluarga YOLO

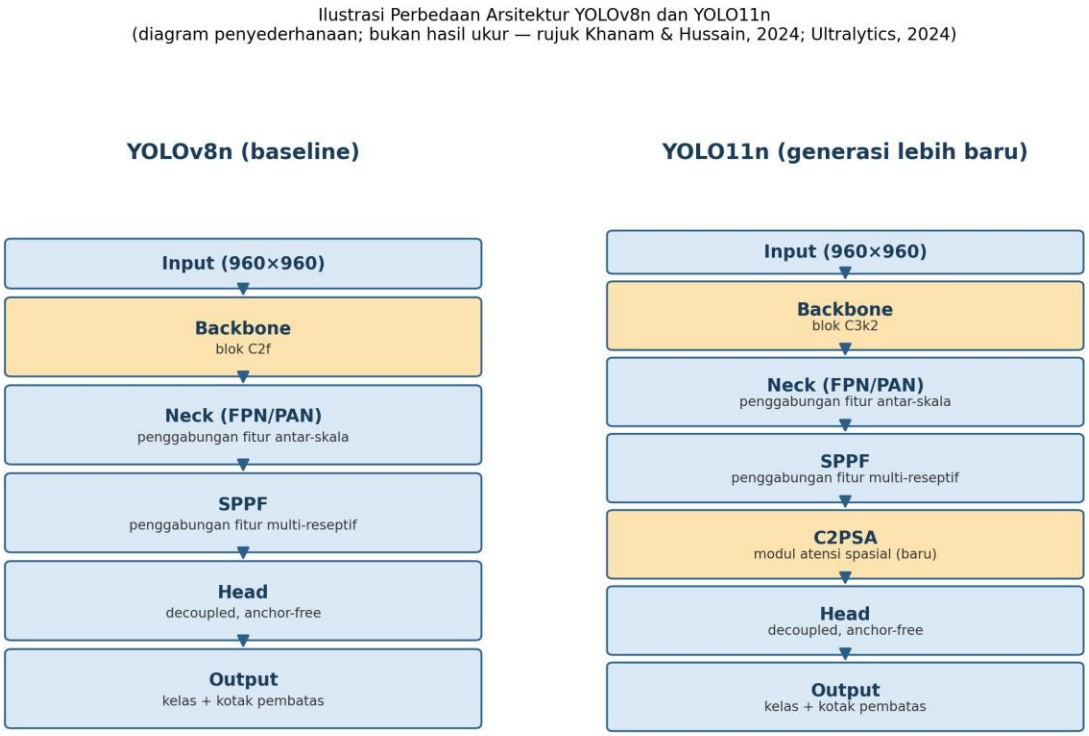
YOLO (You Only Look Once) pertama kali diperkenalkan oleh Redmon dkk. (2016), yang mengubah deteksi objek dari rangkaian dua tahap (usulan wilayah, lalu klasifikasi tiap usulan) menjadi satu persoalan regresi tunggal: sebuah jaringan konvolusi memprediksi langsung koordinat kotak pembatas beserta probabilitas kelas dari seluruh citra dalam satu kali proses forward pass, dari sinilah nama "You Only Look Once" berasal. Pendekatan satu-lintasan ini menjadi dasar filosofi seluruh generasi YOLO berikutnya: kecepatan inferensi mendekati waktu-nyata diprioritaskan tanpa mengorbankan akurasi secara signifikan, menjadikannya pilihan yang lazim untuk aplikasi yang menuntut pemrosesan cepat seperti pemantauan lalu lintas dan infrastruktur jalan. Sejak rilis pertamanya, keluarga YOLO berkembang melalui banyak generasi yang dirawat oleh berbagai kelompok riset dan perusahaan (bukan satu penulis atau organisasi tunggal), dengan tren umum berupa backbone yang semakin efisien, pergeseran dari deteksi berbasis anchor menuju anchor-free, serta penambahan mekanisme atensi pada generasi-generasi yang lebih baru (Sapkota et al., 2025; Jegham et al., 2024).

Arsitektur YOLO tersusun atas *backbone* (ekstraksi fitur pada beberapa tingkat resolusi), *neck* (penggabungan fitur antar-skala melalui piramida fitur), dan *head* (prediksi kelas dan koordinat). Membandingkan dua generasi YOLO berarti membandingkan efek perubahan pada komponen-komponen ini.

2.7.3 YOLOv8n dan YOLO11n

YOLOv8 (Ultralytics, 2023) mengganti struktur C3 pada YOLOv5 dengan C2f yang memperbanyak jalur aliran gradien, memakai kepala terpisah (decoupled head) yang memisahkan cabang klasifikasi dari cabang regresi kotak, dan meninggalkan mekanisme berbasis anchor (Zeng & Zhong, 2024). Pada pendekatan berbasis anchor, model menetapkan sejumlah kotak jangkar (anchor box) berukuran tetap sebagai acuan prediksi; pendekatan anchor-free menghapus kotak jangkar tersebut dan langsung memprediksi posisi serta ukuran kotak, sehingga lebih ringkas dan lebih adaptif terhadap objek dengan bentuk tidak beraturan seperti kerusakan jalan. YOLO11 (Khanam & Hussain, 2024; Ultralytics, 2024) mengganti C2f dengan C3k2 (bottleneck lintas-tahap dengan dua konvolusi berukuran lebih kecil sehingga menurunkan FLOPs), mempertahankan SPPF, dan menambahkan

modul atensi C2PSA setelah SPPF untuk memberi bobot spasial pada peta fitur. Perubahan inilah yang membuat YOLO11n, pada penelitian ini, memuat 2.590.425 parameter berbanding 3.011.433 pada YOLOv8n (lebih sedikit 14,0%) dan 14,5 GFLOPs berbanding 18,4 (lebih rendah 21,2%).



Gambar 2.1 Ilustrasi Perbedaan Arsitektur YOLOv8n dan YOLO11n

Gambar 2.1 adalah diagram penyederhanaan (bukan hasil ukur) yang merangkum letak perbedaan utama kedua arsitektur pada tingkat komponen, dirujuk dari Khanam dan Hussain (2024) serta dokumentasi Ultralytics (2024): penggantian blok backbone dari C2f (YOLOv8n) menjadi C3k2 (YOLO11n), dan penambahan modul atensi C2PSA setelah SPPF pada YOLO11n yang tidak dimiliki YOLOv8n. Kedua arsitektur tetap memakai neck FPN/PAN dan kepala deteksi yang sama-sama decoupled dan anchor-free, sehingga perbandingan pada penelitian ini terkendali pada tingkat komponen sebagaimana dituntut oleh lapis metodologis pada Bab 1.2 Rumusan Masalah: selisih hasil yang akan diamati dapat diatribusikan secara lebih spesifik pada perbedaan blok backbone dan keberadaan modul atensi ini, bukan pada perbedaan kerangka kepala deteksi atau strategi penggabungan fitur.

Perlu dicatat bahwa per tanggal penulisan, YOLO11 (rilis September 2024) bukan generasi paling mutakhir. YOLO12 (Februari 2025) memperkenalkan pendekatan berbasis

atensi (Area Attention dan R-ELAN) yang tetap menyediakan varian nano (yolo12n), namun pendekatan atensinya cenderung lebih berat pada CPU/perangkat tepi dibanding desain murni-konvolusi (Tian et al., 2025). Ultralytics kemudian mengumumkan YOLO26 pada September 2025 (YOLO Vision 2025) dan merilisnya sebagai generasi Januari 2026, dengan desain berorientasi perangkat tepi (edge-first), bebas-NMS (NMS-free), penghapusan cabang Distribution Focal Loss (DFL), serta varian nano (yolo26n); Ultralytics melaporkan inferensi CPU hingga 43% lebih cepat dibanding YOLO11 pada benchmark COCO (Ultralytics, 2026), meskipun klaim tersebut adalah hasil benchmark Ultralytics sendiri dan belum diukur ulang pada dataset penelitian ini. YOLO11 dipilih pada penelitian ini karena merupakan rilis yang matang dan banyak diadopsi untuk produksi, sekaligus generasi yang lebih baru dibanding baseline YOLOv8; evaluasi terhadap generasi yang lebih mutakhir seperti YOLO26n menjadi arah pengembangan lanjutan (Jegham et al., 2024; Sapkota et al., 2025).

Setiap keluarga YOLO menyediakan lima varian skala—nano (n), small (s), medium (m), large (l), dan extra-large (x)—yang dibedakan oleh faktor lebar (jumlah kanal) dan kedalaman (jumlah blok) jaringan. Varian nano adalah varian dengan faktor lebar dan kedalaman paling rendah, sehingga menghasilkan jumlah parameter, FLOPs, dan ukuran berkas paling sedikit di antara kelima varian. Pada penelitian ini, varian nano terukur sebesar 2.590.425 parameter/14,5 GFLOPs/5,5 MB (YOLO11n) dan 3.011.433 parameter/18,4 GFLOPs/6,3 MB (YOLOv8n) pada masukan 960. Istilah “ringan/berbiaya rendah” pada laporan ini merujuk pada rentang ukuran tersebut, yaitu model berukuran di bawah 7 MB dengan beban di bawah 20 GFLOPs, yang dapat dijalankan pada perangkat tanpa akselerator khusus.

2.7.4 Karakteristik Kerusakan Jalan

Ketiga kelas pada penelitian ini didefinisikan mengikuti taksonomi distres perkerasan (Arya et al., 2022; Pavement Interactive, n.d.):

- longitudinal_crack (retak memanjang): retak yang sejajar arah lalu lintas atau garis tengah perkerasan; kerap merupakan tanda awal kelelahan (fatigue) dan dapat berkembang menjadi retak buaya bila berlanjut.
- alligator_crack (retak buaya): jaringan retak saling terhubung berpola menyerupai kulit buaya, merupakan indikator kegagalan struktural akibat beban lalu lintas berulang; memungkinkan infiltrasi air ke lapisan dasar dan, bila tidak ditangani, berkembang menjadi lubang.

- pothole (lubang): cekungan akibat terlepasnya bagian perkerasan, yang kerap merupakan tahap lanjut dari retak yang dibiarkan; menempati sedikit piksel pada citra jalan sehingga menantang untuk dideteksi.

2.7.5 Metrik Evaluasi

Evaluasi deteksi bertumpu pada Intersection over Union (IoU), sebagaimana lazim digunakan pada studi benchmark YOLO (Jegham et al., 2024), yaitu rasio antara luas irisan dan luas gabungan kotak prediksi dengan kotak sebenarnya (ground truth), dirumuskan sebagai berikut: Nilai IoU berada pada rentang 0 hingga 1: bernilai 0 ketika kedua kotak tidak beririsan sama sekali, dan 1 ketika keduanya berimpit sempurna. Dalam pencocokan, sebuah kotak prediksi dipasangkan dengan kotak sebenarnya yang menghasilkan skor IoU maksimum di antara seluruh kotak sebenarnya pada citra tersebut; pasangan dengan IoU tertinggi inilah yang dianggap sebagai kandidat kecocokan. Prediksi kemudian dinyatakan benar (true positive) apabila skor IoU maksimum tersebut mencapai atau melampaui suatu ambang dan kelasnya sesuai; bila di bawah ambang, prediksi dihitung sebagai false positive, dan kotak sebenarnya yang tidak terpasangi dihitung sebagai false negative. Setiap kotak sebenarnya hanya dapat dipasangkan satu kali sehingga prediksi ganda pada objek yang sama tidak dihitung berulang.

$$\mathrm{IoU}(B_p, B_{gt}) = |B_p \cap B_{gt}| \,/\, |B_p \cup B_{gt}| \tag{2.1}$$

Bp adalah kotak pembatas hasil prediksi (predicted box) dan Bgt adalah kotak pembatas sebenarnya (ground truth box). Pemilihan ambang IoU menentukan ketegasan kriteria lokalisasi. Nilai mAP50 menggunakan ambang tunggal 0,50—relatif longgar terhadap ketepatan posisi—sedangkan mAP50-95 merata-ratakan hasil pada sepuluh ambang dari 0,50 hingga 0,95 dengan langkah 0,05, sehingga menuntut lokalisasi yang lebih presisi. Dari nilai true positive (TP), false positive (FP), dan false negative (FN) pada berbagai ambang keyakinan, precision dan recall dirumuskan sebagai berikut, dengan precision menyatakan proporsi prediksi benar di antara seluruh prediksi positif dan recall menyatakan proporsi objek sebenarnya yang berhasil terdeteksi:

$$\text{Precision} = \mathrm{TP} \,/\, (\mathrm{TP} + \mathrm{FP}) \tag{2.2}$$

$$\text{Recall} = \mathrm{TP} \,/\, (\mathrm{TP} + \mathrm{FN}) \tag{2.3}$$

Pada objek retak yang batasnya tidak tegas, selisih antara mAP50 dan mAP50-95 cenderung membesar karena model dapat menemukan keberadaan retak namun sulit

menepatkan tepinya. Precision dan recall pada dasarnya saling bertukar (trade-off): menaikkan ambang keyakinan cenderung menaikkan precision namun menurunkan recall, dan sebaliknya. Average precision (AP) adalah luas di bawah kurva Precision-Recall, dirumuskan sebagai berikut, dengan p(r) adalah nilai precision pada tingkat recall r:

$$AP = \int_0^1 p(r)\, dr \tag{2.4}$$

Mean average precision (mAP) kemudian dirata-ratakan antar-kelas:

$$mAP = (1/N) \sum_i AP_i \tag{2.5}$$

N adalah jumlah kelas kerusakan yang dievaluasi (N = 3 pada penelitian ini: pothole, longitudinal_crack, alligator_crack) dan APi adalah average precision kelas ke-i. Skor F1 merangkum pertukaran precision-recall sebagai rata-rata harmonik keduanya pada satu ambang keyakinan tertentu:

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \tag{2.6}$$

Sehingga F1 berguna untuk membandingkan performa pada satu titik operasi, berbeda dengan mAP yang merangkum performa di seluruh rentang ambang keyakinan sekaligus.

2.7.6 Kompleksitas dan Kecepatan Model

Selain metrik akurasi, indikator efisiensi—jumlah parameter, FLOPs, ukuran model, dan FPS—digunakan untuk menilai potensi penerapan, sejalan dengan kajian efisiensi model deteksi objek (Tan et al., 2020). Ketiga indikator pertama melekat pada arsitektur dan tidak bergantung perangkat, sedangkan FPS bergantung pada perangkat keras tempat model dijalankan dan dirumuskan sebagai kebalikan dari waktu inferensi rata-rata per citra:

$$FPS = 1 / t_{\text{inferensi}} \tag{2.7}$$

Tinferensi adalah waktu rata-rata (dalam detik) yang dibutuhkan model untuk memproses satu citra pada perangkat tertentu; semakin kecil tinferensi, semakin tinggi FPS yang dicapai.

2.7.7 Prapemrosesan dan Augmentasi Data

Sebelum digunakan untuk pelatihan, citra melewati prapemrosesan agar seragam. Penyeragaman ukuran dilakukan dengan mengubah setiap citra ke satu resolusi masukan; salah satu metodenya adalah letterbox, yaitu penskalaan dengan mempertahankan rasio aspek asli dan menambahkan bilah pengisi (padding) pada sisi yang kurang agar objek tidak

terdistorsi. Normalisasi menyetarakan rentang nilai piksel (umumnya ke 0–1) agar pelatihan lebih stabil.

Augmentasi data memperluas keragaman data pelatihan melalui transformasi citra sehingga model tidak menghafal citra pelatihan dan lebih tahan terhadap variasi lapangan. Beberapa teknik augmentasi yang lazim pada deteksi objek, sebagaimana diterapkan pada kerangka kerja Ultralytics (Ultralytics, 2023), adalah: mosaic, yang menggabungkan empat citra menjadi satu citra komposit agar model melihat lebih banyak konteks dan variasi skala; penyesuaian ruang warna HSV (hue, saturation, value), yang mengubah rona, saturasi, dan kecerahan untuk menirukan variasi pencahayaan; serta pencerminan (flip), penskalaan (scale), dan translasi (translate) untuk variasi orientasi, ukuran, dan posisi objek.

2.7.8 Pembagian Data dan Pencegahan Kebocoran

Pembagian data memisahkan dataset menjadi subset pelatihan, validasi, dan pengujian. Pembagian terstratifikasi menjaga proporsi tiap kelas tetap serupa di seluruh subset, yang penting ketika distribusi kelas tidak seimbang. Ancaman terhadap keabsahan evaluasi adalah kebocoran data (data leakage), yaitu ketika citra yang sama atau nyaris identik muncul pada lebih dari satu subset sehingga model dinilai atas data yang pernah dilihatnya dan performa yang dilaporkan menjadi terlalu optimistis (Kapoor & Narayanan, 2023); salah satu pencegahannya adalah deduplikasi berbasis hash konten, yaitu menghitung sidik jari (hash) dari isi setiap citra untuk mengenali dan menyatukan citra yang identik sebelum pembagian.

2.7.9 Istilah dan Proses Pelatihan

Pada tahap pelatihan menggunakan kerangka kerja Ultralytics (Ultralytics, 2024), beberapa istilah digunakan secara berulang. Satu epoch adalah satu kali lintasan penuh atas seluruh data pelatihan; batch adalah jumlah citra yang diproses sekaligus dalam satu langkah pembaruan bobot; optimizer adalah algoritma yang memperbarui bobot untuk meminimalkan fungsi rugi, dengan besar langkah pembaruan ditentukan oleh laju pembelajaran (learning rate). Early stopping adalah mekanisme menghentikan pelatihan lebih awal ketika performa validasi tidak lagi membaik, sedangkan checkpoint adalah simpanan bobot model pada suatu titik pelatihan—penelitian ini menggunakan checkpoint dengan performa validasi terbaik. Karena inisialisasi bobot dan pengacakan data bersifat acak, seed (bilangan awal pembangkit acak) ditetapkan agar hasil dapat direproduksi;

pengulangan pelatihan dengan beberapa seed berbeda memungkinkan penilaian kestabilan hasil.

2.7.10 Variabilitas Antar-Seed dan Simpangan Baku

Ketika pelatihan diulang dengan beberapa seed, hasilnya dilaporkan sebagai rata-rata beserta simpangan baku antar-seed, dirumuskan sebagai berikut:

$$\bar{x} = (1/n)\ \Sigma_i\ x_i \tag{2.8}$$

$$s = \sqrt{[(1/(n-1))\ \Sigma_i\ (x_i - \bar{x})^2]} \tag{2.9}$$

Rata-rata menunjukkan estimasi performa yang tidak bergantung pada satu inisialisasi acak tertentu, sedangkan simpangan baku menunjukkan seberapa konsisten performa tersebut di antara pengulangan—simpangan baku yang kecil mengindikasikan hasil yang stabil dan tidak kebetulan, sedangkan simpangan baku yang besar menandakan performa yang sensitif terhadap inisialisasi atau urutan data. Konsekuensinya, selisih rata-rata antara dua kondisi yang lebih kecil daripada simpangan bakunya perlu dibaca hati-hati, karena rentang variasi antar-seed dapat menjelaskan sebagian atau seluruh selisih yang teramati. Untuk menguji apakah suatu selisih rata-rata cukup besar dibandingkan variasi antar-seed, dapat digunakan uji signifikansi statistik seperti uji-t berpasangan (Bouthillier et al., 2021), dirumuskan sebagai berikut, yang menghasilkan nilai-p sebagai ukuran seberapa mungkin selisih sebesar itu muncul semata karena variasi acak:

$$t = \bar{d} / (s_d / \sqrt{n}) \tag{2.10}$$

n adalah jumlah pengulangan seed (untuk Persamaan 2.8-2.9) atau jumlah pasangan seed (untuk Persamaan 2.10), xi adalah nilai metrik pada pengulangan seed ke-i, $\bar{x}$ adalah rata-rata, s adalah simpangan baku antar-seed, $\bar{d}$ adalah rata-rata selisih berpasangan antar-seed dari dua kondisi yang dibandingkan, dan sd adalah simpangan baku selisih tersebut. Nilai-p yang kecil (lazimnya di bawah 0,05) menunjukkan selisih yang tidak mudah dijelaskan oleh variasi acak semata. Namun, dengan jumlah pengulangan seed yang kecil seperti tiga, uji signifikansi semacam ini bersifat indikatif dan rentan terhadap fluktuasi sampel kecil, sehingga hasilnya perlu dibaca sebagai petunjuk arah, bukan bukti definitif.

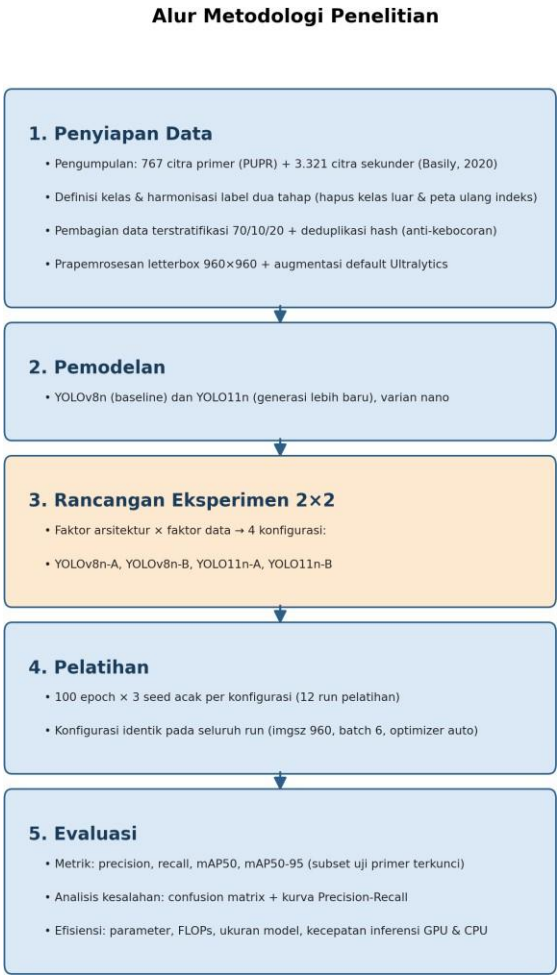
2.7.11 Confusion Matrix, Kurva Evaluasi, dan Overfitting

Confusion matrix ternormalisasi memetakan setiap objek sebenarnya terhadap kelas prediksinya, sebagaimana lazim digunakan pada evaluasi model YOLO (Sapkota et al., 2025), termasuk kategori background, sehingga dapat dibedakan kesalahan antar-kelas

(model bingung membedakan jenis kerusakan) dari objek yang gagal terdeteksi (jatuh ke background). Kurva Precision-Recall menampilkan pertukaran precision–recall pada seluruh rentang ambang keyakinan. Adapun overfitting—ketika rugi pelatihan terus menurun sementara rugi validasi berbalik naik—menjadi alasan digunakannya checkpoint validasi terbaik untuk evaluasi, karena pada titik itu model belum kehilangan kemampuan generalisasinya.

BAB III
Metodologi Penelitian

Bab ini memaparkan tahapan penelitian secara berurutan, mulai dari penyiapan data hingga evaluasi model. Secara garis besar, penelitian ini menempuh lima tahap: penyiapan data (mencakup pengumpulan, definisi kelas dan anotasi, pembagian data, serta prapemrosesan dan augmentasi), pemodelan, rancangan eksperimen terkendali, pelatihan, dan evaluasi. Alur keseluruhan tahapan tersebut ditunjukkan pada Gambar 3.1.



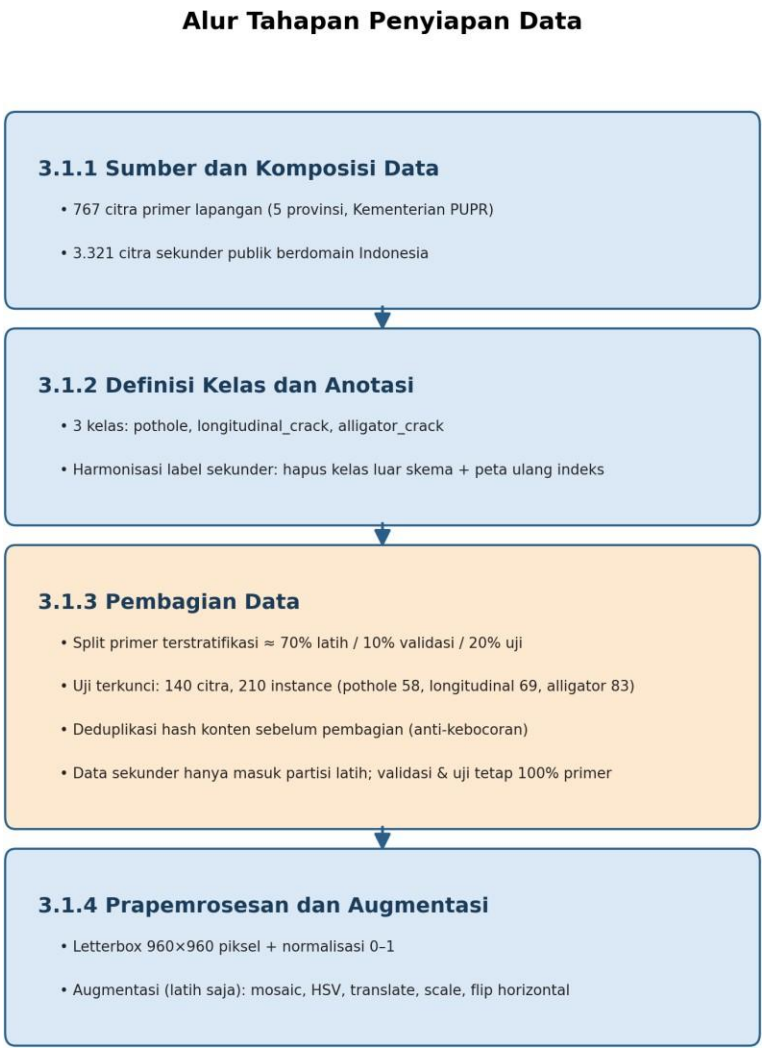
Gambar 3.1 Alur Metodologi Penelitian

Gambar 3.1 menyajikan urutan lima tahap penelitian beserta arah alirannya, yaitu (1) penyiapan data, (2) pemodelan, (3) rancangan eksperimen 2×2, (4) pelatihan, dan (5) evaluasi, dari penyiapan data di bagian atas hingga evaluasi di bagian bawah. Diagram tersebut menegaskan bahwa rancangan eksperimen 2×2 menghasilkan empat konfigurasi

(YOLOv8n-A, YOLOv8n-B, YOLO11n-A, YOLO11n-B) yang masing-masing dilatih selama 100 epoch dengan tiga seed acak (total 12 run pelatihan), lalu dievaluasi pada subset pengujian primer terkunci yang sama, sehingga alur data dari sumber hingga metrik dapat ditelusuri secara utuh.

3.1 Penyiapan Data

Tahap penyiapan data mencakup empat kegiatan yang saling terkait, yaitu pengumpulan data dari dua sumber, pendefinisian kelas beserta harmonisasi anotasi, pembagian data yang bebas kebocoran, serta prapemrosesan dan augmentasi. Keempatnya disatukan pada bagian ini karena membentuk satu rangkaian alur data tunggal: keluaran satu kegiatan menjadi masukan kegiatan berikutnya, hingga menghasilkan data siap latih yang identik bagi seluruh konfigurasi model.



Gambar 3.2 Alur Tahapan Penyiapan Data

Gambar 3.2 memvisualisasikan keempat kegiatan pada Subbab 3.1 sebagai satu rangkaian alur data tunggal: (1) sumber dan komposisi data (767 citra primer dan 3.321 citra sekunder), (2) definisi kelas dan anotasi (harmonisasi label dua tahap), (3) pembagian data (split terstratifikasi sekitar 70/10/20 dengan deduplikasi hash anti-kebocoran), dan (4) prapemrosesan dan augmentasi (letterbox 960×960 serta augmentasi standar Ultralytics), dari pengumpulan data primer dan sekunder hingga data siap latih yang identik bagi seluruh konfigurasi model.

3.1.1 Sumber dan Komposisi Data

Penelitian ini menggunakan dataset gabungan citra jalan Indonesia dari sumber primer dan sekunder. Sumber primer berupa 767 citra lapangan yang diperoleh dari kantor wilayah Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) di lima provinsi, yaitu Bali, Banten, Daerah Istimewa Yogyakarta, Jawa Tengah, dan Jawa Timur. Karena citra tersebut diambil pada kondisi kerja rutin instansi, kualitas akuisisinya mencerminkan inspeksi jalan di dunia nyata, bukan benchmark yang terkurasi. Sumber sekunder berupa dataset kerusakan jalan publik (Basily, 2020) yang terdiri atas 3.321 citra yang diambil menggunakan kamera ponsel. Dataset tersebut tidak mendokumentasikan negara asalnya, sehingga provenansi Indonesia ditetapkan melalui inspeksi visual atas marka jalan dan pelat kendaraan yang terdaftar secara lokal; dataset yang sama juga digunakan sebagai data uji generalisasi oleh Zeng dan Zhong (2024) tanpa menyebut negara asal, sehingga penetapan provenansi pada penelitian ini merupakan kontribusi tersendiri. Dengan demikian, dataset gabungan berjumlah 4.088 citra. Tabel 3.1 merinci komposisi dataset tersebut menurut sumber (primer PUPR, sekunder publik, dan gabungan) dan menurut ketiga kelas kerusakan, dengan jumlah citra dan jumlah anotasi dilaporkan pada baris terpisah untuk setiap sumber. Dari tabel tersebut terlihat bahwa data sekunder menyumbang 3.321 dari 4.088 citra gabungan (81,2%) dan, pada tingkat anotasi, mendominasi kelas pothole dengan 2.657 dari 2.943 anotasi pothole gabungan (90,3%), sedangkan data primer menyumbang 767 citra (18,8%).

Tabel 3.1 Komposisi Dataset Menurut Sumber dan Kelas

Sumber	Total citra	Satuan	pothole	longitudinal_crack	alligator_crack
Primer (PUPR)	767	citra	219	238	310
		anotasi	286	304	424
Sekunder (publik; Basily, 2020)	3.321	citra	1.331	892	790
		anotasi	2.657	1.222	964
Gabungan	4.088	citra	1.550	1.130	1.100
		anotasi	2.943	1.526	1.388

Jumlah citra dan anotasi dilaporkan terpisah karena satu citra dapat memuat beberapa anotasi. Pada data primer, jumlah citra per kelas berjumlah tepat sama dengan total (219 + 238 + 310 = 767), yang menunjukkan setiap citra primer memuat satu kelas; pada data sekunder, jumlah citra per kelas saling tumpang tindih karena sebagian citra memuat lebih dari satu kelas. Rasio anotasi per citra adalah sekitar 1,3 pada data primer dan 1,5 pada data sekunder. Komposisi kelas juga berbeda tajam antar-sumber: pada data primer, alligator_crack paling banyak (310 citra/424 anotasi) sedangkan pothole paling sedikit (219 citra/286 anotasi); pada data sekunder, pola ini berbalik—pothole mendominasi (1.331 citra/2.657 anotasi, lebih dari separuh total anotasi sekunder) sedangkan longitudinal_crack dan alligator_crack porsinya jauh lebih kecil. Perbedaan komposisi ini menjadi salah satu sumber kesenjangan domain antara data primer dan sekunder yang turut memengaruhi performa deteksi pada kelas pothole.

3.1.2 Definisi Kelas dan Anotasi

Tugas deteksi didefinisikan atas tiga kelas kerusakan jalan yang dianotasi dalam format kotak pembatas (bounding box) YOLO, sebagaimana didefinisikan pada Bab 2: pothole (indeks 0), longitudinal_crack (indeks 1), dan alligator_crack (indeks 2). Gambar 3.3 menampilkan satu citra representatif dari subset pengujian primer terkunci untuk masing-masing kelas.



Gambar 3.3 Contoh Citra Tiap Kelas Kerusakan

Gambar 3.3 menampilkan satu citra representatif untuk masing-masing kelas guna memperlihatkan perbedaan visual antarkelas, disusun dari kiri ke kanan sebagai panel 3.3(a), 3.3(b), dan 3.3(c): panel 3.3(a) pothole berupa cekungan dengan batas relatif jelas; panel 3.3(b) longitudinal_crack berupa garis retak tunggal sejajar arah lalu lintas; dan panel 3.3(c) alligator_crack berupa pola jaring retak saling terhubung menyerupai kulit buaya. Kotak merah pada tiap panel menandai lokasi anotasi ground truth untuk kelas yang bersangkutan.

Citra primer telah dianotasi langsung mengikuti skema tiga kelas di atas. Sebaliknya, dataset sekunder semula menyediakan empat kelas, yaitu ketiga kelas di atas ditambah kelas retak lateral (retak melintang) yang berada di luar cakupan penelitian ini, dengan urutan indeks kelas yang berbeda dari skema primer. Untuk menyatukan kedua sumber ke dalam satu ruang label yang konsisten, diterapkan harmonisasi dua tahap pada setiap berkas label sekunder. Istilah "dua tahap" merujuk pada dua operasi berurutan berikut.

1. Tahap penghapusan kelas. Seluruh anotasi berlabel retak lateral dihapus dari berkas label, karena kelas tersebut tidak termasuk dalam tugas deteksi. Sebagai ilustrasi, sebuah berkas label sekunder yang semula memuat baris untuk kelas retak lateral akan kehilangan baris tersebut, sementara baris untuk kelas lain dipertahankan.
2. Tahap pemetaan ulang indeks. Indeks kelas yang tersisa kemudian dipetakan ulang agar sesuai dengan skema primer. Sebagai contoh, apabila pada dataset sekunder lubang berindeks 3 sedangkan pada skema primer lubang berindeks 0, maka seluruh indeks 3 pada berkas label sekunder diubah menjadi 0; hal serupa diterapkan pada kelas lainnya sehingga kedua sumber menggunakan penomoran kelas yang identik.

Mengikuti perlakuan pada data primer, citra sekunder yang seluruh anotasinya hanya berupa retak lateral tetap dipertahankan dengan berkas labelnya dikosongkan, sehingga citra tersebut berperan sebagai background selama pelatihan. Pilihan ini menjaga konsistensi antara kedua sumber, tetapi memunculkan derau label karena retak lateral dan longitudinal memiliki kemiripan bentuk garis dan hanya berbeda orientasi. Karena dataset dan skema pelabelan yang identik digunakan pada seluruh model, pengaruh derau ini berlaku setara pada setiap konfigurasi dan tidak mencederai keadilan perbandingan.

3.1.3 Pembagian Data

Setelah anotasi diselaraskan, dataset primer dibagi menjadi subset pelatihan, validasi, dan pengujian menggunakan skema terstratifikasi, dengan subset pengujian dikunci agar berfungsi sebagai target dunia nyata yang tetap pada seluruh eksperimen. Stratifikasi dilakukan berdasarkan kelas dengan frekuensi kemunculan paling sedikit pada tingkat

dataset (kelas minoritas) yang terdapat pada setiap citra; label stratifikasi setiap citra ditetapkan sebagai kelas paling jarang yang dikandungnya, sehingga proporsi kelas minoritas pada subset pelatihan, validasi, dan pengujian mendekati proporsinya pada keseluruhan dataset dan tidak terkonsentrasi pada salah satu subset. Pembagian yang dihasilkan berkisar 70% pelatihan, 10% validasi, dan 20% pengujian; subset pengujian terkunci memuat 140 citra dan 210 instance (pothole 58, longitudinal_crack 69, alligator_crack 83), sedangkan subset validasi memuat 70 citra dan 125 instance.

Untuk menjaga evaluasi tetap bebas kebocoran (leakage-safe), citra dideduplikasi berdasarkan hash konten sebelum pembagian, sehingga citra yang identik secara bita—termasuk yang muncul pada lebih dari satu folder kelas—diperlakukan sebagai satu unit dan tidak dapat tersebar ke subset yang berbeda. Data sekunder hanya ditambahkan ke partisi pelatihan pada eksperimen yang relevan; subset validasi dan pengujian tetap tersusun sepenuhnya dari citra primer lapangan. Dengan demikian, setiap metrik yang dilaporkan mencerminkan performa pada kondisi jalan primer yang ditahan, dan subset pengujian bersifat identik pada keempat run.

3.1.4 Prapemrosesan dan Augmentasi

Sebelum masuk ke pelatihan, seluruh citra melewati prapemrosesan yang seragam. Setiap citra diubah ukurannya ke resolusi masukan 960 × 960 piksel menggunakan skema letterbox, yaitu penskalaan dengan mempertahankan rasio aspek asli dan menambahkan bilah pengisi (padding) pada sisi yang kurang, agar objek tidak terdistorsi. Nilai piksel kemudian dinormalisasi ke rentang 0–1 dan citra disusun dalam urutan kanal RGB. Resolusi masukan ditetapkan 960 × 960 piksel, yaitu 1,5 kali nilai bawaan Ultralytics sebesar 640 × 640 piksel, sehingga jumlah piksel per citra menjadi sekitar 2,25 kali lipat. Pemilihan ini dilatarbelakangi karakteristik objek retak yang lebarnya kerap hanya beberapa piksel; pada resolusi 640 sebagian retak dapat menyusut hingga di bawah 2 piksel dan hilang saat penurunan resolusi, sedangkan pada resolusi 960 lebar piksel tersebut bertambah sekitar 1,5 kali sehingga lebih terpelihara. Nilai resolusi yang sama diterapkan pada kedua arsitektur agar tidak menjadi variabel pembeda.

Augmentasi data diterapkan hanya pada subset pelatihan dan menggunakan konfigurasi bawaan (default) kerangka kerja Ultralytics, sehingga hasil dapat direproduksi dan tidak menjadi sumber perbedaan antararsitektur. Augmentasi bawaan tersebut mencakup: mosaic (probabilitas 1,0), yang menggabungkan empat citra berbeda menjadi satu citra komposit sehingga model melihat lebih banyak konteks dan variasi skala dalam

satu langkah pelatihan; penyesuaian ruang warna HSV, yaitu perubahan acak pada rona (hue, 0,015), saturasi (0,7), dan kecerahan (value, 0,4) untuk menirukan variasi pencahayaan dan kondisi kamera di lapangan; translasi (0,1) dan penskalaan (0,5), yaitu pergeseran dan perubahan ukuran acak agar model tahan terhadap variasi posisi dan jarak objek; pencerminan horizontal (horizontal flip, probabilitas 0,5), yang membalik citra secara mendatar—pencerminan vertikal tidak digunakan karena orientasi atas–bawah jalan bermakna; serta penonaktifan mosaic pada akhir pelatihan (close mosaic, 10 epoch terakhir), yaitu augmentasi mosaic dimatikan menjelang akhir pelatihan agar model menyesuaikan diri pada citra utuh yang lebih menyerupai kondisi pengujian.

Karena seluruh konfigurasi prapemrosesan dan augmentasi diterapkan secara identik pada kedua arsitektur dan kedua kondisi data, setiap perbedaan performa yang teramati dapat diatribusikan pada arsitektur atau pada data pelatihan, bukan pada pipeline masukan. Data siap latih inilah yang selanjutnya digunakan pada tahap pemodelan.

3.2 Model Deteksi

Berdasarkan data siap latih yang telah disiapkan, tahap pemodelan membandingkan dua detektor satu tahap dari keluarga YOLO, yaitu YOLOv8n sebagai baseline dan YOLO11n sebagai generasi yang lebih baru, keduanya dalam konfigurasi nano (Sapkota et al., 2025; Ultralytics, 2024). Varian nano (“n”) merupakan konfigurasi dengan jumlah parameter paling sedikit di antara lima varian skala pada masing-masing keluarga (n, s, m, l, dan x), yang diperoleh dengan menurunkan faktor lebar dan kedalaman jaringan. Pada penelitian ini, jumlah parameter terukur adalah 2.590.425 untuk YOLO11n dan 3.011.433 untuk YOLOv8n (lebih sedikit 420.008 parameter atau 14,0%), dengan beban komputasi 14,5 GFLOPs berbanding 18,4 GFLOPs (lebih rendah 21,2%) dan ukuran berkas model 5,5 MB berbanding 6,3 MB (lebih kecil 12,7%), sebagaimana diukur pada ukuran masukan 960 dan dirinci pada Tabel 3.6. Varian nano dipilih karena ketiga indikator tersebut menempatkannya sebagai varian paling ringan untuk dijalankan pada perangkat dengan keterbatasan memori dan komputasi, sesuai target penerapan berbiaya rendah pada penelitian ini. Ringkasan karakteristik arsitektur kedua model disajikan pada Tabel 3.2, sedangkan perbandingan indikator efisiensi kuantitatifnya (jumlah parameter, FLOPs, ukuran model, waktu inferensi, dan FPS) dirangkum pada Tabel 3.6.

Tabel 3.2 Ringkasan Karakteristik Arsitektur YOLOv8n dan YOLO11n

Aspek	**YOLOv8n (baseline)**	**YOLO11n (generasi lebih baru)**
Tahun rilis	2023	2024
Peran dalam penelitian	Pembanding (baseline)	Arsitektur yang dievaluasi
Blok utama backbone/neck	C2f	C3k2
Modul penggabung fitur	SPPF	SPPF
Modul atensi	—	C2PSA (atensi spasial)
Kepala deteksi	Decoupled, anchor-free	Decoupled, anchor-free

Tabel 3.2 meringkas persamaan dan perbedaan utama kedua arsitektur pada tingkat komponen. Keduanya sama-sama memakai kepala deteksi terpisah yang anchor-free dan modul SPPF, sehingga tetap sebanding; perbedaan utamanya terletak pada penggantian blok C2f menjadi C3k2 dan penambahan modul atensi C2PSA pada YOLO11n, yang menjadi sumber selisih jumlah parameter sebesar 14,0% (2.590.425 berbanding 3.011.433) serta perbedaan kemampuan ekstraksi fitur yang diuji secara empiris pada penelitian ini. Pemilihan dua arsitektur dari satu keluarga merupakan keputusan metodologis yang disengaja: pertanyaan penelitian berfokus pada apakah peningkatan arsitektur antargenerasi membawa manfaat, sehingga perbandingan perlu dilakukan pada keluarga yang menyediakan rangkaian generasi detektor nano yang sebanding dalam satu kerangka kerja. Arsitektur ringan lain seperti EfficientDet (dirilis 2019 dan tanpa generasi baru sejak itu) maupun detektor MobileNet-SSD (yang mapan pada era backbone MobileNet V1–V3, 2017–2019) tidak menyediakan rangkaian generasi detektor nano yang dirawat dan langsung sebanding seperti YOLO, dan membandingkannya menuntut kerangka kerja berbeda yang justru memunculkan percampuran variabel (confounding). Perlu dicatat bahwa YOLO11 bukan generasi paling mutakhir pada saat penulisan—telah muncul YOLO12 (Februari 2025, berbasis atensi; Tian et al., 2025) dan YOLO26 (diumumkan September 2025, dirilis sebagai generasi Januari 2026, edge-first dan bebas-NMS; Ultralytics, 2026)—namun YOLO11 dipilih karena merupakan rilis yang matang dan banyak diadopsi, sekaligus generasi yang lebih baru dibanding baseline YOLOv8; perbandingan lintas-keluarga maupun dengan generasi yang lebih mutakhir seperti YOLO26n menjadi arah pengembangan lanjutan.

3.3 Rancangan Eksperimen dan Justifikasinya

Untuk menjawab pertanyaan penelitian secara terkendali, evaluasi mengikuti rancangan terkendali 2 × 2. Istilah "2 × 2" merujuk pada persilangan dua faktor yang masing-masing memiliki dua taraf, sehingga menghasilkan empat kombinasi (konfigurasi) yang diuji. Faktor pertama adalah arsitektur, dengan dua taraf yaitu YOLOv8n dan YOLO11n; faktor kedua adalah kondisi data, dengan dua taraf yaitu pelatihan hanya pada data primer (kondisi A) dan pelatihan pada data gabungan primer dan sekunder (kondisi B). Persilangan kedua faktor tersebut menghasilkan empat konfigurasi: YOLOv8n-A, YOLOv8n-B, YOLO11n-A, dan YOLO11n-B, sebagaimana diringkas pada Tabel 3.3. Struktur ini dipilih karena memungkinkan dua jenis perbandingan sekaligus dari satu rangkaian eksperimen: membandingkan sepanjang sumbu arsitektur (YOLOv8n terhadap YOLO11n pada kondisi data yang sama) untuk mengisolasi pengaruh arsitektur, dan membandingkan sepanjang sumbu data (kondisi A terhadap B pada arsitektur yang sama) untuk mengukur pengaruh penambahan data sekunder. Karena keempat konfigurasi berbagi subset pengujian, skema kelas, prapemrosesan, augmentasi, dan hyperparameter yang identik, setiap selisih yang teramati dapat diatribusikan pada faktor yang divariasikan, bukan pada pengaturan eksperimen.

Tabel 3.3 Empat Konfigurasi Hasil Persilangan Rancangan 2×2

Arsitektur \ Kondisi data	**A (primer)**	**B (primer + sekunder)**
YOLOv8n	YOLOv8n-A	YOLOv8n-B
YOLO11n	YOLO11n-A	YOLO11n-B

Tabel 3.3 memetakan kedua faktor pada baris (arsitektur) dan kolom (kondisi data); setiap sel adalah satu konfigurasi yang dilatih dengan tiga seed dan dievaluasi pada subset pengujian primer terkunci yang sama. Selanjutnya, keterkaitan setiap eksperimen dengan pertanyaan penelitian yang hendak dijawab dirinci pada Tabel 3.4.

Tabel 3.4 memuat tiga kolom, yaitu eksperimen yang dilakukan, hal yang hendak dibuktikan oleh masing-masing eksperimen, dan alasan pelaksanaannya, sehingga keterkaitan setiap run dengan pertanyaan penelitian dapat dibaca secara langsung.

Tabel 3.4 Pemetaan Eksperimen terhadap Pertanyaan Penelitian

Eksperimen	Yang hendak dibuktikan	Alasan pelaksanaan
YOLOv8n-A dan YOLO11n-A (data primer)	Apakah selisih performa antararsitektur muncul ketika data pelatihan terbatas	Menjawab pertanyaan pertama pada rezim data yang mencerminkan ketersediaan data lapangan yang realistis
YOLOv8n-B dan YOLO11n-B (data gabungan)	Apakah selisih antararsitektur bertahan ketika data pelatihan diperbanyak	Menguji ketahanan temuan pertama sekaligus menjadi pembanding untuk mengukur besar pengaruh data
Perbandingan A terhadap B pada tiap arsitektur	Seberapa besar penambahan data sekunder memperbaiki performa dan apakah manfaatnya setara	Menjawab pertanyaan kedua; selisih manfaat menjadi dasar penilaian efisiensi data
Pengukuran parameter, FLOPs, ukuran model, dan FPS	Besarnya biaya komputasi masing-masing arsitektur pada perangkat yang sama	Menjawab pertanyaan ketiga; menentukan kelayakan penerapan pada perangkat bersumber daya terbatas
Confusion matrix dan kurva Precision-Recall	Jenis kesalahan yang dominan dan kelas yang paling menantang	Menjawab pertanyaan keempat; metrik agregat tidak menunjukkan letak kegagalan
Pengulangan tiga seed acak pada seluruh konfigurasi	Apakah selisih yang teramati konsisten atau merupakan fluktuasi acak	Selisih antararsitektur berpotensi kecil sehingga rentan terhadap variabilitas stokastik

Sejumlah keputusan eksperimen lain juga memerlukan penjelasan. Varian nano dipilih untuk kedua arsitektur karena penelitian ini menyasar penerapan berbiaya rendah, sehingga perbandingan dilakukan pada varian dengan jumlah parameter, FLOPs, dan ukuran model paling rendah di antara kelima varian skala (n, s, m, l, x), sesuai target perangkat bersumber daya terbatas. Ukuran masukan 960 ditetapkan lebih besar daripada nilai bawaan 640 karena retak bersifat tipis dan memanjang, sehingga resolusi yang lebih tinggi mempertahankan detail yang menentukan deteksinya; nilai yang sama diterapkan pada kedua arsitektur agar tidak menjadi variabel pembeda. Batch size ditetapkan pada nilai tetap, bukan dipilih otomatis, karena penentuan otomatis dapat memberikan nilai berbeda pada kedua model dan dengan demikian mencemari perbandingan.

Subset pengujian dikunci dan disusun sepenuhnya dari citra primer lapangan agar seluruh konfigurasi dinilai terhadap target yang sama dan mencerminkan kondisi penerapan sebenarnya. Deduplikasi berbasis hash konten dilakukan sebelum pembagian data untuk memastikan citra yang identik tidak tersebar ke subset yang berbeda, karena kebocoran semacam itu akan menaikkan metrik secara semu. Data sekunder hanya ditambahkan ke subset pelatihan sehingga sumbu data benar-benar merupakan ablasi terhadap data pelatihan, bukan perubahan pada target evaluasi. Terakhir, checkpoint dengan performa validasi terbaik dipertahankan karena kurva validasi menunjukkan performa memuncak sebelum epoch terakhir, sehingga penggunaan checkpoint epoch terakhir akan menilai model pada kondisi yang telah ter-overfit.

3.4 Konfigurasi Pelatihan

Setelah rancangan ditetapkan, seluruh konfigurasi dilatih dengan pengaturan yang identik berikut. Setiap model dilatih selama 100 epoch pada ukuran masukan 960 dengan batch size 6, menggunakan kerangka kerja Ultralytics pada GPU NVIDIA RTX 3050. Pengoptimal ditetapkan ke mode otomatis, yang memilih pengoptimal dan laju pembelajaran awal secara otomatis berdasarkan konfigurasi pelatihan; karena mode ini bersifat deterministik terhadap konfigurasi yang sama, kedua arsitektur menerima pengoptimal dan laju pembelajaran yang setara. Early stopping tidak diaktifkan, sehingga pelatihan berjalan penuh selama 100 epoch.

Untuk menjaga keadilan perbandingan, batch size ditetapkan pada nilai bilangan bulat yang sama untuk kedua arsitektur, bukan dipilih otomatis. Karena early stopping tidak digunakan sementara kurva validasi menunjukkan performa memuncak sebelum epoch terakhir, checkpoint dengan performa validasi terbaik dipertahankan untuk seluruh evaluasi pengujian.

Tabel 3.5 Hyperparameter Pelatihan

Hyperparameter	Nilai
Kerangka kerja	Ultralytics YOLO, versi 8.4.60
Epochs	100 (tetap, early stopping dinonaktifkan)
Ukuran masukan (imgsz)	960 × 960 piksel
Batch size	6 (nilai tetap, tidak dipilih otomatis)
Optimizer	auto (dipilih otomatis oleh kerangka kerja berdasarkan konfigurasi)
Learning rate awal (lr0)	0,01
Learning rate akhir (lrf)	0,01 (faktor pengali terhadap lr0 pada akhir jadwal)
Momentum	0,937
Weight decay	0,0005
Warmup epochs	3,0
Cosine LR schedule	Tidak diaktifkan (cos_lr = false)
Patience (early stopping)	0 (dinonaktifkan; pelatihan berjalan penuh 100 epoch)
Automatic Mixed Precision (AMP)	Diaktifkan
Checkpoint yang dipakai	best.pt (performa validasi terbaik)
Seed acak	0, 1, 2 (tiga pengulangan per konfigurasi)
Perangkat keras	GPU NVIDIA RTX 3050

Untuk memperhitungkan variabilitas stokastik pelatihan, yaitu inisialisasi bobot dan pengacakan data, setiap konfigurasi dilatih ulang dengan tiga seed acak menggunakan konfigurasi yang identik. Hasil dilaporkan sebagai rata-rata beserta simpangan baku, dan perbandingan antararsitektur dilakukan pada seed yang berpasangan. Tabel 3.5 merinci seluruh hyperparameter pelatihan, diambil langsung dari berkas konfigurasi (args.yaml) tiap run agar tercatat sebagai nilai yang benar-benar digunakan, bukan nilai bawaan yang diasumsikan.

Tabel 3.5 menunjukkan bahwa seluruh hyperparameter, kecuali seed acak, ditetapkan identik pada kedua arsitektur dan kedua kondisi data, sehingga selisih performa yang diamati dapat diatribusikan pada arsitektur atau data, bukan pada perbedaan pengaturan optimisasi. Nilai optimizer, learning rate, dan momentum ditentukan melalui mode otomatis (auto) milik Ultralytics, yang memilih konfigurasi berdasarkan karakteristik dataset dan arsitektur; karena mode ini deterministik terhadap masukan yang sama, kedua arsitektur menerima perlakuan yang setara.

3.5 Metrik Evaluasi

Performa model dinilai menggunakan metrik deteksi objek standar, yaitu precision, recall, mean average precision pada ambang IoU 0,50 (mAP50), dan mean average precision yang dirata-ratakan atas ambang IoU 0,50 hingga 0,95 (mAP50-95). Confusion matrix ternormalisasi digunakan sebagai pelengkap untuk memeriksa distribusi kesalahan antarkelas dan terhadap kategori background. Karena setiap konfigurasi dilatih ulang dengan tiga seed, hasil dilaporkan sebagai rata-rata beserta simpangan baku sampel ($n-1$) atas ketiga seed tersebut, konvensi yang diterapkan secara konsisten pada seluruh tabel hasil pada laporan ini.

Untuk menilai kesesuaian dengan penerapan berbiaya rendah, indikator efisiensi berikut turut dicatat pada setiap model, yaitu jumlah parameter, floating-point operations (FLOPs), ukuran model, dan kecepatan inferensi (FPS). FPS dihitung dari waktu inferensi murni pada batch sama dengan satu di GPU yang sama, dirata-ratakan atas seluruh subset pengujian setelah tahap pemanasan. Seluruh metrik akurasi dihitung pada subset pengujian primer terkunci yang identik untuk menjamin komparabilitas antara keempat run. Hasil pengukuran efisiensi disajikan pada Tabel 3.6. Untuk menguji independensi temuan ini terhadap jenis perangkat keras, kecepatan inferensi juga diukur ulang pada CPU dengan protokol pengukuran yang identik; kedua hasil disajikan bersama pada Tabel 3.6.

Tabel 3.6 Kompleksitas dan Kecepatan Model pada GPU dan CPU (imgsz 960, batch 1)

Model	Parameter	GFLOPs	Ukuran (MB)	Inferensi GPU (ms)	FPS GPU	Inferensi CPU (ms)	FPS CPU
YOLOv8n	3.011.433	18,4	6,3	9,0 ± 1,7	111,3	93,2 ± 8,5	10,7
YOLO11n	2.590.425	14,5	5,5	11,7 ± 1,3	85,6	94,7 ± 5,4	10,6

Tabel 3.6 menjadi rujukan efisiensi yang dibahas bersama hasil akurasi deteksi, sehingga perbandingan kedua arsitektur mempertimbangkan baik performa maupun biaya komputasi secara bersamaan. Meskipun YOLO11n unggul pada ketiga indikator yang melekat pada arsitektur (2.590.425 parameter, 14,5 GFLOPs, dan ukuran berkas 5,5 MB, seluruhnya lebih kecil dibanding YOLOv8n), kecepatan inferensi pada GPU justru berbalik: YOLOv8n mencapai 111,3 FPS berbanding 85,6 FPS pada YOLO11n, sedangkan pada CPU kedua arsitektur nyaris setara (10,7 berbanding 10,6 FPS). Pola ini menunjukkan bahwa keunggulan efisiensi arsitektural YOLO11n tidak serta-merta diterjemahkan menjadi kecepatan inferensi GPU yang lebih tinggi, kemungkinan karena modul atensi C2PSA menambah overhead komputasi yang tidak tertangkap oleh hitungan FLOPs murni.

Sebagai pelengkap evaluasi kuantitatif pada citra statis, penelitian ini juga melakukan evaluasi kualitatif pada video jalan menggunakan mekanisme tracking objek yang disediakan kerangka kerja Ultralytics. Karena objek kerusakan pada rekaman video dapat muncul pada banyak frame berurutan, penghitungan langsung dari deteksi per-frame akan menghitung objek fisik yang sama berulang kali. Oleh karena itu, digunakan metrik jumlah objek unik hasil tracking per kelas, yaitu banyaknya identitas (ID) berbeda yang dihasilkan algoritma tracking sepanjang video, dibandingkan terhadap jumlah objek sebenarnya (ground truth) yang dihitung manual oleh peneliti pada video yang sama. Perlu ditegaskan bahwa mekanisme tracking ini bukan bagian dari arsitektur YOLOv8n maupun YOLO11n itu sendiri, melainkan algoritma terpisah bawaan Ultralytics—ByteTrack (Zhang et al., 2022) dan BoT-SORT (Aharon et al., 2022)—yang diterapkan identik pada kedua arsitektur, sehingga selisih hasil antararsitektur mencerminkan perbedaan kualitas deteksi yang menjadi masukan tracker, bukan perbedaan algoritma tracking itu sendiri.

Evaluasi ini menggunakan tiga video jalan berdurasi pendek (18–32 detik, resolusi 1080×1920, sekitar 30 FPS) yang direkam di Kabupaten Sleman, Daerah Istimewa

Yogyakarta, di luar subset pengujian primer yang telah dipakai pada evaluasi utama. Jumlah objek sebenarnya per kelas pada tiap video dihitung manual oleh peneliti sebagai ground truth. Karena ketiga video berasal dari satu kabupaten dan tidak mencakup kelima provinsi pada dataset primer, hasil evaluasi ini bersifat ilustratif dan tidak dimaksudkan mewakili generalisasi penuh ke seluruh wilayah cakupan penelitian.

BAB IV
Hasil dan Pembahasan

4.1 Perbandingan Performa Keseluruhan

Tabel 4.1 merangkum performa keempat konfigurasi, yaitu dua arsitektur pada dua kondisi data, sebagai rata-rata beserta simpangan baku atas tiga seed, yang dievaluasi pada subset pengujian primer terkunci yang identik.

Tabel 4.1 Performa Deteksi Keseluruhan (rata-rata ± simpangan baku, n = 3 seed)

Model	Kondisi data	Precision	Recall	mAP50	mAP50-95
YOLOv8n	A (primer)	0,418 ± 0,113	0,365 ± 0,064	0,314 ± 0,015	0,128 ± 0,006
YOLOv8n	B (+sekunder)	0,455 ± 0,061	0,371 ± 0,008	0,357 ± 0,013	0,138 ± 0,006
YOLO11n	A (primer)	0,466 ± 0,043	0,385 ± 0,027	0,354 ± 0,006	0,144 ± 0,003
YOLO11n	B (+sekunder)	0,477 ± 0,054	0,386 ± 0,019	0,369 ± 0,017	0,151 ± 0,013

Hasil menunjukkan dua pola yang saling berkaitan. Pada kondisi data primer (A), YOLO11n memperoleh performa yang lebih tinggi daripada YOLOv8n, yaitu mAP50 0,354 dibanding 0,314 dan mAP50-95 0,144 dibanding 0,128, dengan selisih yang konsisten di ketiga seed. YOLO11n juga menunjukkan simpangan baku yang lebih kecil pada kondisi ini, yaitu mAP50 ± 0,006 dibanding ± 0,015 dan mAP50-95 ± 0,003 dibanding ± 0,006, yang mengindikasikan performa yang lebih stabil antar-seed. Pada kondisi data gabungan (B), pola serupa berlanjut: YOLO11n-B mencapai mAP50 0,369 dan mAP50-95 0,151, sedikit di atas YOLOv8n-B (0,357 dan 0,138). Penambahan data sekunder meningkatkan mAP50 pada kedua arsitektur (YOLOv8n: 0,314→0,357; YOLO11n: 0,354→0,369), menunjukkan bahwa manfaat data tambahan konsisten terlepas dari arsitektur yang digunakan. Precision dan recall menunjukkan pola yang sama: YOLO11n unggul tipis pada precision (0,466-0,477 berbanding 0,418-0,455) dan recall (0,385-0,386 berbanding 0,365-0,371) pada kedua kondisi data.

Pada kondisi data gabungan (B), selisih antararsitektur menyempit, yaitu mAP50 0,369 dibanding 0,357. Penyempitan ini berkaitan dengan besarnya pengaruh penambahan data pada masing-masing arsitektur. Penambahan data sekunder meningkatkan mAP50 YOLOv8n secara substansial dari 0,314 menjadi 0,357 dengan selisih 0,043, sedangkan pada

YOLO11n peningkatannya lebih kecil, yaitu dari 0,354 menjadi 0,369 dengan selisih 0,015. Dengan kata lain, YOLO11n mencapai performa mendekati konfigurasi terbaiknya dengan data primer saja, sementara YOLOv8n memerlukan tambahan data untuk mendekatinya. Konfigurasi terbaik secara keseluruhan tetap YOLO11n dengan data gabungan.

Uji-t berpasangan antar seed menghasilkan p sebesar 0,047 untuk selisih mAP50 pada kondisi A dan p di atas 0,05 untuk perbandingan lainnya. Namun, karena jumlah seed terbatas pada tiga, nilai p tersebut bersifat indikatif; bukti utama pada penelitian ini adalah rata-rata beserta simpangan baku dan konsistensi arah selisih antar-seed.

4.2 Hasil Per-Seed

Tabel 4.1 dan Tabel 4.6 melaporkan performa sebagai rata-rata dan simpangan baku atas tiga seed. Untuk menunjukkan sumber angka tersebut secara transparan, Tabel 4.2 merinci precision, recall, mAP50, dan mAP50-95 keseluruhan pada tiap satu dari dua belas run (empat konfigurasi × tiga seed), diikuti baris rata-rata ± simpangan baku pada setiap konfigurasi. Seluruh run dievaluasi pada subset pengujian primer terkunci yang identik (140 citra, 210 instance).

Tabel 4.2 Performa Keseluruhan per Seed (12 Run)

Konfigurasi	**Seed**	**Precision**	**Recall**	**mAP50**	**mAP50-95**
YOLOv8n-A	seed0	0,334	0,431	0,322	0,131
YOLOv8n-A	seed1	0,374	0,362	0,325	0,133
YOLOv8n-A	seed2	0,547	0,302	0,297	0,121
YOLOv8n-A	*rata-rata ± std*	*0,418 ± 0,113*	*0,365 ± 0,064*	*0,314 ± 0,015*	*0,128 ± 0,006*
YOLOv8n-B	seed0	0,388	0,376	0,341	0,131
YOLOv8n-B	seed1	0,468	0,363	0,365	0,142
YOLOv8n-B	seed2	0,508	0,375	0,364	0,141
YOLOv8n-B	*rata-rata ± std*	*0,455 ± 0,061*	*0,371 ± 0,008*	*0,357 ± 0,013*	*0,138 ± 0,006*
YOLO11n-A	seed0	0,464	0,355	0,348	0,141
YOLO11n-A	seed1	0,509	0,408	0,361	0,144
YOLO11n-A	seed2	0,424	0,392	0,353	0,148
YOLO11n-A	*rata-rata ± std*	*0,466 ± 0,043*	*0,385 ± 0,027*	*0,354 ± 0,006*	*0,144 ± 0,003*
YOLO11n-B	seed0	0,489	0,366	0,368	0,166
YOLO11n-B	seed1	0,524	0,404	0,386	0,146
YOLO11n-B	seed2	0,418	0,388	0,352	0,142
YOLO11n-B	*rata-rata ± std*	*0,477 ± 0,054*	*0,386 ± 0,019*	*0,369 ± 0,017*	*0,151 ± 0,013*

Rata-rata pada Tabel 4.2 memulihkan angka pada Tabel 4.1 (selisih di bawah 0,001 pada seluruh sel setelah pembulatan), sehingga kedua tabel konsisten. Pola antar-seed bervariasi menurut metrik: pada YOLOv8n-A, precision berkisar dari 0,334 (seed0) hingga 0,547 (seed2) sementara recall bergerak berlawanan arah, dari 0,431 turun ke 0,302—pertukaran precision-recall semacam ini lazim terjadi antar-seed karena ambang keyakinan optimal dapat bergeser tanpa banyak mengubah mAP. Sebaliknya, mAP50 dan mAP50-95 relatif lebih stabil antar-seed pada YOLO11n-A (simpangan baku mAP50 hanya 0,006), konsisten dengan pengamatan pada Tabel 4.1 bahwa YOLO11n pada kondisi data primer menunjukkan performa yang paling stabil di antara keempat konfigurasi. Simpangan baku terbesar pada mAP50 justru muncul pada YOLO11n-B (0,017), yang selanjutnya ditelusuri pada tingkat kelas berikut.

Tabel 4.3 memfokuskan pada kelas pothole, yang secara konsisten menunjukkan variabilitas antar-seed paling besar. Pada YOLO11n-B, mAP50-95 pothole berkisar dari 0,068 (seed2) hingga 0,114 (seed0), simpangan baku 0,021—lebih dari dua kali lipat simpangan baku mAP50-95 keseluruhan pada konfigurasi yang sama. Pola serupa, meski lebih ringan, teramati pada konfigurasi lain.

Tabel 4.3 Performa Kelas Pothole per Seed (12 Run)

Konfigurasi	**Seed**	**mAP50 (pothole)**	**mAP50-95 (pothole)**
YOLOv8n-A	seed0	0,213	0,080
YOLOv8n-A	seed1	0,207	0,067
YOLOv8n-A	seed2	0,224	0,076
YOLOv8n-A	*rata-rata ± std*	*0,215 ± 0,009*	*0,075 ± 0,007*
YOLOv8n-B	seed0	0,222	0,047
YOLOv8n-B	seed1	0,219	0,081
YOLOv8n-B	seed2	0,213	0,064
YOLOv8n-B	*rata-rata ± std*	*0,218 ± 0,004*	*0,064 ± 0,017*
YOLO11n-A	seed0	0,229	0,089
YOLO11n-A	seed1	0,243	0,086
YOLO11n-A	seed2	0,226	0,094
YOLO11n-A	*rata-rata ± std*	*0,233 ± 0,009*	*0,089 ± 0,004*
YOLO11n-B	seed0	0,292	0,114
YOLO11n-B	seed1	0,226	0,072
YOLO11n-B	seed2	0,196	0,068
YOLO11n-B	*rata-rata ± std*	*0,238 ± 0,049*	*0,085 ± 0,025*

Variabilitas ini konsisten dengan jumlah instance pothole yang paling sedikit pada subset pengujian (58 dari 210 instance), sehingga setiap kesalahan atau keberhasilan deteksi pada beberapa instance memberi pengaruh proporsional lebih besar terhadap metrik dibanding kelas retak yang instance-nya lebih banyak. Tabel 4.3 menunjukkan bahwa variabilitas antar-seed pada kelas pothole cukup besar, terutama pada YOLO11n-B: mAP50 berkisar dari 0,196 (seed2) hingga 0,292 (seed0), menghasilkan simpangan baku 0,049—jauh lebih besar dibanding ketiga konfigurasi lain (0,004 hingga 0,009). Sebaran ini konsisten dengan simpangan baku precision yang besar pada Tabel 4.1 dan menegaskan bahwa performa pothole adalah metrik yang paling berisik (noisy) di antara ketiga kelas kerusakan. Karena variabilitas ini memengaruhi seed mana yang paling representatif terhadap rata-rata, pemilihan seed representatif untuk Gambar 4.2 dan Gambar 4.3 pada penelitian ini didasarkan pada kedekatan mAP50 keseluruhan (bukan performa satu kelas) terhadap rata-rata tiga seed, sebagaimana dijelaskan pada keterangan gambar terkait.

Tabel 4.4 melengkapi pemecahan per-seed untuk kelas longitudinal_crack, mengikuti urutan kelas eksplisit pothole=0, longitudinal_crack=1, alligator_crack=2 sebagaimana didefinisikan pada konfigurasi dataset.

Tabel 4.4 Performa Kelas Longitudinal Crack per Seed (12 Run)

Konfigurasi	**Seed**	**mAP50 (longitudinal_crack)**	**mAP50-95 (longitudinal_crack)**
YOLOv8n-A	seed0	0,342	0,150
YOLOv8n-A	seed1	0,396	0,175
YOLOv8n-A	seed2	0,313	0,140
YOLOv8n-A	rata-rata ± std	0,350 ± 0,042	0,155 ± 0,018
YOLOv8n-B	seed0	0,416	0,188
YOLOv8n-B	seed1	0,401	0,173
YOLOv8n-B	seed2	0,434	0,185
YOLOv8n-B	rata-rata ± std	0,417 ± 0,016	0,182 ± 0,008
YOLO11n-A	seed0	0,425	0,170
YOLO11n-A	seed1	0,415	0,187
YOLO11n-A	seed2	0,403	0,187
YOLO11n-A	rata-rata ± std	0,414 ± 0,011	0,181 ± 0,010
YOLO11n-B	seed0	0,383	0,185
YOLO11n-B	seed1	0,450	0,170
YOLO11n-B	seed2	0,419	0,169
YOLO11n-B	rata-rata ± std	0,417 ± 0,033	0,175 ± 0,009

Rata-rata pada Tabel 4.4 memulihkan angka pada Tabel 4.6 (mAP50 0,350/0,417/0,414/0,417 dan mAP50-95 0,155/0,182/0,181/0,175 berturut-turut untuk YOLOv8n-A/YOLOv8n-B/YOLO11n-A/YOLO11n-B, selisih di bawah 0,001 pada seluruh sel setelah pembulatan), sehingga kedua tabel konsisten. Simpangan baku mAP50 pada kelas ini (0,042/0,016/0,011/0,033) justru lebih besar dibanding simpangan baku pothole pada Tabel 4.3 (0,009/0,004/0,009/0,049) pada tiga dari empat konfigurasi, kecuali pada YOLO11n-B, di mana simpangan baku pothole (0,049) melampaui simpangan baku longitudinal_crack (0,033). Dengan demikian, tidak ditemukan pola konsisten bahwa kelas retak berbentuk lebih linier ini lebih stabil antar-seed dibanding pothole; variabilitas pothole yang besar tampak terkonsentrasi pada satu konfigurasi tertentu, bukan berlaku umum di seluruh konfigurasi. Tabel 4.5 melengkapi pemecahan per-seed untuk kelas alligator_crack, dengan urutan kelas yang sama.

Tabel 4.5 Performa Kelas Alligator Crack per Seed (12 Run)

Konfigurasi	**Seed**	**mAP50 (alligator_crack)**	**mAP50-95 (alligator_crack)**
YOLOv8n-A	seed0	0,410	0,162
YOLOv8n-A	seed1	0,371	0,157
YOLOv8n-A	seed2	0,354	0,148
YOLOv8n-A	rata-rata ± std	0,378 ± 0,029	0,155 ± 0,007
YOLOv8n-B	seed0	0,387	0,159
YOLOv8n-B	seed1	0,474	0,173
YOLOv8n-B	seed2	0,445	0,175
YOLOv8n-B	rata-rata ± std	0,435 ± 0,044	0,169 ± 0,009
YOLO11n-A	seed0	0,391	0,166
YOLO11n-A	seed1	0,424	0,160
YOLO11n-A	seed2	0,431	0,162
YOLO11n-A	rata-rata ± std	0,415 ± 0,022	0,163 ± 0,003
YOLO11n-B	seed0	0,427	0,199
YOLO11n-B	seed1	0,484	0,197
YOLO11n-B	seed2	0,441	0,188
YOLO11n-B	rata-rata ± std	0,451 ± 0,029	0,194 ± 0,006

Rata-rata pada Tabel 4.5 memulihkan angka pada Tabel 4.6 (mAP50 0,378/0,435/0,415/0,451 dan mAP50-95 0,155/0,169/0,163/0,194 berturut-turut untuk

YOLOv8n-A/YOLOv8n-B/YOLO11n-A/YOLO11n-B, selisih di bawah 0,001 pada seluruh sel setelah pembulatan), sehingga kedua tabel konsisten. Simpangan baku mAP50 pada kelas ini (0,029/0,044/0,022/0,029) juga lebih besar dibanding pothole pada tiga dari empat konfigurasi, sejalan dengan pola pada longitudinal_crack: variabilitas pothole yang tinggi tampak spesifik pada konfigurasi YOLO11n-B, bukan ciri umum yang membuat pothole sebagai kelas paling tidak stabil antar-seed secara keseluruhan.

4.3 Analisis Per-Kelas

Tabel 4.6 menyajikan performa per kelas sebagai rata-rata beserta simpangan baku atas tiga seed. Nilai ini konsisten dengan Tabel 4.1, karena rata-rata mAP ketiga kelas pada setiap konfigurasi memulihkan nilai keseluruhan pada Tabel 4.1.

Tabel 4.6 Performa Per Kelas (rata-rata ± simpangan baku, n = 3 seed)

Kelas	Metrik	YOLOv8n-A	YOLOv8n-B	YOLO11n-A	YOLO11n-B
pothole	mAP50	0,215 ± 0,009	0,218 ± 0,004	0,233 ± 0,009	0,238 ± 0,049
pothole	mAP50-95	0,075 ± 0,007	0,064 ± 0,017	0,089 ± 0,004	0,085 ± 0,025
longitudinal_crack	mAP50	0,350 ± 0,042	0,417 ± 0,016	0,414 ± 0,011	0,417 ± 0,033
longitudinal_crack	mAP50-95	0,155 ± 0,018	0,182 ± 0,008	0,181 ± 0,010	0,175 ± 0,009
alligator_crack	mAP50	0,378 ± 0,029	0,435 ± 0,044	0,415 ± 0,022	0,451 ± 0,029
alligator_crack	mAP50-95	0,155 ± 0,007	0,169 ± 0,009	0,163 ± 0,003	0,194 ± 0,006

Beberapa pola teramati. Pertama, pothole konsisten menjadi kelas dengan performa terendah pada seluruh konfigurasi, dengan mAP50 sekitar 0,215 hingga 0,238, sekaligus memiliki simpangan baku terbesar, terutama pada YOLO11n-B, sejalan dengan jumlah instance pothole yang paling sedikit pada subset pengujian. Kedua, pada kondisi data primer, YOLO11n memperoleh mAP50 yang lebih tinggi daripada YOLOv8n pada ketiga kelas. Ketiga, pengaruh penambahan data sekunder terpusat pada kelas retak: pada YOLOv8n, mAP50 longitudinal_crack meningkat dari 0,350 menjadi 0,417 dan alligator_crack dari 0,378 menjadi 0,435; pada YOLO11n, peningkatan terjelas teramati pada alligator_crack dari 0,415 menjadi 0,451. Sebaliknya, mAP50 pothole hampir tidak berubah pada kedua arsitektur.

4.4 Efisiensi Model

Tabel 3.6 melaporkan indikator efisiensi kedua arsitektur. YOLO11n memiliki parameter yang lebih sedikit, yaitu 2.590.425 dibanding 3.011.433, FLOPs yang lebih rendah, yaitu 14,5 dibanding 18,4 GFLOPs, dan ukuran berkas yang lebih kecil, yaitu 5,5 dibanding 6,3 MB. Meskipun demikian, waktu inferensi YOLO11n lebih lama, yaitu 11,7 ± 1,3 ms dibanding 9,0 ± 1,7 ms per citra, sehingga menghasilkan laju yang lebih rendah, yaitu 85,6 dibanding 111,3 FPS pada perangkat yang sama. Kedua model beroperasi jauh di atas ambang waktu-nyata yang umum, yaitu 30 FPS.

Untuk melengkapi penilaian efisiensi pada perangkat berbiaya rendah, kecepatan inferensi kedua arsitektur turut diukur pada mode CPU-only sebagai proksi komputasi tanpa akselerator perangkat keras, yang lebih merepresentasikan lingkungan penerapan berbiaya rendah dibandingkan GPU. Pengukuran dilakukan pada prosesor Intel64 Family 6 Model 154 Stepping 3 (16 inti) menggunakan protokol yang sama dengan pengukuran GPU, yaitu ukuran masukan 960, batch satu, dan tahap pemanasan sebelum pengukuran. Hasilnya disajikan pada Tabel 4.7.

Tabel 4.7 Perbandingan Kecepatan Inferensi pada GPU dan CPU

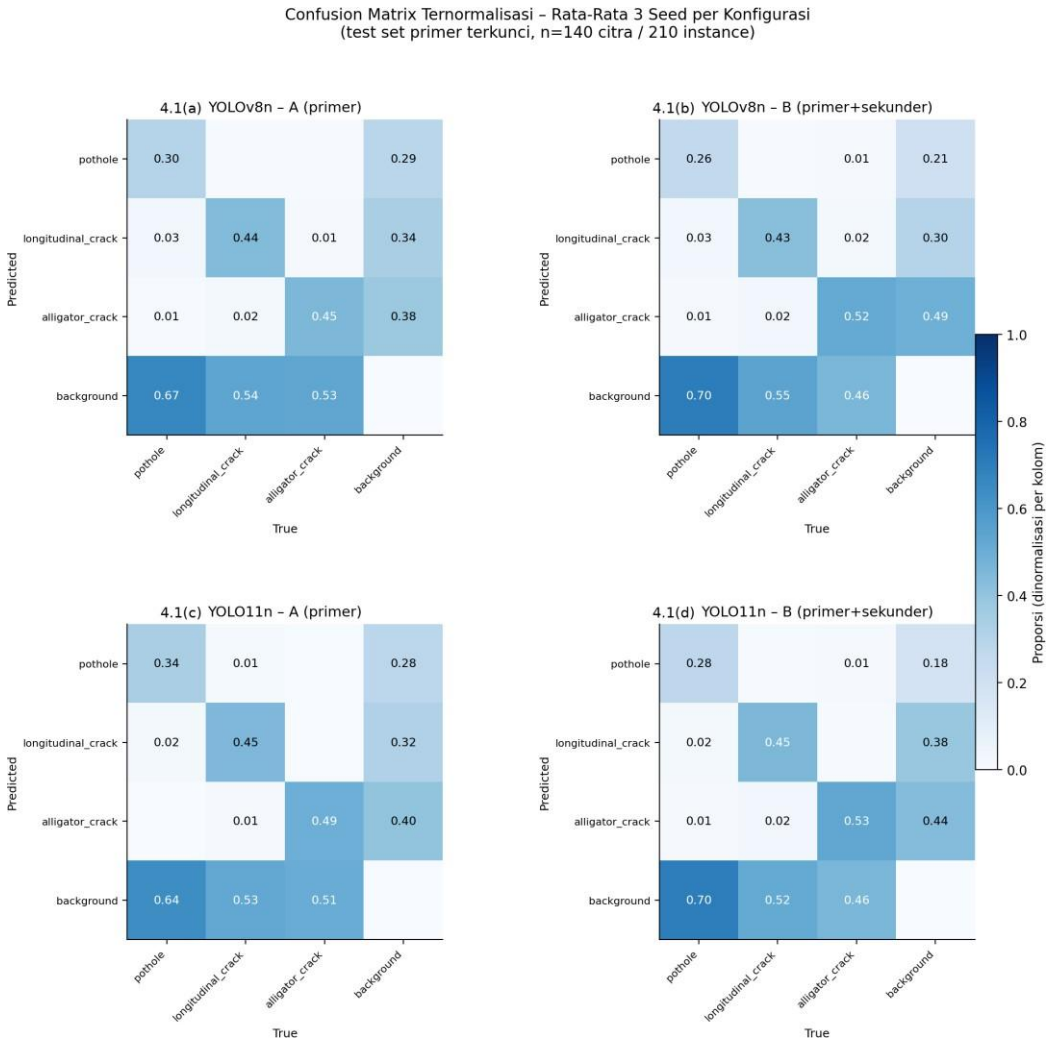
Model	Inferensi GPU (ms)	FPS GPU	Inferensi CPU (ms)	FPS CPU
YOLOv8n	9,0 ± 1,7	111,3	93,2 ± 8,5	10,7
YOLO11n	11,7 ± 1,3	85,6	94,7 ± 5,4	10,6

Pada GPU, YOLO11n memiliki waktu inferensi yang lebih lama daripada YOLOv8n meskipun jumlah parameter dan FLOPs-nya lebih rendah. Sebaliknya, pada CPU, waktu inferensi kedua arsitektur praktis setara, yaitu 94,7 ± 5,4 ms (10,6 FPS) dibanding 93,2 ± 8,5 ms (10,7 FPS) dengan selisih 1,5 ms yang berada dalam rentang simpangan baku, sehingga kerugian kecepatan YOLO11n yang teramati pada GPU tidak muncul pada CPU. Pola ini mengindikasikan bahwa keunggulan efisiensi komputasi YOLO11n yang tercermin pada FLOPs yang lebih rendah menjadi lebih relevan pada perangkat tanpa akselerator. Perlu ditegaskan bahwa indikator utama efisiensi model, yaitu jumlah parameter, FLOPs, dan ukuran model, tidak bergantung pada perangkat, sehingga keunggulan YOLO11n pada ketiga indikator tersebut berlaku konsisten lintas perangkat, sedangkan nilai FPS absolut akan berbeda menurut perangkat yang digunakan. Dengan demikian, penilaian potensi penerapan berbiaya rendah pada penelitian ini bertumpu terutama pada indikator yang tidak

bergantung perangkat, bukan semata pada nilai FPS absolut yang diukur pada satu perangkat tertentu.

4.5 Analisis Kesalahan

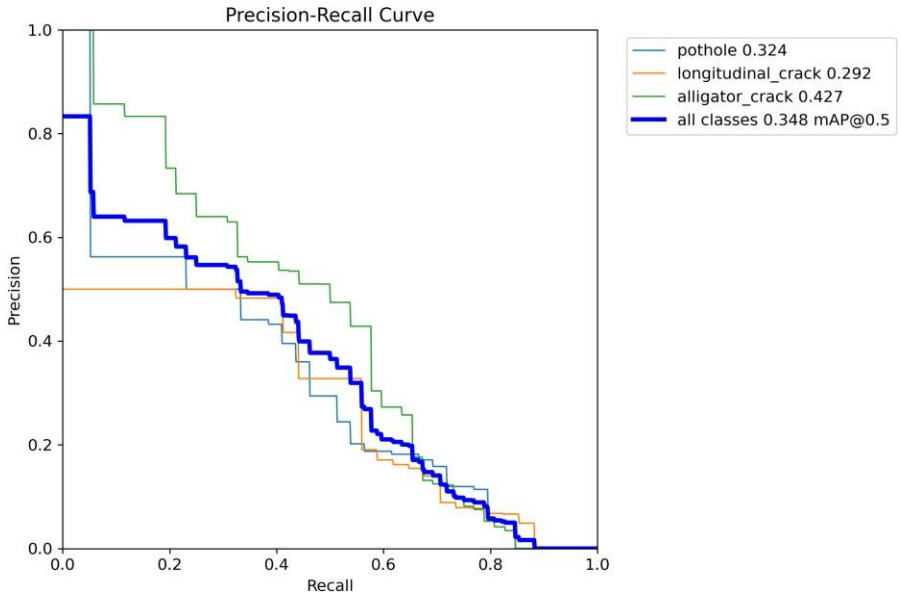
Confusion matrix ternormalisasi yang dirata-ratakan atas tiga seed untuk tiap konfigurasi menunjukkan pola kesalahan yang konsisten di seluruh konfigurasi dan seluruh seed. Nilai antar-kelas mendekati nol, umumnya kurang dari atau sama dengan 0,03, sedangkan proporsi objek asli yang tidak terdeteksi dan jatuh ke kategori background mendominasi, berkisar sekitar 0,46 hingga 0,70. Dengan demikian, kesalahan model didominasi oleh objek yang gagal terdeteksi atau false negative, bukan oleh kekeliruan menetapkan kelas.



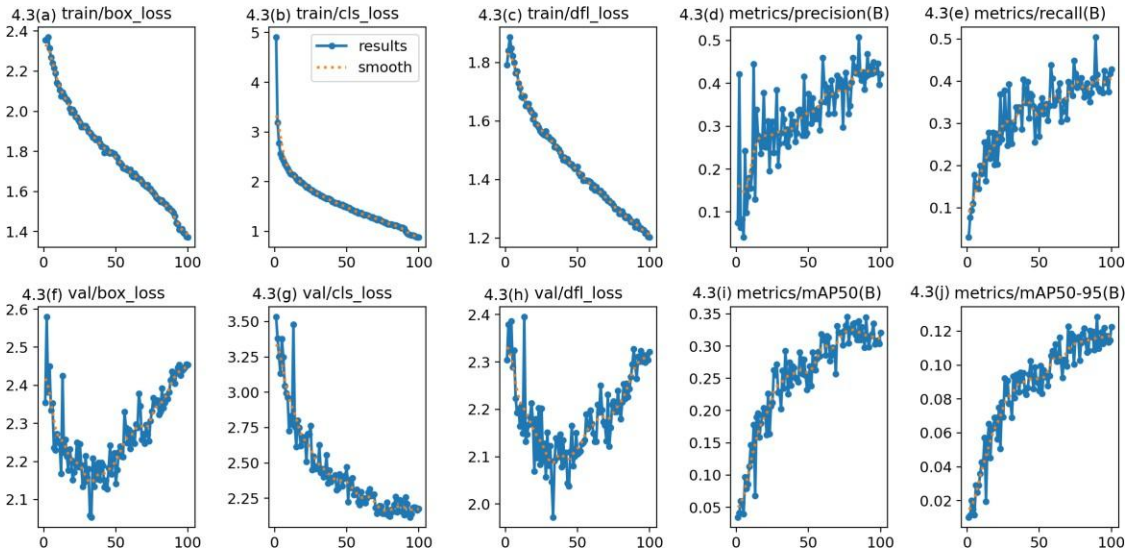
Gambar 4.1 Confusion matrix ternormalisasi, rata-rata 3 seed, keempat konfigurasi

Kelas pothole secara konsisten memperlihatkan proporsi objek terlewat tertinggi, yaitu rata-rata sekitar 0,64 hingga 0,70 ke background, sedangkan kedua kelas retak lebih rendah, yaitu sekitar 0,46 hingga 0,55. Karena pola ini bertahan pada ketiga seed, ia merupakan karakteristik model dalam pengaturan ini, bukan artefak satu kali pelatihan. Pola ini ditunjukkan secara konsisten pada seluruh panel Gambar 4.1(a)–4.1(d), yang masing-masing menyajikan YOLOv8n-A, YOLOv8n-B, YOLO11n-A, dan YOLO11n-B secara berurutan. Kurva Precision-Recall memperkuat pola tersebut, dengan alligator_crack umumnya berada di atas longitudinal_crack dan pothole, konsisten dengan urutan performa per-kelas pada Tabel 4.6: pada konfigurasi YOLO11n-B yang digunakan sebagai representasi, mAP50 alligator_crack (0,451) sedikit lebih tinggi daripada longitudinal_crack (0,417), sedangkan pothole jauh tertinggal (0,238). Kurva ditunjukkan pada Gambar 4.2, diambil dari seed representatif YOLO11n-B (seed dengan mAP50 keseluruhan paling dekat ke rata-rata tiga seed pada Tabel 4.2).

Kurva pelatihan pada Gambar 4.3, juga dari run representatif YOLO11n-B seed0, menunjukkan pola yang menjadi dasar penggunaan checkpoint validasi terbaik. Rugi lokalisasi pada subset validasi (val/box_loss, panel 4.3(f)) menurun dari 2,354 pada epoch 1 menjadi minimum 2,054 pada epoch 33, kemudian berbalik naik hingga 2,454 pada epoch 100; rugi distribusi fokal (val/dfl_loss, panel 4.3(h)) menunjukkan pola serupa, dari 1,972 pada titik minimum epoch 33 menjadi 2,322 pada epoch 100. Sebaliknya, rugi klasifikasi pada validasi (val/cls_loss, panel 4.3(g)) terus membaik dari 3,534 pada epoch 1 menjadi 2,176 pada epoch 100 (minimum 2,116 pada epoch 94). Dengan demikian, peningkatan rugi validasi setelah epoch 33 bersumber dari komponen lokalisasi kotak pembatas (box dan dfl), bukan dari komponen klasifikasi kelas—sebuah nuansa yang relevan karena menunjukkan bahwa overfitting pada pengaturan ini memengaruhi ketepatan posisi kotak lebih daripada ketepatan penentuan jenis kerusakan. Rugi pelatihan (train/box_loss, panel 4.3(a), dan train/cls_loss, panel 4.3(b)) terus menurun sepanjang 100 epoch tanpa tanda berbalik arah, sehingga kesenjangan yang melebar antara rugi pelatihan dan rugi validasi pada komponen lokalisasi merupakan indikasi overfitting standar. Pola serupa—titik minimum val/box_loss pada rentang epoch 16 hingga 43 diikuti kenaikan hingga akhir pelatihan, sementara val/cls_loss terus membaik—teramati pada seluruh 11 run lain yang datanya tersedia, sehingga pola ini bersifat konsisten, bukan artefak satu run tertentu; hal ini menjadi dasar penggunaan checkpoint validasi terbaik, bukan checkpoint epoch terakhir, untuk seluruh evaluasi pada penelitian ini.



Gambar 4.2 Kurva Precision-Recall YOLO11n pada subset pengujian primer, dari seed representatif



Gambar 4.3 Kurva pelatihan (loss & metrik per epoch), dari seed representatif

Sebagai bukti visual atas dominasi kesalahan jenis false negative pada Gambar 4.1, Gambar 4.4 menampilkan satu contoh konkret dari subset pengujian: citra banten_l1_frame03181, yang berisi satu anotasi pothole sebenarnya, dievaluasi dengan model YOLO11n-B (seed0) pada ambang keyakinan default 0,25.



Gambar 4.4 Contoh Kasus False Negative pada Kelas Pothole

Gambar 4.4 mengilustrasikan konvensi visualisasi yang dipakai konsisten pada Gambar 4.4-4.6: panel 4.4(a) menandai kotak pembatas ground truth (hijau, dengan label nama kelas) hasil anotasi manual, sedangkan panel 4.4(b) menampilkan hasil prediksi model pada citra yang sama, dengan label merah "tidak terdeteksi" dan kotak putus-putus abu-abu pada lokasi ground truth apabila model tidak menghasilkan satu pun kotak prediksi di lokasi tersebut (kotak putus-putus bukan hasil model, melainkan penanda referensi lokasi). Pada Gambar 4.4, panel 4.4(a) menandai kotak pembatas sebenarnya untuk objek pothole; panel 4.4(b) menunjukkan bahwa model tidak menghasilkan satu pun kotak prediksi pada citra ini. Objek tersebut menempati area yang relatif kecil dan kontrasnya rendah terhadap permukaan jalan di sekitarnya, konsisten dengan karakteristik pothole yang diuraikan pada Bab 2 dan dengan pola bahwa pothole memiliki proporsi objek terlewat tertinggi di antara ketiga kelas (Gambar 4.1). Kasus ini mengilustrasikan mekanisme kesalahan yang mendasari temuan kuantitatif pada Bab ini: kegagalan model lebih sering berupa objek yang luput terdeteksi sepenuhnya, bukan kesalahan penentuan kelas pada objek yang berhasil ditemukan.

Dua contoh tambahan dari citra berbeda memperkuat pola tersebut dan sekaligus menunjukkan keragaman kondisi kegagalan deteksi. Kedua contoh diambil dari model dan test set yang sama dengan Gambar 4.4 (YOLO11n-B seed0, ambang keyakinan 0,25), dipilih secara terprogram melalui pencocokan IoU antara kotak sebenarnya dan prediksi, bukan pemeriksaan visual manual.



4.5(a) Ground truth 4.5(b) Prediksi model

Gambar 4.5 Contoh Kasus Tiga Pothole Terlewat Sekaligus

Mengikuti konvensi warna yang sama dengan Gambar 4.4, Gambar 4.5 menunjukkan tiga anotasi pothole pada satu citra (panel 4.5(a)) yang ketiganya tidak menghasilkan satu pun prediksi (panel 4.5(b)), meskipun ketiga area tampak sebagai genangan air dan tekstur permukaan yang tidak rata—ciri visual pothole yang biasanya cukup jelas. Sebagai ilustrasi mekanisme (bukan hasil ukur), kegagalan pada kasus ini kemungkinan terkait dengan kemiripan visual antara genangan air dan pantulan cahaya di sekitarnya dengan permukaan aspal basah yang tidak rusak, sehingga batas objek menjadi kabur bagi model.



4.6(a) Ground truth 4.6(b) Prediksi model

Gambar 4.6 Contoh Kasus Dua Pothole dengan Ukuran Berbeda Terlewat

Mengikuti konvensi warna yang sama dengan Gambar 4.4 (panel 4.6(a): ground truth; panel 4.6(b): prediksi model), Gambar 4.6 menunjukkan dua anotasi pothole dengan ukuran berbeda—satu kecil dan satu lebih besar—yang keduanya luput dari deteksi. Sebagai ilustrasi mekanisme, kegagalan pada objek yang lebih kecil sejalan dengan penjelasan pada Gambar 4.4 (area kecil relatif terhadap ukuran masukan 960 × 960 piksel), sedangkan luputnya objek yang lebih besar menunjukkan bahwa ukuran objek bukan satu-satunya faktor; jarak pengambilan citra yang jauh dan kontras permukaan jalan yang rendah pada

kondisi pencahayaan tertentu turut berperan sebagai kemungkinan penyebab, meskipun penelitian ini tidak mengukur kontribusi masing-masing faktor secara terpisah.

Secara keseluruhan, ketiga contoh (Gambar 4.4–4.6) memberikan bukti visual yang konsisten dengan Gambar 4.1: kesalahan yang dominan adalah objek yang gagal terdeteksi sama sekali (false negative ke background), bukan kekeliruan menetapkan kelas pada objek yang berhasil ditemukan. Pola ini juga konsisten dengan Tabel 4.3, yang menunjukkan variabilitas antar-seed paling besar justru pada kelas pothole.

4.6 Efisiensi Data sebagai Keunggulan Utama YOLO11n

Hasil pada Tabel 4.1 mengindikasikan bahwa keunggulan YOLO11n atas YOLOv8n paling menonjol pada rezim data terbatas. Pada kondisi data primer, YOLO11n tidak hanya memperoleh mAP yang lebih tinggi, tetapi juga performa yang lebih stabil antar-seed yang ditunjukkan oleh simpangan baku yang lebih kecil. Stabilitas antar-seed ini merupakan indikator yang kerap terabaikan namun penting: simpangan baku yang lebih kecil menandakan bahwa keunggulan YOLO11n bukan hasil satu kali pelatihan yang beruntung, melainkan cenderung berulang pada inisialisasi dan pengacakan data yang berbeda. Ketika data pelatihan diperbanyak dengan data sekunder, keunggulan tersebut menyempit karena YOLOv8n memperoleh peningkatan yang lebih besar. Pola ini konsisten dengan interpretasi bahwa YOLO11n lebih efisien-data, yaitu arsitektur ini mendekati performa terbaiknya dengan data yang lebih sedikit sehingga memperoleh manfaat marginal yang lebih kecil dari tambahan data dibanding YOLOv8n. Secara intuitif, jika suatu arsitektur telah mampu mengekstraksi fitur yang relevan dari data yang terbatas, maka tambahan data memberikan informasi baru yang lebih sedikit baginya dibandingkan bagi arsitektur yang belum mencapai titik tersebut; inilah yang tampak membedakan kedua model pada penelitian ini. Kecenderungan ini sejalan dengan deskripsi penyempurnaan arsitektur YOLO11 yang menekankan ekstraksi fitur yang lebih efektif dan efisiensi parameter (Khanam & Hussain, 2024; Ultralytics, 2024).

Temuan ini juga memberikan bukti empiris terhadap asumsi yang dinyatakan pada literatur. Xu dkk. (2025) menetapkan YOLOv8n sebagai model dasar dengan alasan bahwa strategi pemangkasan dinamis YOLO11 dinilai kurang spesifik untuk ekstraksi fitur retak. Hasil penelitian ini menunjukkan bahwa pada rezim data terbatas, YOLO11n justru memperoleh mAP yang lebih tinggi pada ketiga kelas, termasuk kedua kelas retak. Dengan demikian, asumsi tersebut tidak didukung pada pengaturan yang diuji di sini, meskipun perlu dicatat bahwa perbandingan dilakukan pada varian nano tanpa modifikasi arsitektur.

Efisiensi data ini relevan secara praktis. Pada pemantauan kerusakan jalan berbiaya rendah, data lapangan yang teranotasi kerap terbatas dan mahal diperoleh, sehingga arsitektur yang mencapai performa lebih baik dengan data lebih sedikit bernilai. Anotasi kerusakan jalan menuntut penandaan kotak pembatas secara manual pada ribuan citra oleh tenaga yang memahami jenis kerusakan, sehingga penghematan kebutuhan data berdampak langsung pada biaya dan waktu penyiapan sistem. Namun, keunggulan akurasi ini disertai waktu inferensi yang lebih lama, pola yang konsisten dengan pengamatan pada literatur bahwa varian YOLO yang lebih baru dapat mengorbankan kecepatan demi efisiensi parameter dan komputasi (Xu et al., 2025). Sebagaimana ditunjukkan pula pada Tabel 4.7, kerugian kecepatan YOLO11n yang teramati pada GPU tidak muncul pada CPU, sehingga bobot pertimbangan ini dapat berubah menurut perangkat yang menjadi sasaran penerapan.

4.7 Pengaruh Penambahan Data Sekunder

Penambahan data sekunder domain-Indonesia meningkatkan performa keseluruhan kedua arsitektur, tetapi besar pengaruhnya berbeda. Peningkatan lebih besar teramati pada YOLOv8n dibandingkan YOLO11n (mAP50 +0,043 berbanding +0,015). Analisis per kelas pada Tabel 4.6 memperjelas asal peningkatan tersebut, yaitu terpusat pada kelas retak, terutama longitudinal_crack dan alligator_crack pada YOLOv8n, sedangkan pothole hampir tidak terpengaruh pada kedua arsitektur. Terpusatnya perbaikan pada kelas retak menunjukkan bahwa tambahan data sekunder paling bermanfaat justru pada kelas yang karakteristik visualnya relatif lebih dapat dialihkan antar-sumber, sementara kelas yang lebih bergantung pada konteks lapangan tidak memperoleh manfaat serupa.

Temuan pada pothole ini perlu digarisbawahi karena tampak berlawanan dengan intuisi komposisi data. Sumber sekunder justru menyumbang mayoritas anotasi pothole, yaitu 2.657 dari 2.943 anotasi pothole gabungan sebagaimana disajikan pada Tabel 3.1, namun performa deteksi pothole pada subset pengujian primer tidak meningkat. Pola ini mengindikasikan kemungkinan kesenjangan domain untuk kelas pothole antara citra sekunder yang lebih terkurasi dan citra primer lapangan, sehingga tambahan contoh pothole dari sumber sekunder tidak serta-merta memperbaiki deteksi pada kondisi lapangan. Dugaan ini selaras dengan temuan pada penyusunan benchmark multi-negara RDD2022, yang mencatat bahwa model yang dilatih pada citra satu distribusi cenderung menurun performanya ketika diuji pada distribusi yang berbeda (Arya et al., 2021, 2024); dengan kata lain, kesamaan label kelas tidak menjamin kesamaan distribusi citra. Perbedaan sudut pengambilan, ketinggian kamera, pencahayaan, dan tekstur perkerasan antara data sekunder

yang terkurasi dan data primer lapangan merupakan sumber kesenjangan domain yang masuk akal. Interpretasi ini perlu disikapi hati-hati mengingat jumlah instance pothole yang sedikit pada subset pengujian, yaitu 58 instance, yang membuat metrik pothole memiliki variabilitas antar-seed yang lebih besar. Variabilitas ini dikuantifikasi lebih rinci pada Tabel 4.3 (subbab Hasil Per-Seed), dan contoh kualitatif kegagalan deteksi pothole ditunjukkan pada Gambar 4.4.

Secara kuantitatif, kontras ini tampak jelas pada Tabel 4.6: mAP50 pothole nyaris tidak bergerak akibat penambahan data sekunder (YOLOv8n 0,215 menjadi 0,218, naik 0,003; YOLO11n 0,233 menjadi 0,238, naik 0,005), sedangkan kelas retak naik jauh lebih besar—longitudinal_crack pada YOLOv8n naik dari 0,350 menjadi 0,417 (naik 0,067), dan alligator_crack naik pada kedua arsitektur (YOLOv8n dari 0,378 menjadi 0,435, naik 0,057; YOLO11n dari 0,415 menjadi 0,451, naik 0,036).

Secara keseluruhan, perbedaan besar pengaruh data antararsitektur memperkuat interpretasi efisiensi data. YOLOv8n memperoleh manfaat lebih besar dari tambahan contoh karena performa awalnya pada data primer lebih rendah, sedangkan YOLO11n telah mendekati performa terbaiknya dengan data primer.

4.8 Implikasi untuk Pemantauan Berbiaya Rendah

Kedua model bersifat ringkas dan beroperasi jauh di atas ambang waktu-nyata pada GPU kelas menengah, sehingga keduanya berpotensi dipertimbangkan untuk pemantauan kondisi jalan berbiaya rendah. Pilihan di antara keduanya menghadirkan pertimbangan yang jelas. YOLO11n lebih unggul pada akurasi serta lebih hemat parameter (2.590.425 berbanding 3.011.433, −14,0%), FLOPs (14,5 berbanding 18,4 GFLOPs, −21,2%), dan ukuran model (5,5 berbanding 6,3 MB, −12,7%), karakteristik yang menguntungkan potensi penerapan pada perangkat tepi dengan memori terbatas, dan lebih efisien-data. Sebaliknya, YOLOv8n memberikan laju inferensi yang lebih tinggi. Perlu ditegaskan bahwa keunggulan efisiensi YOLO11n pada jumlah parameter, FLOPs, dan ukuran model bersifat tidak bergantung perangkat, sehingga berlaku konsisten baik pada GPU maupun CPU; sebaliknya, keunggulan laju inferensi YOLOv8n bergantung pada perangkat dan, sebagaimana ditunjukkan Tabel 4.7, menyusut menjadi setara ketika pengukuran dilakukan pada CPU. Dengan demikian, penilaian potensi penerapan berbiaya rendah pada penelitian ini bertumpu terutama pada indikator yang tidak bergantung perangkat. Karena keduanya melampaui 30 FPS, kecepatan bukan penghambat utama pada konteks ini, sehingga efisiensi memori, efisiensi data, dan akurasi menjadi pertimbangan yang lebih menentukan. Pertimbangan

serupa antara akurasi dan biaya komputasi juga menjadi perhatian pada berbagai kajian deteksi kerusakan jalan untuk perangkat tepi (Geng et al., 2024; Wang et al., 2024a; Xu et al., 2025).

Secara ringkas, panduan berikut dapat menjadi acuan pemilihan di antara keduanya. YOLO11n lebih sesuai ketika prioritas utama adalah akurasi, efisiensi memori perangkat, atau efisiensi data pelatihan—misalnya pada perangkat tepi dengan penyimpanan terbatas, atau ketika anotasi lapangan mahal diperoleh sehingga performa yang baik dengan data lebih sedikit menjadi nilai tambah. Sebaliknya, YOLOv8n tetap relevan ketika prioritas utama adalah laju pemrosesan pada perangkat berbasis GPU, karena keunggulan kecepatannya nyata pada GPU meski menghilang pada CPU. Sebagai konteks tambahan, kecepatan CPU yang terukur pada kedua arsitektur (sekitar 10,6–10,7 FPS pada Tabel 4.7) berada di bawah laju video pemantauan jalan yang umum (sekitar 24–30 FPS); dengan demikian, deployment berbasis CPU pada penelitian ini lebih sesuai untuk pemrosesan rekaman secara batch/offline setelah perjalanan survei selesai, bukan untuk analisis langsung (live) selama perjalanan, sedangkan deployment berbasis GPU tetap mampu mengikuti laju video secara langsung pada kedua arsitektur.

4.9 Nilai Absolut dan Konteks Dataset

Nilai mAP pada penelitian ini, dengan mAP50 keseluruhan tertinggi 0,369, lebih rendah daripada sejumlah studi deteksi kerusakan jalan yang melaporkan mAP50 di atas 0,80, misalnya EVA-YOLOv8 dengan mAP50 89,7% (Liu et al., 2026) dan YOLOv8-GAM-Wise-IoU dengan mAP50 96,5% pada deteksi retak permukaan jembatan (Xiong et al., 2024). Perbedaan ini perlu ditafsirkan dengan mempertimbangkan perbedaan pengaturan, bukan semata perbedaan metode. Kedua studi tersebut menggunakan arsitektur yang dimodifikasi secara khusus dengan mekanisme atensi (Liu et al., 2026; Xiong et al., 2024), dataset benchmark yang lebih besar dan terkurasi, atau evaluasi pada distribusi data yang seragam. Sebagai perbandingan, studi dengan pengaturan yang lebih moderat melaporkan nilai yang lebih dekat dengan penelitian ini: mAP 62,9% pada studi lapangan di Palangkaraya (Jonathan et al., 2026), dan mAP50 70,6% pada YOLOv8n yang dimodifikasi dan dilatih pada benchmark RDD2022 (Zeng & Zhong, 2024).

Pola pada studi-studi yang disitir di atas juga tampak pada studi dengan model dasar tanpa modifikasi. Xu dkk. (2025) melaporkan bahwa YOLOv8n tanpa modifikasi sudah mencapai mAP50 93,8% pada dataset gabungan citra retak yang mereka susun sendiri (Crack Forest, CRACK500, dan citra swadaya, tanpa kelas pothole), sebelum modifikasi

CBAM dan WIoU menaikkannya lebih lanjut menjadi 95,6%. Temuan ini mengindikasikan bahwa varian nano tanpa modifikasi bukan pembatas utama capaian mAP secara umum; pada studi-studi yang disitir di sini, mAP tinggi lebih tampak berkaitan dengan karakteristik dataset yang dipakai—didominasi kelas retak yang relatif homogen dan terkurasi, tanpa kelas pothole yang secara konsisten menjadi kelas tersulit di seluruh studi yang disitir di sini, termasuk pada studi dengan arsitektur termodifikasi. Pola ini bersifat indikatif berdasarkan studi yang direview pada penelitian ini dan tidak serta-merta berlaku umum, karena literatur deteksi kerusakan jalan sangat luas dan tidak seluruhnya tercakup di sini.

Sebaliknya, penelitian ini menggunakan model nano tanpa modifikasi, data lapangan primer dengan kualitas akuisisi apa adanya, serta evaluasi pada subset pengujian lapangan yang dikunci. Ketiga pilihan tersebut, yaitu model dasar tanpa modifikasi, data lapangan apa adanya, dan pengujian yang dikunci, memang cenderung menghasilkan angka absolut yang lebih rendah, tetapi sekaligus membuat hasil lebih mencerminkan kondisi penerapan sebenarnya dan lebih dapat diperbandingkan secara adil antararsitektur. Deteksi retak juga dikenal menantang karena bentuk objek yang tipis, memanjang, dan berkontras rendah terhadap latar yang kompleks (Xu et al., 2025; Yu et al., 2024b). Perlu dicatat pula bahwa kesulitan pada kelas pothole bukan khas penelitian ini, karena studi pada benchmark besar dengan arsitektur termodifikasi pun melaporkan pothole sebagai kelas dengan akurasi terendah, yaitu 53,1% (Zeng & Zhong, 2024). Dengan demikian, nilai absolut yang lebih rendah mencerminkan pengaturan yang lebih menantang dan realistis, bukan kelemahan metode; nilai perbandingan antararsitektur dan antarkondisi data tetap sahih karena diukur pada pengaturan yang identik.

4.10 Keterbatasan

Beberapa keterbatasan perlu dicatat. Pertama, anotasi retak lateral pada sumber sekunder dihapus sementara citranya dipertahankan sebagai background; karena retak lateral dan longitudinal memiliki kemiripan bentuk garis, sebagian retak lateral yang tak berlabel dapat menjadi sinyal yang tidak konsisten dan berpotensi memengaruhi metrik longitudinal_crack. Kedua, jumlah data primer relatif kecil, yaitu 767 citra, dan model menunjukkan gejala overfitting, sehingga performa absolut kemungkinan terbatas oleh ketersediaan data. Ketiga, jumlah instance pothole pada subset pengujian relatif sedikit, yaitu 58 instance, sehingga metrik pothole memiliki derau yang lebih besar; simpangan baku precision yang cukup besar pada beberapa konfigurasi juga menunjukkan precision sebagai metrik yang paling berisik dalam pengaturan ini. Keempat, jumlah pengulangan seed

terbatas pada tiga, sehingga uji signifikansi statistik hanya bersifat indikatif. Kelima, evaluasi dilakukan pada data lapangan dari lima provinsi, sehingga generalisasi ke wilayah lain di luar cakupan ini masih perlu diuji. Keenam, evaluasi kecepatan inferensi dilakukan pada GPU kelas menengah (RTX 3050) dan CPU desktop, bukan pada perangkat tepi (edge) sesungguhnya seperti Raspberry Pi atau Jetson; meskipun demikian, kesetaraan kecepatan kedua arsitektur pada CPU memperkuat indikasi bahwa keunggulan efisiensi bersifat perangkat-independen, sedangkan pengujian pada perangkat tepi nyata tetap diperlukan sebagai langkah verifikasi lanjutan. Ketujuh, evaluasi tracking objek pada video bersifat awal dan terbatas: hanya tiga video berdurasi pendek dari satu kabupaten (Sleman, Daerah Istimewa Yogyakarta) yang diuji, sehingga belum mewakili kelima provinsi pada dataset primer; ground truth dihitung manual oleh peneliti tanpa verifikasi independen; dan hanya dua algoritma tracking bawaan Ultralytics (ByteTrack dan BoT-SORT) yang diujikan tanpa penalaan hyperparameter tracker.

4.11 Jawaban atas Pertanyaan Penelitian

Bagian ini merangkum temuan Bab ini sebagai jawaban langsung atas keempat pertanyaan penelitian yang diajukan pada Bab 1, tanpa mengulang uraian rinci yang telah disajikan pada subbab-subbab sebelumnya.

Pertanyaan pertama menanyakan perbandingan performa YOLO11n terhadap baseline YOLOv8n ketika keduanya dilatih dengan konfigurasi identik. Pada kondisi data primer (A), Tabel 4.1 menunjukkan YOLO11n-A memperoleh mAP50 0,354 ± 0,006 dibanding YOLOv8n-A sebesar 0,314 ± 0,015, dengan simpangan baku yang juga lebih kecil. Hasil ini mengindikasikan kecenderungan YOLO11n lebih unggul pada rezim data terbatas, meskipun dengan n=3 pengulangan seed, kesimpulan ini bersifat indikatif.

Pertanyaan kedua menanyakan sejauh mana penambahan data sekunder memengaruhi performa kedua arsitektur. Penambahan data sekunder meningkatkan mAP50 YOLOv8n dari 0,314 menjadi 0,357 (selisih +0,043), sedangkan YOLO11n meningkat dari 0,354 menjadi 0,369 (selisih +0,015). Selisih antararsitektur pada kondisi B karenanya menyempit dibanding kondisi A, mengindikasikan bahwa YOLOv8n memperoleh manfaat lebih besar dari tambahan data—pola yang konsisten dengan interpretasi YOLO11n sebagai arsitektur yang lebih efisien-data, yaitu mendekati performa terbaiknya dengan data primer yang lebih sedikit.

Pertanyaan ketiga menanyakan perbandingan efisiensi kedua arsitektur. Tabel 3.6 menunjukkan YOLO11n lebih hemat pada seluruh indikator yang tidak bergantung

perangkat: parameter 2.590.425 berbanding 3.011.433 (−14,0%), FLOPs 14,5 berbanding 18,4 GFLOPs (−21,2%), dan ukuran model 5,5 berbanding 6,3 MB (−12,7%). Pada kecepatan inferensi, keunggulan berbalik dan bergantung perangkat: pada GPU, YOLOv8n lebih cepat (9,0 berbanding 11,7 ms), tetapi Tabel 4.7 menunjukkan selisih ini menyusut menjadi setara pada CPU (93,2 berbanding 94,7 ms, selisih 1,5 ms berada dalam rentang simpangan baku). Dengan demikian, untuk konteks pemantauan berbiaya rendah, indikator yang tidak bergantung perangkat memberikan dasar perbandingan yang lebih konsisten daripada kecepatan absolut pada satu jenis perangkat keras saja.

Pertanyaan keempat menanyakan kondisi dan jenis kesalahan yang dominan. Confusion matrix pada Gambar 4.1 menunjukkan bahwa kesalahan antar-kelas mendekati nol ($\leq 0{,}03$) pada seluruh konfigurasi, sedangkan proporsi objek yang gagal terdeteksi (jatuh ke background) berkisar 0,46 hingga 0,70, dengan pothole secara konsisten menunjukkan proporsi objek terlewat tertinggi. Bukti visual pada Gambar 4.4–4.6 mengilustrasikan mekanisme di balik temuan ini: objek pothole yang berukuran kecil relatif terhadap resolusi masukan atau berbatas kabur dengan permukaan jalan di sekitarnya cenderung luput terdeteksi sepenuhnya, bukan salah diklasifikasikan sebagai kelas lain. Dengan demikian, kegagalan model pada penelitian ini didominasi oleh objek yang terlewat, bukan oleh kekeliruan menetapkan jenis kerusakan pada objek yang berhasil ditemukan.

4.12 Evaluasi Tambahan: Tracking Objek pada Video

Sebagaimana dijelaskan pada Bab 3.5, evaluasi tambahan dilakukan pada tiga video jalan untuk menilai kemampuan menghitung objek kerusakan secara unik melalui tracking objek, menggunakan model YOLOv8n dan YOLO11n pada kondisi data gabungan (B). Tabel 4.8 menyajikan perbandingan jumlah objek unik hasil tracking menggunakan ByteTrack (algoritma tracking bawaan/default Ultralytics) terhadap ground truth. Tabel 4.8 menunjukkan pola yang konsisten pada kedua arsitektur: kelas pothole cenderung dihitung kurang (undercounting) terhadap ground truth—sejalan dengan temuan Bab 4.5 bahwa pothole paling sering tidak terdeteksi sama sekali—sedangkan kelas alligator_crack dan longitudinal_crack justru dihitung jauh berlebih (overcounting), pada beberapa video mencapai 3 hingga 17 kali lipat ground truth.

Tabel 4.8 Perbandingan Jumlah Objek Unik Hasil Tracking (ByteTrack) terhadap Ground Truth

Video	Kelas	Ground Truth	YOLOv8n-B	YOLO11n-B
video1	pothole	17	5	4
video1	longitudinal_crack	1	0	2
video1	alligator_crack	6	31	13
video2	pothole	23	11	8
video2	longitudinal_crack	2	6	10
video2	alligator_crack	2	34	21
video3	pothole	35	22	21
video3	longitudinal_crack	2	0	3
video3	alligator_crack	4	17	22

Pola overcounting ini bukan disebabkan oleh deteksi berlebih, melainkan fragmentasi track: bentuk retak yang tidak beraturan dan memanjang menyebabkan kotak pembatas bergeser signifikan antar-frame, sehingga algoritma tracking berbasis kecocokan IoU gagal mengenali retak yang sama sebagai objek yang identik dan membentuk identitas baru berulang kali. Untuk menguji apakah keterbatasan ini spesifik pada algoritma ByteTrack, pengujian yang sama diulang dengan BoT-SORT, algoritma tracking alternatif bawaan Ultralytics yang menambahkan kompensasi gerakan kamera. Tabel 4.9 menyajikan hasilnya.

Tabel 4.9 Perbandingan Jumlah Objek Unik Hasil Tracking (BoT-SORT) terhadap Ground Truth

Video	Kelas	Ground Truth	YOLOv8n-B	YOLO11n-B
video1	pothole	17	7	3
video1	longitudinal_crack	1	0	1
video1	alligator_crack	6	24	11
video2	pothole	23	8	8
video2	longitudinal_crack	2	5	13
video2	alligator_crack	2	30	23
video3	pothole	35	8	12
video3	longitudinal_crack	2	0	2
video3	alligator_crack	4	13	22

Tabel 4.9 menunjukkan bahwa BoT-SORT tidak memperbaiki masalah secara meyakinkan: overcounting pada kelas alligator_crack memang sedikit berkurang pada beberapa video (misalnya dari 31 menjadi 24 objek pada video1 untuk YOLOv8n) tetapi tetap jauh di atas ground truth, sedangkan pada kelas pothole hasil BoT-SORT pada sebagian video justru lebih jauh dari ground truth dibanding ByteTrack (misalnya video3 pada YOLOv8n, dari 22 menjadi 8 objek, dibandingkan ground truth 35). Dengan demikian, keterbatasan yang teramati tidak spesifik pada satu algoritma tracking, melainkan mengindikasikan ketidakcocokan yang lebih mendasar antara asumsi algoritma tracking generik—yang dirancang untuk objek berbentuk relatif kompak dan kaku seperti kendaraan atau pejalan kaki—dengan karakteristik visual kerusakan jalan berbentuk retak yang amorf dan memanjang.Evaluasi ini melengkapi jawaban atas pertanyaan penelitian keempat (Bab 4.11) dengan menunjukkan bahwa keterbatasan model tidak hanya terjadi pada deteksi objek diam pada citra statis, tetapi juga pada penghitungan objek bergerak pada video ketika deteksi tersebut digabungkan dengan mekanisme tracking standar. Karena mekanisme tracking pada kedua arsitektur identik, temuan ini tidak mengubah kesimpulan perbandingan YOLOv8n dan YOLO11n pada subbab-subbab sebelumnya, tetapi menjadi catatan penting bagi penerapan praktis sistem pemantauan berbasis video.

BAB V
Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini membandingkan dua detektor satu tahap, yaitu YOLOv8n dan YOLO11n, untuk deteksi kerusakan jalan multi-kelas pada tiga kelas kerusakan dalam kondisi pelatihan yang identik, menggunakan dataset gabungan 4.088 citra jalan Indonesia yang memadukan data primer lapangan dari lima provinsi dengan data sekunder publik. Melalui rancangan terkendali 2 x 2 yang menyilangkan dua arsitektur dengan dua kondisi data dan dievaluasi pada subset pengujian primer terkunci yang identik, penelitian ini mengisolasi pengaruh arsitektur dari pengaruh penambahan data.

Keberhasilan solusi ini dinilai dari dua parameter yang saling melengkapi, yaitu akurasi deteksi dan efisiensi komputasi, dengan pemantauan jalan berbiaya rendah secara berkala—bukan deteksi waktu-nyata pada kendaraan bergerak—sebagai use case utama yang disasar sejak awal penelitian ini.

Berdasarkan pelatihan dengan tiga seed acak dan pelaporan rata-rata beserta simpangan baku, YOLO11n memperoleh performa deteksi yang lebih baik daripada YOLOv8n, dengan keunggulan yang paling menonjol pada rezim data terbatas. Pada data primer, YOLO11n mencapai mAP50 sebesar 0,354 ± 0,006 dibanding 0,314 ± 0,015, sekaligus lebih stabil antar-seed. Keunggulan ini menyempit ketika data pelatihan diperbanyak dengan data sekunder, karena YOLOv8n memperoleh peningkatan yang lebih besar dari tambahan data. Pola ini mengindikasikan bahwa YOLO11n lebih efisien-data, yaitu mencapai performa yang baik dengan data pelatihan yang lebih sedikit. Konfigurasi terbaik, yaitu YOLO11n dengan data gabungan, mencapai mAP50 sebesar 0,369 ± 0,017 dan mAP50-95 sebesar 0,151 ± 0,013.

Dari sisi efisiensi, YOLO11n lebih hemat parameter (−14,0%), FLOPs (−21,2%), dan ukuran model (−12,7%) dibanding YOLOv8n, meskipun waktu inferensinya sedikit lebih lama pada GPU (+2,7 ms, +30,0%). Kedua model beroperasi jauh di atas ambang waktu-nyata, sehingga keduanya berpotensi dipertimbangkan untuk pemantauan kondisi jalan berbiaya rendah, dengan pilihan yang bergantung pada prioritas antara efisiensi memori dan laju inferensi. Analisis kesalahan menunjukkan bahwa kegagalan model didominasi oleh objek yang tidak terdeteksi, bukan oleh kekeliruan antar-kelas, dengan pothole sebagai kelas paling menantang yang performanya tetap terbatas meskipun sumber sekunder menyumbang

mayoritas anotasi pothole, yang mengindikasikan kemungkinan kesenjangan domain antara data sekunder terkurasi dan data primer lapangan.

Evaluasi tambahan pada video (Bab 4.12) menunjukkan bahwa penghitungan objek unik melalui tracking masih jauh dari akurat: kelas pothole cenderung dihitung kurang, sedangkan kelas retak mengalami kelebihan hitung yang signifikan akibat fragmentasi track pada objek berbentuk tidak beraturan—keterbatasan yang tidak terselesaikan baik dengan ByteTrack maupun BoT-SORT. Karena mekanisme tracking ini identik pada kedua arsitektur, temuan ini tidak mengubah kesimpulan perbandingan YOLOv8n dan YOLO11n, tetapi menjadi catatan penting bagi penerapan pada sistem pemantauan berbasis video.

Temuan ini perlu ditafsirkan dengan hati-hati. Karena jumlah pengulangan seed terbatas pada tiga, perbedaan yang teramati dilaporkan sebagai rata-rata beserta simpangan baku dan sebaiknya dibaca sebagai kecenderungan yang konsisten antar-seed, bukan sebagai hasil uji signifikansi statistik berskala besar. Nilai mAP absolut mencerminkan pengaturan lapangan yang menantang dan realistis.

Lima keterbatasan utama perlu ditegaskan sebagai konteks pembacaan kesimpulan di atas, bukan sebagai kegagalan penelitian. Pertama, jumlah pengulangan seed yang terbatas pada tiga membuat uji signifikansi statistik bersifat indikatif, bukan konklusif. Kedua, seluruh pengukuran kecepatan GPU dilakukan pada satu perangkat (RTX 3050), sehingga angka kecepatan bersifat relatif terhadap perangkat tersebut, meskipun pengukuran CPU pada Bab 4 memperkuat bahwa keunggulan efisiensi parameter dan FLOPs bersifat tidak bergantung perangkat. Ketiga, anotasi retak lateral pada sumber sekunder dihapus namun citranya dipertahankan sebagai background, sehingga berpotensi memunculkan derau label yang berlaku setara pada seluruh konfigurasi. Keempat, kesenjangan domain pada kelas pothole—sumber sekunder menyumbang mayoritas anotasi pothole namun tidak memperbaiki performanya—menunjukkan bahwa penambahan data sekunder tidak serta-merta menutup kebutuhan data lapangan yang representatif. Kelima, evaluasi tracking objek pada video (Bab 4.12) bersifat awal dan terbatas pada tiga video dari satu kabupaten, sehingga belum mewakili kelima provinsi pada dataset primer.

Secara keseluruhan, temuan ini juga menggambarkan karakteristik deteksi kerusakan jalan pada konteks Indonesia: data lapangan primer yang representatif masih terbatas dan mahal diperoleh (767 citra dari lima provinsi), kondisi visual antarwilayah cukup beragam (sudut pengambilan, pencahayaan, tekstur perkerasan), dan pothole secara konsisten menjadi kelas paling menantang—sejalan dengan literatur internasional—sehingga penambahan data

sekunder saja tidak cukup untuk menutup kesenjangan domain tanpa disertai data lapangan yang representatif. Karakteristik ini menegaskan bahwa keberhasilan deteksi kerusakan jalan di Indonesia bergantung tidak hanya pada pemilihan arsitektur, tetapi juga pada kecukupan dan representativitas data lapangan yang menjadi dasar pelatihannya.

Sebagai penutup, meskipun akurasi absolut yang dicapai masih perlu ditingkatkan, efisiensi komputasi dan kestabilan performa yang ditunjukkan YOLO11n tetap mendukung potensinya sebagai alternatif pemantauan kerusakan jalan yang lebih murah dan dapat diskalakan dibandingkan inspeksi manual yang lambat, mahal, dan sulit diterapkan pada jaringan jalan yang luas sebagaimana diuraikan pada Bab 1, sejalan dengan use case pemantauan berkala yang disasar sejak awal penelitian ini.

5.2 Saran

Berdasarkan temuan dan keterbatasan di atas, beberapa saran diajukan untuk penelitian selanjutnya. Pertama, pelatihan sebaiknya diulang dengan jumlah seed yang lebih banyak, misalnya lima pengulangan, atau menggunakan validasi silang, disertai uji signifikansi statistik, sehingga perbedaan performa antararsitektur yang saat ini masih berupa kecenderungan dapat dinyatakan sebagai hasil yang teruji. Penambahan jumlah seed penting karena, dengan hanya tiga pengulangan, sebaran nilai belum cukup untuk uji statistik yang kuat; menambah pengulangan akan mempersempit selang kepercayaan dan memungkinkan kesimpulan yang lebih tegas mengenai selisih yang kecil, seperti yang teramati pada kondisi data gabungan.

Kedua, kualitas anotasi perlu diaudit dan diperbaiki, terutama pada kelas pothole yang memperoleh performa terendah, serta pada penanganan retak lateral yang saat ini menjadi background, karena konsistensi kotak pembatas berpotensi meningkatkan metrik lokalisasi tanpa mengubah arsitektur. Perbaikan anotasi merupakan langkah berbiaya relatif rendah namun berdampak, karena menyasar sumber galat pada data alih-alih menambah kompleksitas model. Ketiga, jumlah data primer lapangan sebaiknya diperbanyak untuk mengurangi overfitting yang teramati, mengingat performa absolut tampak dibatasi oleh ketersediaan data. Penambahan data primer juga berpotensi memperkecil kesenjangan domain pada kelas pothole, karena memberi model lebih banyak contoh yang benar-benar berasal dari distribusi lapangan yang menjadi sasaran.

Keempat, evaluasi lintas-wilayah dianjurkan, misalnya dengan menahan satu provinsi sebagai pengujian eksternal, untuk menguji generalisasi model pada distribusi jalan yang berbeda dari data pelatihan. Skema semacam ini akan menguji secara langsung apakah

model mempertahankan performanya ketika dihadapkan pada kondisi wilayah yang tidak pernah dilihat saat pelatihan, yang merupakan syarat penting bagi penerapan pada jaringan jalan nasional yang beragam. Kelima, penyempurnaan arsitektur ringan yang telah terbukti pada literatur deteksi retak, seperti mekanisme atensi atau modul konvolusi yang efisien, dapat dieksplorasi dengan syarat diterapkan secara identik pada kedua arsitektur agar keadilan perbandingan tetap terjaga. Syarat kesetaraan ini penting agar penelitian lanjutan tidak terjebak pada percampuran variabel yang justru menjadi salah satu keterbatasan literatur terdahulu.

Keenam, untuk mendukung klaim penerapan berbiaya rendah, pengujian langsung pada perangkat tepi yang sesungguhnya, disertai pengukuran konsumsi daya dan latensi ujung-ke-ujung, dapat memperkuat relevansi praktis penelitian. Benchmark pada mode CPU-only yang telah dilakukan pada penelitian ini merupakan langkah awal ke arah tersebut, namun pengujian pada perangkat tepi nyata seperti Jetson atau ponsel akan memberikan gambaran yang lebih representatif mengenai perilaku model di lapangan. Ketujuh, implementasi model terbaik ke dalam aplikasi pemantauan yang dapat digunakan petugas lapangan merupakan arah pengembangan lanjutan yang berada di luar cakupan penelitian ini.

Kedelapan, penelitian lanjutan dapat menguji generasi YOLO yang lebih mutakhir, khususnya YOLO26n, pada dataset jalan Indonesia yang sama. YOLO26 dirancang edge-first dan bebas-NMS dengan klaim inferensi CPU hingga 43% lebih cepat dibanding YOLO11 pada benchmark COCO milik Ultralytics (Ultralytics, 2026); pengujian langsung pada dataset dan protokol penelitian ini akan memverifikasi apakah keunggulan kecepatan tersebut turut terwujud pada domain kerusakan jalan Indonesia, sekaligus relevan langsung dengan fokus tesis pada kelayakan perangkat terbatas.

Kesembilan, mengingat performa tracking objek pada evaluasi awal ini masih terbatas—khususnya overcounting parah pada kelas retak akibat fragmentasi track pada objek berbentuk tidak beraturan, yang tidak terselesaikan baik dengan ByteTrack maupun BoT-SORT—penelitian lanjutan disarankan untuk menguji penalaan hyperparameter tracker (misalnya memperbesar track buffer), algoritma tracking lain di luar bawaan Ultralytics, atau pendekatan berbasis segmentasi maupun lokasi (GPS) yang berpotensi lebih sesuai untuk objek amorf seperti retak jalan, serta memperluas pengujian ke lebih banyak video dari seluruh provinsi pada dataset primer untuk memperkuat klaim generalisasi.

Daftar Pustaka

Adebiyi, R., et al. (2024). Performance analysis of various image feature extractor filters for pothole anomaly classification. International Journal of Image, Graphics and Signal Processing, 16, 25–37.

Aharon, N., Orfaig, R., & Bobrovsky, B.-Z. (2022). BoT-SORT: Robust associations multi-pedestrian tracking (arXiv:2206.14651). arXiv.

Ali, M. L., & Zhang, Z. (2024). The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. Computers, 13(12), 336.

Arya, D., et al. (2021). Deep learning-based road damage detection and classification for multiple countries. Automation in Construction, 132, 103935.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2024). RDD2022: A multi-national image dataset for automatic road damage detection. Geoscience Data Journal, 11(3).

Badan Pusat Statistik. (2022). Statistik transportasi darat 2021. Badan Pusat Statistik.

Basily, A. (2020). Road damage [Data set]. Kaggle. https://www.kaggle.com/datasets/alvarobasily/road-damage

Bhavan Kumar, S. B., Guhan, S., Kishore, M., Santhosh, R., & Alfred Daniel, J. (2023). Real-time pothole detection using YOLOv5 algorithm: A feasible approach for intelligent transportation systems. In Proc. 2023 Second Int. Conf. Electronics and Renewable Systems (ICEARS) (pp. 1678–1683).

Bouthillier, X., et al. (2021). Accounting for variance in machine learning benchmarks. Proceedings of Machine Learning and Systems, 3.

Denaro, L. G., & Lim, R. (2025). Analisis deteksi kerusakan pada jalan aspal menggunakan deep learning untuk mendukung efisiensi biaya dan waktu dalam pemantauan berkelanjutan. Jurnal Dimensi Insinyur Profesional.

Geng, H., Liu, Z., Jiang, J., Fan, Z., & Li, J. (2024). Embedded road crack detection algorithm based on improved YOLOv8. Journal of Computer Applications, 44(5), 1613–1618.

Ismail, D. R., & Rahmadewi, R. (2025). Sistem deteksi jalan berlubang secara real-time menggunakan YOLOv11: Integrasi data dan lokasi melalui website, studi kasus daerah Karawang. JATI (Jurnal Mahasiswa Teknik Informatika), 9(3), 3953–3961.

Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2024). YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions (arXiv:2411.00201). arXiv.

Jonathan, R., Sari, N. N. K., & Christian, E. (2026). Implementasi model deep learning untuk deteksi dan klasifikasi jenis kerusakan jalan di Kota Palangkaraya. Prosiding Seminar Nasional KONSTELASI, 3(1), 55–65.

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), 100804.

Khan, M., Raza, M. A., Abbas, G., Othmen, S., Yousef, A., & Jumani, T. A. (2024). Pothole detection for autonomous vehicles using deep learning: A robust and efficient solution. Frontiers in Built Environment, 9, 1323792.

Khanam, R., & Hussain, M. (2024). YOLOv11: An overview of the key architectural enhancements (arXiv:2410.17725). arXiv.

Klčo, M., Koniar, D., Hargas, L., & Paskala, M. (2023). Automated detection of potholes using YOLOv5 neural network. *Transportation Research Procedia*.

Liu, Y., Wang, F., Song, R., Wu, Y., Shen, H., Wu, H., & Zhang, C. (2026). EVA-YOLOv8: An improved YOLOv8 model integrating multi-scale attention mechanism and vision transformer for multi-class road crack detection. Scientific Reports.

Ma, L., & Chen, M. (2025). Road damage detection based on improved YOLO algorithm. Scientific Reports, 15, 28506.

Pavement Interactive. (n.d.). Pavement distresses. https://pavementinteractive.org/reference-desk/pavement-management/pavement-distresses/

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proc. IEEE CVPR (pp. 779–788).

Roy, A. M., & Bhaduri, J. (2023). DenseSPH-YOLOv5: An automated damage detection model based on DenseNet and Swin-transformer prediction head-enabled YOLOv5 with attention mechanism. Advanced Engineering Informatics, 56, 102007.

Safyari, Y., Mahdianpari, M., & Shiri, H. (2024). A review of vision-based pothole detection methods using computer vision and machine learning. Sensors, 24(17), 5652.

Sapkota, R., et al. (2025). YOLO advances to its genesis: A decadal and comprehensive review of the YOLO series. Artificial Intelligence Review, 58, 274.

Sasmito, B., Setiadji, B. H., & Isnanto, R. R. (2023). Deteksi kerusakan jalan menggunakan pengolahan citra deep learning di Kota Semarang. TEKNIK, 44(1), 7–14.

Takyudin, T., Rais, M. S., Putra, J. A., & Hamsar, A. (2025). Metode deteksi dan estimasi luas lubang jalan menggunakan deep learning berbasis YOLOv11. RIGGS: Journal of Artificial Intelligence and Digital Business, 4(2), 6786–6790.

Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. In Proc. IEEE/CVF CVPR (pp. 10781–10790).

Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-centric real-time object detectors (arXiv:2502.12524). arXiv.

Ultralytics. (2023). YOLOv8. https://docs.ultralytics.com/models/yolov8/

Ultralytics. (2024). YOLO11. https://docs.ultralytics.com/models/yolo11/

Ultralytics. (2026). *YOLO26*. https://docs.ultralytics.com/

Wang, J., et al. (2024b). Road defect detection based on improved YOLOv8s model. Scientific Reports, 14, 16758.

Wang, S., Jiao, H., Su, X., & Yuan, Q. (2024a). An ensemble learning approach with attention mechanism for detecting pavement distress and disaster-induced road damage. IEEE Transactions on Intelligent Transportation Systems, 25(10), 13667–13681.

Xiong, C., Zayed, T., & Abdelkader, E. M. (2024). A novel YOLOv8-GAM-Wise-IoU model for automated detection of bridge surface cracks. Construction and Building Materials, 414, 135025.

Xu, W., Xu, J., Ji, Y., Li, G., Li, H., & Zang, Z. (2025). Research on enhancing road apparent crack detection based on the improved YOLOv8n model. PLOS ONE.

Yu, J., Yin, Y., Cai, H., Chen, J., Liu, Y., & Dong, Z. (2024a). Road surface defect detection—From image-based to non-image-based: A survey. IEEE Transactions on Intelligent Transportation Systems, 25(9).

Yu, J., Yin, Y., Cai, H., Chen, J., Liu, Y., & Dong, Z. (2024b). Research on road crack detection algorithm based on improved YOLOv8. In Proc. 2024 2nd Int. Conf. Frontiers of Intelligent Manufacturing and Automation (CFIMA).

Zeng, J., & Zhong, H. (2024). YOLOv8-PD: An improved road damage detection algorithm based on YOLOv8n model. Scientific Reports, 14, 12052.

Zhang, A. A., et al. (2024). Intelligent pavement condition survey: Overview of current researches and practices. Journal of Road Engineering, 4(3).

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-object tracking by associating every detection box. In Proc. European Conference on Computer Vision (ECCV).