

IDENTIFIKASI TINGKAT KERUSAKAN JALAN BERLUBANG MENGUNAKAN PENGOLAHAN CITRA DIGITAL

Rizki Irama¹, Leonardus Sandy Ade Putra², Redi Ratiandi Yacoub³

d1021201020@student.untan.ac.id¹, leonardusandy@ee.untan.ac.id², rediyacoub@ee.untan.ac.id³

Universitas Tanjungpura

ABSTRAK

Jalan raya merupakan infrastruktur vital yang menunjang kegiatan ekonomi dan mobilitas masyarakat. Namun, keberadaan jalan berlubang menjadi masalah serius yang mengancam keselamatan pengendara dan memerlukan biaya perawatan yang tinggi. Proses identifikasi kerusakan jalan secara manual sering kali tidak efisien, memakan waktu, dan membutuhkan tenaga yang besar. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sebuah sistem yang mampu mengidentifikasi serta mengklasifikasikan tingkat kerusakan jalan berlubang secara otomatis menggunakan metode *deep learning*. Penelitian ini mengimplementasikan algoritma *You Only Look Once version 8* (YOLOv8) untuk mendeteksi dan mengelompokkan kerusakan ke dalam tiga kategori: rusak ringan, rusak sedang, dan rusak tinggi. Proses pelatihan model menggunakan total 200 citra data sekunder dan diuji menggunakan 99 citra data primer yang diambil langsung dari lapangan. Hasil penelitian menunjukkan bahwa model yang dikembangkan memiliki kinerja yang sangat baik, dengan pencapaian metrik *mean Average Precision* (mAP) pada ambang batas IoU 0.5 sebesar 90,6%. Pada tahap pengujian dengan data baru, sistem berhasil mencapai tingkat akurasi rata-rata sebesar 96% di semua kelas kerusakan. Hasil ini membuktikan bahwa metode YOLOv8 sangat efektif dan akurat untuk otomatisasi survei kondisi jalan, sehingga dapat menjadi solusi yang efisien untuk mendukung pemeliharaan infrastruktur jalan yang tepat waktu.

Kata Kunci: Jalan Berlubang, Pengolahan Citra, Klasifikasi Kerusakan, YOLOv8, *Deep Learning*.

ABSTRACT

Highways are vital infrastructures that support economic activities and public mobility. However, the presence of potholes poses a serious problem that threatens driver safety and requires high maintenance costs. Manual identification of road damage is often inefficient, time-consuming, and labor-intensive. Therefore, this research aims to design and build a system capable of automatically identifying and classifying the severity of pothole damage using a deep learning method. This study implements the *You Only Look Once version 8* (YOLOv8) algorithm to detect and categorize damage into three classes: light, medium, and high. The model training process utilized a total of 200 secondary data images and was tested using 99 primary data images captured directly from the field. The results show that the developed model performed excellently, achieving a mean Average Precision (mAP) metric at an IoU threshold of 0.5 of 90.6%. In the testing phase with new data, the system achieved an average accuracy rate of 96% across all damage classes. These results demonstrate that the YOLOv8 method is highly effective and accurate for automating road condition surveys, thus offering an efficient solution to support timely road infrastructure maintenance.

Keywords: Potholes, Image Processing, Damage Classification, YOLOv8, *Deep Learning*.

PENDAHULUAN

Jalan raya merupakan jalur darat bagi masyarakat untuk mempermudah akses transportasi ke tempat tujuan. Jalan raya sangat penting dalam memperlancar kegiatan hubungan perekonomian, baik antara satu kota dengan kota lainnya, maupun antara kota dengan desa serta antara satu desa dengan desa lainnya [1]. Keberadaan jalan berlubang banyak menyebabkan kecelakaan dan kerusakan pada kendaraan. Dengan jumlah

kendaraan yang terus meningkat, perbaikan dan pemeliharaan jalan menjadi semakin mendesak. Oleh karena itu, diperlukan sistem yang efisien untuk mengidentifikasi seberapa parah kerusakan pada jalan raya untuk mendukung upaya perbaikan yang tepat waktu.

Pengolahan citra digital merupakan metode yang digunakan beberapa orang dalam mendeteksi lubang menggunakan warna sebagai fitur utama yang di ekstrak [2]. Dalam konteks deteksi jalan berlubang, teknologi ini dapat digunakan untuk mengidentifikasi kerusakan pada permukaan jalan melalui gambar yang diambil oleh kamera. Dengan memanfaatkan algoritma pengolahan citra, sistem dapat secara otomatis mendeteksi dan mengklasifikasi kerusakan jalan [3].

Sistem pendeteksi jalan berlubang berbasis pengolahan citra digital juga dapat berkontribusi pada manajemen aset infrastruktur yang lebih baik. Data tersebut bisa memberikan informasi yang berharga bagi pemerintah untuk menangani masalah kerusakan jalan dalam merencanakan dan mengalokasikan anggaran perbaikan jalan secara efektif [4]. Selain itu, para pekerja lapangan tidak perlu mengukur satu-satu luas dan jumlah lubang pada jalan raya.

Dalam pengembangannya, pengolahan citra dalam membangun sistem klasifikasi jalan dan sistem peringatan dini untuk pendukung kendaraan sesuai dengan tingkat kerusakan jalan tersebut dianggap sebagai tugas yang sulit, sehingga menggunakan metode deep learning adalah teknologi yang cukup memadai untuk membuat sistem yang dapat mengidentifikasi properti dengan metode ekstraksi fitur dari data pelatihan tanpa label, terutama dalam jumlah data yang besar [5].

Maka pada penelitian ini bertujuan mengklasifikasi dan mengidentifikasi kerusakan jalan pada jalan raya. Dalam mengetahui keadaan jalan berlubang, penelitian ini secara langsung mendukung para pekerja untuk merencanakan perbaikan jalan. Yang di mana eksistensi teknologi menggunakan pengolahan citra sedang berkembang untuk mewujudkan inovasi yang memungkinkan dalam mengidentifikasi dan mengklasifikasi berbagai objek.

Selain itu penulis memantau perkembangan pengolahan citra digital menjadi sangat menarik dalam beberapa waktu ini. Penulis menemukan banyak sekali penerapan pengolahan citra digital dalam kehidupan sehari-hari. Demikian juga untuk pemantauan lingkungan yang efektif dan efisien, serta memperluas pemahaman tentang teknologi pengolahan citra digital dalam konteks mendeteksi jumlah dan luas lubang pada jalan.

METODE PENELITIAN

Bahan Penelitian

Dalam pengerjaan penelitian ini, penulis mengambil bahan penelitian berupa literatur yang membahas tentang penggunaan citra digital seperti yang dibahas penelitian yang terdahulu dalam tinjauan pustaka pada kerusakan jalan raya. Adapun tambahan penulis telah mengambil dari sumber-sumber terpercaya seperti jurnal terdahulu dalam daftar pustaka.

HASIL DAN PEMBAHASAN

Perancangan Data

Proses pengidentifikasian dan pengklasifikasian objek menggunakan citra digital memerlukan sejumlah besar data untuk dilatih, sehingga mendapatkan bobot optimal dalam pengidentifikasian dan pengklasifikasian objek. Selain itu, diperlukan juga sejumlah data lain untuk proses pengujian untuk menguji keakuratan dari metode. Pada penelitian

ini terdiri dari tiga kelas kerusakan jalan berlubang yaitu jalan rusak ringan, rusak sedang dan rusak tinggi.

Kebutuhan Perangkat Lunak

Hal yang diperlukan dalam mengimplementasikan metode yang digunakan dalam penelitian ini yaitu perangkat keras dan perangkat lunak. Rincian tersebut disajikan dalam Tabel 1.

Tabel 1. Kebutuhan Perancangan

Perangkat Keras (<i>Hardware</i>)	1. Intel Core i3-7020U dual-core 2,3GHz 2. 4GB DDR4-2400MHz
Perangkat Lunak (<i>Software</i>)	1. Microsoft Windows 10 Home 64 bit 2. Web Roboflow 3. Web Google colab

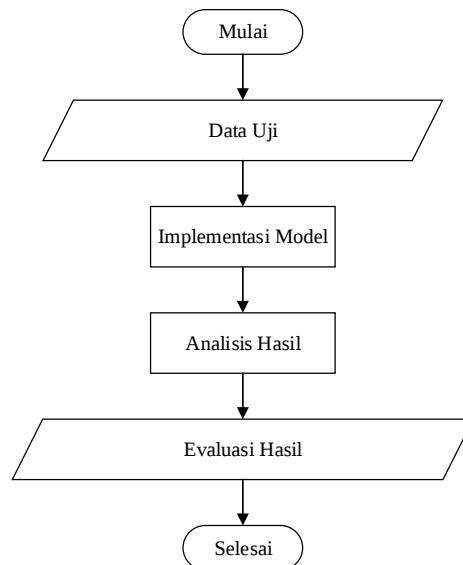


Diagram 1. Alur Pengujian

Data Awal

Proses yang akan dibangun adalah proses identifikasi dan klasifikasi kerusakan jalan berlubang menggunakan pengolahan citra digital. Selain itu, diperlukan pula sejumlah data lain untuk proses pengujian untuk menguji keakuratan dari metode. Data yang telah terkumpul dalam penelitian ini sejumlah 200 data citra untuk pelatihan dan 99 data citra untuk pengujian yang terdiri dari tiga kelas kerusakan jalan berlubang yaitu rusak ringan, rusak sedang dan rusak tinggi. Proses identifikasi dan klasifikasi pada penelitian ini mampu mengenali beberapa objek kerusakan dalam suatu citra dan menunjukkan pada sebuah kotak pembatas (*bounding box*).



(a) Rusak ringan



(b) Rusak sedang



(c) Rusak tinggi

Gambar 1. Citra bentuk awal

Pada Gambar 1, merupakan citra bentuk awal sebelum melakukan proses pelatihan dan proses pengujian. Data citra awal adalah citra RGB dengan format .jpg, .jpeg dan .png. Sebelum melakukan proses pelatihan dan pengujian, dilakukan tahap anotasi data, *resizing* dan *augmentasi*.

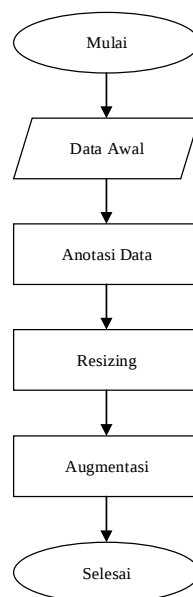


Diagram 2. Alur Sebelum Pelatihan Model

Anotasi Data

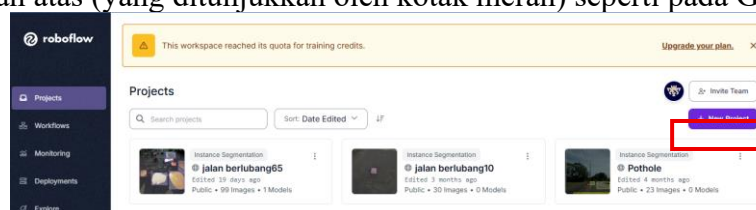
Proses ini melibatkan *roboflow* dalam pembuatan *bounding box* (kotak pembatas) di sekitar objek jalan berlubang dan memberinya label. Tahapan dalam proses anotasi data.

1. Membuka web Roboflow yang sudah tertaut dengan akun google kita.



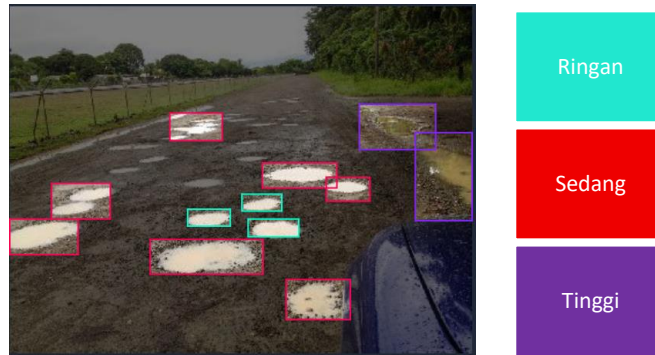
Gambar 2. Roboflow

2. Klik pada bagian project dan tambahkan project baru dengan klik new project pada bagian kanan atas (yang ditunjukkan oleh kotak merah) seperti pada Gambar 3.



Gambar 3. Membuat project baru

- Setelah itu unggah seluruh data latih untuk melakukan pelabelan dan anotasi secara manual seperti pada gambar 4.

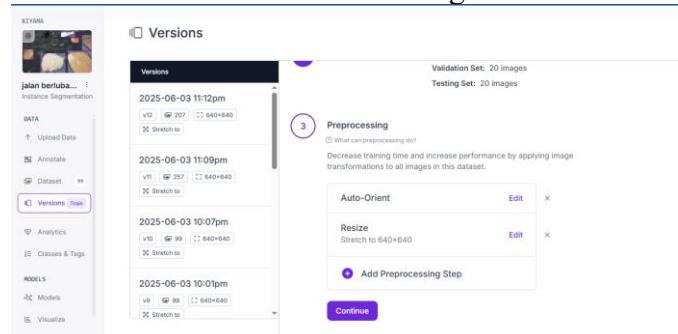


Gambar 4. Hasil pelabelan dan anotasi

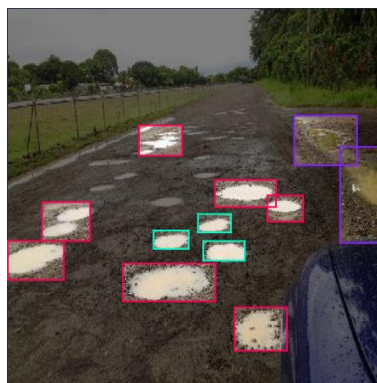
Resizing

Pada tahap *resizing* kita akan mengubah ukuran data menjadi 640×640 piksel. Proses *resizing* pada roboflow yaitu setelah kita melakukan anotasi data, proses ini secara otomatis mengubah semua data yang telah kita unggah sebelumnya. Tujuan *resizing* ini adalah untuk mempermudah program untuk melatih data kita dengan ukuran yang sama. Tahapan pada *resizing* sebagai berikut:

- Pertama kita masuk ke menu version untuk mengubah ukuran 640×640 piksel.



Gambar 4. Proses *resizing*



Gambar 5. Hasil citra *resizing* pada anotasi

Augmentasi

Pada proses *augmentasi* yaitu kita menambahkan beberapa transformasi data berupa *flip horizontal*, *brightness*, *blur* dan *mosaic*. Tujuan dari *augmentasi* adalah untuk meningkatkan jumlah dan keragaman data latih serta meningkatkan performa akurasi model. Bisa dilihat hasil dari *augmentasi* tersebut pada Gambar 4.6, 4.7, 4.8 dan 4.9.



Gambar 6. Hasil *augmentasi* dari flip: *horizontal*



Gambar 7.. Hasil *augmentasi* dari *brightness* terang



Gambar 8. Hasil *augmentasi* dari *blur*



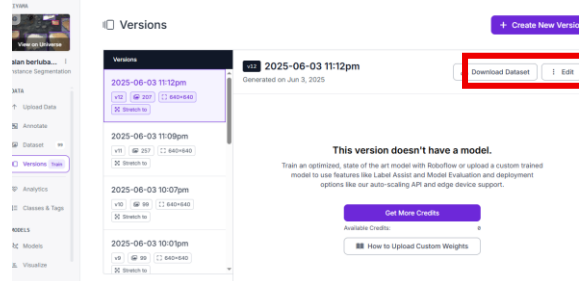
Gambar 9. Hasil *augmentasi* dari *mosaic*

Pelatihan Model

Pada subbab ini dilakukan pelatihan model identifikasi kerusakan jalan berlubang dengan target keluaran berupa kelas lubang yaitu, kelas ringan, sedang dan tinggi. Masing-

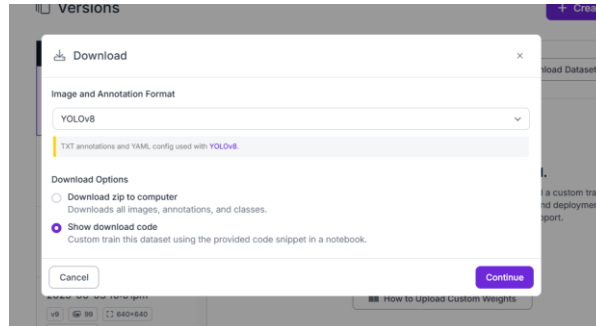
masing hasil metode penelitian yang dilakukan mulai dari penyiapan data penelitian sampai evaluasi data akan diuraikan secara rinci pada subbab ini. Pada pelatihan model jalan berlubang akan dilakukan pengambilan model dengan tingkat akurasi tertinggi.

1. Pertama-tama buka roboflow pada bagian version, selanjutnya download dataset seperti pada Gambar 11.



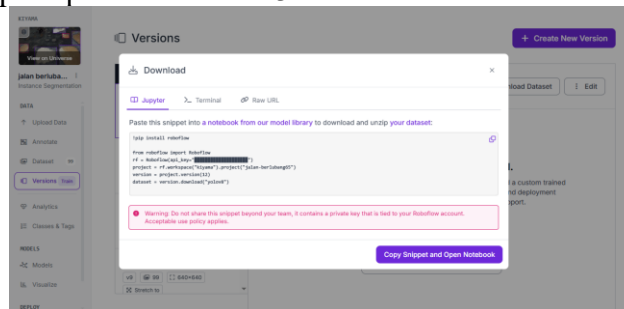
Gambar 10. Data untuk panggil ke google colab

2. Setelah itu pilih bagian *show download code* untuk menampilkan program dataset anotasi seperti pada Gambar 12.



Gambar 11. Memilih *Show Download Code*

3. Salin program pada data yang telah kita anotasi sebelumnya untuk memanggil data ke google colab seperti pada Gambar 13.



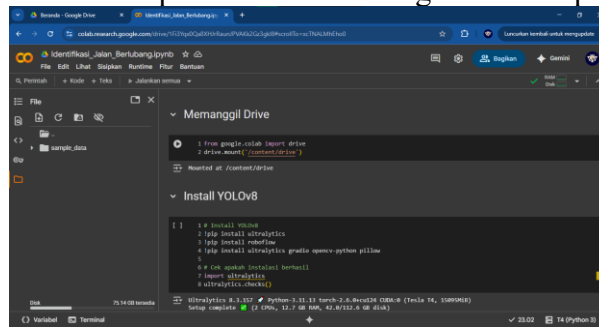
Gambar 12. Tampilan program dataset anotasi

4. Selanjutnya membuka web Google Colab dengan cara membuka google drive dan telusuri kata kunci tersebut



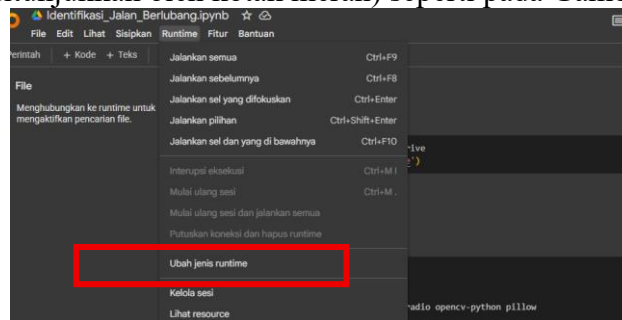
Gambar 13. Google Colab

5. Setelah itu, akan muncul tampilan awal web Google Colab seperti pada Gambar 15.



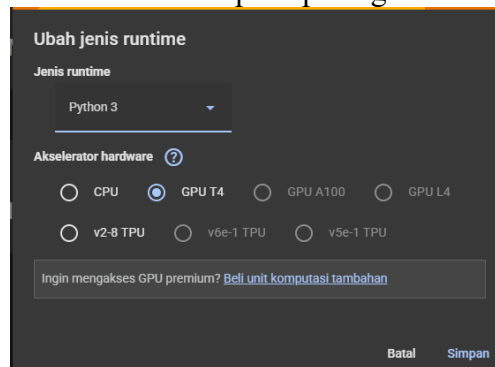
Gambar 14. Tampilan awal Google Colab

6. Setelah tampilan utama dari Google Colab muncul, langkah selanjutnya untuk menjalankan program identifikasi kerusakan jalan berlubang dengan mengatur jenis runtime (yang ditunjukkan oleh kotak merah) seperti pada Gambar 16.



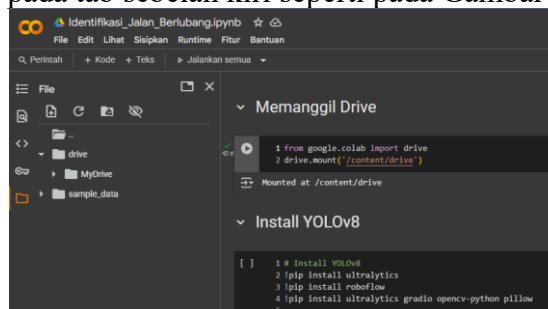
Gambar 15. Membuka ubah jenis runtime

7. Setelah membuka runtime, langkah selanjutnya klik ubah jenis runtime dan akan muncul tampilan seperti pada Gambar 12. Kemudian ganti jenis runtime ke Python 3 dan Akselerator hardware ke GPU T4 seperti pada gambar 17.



Gambar 16. Mengganti jenis runtime

8. Kemudian setelah jenis runtime diubah, selanjutnya menjalankan program untuk memanggil drive supaya mempermudah dalam pengunggahan, maka akan ada tampilan MyDrive pada tab sebelah kiri seperti pada Gambar 4.18.



Gambar 17. Menampilkan Drive

Pemodelan Menggunakan Yolo

Pemodelan kerusakan jalan berlubang dengan target keluaran kelas lubang menggunakan *Yolo v8*. Proses pelatihan model ini menggunakan model YOLOv8n (varian 'nano' dari *Yolo v8* yang ringan dan cepat) pada dataset yang telah disiapkan.

Menginstall *library* Python *YOLO v8*

Tahapan awal dalam pemodelan pada penelitian ini adalah mempersiapkan lingkungan pengembang (*development environment*). Proses ini mencakup instalasi serangkaian *library* Python yang menjadi fondasi untuk implementasi dan evaluasi model. List program berikut merincikan perintah-perintah yang dijalankan untuk mengonfigurasi lingkungan tersebut.

Kode Program 1. Install *Library* Python *YOLO v8*

```
!pip install ultralytics
!pip install roboflow
!pip install ultralytics gradio opencv-python pillow

import ultralytics ultralytics.checks()
!pip uninstall -y gradio
!pip install gradio==3.50.2
```

Setelah proses instalasi selesai dieksekusi, pada program tersebut memiliki fungsi yang bertujuan untuk melakukan analisis otomatis pada lingkungan yang telah terpasang, memastikan bahwa seluruh dependensi seperti *framework* PyTorch dan ketersediaan GPU telah terkonfigurasi dengan benar dan siap untuk digunakan dalam proses pelatihan dan inferensi model. Secara keseluruhan, program tersebut bertujuan untuk membangun lingkungan komputasi yang valid dan fungsional, yang menjadi syarat sebelum melangkah ke tahap pengembangan model deteksi objek.

Memanggil Data Latih Dari Roboflow

Setelah lingkungan pengembang berhasil dikonfigurasi, tahap selanjutnya dalam penelitian ini adalah akuisisi dataset. Proses pengumpulan data dilakukan secara terprogram dengan mengunduh dataset yang telah disiapkan dan di anotasi sebelumnya pada platform dataset Roboflow. List program berikut merincikan kode yang digunakan untuk mengakses dan mengunduh dataset tersebut.

Kode Program 2. Memanggil data latih anotasi dari Roboflow

```
from roboflow import Roboflow
rf = Roboflow(api_key="2mQkOI0tcjA3yKII9en0")
project = rf.workspace("kiyama").project("jalan-berlubang65")
version = project.version(12)
dataset = version.download("yolov8")
```

Dengan dieksekusinya program tersebut, dataset yang relevan untuk penelitian ini secara otomatis terunduh ke dalam google colab. Ketersediaan dataset dalam format yang tepat memungkinkan transisi yang efisien untuk langsung menuju tahap pelatihan model.

Pengaturan Parameter Pelatihan Model

Pemilihan pelatihan model sangat mempengaruhi hasil pelatihan. Parameter yang digunakan dalam penelitian ini merupakan nilai yang telah direkomendasikan oleh *Yolo v8* dan sedikit perubahan nilai. Program ini berfungsi untuk membuat model pelatihan.

Kode Program 3. Pelatihan Model

```
from ultralytics import YOLO
model = YOLO("yolov8n.pt")
results = model.train(data="/content/jalan-berlubang65-12/data.yaml",
epochs=100,
```

```
imgsz=640,  
batch=8,)
```

Penjelasan parameter program yang telah dilampirkan di atas.

Tabel 2. Parameter yang digunakan

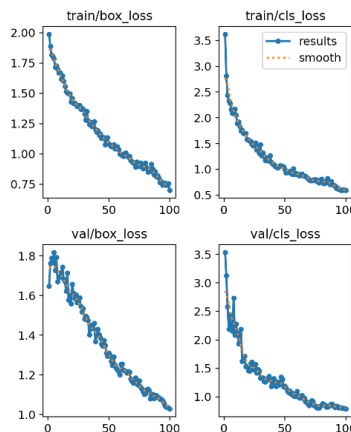
Parameter	Nilai	Deskripsi
Model	yolov8n.pt	Menggunakan model nano sebagai bobot awal (<i>pre-trained on COCO</i>).
Ukuran Citra	640x640	Semua citra diubah ukurannya menjadi 640x640 piksel sebelum masuk ke jaringan.
Epochs	100	Jumlah siklus pelatihan penuh pada seluruh dataset latih.
Batch Size	8	Jumlah citra yang diproses bersamaan dalam satu iterasi.

Analisis Kurva Pelatihan

Kurva pelatihan menunjukkan performa selama proses pelatihan sebuah model objek deteksi selama 100 epoch. Grafik ini menunjukkan bahwa proses pelatihan berjalan dengan sangat baik.

Loss Curves (Kurva Loss)

Kurva *loss* mengukur seberapa besar kesalahan model. Penurunan kurva *loss* menandakan model sedang belajar. Pada Gambar 4.6 menunjukkan kurva *loss* untuk *bounding box* (*box_loss*) dan klasifikasi (*cls_loss*) pada data latih dan validasi.



Gambar 18.. Kurva *loss* selama pelatihan

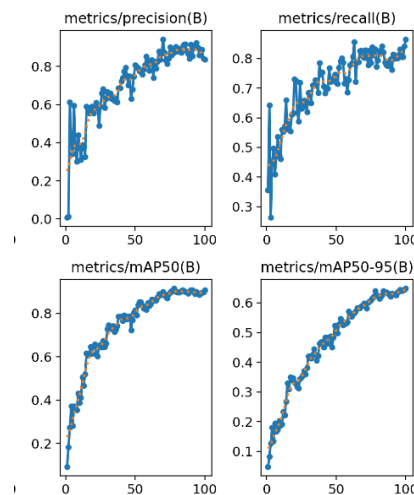
Terlihat pada Gambar 6. bahwa kurva *loss* untuk data latih (*train*) maupun data validasi (*val*) menunjukkan penurunan secara stabil. Hal ini menandakan bahwa model berhasil belajar dan terus mengurangi kesalahan dari waktu ke waktu. Terlihat kurva validasi (*val*) mengikuti pola kurva pelatihan (*train*) menandakan bahwa model menyamakan dengan baik dan tidak mengalami *overfitting* yang berlebihan.

Kurva Metrik Performa

Untuk menilai efektivitas dan kemampuan model yang telah dilatih, dilakukan evaluasi kuantitatif terhadap performa objek deteksi. Evaluasi ini menggunakan metrik

standar dalam bidang objek deteksi, yaitu *Precision*, *Recall* dan *mean Average Precision* (*mAP*). *Precision* mengukur akurasi dari deteksi positif yang dihasilkan oleh model, sedangkan *Recall* menilai kemampuan model dalam mengidentifikasi semua objek relevan (*ground truth*) yang terdapat dalam data.

Sebagai metrik utama, *mAP* memberikan gambaran menyeluruh tentang performa model dengan mempertimbangkan kurva *Precision-Recall*. Evaluasi dilakukan pada data tingkat: *mAP50* (dengan ambang batas *Intersection over Union*(IoU) 50%) yang menilai kualitas deteksi secara umum, dan *mAP50-95* (rata-rata *mAP* pada IoU dari 50% hingga 90%) yang mengukur kemampuan model dalam menghasilkan *bounding box* yang sangat akurat. Perkembangan keempat metrik ini selama 100 epoch pelatihan ditampilkan pada Gambar 7.



Gambar 19. Kurva Metrik Performa selama pelatihan

Berdasarkan hasil visualisasi yang terdapat pada Gambar 4.7, analisis mendalam dapat dilakukan terhadap proses pelatihan model. Secara keseluruhan, keempat grafik metrik menunjukkan tren peningkatan yang positif dan signifikan dari awal hingga akhir pelatihan, yang secara tetap membuktikan bahwa model berhasil belajar serta meningkatkan kinerjanya seiring waktu.

Berikut adalah analisis secara rinci:

1. *Precision* dan *Recall* : Kurva *Precision* (*metrics/precision(B)*) dan *Recall* (*metrics/recall(B)*) menunjukkan peningkatan yang konsisten. Pada akhir pelatihan (epoch 100), model mencapai nilai *Precision* sekitar 0.83 dan *Recall* sekitar 0.86. Hal ini menunjukkan bahwa model yang telah dilatih memiliki keseimbangan yang baik antara kemampuan untuk menghasilkan deteksi yang akurat (meminimalkan *false positive*) dan kemampuan untuk menemukan sebagian besar objek target (meminimalkan *false negative*).
2. *mAP50*: Kurva *metrics/mAP50(B)* memperlihatkan kenaikan yang stabil dan mencapai nilai puncak mendekati 0.90. Nilai tersebut menegaskan bahwa model memiliki performa yang sangat baik dalam mendeteksi keberadaan objek lubang dengan tumpang tindih (*overlap*) minimal 50% dengan *ground truth*. Ini merupakan indikator yang kuat bahwa model bisa untuk mendeteksi keberadaan suatu objek lubang.
3. *mAP50-90*: Kurva *metrics/mAP50-90(B)* merepresentasikan evaluasi yang paling ketat. Walaupun nilainya lebih rendah dibandingkan dengan *mAP50* dan mencapai puncak sekitar 0.65, tren peningkatannya yang stabil tetap menunjukkan kemampuan model yang baik. Nilai ini mengartikan bahwa model tidak hanya mampu mendeteksi

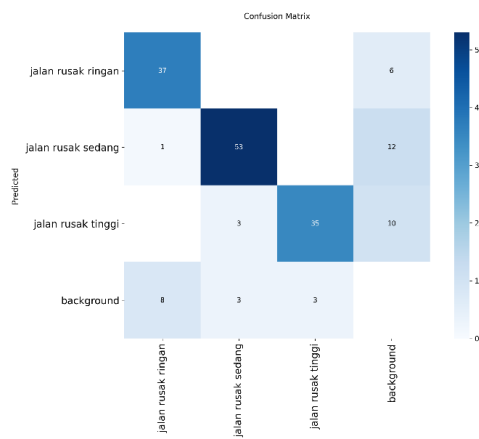
objek, tetapi juga mampu menghasilkan lokasi *bounding box* dengan tingkat presisi yang tinggi pada berbagai ambang batas IoU.

Fenomena di mana kurva-kurva mulai menurun (mendekati konvergensi) pada epoch 80 hingga 100 menunjukkan bahwa model telah mencapai titik jenuh dalam pelatihan, di mana penambahan epoch lebih lanjut tidak memberikan peningkatan performa yang signifikan. Secara keseluruhan, hasil evaluasi ini mengonfirmasi bahwa proses pelatihan telah berhasil dan model yang dihasilkan memiliki kemampuan deteksi objek yang kuat dan dapat diandalkan.

Matriks Kebingungan (*Confusion Matrix*)

Selain melakukan evaluasi dengan metrik performa seperti mAP, analisis yang lebih mendalam diperlukan untuk memahami perilaku model dalam mengenali tiga kelas kerusakan jalan yaitu rusak ringan, rusak sedang dan rusak tinggi setiap kategori secara spesifik, digunakan *confusion matrix* atau matriks kebingungan sebagai alat evaluasi. Matriks ini tidak hanya menunjukkan seberapa baik model dalam mengklasifikasikan setiap kategori kerusakan, tetapi juga penting untuk memahami dua jenis kesalahan utama yaitu kegagalan dalam mendeteksi kerusakan yang ada (*False Negative*) dan kesalahan deteksi pada area yang tidak rusak (*False Positive*).

Melalui matriks ini, dapat diidentifikasi secara rinci jenis-jenis kesalahan yang dilakukan model, seperti kelas mana yang sering tertukar atau salah diklasifikasikan. Analisis ini sangat penting untuk mengetahui kekuatan dan kelemahan model pada tingkat individual. Hasil evaluasi model pada data uji menggunakan *confusion matrix* bisa dilihat pada Gambar 8.



Gambar 20. *Confusion Matrix*

Berdasarkan Gambar 8, *confusion matrix* menggambarkan performa klasifikasi model terhadap tiga kelas yaitu jalan rusak ringan, jalan rusak sedang dan jalan rusak tinggi. Pada kelas background tidak dianggap sebagai kelas target, melainkan sebagai representasi area di mana seharusnya tidak ada kerusakan yang terdeteksi. Oleh karena itu, interaksi antara kelas target dan background dalam matriks ini akan menyoroti kelemahan model dalam membedakan objek dari lingkungannya. Sumbu vertikal (*Predict*) menunjukkan hasil prediksi model, sedangkan sumbu horizontal (*True*) menunjukkan label asli. Dari matriks tersebut, ada beberapa kesimpulan yang dijelaskan sebagai berikut:

1. Performa tinggi pada klasifikasi antar kelas kerusakan: diagonal utama matriks menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengklasifikasikan berbagai tingkat kerusakan. Model berhasil mengidentifikasi 37 data jalan rusak ringan, 53 data jalan rusak sedang dan 35 data jalan rusak tinggi dengan akurat. Tingkat kebingungan antar kelas kerusakan juga sangat rendah, contohnya hanya terdapat 1 kasus jalan rusak ringan yang salah diprediksi sebagai

jalan rusak sedang dan hanya 3 kasus rusak sedang yang salah diprediksi sebagai rusak tinggi. Hal ini membuktikan bahwa model mampu membedakan tingkat keparahan kerusakan dengan efektif.

2. Analisis kesalahan terkait background: Peran background dalam matriks ini adalah untuk mengukur dua jenis kesalahan:

- *False Negative*: Pada baris *Predicted*, background menunjukkan jumlah objek kerusakan yang tidak terdeteksi oleh model. Terdapat 8 kasus jalan rusak ringan, 3 kasus jalan rusak sedang dan 3 kasus jalan rusak tinggi yang salah diklasifikasikan sebagai background. Ini berarti model melewatkan total 14 objek kerusakan yang seharusnya terdeteksi.
- *False Positive*: Pada kolom *True*, menunjukkan jumlah deteksi yang salah, di mana model mengidentifikasi kerusakan pada area yang sebenarnya tidak rusak. Ini merupakan kesalahan terbesar bagi model, dengan total 18 kekeliruan yaitu 6 kasus sebagai rusak ringan, 12 sebagai rusak sedang dan 10 sebagai rusak tinggi. Kesalahan ini menunjukkan bahwa model cenderung menghasilkan *false positive* pada background.

Hal ini menunjukkan bahwa model yang dikembangkan memiliki kemampuan utama dalam membedakan tingkat keparahan kerusakan jalan. Namun, masalah utamanya terletak pada kemampuannya untuk secara akurat memisahkan antara area yang benar-benar rusak dan background, yang menjadi penyebab utama *False Negative* dan *False Positive*.

Evaluasi Kinerja Metrik Pelatihan Model

Proses evaluasi kinerja model dilakukan dengan menghitung nilai *Precision*, *Recall*, dan *F1-score* untuk masing-masing kelas yang digunakan dalam penelitian, yaitu kelas jalan rusak ringan, jalan rusak sedang dan jalan rusak tinggi. Perhitungan dilakukan dengan menggunakan data hasil *confusion matrix*, di mana setiap metrik dihitung berdasarkan jumlah *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) pada masing-masing kelas. Selanjutnya, untuk memperoleh gambaran performa model secara umum, dilakukan perhitungan nilai rata-rata keseluruhan (*macro average*) dari ketiga metrik tersebut, sehingga dapat memberikan evaluasi menyeluruh terhadap kemampuan model dalam melakukan identifikasi dan klasifikasi objek pada semua kelas. Berikut adalah rumus dari *Precision*, *Recall*, dan *F1-score*:

a. *Precision*

Precision mengukur seberapa tepat model dalam mendeteksi objek yang benar dari semua deteksi yang dibuat oleh model.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

b. *Recall*

Recall mengukur seberapa baik model dalam menemukan semua objek yang seharusnya terdeteksi.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

c. *F1-score*

F1-score merupakan rata-rata keseimbangan antara *precision* dan *recall*

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

Berikut adalah hasil dari perhitungan *Precision*, *Recall*, dan *F1-score*, pada masing-masing kelas:

a. Jalan rusak ringan

Pada jalan rusak ringan TP = 37, FP = 6 dan FN = 9

- *Precision*

$$Precision = \frac{TP}{TP+FP} = \frac{37}{37+6} = \frac{37}{43} \approx 0.860 \quad (4)$$

- *Recall*

$$Recall = \frac{TP}{TP+FN} = \frac{37}{37+9} = \frac{37}{46} \approx 0.804 \quad (5)$$

- *F1-score*

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{0.860 \times 0.804}{0.860 + 0.804} \approx 0.831 \quad (6)$$

b. Jalan rusak sedang

Pada jalan rusak sedang TP = 53, FP = 13 dan FN = 6

- *Precision*

$$Precision = \frac{TP}{TP+FP} = \frac{53}{53+13} = \frac{53}{66} \approx 0.803 \quad (7)$$

- *Recall*

$$Recall = \frac{TP}{TP+FN} = \frac{53}{53+6} = \frac{53}{59} \approx 0.898 \quad (8)$$

- *F1-score*

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{0.803 \times 0.898}{0.803 + 0.898} \approx 0.848 \quad (9)$$

c. Jalan rusak tinggi

Pada jalan rusak tinggi TP = 35, FP = 13 dan FN = 3

- *Precision*

$$Precision = \frac{TP}{TP+FP} = \frac{35}{35+13} = \frac{35}{48} \approx 0.729 \quad (10)$$

- *Recall*

$$Recall = \frac{TP}{TP+FN} = \frac{35}{35+3} = \frac{35}{38} \approx 0.921 \quad (11)$$

- *F1-score*

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{0.729 \times 0.921}{0.729 + 0.921} \approx 0.814 \quad (12)$$

Berikut adalah hasil dari perhitungan menggunakan data *confusion matrix*.

Tabel 3. Metrik hasil pelatihan model

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Ringan	0.860	0.804	0.831
Sedang	0.803	0.898	0.848
Tinggi	0.729	0.921	0.814
Rata-rata (all)	0.797	0.871	0.831

Berdasarkan Tabel 3, model menunjukkan kinerja terbaik pada kelas kerusakan ringan dari sisi *Precision* (0,860), sedangkan *Recall* tertinggi dicapai oleh kelas kerusakan tinggi (0,921). Hal ini menandakan bahwa prediksi kerusakan ringan cenderung lebih tepat, namun kerusakan berat lebih mudah terdeteksi secara menyeluruh oleh model. Nilai *F1-Score* tertinggi terdapat pada kelas kerusakan sedang (0,848), yang menunjukkan keseimbangan optimal antara ketepatan dan kelengkapan deteksi. Rata-rata nilai untuk semua kelas, tanpa menghitung kelas latar belakang, adalah 0,797 untuk *Precision*, 0,871 untuk *Recall*, dan 0,831 untuk *F1-Score*. Dengan demikian, model dapat dikategorikan memiliki performa deteksi yang baik dan seimbang di ketiga kategori kerusakan jalan.

Hasil Pengujian

Dari hasil pemodelan yang telah dilatih sebelumnya menghasilkan nilai *mAP* yang tinggi maka akan dilakukan pengujian pada data jalan berlubang. Hasil deteksi yang menunjukkan deteksi sukses (*true positive*), di mana model berhasil mengidentifikasi jalan berlubang dengan benar dan deteksi gagal (*false negative*), di mana model gagal mengenali lubang yang seharusnya terdeteksi. Secara visual, setiap hasil deteksi ditampilkan dalam bentuk gambar asli dengan *bounding box* beserta label kelas kerusakan dan nilai keyakinan (*confidence score*) pada gambar.

Pengujian ini bertujuan untuk memvalidasi kemampuan model dalam mengklasifikasikan tiga kategori kerusakan jalan secara akurat. Kinerja model berdasarkan metrik akurasi, yang menunjukkan persentase yang benar untuk setiap kelas. Hasil tingkat akurasi model pada data uji dapat dilihat pada Tabel 3.

Tabel 4. Tingkat Akurasi Model Pada Data Uji

Data Kerusakan	Jumlah Gambar	Jumlah Lubang	Akurasi
Ringan	31	85	96%
Sedang	33	171	95%
Tinggi	35	156	97%
Total	99	412	288%
Rata-rata			96%

Berdasarkan data yang disajikan pada Tabel 4, dapat dilakukan analisis mendalam mengenai performa model pada data uji. Tabel tersebut merinci hasil pengujian pada 99 gambar yang secara keseluruhan berisi 412 lubang jalan, yang terbagi ke dalam tiga kategori. Secara keseluruhan, model menunjukkan kinerja yang sangat memuaskan dengan mencapai rata-rata akurasi sebesar 96%.

Pada setiap kelas menunjukkan bahwa model memiliki kemampuan yang konsisten dan tinggi di semua kategori. Performa tertinggi dicapai pada klasifikasi kerusakan Tinggi dengan tingkat akurasi 97%, berdasarkan pengujian terhadap 35 gambar yang memuat 156 lubang. Tingginya akurasi ini kemungkinan besar disebabkan oleh fitur visual kerusakan berat yang lebih jelas dan mudah dikenali oleh model.

Untuk kelas Ringan, model berhasil mencapai akurasi 96% dari total 31 gambar dan 85 lubang. Sementara itu, pada kelas Sedang, model menunjukkan akurasi 95% dari 33 gambar yang memuat 171 lubang. Meskipun akurasi pada kelas Sedang sedikit lebih rendah, nilainya masih tergolong sangat tinggi dan menunjukkan bahwa model tidak mengalami kesulitan yang signifikan. Variasi akurasi yang sangat kecil (hanya 2%) antar kelas membuktikan bahwa model yang dikembangkan tidak bias dan memiliki kemampuan generalisasi yang seimbang untuk setiap tingkat keparahan kerusakan.

Deteksi Sukses

Hasil deteksi sukses diperoleh pada kerusakan jalan berlubang yang jika kita amati secara visual mudah didefinisikan, misalnya memiliki tekstur yang berbeda secara nyata jika dibandingkan dengan tekstur di sekelilingnya yang tidak rusak, dan memiliki tepi yang jelas meskipun dengan pencahayaan yang kurang. Deteksi sukses sering ditemui terutama pada jalan berlubang dengan kelas tinggi. Setiap gambar akan menunjukkan beberapa jenis kerusakan, lengkap dengan *bounding box* dan label yang diberikan oleh model. Gambaran deteksi sukses bisa dilihat pada Gambar 9.



(a) Jalan rusak ringan



(b) Jalan rusak sedang



(c) Jalan rusak tinggi

Gambar 21. Hasil deteksi sukses

Berdasarkan Gambar 22, pada setiap kerusakan memiliki nilai keyakinan (*confidence score*) yang dihasilkan untuk setiap deteksi. Pada jalan rusak ringan memiliki nilai keyakinan yang kurang optimal yaitu 0.78, pada kasus ini mengindikasikan adanya tantangan visual bagi model, karena kemungkinan kerusakannya terlihat samar. Warna lubang hampir menyatu dengan warna aspal dan tidak terlalu kontras sehingga objek sulit untuk diidentifikasi. Tapi model masih bisa mendeteksi bahwa itu adalah jalan rusak ringan. Dengan nilai keyakinan tersebut menunjukkan bahwa model bekerja dengan cukup baik.

Pada jalan rusak sedang memiliki nilai keyakinan yang sangat tinggi yaitu 0.93, dengan nilai keyakinan yang tinggi ini tentunya ada beberapa faktor visual yang memengaruhi. Pertama, objek kerusakan memiliki kontras warna yang signifikan dengan permukaan jalan di sekitarnya, sehingga memudahkan model untuk mengidentifikasi jalan rusak tersebut. Kedua, batas tepi kerusakan sangat jelas dan terdefinisi dengan jelas, sehingga model memberikan kemampuan yang baik dalam menentukan *bounding box*. Terakhir, tidak adanya halangan atau gangguan visual seperti bayangan memudahkan dan meningkatkan akurasi ekstraksi fitur oleh model. Kondisi yang ideal ini memungkinkan model untuk melakukan klasifikasi dengan nilai keyakinan yang tinggi.

Untuk jalan rusak tinggi yang memiliki nilai keyakinan yang baik tetapi belum maksimal yaitu 0.79 dan 0.80, dalam deteksi kerusakan ini dapat dijelaskan adanya halangan yang diakibatkan oleh genangan air menutupi fitur visual dari kerusakan, seperti tekstur permukaan dan kedalaman lubang yang sebenarnya. Akibatnya, model harus menyimpulkan berdasarkan fitur sekunder, yaitu bentuk dan keberadaan genangan air yang tidak biasa di permukaan jalan. Kondisi halangan ini menambah ketidakjelasan visual dan kerumitan dalam proses deteksi. Meskipun model berhasil mengklasifikasikan kerusakan dengan benar, ketidakjelasan tersebut menyebabkan nilai keyakinan tidak mencapai nilai yang maksimal seperti objek yang terlihat jelas tanpa halangan.

Deteksi Gagal

Selain memvalidasi keberhasilan, analisis terhadap kasus kegagalan merupakan tahap penting untuk memahami batasan dan kelemahan dari model yang dikembangkan. Analisis ini berfokus pada identifikasi kondisi atau jenis objek yang dapat menyebabkan model menghasilkan prediksi yang keliru. Pada sub bab ini, disajikan sebuah studi kasus deteksi gagal yang menyoroti terjadinya *False Positive*, yaitu ketika model mendeteksi

objek yang sebenarnya bukan merupakan kelas target. Contoh dari objek yang keliru bisa dilihat pada Gambar 23.



Gambar 22. Contoh kasus *False Positive* pada deteksi gagal

Pada Gambar 23. menampilkan sebuah pengujian yang di mana terjadi satu deteksi sukses dan satu deteksi gagal secara bersamaan. Deteksi sukses dilakukan terhadap objek lubang besar yang secara akurat diklasifikasikan sebagai jalan rusak tinggi dengan nilai keyakinan 0.90. Namun, fokus utama dari analisis ini adalah pada deteksi gagal yang teridentifikasi sebagai *False Positive*.

Secara spesifik, model keliru dalam mengidentifikasi area langit yang terlihat antara pepohonan sebagai jalan rusak sedang. Kesalahan klasifikasi ini kemungkinan besar disebabkan oleh kesamaan dalam ciri-ciri visual yang terlihat pada tingkat dasar. Area langit yang cerah dan di kelilingi oleh dahan pohon yang lebih gelap menciptakan sebuah elemen visual dengan kontras tinggi. Pola ini secara kebetulan menyerupai beberapa contoh kerusakan jalan misalnya, lubang tergenang air atau tambalan aspal yang telah dipelajari oleh model selama pelatihan.

Aspek yang paling mendasar untuk dianalisis dari deteksi gagal ini adalah nilai kepercayaan yang dihasilkan, yaitu 0.26. Nilai yang sangat rendah ini mengindikasikan bahwa model itu sendiri memiliki tingkat ketidakpastian yang signifikan terhadap prediksinya. Dalam penerapannya, kesalahan semacam ini dapat diatasi secara efektif dengan menerapkan ambang batas keyakinan (*confidence threshold*), misalnya 0.50. Dengan mekanisme penyaringan ini, setiap deteksi dengan skor di bawah ambang batas akan secara otomatis diabaikan

Kasus ini memberikan dua pemahaman yang penting: pertama, model rentan terhadap elemen visual di latar belakang yang menimbulkan keraguan. Kedua, metrik nilai keyakinan berfungsi sebagai mekanisme internal yang andal untuk mengidentifikasi dan menyaring prediksi yang tidak meyakinkan

KESIMPULAN

Berdasarkan hasil penelitian tugas akhir yang berjudul “Identifikasi Tingkat Kerusakan Jalan Berlubang Menggunakan Pengolahan Citra Digital” dapat disimpulkan sebagai berikut:

1. Penelitian ini telah berhasil merancang dan mengimplementasikan sistem untuk identifikasi dan klasifikasi tingkat kerusakan jalan berlubang menggunakan metode YOLOv8. Sistem ini mampu membedakan tiga kelas kerusakan, yaitu rusak ringan, rusak sedang, dan rusak tinggi.
2. Model yang dilatih menunjukkan kinerja yang sangat baik selama tahap pelatihan dan validasi. Hal ini dibuktikan dengan perolehan metrik evaluasi yang tinggi, dengan nilai Precision rata-rata 79,7%, Recall 87,1%, F1-Score 83,1% dan mAP-0.5 mencapai 90,6%.
3. Pada tahap pengujian menggunakan 99 data uji primer yang berisi 412 lubang, model

menunjukkan kemampuan generalisasi yang sangat memuaskan dengan rata-rata akurasi keseluruhan mencapai 96%. Performa model konsisten di semua kelas, dengan akurasi 97% untuk kelas rusak tinggi, 96% untuk kelas rusak ringan, dan 95% untuk kelas rusak sedang.

4. Analisis kualitatif menunjukkan bahwa performa deteksi dipengaruhi oleh faktor visual. Deteksi dengan keyakinan tinggi (0.93) terjadi pada objek dengan kontras dan batas tepi yang jelas, sementara tantangan seperti oklusi parsial oleh genangan air atau kontras rendah dapat menurunkan skor keyakinan (0.78-0.80).
5. Kelemahan utama model yang teridentifikasi melalui confusion matrix adalah dalam membedakan antara area rusak dengan background. Hal ini menyebabkan terjadinya kesalahan berupa False Negative (14 objek kerusakan terlewatkan) dan False Positive (18 deteksi keliru pada area background).

Saran

Beberapa saran yang dapat penulis berikan untuk penelitian lanjutan yang berhubungan dengan penelitian ini yaitu:

1. Untuk mengurangi kesalahan deteksi pada background (False Positive dan False Negative), disarankan untuk memperkaya dataset pelatihan. Hal ini dapat dilakukan dengan menambahkan lebih banyak variasi gambar background yang tidak memiliki kerusakan namun memiliki elemen yang berpotensi membingungkan (seperti bayangan, genangan air, atau perbedaan tekstur aspal) serta gambar kerusakan dalam berbagai kondisi pencahayaan dan cuaca yang berbeda.
2. Mengingat model terkadang menghasilkan deteksi False Positive dengan skor keyakinan yang sangat rendah (contoh: 0.26), disarankan untuk mengimplementasikan mekanisme penyaringan berbasis ambang batas keyakinan (confidence threshold) pada penerapan praktis. Deteksi dengan skor di bawah ambang batas tertentu (misalnya 0.50) dapat diabaikan untuk meningkatkan keandalan sistem secara keseluruhan.
3. Penelitian ini menggunakan model YOLOv8n yang ringan. Untuk penelitian selanjutnya, dapat dieksplorasi penggunaan varian YOLOv8 yang lebih besar (misalnya YOLOv8s atau YOLOv8m) untuk membandingkan apakah peningkatan kompleksitas model dapat mengatasi kasus-kasus deteksi yang sulit, meskipun mungkin dengan konsekuensi pada kecepatan inferensi.
4. Pengembangan sistem dapat diperluas untuk pemrosesan video secara real-time. Selain itu, mengintegrasikan sistem deteksi dengan Global Navigation Satellite System (GNSS) dapat menjadi pengembangan yang sangat bermanfaat untuk melakukan pemetaan lokasi kerusakan secara otomatis, seperti yang telah dilakukan pada penelitian terkait.
5. Untuk meningkatkan nilai praktis bagi pihak terkait, penelitian di masa depan dapat berfokus tidak hanya pada klasifikasi tingkat kerusakan, tetapi juga pada estimasi dimensi fisik kerusakan (seperti luas atau kedalaman lubang) menggunakan teknik pengolahan citra lebih lanjut.

DAFTAR PUSTAKA

- “Confusion Matrix: Pengertian, Cara Kerja dan Contoh Soal.” Diakses: 19 Agustus 2025. [Daring]. Tersedia pada: <https://kantinit.com/kecerdasan-buatan/confusion-matrix-pengertian-cara-kerja-dan-contoh-soal/>
- “Deteksi Objek dengan Kemampuan Tingkat Lanjut YOLOv8.” Diakses: 6 Mei 2025. [Daring]. Tersedia pada: <https://id.linux-console.net/?p=34055>
- A. Fadjeri, L. Kurniatin, D. K. Adri Ariyanto, dan B. A. Saputra, “Analisis Perbandingan Hasil Pengolahan Citra Asli Dan Cropping Untuk Mengidentifikasi Karakteristik Tanaman Selada Menggunakan Metode Morfologi Dan Ekstraksi Ciri,” Jurnal Ilmiah SINUS, vol. 21, no. 1,

- hlm. 73–82, Jan 2023, doi: 10.30646/sinus.v21i1.664.
- A. Habib Fitriansyah, E. Rachmawati, dan Risnandar, “PENGENALAN JALAN BERLUBANG BERBASIS VISION MENGGUNAKAN PYRAMID HISTOGRAM OF ORIENTED GRADIENTS,” vol. 10, no. 3, hlm. 709–716, Jun 2023, doi: 10.25126/jtiik.2023106820.
- A. Pamungkas, “Transformasi Geometri pada Citra: Flip dan Rotasi.” Diakses: 27 April 2025. [Daring]. Tersedia pada: <https://pemrogramanmatlab.com/2023/09/13/transformasi-geometri-pada-citra-flip-dan-rotasi/#more-10603>
- A. Wira Mulia, I. Ruslianto, dan D. Marisa Midyanti, “Klasifikasi Kerusakan Jalan Pada Citra Jalan Raya Pontianak dan Sekitarnya Dengan Menggunakan Convolutional Neural Network,” 2023.
- annisadev, “Mengenal Roboflow: Solusi Lengkap untuk Pengembangan Model Computer Vision.” Diakses: 27 Mei 2025. [Daring]. Tersedia pada: <http://annisadev.com/news/read/1012/mengenal-roboflow-solusi-lengkap-untuk-pengembangan-model-computer-vision.html>
- Ashari, N. Latif, dan A. Astuti, “PENGOLAHAN CITRA DIGITAL UNTUK MENENTUKAN BOBOT SAPI MENGGUNAKAN METODE CANNY EDGE DETECTION,” Jurnal Ilmiah Ilmu Komputer, vol. 5, no. 1, Apr 2019, [Daring]. Tersedia pada: <http://ejournal.fikom-unasman.ac.id>
- B. Sasmito, B. H. Setiadji, dan R. Isnanto, “Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning di Kota Semarang,” TEKNIK, vol. 44, no. 1, hlm. 7–14, Mei 2023, doi: 10.14710/teknik.v44i1.51908.
- B. Sitohang dan A. Sindar, “Analisis Dan Perbandingan Metode Sobel Edge Detection Dan Prewit Pada Deteksi Tepi Citra Daun Srilangka,” Jurnal Nasional Komputasi dan Teknologi Informasi, vol. 3, no. 3, Des 2020.
- dewaweb, “Mengenal Apa Itu Google Colab dan Cara Menggunakannya.” Diakses: 3 Juni 2025. [Daring]. Tersedia pada: <https://www.dewaweb.com/blog/mengenal-google-colab/>
- E. Darmawan, A. Rosadi, dan E. Kresna Andana, “Eci Darmawan, Pengolahan Citra Menggunakan Border Following Dan Support Vector Machine Pengolahan Citra Menggunakan Border Following Dan Support Vector Machine (SVM) Dalam Mengidentifikasi Kerusakan Jalan Raya,” vol. 2, no. 1, hlm. 14–18, 2023.
- E. Setiadi dan A. Wibowo, “Klasifikasi dan Deteksi Keretakan Pada Trotoar Menggunakan Metode Convolutional Neural Network,” 2023.
- I. B. Pakpahan dan I. C. Dewi, “Pendeteksian Lubang Pada Jalanan Menggunakan Metode SSD-MobileNet,” IJEIS (Indonesian Journal of Electronics and Instrumentation Systems), vol. 11, no. 2, hlm. 213–222, Okt 2021, doi: 10.22146/ijeis.60157.
- J. R. Priandini, “PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN MODEL YOU ONLY LOOK ONCE (YOLO) V8,” Agu 2024.
- J. Tores, “YOLOv8 Architecture Explained: Exploring the YOLOv8 Architecture.” Diakses: 6 Mei 2025. [Daring]. Tersedia pada: <https://yolov8.org/yolov8-architecture-explained/>
- L. Sukmawati dan R. Sadikin, “Segmentasi Jalan Berlubang Citra Jalan Raya Menggunakan Metode Thresholding Dan K-Means,” Jurnal Teknik Komputer, vol. 9, no. 2, hlm. 89–95, Agu 2023, doi: 10.31294/jtk.v9i2.15211.
- N. G. Pramuditasari dan D. Iskandar Mulyana, “Optimasi Image Klasifikasi Kondisi Jalan Rusak dengan Metode Image Enhancement dan Morfologi,” 2022.
- N. Khairunisa, Carudin, dan A. Jamaludin, “ANALISIS PERBANDINGAN ALGORITMA CNN DAN YOLO DALAM MENGIDENTIFIKASI KERUSAKAN JALAN,” Jurnal Informatika dan Teknik Elektro Terapan, vol. 12, no. 3, hlm. 1756–1768, Agu 2024, doi: 10.23960/jitet.v12i3.4434.
- Nurfiyah, Y. Rianto, dan D. Riana, “IDENTIFIKASI TINGKAT KERUSAKAN JALAN RAYA MENGGUNAKAN THRESHOLDING DAN K-MEANS IDENTIFICATION OF ROAD DAMAGE USING THRESHOLDING AND K-MEANS,” Daerah Khusus Ibukota Jakarta, vol. 13, no. 1, hlm. 34–44, Feb 2021, doi: 10.22303/csrid.13.1.2021.34-44.
- P. B. Prakoso, U. Sylvia Lestari, dan Y. Sari, “DETEKSI KERETAKAN PERMUKAAN PERKERASAN LENTUR JALAN RAYA (STUDI KASUS: TANAH LUNAK DI

- BANJARMASIN) Detection of Cracks on Highway Flexible Pavement Surfaces (Case Study: Soft Soils in Banjarmasin),” Prosiding Seminar Nasional Lingkungan Lahan Basah, vol. 4, Apr 2019.
- P. B. Prakoso, Y. Sari, dan M. A. Rahman, “Perkiraan Luas Kerusakan Permukaan Jalan Dengan Menggunakan Metode Thresholding,” 2020.
- P. Bagus, D. Pranata, I. Gede, dan A. Wibawa, “Segmentasi Kerusakan Jalan Menggunakan Metode Binary Thresholding,” vol. 11, no. 4, hlm. 2654–5101, Mei 2023.
- R. Hayu Pramestya, “DETEKSI DAN KLASIFIKASI KERUSAKAN JALAN ASPAL MENGGUNAKAN METODE YOLO BERBASIS CITRA DIGITAL,” 2018.
- R. V. Sukmaningsih, B. Heru Iswanto, dan H. Suhendar, “KLASIFIKASI KERUSAKAN JALAN RAYA BERBASIS CITRA UDARA MENGGUNAKAN OBJECT-BASED IMAGE-ANALYSIS (OBIA),” PRODI Pendidikan Fisika dan Fisika UNJ, Jan 2024. doi: 10.21009/03.1201.fa05.
- S. Ratna, “PENGOLAHAN CITRA DIGITAL DAN HISTOGRAM DENGAN PHYTON DAN TEXT EDITOR PHYCHARM,” Sep 2020.
- Y. Sari1, A. Rizky Baskara, B. Prakoso, dan M. A. Rahman, “PENERAPAN ACTIVE CONTOUR MODEL PADA PENGOLAHAN CITRA UNTUK DETEKSI KERUSAKAN JALAN (APPLICATION OF ACTIVE CONTOUR MODEL ON IMAGE PROCESSING FOR DETECTION OF ROAD DAMAGE),” Nov 2021.