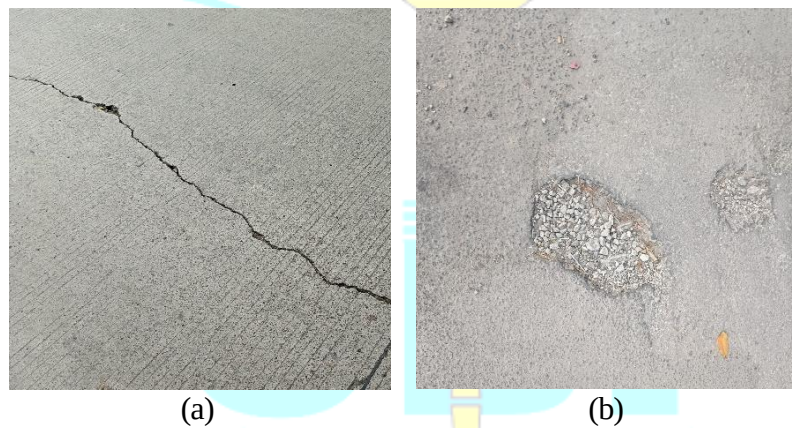


BAB III METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah model deteksi kerusakan jalan menggunakan framework *YOLOv9*, sebuah arsitektur *Convolutional Neural Network (CNN)*. Penelitian ini dilakukan untuk mendeteksi berbagai jenis kerusakan jalan seperti retakan dan lubang, yang merupakan salah satu penyebab utama kecelakaan lalu lintas. Citra jalan yang digunakan dalam penelitian ini mencakup berbagai kondisi pencahayaan dan jenis kerusakan untuk memastikan model yang dihasilkan memiliki performa yang optimal. Berikut contoh citra jalan dengan kerusakan.



Gambar 3. 1. Citra (a)jalan retak, (b)jalan berlubang

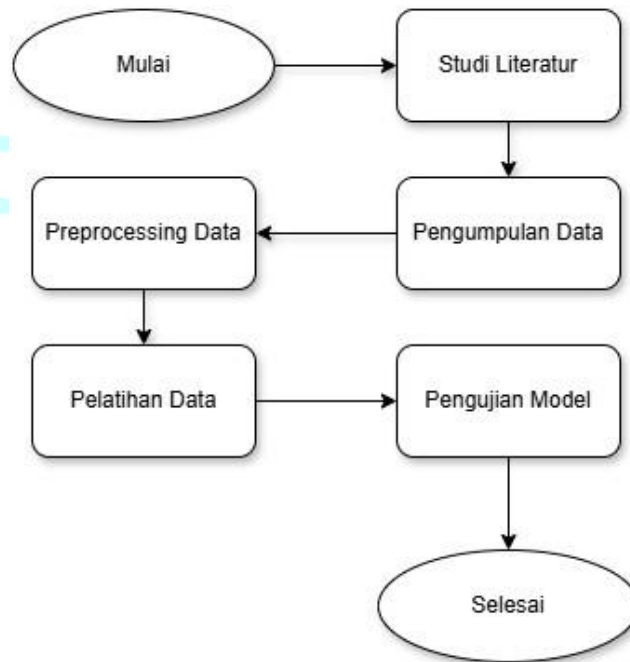
Gambar 3.1.a menunjukkan jalan retak dengan kerusakan berupa celah atau garis pecah akibat tekanan kendaraan, perubahan suhu, atau material yang kurang baik. Sementara itu, Gambar 3.1.b menggambarkan jalan berlubang yang ditandai dengan cekungan pada permukaan jalan akibat erosi air, tekanan kendaraan berat, dan pemeliharaan yang tidak memadai. Waktu penelitian direncanakan berlangsung selama enam bulan, mulai dari studi literatur hingga analisis hasil pelaksanaan penelitian sebagai berikut.

Tabel 3. 1. Waktu Penelitian

No.	Kegiatan	Nov'24	Des'24	Jan'25	Feb'25	Mar'25	April'25
1	Studi Literatur						
2	Pengumpulan Data						
3	Preprocessing dan Augmentasi Data						
4	Pelatihan Model						
5	Pengujian Model						

3.2 Prosedur Penelitian

Prosedur penelitian ini dirancang untuk memastikan implementasi dan evaluasi model deteksi kerusakan jalan menggunakan YOLOv9 dapat dilakukan secara sistematis dan terstruktur. Prosedur ini terdiri dari tiga tahapan utama, yaitu pengumpulan data, *preprocessing* dan *augmentasi* data, pelatihan model, dan skenario pengujian.



Gambar 3. 2. Prosedur Penelitian

3.2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri atas citra jalan yang menunjukkan berbagai jenis kerusakan, seperti retakan, lubang. Proses pengambilan citra dilakukan menggunakan perangkat kamera smartphone merk *Techno* dengan resolusi kamera 108 *Megapixel* untuk memastikan detail kerusakan jalan dapat terlihat dengan jelas. Kamera digunakan dari sudut pandang 45 derajat dengan jarak satu meter satu objek untuk menangkap kondisi jalan secara optimal.

Penelitian dilaksanakan di Kabupaten Karawang dengan pengumpulan data dilakukan secara langsung di lapangan. Citra diambil dari jalanan beraspal dan beton dalam kondisi kering untuk memastikan hasil pencitraan tidak terganggu oleh faktor lingkungan, seperti air atau refleksi. Lokasi pengambilan data mencakup berbagai jenis jalan, seperti jalan utama, jalan lingkungan, dan kawasan industri contoh citra jalan rusak ada pada gambar 3.1. Variasi ini bertujuan untuk mencakup

beragam jenis kerusakan jalan yang mungkin timbul akibat perbedaan kondisi lingkungan dan tingkat penggunaan jalan.

Untuk meningkatkan representasi dataset, citra diambil dalam berbagai kondisi pencahayaan, seperti pada siang hari dengan pencahayaan terang dan sore hari dengan intensitas cahaya lebih redup. Selain itu, jenis jalan yang divariasikan memungkinkan model *YOLOv9* mengenali kerusakan jalan dalam berbagai konteks visual dan kondisi nyata.

3.2.2 Preprocessing Data

Preprocessing data adalah langkah penting dalam mempersiapkan data sebelum dianalisis lebih lanjut atau digunakan untuk melatih model *YOLOv9*. Dalam penelitian ini, tahap *preprocessing* meliputi beberapa proses utama: anotasi, *auto-orientasi*, penyesuaian ukuran citra, konversi ke *grayscale*, penyesuaian kontras otomatis, dan *augmentasi* data untuk memperbanyak jumlah dataset.

1. Anotasi Data

Anotasi dilakukan secara manual menggunakan situs web **Roboflow.com** dengan teknik *bounding box* untuk menandai area objek kerusakan jalan. Proses ini memastikan objek yang relevan teridentifikasi, meskipun dengan ketepatan yang masih bisa diperbaiki.



Gambar 3. 3. Anotasi Data menggunakan Roboflow.com

2. Auto-Orientasi

Langkah ini digunakan untuk memastikan citra memiliki orientasi yang seragam, yang sangat penting agar model dapat mendeteksi kerusakan pada jalan secara lebih akurat. Dalam konteks ini, *auto-orientasi* bertujuan untuk mengatasi masalah citra yang mungkin diambil dalam berbagai orientasi, seperti citra yang diputar atau miring (*landscape* atau *portrait*).

3. Penyesuaian Ukuran (*resize*)

Ukuran citra diubah agar sesuai dengan dimensi standar yang telah ditentukan, misalnya mengubah semua citra menjadi ukuran yang telah ditentukan. Hal ini penting untuk menjaga konsistensi ukuran data yang digunakan dalam pelatihan model. Selain itu, penyesuaian ukuran juga dapat membantu mengurangi beban komputasi saat model dilatih. Citra dengan ukuran besar memerlukan lebih banyak memori dan waktu untuk diproses, sehingga penyesuaian ukuran membantu mempercepat proses pelatihan dan mengurangi kebutuhan sumber daya komputasi.

4. Penyesuaian Kontras Otomatis (*Auto-Adjust Contrast*)

Teknik *adaptive equalization* diterapkan untuk meningkatkan kontras citra. Ini membantu menonjolkan detail yang mungkin kurang jelas dalam citra asli, seperti kerusakan jalan yang memiliki kontras rendah dengan latar belakangnya. Dengan meningkatkan kontras, bagian-bagian penting dari citra, seperti retakan atau lubang, menjadi lebih jelas dan lebih mudah dikenali oleh model. Proses ini penting untuk memastikan bahwa model tidak melewatkan detail kecil yang mungkin penting untuk identifikasi kerusakan.

5. *Augmentasi Data*

Augmentasi data dilakukan untuk memperbesar jumlah dan keragaman dataset pelatihan, yang membantu model belajar mengenali kerusakan jalan dalam berbagai kondisi. Beberapa teknik *augmentasi* yang diterapkan antara lain:

- a. *Konversi ke Grayscale*: Setelah citra diubah menjadi skala abu-abu (*grayscale*), informasi warna yang tidak dibutuhkan dihilangkan. Hal ini membantu untuk menyederhanakan citra dan memungkinkan model untuk fokus pada detail struktural dan tekstural, seperti retakan atau lubang pada jalan, yang lebih relevan dalam deteksi kerusakan. Dengan menghilangkan informasi warna, citra menjadi lebih sederhana untuk diproses, dan model dapat lebih cepat mendeteksi pola yang relevan tanpa terganggu oleh variasi warna yang mungkin tidak penting.

- b. Membalik citra secara horizontal dan vertikal: Untuk menambah variasi citra, misalnya, jika model dilatih hanya dengan citra yang menghadap ke kanan, maka citra yang dibalik secara horizontal dapat membantu model mengenali kerusakan yang berada pada arah berlawanan.
- c. Memutar citra sebesar 90 derajat atau dalam rentang tertentu: Rotasi citra meningkatkan variasi data, sehingga model tidak terlalu bergantung pada posisi spesifik objek dalam citra.
- d. Menggeser posisi citra secara linier: Menggeser citra secara horizontal atau vertikal memungkinkan model belajar mengenali kerusakan meski objeknya tidak berada di tengah citra.
- e. Menyesuaikan tingkat eksposur atau kecerahan citra: Mengubah tingkat kecerahan citra dapat membantu model mengenali kerusakan dalam berbagai kondisi pencahayaan.
- f. Menambahkan noise pada citra: Penambahan noise bertujuan untuk melatih model agar tetap dapat mengenali objek meskipun ada gangguan visual, yang dapat terjadi di dunia nyata.

Augmentasi ini bertujuan untuk meningkatkan ketahanan model terhadap variasi citra yang berbeda, seperti sudut pengambilan citra, pencahayaan, atau gangguan visual, yang mungkin muncul dalam citra kerusakan jalan yang berbeda.

6. Pembagian Dataset

Dataset dibagi menjadi tiga bagian utama: 70% untuk data pelatihan, 10% untuk data pengujian, dan 20% untuk data validasi. Pembagian ini penting untuk memastikan bahwa model dilatih dengan data yang cukup, diuji dengan data yang belum pernah dilihat sebelumnya, dan divalidasi untuk mengevaluasi kinerja model secara objektif. Pembagian ini memungkinkan model untuk belajar dari data pelatihan, menguji kemampuannya dengan data pengujian, dan mengevaluasi apakah model bekerja dengan baik pada data yang baru dan tidak dilihat sebelumnya menggunakan data validasi.

Tahapan preprocessing ini bertujuan untuk menghasilkan dataset yang konsisten dan berkualitas tinggi, sehingga model YOLOv9 dapat dilatih secara

efektif dan mampu mendeteksi kerusakan jalan dengan baik. Data yang sudah dilabeli dan diproses kemudian digunakan untuk konfigurasi model YOLOv9 agar dapat menjalankan deteksi dengan akurasi yang optimal.

3.2.3 Pelatihan Model YOLOv9

Setelah tahap preprocessing data selesai, langkah berikutnya adalah melatih model YOLOv9 untuk mendeteksi kerusakan jalan. Proses pelatihan dilakukan menggunakan *Google Colaboratory*. Penelitian ini menggunakan laptop dengan *processor* i5 1035G7 dengan *RAM (Random Access Memory)* sebesar 8 GB. Berikut adalah *pseudocode* pelatihan model YOLOv9 untuk mendeteksi kerusakan jalan:

Langkah 1: Mulai

Langkah 2: Membaca dataset

Langkah 3: Persiapan model YOLOv9 dan argumen

Langkah 4: Pelatihan model YOLOv9

Langkah 5: Menyimpan bobot terbaik

Langkah 6: Selesai

Pada *pseudocode* diatas peneliti membaca data yang sudah disiapkan sebelumnya yang sudah dibagi datanya menjadi data latih, validasi dan pengujian. Persiapan model YOLOv9 dan memasangnya di *google drive* untuk mempersiapkan pelatihan. Argumen-argumen yang diterapkan di pelatihan ini yaitu: *batch size* 16, *epoch* 110, resolusi input 640x640 piksel dan pelatihan model pun dimulai dengan *GPU* yang tersedia di *Google Colaboratory*. Pelatihan ini menghasilkan bobot untuk setiap iterasinya dan hanya bobot terbaik saja yang akan disimpan.

3.2.4 Pengujian Model

Setelah model YOLOv9 selesai dilatih, tahap pengujian dilakukan untuk mengevaluasi performa model dalam mendeteksi kerusakan jalan. Proses pengujian dilakukan menggunakan dataset uji yang sebelumnya telah disiapkan (20% dari total dataset). Model diuji pada dataset uji dengan melakukan deteksi kerusakan jalan seperti retakan dan lubang. Hasil deteksi dibandingkan dengan label *ground truth* untuk menghitung metrik evaluasi berikut:

1. *Precision*: Mengukur sejauh mana prediksi model benar dibandingkan dengan keseluruhan prediksi positif.

2. *Recall*: Mengukur sejauh mana model berhasil mendeteksi kerusakan jalan yang benar-benar ada.
3. *F1-Score*: Menggabungkan nilai precision dan recall untuk memberikan ukuran kinerja yang seimbang.

Berikut adalah *pseudocode* pengujian model YOLOv9 untuk mendeteksi kerusakan jalan:

Langkah 1: Mulai

Langkah 2: Masukkan citra

Langkah 3: Persiapkan model deteksi objek

Langkah 4: Lakukan deteksi pada citra

Langkah 5: Bandingkan dengan ground truth

Langkah 6: Hitung metrik evaluasi

Langkah 7: Hitung nilai performa model

Langkah 8: Tampilkan hasil evaluasi

Langkah 9: Selesai

Pseudocode pseudocode diatas memaparkan proses pengujian model YOLOv9 untuk mendeteksi kerusakan jalan. Model akan memproses citra input, menghasilkan bounding box yang menunjukkan lokasi dan jenis kerusakan, lalu membandingkan hasil dengan label ground truth. Metrik evaluasi seperti *precision*, *recall* dan *F1-score* akan dihitung juga dengan nilai rata-ratanya dengan menggunakan rumus sebagai berikut:

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}$$

$$\text{F1 - Score} = 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Di mana:

- TP: *True Positive* (kerusakan jalan yang terdeteksi dengan benar).
- FP: *False Positive* (prediksi salah, objek yang terdeteksi bukan kerusakan jalan).
- FN: *False Negative* (kerusakan jalan yang tidak terdeteksi).

Hasil deteksi model divisualisasikan dalam bentuk bounding box pada gambar uji. Setiap *bounding box* menunjukkan lokasi kerusakan jalan yang terdeteksi, jenis kerusakan (misalnya, retakan atau lubang), dan tingkat kepercayaan (confidence score). Visualisasi ini dilakukan untuk memastikan bahwa deteksi model sesuai dengan ground truth. Hasil pengujian ini akan digunakan untuk menilai apakah model *YOLOv9* sudah cukup akurat dan efisien untuk aplikasi pemeliharaan jalan secara otomatis.

