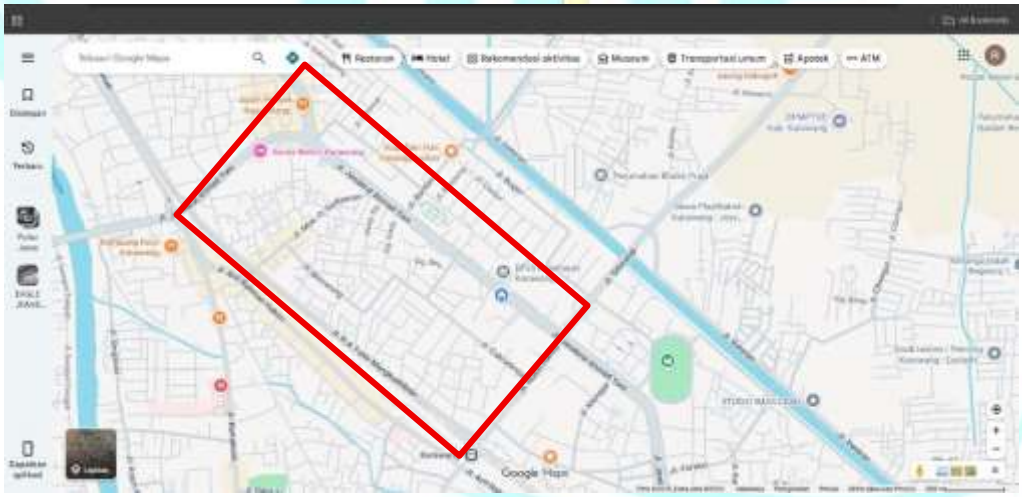
BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian dalam studi ini adalah model deteksi objek berbasis *deep learning* yang digunakan untuk mengidentifikasi kondisi kerusakan jalan berupa jalan berlubang (*potholes*) dan jalan retak (*cracks*). Penelitian ini berfokus pada pemanfaatan citra digital sebagai sumber data dalam proses pelatihan dan pengujian model, dengan tujuan menilai kemampuan model dalam menghasilkan prediksi yang akurat, cepat, dan konsisten. Objek penelitian mencakup data citra yang bersumber dari dataset publik Kaggle serta dokumentasi pribadi yang dikumpulkan secara langsung di lapangan. Penelitian ini juga memfokuskan perhatian pada dua model deteksi objek, yaitu YOLOv8 dan YOLOv11, yang dipilih untuk dianalisis performanya dalam mendeteksi kerusakan jalan serta kemampuan dalam mengklasifikasikan jenis kerusakan berdasarkan hasil deteksi, sehingga menghasilkan informasi kondisi permukaan jalan yang lebih komprehensif.



Gambar 3. 1 Lokasi pengujian deteksi lubang jalan
sumber: https://maps.google.com/

Gambar 3. 1 menunjukkan lokasi penelitian yang berada di wilayah Kabupaten Karawang, Jawa Barat, khususnya di sekitar Jalan Jenderal Ahmad Yani, Jalan Arif Rahman Hakim, dan area permukiman sekitarnya. Lokasi ini dipilih karena memiliki variasi kondisi permukaan jalan serta intensitas lalu lintas yang cukup tinggi, sehingga sering memunculkan kerusakan berupa jalan berlubang atau retak akibat perubahan cuaca dan beban kendaraan. Keberagaman kondisi

visual pada area tersebut mulai dari pencahayaan, tekstur permukaan aspal, hingga sudut pengambilan citra menjadikan lokasi ini representatif sebagai area uji untuk mengevaluasi performa model YOLO dalam mendeteksi lubang jalan pada situasi nyata.

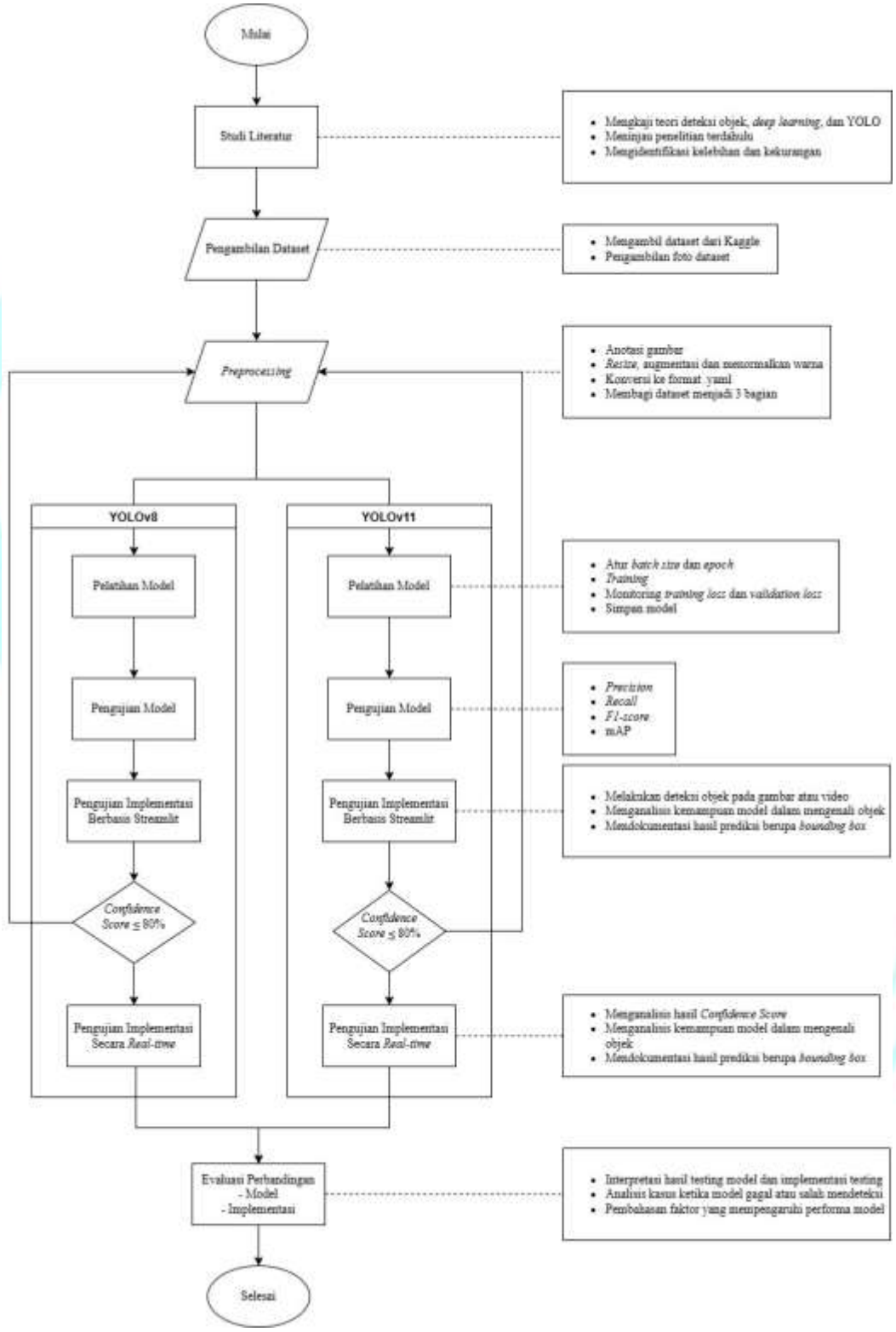
Hasil prediksi yang menjadi bagian dari objek penelitian meliputi keluaran berupa *bounding box* yang menunjukkan posisi objek, tingkat keyakinan model atau *confidence score*, serta label prediksi yang menggambarkan potensi kerusakan jalan pada citra yang dianalisis. Seluruh rangkaian penelitian dilakukan menggunakan perangkat komputasi pribadi.

3.2 Prosedur Penelitian

Prosedur penelitian ini dirancang secara sistematis untuk menghasilkan proses pengembangan dan evaluasi model deteksi objek yang komprehensif. Setiap tahapan dirumuskan agar saling berkaitan dan membentuk alur kerja yang runtut, sehingga model YOLOv8 dan YOLOv11 dapat dibandingkan secara objektif. Pendekatan metodologis yang diterapkan mencakup pengumpulan data, persiapan dataset, pelatihan model, pengujian, hingga analisis mendalam terhadap hasil yang diperoleh.

Selain menghasilkan model deteksi yang akurat, prosedur penelitian ini juga bertujuan untuk memastikan bahwa proses dapat direplikasi oleh peneliti lain yang tertarik pada bidang deteksi kerusakan jalan berbasis *computer vision*. Oleh karena itu, setiap tahapan disusun dengan deskripsi yang terperinci, termasuk pemilihan alat, metode pemrosesan data, serta teknik evaluasi yang digunakan. Alur penelitian yang telah disusun divisualisasikan dalam bentuk diagram alir sehingga setiap langkah dapat dipahami secara sistematis. Visualisasi ini berfungsi untuk memperjelas hubungan antartahapan dan memberikan gambaran menyeluruh mengenai proses penelitian yang dilakukan. Dengan adanya diagram alir tersebut, proses penelitian dapat dipahami sebagai suatu rangkaian tahapan yang saling terintegrasi, mulai dari tahap awal hingga diperolehnya hasil akhir. Setiap tahapan dirancang untuk mendukung keberhasilan tahap berikutnya, sehingga membentuk alur kerja yang logis, terstruktur, dan mudah untuk direplikasi. Selain itu, visualisasi ini juga membantu dalam mengidentifikasi keterkaitan antarproses serta memastikan bahwa seluruh tahapan penelitian telah dilaksanakan secara sistematis

dan sesuai dengan tujuan yang telah ditetapkan. Gambar 3. 2 adalah alur penelitian yang digunakan dalam penelitian ini.



Gambar 3. 2 Alur Penelitian

3.2.1 Studi Literatur

Tahap studi literatur merupakan fondasi awal dalam penelitian ini. Pada tahap ini, peneliti mengumpulkan berbagai referensi dari jurnal internasional, konferensi ilmiah, buku, dan artikel terkait yang membahas deteksi kerusakan jalan, algoritma YOLO, serta perkembangan *deep learning* dalam *computer vision*. Literatur tersebut memberikan gambaran mengenai bagaimana teknologi deteksi objek berkembang dari waktu ke waktu, serta bagaimana YOLO digunakan secara luas dalam berbagai bidang seperti kesehatan, pertanian, industri, dan transportasi. Dengan mempelajari berbagai penelitian terdahulu, peneliti dapat memahami tren teknologi terbaru dan melihat peluang inovasi pada topik deteksi jalan berlubang.

Selain itu, studi literatur membantu peneliti dalam memahami karakteristik khusus dari masing-masing versi YOLO. YOLOv8 dan YOLOv11 merupakan dua arsitektur modern yang memiliki perbedaan dalam struktur jaringan, modul ekstraksi fitur, efisiensi komputasi, serta stabilitas deteksi. Dengan memahami aspek ini melalui literatur, peneliti dapat menentukan parameter pelatihan, teknik evaluasi, serta pendekatan komparatif yang tepat. Hal ini penting agar perbandingan performa kedua model tidak hanya dilakukan berdasarkan hasil numerik, tetapi juga berdasarkan pemahaman mendalam tentang arsitektur dan mekanisme kerja model.

Selanjutnya, studi literatur juga memungkinkan peneliti mengindentifikasi kelemahan penelitian terdahulu. Salah satu celah yang ditemukan adalah kurangnya penelitian yang membandingkan performa YOLO versi terbaru seperti YOLOv11 dengan versi sebelumnya secara langsung khusus pada kasus deteksi jalan berlubang. Celah ini menjadi motivasi utama penelitian, sekaligus menjadi dasar untuk menyusun metodologi penelitian yang lebih komprehensif. Dengan demikian, studi literatur tidak hanya berfungsi sebagai referensi, tetapi juga menjadi landasan logis dalam penyusunan fokus penelitian.

3.2.2 Pengambilan Dataset

Pengambilan dataset dilakukan dengan memilih sumber data yang relevan untuk kasus deteksi kondisi permukaan jalan. Sumber dataset berasal dari platform Kaggle, yaitu dataset *Annotated Potholes with Severity Levels* yang disusun oleh Idan Baruch, dengan jumlah sebanyak 347 citra. Dataset tersebut memuat variasi kondisi jalan berlubang dengan tingkat kerusakan yang beragam, mulai dari lubang berukuran kecil yang tampak sebagai retakan dangkal hingga lubang berukuran besar yang berpotensi membahayakan pengguna jalan.

Selain itu, penelitian ini juga menambahkan dataset *Roads and Bridges Cracks (YOLOv8 Format)* yang disusun oleh Danish Ghaffar, dengan total sebanyak 317 citra jalan retak sebagai pelengkap untuk meningkatkan variasi data, khususnya dalam mengenali kondisi jalan yang mengalami keretakan. Penambahan dataset ini bertujuan agar model tidak hanya mampu mendeteksi jalan berlubang, tetapi juga dapat mengklasifikasikan jenis kerusakan jalan lainnya.

Selain dataset dari Kaggle, penelitian ini juga menggunakan data tambahan berupa citra jalan berlubang yang diambil secara mandiri oleh peneliti. Pengambilan citra dilakukan untuk menambah variasi data serta menyesuaikan kondisi lingkungan jalan yang lebih merepresentasikan keadaan nyata di lapangan. Citra tambahan tersebut diambil dengan mempertimbangkan perbedaan pencahayaan, sudut pengambilan gambar, latar belakang, serta ukuran lubang jalan.

Secara keseluruhan, dataset yang digunakan dalam penelitian ini merupakan gabungan dataset Kaggle dan hasil dokumentasi mandiri. Variasi pada dataset meliputi perbedaan tingkat kerusakan jalan (berlubang dan retak), kondisi pencahayaan, sudut pengambilan gambar, gangguan visual, serta ukuran kerusakan jalan, sehingga diharapkan model YOLOv8 dan YOLOv11 dapat dilatih dengan data yang beragam dan mampu mencerminkan kondisi nyata di lapangan. Gambaran umum mengenai kumpulan dataset yang digunakan dalam penelitian ini dapat dilihat pada

Gambar 3. 3, yang menunjukkan contoh distribusi citra jalan berlubang dan jalan retak sebagai objek utama deteksi model.



Gambar 3. 3 Dataset Sumber Kaggle

3.2.3 *Preprocessing*

Tahap berikutnya adalah melakukan proses anotasi menggunakan platform Roboflow. Anotasi dilakukan dengan menandai area lubang menggunakan *bounding box*. Proses ini dilakukan secara manual untuk memastikan ketepatan label data, karena kualitas anotasi sangat berpengaruh terhadap kemampuan model dalam mendeteksi objek. Setelah anotasi selesai, dataset diproses melalui tahap *preprocessing* seperti mengubah resolusi gambar menjadi 640×640 piksel, menormalkan warna, dan mengonversi format anotasi menjadi standar YOLO. Tahapan ini bertujuan agar data siap digunakan dalam proses pelatihan kedua model (Bagherpour, 2025).

Kemudian dataset dibagi menjadi tiga kelompok utama: *training set* (70%), *validation set* (20%), dan *test set* (10%). Pembagian dataset secara proporsional ini bertujuan untuk menghindari bias data sekaligus memastikan model diuji secara objektif menggunakan data yang benar-benar baru bagi model (Indra et al., 2025).

Setelah pembagian dataset dilakukan, seluruh anotasi yang telah dibuat kemudian diekspor ke dalam format YOLO berupa file .txt pada setiap citra, berisi koordinat *bounding box* dan kelas objek. Tahap berikutnya adalah melakukan penyusunan struktur dataset dan membuat file konfigurasi data.yaml, yang berfungsi sebagai penghubung antara dataset dengan model saat proses pelatihan dijalankan. File .yaml memuat informasi mengenai lokasi direktori *data train*, *validation*, dan *test*, serta daftar label kelas yang digunakan dalam penelitian. Penyiapan file .yaml memastikan bahwa dataset dapat dibaca secara otomatis oleh *frame* work

YOLOv8 dan YOLOv11 saat *train*ing dimulai, sehingga setiap citra dan anotasi dapat dipanggil dengan benar oleh model.

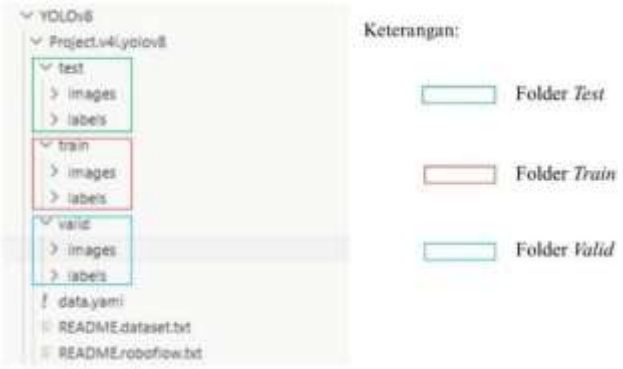
3.2.4 Pelatihan Model

Pelatihan model dilakukan terhadap dua arsitektur deteksi objek modern, yaitu YOLOv8 dan YOLOv11, menggunakan dataset yang telah melalui tahap anotasi dan pembagian menjadi tiga bagian utama *training set*, *validation set*, dan *test set*. Proses pembagian dataset dilakukan dengan proporsi 70% data untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian akhir. Ketiga subset data tersebut kemudian diproses pada masing-masing model secara terpisah agar proses pembelajaran berlangsung terstruktur dan dapat dibandingkan secara objektif.

Pelatihan model dijalankan pada perangkat dengan dukungan GPU untuk memastikan proses komputasi berjalan efisien. Parameter dasar seperti *epoch*, *batch size*, *learning rate*, dan resolusi *input* (640×640 px) diseragamkan untuk menghindari bias performa antar-model. Selama proses pelatihan berlangsung, *training loss* dan *validation loss* diamati secara berkala guna mendeteksi indikasi *underfitting* maupun *overfitting*. Model dengan performa terbaik pada validasi kemudian disimpan sebagai *best weights* untuk dijalankan pada tahap pengujian.

A. Pelatihan Model YOLOv8

Pada YOLOv8, dataset yang telah dibagi menjadi *train*, *validation*, dan *test* digunakan sesuai perannya masing-masing. *Train set* digunakan sebagai data utama bagi model untuk mempelajari pola jalan berlubang melalui *proses forward* dan *backpropagation*. Seluruh citra yang berada di dalam folder *train* digunakan untuk memperbaharui bobot jaringan melalui komputasi convolution pada setiap iterasi. Dengan demikian, model perlahan memahami struktur visual dari lubang jalan, tepi objek, tekstur permukaan aspal, serta pola pencahayaan. Struktur direktori penyimpanan dataset untuk YOLOv8 dapat dilihat pada Gambar 3. 4, yang menampilkan pembagian folder *train*, *validation*, dan *test* secara terorganisasi sebagai dasar pelatihan model.



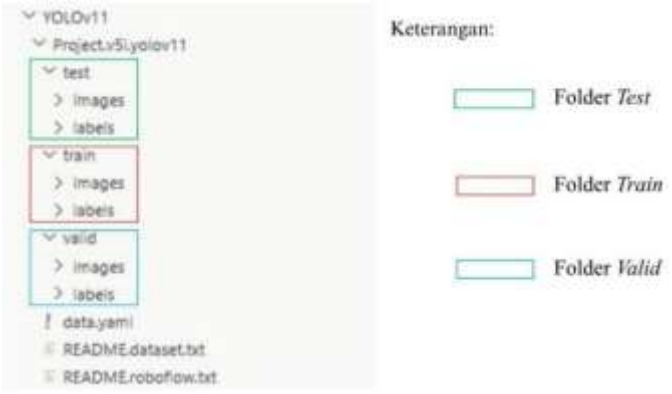
Gambar 3. 4 Folder YOLOv8

Selanjutnya, *validation set* berfungsi sebagai data pemantau performa selama pelatihan. Data validasi tidak digunakan untuk pembaruan bobot, tetapi hanya untuk mengevaluasi apakah model generalisasi dengan baik pada data yang belum dilihat. Apabila *validation loss* menurun sebanding dengan *training loss*, maka proses pelatihan berjalan stabil. Namun apabila *training loss* menurun sementara *validation loss* meningkat, maka indikasi *overfitting* perlu diperhatikan.

Kemudian, *test set* tidak digunakan pada proses *training* maupun validasi. Data ini baru digunakan setelah pelatihan selesai untuk mengukur performa akhir YOLOv8 secara objektif.

B. Pelatihan Model YOLOv11

Seperti pada YOLOv8, model YOLOv11 juga dilatih menggunakan dataset yang terdiri dari *train set, validation set*, dan *test set*. Pembagian dataset sama, namun cara pemrosesan fitur dalam arsitektur model memiliki perbedaan. Pada *train set*, YOLOv11 mempelajari objek lubang jalan melalui proses ekstraksi fitur dengan dukungan modul *attention* yang lebih adaptif terhadap variasi ukuran objek dan tekstur aspal. Model ini membaca citra dan label *bounding box* dari direktori *training* kemudian memperbarui bobot secara bertahap setiap iterasi hingga mencapai kestabilan. Pembagian direktori dataset YOLOv11 ditunjukkan pada Gambar 3. 5, yang memperlihatkan susunan folder yang digunakan model untuk membaca data selama proses *train*ing.



Gambar 3. 5 Folder YOLOv11

Validation set kembali berperan sebagai indikator kualitas pelatihan. Model dievaluasi secara berkala menggunakan data validasi untuk melihat apakah bobot yang dihasilkan mampu bekerja pada data baru tanpa penurunan kinerja. YOLOv11 biasanya menunjukkan respons lebih cepat pada stabilisasi *validation loss* dibandingkan YOLOv8 karena memiliki mekanisme penangkapan fitur multi-skala yang lebih efisien. Selama proses ini nilai mAP, *precision*, *recall*, dan *F1-score* juga ikut naik seiring pengalaman model bertambah.

Sementara itu, *test set* digunakan setelah pelatihan selesai dan bobot terbaik telah diperoleh.

3.2.5 Pengujian Model

Pengujian model dilakukan setelah proses pelatihan selesai untuk mengevaluasi kemampuan YOLOv8 dan YOLOv11 dalam mendeteksi objek pada citra yang tidak digunakan selama proses training. Tahapan ini bertujuan untuk memastikan bahwa model tidak hanya mempelajari data latih, tetapi juga mampu melakukan generalisasi terhadap data baru.

Pengujian dilakukan menggunakan data pada validation set dan test set, dengan fokus utama pada test set sebagai data evaluasi akhir. Model yang telah tersimpan dalam bentuk best weights dijalankan untuk melakukan prediksi terhadap citra uji. Hasil prediksi berupa bounding box, label kelas, serta nilai confidence yang menunjukkan tingkat keyakinan model terhadap objek yang terdeteksi.

Selanjutnya, hasil prediksi dibandingkan dengan ground truth anotasi untuk menghitung metrik evaluasi. Perhitungan metrik dilakukan menggunakan fungsi evaluasi bawaan Ultralytics. Metrik yang digunakan meliputi precision, recall, F1-score, mAP@50, serta mAP@50-95. Metrik-metrik tersebut digunakan sebagai acuan untuk mengukur tingkat akurasi deteksi serta kemampuan model dalam mengenali objek secara konsisten.

Selain evaluasi kuantitatif, pengujian juga mencakup analisis visual terhadap hasil deteksi. Visualisasi bounding box pada citra uji digunakan untuk melihat kesesuaian antara area yang terdeteksi dengan objek sebenarnya. Analisis ini membantu dalam memahami bagaimana model mengenali objek pada berbagai kondisi, seperti perbedaan pencahayaan, ukuran objek, serta tekstur permukaan jalan.

A. Pengujian Model YOLOv8

Pada model YOLOv8, pengujian dilakukan dengan menjalankan best weights terhadap seluruh citra pada *test set*. Model menghasilkan prediksi berupa *bounding box* yang menandai area objek serta nilai *confidence* sebagai indikator keyakinan deteksi.

Hasil prediksi kemudian dibandingkan dengan anotasi *ground truth* untuk menentukan tingkat kesesuaian deteksi. Proses ini melibatkan perhitungan *Intersection over Union* (IoU) sebagai dasar dalam menentukan apakah suatu prediksi termasuk kategori benar atau tidak. Berdasarkan hasil tersebut, dihitung nilai *precision, recall, F1-score*, serta mAP sebagai indikator performa model.

B. Pengujian Model YOLOv11

Pengujian model YOLOv11 dilakukan menggunakan prosedur yang sama dengan YOLOv8, yaitu dengan memanfaatkan best weights untuk melakukan prediksi pada data test set. Model menghasilkan bounding box, label kelas, serta nilai confidence untuk setiap objek yang terdeteksi.

Evaluasi dilakukan dengan membandingkan hasil prediksi terhadap ground truth menggunakan perhitungan IoU. Berdasarkan hasil tersebut, diperoleh nilai precision, recall, F1-score, serta mAP sebagai indikator

performa model. Penggunaan metrik yang sama bertujuan untuk memastikan bahwa hasil evaluasi kedua model dapat dibandingkan secara objektif.

Selain itu, hasil visualisasi bounding box juga digunakan sebagai bagian dari analisis untuk melihat bagaimana model mendeteksi objek pada berbagai kondisi citra. Tahapan ini memberikan gambaran mengenai kemampuan model dalam mengenali objek secara konsisten pada data yang beragam.

3.2.6 Pengujian Implementasi Berbasis Streamlit

Pengujian implementasi berbasis Streamlit dilakukan untuk menguji bagaimana model yang telah dilatih dapat diintegrasikan ke dalam sebuah aplikasi berbasis web sederhana. Tujuan dari tahap ini adalah untuk menyediakan antarmuka interaktif yang memungkinkan pengguna melakukan deteksi objek secara langsung tanpa harus menjalankan kode secara manual.

Pada tahap ini, model yang telah dilatih diintegrasikan ke dalam aplikasi Streamlit dengan memanfaatkan file *best weights*. Aplikasi dirancang untuk menerima input berupa citra yang diunggah oleh pengguna, kemudian sistem akan memproses citra tersebut menggunakan model yang telah dilatih dan menampilkan hasil deteksi dalam bentuk *bounding box* serta label objek.

Proses pengujian dilakukan dengan mengunggah beberapa citra uji ke dalam aplikasi untuk memastikan bahwa sistem dapat berjalan dengan baik. Hasil deteksi yang ditampilkan kemudian diamati untuk melihat kesesuaian antara objek yang terdeteksi dengan kondisi sebenarnya pada citra. Selain itu, pengujian ini juga bertujuan untuk memastikan bahwa integrasi antara model dan antarmuka aplikasi berjalan dengan stabil tanpa kendala teknis.

3.2.7 Pengujian Implementasi Secara *Real-time*

Pengujian implementasi dilakukan sebagai tahap lanjutan setelah pengujian implementasi berbasis streamlit untuk melihat bagaimana kinerja YOLOv8 dan YOLOv11 ketika diterapkan secara langsung pada citra jalan berlubang di luar skenario pengujian terkontrol. Tahapan ini bertujuan

untuk mengevaluasi kemampuan implementatif model dalam kondisi yang lebih mendekati penggunaan nyata, sesuai dengan alur penelitian pada flowchart yang menekankan pengujian berbasis *confidence score* dan hasil deteksi visual.

Pada tahap ini, model YOLOv8 dan YOLOv11 yang telah melalui proses pelatihan dan pengujian menggunakan *best weight* masing-masing dijalankan untuk melakukan inferensi pada citra jalan berlubang yang tidak termasuk ke dalam dataset pelatihan maupun pengujian sebelumnya. Citra uji implementasi mencerminkan variasi kondisi lapangan, seperti perbedaan pencahayaan, sudut pengambilan gambar, tekstur aspal, serta ukuran dan bentuk lubang jalan.

Hasil pengujian implementasi ditampilkan dalam bentuk *bounding box* yang menandai lokasi lubang jalan beserta nilai *confidence score* pada setiap prediksi. *Confidence score* digunakan sebagai indikator tingkat keyakinan model terhadap hasil deteksi yang dihasilkan. Semakin tinggi nilai *confidence*, semakin besar kemungkinan bahwa area yang terdeteksi merupakan lubang jalan yang valid. Pada tahap ini, analisis tidak hanya berfokus pada keberadaan deteksi, tetapi juga pada kestabilan model dalam mempertahankan akurasi prediksi pada berbagai kondisi visual.

Pengujian implementasi ini memberikan gambaran nyata mengenai performa model saat diaplikasikan secara langsung, serta menjadi penghubung antara evaluasi teknis berbasis metrik dan evaluasi fungsional berbasis kebutuhan lapangan. Dengan demikian, hasil pengujian implementasi menjadi dasar penting dalam menganalisis perbedaan performa antara YOLOv8 dan YOLOv11.

3.2.8 Evaluasi Perbandingan

Evaluasi perbandingan dilakukan untuk menjawab rumusan masalah penelitian yang berkaitan dengan perbandingan tingkat akurasi dan performa deteksi antara model YOLOv8 dan YOLOv11. Tahapan ini merupakan tahap akhir dalam alur penelitian sesuai dengan flowchart, yang mengintegrasikan hasil pengujian model dan pengujian implementasi ke dalam analisis komparatif yang menyeluruh.

Evaluasi dilakukan dengan menggabungkan hasil pengujian kuantitatif berbasis metrik deteksi objek dan hasil pengujian implementasi berbasis pengamatan visual serta *confidence score*. Dengan pendekatan ini, perbandingan performa tidak hanya didasarkan pada nilai numerik, tetapi juga pada perilaku model saat diterapkan dalam kondisi nyata.

1. Evaluasi Perbandingan Model

   Evaluasi perbandingan model dilakukan dengan membandingkan hasil pengujian YOLOv8 dan YOLOv11 berdasarkan metrik evaluasi yang telah diperoleh pada tahap pengujian model, yaitu *precision*, *recall*, *F1-score*, mAP@50, dan mAP@50–95. Metrik tersebut digunakan untuk menilai kemampuan masing-masing model dalam mendeteksi lubang jalan secara akurat dan konsisten.

   Precision digunakan untuk menilai tingkat ketepatan deteksi model dalam menghindari kesalahan prediksi, sedangkan *recall* mengukur kemampuan model dalam menemukan seluruh objek lubang jalan yang ada pada citra. *F1-score* digunakan sebagai indikator keseimbangan antara *precision* dan *recall*, sehingga memberikan gambaran performa umum model secara objektif. Sementara itu, mAP@50 dan mAP@50–95 menjadi indikator utama dalam menilai kualitas deteksi *bounding box* pada berbagai tingkat ambang *Intersection over Union* (IoU).

   Berdasarkan hasil evaluasi, model dengan nilai mAP dan *F1-score* yang lebih tinggi dapat dinilai memiliki performa deteksi yang lebih baik secara keseluruhan. Apabila YOLOv11 menunjukkan peningkatan nilai mAP dibandingkan YOLOv8, maka hal tersebut mengindikasikan bahwa pembaruan arsitektur dan mekanisme ekstraksi fitur pada YOLOv11 memberikan dampak positif terhadap akurasi dan stabilitas deteksi kondisi jalan.

2. Evaluasi Perbandingan Implementasi

   Evaluasi perbandingan implementasi dilakukan untuk membandingkan kinerja YOLOv8 dan YOLOv11 saat diterapkan secara langsung pada citra uji implementasi. Proses evaluasi ini bertujuan

untuk mengetahui perbedaan performa kedua model dalam kondisi penggunaan nyata di luar dataset pelatihan dan pengujian.

Analisis dilakukan berdasarkan beberapa aspek utama, yaitu hasil visual deteksi, kestabilan *bounding box* dalam mengidentifikasi objek, serta nilai confidence score yang dihasilkan oleh masing-masing model. Hasil visual digunakan untuk mengamati kesesuaian area deteksi terhadap objek sebenarnya, sedangkan confidence score digunakan untuk melihat tingkat keyakinan model terhadap prediksi yang dihasilkan.

Pengujian dilakukan pada berbagai kondisi citra, seperti variasi pencahayaan, tekstur permukaan jalan, sudut pengambilan gambar, serta ukuran objek yang berbeda. Dengan demikian, evaluasi ini dapat memberikan gambaran mengenai kemampuan masing-masing model dalam menghadapi variasi kondisi lapangan.

Hasil dari evaluasi perbandingan implementasi ini selanjutnya digunakan sebagai dasar dalam analisis performa model pada tahap pembahasan untuk menentukan model yang memiliki kinerja paling optimal.