

SISTEM DETEKSI JALAN BERLUBANG SECARA REAL-TIME MENGGUNAKAN YOLOV11: INTEGRASI DATA DAN LOKASI MELALUI WEBSITE (STUDI KASUS : DAERAH KARAWANG)

Dzagar Raya Ismail, Reni Rahmadewi

Teknik Elektro, Universitas Singaperbangsa Karawang

HS.Ronggo Waluyo, Puseurjaya, Kec. Telukjambe Tim., Kabupaten Karawang, Jawa Barat 41361.

2110631160080@student.unsika.ac.id

ABSTRAK

Kerusakan jalan berlubang di daerah Karawang berdampak pada keselamatan pengguna jalan dan efisiensi transportasi. Identifikasi jalan berlubang masih dilakukan secara manual, sehingga sering tidak akurat dan memerlukan waktu lama. Oleh karena itu, penelitian ini mengembangkan sistem deteksi jalan berlubang berbasis YOLOv11 yang terintegrasi dengan platform web untuk mendeteksi jalan berlubang secara *real-time*. Metode yang digunakan mencakup pengumpulan dataset dari Roboflow, pelatihan model menggunakan Google Colaboratory, serta implementasi sistem berbasis web yang mengintegrasikan *frontend*, *backend*, dan *database* JSON sebagai penyimpanan data. Sistem memungkinkan pengguna mengaktifkan kamera untuk menangkap jalan yang diproses oleh YOLOv11 guna mengidentifikasi keberadaan jalan berlubang. Hasil deteksi berupa gambar dengan *bounding box*, lokasi GPS, dan waktu kejadian akan disimpan dalam format JSON dan ditampilkan di halaman web. Pengujian menunjukkan model memiliki akurasi 83.5%, presisi 89.6%, dan *recall* 92.4%, yang menunjukkan performa cukup baik dalam mengenali jalan berlubang. Sistem ini diharapkan dapat meningkatkan efektivitas pemantauan kondisi jalan dan menjadi pendukung keputusan dalam perbaikan infrastruktur secara lebih cepat dan akurat.

Kata kunci : YOLOv11, deteksi jalan berlubang, web-based system, computer vision, JSON storage.

1. PENDAHULUAN

Jalan merupakan infrastruktur yang sangat penting untuk mendukung pergerakan pengguna kendaraan motor maupun mobil, jalan yang baik merupakan salah satu faktor yang mempengaruhi keselamatan pengguna jalan. Di Indonesia, termasuk di daerah Karawang, banyak sekali jalan yang mengalami kerusakan, seperti jalan berlubang, yang dapat menyebabkan kecelakaan kepada pengguna kendaraan dan juga membuat kerugian ekonomi.

Menurut laporan dari ONTV.CO.ID, intensitas hujan yang tinggi di awal tahun 2025 telah menyebabkan kerusakan infrastruktur jalan di Kabupaten Karawang, Jawa Barat. Di Jl. Raya Klari arah Cikampek, banyak terdapat lubang dan genangan air yang mengganggu kelancaran lalu lintas. Banyak pengendara mengeluhkan kondisi jalan yang berbahaya, terutama saat hujan deras, di mana lubang-lubang tersebut tidak terlihat dan dapat menyebabkan kecelakaan, sehingga meminta pemerintah setempat untuk segera memperbaikinya [1].

Dalam era digital ini, sudah terdapat teknologi kecerdasan buatan (*Artificial Intelligence/ AI*) dan visi komputer yang telah memberikan solusi yang inovatif untuk permasalahan ini. Salah satu algoritma yang sudah banyak digunakan dalam berbagai deteksi objek yaitu YOLO (*You Only Look Once*), yang sudah terkenal karena kemampuannya untuk melakukan deteksi objek secara *real-time* dengan tingkat akurasi yang tinggi [2].

YOLOv11, sebagai versi terbaru dari algoritma ini, menawarkan peningkatan terutama dalam hal

kecepatan dan akurasi deteksi, menjadikannya pilihan yang paling ideal untuk website *real-time* deteksi jalan berlubang.

Penelitian ini bertujuan untuk merancang sebuah sistem pendeteksian jalan berlubang secara *real-time* dengan memanfaatkan algoritma YOLOv11 yang terhubung dengan platform web. Sistem ini memungkinkan pengguna, termasuk pihak berwenang, untuk mengunggah gambar jalan yang diduga berlubang dan dapat berpotensi membahayakan kendaraan. Setelah diunggah, sistem akan menganalisis gambar menggunakan model YOLOv11 untuk memastikan keberadaan lubang pada jalan. Hasil deteksi yang diperoleh mencakup sejumlah gambar jalan berlubang, lokasi berdasarkan peta, serta tanggal dan waktu kejadian, sehingga memudahkan akses informasi bagi pihak berwenang.

Studi kasus ini difokuskan khusus daerah Karawang, sebuah daerah yang sering menghadapi permasalahan infrastruktur jalan, termasuk banyaknya jalan berlubang yang berisiko bagi pengguna kendaraan. Dengan adanya sistem ini, diharapkan respons terhadap perbaikan jalan berlubang di Karawang menjadi lebih cepat dan efisien. Selain itu, data yang dikumpulkan dapat digunakan untuk perencanaan pemeliharaan jalan jangka panjang, sehingga mendukung upaya pemerintah dalam meningkatkan kualitas infrastruktur. Melalui integrasi teknologi dan informasi lokasi yang akurat, sistem ini diharapkan dapat membantu dalam pengambilan keputusan yang lebih baik terutama menangani masalah jalan berlubang.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang berarti dalam pengembangan teknologi deteksi jalan berlubang dan meningkatkan kualitas infrastruktur jalan di daerah Karawang, serta menjadi model bagi daerah lain yang menghadapi masalah serupa.

2. TINJAUAN PUSTAKA

2.1. Definisi dan Permasalahan Jalan Berlubang

Jalan berlubang merupakan salah satu bentuk kerusakan pada jalan raya yang terjadi akibat faktor alam maupun aktivitas manusia. Lubang pada permukaan jalan terbentuk dikarenakan cuaca ekstrem, genangan air, beban kendaraan yang terlalu berlebih, serta kualitas pada konstruksi yang kurang baik. Kerusakan ini sangat mengganggu kenyamanan berkendara selain itu dapat menyebabkan terjadinya kecelakaan lalu lintas dan meningkatkan biaya perawatan kendaraan.

Kerusakan infrastruktur jalan banyak dijumpai di sekitar lingkungan kita, seperti jalan pemukiman dan jalur desa, serta jalan umum, terjadi akibat dilaluinya beban gandar ringan maupun berat. Adapun dampak-dampak dari kerusakan infrastruktur jalan yaitu meliputi adanya kecelakaan kendaraan, adanya ketidak nyamanan pengendara, terganggunya pada perekonomian masyarakat dan dampak negatif lainnya [3].

Oleh karena itu dibutuhkan sistem pemantauan yang lebih efektif untuk mengidentifikasi dan melaporkan jalan berlubang secara *real-time*.

2.2. Teknologi Kecerdasan Buatan dalam Deteksi Jalan Berlubang

Artificial Intelligence (AI) adalah kecerdasan buatan yang merupakan wujud pemodelan dari kecerdasan manusia yang diterapkan dalam suatu mesin untuk pembuatan mesin cerdas [4].

Salah satu cabang AI yang banyak digunakan dalam pemrosesan visual adalah visi komputer (*Computer Vision*), merupakan ilmu yang menggunakan *image processing* untuk membuat sebuah keputusan berdasarkan citra yang didapat dari sensor seperti kamera [5].

Dengan kata lain, computer vision bertujuan untuk membangun sebuah mesin pintar yang dapat melihat. Kerangka kerja umum yang biasa dilakukan dalam *computer vision* adalah: *image acquisition* (akuisisi citra), *pre-processing* (pra pemrosesan), *feature extraction* (ekstraksi fitur), deteksi/segmentasi, pemrosesan tingkat tinggi, dan pengambilan keputusan [6].

Sehingga computer vision dapat memungkinkan sistem untuk memahami berbagai informasi dari gambar atau video.

Dengan perkembangan teknologi AI dan visi komputer saat ini menjadikan sebuah bantuan dalam deteksi jalan berlubang yang lebih akurat dan efisien. Sistem ini juga memiliki berpotensi menjadi solusi efektif dalam pemeliharaan khususnya

infrastruktur jalan, terutama di daerah seperti Karawang, dimana jalan berlubang sering menjadi permasalahan utama bagi pengguna kendaraan.

2.3. Algoritma YOLO dalam Deteksi Objek

YOLO (*You Only Look Once*) merupakan algoritma *deep learning* untuk melakukan deteksi objek secara *real time* dengan memanfaatkan algoritma *Convolutional Neural Network* (CNN) untuk memproses data visual. YOLO mendeteksi sebuah objek dengan menerima input gambar kemudian dibagi menjadi kotak dengan ukuran $S \times S$ untuk membuat bounding box dan sistem memprediksi probabilitas kelas berdasarkan tingkat keyakinan dalam mendeteksi sebuah objek [7].

Dalam penelitian ini, YOLOv11 dipilih sebagai algoritma utama dikarenakan memiliki akurasi yang lebih tinggi dalam mendeteksi objek kecil, termasuk lubang pada jalan, yang sering kali sulit terdeteksi oleh model sebelumnya. Selain itu, YOLOv11 menawarkan kecepatan inferensi real-time, memungkinkan deteksi dilakukan dalam hitungan milidetik, sehingga sangat ideal untuk aplikasi berbasis video dan kamera mobil [8].

Algoritma ini juga mampu bekerja dengan baik dalam berbagai macam kondisi lingkungan, seperti pencahayaan rendah atau refleksi cahaya, yang sering menjadi tantangan dalam deteksi jalan berlubang. Kemudahan integrasinya dengan sistem berbasis web memungkinkan hasil deteksi langsung diunggah ke platform pemantauan, mempermudah pihak berwenang dalam mengambil tindakan cepat terhadap perbaikan infrastruktur jalan.

2.4. Implementasi Sistem Berbasis Web untuk Deteksi Jalan Berlubang

Dalam era digital saat ini, sistem berbasis web telah menjadi solusi efektif untuk pemantauan infrastruktur jalan, termasuk dalam deteksi jalan berlubang. Dengan memanfaatkan teknologi *Artificial Intelligence* (AI) dan visi komputer (*Computer Vision*), sistem ini memiliki kemampuan mengidentifikasi kerusakan jalan secara *real-time* dan menyajikan informasi kepada pihak berwenang maupun masyarakat umum melalui platform berbasis web.

Sistem ini bekerja dengan cara mengarahkan kamera ke jalan-jalan berlubang oleh pengguna ketika menggunakan web tersebut secara langsung. Data tersebut kemudian diproses oleh model YOLOv11, yang mampu melakukan deteksi keberadaan jalan berlubang dengan akurat dan cepat. Hasil dari deteksi ini mencakup gambar jalan yang telah dianalisis, lokasi berbasis Gmaps, serta waktu kejadian ketika melakukan deteksi, yang kemudian ditampilkan dalam antarmuka web agar dapat diakses oleh pengguna yang berkepentingan, seperti pemerintah daerah dan dinas perhubungan.

Beberapa contoh implementasi yang berkaitan dengan sistem berbasis web dalam pemantauan jalan,

sudah banyak diterapkan di berbagai negara. Misalnya, *Google Street View* AI digunakan untuk menganalisis atau memantau kondisi jalan di beberapa kota besar, sedangkan di India, terdapat proyek *RoadCare* telah memanfaatkan AI untuk mendeteksi objek khususnya kerusakan jalan dengan kamera ponsel yang terhubung ke aplikasi berbasis web. Konsep serupa dapat diterapkan dalam penelitian ini dengan menyesuaikan pada kondisi infrastruktur jalan di Indonesia, khususnya di Kabupaten Karawang.

2.5. Penelitian Terdahulu dalam Deteksi Jalan Berlubang

Terdapat beberapa penelitian terdahulu yang telah dilakukan untuk mendeteksi jalan berlubang menggunakan teknologi berbasis computer vision dan AI. Penelitian yang dilakukan oleh (Riau *et al.*, 2024) menggunakan YOLOv5 berbasis drone yang digunakan untuk mendeteksi jalan berlubang secara otomatis dan memperoleh hasil yang menjanjikan dengan keyakinan F1 sebesar 0,83, keyakinan Presisi sebesar 0,96, Presisi-Recall sebesar 0,716, dan keyakinan Recall sebesar 0,8 [9].

Sementara itu, penelitian oleh (Habibi & Wibowo, 2023) mengembangkan aplikasi Android yang memungkinkan pengguna melaporkan kerusakan jalan dengan gambar yang diunggah melalui aplikasi [10].

Penelitian ini mengadopsi konsep serupa, tetapi dengan pendekatan yang lebih komprehensif, yaitu dengan menerapkan YOLOv11 berbasis web untuk mendeteksi jalan berlubang secara real-time, menyimpan hasil deteksi ke dalam JSON Storage, serta menampilkan informasi berupa gambar hasil deteksi, lokasi GPS, dan waktu kejadian pada platform web. Dengan integrasi ini, sistem dapat menjadi alat yang lebih efektif, cepat, dan mudah diakses untuk pemantauan infrastruktur jalan, khususnya di Kabupaten Karawang.

3. METODE PENELITIAN

Pada bagian ini menjelaskan metode pengembangan yang akan digunakan oleh sistem deteksi jalan berlubang berbasis web dengan integrasi algoritma YOLOv11. Sistem ini dirancang agar dapat diakses oleh pengguna umum dan pihak berwenang, dengan fitur utama seperti pengunggahan gambar secara *real time*, deteksi otomatis menggunakan AI, serta penyimpanan dan penyajian data dalam sistem berbasis web.

3.1. Rancangan Penelitian

Untuk memastikan agar sistem dapat berjalan dengan efisien, penelitian ini perlu melakukan pendekatan *Full-Stack Web Development* dengan terintegrasi AI untuk visi komputer. Rancangan sistem ini terdiri dari dua komponen utama yaitu:

- a. *Frontend* (Tampilan Pengguna UI/UX)
Frontend merupakan bagian dari *website* yang bertujuan memberikan interaksi langsung dengan

pengguna, mencakup elemen visual seperti tata letak, tobo, warna, dan navigasi, yang dikembangkan oleh HTML, CSS, dan Javascript [11].

UI (*User Interface*) memiliki tujuan yaitu fokus terhadap desain tampilan yang menarik dan fungsional, sedangkan untuk UX (*User Experience*) memastikan bahwa navigasi dapat memudahkan pengguna dalam berinteraksi dengan sistem [11].

Pada penelitian ini dimana sistem menggunakan Bootstrap untuk antarmuka interaktif dan terdiri hanya dua halaman utama yang terhubung untuk mendeteksi jalan berlubang. Halaman pertama memungkinkan pengguna dapat melakukan rekaman real time melalui kamera perangkat serta sistem akan secara otomatis mengambil lokasi menggunakan *Geolocation* API dan waktu melakukan pendeteksiannya. Pada halaman kedua terdapat hasil dari jalan yang terdeteksi berlubang dalam bentuk gambar yang telah di *bounding box* berserta gmaps dan waktu kejadiannya.

- b. *Backend* (Pemrosesan data dan penyimpanan)
Backend merupakan bagian dari sistem yang memiliki tanggung jawab dalam mengelola logika bisnis, pemrosesan data, dan komunikasi dengan database, bekerja di *server-side* untuk mengatasi berbagai permintaan dari frontend dan mengirimkan hasil kembali ke pengguna [12].
Backend sistem yang dikembangkan pada penelitian ini menggunakan Python (Flask) untuk menjalankan YOLOv11 dalam melakukan deteksi jalan berlubang secara *real time*. Hasil dari deteksi termasuk gambar jalan berlubang, kelas objek, lokasi GPS yang bisa diakses melalui Gmaps, dan waktu kejadian, disimpan dalam database berbasis JSON, yang mempermudah akses dan pengelolaan data. Selain itu, backend juga terintegrasi dengan PHP, yang memiliki fungsi sebagai penghubung antara frontend dan sistem AI, untuk memastikan bahwa komunikasi dapat berlangsung secara efisien terutama dalam pemrosesan dan tampilan hasil deteksi di antarmuka pengguna.

3.2. Arsitektur Sistem

Arsitektur sistem yang akan dirancang dalam penelitian ini menggunakan sebuah pendekatan *Client-Server* yang terdiri dari tiga lapisan utama, yaitu Frontend (Client), Backend (Server), dan Database (JSON Storage). Sistem ini memungkinkan pengguna untuk melakukan pengunggahan gambar jalan berlubang, yang kemudian akan diproses oleh YOLOv11 di backend, dan hasilnya dapat langsung disimpan dalam JSON database. Informasi yang tersimpan yaitu lokasi GPS, gambar, waktu deteksi, dan klasifikasi objek.

Komponen Arsitektur Sistem:

- a. *Frontend (Client Side)*
Dibangun menggunakan beberapa bahasa pemrograman yaitu HTML, CSS, dan Javascript

sebagai antarmuka pengguna, dimana memungkinkan pengguna untuk dapat mengunggah hasil akhir deteksi yang berbentuk gambar dari hasil deteksi menggunakan kamera rekaman dan mengakses informasi deteksi.

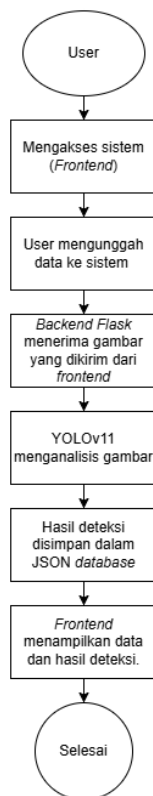
b. Backend (Processing Server)

Menggunakan flask dari python sebagai API server yang dapat mengatasi berbagai permintaan dari frontend, dimana menerapkan sebuah algoritma YOLOv11 untuk mendeteksi objek dari jalan berlubang yang direkam secara *real time* oleh pengguna dan dapat mengelola penyimpanan gambar dan metada dalam JSON database. Semua itu terintegrasi dengan PHP yang memiliki fungsi penghubung antara frontend dan backend.

c. Database (JSON Storage)

JSON Storage merupakan sebuah metode untuk menyimpan data dalam format JavaScript Object Notation (JSON) yang biasanya digunakan sebagai alternatif database relasional [13].

JSON ini bersifat ringan, fleksibel, dan mudah dibaca, sehingga lebih banyak digunakan dalam sistem berbasis web dan API. Dalam penelitian ini sistem pada deteksi jalan berlubang, JSON digunakan sebagai database utama dalam menyimpan lokasi, gambar hasil deteksi melalui rekaman, waktu kejadian, dan hasil klasifikasi YOLOv11. Penyimpanan pada JSON ini memungkinkan akses data yang lebih cepat dan efisien tanpa memerlukan database relasional, sehingga mempermudah dalam pengelolaan hasil dari deteksi dalam website.

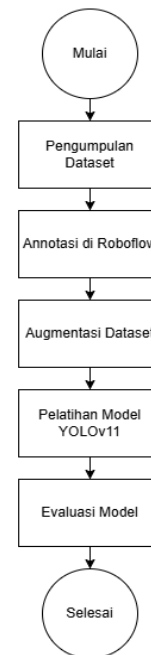


Gambar 1. Flowchart arsitektur sistem

Pada gambar 1, menampilkan alur kerja dari arsitektur sistem yang dimana terdapat tahap-tahap proses dalam mengembangkan sistem.

3.3. Pemrosesan Data dengan YOLOv11

Proses untuk penelitian ini terutama dalam deteksi jalan berlubang dilakukan dengan memanfaatkan Roboflow sebagai platform untuk melakukan pengolahan dataset dan pelatihan untuk model YOLOv11. Berikut adalah tahapan utama dalam pemrosesan data:

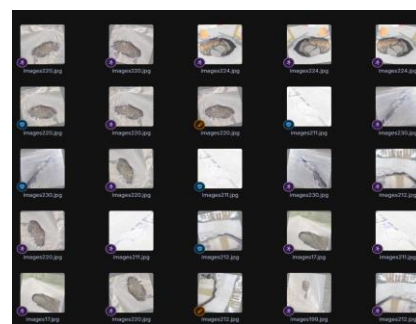


Gambar 2. Flowchart YOLOv11

Seperti yang ditampilkan pada gambar 2, merupakan alur kerja dari pemrosesan data untuk YOLO agar bisa digunakan ketika melakukan uji coba dalam deteksi objek khususnya pada penelitian ini adalah jalan berlubang.

a. Pengumpulan Dataset

Dataset yang digunakan untuk penelitian ini didapat dari Roboflow Universe Datasets, yang memiliki 515 gambar jalan berlubang. Dataset ini hanya dibuat untuk satu kelas, yaitu jalan berlubang, sehingga memungkinkan model untuk lebih fokus dalam mendeteksi detail karakteristik jalan berlubang secara lebih akurat.



Gamabr 3. Pengumpulan dataset

Data yang dikumpulkan pada gambar 3, digunakan sebagai proses yang akan dipelajari oleh YOLO, pada penelitian ini penulis melakukan pengumpulan dataset melalui platform Roboflow yang dimana mempermudah dalam proses pengerjaannya.

b. Anotasi Dataset

Setelah pengumpulan dataset, langkah berikutnya dilakukan proses membuat anotasi menggunakan Roboflow. Anotasi ini dilakukan dengan cara memberikan label pada setiap gambar jalan berlubang.



Gambar 4. Anotasi dataset

Proses dalam melakukan anotasi seperti gambar 4, melibatkan penandaan pada sebuah area di jalan berlubang menggunakan *bounding box* untuk memastikan bahwa setiap gambar yang berada dalam dataset memiliki informasi yang jelas untuk melakukan pelatihan, setiap gambar yang telah di *bounding box* di berikan *class* yaitu "Jalan Berlubang".

c. Pelatihan Model YOLOv11

Model YOLO yang digunakan dalam penelitian ini adalah YOLOv11, yang difokuskan untuk melakukan deteksi sebuah jalan berlubang dengan satu kelas utama, yaitu "Jalan Berlubang". Dataset yang digunakan terdiri dari 515 gambar, yang sudah diperoleh dari Roboflow Universe Datasets. Model ini memanfaatkan *transfer learning*, di mana bobot awal telah dilatih menggunakan dataset besar sebelumnya untuk meningkatkan akurasi dan efisiensi dalam mendeteksi jalan berlubang.

Tabel 1. Parameter training

Parameter	Nilai	Keterangan
Model	YOLOv11	Model YOLO versi 11 skala kecil
Dataset	Roboflow Universe	Dataset berisi 515 gambar
Epochs	100	Jumlah iterasi training
Batch size	32	Ukuran batch untuk efisiensi training
Learning rate	0.001	Kecepatan pembelajaran

Parameter	Nilai	Keterangan
Optimizer	AdamW	Optimizer untuk stabilitas training
Image size	800 x 800	Resolusi gambar saat training
Augmentasi	Aktif (mosaic, flip)	Augmentasi untuk memperkaya data

Tabel 1 di atas merangkum parameter yang digunakan dalam proses pelatihan model YOLOv11 untuk mendeteksi jalan berlubang. Model yang digunakan adalah YOLOv11s, versi skala kecil dari YOLOv11, yang dioptimalkan untuk efisiensi dan kecepatan. Dataset yang digunakan berasal dari Roboflow Universe, terdiri dari 515 gambar yang telah diberi label.

d. Evaluasi Model

Evaluasi model dilakukan dengan menggunakan beberapa metrik utama, yaitu Presisi (*Precision*), Recall, *mean Average Precision (mAP)*, dan *F1-Score*. Selain itu, *confusion matrix* digunakan untuk memberikan gambaran lebih rinci tentang kinerja deteksi model. Berikut adalah penjelasan dari masing-masing metrik evaluasi:

Presisi mengukur seberapa akurat model dalam mengidentifikasi jalan berlubang dengan benar. Nilai presisi dihitung sebagai:

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

di mana TP (*True positive*) adalah jumlah jalan berlubang yang berhasil dideteksi dengan benar, dan FP (*False positive*) adalah jumlah objek yang salah diklasifikasikan sebagai jalan berlubang.

Recall mengukur seberapa baik model dalam menemukan semua jalan berlubang yang ada dalam dataset. Rumus recall adalah:

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

di mana FN (*False negative*) adalah jumlah jalan berlubang yang tidak terdeteksi oleh model.

mAP mengukur rata-rata akurasi deteksi model untuk berbagai nilai IoU (*Intersection over Union*). Semakin tinggi nilai *mAP*, semakin baik model dalam mengenali jalan berlubang dengan akurat di berbagai skenario pencahayaan dan sudut gambar. *F1-Score* adalah ukuran keseimbangan antara presisi dan recall, yang dihitung dengan rumus:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Nilai *F1-Score* yang tinggi menunjukkan keseimbangan yang baik antara mendeteksi jalan berlubang dengan benar dan menghindari kesalahan deteksi.

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi performa suatu model atau

algoritma dalam mengklasifikasikan data. Selain itu, terdapat beberapa metrik evaluasi tambahan, seperti *Box Loss*, yang mengukur sejauh mana model mampu menentukan posisi pusat objek dan memprediksi bounding box dengan akurat. *Objectness Loss* menunjukkan probabilitas keberadaan suatu objek dalam wilayah yang diprediksi oleh model, sedangkan *Classification Loss* menggambarkan tingkat kesalahan model dalam mengklasifikasikan suatu objek ke dalam kategori yang sesuai.

3.4. Implementasi Sistem Berbasis Web

Sistem deteksi jalan berlubang berbasis YOLOv11 diimplementasikan dalam sebuah platform web yang memungkinkan pengguna untuk mengunggah gambar jalan, melakukan deteksi secara *real time*, serta melihat hasil analisis yang mencakup lokasi dan tingkat keparahan jalan berlubang. Implementasi ini terdiri dari tiga komponen utama, yaitu Frontend, Backend, dan Database JSON yang saling terhubung untuk memastikan kelancaran proses deteksi dan penyimpanan data.

3.4.1. Arsitektur Sistem Web

Sistem ini dirancang menggunakan pendekatan client-server, dengan struktur sebagai berikut:

- Frontend (*Client side*): Dibangun menggunakan HTML, CSS, dan JavaScript (PHP) untuk memberikan tampilan interaktif kepada pengguna.
- Backend (Server Side): Menggunakan Flask (Python) untuk menangani pemrosesan data, mengintegrasikan YOLOv11, serta menyimpan hasil deteksi ke dalam JSON database.
- Database (JSON Storage): Berfungsi sebagai tempat penyimpanan hasil deteksi, termasuk gambar, informasi lokasi, dan waktu deteksi.

3.4.2. Alur Kerja Sistem

- Pengguna mengunggah gambar jalan melalui antarmuka web.
- Gambar dikirim ke backend Flask, yang memprosesnya menggunakan model YOLOv11.
- Jika jalan berlubang terdeteksi, hasilnya disimpan dalam database JSON bersama metadata seperti lokasi GPS dan timestamp.
- Hasil deteksi ditampilkan di halaman web, sehingga pengguna dapat melihat kondisi jalan secara *real-time*.

3.4.3. Teknologi yang Digunakan

Tabel 3. Komponen teknologi yang digunakan

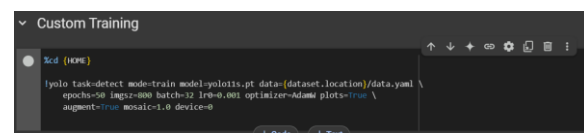
Komponen	Teknologi yang Digunakan
Frontend	JavaScripts, HTML, CSS
Backend	PHP, Python (Flask), YOLOv11
Database	JSON Storage
Web Server	XAMPP (untuk pengujian)

4. HASIL DAN PEMBAHASAN

Pada bagian ini membahas hasil penelitian yang telah dilakukan, termasuk proses pelatihan dan pengujian model deteksi lubang jalan berbasis YOLOv11. Analisis dilakukan berdasarkan hasil prediksi model terhadap dataset yang digunakan serta melakukan implementasi berbasis web.

4.1. Training Dataset

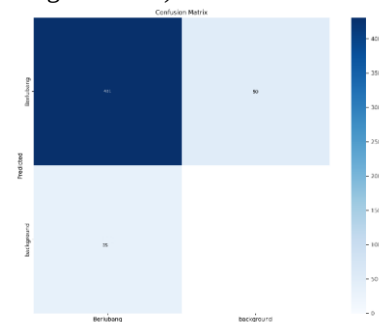
Melakukan training dataset yang telah dibuat pada roboflow yang dilakukan oleh Google Colaboratory dalam Bahasa pemrograman python, training ini menggunakan parameter pada tabel 2. Parameter training, yang diharapkan dapat mengurangi false positive dan false negative, meningkatkan precision, recall, dan mAP, serta lebih optimal untuk dapat digunakan dalam aplikasi deteksi jalan berlubang secara *real-time*.



Gambar 4. Training dataset

Kemudian setelah melakukan training dataset dengan code seperti gambar 4, dilanjutkan running untuk confusion matrix, confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan hasil prediksi dari model dengan label sebenarnya, matriks ini berisi jumlah prediksi benar dan salah untuk setiap kelasnya, yang dimana dapat digunakan untuk menghitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Confusion matrix terdiri dari empat elemen utama yaitu:

- True positive* (TP): Prediksi positif yang benar (model mendeteksi jalan berlubang dan benar adanya).
- False positive* (FP): Prediksi positif yang salah (model mendeteksi jalan berlubang tetapi sebenarnya tidak).
- False negative* (FN): Prediksi negatif yang salah (model tidak mendeteksi jalan berlubang padahal ada).
- True negative* (TN): Prediksi negatif yang benar (model tidak mendeteksi jalan berlubang dan memang tidak ada).



Gambar 5. Confusion matrix

Berdasarkan gambar 5 confusion matrix yang ditampilkan, berikut analisisnya:

- True positive* (TP) = 431. Model berhasil mendeteksi 431 gambar jalan berlubang dengan benar.
- False positive* (FP) = 50. Model salah mendeteksi 50 gambar sebagai jalan berlubang, padahal seharusnya bukan.
- False negative* (FN) = 35. Model gagal mendeteksi 35 gambar jalan berlubang, sehingga diklasifikasikan sebagai background.
- True negative* (TN) = 0. Model berhasil mendeteksi 0 gambar background dengan benar.

Evaluasi kinerja model dari confusion matrix tersebut, dapat di hitung beberapa matrik evaluasi sebagai berikut:

- Akurasi (*Accuracy*)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{431 + 0}{431 + 0 + 50 + 35}$$

$$Accuracy = \frac{431}{516} = 0.835$$

- Presisi (*Precision*)

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

$$Precision = \frac{431}{431 + 50} = 0.896$$

- Recall*

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

$$Recall = \frac{431}{431 + 35} = 0.924$$

- F1-Score*

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

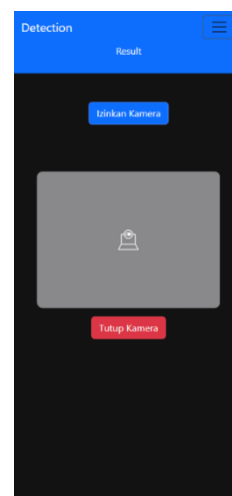
$$F1 = 2 \times \frac{0.896 \times 0.924}{0.896 + 0.924} = 0.91$$

Berdasarkan hasil evaluasi confusion matrix, model deteksi jalan berlubang menunjukkan kinerja yang cukup baik dengan akurasi sebesar 83.5%, yang berarti sebagian besar prediksi model sudah benar. Presisi sebesar 89.6% menunjukkan bahwa model jarang salah mengklasifikasikan *background* sebagai berlubang, sementara recall sebesar 92.4% mengindikasikan bahwa model cukup andal dalam menangkap hampir semua jalan berlubang yang ada. Dengan F1-score sebesar 91%, model memiliki keseimbangan yang baik antara presisi dan recall, menjadikannya cukup andal dalam deteksi. Namun, masih terdapat *False Positive* (FP) dan *False Negative* (FN) yang bisa diminimalkan lebih lanjut dengan

teknik *fine-tuning* model, penyesuaian threshold, atau peningkatan kualitas dataset.

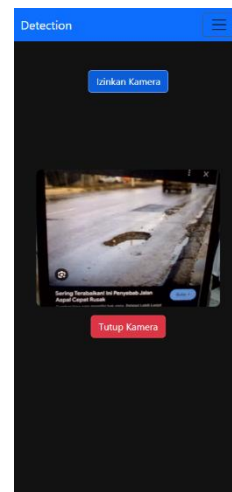
4.2. Hasil Pengujian dengan Web

Pada tahap ini dimana megabungkan antara model deteksi jalan berlubang ke dalam aplikasi berbasis web. Pengujian ini bertujuan untuk mengevaluasi kinerja sistem dalam proses mendeteksi jalan berlubang secara real-time melalui antarmuka pengguna. Proses ini juga mencakup penggabungan model YOLOv11 yang telah dilatih dengan *backend* web menggunakan bahasa pemrograman seperti python serta frontend berbasis HTML, CSS, dan JavaScript. Selain itu, dilakukan pengujian terhadap fitur-fitur utama seperti akses kamera, pemrosesan gambar, serta tampilan hasil dari deteksi untuk memastikan model ini dapat bekerja dengan baik dalam kondisi nyata.



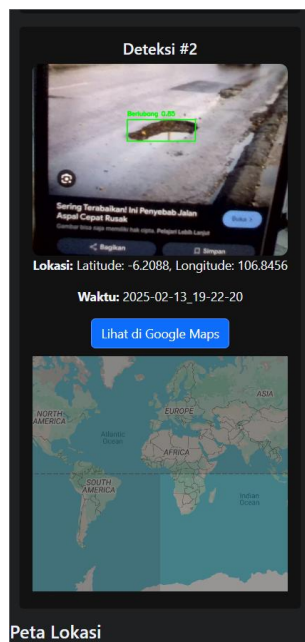
Gambar 6. Tampilan Web pada Handphone

Website ini memiliki menu result dan fitur untuk mengaktifkan kamera, yang memerlukan izin akses dari perangkat pengguna yang bisa dilihat pada gambar 6. Selain itu, izin akses lokasi juga diperlukan agar sistem dapat berfungsi dengan optimal dan memungkinkan pengguna untuk menggunakan web dengan baik.



Gambar 7. Uji coba web

Setelah kamera diaktifkan, pengguna dapat langsung mengarahkan perangkat ke area jalan yang diduga berlubang secara *real time*, dapat dilihat pada gambar 7. Sistem kemudian akan memproses gambar menggunakan YOLOv11 untuk memastikan apakah area yang ditunjukkan benar merupakan jalan berlubang. Jika terdeteksi sebagai jalan berlubang, data tersebut akan disimpan dalam format JSON bersama dengan informasi lokasi dan waktu, lalu hasilnya akan ditampilkan pada menu Result.



Gambar 8. Menu result

Menu Result dapat dilihat tampilannya pada gambar 8, dimana menampilkan hasil deteksi jalan berlubang yang telah diverifikasi secara *real-time*. Hasil ini mencakup gambar dengan *bounding box* pada area berlubang, serta informasi lokasi dan waktu deteksi. Selain itu, pengguna juga dapat langsung mengakses lokasi tersebut melalui Google Maps untuk melihat titik deteksi secara lebih akurat.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sistem deteksi jalan berlubang berbasis YOLOv11 yang terintegrasi dengan website untuk mendeteksi dan mencatat lokasi jalan berlubang secara *real-time*. Hasil pengujian menunjukkan bahwa model memiliki akurasi sebesar 83.5%, dengan presisi 89.6% dan *recall* 92.4%, yang menunjukkan kinerja yang cukup baik dalam mengenali jalan berlubang. Sistem berbasis web ini memungkinkan pengguna untuk mengunggah gambar, memproses deteksi secara otomatis, dan menyimpan hasil dalam format JSON, serta menampilkan lokasi deteksi melalui Google Maps. Namun, masih terdapat beberapa false positive dan false negative, sehingga penelitian selanjutnya dapat berfokus pada peningkatan akurasi model, penyesuaian parameter deteksi, serta penggunaan

dataset yang lebih luas dan bervariasi untuk meningkatkan performa dalam berbagai kondisi lingkungan dan pencahayaan.

DAFTAR PUSTAKA

- [1] G. I. Nusantara, "Jalan Berlubang Akibat Kemacetan dan Beresiko Kecelakaan Bagi Pengguna Jalan," ontv.co.id.
- [2] D. Iskandar Mulyana and M. A. Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," *J. Pendidik. Tambusai*, vol. 6, no. 3, pp. 13971–13982, 2022, doi: 10.31004/jptam.v6i3.4825.
- [3] I. G. A. Istri Lestari, I. G. Angga Diputera, I. K. D. Kubon Tubuh, and A. S. Jiman, "Analisis Penyebab dan Dampaknya Kerusakan Infrastruktur Jalan Terhadap Para Pengguna Jalan dan Masyarakat Sekitar," *J. Ilm. Kurva Tek.*, vol. 11, no. 2, pp. 32–36, 2022, doi: 10.36733/jikt.v11i2.5427.
- [4] K. Maso *et al.*, "PERAN ARTIFICIAL INTELLIGENCE (AI) UNTUK MENDUKUNG PEMBELAJARAN DI MASA PANDEMI COVID-19 Luh," *Crit. Care Med.*, vol. 43, no. 1, p. 267, 2015, doi: 10.1097/01.ccm.0000474893.34162.5c.
- [5] K. Upton, "Computer Vision A Modern Approach," *Manuf. Eng.*, vol. 74, no. 3, pp. 111–113, 1995, doi: 10.1049/me:19950308.
- [6] R. C. Hardie and M. M. Hayat, "Digital Images: Processing," *Encycl. Opt. Photonic Eng. Second Ed.*, pp. 567–573, 2015, doi: 10.1081/E-EOE2-120009509.
- [7] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023, doi: 10.1007/s11042-022-13644-y.
- [8] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.
- [9] P. Riau *et al.*, "Deteksi Lubang Jalan Secara Otomatis dari Rekaman Drone Menggunakan Model Machine Learning Berbasis YOLOv5 Instance Segmentation di Kota Deteksi Lubang Jalan Secara Otomatis dari Rekaman Drone Menggunakan Model Machine Learning Berbasis YOLOv5 Instance Se," no. October, 2024, doi: 10.52364/zona.v8i2.126.
- [10] M. B. Habibi and S. Wibowo, "Pengembangan Aplikasi Deteksi Kerusakan Lubang Jalan Berbasis Android," *Pros. Semin. Nas. ...*, vol. 1, no. 1, pp. 260–267, 2023, [Online]. Available: <https://conference.upgris.ac.id/index.php/infest/article/view/3765%0Ahttps://conference.upgris.ac.id/index.php/infest/article/download/3765/24>

- 19
- [11] H. Hambrose, "The Design of Everyday Work," *Des. Manag. Rev.*, vol. 26, no. 2, pp. 10–15, 2015, doi: 10.1111/drev.10319.
- [12] M. Kaluža, M. Kalanj, and B. Vukelić, "A COMPARISON OF BACK-END FRAMEWORKS FOR WEB APPLICATION DEVELOPMENT," *Zb. Veleučilišta u Rijeci*, vol. 7, no. 1, pp. 317–332, 2019.
- [13] M. Garba and H. Abubakar, "A Comparison of NoSQL and Relational Database Management Systems (RDBMS)," *Kasu J. Math. Sci.*, vol. 1, no. August, 2020, [Online]. Available: <http://www.journal.kasu.edu.ng/index.php/kjms>