

PENGEMBANGAN SISTEM DETEKSI LUBANG PADA JALAN MENGUNAKAN ALGORITMA YOLO BERBASIS ESP32-CAM

Ekmalzdaki Royhan Mahar Beta Adi Sucipto¹, Rekyan Regasari Mardi Putri², Budi Darma Setiawan³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ekmalroyhan@student.ub.ac.id, ²rekyan.rmp@ub.ac.id, ³s.budidarma@ub.ac.id

Abstrak

Jalan merupakan infrastruktur penting yang mendukung transportasi, pembangunan wilayah, dan pemerataan hasil pembangunan. Namun, kerusakan jalan berupa lubang sering terjadi akibat faktor alam dan aktivitas manusia, seperti cuaca ekstrem dan beban kendaraan berat. Hal ini meningkatkan risiko kecelakaan dan biaya perawatan kendaraan. Penelitian ini bertujuan mengembangkan sistem deteksi lubang pada jalan secara real-time dengan memanfaatkan algoritma YOLOv8 dan perangkat ESP32-CAM. Sistem ini diharapkan mengatasi keterbatasan metode manual yang membutuhkan banyak waktu, tenaga, dan rentan terhadap human error. Pendekatan penelitian ini menggunakan metodologi Agile dengan pengujian sistem pada jalan sepanjang 3,5 km. Hasil pengujian menunjukkan bahwa sistem dapat mencapai F1-Score sebesar 0,73, dengan precision 0,70 dan recall 0,78. Waktu rata-rata untuk inference adalah 300-350 ms, preprocess 0-12 ms, dan postprocess 0-3 ms. Berdasarkan hasil analisis waktu proses dan mengacu pada kriteria sistem real-time, sistem ini dapat dikategorikan sebagai real-time karena mampu merespons tanpa keterlambatan yang signifikan. Kinerja sistem stabil meskipun dipengaruhi oleh faktor pencahayaan, kompleksitas objek, dan daya yang tersedia. Meskipun sistem ini masih memerlukan komputer untuk menjalankan YOLOv8, penelitian ini merupakan langkah penting menuju pengembangan sistem deteksi lubang pada jalan yang lebih mandiri dan akurat. Hasil penelitian ini menunjukkan bahwa integrasi ESP32-CAM dengan YOLOv8 dapat mendeteksi lubang pada jalan secara real-time dengan tingkat akurasi yang memadai.

Kata kunci: deteksi objek, deteksi lubang pada jalan, YOLOv8, ESP32-CAM, real-time, kecerdasan buatan

Abstract

Roads are important for transportation, area development, and spreading the benefits of progress. However, potholes often form due to natural causes and human activities, such as extreme weather and heavy vehicle traffic. This increases the risk of accidents and vehicle repair costs. This study aims to create a real-time pothole detection system using the YOLOv8 algorithm and ESP32-CAM devices. The system is designed to solve the problems of manual methods that take a lot of time, effort, and are prone to human mistakes. The research uses the Agile methodology and tests the system on a 3.5 km stretch of road. The results show that the system can achieve an F1-Score of 0.73, with precision of 0.70 and recall of 0.78. The average inference time is 300-350 ms, preprocess time is 0-12 ms, and postprocess time is 0-3 ms. Based on an analysis of the processing time and referencing real-time system criteria, the system can be categorized as real-time, as it can respond without significant delays. The system's performance stays stable even with different lighting conditions, object complexity, and available power. Although the system still needs a computer to run YOLOv8, this research is an important step toward developing a more independent and accurate pothole detection system. The study shows that combining ESP32-CAM with YOLOv8 can detect potholes on the road in real-time with good accuracy.

Keywords: object detection, pothole detection, YOLOv8, ESP32-CAM, real-time, artificial intelligence

1. PENDAHULUAN

Berdasarkan Undang-Undang Republik

Indonesia Nomor 38 Tahun 2004 tentang Jalan, jalan didefinisikan sebagai prasarana transportasi yang memiliki peranan penting

dalam mendukung perkembangan wilayah, pemerataan hasil pembangunan, dan pencapaian pembangunan nasional. Namun, kerusakan jalan sering terjadi akibat faktor manusia dan alam, seperti perubahan suhu, cuaca, serta muatan kendaraan yang semakin berat. Kondisi ini berdampak buruk pada pengguna jalan dan lingkungan sekitar. Menurut Nabawi et al. (2021), kerusakan jalan dapat berupa retak, kegemukan, distorsi, pengausan, dan lubang, yang memberikan dampak negatif dari tiga aspek: biaya, kenyamanan, dan keamanan.

Pada aspek biaya, kerusakan jalan meningkatkan pengeluaran masyarakat, penggunaan bahan bakar, serta biaya perawatan kendaraan. Dari sisi kenyamanan, kerusakan ini memengaruhi efektivitas perjalanan, kebisingan, dan kesehatan fisik. Sementara itu, dari aspek keamanan, kerusakan jalan meningkatkan risiko kecelakaan, ketidakstabilan kendaraan, serta kecemasan pengendara. Di antara berbagai jenis kerusakan jalan, lubang menjadi fokus utama karena menjadi salah satu penyebab utama kecelakaan dan kerusakan kendaraan, terutama pada jalan dengan volume kendaraan tinggi dan kecepatan yang cukup besar. Pengendara yang tidak menyadari keberadaan lubang sering kali kehilangan kendali atas kendaraannya, yang dapat berujung pada kecelakaan dan biaya perawatan kendaraan yang lebih besar.

Solusi yang diusulkan dalam penelitian ini adalah pengembangan sistem pendeteksian lubang pada jalan secara real-time, yang diharapkan dapat mengatasi keterbatasan perhitungan manual dari lubang pada jalan yang memerlukan waktu, tenaga, dan sumber daya yang besar, serta rawan terhadap human error. Dengan menggunakan teknologi deteksi objek berbasis kecerdasan buatan dan sistem portable, proses pendeteksian diharapkan dapat dilakukan secara real-time dengan tingkat akurasi yang tinggi, sehingga mempercepat identifikasi dan memungkinkan penanganan dini sebelum kerusakan bertambah parah.

Penelitian ini terinspirasi dari penelitian Putri (2024) yang menggunakan algoritma You Only Look Once (YOLO) versi 8 untuk mendeteksi berbagai jenis kerusakan jalan. Meskipun hasil penelitian tersebut menunjukkan nilai precision sebesar 0,66, recall sebesar 0,563, dan F1-Score sebesar 0,614, terdapat ruang untuk pengembangan lebih lanjut. Fokus pada deteksi lubang dengan tujuan untuk mengeksplorasi potensi peningkatan kinerja model, baik dari segi precision, recall, maupun

F1-Score. Selain itu, penelitian ini juga terinspirasi oleh penelitian Fauzan (2021) dan Khairunnisa (2023), yang mengintegrasikan computer vision dengan sistem embedded, menggunakan hardware alternatif seperti Raspberry Pi dan ESP32-CAM.

Penelitian ini bertujuan untuk mengeksplorasi kemampuan algoritma YOLOv8 dalam mendeteksi lubang pada jalan, dengan memanfaatkan ESP32-CAM sebagai mikrokontroler utama. Diharapkan sistem yang dikembangkan dapat memberikan solusi yang praktis, efisien, dan mudah diimplementasikan secara luas dalam pengelolaan infrastruktur jalan. Analisis terhadap berbagai faktor, seperti kondisi pencahayaan, ukuran, dan variasi bentuk lubang, juga akan dilakukan untuk meningkatkan akurasi sistem. Dengan pendekatan ini, penelitian diharapkan memberikan kontribusi nyata dalam peningkatan keselamatan lalu lintas dan efisiensi pengelolaan jalan.

2. DASAR TEORI

2.1 Lubang Jalan

Lubang jalan seperti pada gambar 1 merupakan salah satu bentuk kerusakan pada jalan yang menyebabkan berbagai macam masalah bagi pengguna jalan, seperti ketidaknyamanan ketika melintas hingga kecelakaan yang dapat merenggut nyawa dari pengguna jalan. Lubang jalan biasanya terbentuk akibat faktor-faktor seperti beban kendaraan berlebih, dan kualitas konstruksi jalan yang buruk.



Gambar 1 Lubang Jalan
Sumber: detik.com, 2023

2.2 YOLOv8

Algoritma YOLOv8 merupakan algoritma versi terbaru dari rangkaian model YOLO yang terkenal untuk deteksi objek dalam citra dan video. Algoritma YOLOv8 merupakan algoritma yang digunakan dalam penelitian ini. YOLOv8 dirancang untuk menggabungkan kecepatan dan efisiensi dalam mendeteksi objek dengan tingkat akurasi yang tinggi. Model ini

memiliki arsitektur yang lebih ringan dan lebih cepat dibandingkan dengan versi sebelumnya, seperti YOLOv5, sambil tetap mempertahankan performa yang optimal. YOLOv8 menggunakan teknik deep learning dan convolutional neural networks (CNN) yang memungkinkan deteksi objek dalam satu kali pencitraan, membuatnya sangat cocok untuk aplikasi real-time seperti pemantauan video, sistem keamanan, dan kendaraan otonom.

2.3 ESP32-CAM

ESP32-CAM adalah modul kamera dengan ukuran kecil yang menggunakan ESP32-S sebagai sirkuitnya. Terdapat kamera OV2640 sebagai kamera yang digunakan, dan beberapa GPIOs untuk terhubung dengan perangkat lain. ESP32-CAM dapat digunakan dalam berbagai macam pengaplikasian IoT, seperti smart home devices, industrial wireless control, wireless monitoring, dan pengaplikasian IoT lainnya. ESP32-CAM digambarkan pada gambar 2 sebagai sebuah mikrokontroler.

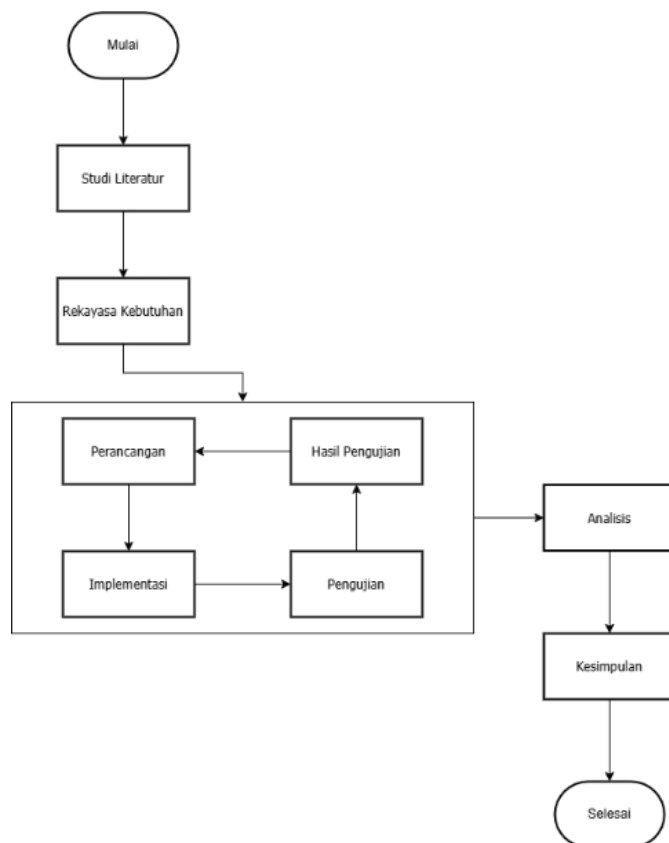


Gambar 2 ESP32-CAM

3. METODOLOGI PENELITIAN

Penelitian ini termasuk dalam kategori penelitian implementatif dengan pendekatan pengembangan (development). Penelitian implementatif bertujuan untuk menerapkan teori yang sudah ada ke dalam praktik nyata, dengan fokus pada pengembangan atau penerapan suatu metode untuk memecahkan masalah spesifik. Penelitian ini tergolong sebagai penelitian pengembangan karena selain menguji teori yang sudah ada, pengembangan sebuah sistem yang dapat diimplementasikan dalam situasi nyata untuk memenuhi kebutuhan tertentu seperti memecahkan masalah di dunia nyata dilakukan.

Penelitian ini menggunakan strategi penelitian dengan pendekatan Agile sebagai tahapan pengembangan sistemnya. Agile dipilih karena memiliki kemampuan untuk mendukung pengembangan yang bersifat iteratif dan inkremental. Hal tersebut memungkinkan terjadinya penyesuaian pada sistem ketika sistem dilatih untuk mendapat hasil yang diinginkan selama proses penelitian.



Gambar 3 Diagram Alir Strategi Penelitian

Metodologi yang digunakan dalam penelitian ini melibatkan tujuh tahapan utama seperti pada gambar 3 diagram alir strategi penelitian, yaitu studi literatur, rekayasa kebutuhan, perancangan, implementasi, pengujian, analisis, dan kesimpulan. Studi literatur dilakukan untuk memahami konsep terkait sistem deteksi lubang pada jalan, termasuk penelitian terdahulu atau penelitian serupa. Pemahaman ini mencakup penggunaan ESP32-CAM sebagai mikrokontroler, algoritma YOLO dan YOLOv8 sebagai algoritma klasifikasi, serta integrasi embedded system dan computer vision yang relevan dengan penelitian ini.

Tahapan rekayasa kebutuhan bertujuan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional yang diperlukan untuk membangun sistem. Kebutuhan ini dianalisis untuk memastikan semua aspek sistem terpenuhi. Selanjutnya, pada tahapan perancangan, dilakukan perencanaan terhadap seluruh komponen perangkat keras dan perangkat lunak. Hasil dari tahap ini adalah rancangan sistem yang memuat detail bagaimana setiap komponen bekerja sesuai kebutuhan.

Pada tahapan implementasi, dilakukan pengembangan sistem secara langsung, termasuk penulisan kode program untuk ESP32-

CAM, pembuatan model YOLOv8, dan integrasi keduanya. Sistem ini kemudian diuji untuk membaca dan melakukan klasifikasi lubang pada jalan secara real-time. Setelah implementasi, dilakukan pengujian sistem secara keseluruhan untuk memastikan semua komponen bekerja sesuai dengan rencana. Pengujian ini menitikberatkan pada performa ESP32-CAM ketika diintegrasikan dengan algoritma YOLOv8, terutama terkait akurasi dan kinerja sistem. Jika hasil pengujian belum memadai, akan dilakukan iterasi kembali ke tahap perancangan.

Setelah pengujian, dilakukan analisis terhadap hasil sistem untuk mengevaluasi performa akurasi dan waktu proses sistem deteksi lubang. Analisis ini mencakup perhitungan metrik evaluasi seperti confusion matrix dan F1-score untuk menilai performa akurasi, serta evaluasi performa ESP32-CAM dari aspek latency, daya tahan, dan komunikasi dengan komputer. Hasil analisis ini digunakan untuk mengidentifikasi masalah atau kendala sistem yang mungkin terjadi serta merancang perbaikan yang diperlukan.

Tahapan terakhir adalah penarikan kesimpulan, yang dituliskan berdasarkan hasil analisis dari seluruh tahapan sebelumnya.

Kesimpulan ini mencakup evaluasi akhir dari performa akurasi dan waktu proses sistem, serta memberikan kontribusi berupa referensi untuk penelitian dan pengembangan lebih lanjut di bidang serupa.

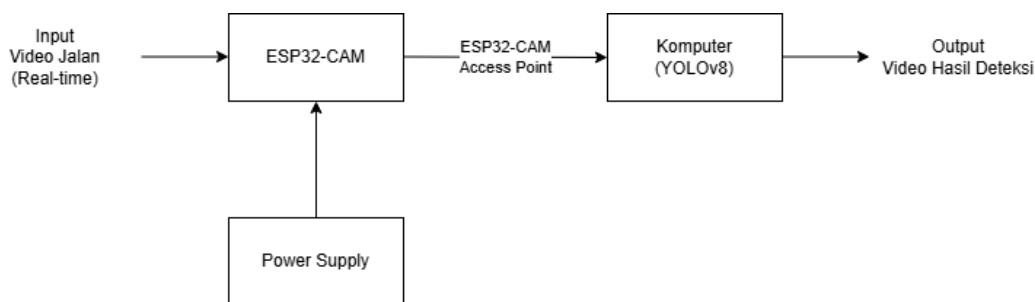
Subjek penelitian ini adalah lubang yang terdapat pada jalan, dengan lokasi penelitian berada di sepanjang Jalan Mayjen Sukertiyo, mencakup Kecamatan Rowokangkung dan Kecamatan Tekung, Kabupaten Lumajang. Data penelitian dikumpulkan secara real-time sebagai data primer, yang mencakup pengamatan terhadap lubang di sepanjang lokasi penelitian. Data ini kemudian digunakan untuk menguji sistem sesuai kebutuhan pengujian.

Analisis data dalam penelitian ini dibagi menjadi dua bagian, yaitu analisis akurasi dan analisis waktu proses. Analisis akurasi meliputi evaluasi akurasi yang dihasilkan oleh ESP32-CAM yang terintegrasi dengan algoritma YOLOv8 dalam mendeteksi lubang pada jalan.

Evaluasi dilakukan dengan menggunakan perhitungan confusion matrix dan F1-score sebagai tolak ukur performa akurasi sistem. Analisis waktu proses mencakup evaluasi performa waktu proses sistem secara real-time, seperti latency, waktu yang diperlukan untuk preprocessing, inference, dan postprocessing, serta daya tahan ESP32-CAM selama proses pengujian. Analisis ini bertujuan untuk menilai efisiensi sistem dalam mendeteksi lubang pada jalan secara langsung.

4. PERANCANGAN SISTEM

Pada proses perancangan sistem dimulai dari membuat diagram blok untuk menjadi gambaran keseluruhan sistem deteksi lubang pada jalan. Diagram blok digambarkan dengan beberapa komponen penyusun sistem yang saling terhubung dan berinteraksi. Gambar 4 merupakan diagram blok dari sistem ini.



Gambar 4 Diagram Blok Sistem

4.1 Perancangan Perangkat Keras

Perancangan perangkat keras untuk sistem deteksi lubang pada jalan terdiri dari beberapa komponen. Komponen pertama adalah ESP32-CAM yang digunakan untuk membaca data, menerima data, dan mengirim data. Komponen kedua adalah power supply dalam bentuk power bank yang berfungsi untuk memberikan daya pada komponen pertama. Kedua komponen tersebut dihubungkan dengan bantuan kabel USB yang seperti pada gambar 5 digambarkan dengan garis berwarna merah untuk saling menghubungkan. Pada ESP32-CAM terdapat ESP32-CAM MB Adapter yang berfungsi sebagai adaptor untuk ESP32-CAM supaya dapat menggunakan kabel USB baik untuk menerima daya dan kode program.



Gambar 5 Rancangan Perangkat Keras Sistem

4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak untuk sistem deteksi lubang pada jalan mencakup beberapa proses utama, yaitu perancangan model, pembacaan data, pengiriman dan penerimaan data, serta pembacaan dan pengolahan data.

Perancangan Model dimulai dengan identifikasi kebutuhan dataset berupa gambar lubang pada jalan yang diperoleh dari

pengambilan gambar lapangan atau dataset yang tersedia. Dataset dianotasi menggunakan platform Roboflow untuk menandai area lubang pada jalan sebagai target deteksi. Setelah anotasi selesai, model awal dibuat dan dilatih menggunakan YOLOv8 di Google Colab hingga siap digunakan.

Perancangan Pembacaan Data melibatkan inisialisasi ESP32-CAM, mencakup konfigurasi perangkat keras dan perangkat lunak untuk mengambil gambar. Library `esp_camera.h` digunakan untuk mengaktifkan fitur kamera dan mengatur resolusi serta format gambar. Proses ini dianggap berhasil jika ESP32-CAM dapat mengambil gambar.

Perancangan Pengiriman dan Penerimaan Data dimulai dengan mengonfigurasi ESP32-CAM sebagai Wi-Fi Access Point untuk mentransfer data gambar ke komputer. Gambar yang diambil oleh ESP32-CAM dikirimkan melalui jaringan Wi-Fi dan diterima di komputer untuk pengolahan lebih lanjut. Proses ini berhasil jika data gambar dapat diterima di komputer tanpa kendala.

Perancangan Pembacaan dan Pengolahan Data mencakup penerimaan data gambar di komputer melalui jaringan Wi-Fi. Data gambar diproses menggunakan algoritma YOLOv8 melalui program Python di Visual Studio Code (VSCode). Hasil deteksi ditampilkan dalam bentuk visualisasi, seperti gambar dengan kotak deteksi dan informasi klasifikasi. Proses ini dinyatakan berhasil jika sistem mampu mendeteksi lubang pada jalan dan menampilkan hasil deteksinya.

4.3 Implementasi Sistem

Pada proses implementasi sistem deteksi lubang pada jalan dimulai dari menjalankan beberapa rencana yang sudah dirancang sebelumnya. Sistem diimplementasikan mengikuti rencana yang sudah ditetapkan dengan harapan dapat menghasilkan hasil yang diperlukan untuk melakukan pengujian pada sistem.

Pada proses implementasi perangkat keras sistem deteksi lubang pada jalan, dimulai dengan memasang komponen-komponen yang diperlukan dalam sistem deteksi lubang pada jalan sesuai dengan rencana yang sudah ditetapkan. Proses pemasangan komponen ini melibatkan ESP32-CAM, ESP32-CAM MB Adapter, power bank, dan kabel USB. ESP32-

CAM dengan ESP32-CAM MB Adapter dihubungkan dengan power bank dengan kabel USB seperti pada gambar 6. ESP32-CAM kemudian direkatkan di atas power bank untuk menjaga agar ESP32-CAM tetap stabil pada saat pengambilan data gambar.



Gambar 6 Implementasi Perangkat Keras

Pada proses implementasi perangkat lunak sistem dibagi menjadi implementasi model, implementasi pembacaan data, implementasi pengiriman dan penerimaan data, dan implementasi pembacaan dan pengolahan data.

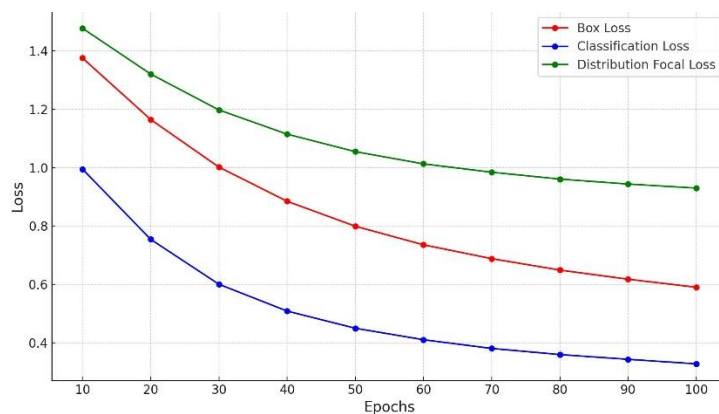
Implementasi model dilakukan sesuai dengan apa yang direncanakan pada perancangan model dimulai dari melakukan pengumpulan dataset, pengamatan dataset, pembuatan model awal di platform Roboflow, dan melatih model di platform Google Colab hingga model siap untuk digunakan.

Tabel 1 Dataset yang Digunakan

Jenis Dataset	Jumlah Data	Persentase
Data Latih	5500	57,89%
Data Validasi	3500	36,84%
Data Uji	1500	15,79%
Total	9500	100%

Jumlah dataset yang dipilih sebesar 9500 gambar untuk mengoptimalkan batas yang diberikan oleh platform Roboflow (10000 gambar) tanpa melampaui limit untuk pengguna gratis. Pembagian dataset ini dilakukan dengan tujuan untuk model memiliki cukup data untuk pelatihan sebesar 57.89%, validasi sebesar 36.84%, dan pengujian sebesar 15.79%.

Hasil pelatihan model yang berjalan selama 1.3 hingga 1.5 jam untuk setiap 10 epochs hingga 100 epochs. Pelatihan tersebut menghasilkan hasil yang paling baik dari 100 epochs yaitu precision bernilai 0.823, recall bernilai 0.705, mAP50 bernilai 0.761, mAP50-95 bernilai 0.506.



Gambar 7 Grafik Loss Metrics

Pada gambar 7 ditunjukkan grafik dari loss metrics yang didapat selama dilakukan pelatihan model, grafik tersebut menunjukkan perubahan nilai box_loss, cls_loss, dan dfl_loss selama 100 epochs. Untuk box_loss, secara konsisten menurun dari 1.376 pada epoch ke-10 hingga 0.59 pada epoch ke-100. Penurunan signifikan terjadi hingga sekitar epoch ke-50, kemudian melambat namun tetap menurun. Dari hasil tersebut, box loss menunjukkan bahwa model semakin baik dalam memprediksi posisi bounding box seiring bertambahnya epoch. Untuk cls_loss, menurun secara stabil dari 0.995 pada epoch ke-10 menjadi 0.3273 pada epoch ke-100. Penurunan awal lebih tajam dibandingkan akhir. Dari hasil tersebut, classification loss menunjukkan bahwa model semakin baik dalam mengklasifikasikan objek ke kelas yang benar dengan stabilitas pada epoch akhir menunjukkan model mulai konvergen. Untuk dfl_loss, turun dari 1.478 pada epoch ke-10 menjadi 0.93 pada epoch ke-100. Penurunan bertahap sepanjang pelatihan, hampir mirip dengan box loss. Dari hasil tersebut, distribution focal loss menunjukkan bahwa model meningkatkan kualitas distribusi prediksi koordinat bounding box, mendekati ground truth. Berdasarkan data-data tersebut dapat dikatakan semua metrik loss menunjukkan penurunan yang stabil, dengan nilai semakin kecil mendekati akhir pelatihan dan penurunan yang melambat setelah epoch ke-50 menunjukkan model mendekati konvergensi. Nilai loss yang rendah pada akhir pelatihan mengindikasikan model telah memahami dataset dengan baik dan tidak ada tanda-tanda overfitting atau underfitting yang jelas dalam data tersebut.

Implementasi pembacaan data dilakukan dengan dimulai dari inisialisasi dan program

ESP32-CAM, penggunaan library `esp_camera.h`, dan penangkapan gambar oleh ESP32-CAM dapat dilakukan. Penggunaan library `esp_camera.h` sebagai pendukung operasi kamera ESP32-CAM. Kamera dikonfigurasi mulai dari model kamera, konfigurasi pin sesuai model yaitu AI-THINKER, dan format kamera.

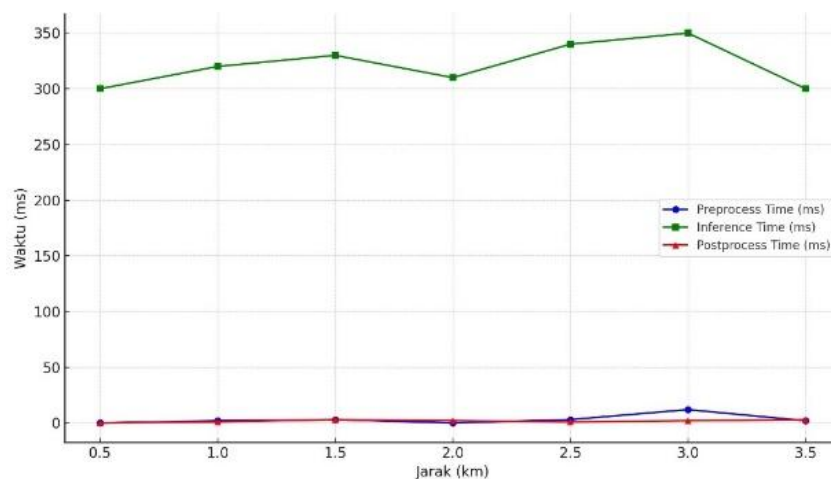
Implementasi pengiriman dan penerimaan data dilakukan dengan dimulai dari setup ESP32-CAM sebagai Wi-Fi Access Point, pengiriman data via Wi-Fi, dan penerimaan data di komputer. Penggunaan library `wifi.h` yang berfungsi sebagai library untuk menangani komunikasi nirkabel dengan menggunakan SSID dan password access point. Web server untuk antarmuka kamera juga digunakan untuk membantu pengguna dalam mengatur dan melihat hasil kamera.

Implementasi pembacaan dan pengolahan data dilakukan dengan dimulai dari penerimaan gambar dari ESP32-CAM oleh komputer, pemrosesan gambar oleh YOLOv8 pada komputer, dan menampilkan hasil proses pada komputer. Penggunaan library `cv2` (OpenCV) dan library `Ultralytics` digunakan untuk menangani operasi kamera dan memuat serta menjalankan model YOLOv8. Hasil deteksi sistem akan ditampilkan pada implementasi ini, dengan menampilkan anotasi (bounding box) pada lubang pada jalan yang terdeteksi. Hasil deteksi juga akan disimpan dalam file terpisah untuk dapat dianalisis lebih lanjut. Penetapan fungsi tombol untuk memberhentikan sistem saat melakukan deteksi juga ditetapkan pada implementasi ini dengan tujuan untuk memudahkan antarmuka pengguna pada saat melakukan deteksi.

5. PENGUJIAN DAN ANALISIS HASIL

5.1 Pengujian Performa Waktu Proses Sistem

Pengujian dilakukan untuk mengetahui performa waktu proses sistem dari ESP32-CAM yang terintegrasi dengan YOLOv8 melalui komputer bekerja. Jalan sepanjang 3,5 km dilewati dalam pengujian ini, dengan sistem diletakkan pada bagian dashboard mobil.



Gambar 8 Grafik Performa Sistem

Grafik pada gambar 8 menggambarkan performa sistem ESP32-CAM yang terintegrasi dengan YOLOv8 melalui komputer dalam hal waktu proses (preprocess, inference, dan postprocess) selama pengujian di jalan sepanjang 3,5 km. Inference time antara 300 hingga 350 ms dapat dianggap stabil untuk sistem berbasis mikrokontroler seperti ESP32-CAM yang digunakan dengan YOLOv8 untuk sistem deteksi lubang pada jalan. Menurut penelitian oleh Choi (2020), untuk sistem real-time yang menggunakan mikrokontroler, waktu inferensi dalam rentang 200-500 ms sudah cukup stabil dan dapat diterima. Hal ini disebabkan oleh keterbatasan mikrokontroler dalam pemrosesan data dibandingkan perangkat yang lebih kuat. Menurut penelitian oleh Zhang (2021), juga menunjukkan meskipun terjadi fluktuasi waktu eksekusi terutama pada operasi pemrosesan gambar dengan algoritma kompleks seperti YOLO, sistem masih dapat beradaptasi dengan perubahan kondisi pencahayaan dan kompleksitas gambar tanpa menyebabkan gangguan yang cukup besar. Preprocess time yang bervariasi dari 0 ms hingga 12 ms juga dapat diterima karena menunjukkan adanya beberapa tambahan dalam menyiapkan data gambar, yang merupakan hal yang wajar dalam sistem berbasis mikrokontroler yang melakukan pengolahan citra real-time. Postprocess time yang sebagian besar mendekati 0 ms, dengan beberapa puncak hingga 3 ms, menunjukkan

bahwa sistem dapat menyaring hasil inferensi dengan cepat, meskipun terdapat beberapa fluktuasi kecil yang tidak signifikan terhadap performa sistem secara keseluruhan.

Variasi waktu proses sistem dalam pendeteksian lubang pada jalan dapat dipengaruhi oleh beberapa faktor. Salah satu faktor utama adalah kondisi pencahayaan, di mana kamera OV2640 pada ESP32-CAM sangat sensitif terhadap perubahan intensitas cahaya. Ketika sistem berada di bawah bayangan pohon atau area dengan pencahayaan terlalu terang, waktu tambahan diperlukan untuk menyesuaikan kontras dan detail gambar, sehingga menyebabkan peningkatan waktu proses. Selain itu, kompleksitas objek dalam gambar juga menjadi faktor penting. Jika gambar mengandung elemen rumit, seperti bayangan yang menyerupai lubang atau pola jalan yang tidak rata, sistem membutuhkan waktu lebih lama untuk membedakan antara lubang asli dan noise. Faktor lain yang memengaruhi adalah kapasitas daya yang disuplai oleh power bank. Ketika daya mulai menurun, performa ESP32-CAM dalam memproses data dapat terganggu, yang pada akhirnya menyebabkan fluktuasi waktu, terutama pada tahap postprocess time.

Berdasarkan hasil pengujian, sistem ini dapat dikategorikan sebagai sistem real-time. Menurut penelitian yang dilakukan oleh Stanković et al. (2019), sistem dapat dianggap

real-time jika waktu proses keseluruhannya (termasuk preprocessing, inferencing, dan postprocessing) berada dalam rentang yang memungkinkan hasil deteksi direspons tanpa keterlambatan yang signifikan. Dalam pengujian ini, inference time berada di kisaran 300-350 ms, yang masih sesuai dengan standar waktu respon real-time untuk perangkat berbasis mikrokontroler. Selain itu, preprocess time (0-12 ms) dan postprocess time (0-3 ms) juga mendukung efisiensi sistem dalam melakukan deteksi pada sistem. Dengan indikator waktu proses yang stabil dan responsif, serta kemampuan sistem untuk beradaptasi terhadap perubahan kondisi lingkungan seperti pencahayaan, sistem ini memenuhi kriteria sebagai sistem real-time.

5.2 Pengujian Performa Akurasi Sistem

Pengujian dilakukan untuk mengetahui performa akurasi sistem dalam melakukan deteksi lubang pada jalan, ESP32-CAM yang sudah terintegrasi dengan YOLOv8 melalui komputer. Jalan sepanjang 3,5 km dilewati dalam pengujian ini, dengan sistem diletakkan pada bagian dashboard mobil.

Perangkat ESP32-CAM menangkap gambar pada jalan dan mengirimkannya pada komputer melalui ESP32-CAM access point, komputer kemudian akan mengolah hasil tangkapan gambar tersebut dengan YOLOv8 untuk dideteksi ada atau tidaknya lubang pada jalan serta lubang pada jalan yang tidak terdeteksi seperti pada gambar 9, gambar 10, dan gambar 11. Jarak penangkapan gambar tersebut memiliki batas kurang lebih 2 meter dari perangkat ESP32-CAM diletakkan. Pelataan posisi perangkat juga mempengaruhi hasil deteksi, maka dari itu perangkat diletakkan tepat pada tengah dari bagian dashboard mobil untuk mendapatkan hasil yang optimal dan menjangkau keseluruhan jalan.



Gambar 9 Sistem Benar dalam Mendeteksi Lubang pada Jalan



Gambar 10 Sistem Salah dalam Mendeteksi Lubang pada Jalan



Gambar 11 Sistem Tidak Mendeteksi Adanya Lubang pada Jalan

Tabel 2 Confusion Matrix Sistem

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP = 65	FP = 27
	Negative	FN = 18	TN = 0

Pada tabel 2 menunjukkan hasil dari sistem dalam melakukan deteksi lubang pada jalan dalam bentuk confusion matrix. Tabel tersebut didapat dari penghitungan manual dari video pada saat sistem melakukan deteksi dengan dianalisa setiap framenya dari video berdurasi 2 menit dengan 24 frame untuk setiap detiknya sehingga sebanyak 2880 frame akan dilakukan perhitungan manual untuk mendapatkan nilai seperti pada tabel 6.2. Pada tabel tersebut memiliki hasil True Positive (TP) sebanyak 65, False Positive (FP) sebanyak 27, dan False Negative (FN) sebanyak 18. Hasil tersebut merupakan hasil pengujian sistem pada jalan dengan jarak 3,5 km dan dengan Actual Positive (AP) sebanyak 83. Berdasarkan beberapa hasil tersebut maka dapat dilakukan perhitungan sebagai berikut.

$$Precision = \frac{TP}{TP + FP} = \frac{65}{65 + 27} = \frac{65}{92} = 0.70$$

$$Recall = \frac{TP}{TP + FN} = \frac{65}{65 + 18} = \frac{65}{83} = 0.78$$

$$\begin{aligned} F1\ Score &= 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \\ &= 2 \times \left(\frac{0.70 \times 0.78}{0.70 + 0.78} \right) = 0.73 \end{aligned}$$

Dari hasil perhitungan tersebut diperoleh nilai Precision sebesar 0,70, Recall sebesar 0,78, dan F1-Score sebesar 0,73. Nilai Recall yang dihasilkan terbilang cukup, yang menunjukkan bahwa sistem mampu mendeteksi sebagian besar lubang pada jalan yang ada. Nilai Precision yang dihasilkan juga terbilang cukup, yang menunjukkan bahwa sistem dapat dengan benar mendeteksi objek berupa lubang pada jalan, namun ada kalanya sistem salah mendeteksi objek sebagai lubang pada jalan.

Hasil nilai F1-Score yang sebesar 0,73 menunjukkan bahwa sistem memiliki keseimbangan yang cukup baik antara Precision dan Recall. F1-Score merupakan ukuran yang menggabungkan kedua metrik tersebut, dengan memberikan perhatian khusus pada keterimbangan antara kemampuan sistem dalam mendeteksi objek dengan benar (Precision) dan kemampuan sistem dalam mendeteksi objek yang relevan (Recall). Nilai F1-Score yang mendekati 1,0 menandakan kinerja deteksi yang sangat baik, sementara nilai yang lebih rendah menunjukkan adanya ruang untuk perbaikan. Dengan nilai F1-Score sebesar 0,73, dapat dikatakan bahwa sistem ini cukup efektif dalam mendeteksi lubang pada jalan, meskipun ada potensi untuk meningkatkan akurasi, baik dalam meningkatkan Precision agar lebih sedikit kesalahan deteksi objek, maupun dalam meningkatkan Recall agar lebih banyak lubang yang terdeteksi.

Setelah dilakukan analisis ulang terhadap hasil yang diperoleh, beberapa faktor penyebab tingkat akurasi sistem telah diidentifikasi. Salah satu faktor utama adalah kesalahan dalam pengambilan gambar. Bayangan sering kali terdeteksi sebagai lubang, sehingga meningkatkan jumlah False Positive. Selain itu, banyak lubang pada jalan yang tidak terdeteksi karena model kurang memahami variasi bentuk lubang yang ditemui selama pengujian langsung. Kondisi pencahayaan juga memengaruhi kinerja sistem, di mana pencahayaan yang terlalu terang atau terlalu gelap dapat menyebabkan lubang pada jalan tidak terdeteksi. Faktor berikutnya adalah kurangnya keragaman dalam dataset. Dataset yang digunakan belum sepenuhnya mencerminkan karakteristik gambar dari kamera OV2640 pada ESP32-CAM, sehingga

mengurangi kemampuan model untuk mengenali berbagai kondisi lapangan. Terakhir, terdapat keterbatasan pada platform pelatihan model seperti Roboflow dan Google Colab, termasuk jumlah dataset yang dapat digunakan dan jumlah epochs pelatihan. Hal ini membatasi kemampuan untuk menghasilkan model yang lebih optimal.

6. KESIMPULAN

Berdasarkan rumusan masalah yang dibuat pada awal penelitian serta hasil dan analisis pengujian yang telah dilakukan, dapat disimpulkan beberapa hal. Pertama, sistem menunjukkan performa yang cukup stabil dalam pengujian real-time pada jarak 3,5 km, dengan rata-rata waktu inference 300-350 ms, preprocess 0-12 ms, dan postprocess 0-3 ms. Berdasarkan analisis waktu proses dan mengacu pada kriteria sistem real-time, sistem ini dapat dikategorikan sebagai sistem real-time karena mampu merespons tanpa keterlambatan yang signifikan. Integrasi ESP32-CAM dengan algoritma YOLOv8 membuktikan bahwa pendekatan ini dapat dilakukan sebagai pengembangan sistem deteksi lubang pada jalan dengan pendekatan yang berbeda. Kedua, sistem berhasil mencapai nilai F1-Score sebesar 0.73, dengan nilai precision sebesar 0.70 dan recall sebesar 0.78. Hasil ini menunjukkan bahwa sistem memiliki akurasi yang cukup untuk mendeteksi lubang pada jalan secara real-time, meskipun masih terdapat potensi untuk pengembangan lebih lanjut agar lebih akurat dan optimal.

Berdasarkan hasil dan analisis pengujian penelitian, beberapa saran dapat diberikan untuk pengembangan penelitian selanjutnya. Untuk meningkatkan akurasi sistem, langkah pertama yang disarankan adalah melatih model dengan dataset yang lebih besar dan beragam. Dataset saat ini terbatas pada 9.500 gambar akibat keterbatasan platform yang digunakan, sehingga variasi data yang mencakup kondisi jalan dan pencahayaan masih belum optimal. Selain itu, dataset yang digunakan harus relevan dengan karakteristik kamera OV2640 pada ESP32-CAM, mengingat spesifikasi kamera seperti resolusi dan kualitas warna memengaruhi hasil tangkapan gambar. Dataset yang sesuai dengan kemampuan kamera ini akan membantu meningkatkan performa model dalam mendeteksi lubang jalan secara lebih akurat dalam berbagai kondisi. Untuk kinerja sistem,

saat ini masih terdapat variasi waktu proses terutama pada tahap preprocessing dan postprocessing. Hal ini dapat diatasi dengan menggunakan sumber daya yang lebih stabil, seperti baterai berkapasitas besar atau catu daya yang mendukung kebutuhan ESP32-CAM secara konsisten. Optimasi pada tahapan preprocessing dan postprocessing, termasuk penyesuaian parameter model dan pengurangan beban komputasi yang tidak diperlukan, juga dapat meningkatkan konsistensi dan efektivitas waktu proses sistem tanpa penurunan performa yang signifikan.

Selain itu, untuk menghasilkan sistem yang lebih praktis dan efisien, disarankan memahami algoritma machine learning alternatif yang dapat diterapkan langsung pada perangkat ESP32-CAM. Saat ini, model YOLOv8 memerlukan komputer tambahan untuk pemrosesan karena beban komputasi yang tinggi. Algoritma yang lebih ringan dapat menjadi alternatif untuk memungkinkan sistem bekerja secara mandiri. Pengintegrasian model ini dengan perangkat keras ESP32-CAM secara optimal akan memungkinkan pengembangan sistem deteksi lubang yang lebih praktis dan efisien. Dengan memahami dan menerapkan algoritma yang lebih sederhana, pengembangan selanjutnya dapat menghasilkan sistem yang lebih siap untuk aplikasi lapangan. Keterbatasan platform seperti Roboflow dan Google Colab juga menjadi hambatan utama dalam pelatihan model dan pengumpulan dataset. Oleh karena itu, disarankan menggunakan platform alternatif tanpa batasan, baik dari segi jumlah dataset maupun durasi pelatihan model. Dengan platform yang lebih fleksibel, model dapat dilatih lebih lama dengan dataset yang lebih banyak, sehingga meningkatkan potensi akurasi dan performa. Penggunaan platform ini akan memungkinkan pengembangan mencapai hasil yang lebih maksimal tanpa terganggu oleh batasan teknis.

7. DAFTAR PUSTAKA

- Choi, H., Kim, S., & Lee, Y. (2020). Real-time object detection on embedded systems using YOLO. *Journal of Embedded Systems*, 15(3), 34-4.
- ESP32.net. (n.d.). ESP32 overview. Diakses dari <https://www.esp32.net>.
- Fauzan, A. A. A., & Utaminigrum, F. (2021). Sistem Pendeteksi Dini Lubang pada Jalan menggunakan Gray Level Co-Occurrence Matrix berbasis Raspberry Pi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(3), 845-851.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Pearson Education.
- KMPedia. (2022, Jan 18). Embedded System, Apa Artinya?. Diakses dari <https://www.kmtech.id/post/embedded-system-apa-artinya>.
- Khairunnisa, N., Yuniarti, E., & Purwowibowo. (2023). Detektor angka real-time berbasis mikrokontroler ESP32 CAM dengan pengolahan data menggunakan algoritma YOLO. Diakses dari <https://repository.uinjkt.ac.id/dspace/handle/123456789/75022>.
- Kumar, P., & Kumar, V. (2023). Exploring the frontier of object detection: A deep dive into YOLOv8 and the COCO dataset. In *2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)* (pp. 1-6). IEEE.
- Last Minute Engineers. (2024). Getting started with ESP32-CAM. Diakses dari <https://lastminuteengineers.com/getting-started-with-esp32-cam/>.
- Liang, J. (2022). Confusion matrix: Machine learning. *POGIL Activity Clearinghouse*, 3(4).
- McKinsey & Company. (2023, 6 Oktober). It's coming home: The return of agile hardware product development. McKinsey & Company. Diakses dari <https://www.mckinsey.com/capabilities/operations/our-insights/its-coming-home-the-return-of-agile-hardware-product-development> (diakses pada Januari 2025).
- Morales, I. H. (2023). Brief overview of embedded systems for Industry 4.0 Applications and Networks. In *2023 IEEE Integrated STEM Education Conference (ISEC)* (pp. 18-21). Laurel, MD, USA: IEEE.
- Nabawi, I., Wahidin, W., Feriska, Y., Diantoro, W., & Imron, I. (2021). Analisis Dampak Kerusakan Jalan terhadap Pengguna Jalan dan Lingkungan di Ruas Jalan Pebatan-Rengaspendawa Brebes. *Infratech Building Journal*, 2(1), 28-34.

- P. Agrawal et al., "YOLO Algorithm Implementation for Real Time Object Detection and Tracking," 2022 IEEE Students Conference on Engineering and Systems (SCES), Prayagraj, India, 2022, pp. 01-06.
- Putri, S. A. (2024). Penerapan Model Deteksi Jenis Kerusakan Jalan Dengan Menggunakan Algoritma You Only Look Once (YOLO) Versi 8 (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- Salikhov, R. B., Abdrakhmanov, V. K., & Safargalin, I. N. (2021, November). Internet of things (IoT) security alarms on ESP32-CAM. In *Journal of Physics: Conference Series* (Vol. 2096, No. 1, p. 012109). IOP Publishing.
- SunFounder. (2024). ESP32 CAM. SunFounder GalaxyRVR Kit for Arduino 1.0 documentation. Diakses dari ESP32 CAM — SunFounder GalaxyRVR Kit for Arduino 1.0 documentation.
- Tafida, A. I., Bt Wan Zawawi, N. A., Alaloul, W. S., & Musarat, M. A. (2024). A systematic review of the use of computer vision and photogrammetry tools in learning-based dimensional road pavement defect detection for smart transportation. In *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)* (pp. 1-9). Harrisonburg, VA, USA: IEEE.
- Undang-Undang No.38 Tahun 2004 tentang Jalan (Indonesia). Diakses dari UU No. 38 Tahun 2004 (bpk.go.id).
- Zhang, J., Wang, L., & Xu, D. (2021). Performance analysis of real-time object detection on embedded devices. *IEEE Transactions on Embedded Computing*, 22(6), 180-192.