Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Sistem Identifikasi Cerdas: Integrasi IOT dengan YOLOv8 Untuk Identifikasi Visual Kerusakan Dinding Bangunan

Kamdan Kamdan*, Somantri Somantri, Satria Rizki Rohmat, Agung Gumelar, Ivana Lucia Kharisma

Fakultas Teknik Komputer dan Desain, Program Studi Teknik Informatika, Universitas Nusa Putra, Sukabumi, Indonesia
Email: [1,*]kamdan@nusaputra.ac.id, [2]somantri@nusaputra.ac.id, [3]satria.rizki_ti21@nusaputra.ac.id,
[4]agung.gumelar_ti21@nusaputra.ac.id, [5]ivana.lucia@nusaputra.ac.id
Email Penulis Korespondensi: kamdan@nusaputra.ac.id
Submitted: **04/07/2025**; Accepted: **04/09/2025**; Published**: 05/09/2025**

Abstrak-Kerusakan pada elemen non-struktural bangunan, khususnya dinding, dapat menjadi indikator awal dari potensi kerusakan yang lebih serius. Identifikasi keretakan secara manual sering kali memakan waktu, bergantung pada subjektivitas tenaga ahli, dan tidak selalu konsisten. Penelitian ini mengembangkan sistem identifikasi otomatis berbasis *computer vision* menggunakan arsitektur YOLOv8 yang terintegrasi dengan teknologi Internet of Things (IoT) melalui perangkat ESP32-CAM. Sistem ini dirancang untuk mengidentifikasi dan mengklasifikasikan kerusakan dinding secara visual ke dalam kategori ringan, sedang, atau berat berdasarkan citra yang ditangkap di lapangan. Model dilatih dan dievaluasi menggunakan metrik *confusion matrix* untuk mengukur kinerja klasifikasinya. Berdasarkan pengujian yang telah dilakukan, sistem menunjukkan kinerja yang optimal dengan perolehan nilai mAP@50 sebesar 0.822, serta mAP@50-95 yang lebih ketat sebesar 0.522, menandakan kapabilitas sistem dalam mendeteksi objek kerusakan dengan tingkat ketelitian yang baik. Implementasi sistem ini diharapkan dapat mendukung proses inspeksi bangunan secara lebih terstandar, objektif, dan berkelanjutan, serta membantu dalam pengambilan keputusan terkait perawatan dan perbaikan struktur bangunan.

Kata Kunci: YOLOv8; Identifikasi Dinding Bangunan; Computer Vision; Internet of Things; ESP32-CAM; Kecerdasan Buatan

Abstract-Damage to non-structural building elements, particularly walls, can serve as an early indicator of more serious structural issues. Manual crack identification is often time-consuming, subjective, and lacks consistency. This study develops an automated identification system based on computer vision using the YOLOv8 architecture, integrated with Internet of Things (IoT) technology through the ESP32-CAM device. The system is designed to visually detect and classify wall damage into light, moderate, or severe categories based on field-captured images. The model was trained and evaluated using the confusion matrix metric to assess its classification performance. The test results show that the system achieved a solid performance with an mAP@50 score of 0.822 and a stricter mAP@50-95 score of 0.522**,** indicating the system's strong capability in detecting damage objects with a good level of precision. The implementation of this system is expected to support building inspection processes in a more standardized, objective, and sustainable manner, and assist in decision-making regarding building maintenance and repair.

Keywords: YOLOv8; Building Wall Identification; Computer Vision; Internet of Things; ESP32-CAM, Artificial Intelligence

1. PENDAHULUAN

Bangunan memiliki peran penting dalam kehidupan manusia, baik sebagai hunian, ruang kerja, maupun sarana fasilitas umum. Bangunan terdiri dari 2 elemen, yaitu elemen struktural dan nonstruktural. Bagian elemen struktural meliputi kolom, balok, lantai dan lain-lain. Sedangkan elemen nonstruktural meliputi, pintu, jendela, langit-langit, dinding, serta elemen pendukung lainnya[1]. Ketahanan bangunan bergantung pada elemen struktural dan nonstrukturalnya. Salah satu nya elemen dinding, yang merupakan bagian dari elemen nonstruktural. Dalam banyak kasus, dinding menjadi salah satu elemen bangunan yang paling rentan terhadap kerusakan[2].

Kerusakan pada dinding tidak hanya mengurangi estetika bangunan, tetapi juga dapat menjadi potensi bahaya jika tidak segera ditangani. Kerusakan tersebut dapat berkembang lebih besar, merusak lapisan cat, memengaruhi bagian lainnya seperti jendela dan atap, hingga meningkatkan risiko keruntuhan. Untuk mencegah kerusakan semakin memburuk, diperlukan tindakan perbaikan. Namun, perbaikan tersebut tidak boleh dilakukan sembarangan. Penting untuk memahami penyebab kerusakan terlebih dahulu agar penanganannya dapat dilakukan secara tepat sesuai kebutuhan[3]. Berdasarkan sifat dan tingkat keparahannya, kerusakan pada dinding bangunan umumnya diklasifikasikan menjadi tiga kategori utama, yaitu: (1) kerusakan *hairline* yang ditandai dengan retak halus pada permukaan, (2) kerusakan *struktural* yang berkaitan dengan gangguan pada elemen penopang dinding, dan (3) kerusakan *spalling* yang ditandai dengan pengelupasan atau terlepasnya material permukaan dinding akibat korosi atau kelembaban. Pemahaman terhadap klasifikasi ini menjadi dasar penting dalam menentukan metode penanganan yang tepat dan berkelanjutan[4].

Secara umum, Identifikasi kerusakan pada dinding dilakukan secara manual oleh tenaga ahli atau teknisi bangunan. Metode ini memakan waktu yang cukup lama, tergantung pada penilaian individu, dan tidak efisien. Oleh karena itu, dibutuhkan sebuah sistem otomatis yang dapat mengindentifikasi keretakan dengan cepat dan akurat. Salah satu solusi yang dapat diterapkan adalah dengan memanfaatkan teknologi *computer vision*[5]. *Computer vision*, sebagai bagian dari *artificial intelligence*, berfokus pada bagaimana sebuah mesin atau sistem dapat melihat dan memahami berbagai hal, seperti pengenalan objek, pendeteksian gerakan, rekonstruksi adegan, dan pemulihan gambar[6]. Penelitian ini mengimplementasikan algoritma *You Only Look Once* (YOLO) untuk mendeteksi objek. Cara kerja model ini adalah dengan mempartisi sebuah gambar menjadi beberapa bagian *grid*. Selanjutnya, *feature map* yang menjadi keluaran dari YOLO akan berisi prediksi *bounding box* (kotak pembatas), skor kepercayaan objek,

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

dan skor probabilitas kelas. Algoritma YOLO dipilih karena dikenal sebagai salah satu metode deteksi objek yang paling cepat namun tetap mampu mempertahankan performa dan akurasi yang tinggi[7]. Secara umum, identifikasi kerusakan pada dinding dilakukan secara manual oleh tenaga ahli atau teknisi bangunan. Metode ini memakan waktu yang cukup lama, bergantung pada penilaian individu, dan tidak efisien. Oleh karena itu, dibutuhkan sebuah sistem otomatis yang dapat mengidentifikasi keretakan dengan cepat dan akurat dengan memanfaatkan teknologi *computer vision* dan *Internet of Things* (IoT) [8].

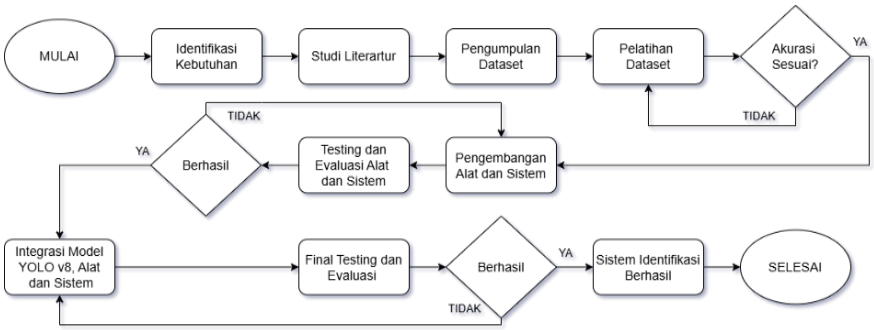
Metode Identifikasi kerusakan pada dinding bangunan berbasis deep learning bukanlah hal yang sepenuhnya baru. Sebelumnya, penelitian seperti implementasi menggunakan algoritma CNN untuk mendeteksi dan mengidentifikasi tingkat kerusakan pascagempa, namun fokusnya hanya pada kerusakan skala besar dan tidak dirancang untuk deteksi dini retakan halus[9]. Penelitian dengan mengkomparasi berbagai model kecerdasan buatan untuk klasifikasi kerusakan dinding bangunan, seperti algoritma *VGG16-ANN*, dan *VGG16-ANN.* Arsitektur tersebut cenderung berat dan kurang optimal untuk implementasi pada perangkat IoT berdaya rendah di lapangan[4]. Penelitian dengan mengembangkan sistem deteksi keretakan permukaan bangunan berbasis web menggunakan algoritma *yolov8* dengan akurasi *mAP* sebesar 75%, tetapi masih memerlukan pengguna untuk mengunggah gambar secara manual dan belum terintegrasi dengan sistem akuisisi gambar otomatis[10]. Penelitian dengan komparasi beberapa model *deep learning* seperti *VGG19, ResNet50, InceptionV3, EfficientNetV2* untuk identifikasi kerusakan dinding bangunan, namun arsitektur tersebut cenderung berat untuk penerapan sebagai algoritma sistem deteksi *real-time*[11]. Serta penelitian sistem klasifikasi kerusakan dinding bangunan dengan menggunakan algoritma *naïve bayes*[12].

Meskipun berbagai pendekatan sebelumnya telah dikembangkan untuk mendeteksi kerusakan dinding, sebagian besar masih menghadapi tantangan dalam penerapan lapangan, seperti ketergantungan pada unggahan gambar manual, kompleksitas arsitektur yang tinggi untuk perangkat edge, serta keterbatasan dalam mengenali kerusakan minor secara akurat. Penelitian ini menawarkan pendekatan alternatif dengan mengembangkan sistem deteksi kerusakan berbasis YOLOv8 yang terintegrasi penuh dengan perangkat IoT seperti ESP32-CAM dan dashboard visualisasi. Pendekatan ini ditujukan untuk menghadirkan sistem inspeksi bangunan yang efisien, ringan, serta mampu bekerja secara real-time di lingkungan lapangan dengan sumber daya terbatas.

2. METODOLOGI PENELITIAN

2.1 Metode Penelitian

Penelitian ini menggunakan metode prototipe rekayasa sistem, yang mencakup tahapan identifikasi kebutuhan, perancangan sistem, implementasi perangkat keras dan perangkat lunak, serta evaluasi kinerja sistem. Pendekatan ini dipilih agar sistem yang dikembangkan dapat langsung diuji fungsionalitasnya di lapangan sesuai kebutuhan[13]. Alur lengkap proses penelitian ditampilkan pada Gambar 1



Gambar 1. Tahapan Penelitian

2.2 Pengumpulan Dataset dan Pelatihan Model

Pada tahap pengumpulan dataset ini, penulis akan mengumpulkan gambar yang menunjukkan berbagai jenis keretakan atau kerusakan pada struktur dinding bangunan, khususnya bagian non-struktural Gambar-gambar ini akan diambil dari berbagai sumber terbuka dari internet. Data yang terkumpul akan digunakan untuk melatih dan menguji metode identifikasi kerusakan menggunakan arsitektur YOLOv8. Pada gambar 2 dan tabel 1 dibawah ini merupakan rincian dataset dan citra gambar kerusakan dinding bangunan yang akan di gunakan:

Gambar 2. Citra Gambar dataset

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Tabel 1. Pengumpulan Dataset

Dataset	Sumber	Jumlah
Hairline	Roboflow.[14]	1280
Structural	Roboflow.[15]	1280
Spalling	Roboflow[15]	1280
Hairline,Structural, Spalling	Diambil secara langsung di berbagai lokasi bangunan	260
Gambar non kerusakan	Pexels[16]	1100
Total		5200

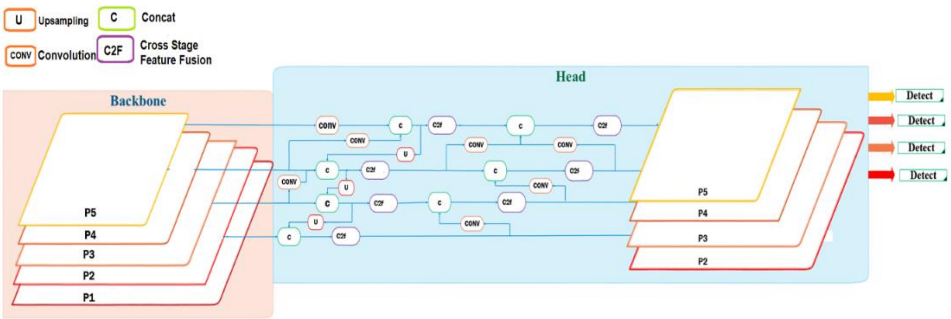
Sebanyak 260 gambar kerusakan diambil secara langsung di berbagai lokasi bangunan, menggunakan kamera ESP32-CAM. Pengambilan dilakukan dalam berbagai kondisi pencahayaan alami, yaitu pada pagi hari, sore dan malam hari, dengan jarak pengambilan 30–150 cm dari permukaan dinding bangunan. Gambar ini melengkapi data lain dari sumber daring sehingga total dataset terdiri dari 5200 gambar, yang mencakup 4160 gambar kerusakan dan 1040 gambar non-kerusakan. Dataset kemudian dibagi menjadi tiga subset: 80% untuk *training*, 10% untuk *testing*, dan 10% untuk *validation*. Pembagian ini dirancang untuk memastikan pelatihan model berlangsung optimal dan representatif terhadap kondisi lapangan.[17]. Pada tabel 2 dibawah ini, merupakan salah satu skema untuk *training, testing dan validation:*

Tabel 2. Skema Pembagian Dataset untuk Pelatihan dataset

Dataset	Proporsi	Jumlah gambar
Train	80 %	4160
Validation	10%	520
Testing	10%	520
Total	100%	5200

Proses pelatihan (training) model merupakan tahap krusial dalam pengembangan alat identifikasi kerusakan dinding bangunan berbasis IoT yang dirancang dalam penelitian ini. Melalui proses ini, model diharapkan mampu mencapai nilai *mean Average Precision* (mAP) yang optimal. Pelatihan dilakukan selama 150 epoch dengan *batch size* sebesar 16, menggunakan *optimizer* Adam dan ukuran input gambar sebesar 640×640 piksel. Selain itu, diterapkan juga beberapa teknik augmentasi seperti *horizontal flip, scaling, mosaic, dan HSV augmentation* untuk meningkatkan keragaman data latih dan kemampuan generalisasi model.

Sebagai bagian dari upaya peningkatan performa, dilakukan pula kustomisasi pada arsitektur model, khususnya pada bagian *neck* dengan menerapkan *GFPN (Generalized Feature Pyramid Network)***.** GFPN digunakan untuk memperkuat proses ekstraksi fitur dari berbagai level piramida, sehingga model dapat mendeteksi kerusakan dengan tingkat presisi yang lebih tinggi, baik pada objek kecil maupun besar[18]. Kustomisasi ini diharapkan dapat meningkatkan efektivitas model dalam menghadapi variasi bentuk dan ukuran kerusakan dinding yang ditemukan di lapangan. Pada gambar 3 dibawah ini merupakan visualisasi arsitektur YOLOv8-*GFPN*



Gambar 3. Arsitektur YOLOv8-*GFPN*[18]

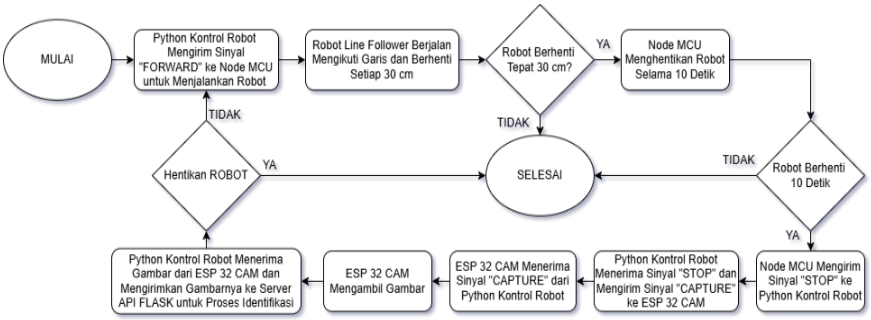
2.3 Metode Evaluasi Hasil Training

*Mean Aver*age Precision *(mAP)* merupakan metrik yang digunakan untuk menilai kinerja model dalam tugas seperti deteksi objek dan pengambilan informasi. Dalam konteks deteksi objek, *mAP* mengukur performa model dengan mempertimbangkan keseimbangan antara presisi dan *recall* pada berbagai nilai ambang *Intersection over Union (IoU).* Dengan demikian, *mAP* tidak hanya menilai jumlah objek yang berhasil dideteksi, tetapi juga mengevaluasi ketepatan bounding box yang dihasilkan [19].

2.4 Tools Development

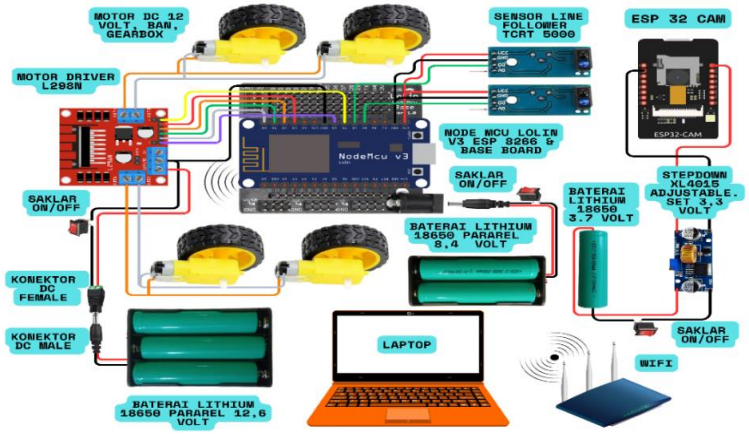
Perencanaan alat identifikasi kerusakan struktur bangunan dilakukan dalam bentuk prototipe nyata yang dapat diimplementasikan langsung di lapangan. Perancangan alur dan perangkat disusun secara optimal agar selaras dengan tujuan penelitian dan kebutuhan kondisi lapangan, untuk detailnya dapat diamati pada Gambar 4:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882



Gambar 4. Alur kerja alat

Pada Gambar 4, sistem ini menjelaskan sebuah robot line follower yang dikontrol oleh Python dan NodeMCU untuk melakukan inspeksi visual otomatis. Robot bergerak mengikuti garis, berhenti setiap 30 cm selama 10 detik, lalu ESP32 CAM diperintahkan untuk mengambil gambar. Gambar tersebut kemudian dikirim oleh Python ke sebuah server API Flask untuk proses identifikasi, setelah itu sistem akan memutuskan apakah robot akan melanjutkan siklus pergerakan dan pengambilan gambar berikutnya atau menghentikan seluruh operasi. Selanjutnya merupakan Proses perancangan alat untuk mengidentifikasi kerusakan pada struktur bangunan yang berbasis kecerdasan buatan dan IoT. Rancangan instalasi perangkat di gambarkan pada Gambar 5 berikut:



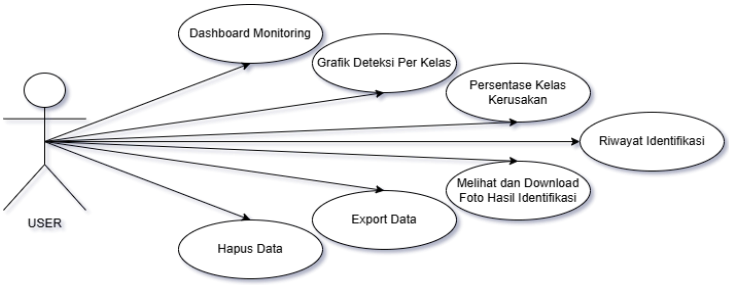
Gambar 5. Instalasi alat

2.5 Software Development

Pengembangan perangkat lunak untuk sistem identifikasi kerusakan dinding ini dilakukan melalui beberapa tahapan terstruktur. Tahap pertama adalah perancangan fungsional untuk mendefinisikan kebutuhan dan interaksi pengguna atau *use case*. Tahap selanjutnya adalah implementasi teknis, yang dipecah menjadi pengembangan ***server*** untuk logika pemrosesan model dan *frontend* untuk visualisasi data pada dashboard.

a. *Use Case Diagram*

Untuk merancang kebutuhan sistem, interaksi antara pengguna dan *Dashboard YOLO Wall Identification System* divisualisasikan melalui sebuah *use case diagram*. Pada Gambar 6 berikut ini secara spesifik menyajikan *use case diagram* yang merepresentasikan berbagai fungsi yang tersedia pada *dashboard monitoring*.

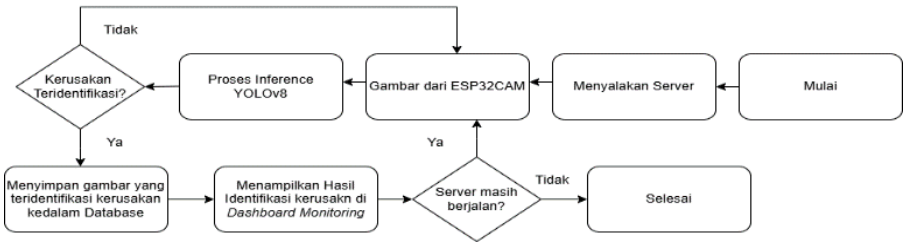


Gambar 6. *Use Case Diagram*

Gambar 6 menunjukkan *use case diagram* dari sistem dashboard identifikasi kerusakan struktur bangunan. Diagram ini menggambarkan interaksi antara pengguna (*user*) dengan fitur-fitur utama yang tersedia di dalam sistem. Diagram ini merepresentasikan keseluruhan fungsionalitas utama yang dirancang untuk mempermudah pengguna dalam memantau serta mengevaluasi hasil identifikasi kerusakan bangunan secara terstruktur.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

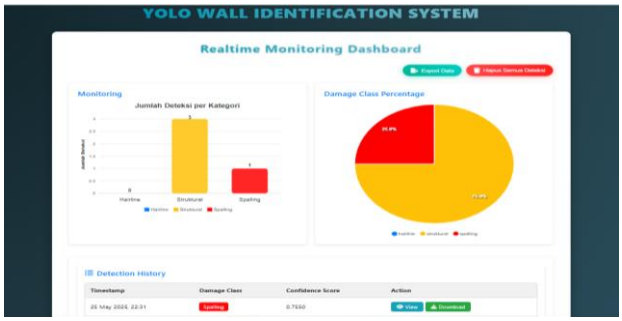
b. Pengembangan *Server*
Tahap ini mencakup proses pembangunan sisi *server,* yang berfungsi sebagai pusat pemrosesan untuk sistem identifikasi kerusakan. Server tersebut dibangun menggunakan bahasa pemrograman *python* dan *framework* Flask. Alur kerja utamanya dapat dilihat pada Gambar 7 berikut:



Gambar 7. Alur Kerja *Server*

Gambar 7 menunjukkan alur kerja sistem identifikasi kerusakan bangunan berbasis YOLOv8 yang terintegrasi dengan ESP32-CAM. Proses diawali dengan mengaktifkan server utama, yang kemudian menerima citra dari ESP32-CAM untuk dianalisis menggunakan YOLOv8 guna mendeteksi kerusakan pada dinding bangunan. Jika tidak ada kerusakan terdeteksi, sistem menunggu input berikutnya. Jika ada, hasil deteksi disimpan ke basis data *MySQL* dan ditampilkan di dashboard monitoring. Proses ini berlangsung otomatis selama server aktif, mencakup akuisisi citra, analisis AI, hingga penyimpanan dan visualisasi secara real-time.

c. Pengembangan *Front End Dashboard Monitoring*
Pada Tahap ini merupakan pengembangan antarmuka pengguna (frontend) dashboard dilakukan menggunakan bahasa pemrograman JavaScript dengan framework vue.js. Vue.js merupakan kerangka kerja progresif untuk membangun antarmuka pengguna yang interaktif dan modular[20]. Pemilihan Vue.js didasarkan pada kemudahannya dalam pengelolaan state dan integrasi komponen, sehingga mempermudah proses pengembangan tampilan dashboard yang dinamis dan responsif. Tampilan awal dashboard hasil pengembangan frontend dapat dilihat pada Gambar 8, yang kemudian direalisasikan secara menyeluruh dalam tahap implementasi.



Gambar 8. Pengembangan *Dashboard Monitoring*

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Alat

Tahap implementasi alat merupakan realisasi fisik dari sistem inspeksi, di mana robot line follower digunakan sebagai wahana bergerak. NodeMCU berperan sebagai pengendali pergerakan, sementara ESP32-CAM berfungsi mengambil citra visual untuk identifikasi kerusakan. Gambar 9 dibawah menunjukkan implementasi alat tersebut.:

Gambar 9. Implementasi Alat

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Gambar 9 menunjukkan implementasi sistem inspeksi otomatis, di mana robot mengikuti jalur garis, berhenti setiap 0,5 detik gerakan selama 10 detik untuk mengambil gambar menggunakan ESP32-CAM. Citra yang ditangkap dikirim secara real-time ke server melalui Wi-Fi untuk dianalisis oleh model YOLOv8 dalam mendeteksi kerusakan dinding. Proses ini berlangsung berulang hingga seluruh area selesai diinspeksi.

3.2 Hasil Pelatihan Model

Proses pelatihan model dilakukan untuk mengevaluasi performa dan menentukan arsitektur terbaik dalam mendeteksi kerusakan dinding bangunan. Tiga skenario eksperimen diterapkan dengan konfigurasi yang sama, yaitu 150 epoch, batch size 16, pembagian dataset, dan teknik augmentasi. Kustomisasi juga dilakukan pada bagian neck dan head model, termasuk penerapan GFPN. Hasil pelatihan ketiga skenario ditampilkan pada Tabel 3 berikut..

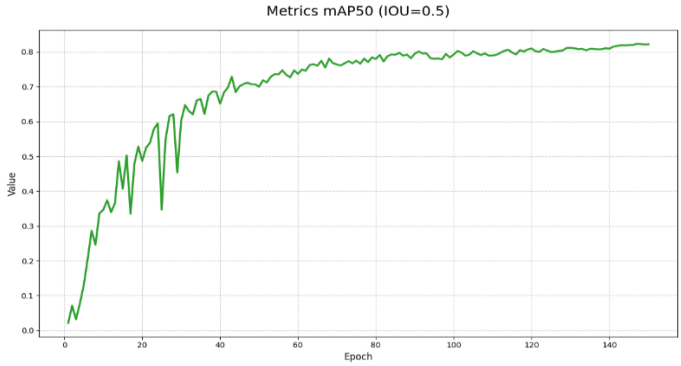
Tabel 3. Hasil Pelatihan Model

No	Nama Eksperimen	Epoch & Batch Size		Optimizer	mAP@50	mAP@50-95
1	YOLOv8s-Baseline	150	16	AdamW	0.7556	0.4587
2	YOLOv8m-Baseline	150	16	AdamW	0.7780	0.4790
3	YOLOv8s-GFPN	150	16	AdamW	0.8225	0.5225

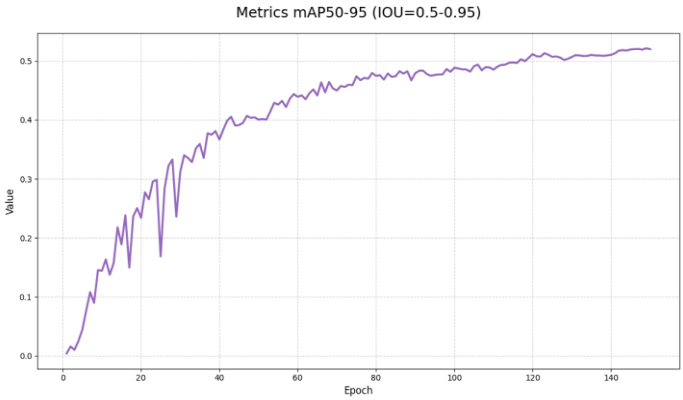
Berdasarkan Tabel 3 terdapat beberapa hasil yang diperoleh, model dengan konfigurasi pada percobaan ketiga dengan kustomisasi pada arsitektur menggunakan *GFPN* di latih dengan jumlah 150 *epoch* dan *optimizer* AdamW memperoleh performa terbaik, dengan nilai mAP@0.5 sebesar 0.8225 dan mAP@0.5:0.95 sebesar 0.5225.

3.3 Evaluasi Model Yolov8s-*GFPN*

a. Evaluasi *Mean Average Precision*
Dalam penilaian performa model dalam mendeteksi objek secara akurat, digunakan metrik *mean Average Precision* (mAP) yang merupakan salah satu indicator dalam evaluasi performa YOLOv8s dalam identifikasi kerusakan bangunan. Metrik ini menunjukkan sejauh mana performa model untuk mengenali dan mengklasifikasikan objek secara tepat pada berbagai tingkat kepercayaan (*confidence threshold*). Dalam evaluasi ini, *mAP* diukur pada dua level, yaitu *mAP*@0.5 dan *mAP*@0.5-0.95. Nilai *mAP*@50 menunjukkan akurasi ketika *Intersection over Union (IoU)* ditetapkan sebesar 0.5, sedangkan *mAP*@0.50-0.95 adalah rata-rata akurasi pada rentang *IoU* dari 0.5 hingga 0.95, yang mencerminkan ketelitian model secara lebih menyeluruh[21]. Pada Gambar 10 dan 11 dibawah ini, merupakan hasil evaluasi mAP@0.5 dan mAP@0.5-0.95:



Gambar 10. mAP@50

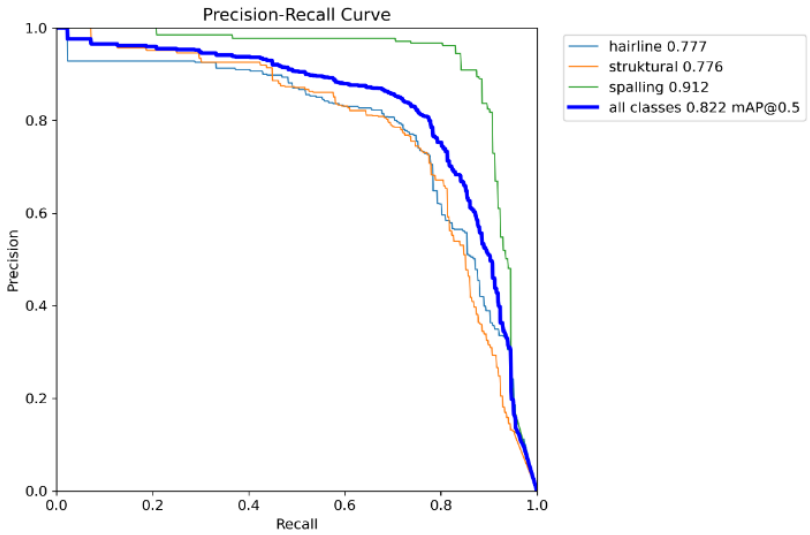


Gambar 11. mAP@0.50-0.95

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Gambar 10 menunjukkan grafik mAP@0.5 per epoch yang mengindikasikan peningkatan akurasi deteksi model seiring proses pelatihan. Nilai mAP@0.5 mencapai stabilitas di angka sekitar0.822 setelah 150 epoch. Sementara itu, Gambar 11 memperlihatkan grafik mAP@0.5-0.95 yang merepresentasikan ketelitian deteksi pada berbagai tingkat IoU. Nilai ini terus meningkat dan mencapai sekitar 0.522, menandakan performa model yang konsisten pada berbagai ambang batas overlap.

Setelah dilakukan evaluasi menggunakan metrik *mAP* pada *mAP@0.5* maupun *mAP@0.5-0.95*, langkah selanjutnya adalah menganalisis kinerja model berdasarkan pada kurva *Precision-Recall* (PR Curve). Visualisasi yang lebih jelas disajikan pada Gambar 12:

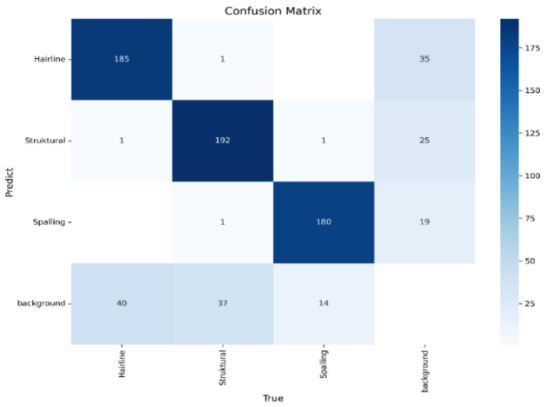


Gambar 12. *Precision-Recall Curve*

Gambar 12 di atas menunjukkan kurva Precision-Recall untuk tiga kelas kerusakan bangunan: *hairline***,** *struktural***,** dan *spalling***.** Model mendeteksi kerusakan *spalling* dengan performa terbaik di angka 0.912, diikuti oleh *hairline* 0.777 dan *structural* 0.776. Nilai *mA*P@0.5 untuk semua kelas sebesar 0.822, menunjukkan bahwa model memiliki akurasi deteksi yang cukup baik secara keseluruhan. Kurva ini menggambarkan keseimbangan antara presisi dan recall untuk masing-masing kelas.

b. Evaluasi *Confusion Matrix*

Setelah mengevaluasi performa model melalui metrik-metrik seperti *mAP, precision, recall, F1 score, ,* analisis selanjutnya dilakukan menggunakan confusion matrix. Visualisasi yang lebih jelas disajikan pada Gambar 13:



Gambar 13. Confusion matrix

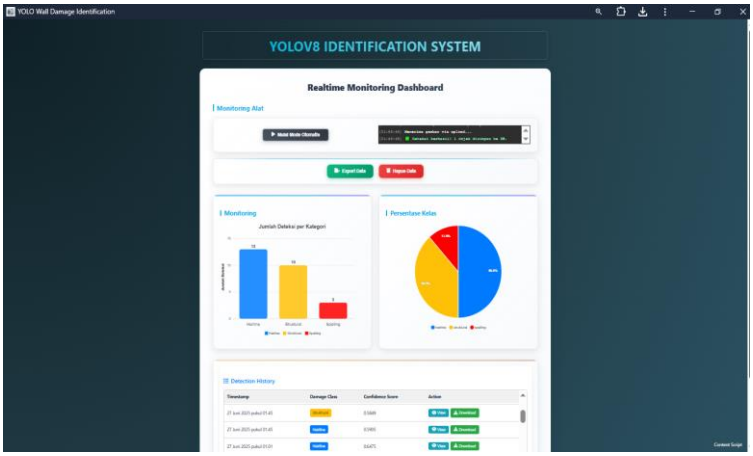
Sebagai alat evaluasi untuk mengukur kinerja algoritma klasifikasi, confusion matrix pada Gambar 13 menunjukkan perbandingan antara label kerusakan dinding yang sebenarnya dengan hasil prediksi model menggunakan YOLOv8 untuk mendeteksi jenis kerusakan Hairline, Struktural, Spalling.

3.4 Implentasi *Dashboard Monitoring*

Tahap implementasi dashboard monitoring merupakan langkah realisasi dari rancangan antarmuka pengguna yang telah disusun sebelumnya. Pada tahap ini, seluruh komponen sistem mulai dari kontrol perangkat, visualisasi hasil deteksi, hingga pengelolaan riwayat inspeksi digabungkan dalam satu tampilan terpusat. Dashboard ini berperan

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

sebagai penghubung antara pengguna dan sistem, menyederhanakan data kompleks menjadi informasi yang mudah dipahami. Visualisasi dashboard monitoring yang telah diimplementasikan ditampilkan pada Gambar 14.



Gambar 14. Implementasi *Dashboard Monitoring*

Gambar 14 menampilkan antarmuka *dashboard monitoring* yang menyajikan data hasil deteksi secara visual. *Dashboard* ini dilengkapi dengan beberapa fitur utama, yaitu terdapat tombol tindakan untuk memulai proses inspeksi alat secara otomatis, visualisasi jumlah deteksi per kategori dan persentase kelas kerusakan. Selain itu, terdapat riwayat deteksi lengkap serta fungsi manajemen data seperti ekspor data dan hapus deteksi.

3.5 Integrasi *Internet of Things* dengan *YOLOv8-GFPN*

Pada Tahap ini merupakan implementasi akhir dari sistem yang telah dirancang, yang mencakup integrasi antara perangkat Internet of Things (IoT) seperti *ESP32-CAM* dan *Robot Line Follower*, dengan model deteksi visual *YOLOv8s-GFPN* serta sistem *dashboard monitoring*. Pengujian dilakukan untuk memastikan bahwa seluruh komponen dapat bekerja secara sinergis sesuai alur sistem yang telah ditentukan, mulai dari pengambilan gambar, pengiriman data ke server, proses deteksi otomatis, hingga visualisasi hasil melalui dashboard.

a. Pengujian Alat
Pada tahap ini, pengujian fungsionalitas dan kinerja alat dilakukan secara komprehensif untuk memastikan semua berjalan sesuai harapan. Untuk lebih jelasnya dapat dilihat pada Tabel 4 dibawah ini:

Tabel 4. Pengujian alat

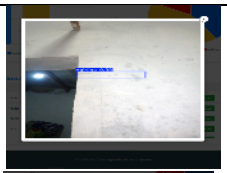
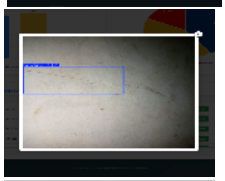
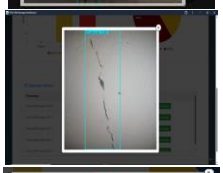
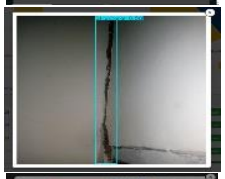
ID Pengujian	Pengujian	Alat yang Duji	Hasil yang Diperlukan	Status
RLF-01	Berjalan Mengikuti Garis	Robot Line Follower	mobil bergerak mengikuti garis hitam tanpa keluar jalur	Berhasil
RLF-02	Berhenti setiap 0.5 detik	Robot Line Follower	mobil berhenti total pada jangka waktu 0.5 detik setelah memulai perjalanan	Berhasil
RLF-03	Berhenti selama 10 detik	Robot Line Follower	mobil berhenti selama tepat 10 detik (toleransi +/- 1 detik)	Berhasil
RLF-04	Memotret gambar	ESP 32 CAM	ESP32-CAM mengambil gambar yang jelas saat robot berhenti.	Berhasil
RLF-05	Menerima dan menjalankan perintah potret dari python	ESP 32 CAM	ESP32-CAM berhasil menerima dan menjalankan perintah pemotretan dari skrip Python	Berhasil
RLF-06	Mengirim gambar ke server YOLO	Python	Gambar yang diambil ESP32-CAM berhasil terkirim ker server YOLO	Berhasil
RLF-07	Berjalan kembali setelah gambar terkirim	Robot Line Follower	Setelah gambar berhasil terkirim ke dashboard, robot otomatis melanjutkan perjalanan mengikuti garis	Berhasil
RLF-08	Siklus berulang hingga dimatikan	Robot Line Follower	Robot mengulang siklus berjalan-berhenti-potret-kirim gambar secara otomatis dan konsisten hingga dimatikan	Berhasil

Seluruh skenario pengujian yang disajikan pada Tabel 4 menunjukkan status "Berhasil". Hasil ini mengonfirmasi bahwa alat yang dikembangkan dapat menjalankan semua tugas yang dirancang, mencakup pergerakan fisik robot dan alur kerja sistem pemrosesan gambar.

b. Pengujian Model YOLOv8s-*GFPN*
Pada bagian ini, dilakukan pengujian terhadap model dengan menerapkan beberapa skenario berdasarkan jarak pengambilan gambar, yang rinciannya disajikan pada Tabel 5 ini:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Tabel 5. Pengujian model

Klasifikasi Kerusakan	Jarak Pengujian	Model Mengidentifikasi Kerusakan	Confidence Treshold	Hasil Gambar Identifikasi
Hairline	10 cm	Ya	0.81	
	20 cm	Ya	0.79	
	30 cm	Ya	0.85	
Struktural	30 cm	Ya	0.80	
	40 cm	Ya	0.85	
	50 cm	Ya	0.80	
Spalling	80 cm	Ya	0.84	
	120 cm	Ya	0.90	
	150 cm	Ya	0.83	

Berdasarkan hasil pengujian pada Tabel 5, semua skenario mampu berfungsi sesuai harapan, yang ditunjukkan oleh keberhasilan model dalam mendeteksi setiap jenis kerusakan *Hairline*, *Struktural*, dan *Spalling* secara konsisten di semua variasi jarak pengujian.

c. Pengujian *Dashboard Monitoring*

Pengujian fungsionalitas dashboard monitoring merupakan langkah krusial untuk memastikan sistem dapat berjalan sesuai kebutuhan pengguna. Tabel 6 menyajikan hasil pengujian skenario fungsionalitas utama dari dashboard monitoring.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Tabel 6. Pengujian *Dashboard Monitoring*

ID Pengujian	Pengujian	Pengguna	Hasil yang diperlukan	Status Pengujian	Catatan/ Saran
DM-01	Mengakses *Dashboard Monitoring*	Pengguna	Pengguna dapat mengakses *Dashboard Monitoring*	Selesai	-
DM-02	Tombol Mulai mode otomatis	Pengguna	Penggna dapat memulai inspeksi otomatis	Selesai	-
DM-03	Mendownload Data	Pengguna	Pengguna dapat mendownload data hasil identifikasi di *Dashboard Monitoring*	Selesai	-
DM-04	Menghapus Data	Pengguna	Pengguna dapat menghapus data yang berada di *Dashboard Monitoring*	Selesai	-
DM-05	Melihat atau mendownload gambar hasil identifikasi	Pengguna	Pengguna dapat melihat atau mendownload gambar hasil identifikasi yang berada di *Dashboard Monitoring*	Selesai	-
DM-06	Memulai mode otomatis identifikasi	Pengguna	Pengguna dapat menyalakan alat dan memulai proses identifikasi secara otomatis melalui *Dashboard Monitoring*	Selesai	-

Berdasarkan Tabel 6 menunjukkan bahwa seluruh skenario pengujian fungsionalitas *dashboard monitoring* telah berhasil dijalankan dengan status "Selesai". Hal ini menegaskan bahwa setiap fitur kunci seperti akses dashboard, kemampuan mengunduh data, menghapus data, melihat dan mengunduh gambar hasil identifikasi serta tombol tindakan untuk memulai inspeksi otomatis berfungsi dengan baik sesuai harapan.

3.6 Pembahasan

Bagian ini membahas analisis mendalam terhadap hasil penelitian, mencakup keunggulan model yang diusulkan, interpretasi temuan, serta implikasi praktis dan arah pengembangan sistem di masa depan.

a. Kinerja YOLOv8s-GFPN
Model YOLOv8s-GFPN menunjukkan kinerja deteksi terbaik dengan mAP@0.5 sebesar 0.8225 dan mAP@0.5-0.95 sebesar 0.5225. Kunci dari performa baik ini adalah adopsi Generalized Feature Pyramid Network (GFPN), yang memperkuat ekstraksi fitur dengan mengintegrasikan informasi dari berbagai skala spasial secara lebih efektif dibandingkan FPN atau PANet konvensional. Kemampuan ini krusial untuk mendeteksi kerusakan dengan ukuran bervariasi secara akurat, mulai dari retakan halus hingga *spalling* yang luas.
Jika dibandingkan, hasil mAP 0.8225 ini unggul secara signifikan dari penelitian sebelumnya[10], hanya mencapai mAP 0.75 menggunakan YOLOv8 standar tanpa integrasi IoT. Model ini juga lebih efisien secara komputasi dibandingkan arsitektur seperti ResNet50 atau InceptionV3[4] yang menghadirkan tantangan untuk implementasi *real-time* pada perangkat *edge*. Dengan demikian, model kami menawarkan kombinasi kinerja kompetitif dan efisiensi praktis.

b. Analisis *Precision-Recall* dan *Confusion Matrix*
Analisis lebih dalam menunjukkan kelas *spalling* mencapai *precision* dan *recall* tertinggi (0.912). Hal ini disebabkan oleh fitur visualnya yang sangat menonjol, seperti kontras tinggi pada area pengelupasan beton dan ukurannya yang relatif besar, sehingga mudah diidentifikasi oleh model.
Sebaliknya, temuan dari *confusion matrix* mengindikasikan adanya kesulitan dalam membedakan kelas retak halus (*hairline*) dan retak struktural. Kesamaan tekstur dan pola visual menjadi penyebab utama model keliru mengklasifikasikan keduanya. Implikasi praktis dari kesalahan ini sangat serius, karena dapat menyebabkan pengabaian kerusakan kritis (false negative) atau prioritas perbaikan yang salah (false positive). Hal ini menegaskan perlunya penyempurnaan dataset untuk meningkatkan akurasi diferensiasi kelas.

c. Implementasi Sistem *End-to-End* Melalui Integrasi IoT dan Dashboard
Integrasi ESP32-CAM dan NodeMCU pada robot *line follower* menjadi kunci untuk mewujudkan sistem inspeksi otomatis. Komponen ini memungkinkan akuisisi citra dan transmisi data secara *real-time* melalui Wi-Fi ke *dashboard monitoring*. Proses yang berjalan secara *end-to-end* ini mulai dari akuisisi data hingga visualisasi hasil menghilangkan intervensi manual dan mendukung inspeksi lapangan yang efisien dan berkelanjutan. *Dashboard* berfungsi sebagai antarmuka terpusat yang meningkatkan pengambilan keputusan. Dengan menyajikan visualisasi

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

data deteksi secara intuitif, sistem ini menjadi informatif dan mudah diakses, bahkan bagi pengguna non-teknis, untuk merencanakan tindakan pemeliharaan.

d. Implikasi Praktis dan Arah Penelitian Masa Depan
Secara keseluruhan, sistem ini menawarkan manfaat konkret untuk inspeksi bangunan rutin, terutama di area fasilitas industri vital.
1. Objektivitas, Kecepatan, dan Konsistensi: Mengurangi bias manusiawi, mempercepat proses inspeksi, dan menjamin hasil yang konsisten.
2. Dukungan Keputusan: Membantu pemilik/manajer fasilitas dalam menetapkan prioritas perbaikan berbasis data.
3. Pemeliharaan Prediktif: Memungkinkan deteksi dini untuk mencegah kerusakan yang lebih parah dan mahal.

Untuk pengembangan selanjutnya, beberapa rekomendasi diajukan:
1. Menguji robustisitas model pada berbagai kondisi lingkungan
2. Menambah variasi kelas kerusakan lain yang relevan misalnya, jamur atau rembesan air.
3. Melakukan optimasi model lebih lanjut untuk meningkatkan efisiensi komputasi pada perangkat *edge*.

4. KESIMPULAN

Penelitian ini bertujuan mengatasi permasalahan identifikasi kerusakan dinding bangunan yang selama ini dilakukan secara manual, lambat, dan berbiaya tinggi dalam jangka panjang, dengan mengembangkan sistem identifikasi cerdas berbasis algoritma YOLOv8-GFPN yang terintegrasi dengan teknologi Internet of Things (IoT) melalui perangkat ESP32-CAM. Sistem yang dirancang mampu beroperasi secara mandiri dan real-time di lingkungan lapangan, dan telah menunjukkan performa yang baik berdasarkan hasil pengujian menyeluruh terhadap aspek perangkat keras maupun perangkat lunak. Model YOLOv8s-GFPN yang digunakan berhasil mencapai tingkat akurasi deteksi tinggi dengan nilai mAP@50 sebesar 0,822 dan mAP@50-95 sebesar 0,522, serta berhasil melewati semua skenario pengujian alat dan dashboard monitoring. Meski demikian, masih ditemukan kasus misklasifikasi antara area kerusakan dan non-kerusakan, sehingga diperlukan penelitian lanjutan dengan dataset yang lebih bervariasi serta optimasi model untuk mendukung komputasi rendah daya. Pengembangan dashboard berbasis aplikasi mobile juga direkomendasikan guna meningkatkan fleksibilitas penggunaan di lapangan. Secara umum, penerapan sistem ini dapat meningkatkan kecepatan, objektivitas, dan konsistensi dalam proses inspeksi bangunan, mendukung strategi pemeliharaan prediktif, serta membantu pengambilan keputusan yang lebih tepat oleh pemilik bangunan atau pengelola fasilitas..

REFERENCES

[1] M. Putra, D. Mabui, and M. Wantoro, "Evaluasi Kinerja Bangunan Gedung Rektorat Universitas Yapis Papua Akibat Pengaruh Gempa Di Jayapura," *Cyclops J. Tek. Sipil dan Perenc.*, vol. 1, no. 2, pp. 63–74, 2023, doi: 10.55098/jtsp.v1i2.492.
[2] D. G. Gutama and R. L. Rahayu, "Ketahanan Bangunan Rumah Sakit Terhadap Bencana Gempa Bumi Di Bantul Daerah Istimewa Yogyakarta," *Lakar J. Arsit.*, vol. 4, no. 2, p. 150, 2021, doi: 10.30998/lja.v4i2.10958.
[3] S. S. Kelty Joria , Sheera Xaviera, Radha P. , Lidia B. Manalu , Erwin , Allan Jali, "Perumahan Marina Park," *JUTEKS - J. Tek. SIPIL*, vol. V, no. I, pp. 11–19, 2020, doi:10.32511/juteks.v5i1.625
[4] A. P. Wibowo, A. Adha, I. F. Kurniawan, and I. Laory, "Wall Crack Multiclass Classification: Expertise-Based Dataset Construction and Learning Algorithms Performance Comparison," *Buildings*, vol. 12, no. 12, 2022, doi: 10.3390/buildings12122135.
[5] Z. Jiang, J. I. Messner, and E. Matts, "Computer Vision Applications in Construction and Asset Management Phases: a Literature Review," *J. Inf. Technol. Constr.*, vol. 28, no. November 2022, pp. 176–199, 2023, doi: 10.36680/J.ITCON.2023.009.
[6] Yudistira Bagus Pratama and Nurzaidah Putri Dalimunthe, "Implementasi Teknik Computer Vision Untuk Deteksi Viridiplantae Pada Lahan Pasca Tambang," *Bull. Comput. Sci. Res.*, vol. 3, no. 1, pp. 64–72, 2022, doi: 10.47065/bulletincsr.v3i1.193.
[7] M. Sarosa and N. Muna, "Implementasi Algoritma You Only Look Once (YOLO) Untuk Deteksi Korban Bencana Alam," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 8, no. 4, pp. 787–792, 2021, doi: 10.25126/jtiik.202184407.
[8] S. Somantri, G. P. Insany, S. Olis, and K. Kamdan, "Perancangan Sistem Otomatisasi Pemberi Pakan Ikan Lele Berdasarkan Suhu Air Menggunakan Logika Fuzzy Sugeno," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, p. 289, 2023, doi: 10.26418/jp.v9i2.65823.
[9] M. Q. Jannah, Z. Amalia, C. N. Asyifa, and M. Afifuddin, "Implementasi Algoritma Convolutional Neural Network Untuk Identifikasi Tingkat Kerusakan Struktur Bangunan Pasca Gempa Bumi," Journal of The Civil Engineering Student, vol. 6, no. 4, pp. 212–221, 2024, doi: 10.24815/journalces.v6i4.30359
[10] R. Soekarta, S. Aras, and A. M. Fitrah Pasaribu, "Deteksi Keretakan Permukaan Gedung Menggunakan Algoritma YOLO Berbasis Web," *Insect (Informatics Secur. J. Tek. Inform.*, vol. 11, no. 1, pp. 116–126, 2025, doi: 10.33506/insect.v11i1.4339.
[11] S. Shomal Zadeh, S. Aalipour birgani, M. Khorshidi, and F. Kooban, "Concrete Surface Crack Detection with Convolutional-based Deep Learning Models," *SSRN Electron. J.*, vol. 10, no. 3, pp. 25–35, 2024, doi: 10.2139/ssrn.4661249.
[12] W. S. N. Hassanah, Y. P. Lestari, and R. A. Saputra, "Digital Image Processing to Detect Cracks in Buildings Using Naïve Bayes Algorithm (Case Study: Faculty of Engineering, Halu Oleo University)," *Telematika*, vol. 20, no. 1, p. 1, 2023, doi: 10.31315/telematika.v20i1.8925.
[13] D. S. Charismana, H. Retnawati, and H. N. S. Dhewantoro, "Motivasi Belajar Dan Prestasi Belajar Pada Mata Pelajaran Ppkn

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1290−1301
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7882

Di Indonesia: Kajian Analisis Meta," *Bhineka Tunggal Ika Kaji. Teor. dan Prakt. Pendidik. PKn*, vol. 9, no. 2, pp. 99–113, 2022, doi: 10.36706/jbti.v9i2.18333.

[14] A. Raimundo, "Crack Detection Computer Vision Project." Access Date April 2025, Available: https://universe.roboflow.com/antonio-raimundo/crack-detection-y5kyg

[15] Tim BR, "Damage-detection Computer Vision Project." Access Date April 2025, Available: https://universe.roboflow.com/tim-4ijf0/damage-detection-0otvb

[16] Pexels, "No Defect Wall." Access Date April 2025, Available: https://www.pexels.com/id-id/pencarian/Foto Dinding/

[17] M. I. Hatta, F. Yanto, I. Afrianty, and L. Afriyanti, "Pengaruh Image Enhancement Contrast Stretching dalam Klasifikasi CT-Scan Tumor Ginjal Menggunakan Deep Learning,"*Inovtek Polbeng-Seri Informatika* vol. 9, no. 1, 2024 , pp. 408–419, 2024, doi:10.35314/isi.v9i1.4233

[18] B. Khalili and A. W. Smyth, "SOD-YOLOv8 - Enhancing YOLOv8 for Small Object Detection in Traffic Scenes," *Sensors*, vol. *24, no.* 19, 2024. doi: 10.3390/s24196209.

[19] Evidently AI Team, "Mean Average Precision (MAP) in ranking and recommendations." [Online]. Available: https://www.evidentlyai.com/ranking-metrics/mean-average-precision-map

[20] A. Efendi Noor and P. Irfan, "Implementasi Progressive Web Apps (PWA) Menggunakan (Implementation of Progressive Web Apps (PWA) Using Laravel and Vue.Js in Making Freelance Service Provider Applications)," *JTIM : Jurnal Teknologi Informasi dan Multimedia,* vol. 2, no. 3, pp. 174–180, 2020, doi:10.35746/jtim.v2i3.109

[21] H. Saputra, K. Muchtar, N. Chitraningrum, A. Andria, and A. Febriana, "Performance evaluation of hyper-parameter tuning automation in YOLOV8 and YOLO-NAS for corn leaf disease detection," *Sinergi (Indonesia)*, vol. 29, no. 1, pp. 197–206, 2025, doi: 10.22441/sinergi.2025.1.018.