

Sistem Deteksi Tingkat Kerusakan Dinding Berbasis YOLOv8n-OB Untuk Mendukung Pemeliharaan

Krisna Dzakyaa Darajat¹, Arilyan Okta Hendriansyah²

Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: *¹krisnadzakyaa@gmail.com, ²arilokta97@gmail.com

Abstrak – Kekuatan struktural bangunan sangat vital bagi keselamatan, namun inspeksi visual manual sering kali bersifat subjektif dan tidak efisien dalam mendeteksi kerusakan dini seperti retak dan jamur. Penelitian ini bertujuan mengatasi masalah tersebut dengan mengusulkan sistem pemantauan kesehatan struktural otomatis berbasis *deep learning*. Metode yang digunakan adalah pendekatan kuantitatif eksperimental dengan mengimplementasikan arsitektur YOLOv8n-OB (*Oriented Bounding Box*). Varian 'nano' dipilih untuk mencapai keseimbangan optimal antara kecepatan inferensi *real-time* dan efisiensi waktu pemrosesan gambar. Model dilatih menggunakan 15.553 citra dinding yang dianotasi secara manual dalam format koordinat delapan titik. Hasil evaluasi menunjukkan performa impresif dengan nilai *mean Average Precision* (mAP@50) mencapai 90,7%, *precision* 87,2%, dan *recall* 84,9%. Secara teknis, model mampu memproses citra dengan kecepatan 15,08 ms, memungkinkan deteksi instan pada aplikasi berbasis web Streamlit yang juga menyediakan fitur estimasi luas area dan skor keparahan. Keberhasilan ini memvalidasi bahwa penggunaan YOLOv8n-OB secara signifikan meningkatkan presisi lokalisasi kerusakan dibandingkan *bounding box* standar. Implementasi sistem ini memberikan kontribusi penting dalam mengotomatisasi inspeksi bangunan, mengurangi biaya pemeliharaan, serta meminimalisir risiko kegagalan struktural melalui deteksi dini yang objektif dan aplikatif bagi masyarakat luas.

Kata Kunci: Berjamur, Dinding, Prediksi, Retak, YOLOv8n - OB

1. PENDAHULUAN

Bangunan adalah dasar utama dalam pondasi perkembangan zaman modern, menyediakan tempat untuk istirahat, keamanan, dan sebagai pendukung di kehidupan masyarakat. Namun, struktur fisik ini tidak bersifat permanen, dalam kata lain bangunan bisa rapuh/rusak. Biasanya terjadi karena paparan langsung seperti faktor lingkungan, kelembapan, beban struktural, dan usia material bisa memengaruhi pada penurunan kualitas yang tak bisa dihiraukan [1]. Memastikan kekuatan dan keamanan struktur ini bukanlah sebuah pilihan, melainkan sebuah kewajiban untuk keselamatan dan keberlanjutan ekonomi, terutama kerusakan yang tidak terdeteksi pada tahap awal telah terbukti memperbesar risiko kegagalan, misalnya pada jembatan yang mengalami penurunan kondisi struktural akibat peristiwa seismik [2].

Structural Health Monitoring (SHM) mengarah pada deteksi dini retakan permukaan, yang meski tampak remeh, seringkali menunjukkan masalah struktural lebih besar/dalam. Retakan adalah salah satu masalah utama dalam penurunan kualitas material yang mengancam struktur bangunan, sehingga deteksi awal sangat diperlukan [3]. Jika diabaikan, kerusakan kecil ini dapat berakibat kerusakan besar, yang mengancam keselamatan dan biaya perawatan menjadi mahal. Untuk mencegah hal ini, pendekatan *deep learning* terbukti bisa meningkatkan akurasi identifikasi anomali sejak fase awal sebelum kerusakan besar terjadi [4].

Dalam penerapan tradisional, proses *Structural Health Monitoring* (SHM) sangat bergantung di inspeksi visual manual. Cara ini memiliki kelemahan mendasar yang terdokumentasi dengan baik, terutama dalam subjektivitas yang tinggi dan inkonsistensi antar inspektor [5]. Proses ini tidak hanya mengandalkan banyak pekerja dan memakan waktu, tetapi juga tidak efisien untuk infrastruktur skala besar dan seringkali menempatkan petugas dalam situasi berbahaya [6].

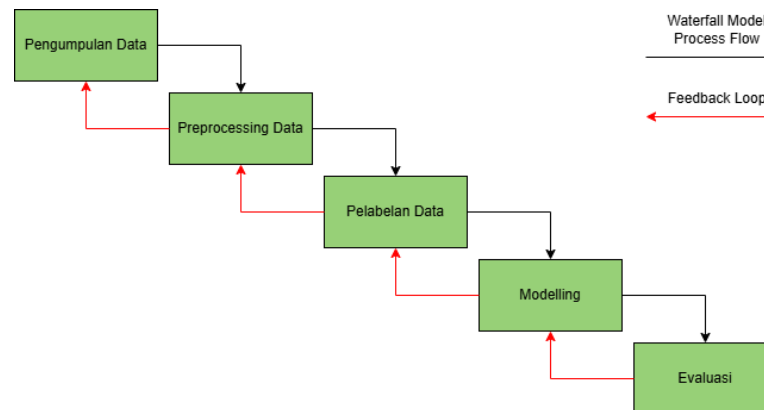
Kelemahan inilah yang memunculkan alasan penelitian kami. Di era di mana infrastruktur global semakin menua sementara tuntutan akan standar keselamatan semakin tinggi, kita membutuhkan metode pemantauan yang cerdas dan otomatis. Kemajuan pesat di bidang Kecerdasan Buatan (AI), khususnya *Deep Learning* [7], menawarkan landasan teori baru untuk mengotomatisasi tugas persepsi visual secara objektif dan kecepatan *super-human*, yang telah terbukti menjadi dasar dalam SHM modern [8].

Dalam domain ini, arsitektur telah menunjukkan keunggulan spesifik: YOLOv8n-OB [9]. YOLOv8n-OB, sebagai varian 'nano' dari arsitektur YOLO [10], yang menawarkan keseimbangan optimal antara kecepatan deteksi *real-time* dan akurasi, memungkinkannya menepatkan area 'terduga' kerusakan dengan sangat efisien [11].

Artikel ini mengusulkan sebuah sistem deteksi kerusakan tembok hibrida yang memanfaatkan keunggulan komputasi YOLOv8n OBB [10] untuk menempatkan kerusakan, mengklasifikasikannya. Penelitian ini bertujuan memudahkan pengguna dalam mengklasifikasi kerusakan tembok.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimental. Fokus utamanya adalah merancang, mengimplementasikan, dan mengevaluasi sebuah sistem berbasis *deep learning* [12] untuk deteksi dan klasifikasi kerusakan pada permukaan dinding. Alur penelitian dirancang untuk secara sistematis menguji kelayakan penggunaan arsitektur YOLOv8n-OBB [10] dalam mengatasi masalah yang diidentifikasi pada bagian pendahuluan, yaitu menggantikan inspeksi visual manual yang subjektif dan kurang efisien [5] [13]. Secara spesifik, metode ini diimplementasikan dalam beberapa tahapan yang berurutan, seperti yang ditampilkan pada Gambar 1.



Gambar 1. Diagram Waterfall Model

2.1. Pengumpulan Data

Dalam penelitian ini, data diunduh dari *Roboflow Universe*, dengan 6.233 gambar kategori jamur [14] dan 9.320 gambar kategori retak [15].

2.2. Preprocessing Data

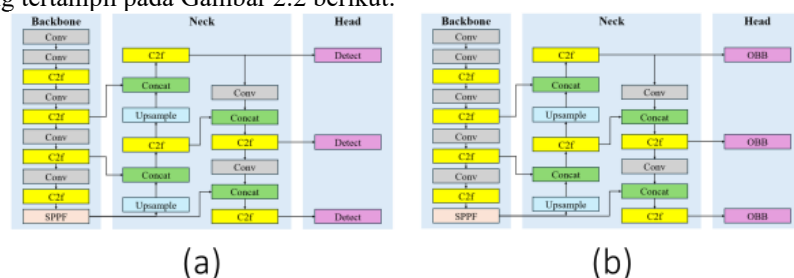
Sebelum masuk ke latih model, citra/gambar yang didapatkan akan melalui pra pemrosesan yaitu dilakukan normalisasi penskalaan kembali terhadap nilai-nilai yang tersimpan pada dataset yang membuat proses pengolahan menjadi lebih mudah [16], lalu *Resize*, artinya mengubah ukuran gambar ke dalam format *pixel* yang telah ditentukan [17].

2.3. Pelabelan Data

Data citra jamur dan retak yang sudah dikumpulkan akan digabungkan menjadi satu *dataset* untuk meningkatkan variansi fitur yang dipelajari model dalam satu ruang lingkup deteksi [18]. Proses selanjutnya adalah pelabelan data citra, dengan menggunakan format YOLO OBB [17].

2.4. Modeling

Menjawab urgensi untuk sistem pemantauan yang cerdas dan otomatis sebagaimana dijelaskan pada pendahuluan, penelitian ini mengadopsi arsitektur *You Only Look Once* (YOLO) versi 8 [19]. Dengan perbedaan gambar seperti yang tertampil pada Gambar 2.2 berikut:



Gambar 2.2 Perbedaan Arsitektur YOLOv8 Standar (a) vs Oriented Bounding Boxes (b)

Secara spesifik, model *pretrained* YOLOv8n-obb.pt dipilih sebagai pilihan yang utama. Pemilihan arsitektur ini didasarkan pada dua justifikasi teknis utama:

1. Varian *Nano* (n): Varian "*nano*" (YOLOv8n) [10] dipilih karena varian ini merupakan model terkecil dalam keluarga YOLOv8 [10]. Pilihan ini sejalan dengan argumen pada pendahuluan untuk menciptakan sistem yang efisien. Model '*nano*' [10] menawarkan keseimbangan optimal antara kecepatan inferensi yang sangat tinggi (mendukung deteksi *real-time*) [10] dan penggunaan sumber daya komputasi yang minimal, tanpa mengorbankan akurasi secara signifikan.
2. Oriented Bounding Box (obb): Penggunaan versi OBB (Kotak Pembatas Berorientasi) [9] adalah keputusan krusial. Kerusakan seperti retakan jarang sekali muncul dalam bentuk kotak horizontal atau vertikal yang sempurna. Model OBB [9] mampu menghasilkan anotasi miring yang mengikuti bentuk aktual dari kerusakan, sehingga menghasilkan lokalisasi yang jauh lebih presisi dibandingkan *bounding box* standar [20].

Meskipun pendahuluan mengusulkan sistem hibrida seperti YOLO untuk lokalisasi dan CNN [21] untuk klasifikasi, arsitektur YOLOv8 [9] itu sendiri secara umum adalah sebuah *Convolutional Neural Network* (CNN) [21] yang canggih. Model ini telah dirancang untuk melakukan kedua tugas lokalisasi (regresi *bounding box*) dan klasifikasi menentukan kelas retak (*crack*) atau berjamur (*mould*) [15] [14] secara bersamaan dalam satu tahap (*single-stage detector*) [22]. Pendekatan ini secara umum lebih efisien daripada model dua-tahap, yang sepenuhnya sejalan dengan tujuan penelitian.

2.5. Evaluasi

Untuk memvalidasi efektivitas metode yang diusulkan, kinerja model dievaluasi secara kuantitatif setelah proses pelatihan *epoch* selesai. Evaluasi ini dilakukan menggunakan set data validasi yang terpisah. Metrik evaluasi standar untuk tugas deteksi objek digunakan untuk mengukur kinerja model, sesuai dengan keluaran (output) dari proses validasi:

1. *Precision (P)*: Mengukur seberapa akurat prediksi positif yang dibuat model.
2. *Recall (R)*: Mengukur seberapa baik model dalam menemukan semua instansi kerusakan yang ada.
3. *Mean Average Precision (mAP@50)*: Metrik utama untuk deteksi objek, mengukur presisi rata-rata pada ambang batas *Intersection over Union* (IoU) sebesar 50.
4. *Mean Average Precision (mAP@50-95)*: Metrik yang lebih ketat, mengukur presisi rata-rata pada berbagai ambang batas IoU (dari 0.5 hingga 0.95).

Lalu, untuk menentukan nilai kepercayaan berdasarkan inputan 0 sampai 1. Maka, rumus matematik yang kami masukkan dalam aplikasi untuk menentukan nilai kepercayaan tersebut seperti pada contoh Persamaan 1.

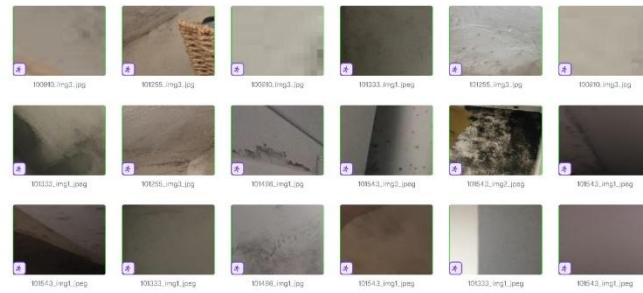
$$Pr(Class_i|Object) \times Pr(Object) \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth} \dots\dots\dots(1)$$

Persamaan 1. Rumus Menentukan Confidence Score.

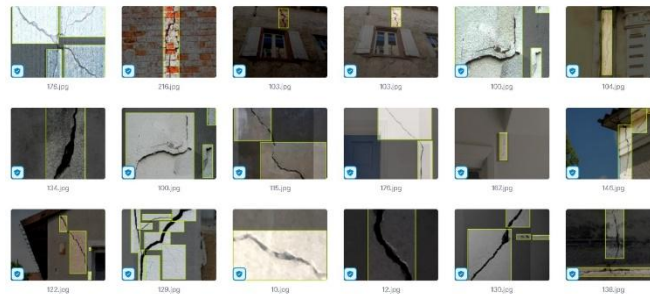
3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan Data

Pada tahap ini, dataset yang berhasil diambil terdiri dari dua kategori: jamur (*mould*) dan retak (*crack*). Total data berjumlah 15.553 citra yang diintegrasikan ke dalam satu repositori untuk melatih model dalam mendeteksi dua kelas kerusakan sekaligus. Distribusi data tersebut mencakup 6.233 citra jamur dan 9.320 citra retak. Visualisasi sampel data pada Gambar 3.1 dan Gambar 3.2 menunjukkan keberagaman kondisi dinding, mulai dari perbedaan tekstur, warna, hingga intensitas cahaya yang ekstrem. Keberagaman ini krusial agar model memiliki kemampuan generalisasi yang tinggi di lingkungan nyata tanpa bergantung pada augmentasi buatan.



Gambar 3.1. Dataset Mould dari Roboflow



Gambar 3.2. Dataset Crack dari Roboflow

Berdasarkan sampel pada Gambar 3.1 dan 3.2, dataset memiliki variansi yang signifikan dalam beberapa aspek:

1. Kondisi pencahayaan : pencahayaan yang ada di dinding dari yang rendah sampai tinggi (gelap dan terang)
2. Tekstur permukaan : jenis permukaan yang ada di dinding
3. Detail latar belakang : kompleksitas dinding tersebut, misal ada lukisan di sekitar dinding itu, dll.

3.2. Preprocessing Data

Sebelum dianalisis lebih lanjut data yang sudah di ambil dari *roboflow* akan melalui *preprocessing data*, dimana akan melewati beberapa proses yaitu:

1. Skala : mengubah nilai nilai dari gambar, supaya bisa di latih oleh model dengan baik
2. *Resize* : mengubah ukuran gambar asli menjadi ukuran yang telah ditentukan

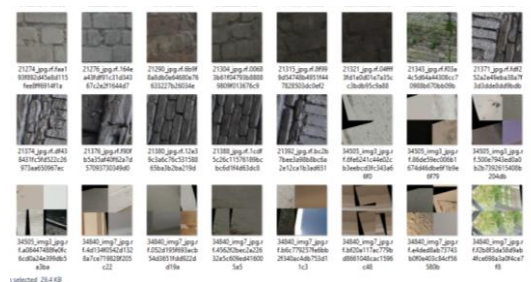
Gambar 3.3. *Resize dan Normalisasi*

Berdasarkan pada Gambar 3.3, seluruh citra dalam *dataset* dirubah ukuran dimensinya ke 640x640 piksel. Pemilihan resolusi ini didasarkan pada standar arsitektur YOLOv8n-OBb yang mengoptimalkan keseimbangan antara detail fitur pada retakan dinding yang tipis dengan kecepatan komputasi (*inference speed*). Pengurangan ukuran dari resolusi asli ke 640 piksel bertujuan untuk meringankan beban memori *GPU* tanpa menghilangkan informasi spasial penting dari kerusakan permukaan.

Selain *resizing*, dilakukan proses normalisasi nilai piksel dari rentang 0-255 menjadi 0-1. Normalisasi ini cukup penting untuk memastikan kestabilan kemiringan garis selama proses pelatihan (*training*), mempercepat konvergensi model, dan meminimalisir *bias* yang disebabkan oleh perbedaan intensitas cahaya yang ekstrem pada sampel data.

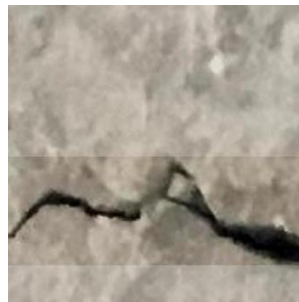
3.3. Pelabelan Data

Setelah tahap *preprocessing* selesai, langkah selanjutnya adalah melakukan integrasi dan pelabelan data secara spesifik. Dataset yang berasal dari dua sumber berbeda digabungkan menjadi satu kesatuan untuk melatih model dalam mendeteksi multi-kelas (jamur dan retak) dalam satu tahap inferensi seperti pada gambar 3.4.



Gambar 3.4. Penggabungan Data

Hasil dari penggabungan dataset ini menciptakan satu kesatuan basis data dengan total 15.553 citra yang siap dianotasi. Proses pelabelan manual dilakukan dengan menggunakan format *Oriented Bounding Box* (OBB) untuk mengakomodasi bentuk kerusakan yang tidak beraturan, terutama pada kelas retak yang memiliki orientasi kemiringan beragam. Setiap objek dalam citra direpresentasikan ke dalam format delapan koordinat ternormalisasi ($x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$). Format ini memastikan bahwa model dapat mempelajari fitur kerusakan secara presisi tanpa terganggu oleh *noise* latar belakang yang dominan seperti pada gambar 3.5.

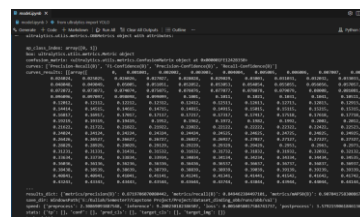


Gambar 3.5. Pelabelan Data

Berdasarkan Gambar 3.5, hasil pelabelan manual menunjukkan bahwa setiap objek kerusakan telah terlindungi oleh *bounding box* yang mengikuti kontur aktual objek. Keunggulan dari representasi delapan koordinat ini adalah kemampuannya dalam menangkap detail kerusakan *linear* (seperti retakan diagonal) yang jika menggunakan kotak *horizontal* standar akan menyertakan terlalu banyak pixel latar belakang (*background pixels*).

3.4. Modeling

Pelatihan model menggunakan YOLOv8n-OBB yang dilatih sebelumnya dilakukan dengan memanfaatkan dataset yang berada di penyimpanan lokal seperti pada Gambar 3.6 berikut:



Gambar 3.6. Proses Latih Model

Berdasarkan gambar 3.6, model menunjukkan performa yang sangat impresif dengan nilai *mean Average Precision* ($mAP@50$) mencapai 90,7%, yang membuktikan tingkat akurasi dan lokalisasi objek yang sangat tinggi. Model ini memiliki keseimbangan performa yang solid dengan nilai *Precision* sebesar 87,2% dan *Recall* sebesar 84,9%, di mana secara spesifik kelas jamur (*mould*) terdeteksi jauh lebih akurat dengan nilai mAP 90,4% dibandingkan kelas retak (*crack*) yang berada di angka 62,7%. Secara teknis, efisiensi model *nano* ini cukup unggul untuk implementasi *real-time* dengan total waktu kecepatan pemrosesan hanya 15,08 ms per gambar (terdiri dari

preprocess 2,3 ms, *inference* 9,2 ms, dan *postprocess* 3,5 ms), menjadikannya solusi otomatisasi inspeksi bangunan yang cepat sekaligus presisi.

3.5. Evaluasi

Tahap evaluasi dilakukan untuk mengukur sejauh mana model YOLOv8n-OBb mampu mengenali dan melokalisasi kerusakan dinding secara akurat setelah melalui proses pelatihan.

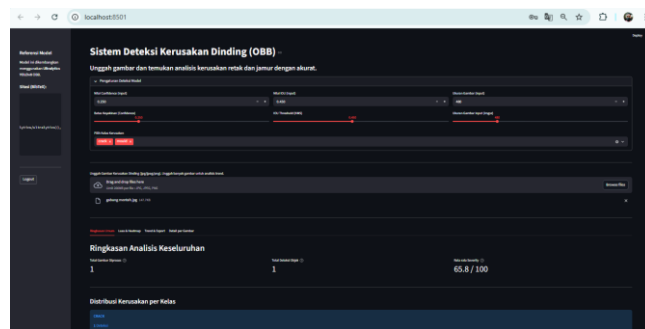
Tabel 3. 1. *Performance Modelling Retak (Crack) & Jamur (Mould)*

Class	Images	Instances	Box Precision	Box Recall	mAP50	mAP50-95
All	1534	3399	0.872	0.849	0.907	0.766
Crack	523	702	0.893	0.785	0.867	0.628
Mould	1011	2697	0.914	0.914	0.947	0.904

Berdasarkan data pada Tabel 3.1, model secara keseluruhan menunjukkan performa yang sangat impresif dengan nilai *mAP50* mencapai 0,907, yang mengindikasikan tingkat akurasi dan lokalisasi objek yang sangat tinggi dalam mendeteksi kerusakan dinding. Jika dianalisis per kategori, kelas *Mould* (Jamur) mendapatkan skor yang sangat unggul dengan nilai *mAP50* sebesar 0,947 dan *Recall* 0,914, sedangkan kelas *Crack* (Retak) menunjukkan nilai *mAP50* yang sedikit lebih rendah yaitu 0,867. Perbedaan ini secara teknis disebabkan oleh karakteristik geometris retakan yang lebih tipis dan kompleks dibandingkan bercak jamur, namun model tetap mempertahankan tingkat *Precision* yang tinggi pada angka 0,893 untuk meminimalisir kesalahan deteksi positif palsu. Pencapaian nilai *mAP50-95* sebesar 0,766 atau 76,6% untuk keseluruhan kelas membuktikan bahwa penggunaan arsitektur YOLOv8n-OBb dengan format kotak miring berhasil memberikan orientasi deteksi yang cukup presisi dan sesuai dengan bentuk aktual kerusakan di lapangan.

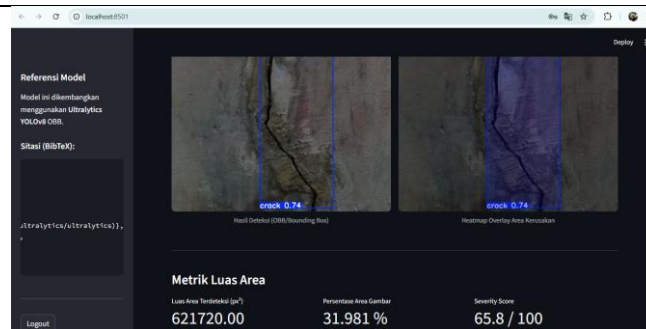
3.6. Implementasi Framework Streamlit

Secara umum, aplikasi web ini berfungsi sebagai *platform* otomatisasi inspeksi kerusakan dinding yang memungkinkan pengguna mengunggah citra permukaan bangunan, mengatur parameter deteksi secara *real-time*, dan mendapatkan analisis kerusakan yang komprehensif. Sistem tidak hanya melakukan deteksi visual, tetapi juga menghitung metrik luas area kerusakan, skor keparahan (*severity score*), serta memberikan rekomendasi tindakan perbaikan berdasarkan jenis kerusakan yang terdeteksi.



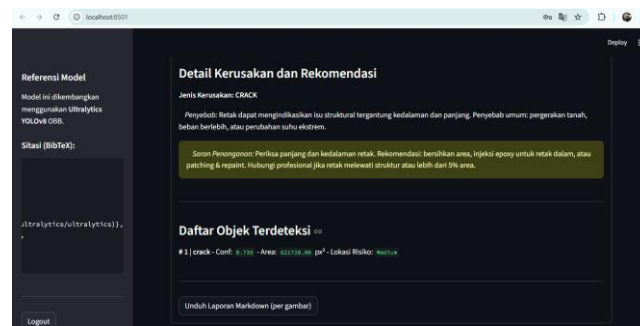
Gambar 3.7. tampilan web upload gambar dan setting parameter

Gambar 3.7 menunjukkan tampilan utama sistem di mana pengguna dapat mengunggah *file* gambar dalam berbagai format (jpg, png, jpeg). Pada bagian ini, tersedia panel pengaturan parameter deteksi seperti *Confidence Threshold* untuk mengatur batas minimal nilai kepercayaan model, dan *IoU Threshold* untuk mengatur sensitivitas tumpang tindih kotak pembatas. Fitur ini memberikan fleksibilitas bagi pengguna untuk menyesuaikan sensitivitas deteksi sesuai dengan kondisi pencahayaan atau kompleksitas tekstur dinding pada gambar yang diuji.



Gambar 3.8. tampilan gambar yang terdeteksi kerusakan dan perkiraan kerusakan

Gambar 3.8 menyajikan *output* visual dari sistem setelah proses inferensi model selesai. Sistem menampilkan dua jendela komparasi: hasil deteksi menggunakan *Oriented Bounding Box* (OBB) dan *Heatmap Overlay* area kerusakan. Di bawah hasil visual, sistem secara otomatis menghitung Metrik Luas Area dalam satuan piksel, persentase area kerusakan terhadap total gambar, serta *Severity Score* (Skor Keparahan) dalam skala 0-100. Penggabungan data statistik ini memberikan objektivitas yang lebih tinggi dibandingkan inspeksi visual manual.



Gambar 3.9. tampilan detail kerusakan dan rekomendasi

Gambar 3.9 menampilkan bagian akhir dari analisis sistem yang berisi detail teknis kerusakan dan saran penanganan. Sistem mengklasifikasikan penyebab kerusakan (seperti pergerakan tanah atau beban berlebih untuk kategori *crack*) dan memberikan rekomendasi teknis, misalnya penggunaan injeksi *epoxy* untuk retak dalam atau pembersihan area untuk kategori jamur. Pengguna juga dapat mengunduh laporan analisis secara otomatis dalam *format Markdown* untuk keperluan dokumentasi inspeksi.

4. SIMPULAN

Dengan mempertimbangkan hasil evaluasi yang dilakukan, beberapa hal dapat disimpulkan sebagai berikut:

1. Efektivitas Arsitektur YOLOv8n-OBB: Penggunaan varian YOLOv8 "*nano*" dengan format *Oriented Bounding Box* (OBB) terbukti sangat efektif untuk deteksi kerusakan dinding. Model berhasil mencapai nilai *mAP50* sebesar 90,7%, yang menunjukkan tingkat akurasi dan presisi lokalisasi yang sangat tinggi untuk objek dengan orientasi miring dan kompleks.
2. Kualitas dan Generalisasi Dataset: Penggabungan dua kategori *dataset* (6.233 citra jamur dan 9.320 citra retak) menghasilkan total 15.553 citra dengan variasi kondisi lingkungan yang nyata (pencahaya ekstrem dan tekstur beragam). Hal ini memberikan kemampuan generalisasi yang kuat pada model tanpa memerlukan proses augmentasi buatan tambahan.
3. Presisi Lokalisasi OBB pada Retakan: Implementasi *format OBB* (delapan titik koordinat) memberikan keunggulan signifikan dalam menangkap detail kerusakan *linear* seperti retakan (*crack*). Berbeda dengan *bounding box* standar, OBB mampu mengikuti kontur aktual kerusakan sehingga meminimalisir *noise* dari latar belakang, yang dibuktikan dengan nilai *Precision* kelas retak mencapai 89,3%.
4. Performa Komputasi *Real-Time*: Model YOLOv8n-OBB menunjukkan efisiensi luar biasa dengan total waktu pemrosesan hanya 15,08 ms per gambar. Kecepatan inferensi sebesar 9,2 ms memastikan sistem ini sangat layak diimplementasikan untuk kebutuhan inspeksi bangunan secara *real-time*.
5. Objektivitas Melalui Parameter Digital: Sistem yang dikembangkan mampu mengubah proses inspeksi subjektif menjadi objektif. Penggabungan fitur kalkulasi Luas Area Kerusakan dan *Severity Score* (skor keparahan) pada aplikasi memberikan data kuantitatif yang akurat bagi pemilik bangunan atau inspektur.
6. Aksesibilitas Sistem via Streamlit: Implementasi antarmuka web menggunakan *framework* Streamlit berhasil menyediakan *platform* yang inklusif dan mudah digunakan. Pengguna dapat melakukan unggah

citra, mengatur ambang batas kepercayaan (*Confidence Threshold*), hingga mengunduh laporan rekomendasi perbaikan dalam *format Markdown* secara otomatis.

7. Dampak Ekonomi dan Praktis: Secara keseluruhan, penelitian ini memberikan solusi yang mampu mengurangi biaya operasional inspeksi manual, meminimalisirkan risiko kesalahan manusia (*human error*), dan mempercepat pengambilan keputusan dalam pemeliharaan bangunan sebelum terjadi kerusakan struktural yang lebih parah.

4. SARAN

Berdasarkan hasil penelitian ini, beberapa saran untuk meningkatkan hasil deteksi kerusakan dinding yang bagus yaitu, penggabungan jenis *dataset*, seperti cat dinding yang mengelupas, perubahan warna. Lalu di pelatihan model bisa ditingkatkan lagi, dengan cara penambahan *epoch* lebih dari 100 (misal : 200) dan teknik *preprocessing* bisa ditambahkan augmentasi data (seperti : *flip data* citra). Selanjutnya di Pengujian model terhadap gambar yang memiliki kondisi pencahayaan yang berbeda, penambahan parameter seperti *patience*, *label smoothing* dapat meningkatkan kemampuan dalam *preprocessing* dan kinerja model dalam kondisi yang ekstrem. Dengan saran yang disampaikan tadi, diharapkan dapat menghasilkan sistem deteksi yang lebih *robust* dan aplikatif, terlebih di bidang arsitektur bangunan.

DAFTAR PUSTAKA

- [1] G. Wang dan J. Ke, "Literature Review on the Structural Health Monitoring (SHM) of Sustainable Civil Infrastructure: An Analysis of Influencing Factors in the Implementation," *Build. 2024 Vol 14 Page 402*, vol. 14, no. 2, hlm. 402, Feb 2024, doi: 10.3390/BUILDINGS14020402.
- [2] C. Karakostas *dkk.*, "Seismic assessment of bridges through structural health monitoring: a state-of-the-art review," *Bull. Earthq. Eng. 2023 223*, vol. 22, no. 3, hlm. 1309–1357, Nov 2023, doi: 10.1007/S10518-023-01819-3.
- [3] H. Abou-Ibrahim dan F. Hamzeh, "Understanding Stakeholders' impact on design workflow dynamics using agent-based modeling," *Autom. Constr.*, vol. 138, hlm. 104254, Jun 2022, doi: 10.1016/J.AUTCON.2022.104254.
- [4] J. Jia dan Y. Li, "Deep Learning for Structural Health Monitoring: Data, Algorithms, Applications, Challenges, and Trends," *Sensors*, vol. 23, no. 21, hlm. 8824, Okt 2023, doi: 10.3390/S23218824/S1.
- [5] F. Poli, M. F. Bado, A. Verzobio, dan D. Zonta, "Bridge structural safety assessment: a novel solution to uncertainty in the inspection practice," *Struct. Infrastruct. Eng.*, vol. 21, no. 3, hlm. 421–435, 2023, doi: 10.1080/15732479.2023.2211956;WGROU:STRING:PUBLICATION.
- [6] Z. Li, Y. Lan, dan W. Lin, "Footbridge damage detection using smartphone-recorded responses of micromobility and convolutional neural networks," *Autom. Constr.*, vol. 166, hlm. 105587, Okt 2024, doi: 10.1016/J.AUTCON.2024.105587.
- [7] Y. Lecun, Y. Bengio, dan G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, hlm. 436–444, Mei 2015, doi: 10.1038/NATURE14539;SUBJMETA.
- [8] F. Luleci dan F. N. Catbas, "A brief introductory review to deep generative models for civil structural health monitoring," *AI Civ. Eng. 2023 21*, vol. 2, no. 1, hlm. 1–11, Agu 2023, doi: 10.1007/S43503-023-00017-Z.
- [9] E. S. Jun, H. J. Sim, dan S. J. Moon, "Advancing YOLOv8-Based Wafer Notch-Angle Detection Using Oriented Bounding Boxes, Hyperparameter Tuning, Architecture Refinement, and Transfer Learning," *Appl. Sci. Switz.*, vol. 15, no. 21, Nov 2025, doi: 10.3390/app15211507.
- [10] M. Bakirci, "Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems," *Trait. Signal*, vol. 41, no. 04, hlm. 1727–1740, 2024, doi: 10.18280/ts.410407.
- [11] Y. J. Cha, R. Ali, J. Lewis, dan O. Büyüköztürk, "Deep learning-based structural health monitoring," *Autom. Constr.*, vol. 161, no. February, 2024, doi: 10.1016/j.autcon.2024.105328.
- [12] T. Wan *dkk.*, "A Survey of Deep Active Learning for Foundation Models," *Intell. Comput.*, vol. 2, Nov 2023, doi: 10.34133/ICOMPUTING.0058;REQUESTEDJOURNAL:JOURNAL:ICOMPUTING;WEBSITE:WEBSITE:SPJ-SITE;ISSUE:ISSUE:DOI.
- [13] B. D. Graham Bettis, P.E., Director, *Bridge Inspections Manual*, no. September. Texas, 2024.
- [14] YOLO-WISLR, "Mould Dataset," Roboflow Universe. Diakses: 27 November 2025. [Daring]. Tersedia pada: <https://universe.roboflow.com/yolo-wislr/mould-0hu11>
- [15] Team5, "Wall Crack Dataset," Roboflow Universe.
- [16] K. Umam dan A. Ardiansyah, "Perbandingan Metode ARIMA dan LSTM pada Prediksi Jumlah Pengunjung Perpustakaan," *MIND Multimed. Artif. Intell. Netw. Database J.*, vol. 8, no. 2, hlm. 119–129, Des 2023, doi: 10.26760/mindjournal.v8i2.119-129.

-
- [17] A. A. Rahman, S. D. Agustin, N. Ibrahim, dan N. C. Kumalasari, "Perbandingan Algoritma YOLOv4 dan Scaled YOLOv4 untuk Deteksi Objek pada Citra Termal," *MIND Multimed. Artif. Intell. Netw. Database J.*, vol. 7, no. 1, hlm. 61–71, Jun 2022, doi: 10.26760/mindjournal.v7i1.61-71.
 - [18] L. Fan *dkk.*, "Impact of Image Preprocessing and Crack Type Distribution on YOLOv8-Based Road Crack Detection," *Sensors*, vol. 25, no. 7, Mar 2025, doi: 10.3390/s25072180.
 - [19] Y. Wang dan J. He, "A Rapid Concrete Crack Detection Method Based on Improved YOLOv8," *IEEE Access*, vol. 13, hlm. 59227–59243, 2025, doi: 10.1109/ACCESS.2025.3555825.
 - [20] M. Zand, A. Etemad, dan M. Greenspan, "Oriented Bounding Boxes for Small and Freely Rotated Objects," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, doi: 10.1109/TGRS.2021.3076050.
 - [21] A. Sharma, V. Kumar, dan L. Longchamps, "Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species," *Smart Agric. Technol.*, vol. 9, hlm. 100648, Des 2024, doi: 10.1016/j.atech.2024.100648.
 - [22] R. Liu, Y. Zhu, Y. Wang, dan Z. Xing, "OB-YOLO: A UAV Image Detection Model for Reducing Computational Resource Consumption," dalam *Proceedings of the International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/IJCNN60899.2024.10649954.