

Deteksi Keretakan Permukaan Gedung Menggunakan Algoritma YOLO Berbasis Web

Rendra Soekarta¹, Suhardi Aras^{*2}, Andi Muhammad Fitrah Pasaribu³

^{1,2,3} Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong

E-mail: ¹rsoekarta@um-sorong.ac.id, ^{2*}suhardi.aras@gmail.com,

³andifitrahpasaribu21@gmail.com,

Abstrak

Keretakan permukaan gedung merupakan salah satu permasalahan yang sering terjadi pada bangunan, yang disebabkan oleh faktor-faktor seperti bencana alam, perubahan temperatur, kelembaban tinggi, dan penggunaan bahan bangunan berkualitas rendah. Permasalahan ini dapat mengancam keselamatan dan kenyamanan pengguna bangunan jika tidak segera ditangani. Metode konvensional dalam mendeteksi keretakan pada bangunan sering kali melibatkan inspeksi manual yang memakan waktu dan berisiko tinggi. Penelitian ini bertujuan untuk mengembangkan sistem deteksi keretakan permukaan gedung menggunakan algoritma YOLO (You Only Look Once) berbasis web. Sistem ini dirancang untuk meningkatkan efisiensi dan akurasi dalam mendeteksi keretakan dengan memanfaatkan teknologi pemrosesan citra dan pembelajaran mendalam (deep learning). Algoritma YOLO dipilih karena kemampuannya dalam melakukan deteksi objek secara dengan tingkat akurasi yang tinggi. Hasil evaluasi menunjukkan bahwa sistem deteksi keretakan permukaan gedung menggunakan algoritma YOLOv8 berbasis web berhasil mencapai tingkat akurasi sebesar 54,8%, dengan nilai precision sebesar 62,9%, recall sebesar 81,1%, F1-score sebesar 70,1%, dan nilai mAP sebesar 75,1%. Dengan demikian, sistem ini diharapkan dapat menjadi solusi yang efektif dan efisien dalam memantau kondisi bangunan, mengurangi risiko kecelakaan kerja, dan meningkatkan keselamatan serta kenyamanan pengguna gedung.

Kata kunci— Keretakan Permukaan gedung, Deteksi Objek, You Only Look Once, YOLO

1. PENDAHULUAN

Bangunan gedung memiliki peranan yang sangat penting dalam kehidupan sehari-hari sebagai tempat untuk berbagai aktivitas manusia, seperti tempat tinggal, kegiatan keagamaan, usaha, hingga kegiatan sosial budaya. Untuk memastikan bangunan berfungsi dengan baik, pemeliharaan dan perawatan yang terstruktur sangat diperlukan. Menurut Pemetakhiran Dapodik Tahun 2021, penilaian tingkat kerusakan bangunan di satuan pendidikan dilakukan oleh dinas pendidikan dan dinas terkait dengan menggunakan instrumen penilaian yang sesuai dengan pedoman Kementerian PUPR No. 24 Tahun 2008 mengenai pemeliharaan bangunan Gedung [1]. Fungsi bangunan gedung tidak hanya mencakup pemenuhan aspek administratif, tetapi juga melibatkan penanganan kerusakan struktural yang mungkin terjadi akibat berbagai faktor seperti bencana alam atau penggunaan bahan bangunan yang kurang sesuai [2]

Keretakan pada bangunan gedung adalah masalah yang sering terjadi dan dapat mengurangi daya tahan serta kekuatan struktur bangunan. Beberapa faktor yang menyebabkan kerusakan ini antara lain bencana alam, seperti gempa bumi, badai, atau banjir, yang dapat memberikan tekanan atau guncangan pada struktur bangunan[3]. Selain itu, perubahan suhu dan kelembaban yang ekstrem dapat mempercepat proses pelapukan pada material bangunan, seperti

kayu, dan menyebabkan retakan pada beton [2]. Penggunaan bahan bangunan yang tidak tepat atau kualitas yang rendah juga berpotensi mempercepat kerusakan bangunan tersebut[1]. Oleh karena itu, diperlukan sistem pemeliharaan yang baik dan rutin untuk menjaga agar bangunan tetap aman digunakan dan dapat bertahan dalam jangka panjang.

Universitas Muhammadiyah Sorong (UNAMIN), yang didirikan pada tahun 1984 di Kota Sorong, Papua Barat Daya, terus berkembang dengan memberikan fasilitas yang mendukung kegiatan belajar mengajar dan menciptakan lingkungan pendidikan yang nyaman bagi mahasiswa dan dosen[4]. Namun, UNAMIN menghadapi tantangan dalam memantau kondisi gedung-gedung bertingkat yang dimilikinya. Pemantauan secara manual dengan pengamatan langsung seringkali tidak cukup efektif, terutama untuk mendeteksi kerusakan yang tidak terlihat atau sulit dijangkau. Selain itu, faktor cuaca yang tidak menentu dan bencana alam turut memperburuk masalah ini, yang mengharuskan pengelola untuk mencari cara yang lebih sistematis dan terotomatisasi untuk menjaga kondisi fisik gedung dengan lebih baik [5].

Pemantauan manual terhadap kondisi bangunan sering melibatkan pekerja yang harus naik ke atap atau menggunakan scaffolding untuk melakukan inspeksi, yang berisiko tinggi terhadap keselamatan. Meskipun langkah-langkah keamanan diterapkan, pekerjaan di ketinggian tetap memiliki potensi bahaya yang besar. Risiko kecelakaan kerja dalam pemantauan konvensional ini dapat mengganggu atau menghentikan seluruh proses inspeksi. Oleh karena itu, penting untuk mencari solusi yang lebih aman dan efisien dalam pemantauan kondisi gedung. Salah satu alternatif yang semakin populer adalah penggunaan kamera smartphone untuk melakukan inspeksi jarak jauh, yang terbukti dapat mengurangi biaya, meningkatkan keandalan, dan mengurangi risiko kecelakaan[6].

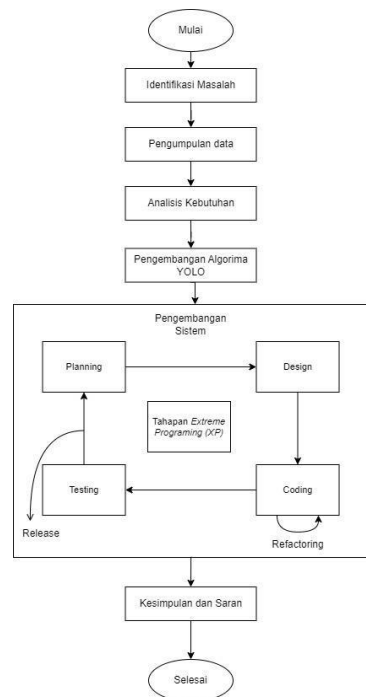
Pada penelitian sebelumnya, sistem deteksi kerusakan gedung menggunakan algoritma YOLOv5 dengan Unmanned Aerial Vehicle (UAV) telah diterapkan[7]. Namun, algoritma ini masih memiliki kekurangan, terutama dalam hal akurasi deteksi, terutama untuk objek kecil atau yang terletak jauh dari kamera. YOLOv5 juga menghadapi masalah dalam trade-off antara akurasi dan kecepatan pendeteksian, dengan varian YOLOv5s lebih cepat namun dengan akurasi lebih rendah, sementara YOLOv5x lebih akurat namun lebih lambat. Penelitian ini akan menggunakan algoritma YOLOv8 yang lebih baru[3], yang diharapkan dapat meningkatkan tingkat akurasi dalam mendeteksi kerusakan pada bangunan gedung. Dengan menggunakan YOLOv8, diharapkan hasil deteksi keretakan dapat lebih optimal, baik dari sisi akurasi maupun kecepatan[8].

Berdasarkan permasalahan dan kebutuhan yang telah dijelaskan, penulis memilih untuk mengangkat judul penelitian Deteksi Keretakan Permukaan Gedung Menggunakan Algoritma YOLO dalam penelitian ini. Tujuan penelitian adalah mengembangkan sistem deteksi keretakan pada bangunan gedung menggunakan teknologi terbaru dalam deteksi objek, yaitu YOLOv8. Penggunaan platform berbasis web diharapkan dapat mempermudah akses dan penerapan sistem ini di berbagai lokasi tanpa perlu perangkat keras yang mahal atau metode pemantauan manual yang berisiko tinggi[9]. Dengan sistem berbasis web, diharapkan dapat membantu mengurangi biaya dan meningkatkan efisiensi dalam pemantauan kondisi gedung, serta memberikan solusi yang lebih aman dan akurat dalam menjaga keselamatan dan kenyamanan penghuni bangunan[10].

2. METODE PENELITIAN

2.1 Alur Penelitian

Dalam melakukan penelitian terdapat beberapa tahapan alur penelitian yaitu dengan menentukan Identifikasi Masalah, Studi Literatur, Menentukan Variabel Dan Operasionalnya, Pengumpulan Data, Analisis Kebutuhan, Pengembangan Model You Only Look Once (YOLO), Metode Pengembangan Extreme Programming Implementasi, Pengujian Sistem, Kesimpulan dan Saran. Berikut adalah gambaran alur penelitian yang dilakukan :



Gambar 1. Alur Penelitian

Langkah awal dalam penelitian ini adalah mengidentifikasi masalah yang akan diteliti, yaitu deteksi keretakan pada gedung dengan menggunakan algoritma You Only Look Once (YOLO). Proses ini penting untuk memperjelas batasan permasalahan dan menjaga agar fokus penelitian tetap pada tujuan utamanya. Peneliti kemudian melakukan studi literatur dengan merujuk pada berbagai sumber, termasuk jurnal, skripsi, dan artikel terkait dengan machine learning, deteksi keretakan, serta YOLO, untuk memberikan pemahaman yang lebih mendalam tentang teknik dan metode yang relevan. Setelah itu, peneliti mengumpulkan data berupa gambar yang berisi objek retakan, yang telah dianotasi dan dilabeli untuk pelatihan model. Dataset ini disimpan di Google Drive dan digunakan untuk menguji algoritma YOLO, dengan harapan memperoleh akurasi yang tinggi dalam mendeteksi keretakan pada gedung.

2.2 Analisis Kebutuhan

1. Kebutuhan perangkat keras

- a. Processor : Intel Core i5 G8
- b. Ram : 12 GB
- c. Penyimpanan : 1000 GB HDD + 256 GB SSD
- d. Smartphone : Iphone 11 Pro Max

2. Kebutuhan perangkat lunak

- a. Sistem operasi : Windows 10 home 64-bit
- b. Text editor : Visual Studio Code (VSC)
- c. Browser : Google Chrome
- d. Google Collaboratory
- e. Draw.io
- f. Figma
- g. Roboflow

3. Kebutuhan dataset

- a. 431 gambar keretakan format jpg
- b. Label dan anotasi bounding box

2.3 Pengembangan Algoritma YOLO

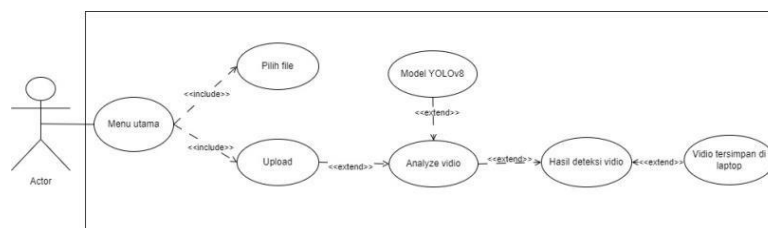
Penelitian ini bertujuan untuk menguji kemampuan model YOLOv8 dalam mendeteksi keretakan pada permukaan gedung. Langkah pertama melibatkan pengumpulan dan anotasi dataset yang berisi 431 gambar dengan satu kategori objek, yaitu retakan, yang diberi label menggunakan bounding box. Data yang digunakan mencakup variasi kondisi seperti perbedaan pencahayaan dan posisi objek. Dataset ini dibagi menjadi tiga bagian: pelatihan, validasi, dan pengujian. Pembagian ini dimaksudkan untuk memastikan bahwa model dapat bekerja dengan baik dalam berbagai kondisi serta menghindari masalah overfitting. YOLOv8 dipilih karena memberikan kombinasi yang baik antara kecepatan, akurasi, dan kemampuan deteksi objek[11].

Proses pengembangan model mencakup tahap pelatihan dengan penyesuaian parameter menggunakan subset validasi untuk menghindari overfitting. Evaluasi dilakukan dengan menggunakan berbagai metrik, seperti akurasi, presisi, recall, dan skor F1, untuk mengukur efektivitas model[12]. Setelah tahap pelatihan dan evaluasi, model yang telah dilatih disimpan dalam format PyTorch dan siap diintegrasikan ke dalam sistem berbasis web untuk deteksi keretakan. Evaluasi ini penting untuk memastikan bahwa model dapat berfungsi dengan baik dalam situasi nyata dan mampu mengenali objek yang belum pernah ditemukan sebelumnya.

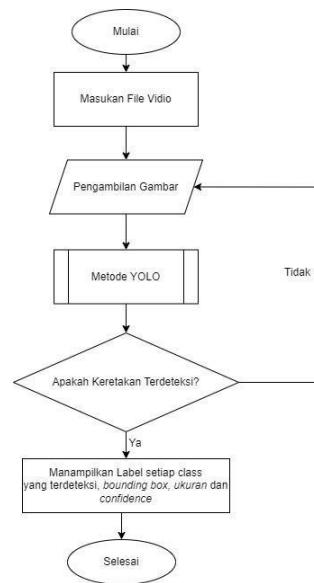
2.4 Tahapan Perancangan Sistem

Dalam penelitian ini, sistem yang digunakan mengacu pada metodologi Extreme Programming yang terdiri dari empat tahapan utama. Tahap pertama adalah Planning, di mana tujuan pengembangan diidentifikasi dengan jelas dan perencanaan terkait aktivitas pengembangan dilakukan dengan rinci dalam jangka waktu yang ditentukan. Tujuan dari tahap ini adalah untuk memastikan bahwa setiap langkah pengembangan dilakukan dengan arah yang jelas dan terstruktur, yang pada gilirannya akan memungkinkan pelaksanaan yang efisien dan pencapaian hasil yang diinginkan.

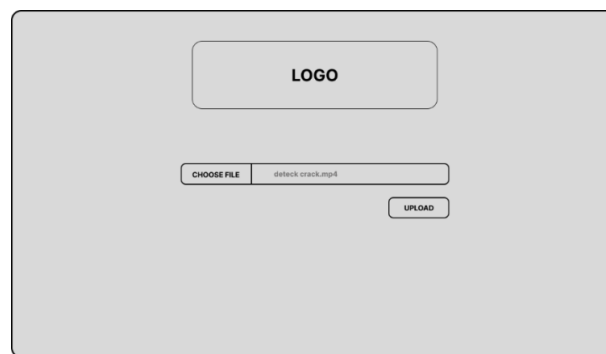
Tahap kedua adalah Design, yang mencakup perancangan struktur dan arsitektur perangkat lunak yang disesuaikan dengan kebutuhan dan preferensi pengguna. Pada tahapan ini, desain sistem dituangkan dalam bentuk diagram alir dan diagram use case, yang berfungsi untuk menggambarkan alur kerja dari sistem yang akan diterapkan. Diagram Use Case menggambarkan bagaimana pengguna dapat mengakses hasil deteksi melalui video, yang diproses oleh model untuk menghasilkan prediksi dan akurasi. Di sisi lain, flowchart aplikasi menggambarkan langkah-langkah alur program, sementara flowchart training mengilustrasikan tahapan pelatihan data dari pengumpulan hingga penyimpanan model yang telah dilatih.



Gambar 2. Use Case Diagram



Gambar 3. Flowchart Sistem



Gambar 4. User Interface

Pada tahap Coding, peneliti mulai mengembangkan aplikasi yang sesuai dengan desain yang telah ditentukan sebelumnya. Aplikasi ini dibangun dengan menggunakan algoritma YOLO, dengan antarmuka pengguna yang dikembangkan menggunakan framework Python Flask. Visual Studio dipilih sebagai text editor untuk penulisan kode. Dalam tahap ini, seluruh proses pengembangan mengikuti metodologi Extreme Programming[13], yang memang umumnya digunakan dalam proyek perangkat lunak dengan skala kecil.

Tahap terakhir adalah Testing, di mana aplikasi yang telah dikembangkan diuji melalui beberapa jenis pengujian, termasuk black box testing dan usability testing. Pada pengujian black box, pengguna melakukan interaksi langsung dengan antarmuka tanpa memperhatikan struktur internal sistem, seperti mengunggah file video untuk proses deteksi dan memverifikasi apakah sistem dapat mendeteksi objek dengan tepat. Misalnya, jika dalam video terdapat keretakan, pengujian dilakukan untuk memastikan sistem dapat mendeteksi keretakan tersebut dengan akurasi yang tinggi tanpa kesalahan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data.

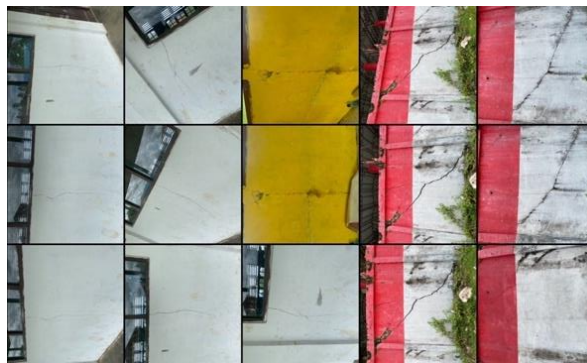
Pada penelitian ini, digunakan dataset berupa gambar keretakan permukaan bangunan

dalam format JPG yang dikumpulkan secara mandiri oleh peneliti. Dataset ini terdiri dari 431 gambar yang digunakan untuk melatih dan menguji model deteksi keretakan permukaan bangunan. Penggunaan dataset yang cukup besar ini diharapkan dapat meningkatkan keakuratan dan kehandalan model deteksi keretakan yang dikembangkan dalam penelitian ini

3.2 Implementasi Algoritma YOLO

1. Pengumpulan Dataset

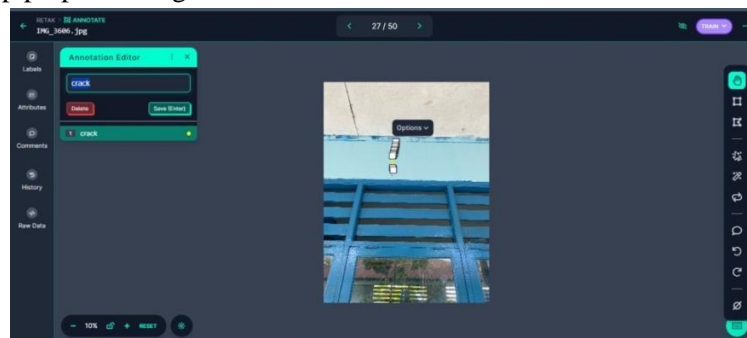
Penelitian ini menggunakan dataset berupa gambar keretakan permukaan bangunan dalam format JPG, yang dikumpulkan oleh peneliti secara mandiri. Setelah dikumpulkan, dataset tersebut diunggah ke Google Drive untuk disimpan dan digunakan dalam lingkungan Google Colaboratory, yang merupakan platform yang umum digunakan untuk pengembangan dan pelatihan model machine learning. Selanjutnya, dataset dimasukkan ke website Roboflow, sebuah platform yang dirancang khusus untuk membantu dalam proses pelabelan (labeling) data, termasuk memberikan label pada kelas keretakan pada gambar-gambar tersebut. Dengan demikian, dataset dapat diolah dan digunakan secara efisien dalam proses pelatihan model deteksi keretakan.



Gambar 1 Dataset Keretakan

2. Preprocessing Data

Pada tahap ini, dilakukan anotasi atau labelling dataset di media Roboflow untuk melakukan selection pada gambar keretakan menggunakan polygon tool yang bertujuan untuk mempermudah machine learning dalam melakukan objek deteksi. Pada tahap ini juga dilakukan pembagian data yang akan digunakan untuk training model yang terdiri dari data training, validation, dan test. Berikut tahap preprocessing dataset di Roboflow.



Gambar 5. Proses Anotasi Data

3. Split Data

Pada tahap ini, setelah dilakukan anotasi pada gambar keretakan permukaan bangunan, gambar-gambar tersebut akan dibagi ke dalam tiga folder: train, test, dan validasi.

4. Training

Pada tahap ini, model yang telah dirancang akan dilatih menggunakan data pelatihan. Proses pelatihan dengan melakukan iterasi melalui epoch dan batch. Setiap epoch mewakili satu siklus melalui seluruh data pelatihan.

```

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
226/250  2.28G    0.6093    0.5252    0.9081    10         640: 100% 14/14 [00:03<00:00, 4.01it/s]
Class   Images  Instances  Box(P)    R         mAP50  mAP50-95): 100% 2/2 [00:00<00:00, 2.31it/s]
all     61      70        0.743    0.787    0.688

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
227/250  2.28G    0.6007    0.5398    0.9131    10         640: 100% 14/14 [00:03<00:00, 3.99it/s]
Class   Images  Instances  Box(P)    R         mAP50  mAP50-95): 100% 2/2 [00:00<00:00, 2.17it/s]
all     61      70        0.827    0.751    0.686

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
228/250  2.28G    0.6487    0.5801    0.9213    9          640: 100% 14/14 [00:04<00:00, 2.84it/s]
Class   Images  Instances  Box(P)    R         mAP50  mAP50-95): 100% 2/2 [00:00<00:00, 2.36it/s]
all     61      70        0.827    0.743    0.598

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
229/250  2.28G    0.61      0.5942    0.9101    4          640: 100% 14/14 [00:03<00:00, 4.15it/s]
Class   Images  Instances  Box(P)    R         mAP50  mAP50-95): 100% 2/2 [00:00<00:00, 2.07it/s]
all     61      70        0.815    0.729    0.599

Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 179, best model saved as best.
To update EarlyStopping(patience=50) pass a new patience value, i.e. 'patience=300' or use 'patience=0' to disable EarlyStopping.

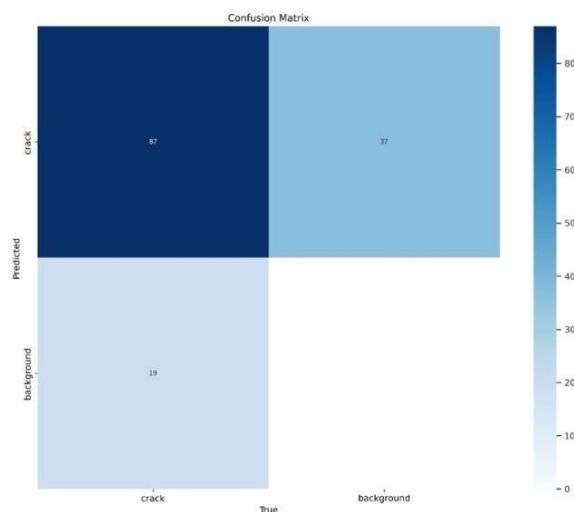
229 epochs completed in 0.359 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.3MB

```

Gambar 6 Training Model

5. Evaluasi Kinerja Model

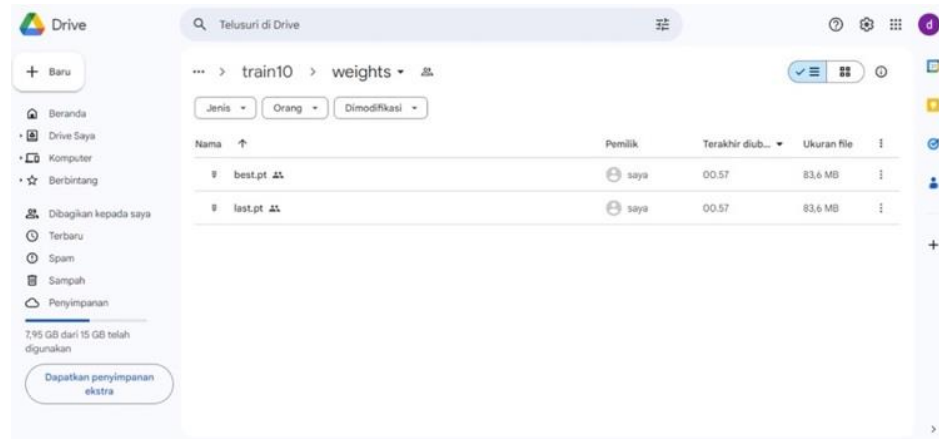
Confusion Matrix digunakan untuk memberikan pemahaman tentang kinerja model dalam melakukan tugas deteksi objek. Dalam konteks deteksi objek, confusion matrix memberikan informasi tentang sejauh mana model dapat mengenali dan membedakan objek yang ada dalam gambar/video. True Positive dan True Negative memberikan informasi ketika classifier melakukan pendeteksian data bernilai benar, sedangkan False Positive dan False Negatif memberikan informasi salah dalam mendeteksi. Confusion matrix dari YOLO v8 ini didapatkan dari hasil perbandingan antara label yang diberikan dan label yang diprediksi oleh sistem. Matriks ini menunjukkan hasil prediksi yang benar dan salah dari model.



Gambar 7. Confusion Matrix

6. Save Model

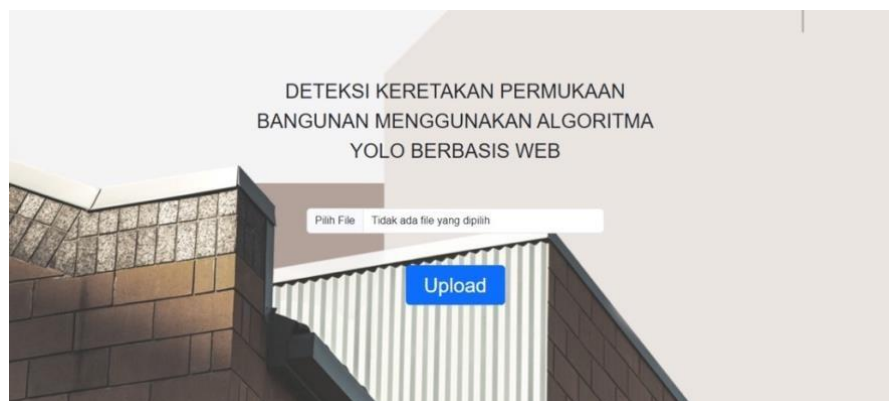
Pada tahap ini, setelah proses training dan hasil dari training akan disimpan dengan nama best.pt, yang merupakan file yang akan sistem gunakan untuk mendeteksi keretakan permukaan bangunan pada inputan gambar, proses penyimpanan best.pt ini dapat dilihat pada gambar 5.



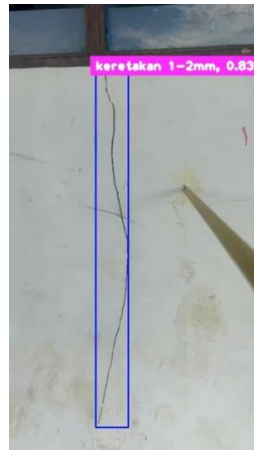
Gambar 8. Save Model

3.3 Implementasi Metode Extreme Programming

Dalam implementasi metode *Extreme Programming*, pembuatan sistem telah direalisasikan dalam bentuk aplikasi web yang menggunakan bahasa pemrograman Python untuk logika backend, JavaScript untuk interaksi dinamis di sisi klien, serta HTML dan CSS dengan Bootstrap framework untuk tata letak dan desain yang responsif. Halaman web yang dihasilkan, seperti yang terlihat pada gambar menunjukkan antarmuka pengguna yang dirancang untuk memudahkan pengguna dalam mengakses dan menggunakan sistem deteksi keretakan permukaan bangunan. Desain yang bersih dan responsif memastikan bahwa pengguna dapat berinteraksi dengan aplikasi tanpa hambatan, sementara fitur-fitur yang diimplementasikan, seperti pemrosesan gambar untuk deteksi keretakan, memberikan nilai tambah yang signifikan dalam pemantauan dan pemeliharaan bangunan. Dengan demikian, hasil dari pembuatan sistem ini tidak hanya mencakup aspek fungsional, tetapi juga estetika dan kegunaan yang dapat meningkatkan pengalaman pengguna secara keseluruhan.



Gambar 9. Tampilan Web



Gambar 10. Hasil Deteksi

1. Balckbox Testing

Pada tahapan ini dilakukan pengujian sistem. Pengujian ini dilakukan untuk memastikan bahwa program dapat berjalan dengan baik saat digunakan. Dari setiap tes yang dilakukan untuk menutup kemungkinan terdapat kesalahan dari sistem yang sudah di tes, namun dengan dilakukannya pengujian ini setidaknya dapat meminimalisir yang ada pada sistem. Pengujian pada tabel dibawah ini, dilakukan sebagai berikut.

Tabel 1. Blackbox Testing

Uji Kasus	Skenario Uji Coba	Hasil Pengujian	Kesimpulan
Klik "PILIH FILE" dan pilih file video	Nama file muncul di bidang input	Nama file ditampilkan	Valid
Klik "PILIH FILE" lalu pilih file non-video dan Klik "UPLOAD"	Pesan kesalahan tentang tipe file yang tidak valid	Pesan kesalahan ditampilkan	Valid
Klik "UPLOAD" tanpa memilih file	Pesan kesalahan tentang file yang tidak ada	Pesan kesalahan ditampilkan	Valid
Pilih file video dan klik "UPLOAD"	Vidio akan diproses dan dilakukan deteksi serta akan disimpan otomatis di laptop	Hasil vidio deteksi	Valid

2. Usability Testing

Untuk mengetahui seberapa puas pengguna dengan layanan atau produk, peneliti membuat sebuah survei singkat menggunakan Google Forms. Survei ini berisi lima pertanyaan yang bertujuan untuk memahami pendapat pengguna. Cara ini memudahkan mendapat masukan dari banyak orang dengan cepat dan rapi, sehingga peneliti bisa tahu apa yang disukai atau perlu diperbaiki. Berdasarkan hasil perhitungan, perolehan interpretasi skor yang didapat yaitu lebih dari 83,4% dengan keterangan Sangat Setuju, maka dapat disimpulkan bahwa pembuatan aplikasi deteksi keretakan pada permukaan bangunan menggunakan algoritma yolo ini sudah sesuai dengan tujuan yang ingin dicapai.

4. KESIMPULAN

Berdasarkan hasil pengujian dan implementasi yang telah di lakukan untuk mendeteksi keretakan permukaan bangunan menggunakan algoritma You Only Look Once (YOLO) berbasis website, maka di peroleh beberapa kesimpulan yaitu Implementasi deteksi keretakan

pada permukaan gedung menggunakan algoritma yolo v8 telah berhasil diwujudkan dalam bentuk sistem berbasis website. Dengan menerapkan metode Black Box Equivalence Partitioning, sistem mencatat keberhasilan pada semua 4 kasus uji, membuktikan keandalan dan kemampuannya dalam mendeteksi keretakan secara akurat. Lebih lanjut, evaluasi penggunaan sistem melalui metode Usability menghasilkan tingkat kesepakatan sebesar 83,4% dengan respons “Sangat Setuju” dari pengguna, menunjukkan bahwa sistem ini efektif dan mudah digunakan. Dalam implementasi deteksi keretakan pada permukaan gedung mendapatkan tingkat akurasi sebesar 60,8%, untuk precision sebesar 70,1%, untuk recall sebesar 82,07%, untuk f1-score sebesar 75,4% dan untuk mAP sebesar 81,6%..

5. SARAN

Beberapa saran untuk penelitian selanjutnya meliputi penggunaan arsitektur selain YOLO untuk mengeksplorasi potensi model lain dalam deteksi objek. Pengembangan algoritma yang lebih canggih dan akurat juga dapat meningkatkan kinerja sistem secara signifikan. Selain itu, penambahan kelas jenis keretakan akan memungkinkan deteksi untuk berbagai tipe keretakan, yang dapat memperluas aplikasi sistem. Selanjutnya, penelitian dapat dikembangkan ke dalam aplikasi Android untuk mempermudah akses dan penggunaan oleh masyarakat. Selain itu, menambahkan kemampuan deteksi objek dan pengukuran selain vertikal juga dapat meningkatkan cakupan dan akurasi sistem dalam mengidentifikasi berbagai kondisi keretakan.

DAFTAR PUSTAKA

- [1] J. Aprilyanto And Yohannes, “Implementasi Arsitektur Vgg-Unet Dalam Melakukan Segmentasi Keretakan Pada Citra Bangunan,” *Mdp Student Conference 2023*, 2023, Doi: <https://doi.org/10.35957/mdp-sc.v2i1.4311>.
- [2] A. S. Ariyanto, “Analisis Jenis Kerusakan Pada Bangunan Gedung Bertingkat (Studi Kasus Pada Gedung Apartemen Dan Hotel Candiland Semarang),” 2020.
- [3] F. Marleny, Muhammad Fitriansyah, Sa’adah, Winda Astria Nuansa Saputri, Rudy Ansari, And Mambang, “Segmentasi Citra Keretakan Dinding Beton Menggunakan Teknik Perbandingan Evaluasi Metrik,” *Tematik*, Vol. 10, No. 1, Pp. 28–33, Jun. 2023, Doi: [10.38204/tematik.v10i1.1261](https://doi.org/10.38204/tematik.v10i1.1261).
- [4] S. Mutmainah, R. Soekarta, And I. Amri, “Rancang Bangun Aplikasi Kkn Pada Kampus Universitas Muhammadiyah Sorong Berbasis Web Dan Android,” Vol. 01, No. 02, 2023.
- [5] Sakti Maulana Muhidin, Rendra Soekarta, Teguh Hidayat Iskandar Alam, And Nirwana Nurdjan, “Rancang Bangun Sistem Alat Pemilah Telur Ayam Siap Jual Menggunakan Mikrokontroler Arduino Dan Firebase Berbasis Android,” *Framework*, Vol. 02, 2023, Doi: <https://doi.org/10.33506/jiki.v2i01.3046>.
- [6] B. Sasmito, B. H. Setiadji, And R. Isnanto, “Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning Di Kota Semarang,” *Teknik*, Vol. 44, No. 1, Pp. 7–14, May 2023, Doi: [10.14710/teknik.v44i1.51908](https://doi.org/10.14710/teknik.v44i1.51908).
- [7] Muhammad Aqil Rayhan Majid, Mohamad Abdul Hady, Mochammad Sahal, And Yurid Eka Nugraha, “Sistem Deteksi Dan Tracking Keretakan Bangunan Dengan Unmanned Aerial Vehicle Menggunakan Algoritma Cnn,” No. Vol. 3 No. 1 (2023): Snte Ii, 2023, Accessed: Mar. 29, 2025. [Online]. Available: <https://snte.fortei.org/list/index.php/snte/article/view/42>
- [8] M. Yue, L. Zhang, J. Huang, And H. Zhang, “Lightweight And Efficient Tiny-Object Detection Based On Improved Yolov8n For Uav Aerial Images,” *Drones*, Vol. 8, No. 7, Jul. 2024, Doi: [10.3390/drones8070276](https://doi.org/10.3390/drones8070276).
- [9] E. Setiadi And A. Wibowo, “Klasifikasi Dan Deteksi Keretakan Pada Trotoar Menggunakan Metode Convolutional Neural Network,” 20203. Doi: <https://doi.org/10.51988/jtsc.v4i1.116>.
- [10] Y. Yulianto And A. Wibowo, “Deteksi Keretakan Jalan Aspal Menggunakan Metode Convolutional Neural Network,” 2023.

- [11] A. A. Permana, R. Muttaqin, And A. Sunandar, “Sistem Deteksi Api Secara Real Time Menggunakan Algoritma You Only Look Once (Yolo) Versi 8,” 2024. [Online]. Available: <https://Universe.Roboflow.Com/>.
 - [12] R. Gilang Wijanarko, A. Ichsan Pradana, And D. Hartanti, “Implementasi Deteksi Drone Menggunakan Yolo (You Only Look Once),” 2024.
 - [13] S. Supriyatna, “Penerapan Metode Extreme Programming Dalam Perancangan Sistem Layanan Pengaduan Masyarakat Berbasis Web,” *Jitu: Jurnal Informatika Utama*, Vol. 1, No. 2, 2023, Doi: 10.55903/Jitu.V1i2.168.
-