



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 2 (2026) pp: 21125-21132

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Implementasi Sistem Deteksi dan Klasifikasi Cacat Produk Berbasis Web Menggunakan Algoritma YOLOv8

Anggarini Puspita Rarasati, Wiwin Winarti

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang

puspitararasati25@gmail.com, dosen02374@unpam.ac.id

Abstrak

Proses quality control pada produk metal cutting dies umumnya masih dilakukan melalui inspeksi visual secara manual, sehingga rentan terhadap ketidakkonsistenan, kelelahan, dan kesalahan manusia, terutama pada industri dengan volume produksi tinggi. Kondisi ini mendorong kebutuhan akan sistem inspeksi otomatis yang lebih cepat, objektif, dan konsisten dibandingkan pengamatan manual. Penelitian ini bertujuan mengembangkan sistem berbasis web untuk mendeteksi dan mengklasifikasi cacat produk menggunakan algoritma YOLOv8. Pendekatan yang digunakan memanfaatkan object detection untuk mengidentifikasi area cacat pada permukaan produk, kemudian menentukan kondisi akhir produk ke dalam dua kategori yaitu defect dan non-defect berdasarkan confidence thresholding dengan ambang batas nilai confidence sebesar 50%. Tahapan penelitian meliputi studi pustaka, pengumpulan dataset, anotasi citra menggunakan platform Roboflow, pra-pemrosesan dan augmentasi data, pelatihan model, evaluasi model, perancangan antarmuka web, serta pengujian fungsional sistem secara menyeluruh. Model YOLOv8n (Nano) dilatih menggunakan dataset sebanyak ± 200 gambar dengan transfer learning dari checkpoint publik MS COCO pada lingkungan komputasi berbasis CPU, dan menghasilkan nilai Precision sebesar 51,5% pada epoch terbaik ke-31, Recall sebesar 44,3%, dan mAP@0.5 sebesar 39,3%. Sistem web berbasis Flask yang dibangun mampu menampilkan bounding box pada area cacat beserta status akhir produk secara otomatis. Pengujian fungsionalitas menggunakan Black Box Testing menunjukkan seluruh skenario, baik positif maupun negatif, berhasil berjalan sesuai rencana dengan tingkat keberhasilan 100%, menegaskan bahwa pendekatan ini dapat diintegrasikan ke antarmuka sederhana tanpa latar belakang teknis kecerdasan buatan. Penelitian ini diharapkan mendukung inspeksi kualitas secara lebih cepat, konsisten, dan efisien, sekaligus menjadi dasar pengembangan lanjutan dengan dataset lebih besar dan akselerasi GPU.

Kata kunci: Cutting Dies, Quality Control, YOLOv8, Deteksi Cacat, Sistem Berbasis Web

1. Latar Belakang

Perkembangan teknologi digital dan otomatisasi industri telah meningkatkan kebutuhan terhadap sistem inspeksi kualitas yang cepat, akurat, dan konsisten. Proses quality control pada industri manufaktur, khususnya pada komponen metal seperti cutting dies insole, masih banyak dilakukan dengan metode inspeksi visual manual. Metode manual memiliki keterbatasan berupa subjektivitas penilaian, kelelahan operator, waktu inspeksi yang lama, dan ketidakkonsistenan hasil antar pemeriksa. Sistem manual pada computer vision sering gagal menjaga stabilitas dan akurasi inspeksi ketika diaplikasikan pada lingkungan industri yang dinamis [1].

Fenomena tersebut juga terjadi pada industri manufaktur di Indonesia, khususnya sektor die casting otomotif. Inspeksi manual pada produk hasil cutting dies masih banyak mengandalkan pengamatan visual operator, yang menyebabkan tingkat kesalahan tinggi dan inkonsistensi antar operator, sementara penerapan algoritma YOLO terbukti mampu meningkatkan akurasi deteksi hingga precision 0,94 dan mAP@0,5 sebesar 0,90 [2]. Cutting dies insole berbahan plat besi memiliki struktur kompleks dan melalui proses pemanasan serta pengelasan yang berpotensi menimbulkan cacat permukaan seperti burn marks, irregular edges, karat, dan patah struktur, sehingga memerlukan inspeksi permukaan yang lebih presisi [3].

Seiring kemajuan deep learning, pendekatan object detection menjadi solusi utama dalam inspeksi otomatis. YOLO (You Only Look Once) merupakan algoritma deteksi objek yang populer karena mampu berjalan cepat secara real-time, dan telah berevolusi dari versi v1 hingga v8 dengan peningkatan signifikan pada arsitektur, akurasi, serta kemampuan generalisasi [4]. YOLOv8 sebagai versi terbaru dirancang dengan arsitektur yang lebih efisien dan performa yang lebih tinggi, sehingga sesuai diterapkan pada berbagai jenis citra dan cacat permukaan [5].

Penerapan deep learning secara umum, termasuk arsitektur YOLO, telah terbukti berkembang pesat pada berbagai domain permasalahan yang melibatkan pengenalan objek pada citra digital [12]. Selain untuk inspeksi cacat produk, keluarga algoritma YOLO juga telah diterapkan pada berbagai kasus lain seperti penghitungan kendaraan secara real-time melalui object tracking [7], deteksi objek kecil semacam pin del pada jalur listrik menggunakan YOLOv7 [10], hingga kajian sistematis terhadap pengembangan YOLO pada citra multispektral yang menegaskan fleksibilitas arsitektur ini pada berbagai jenis sensor dan domain aplikasi [9]. Keberhasilan penerapan YOLO pada berbagai domain tersebut menjadi salah satu dasar pertimbangan pemilihan algoritma YOLOv8 pada penelitian ini, dengan harapan karakteristik object detection yang cepat dan ringan tetap relevan diterapkan pada inspeksi permukaan metal cutting dies insole meskipun dengan keterbatasan sumber daya komputasi.

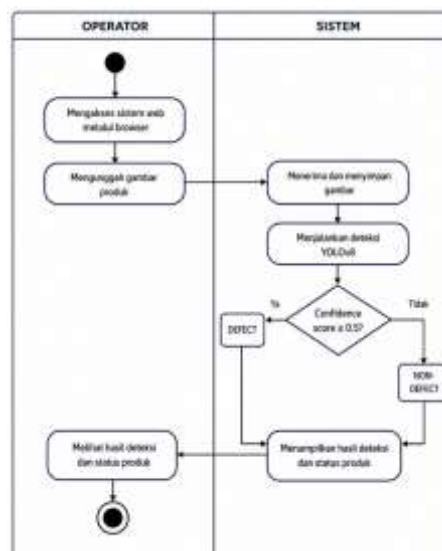
Berdasarkan permasalahan inspeksi manual, karakteristik objek cutting dies yang kompleks, serta perkembangan teknologi deep learning, penelitian ini mengimplementasikan algoritma YOLOv8 untuk mendeteksi area cacat pada permukaan metal cutting dies insole. Hasil deteksi digunakan untuk menentukan kondisi akhir produk ke dalam dua kategori, yaitu defect dan non-defect, yang kemudian diintegrasikan ke dalam sistem berbasis web agar proses inspeksi dapat dilakukan secara otomatis, konsisten, dan mudah digunakan oleh operator, sehingga diharapkan mampu meningkatkan efisiensi proses quality control pada lingkungan industri.

1. Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimental yang meliputi studi pustaka, observasi lapangan di PT Jomu Studio Indonesia, pengumpulan dataset, anotasi citra, preprocessing dan augmentasi data, pelatihan model YOLOv8, pembuatan sistem web, serta pengujian model dan sistem. Tahapan tersebut dilakukan secara berurutan untuk menghasilkan sistem deteksi dan klasifikasi cacat produk yang dapat diandalkan.

2.1. Alur Sistem yang Diusulkan

Sistem yang diusulkan bekerja dengan alur berikut: operator mengunggah gambar produk metal cutting dies insole melalui halaman web, kemudian sistem memproses gambar menggunakan model YOLOv8n yang telah dilatih untuk mendeteksi area cacat pada permukaan produk. Hasil deteksi ditampilkan dalam bentuk bounding box beserta nilai confidence score, dan sistem mengklasifikasikan kondisi akhir produk berdasarkan confidence thresholding dengan ambang batas 0,5. Produk diklasifikasikan sebagai defect apabila terdapat deteksi dengan confidence score $\geq 0,5$, dan non-defect apabila tidak ada deteksi yang memenuhi ambang batas tersebut. Alur sistem usulan ditampilkan pada Gambar 1.



Gambar 1. Alur Sistem Usulan

2.2. Dataset dan Anotasi

Dataset penelitian terdiri dari ± 200 gambar produk metal cutting dies insole yang dikumpulkan langsung oleh peneliti di PT Jomu Studio Indonesia dari berbagai sudut, jarak, dan kondisi pencahayaan. Jenis cacat yang terdapat pada dataset meliputi weld burn marks dan karat (rust). Komposisi dataset ditampilkan pada Tabel 1.

Tabel 1. Komposisi Dataset

Komponen	Detail
Total gambar	±200 gambar
Sumber data	Data primer (pengambilan langsung)
Jenis cacat dideteksi	Weld Burn Marks & Karat (Rust)
Format label	Bounding box (1 class: defect)
Platform anotasi	Roboflow
Pembagian data	70% train / 20% valid / 10% test

Anotasi dataset dilakukan menggunakan platform Roboflow, setiap gambar yang mengandung cacat diberi bounding box pada area cacat dengan satu kelas label yaitu defect, sedangkan gambar non-defect dianotasi tanpa bounding box. Tampilan anotasi dataset pada Roboflow ditunjukkan pada Gambar 2. Pemilihan Roboflow sebagai platform anotasi juga didasari oleh kemudahan integrasinya dalam alur kerja pengumpulan dan pengelolaan dataset berbasis computer vision, sebagaimana ditunjukkan pada implementasi deteksi status pintu otomatis yang turut memanfaatkan platform yang sama [11].



Gambar 2. Tampilan Anotasi Dataset pada Roboflow

2.3. Preprocessing, Augmentasi, dan Pelatihan Model

Dataset melalui tahap preprocessing berupa resize gambar ke ukuran 512×512 piksel dan normalisasi nilai piksel. Augmentasi data diterapkan otomatis oleh framework Ultralytics pada data latih, meliputi horizontal flip, HSV jitter, scale, translate, mosaic, random erasing, dan RandAugment, untuk memperluas variasi data dan mengurangi risiko overfitting. Pelatihan model dilakukan menggunakan algoritma YOLOv8 varian Nano (YOLOv8n) dengan framework Ultralytics dan PyTorch pada lingkungan komputasi lokal berbasis CPU, menggunakan transfer learning dari checkpoint publik MS COCO. Parameter pelatihan ditampilkan pada Tabel 2.

Tabel 2. Parameter Pelatihan Model YOLOv8

Parameter	Nilai
Algoritma	YOLOv8n (Nano)
Framework	Ultralytics + PyTorch (CPU)
Jumlah epoch	200 (early stopping patience 100)
Image size	512×512
Jumlah class	1 (defect)
Optimizer	Auto
Transfer learning	Checkpoint publik MS COCO

2.4. Perancangan Sistem Web dan Pengujian

Sistem web dikembangkan menggunakan framework Flask (Python) sebagai backend dan HTML sebagai frontend, memungkinkan operator mengunggah gambar, menjalankan deteksi otomatis, dan melihat hasil deteksi berupa bounding box beserta status akhir produk. Logika klasifikasi berdasarkan confidence thresholding ditampilkan pada Tabel 3. Pengujian sistem dilakukan menggunakan metode Black Box Testing pada skenario positif dan negatif untuk memastikan seluruh fitur berjalan sesuai rancangan, sedangkan performa model dievaluasi menggunakan metrik precision, recall, dan mAP@0.5. Pemilihan Python sebagai bahasa pemrograman utama didasarkan pada fleksibilitas dan kematangan ekosistemnya untuk aplikasi computer vision maupun pengembangan web [13], sedangkan Flask dipilih karena sifatnya yang ringan dan mudah diintegrasikan dengan model machine learning yang telah dilatih, sejalan dengan penerapan Flask pada pengembangan sistem informasi berbasis web lainnya [14]. Rancangan alur interaksi pengguna dengan sistem, mulai dari proses unggah gambar hingga penampilan hasil deteksi, disusun mengacu pada pendekatan pemodelan berorientasi objek menggunakan diagram UML sebagai panduan perancangan use case dan alur sistem [17].

Secara arsitektur, sistem web dirancang dengan pola client-server sederhana, di mana sisi client menyediakan formulir unggah gambar dan menampilkan hasil deteksi, sedangkan sisi server (Flask) menangani proses

penerimaan file, pemanggilan model YOLOv8 yang telah dilatih untuk melakukan inferensi, penggambaran bounding box pada citra, penentuan status akhir produk berdasarkan confidence thresholding, dan pengembalian hasil ke halaman web dalam bentuk gambar beranotasi beserta label status. Pendekatan ini dipilih karena sederhana untuk diimplementasikan pada skala prototipe penelitian, mudah dikembangkan lebih lanjut, dan tidak memerlukan infrastruktur server yang kompleks, sehingga sesuai dengan tujuan penelitian untuk menghasilkan purwarupa sistem inspeksi yang dapat langsung diujicobakan.

Tabel 3. Logika Klasifikasi Berdasarkan Confidence Thresholding

Kondisi Deteksi Model	Hasil Klasifikasi
Ada deteksi dengan confidence $\geq 0,5$	DEFECT
Tidak ada deteksi / confidence $< 0,5$	NON-DEFECT

Pemilihan metode Black Box Testing pada penelitian ini mengacu pada praktik pengujian fungsional yang umum digunakan untuk memvalidasi perilaku aplikasi berbasis web dari sisi input dan output tanpa perlu meninjau struktur kode program secara internal, sebagaimana diterapkan pada pengujian aplikasi peminjaman kendaraan operasional kantor [15]. Pendekatan ini dipandang sesuai untuk memvalidasi bahwa seluruh fitur utama sistem, yaitu unggah gambar, proses deteksi, dan penampilan status akhir produk, berjalan sebagaimana mestinya baik pada skenario input yang valid maupun tidak valid.

2.5. Lingkungan dan Alat Penelitian

Penelitian ini dilakukan menggunakan perangkat keras berupa laptop dengan prosesor Intel Core i5 dan RAM 8 GB tanpa akselerasi GPU khusus, sehingga seluruh proses pelatihan model dijalankan pada mode CPU. Perangkat lunak yang digunakan meliputi Python sebagai bahasa pemrograman utama, framework Ultralytics YOLOv8 untuk pelatihan dan inferensi model, PyTorch sebagai backend deep learning, Roboflow untuk anotasi dan manajemen dataset, serta Flask untuk membangun antarmuka sistem berbasis web. Visual Studio Code digunakan sebagai code editor, sedangkan Google Colaboratory sesekali dimanfaatkan untuk menjalankan eksperimen tambahan. Pemilihan perangkat lunak tersebut didasarkan pada kemudahan integrasi antara proses pelatihan model machine learning dengan pengembangan aplikasi web, sehingga hasil model dapat langsung diintegrasikan ke dalam sistem tanpa memerlukan konversi format yang rumit. Keterbatasan perangkat keras berbasis CPU turut menjadi pertimbangan dalam pemilihan varian model YOLOv8n (Nano) yang memiliki jumlah parameter lebih sedikit dibandingkan varian YOLOv8 lainnya, sehingga proses pelatihan dan inferensi tetap dapat dilakukan dalam waktu yang wajar meskipun tanpa akselerasi GPU.

2.6. Metrik Evaluasi Model

Evaluasi performa model deteksi pada penelitian ini menggunakan tiga metrik standar pada tugas object detection, yaitu Precision, Recall, dan mean Average Precision (mAP). Precision dihitung sebagai proporsi prediksi cacat yang benar terhadap seluruh prediksi cacat yang dihasilkan model, dengan formulasi $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$, di mana TP (True Positive) adalah jumlah deteksi cacat yang benar dan FP (False Positive) adalah jumlah deteksi cacat yang salah. Recall dihitung sebagai proporsi objek cacat yang berhasil terdeteksi terhadap seluruh objek cacat yang sebenarnya ada pada data uji, dengan formulasi $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$, di mana FN (False Negative) adalah jumlah objek cacat yang gagal terdeteksi oleh model. Sementara itu, mAP@0.5 dihitung dari luas area di bawah kurva precision-recall (Average Precision) pada ambang batas Intersection over Union (IoU) sebesar 0,5, yang merepresentasikan kemampuan model dalam mendeteksi sekaligus melokalisasi area cacat secara bersamaan. Ketiga metrik tersebut dipilih karena secara umum digunakan pada evaluasi model deep learning berbasis object detection [12], sehingga memungkinkan hasil penelitian ini dibandingkan secara objektif dengan penelitian sejenis yang menerapkan algoritma YOLO pada domain lain [2], [6], [8].

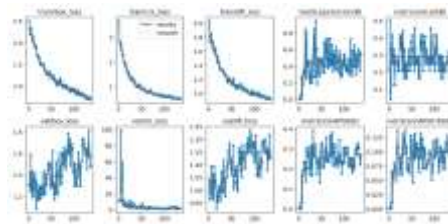
3. Hasil dan Diskusi

3.1. Hasil Pelatihan dan Evaluasi Model

Proses pelatihan model YOLOv8n menunjukkan bahwa nilai train box loss, train class loss, dan train DFL loss menurun secara bertahap seiring bertambahnya epoch, sebagaimana ditampilkan pada Gambar 3. Pada sisi validasi, validation class loss cenderung menurun, sedangkan validation box loss dan validation DFL loss berfluktuasi namun tetap berada pada rentang yang relatif stabil. Metrik precision, recall, mAP@50, dan mAP@50-95 meningkat pada awal pelatihan dan cenderung stabil hingga akhir epoch.

Pola penurunan train loss yang konsisten menunjukkan bahwa model secara bertahap berhasil mempelajari representasi visual dari area cacat pada data latih, sedangkan fluktuasi pada validation box loss dan validation DFL loss mengindikasikan adanya variasi karakteristik gambar pada data validasi yang belum sepenuhnya terwakili

pada data latih, mengingat ukuran dataset yang masih terbatas. Kesenjangan antara tren loss pelatihan dan validasi ini merupakan indikasi umum pada model deep learning yang dilatih dengan volume data kecil, dan menjadi salah satu faktor yang turut menjelaskan nilai Recall serta mAP@0.5 yang masih berada pada kategori sedang pada penelitian ini.



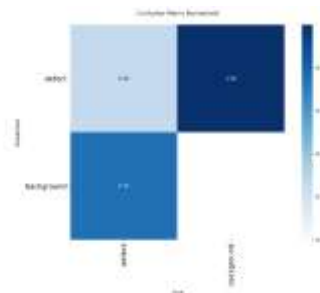
Gambar 3. Grafik Hasil Pelatihan Model YOLOv8

Hasil evaluasi model pada data uji dengan ambang batas IoU sebesar 0,5 ditampilkan pada Tabel 4.

Tabel 4. Hasil Evaluasi Model YOLOv8

Metrik	Nilai
Precision	51,5% (best epoch 31)
Recall	44,3%
mAP@0.5	39,3%

Nilai Precision sebesar 51,5% pada epoch terbaik ke-31 menunjukkan bahwa lebih dari setengah hasil deteksi model merupakan deteksi yang benar. Nilai Recall sebesar 44,3% menunjukkan model berhasil mendeteksi sekitar 44,3% dari seluruh objek cacat pada data uji, sedangkan nilai mAP@0.5 sebesar 39,3% mengindikasikan kemampuan model dalam mendeteksi dan melokalisasi objek cacat masih berada pada kategori sedang. Confusion matrix pada Gambar 4 memperlihatkan bahwa sebanyak 25% objek defect berhasil dideteksi dengan benar sebagai defect, sedangkan 75% sisanya diklasifikasikan sebagai background, yang menegaskan bahwa model masih mengalami kesulitan mendeteksi seluruh objek cacat sehingga menghasilkan nilai recall yang relatif rendah.



Gambar 4. Confusion Matrix

Keterbatasan performa ini sejalan dengan jumlah dataset yang relatif kecil (+200 gambar) dan lingkungan pelatihan berbasis CPU lokal tanpa akselerasi GPU. Hasil ini menunjukkan bahwa performa model masih dapat ditingkatkan melalui penambahan jumlah data pelatihan, perbaikan kualitas anotasi, maupun optimasi parameter model.

3.2. Implementasi Antarmuka Sistem Web

Sistem web yang dibangun menampilkan hasil deteksi berupa bounding box pada area cacat beserta status akhir produk. Pada kondisi produk terdeteksi cacat, sistem menampilkan bounding box berwarna biru dengan label defect beserta nilai confidence score, seperti dicontohkan pada Gambar 5 dengan confidence score sebesar 77,3% yang berada di atas ambang batas 50%, sehingga produk dinyatakan DEFECT dan tidak lolos uji kualitas. Sebaliknya, apabila tidak ditemukan objek dengan confidence score di atas ambang batas, sistem menampilkan status NON-DEFECT tanpa bounding box dan menyatakan produk lolos uji kualitas. Selain menampilkan status akhir produk, halaman hasil deteksi juga menyertakan nilai confidence score dari setiap bounding box yang terdeteksi, sehingga operator dapat menilai tingkat keyakinan model terhadap setiap deteksi yang ditampilkan, bukan sekadar menerima label defect atau non-defect secara langsung. Informasi tambahan ini dimaksudkan untuk

memberi ruang bagi operator melakukan verifikasi ulang secara visual pada kasus dengan confidence score yang mendekati ambang batas 50%, sebelum mengambil keputusan akhir terhadap status kualitas produk.



Gambar 5. Tampilan Output Hasil Deteksi Defect

3.3. Hasil Pengujian Black Box Testing

Pengujian fungsionalitas sistem dilakukan pada skenario positif (data valid) dan skenario negatif (data tidak valid), mencakup pengunggahan gambar defect dan non-defect, format file .jpg dan .png, penekanan tombol deteksi, akses halaman utama, tidak memilih file, format file tidak valid, ukuran file besar, serta gambar berkualitas rendah. Ringkasan hasil pengujian ditampilkan pada Tabel 5. Setiap skenario pengujian disusun dengan mengacu pada kondisi penggunaan nyata oleh operator di lapangan, termasuk kemungkinan kesalahan penggunaan seperti lupa memilih file maupun mengunggah dokumen dengan format yang tidak didukung sistem, sehingga hasil pengujian dapat merepresentasikan kesiapan sistem menghadapi variasi input yang mungkin terjadi pada praktiknya.

Tabel 5. Ringkasan Hasil Black Box Testing

Skenario Pengujian	Jumlah Kasus	Keberhasilan
Positif (data valid)	5	100%
Negatif (data tidak valid)	4	100%

Berdasarkan hasil pengujian tersebut, seluruh skenario baik positif maupun negatif menunjukkan hasil yang sesuai dengan yang diharapkan. Sistem berhasil memproses gambar yang diunggah, menjalankan deteksi cacat secara otomatis, menampilkan bounding box, serta mengklasifikasikan status produk sesuai logika confidence thresholding, dengan tingkat keberhasilan fungsional sebesar 100%. Hal ini mengonfirmasi bahwa sistem web deteksi cacat yang dikembangkan berfungsi sesuai dengan perancangan yang telah dilakukan.

3.4. Diskusi

Hasil pengujian menunjukkan bahwa model YOLOv8n mampu mempelajari pola visual cacat pada permukaan metal cutting dies insole, namun nilai Recall dan mAP@0.5 yang diperoleh masih tergolong sedang apabila dibandingkan dengan penelitian sejenis yang menerapkan YOLOv8 pada objek lain dengan jumlah dataset yang jauh lebih besar dan didukung akselerasi GPU. Perbedaan ini mengindikasikan bahwa performa model sangat dipengaruhi oleh kuantitas dan variasi data latih, di mana dataset sebanyak ± 200 gambar pada penelitian ini relatif terbatas untuk merepresentasikan seluruh variasi kondisi pencahayaan, sudut pengambilan gambar, dan jenis cacat yang mungkin terjadi pada kondisi produksi sesungguhnya.

Dari sisi implementasi sistem, integrasi model YOLOv8 ke dalam antarmuka berbasis web terbukti dapat memberikan gambaran nyata bagaimana model deep learning dapat dimanfaatkan secara praktis dalam proses inspeksi kualitas, tanpa memerlukan operator untuk memahami detail teknis model. Antarmuka yang sederhana memungkinkan operator hanya perlu mengunggah gambar produk untuk memperoleh hasil deteksi dan status akhir produk secara instan. Hasil pengujian Black Box Testing dengan tingkat keberhasilan fungsional 100% juga menunjukkan bahwa sistem telah berjalan sesuai dengan rancangan pada seluruh skenario yang diujikan, baik pada kondisi input yang valid maupun tidak valid, sehingga sistem dapat dikatakan siap untuk diujicobakan lebih lanjut pada lingkungan produksi yang sesungguhnya.

Beberapa keterbatasan pada penelitian ini perlu menjadi perhatian pada penelitian selanjutnya. Pertama, jumlah dataset yang relatif kecil berpotensi menyebabkan model kurang mampu menggeneralisasi pada variasi cacat yang belum pernah ditemui sebelumnya. Kedua, proses pelatihan yang dilakukan sepenuhnya pada CPU membatasi jumlah epoch dan ukuran model yang dapat dieksplorasi dalam waktu penelitian yang tersedia, sehingga varian YOLOv8 yang lebih besar seperti YOLOv8s atau YOLOv8m belum sempat diujicobakan. Ketiga, pengujian sistem pada penelitian ini masih berfokus pada aspek fungsional melalui Black Box Testing dan belum mencakup pengujian pada kondisi jaringan produksi yang sesungguhnya dengan volume gambar yang besar secara

bersamaan. Ketiga keterbatasan tersebut membuka peluang penelitian lanjutan yang lebih komprehensif di masa mendatang.

Ditinjau dari sisi penerapan praktis, sistem yang dihasilkan pada penelitian ini berpotensi memberikan manfaat nyata bagi PT Jomu Studio Indonesia maupun industri sejenis apabila dikembangkan lebih lanjut, khususnya dalam mengurangi beban kerja operator inspeksi visual dan mempercepat proses quality control pada lini produksi. Dengan antarmuka berbasis web yang sederhana, operator dengan latar belakang teknis yang terbatas tetap dapat mengoperasikan sistem tanpa memerlukan pelatihan khusus mengenai model deep learning. Namun demikian, sebelum diterapkan secara penuh pada lingkungan produksi, disarankan dilakukan pengujian lanjutan dengan volume data yang lebih besar dan lebih beragam, penambahan jenis cacat yang dapat dideteksi, serta evaluasi terhadap kecepatan pemrosesan (inference time) pada kondisi beban kerja yang mendekati kondisi produksi sesungguhnya, agar sistem benar-benar siap diadopsi sebagai bagian dari proses quality control industri secara berkelanjutan.

3.5. Perbandingan dengan Penelitian Sejenis

Apabila dibandingkan dengan penelitian penerapan YOLO pada produk hasil aluminium casting yang melaporkan precision sebesar 0,94 dan mAP@0,5 sebesar 0,90 [2], performa model pada penelitian ini masih jauh lebih rendah. Kesenjangan tersebut wajar mengingat penelitian tersebut menggunakan jumlah dataset yang jauh lebih besar dan lingkungan pelatihan yang didukung GPU, sedangkan penelitian ini dibatasi oleh dataset primer berjumlah ± 200 gambar dan pelatihan berbasis CPU. Pola serupa juga terlihat pada penelitian deteksi cacat permukaan baja menggunakan arsitektur YOLOv8 yang telah dimodifikasi [8], di mana penyesuaian arsitektur dan volume data yang lebih besar turut berkontribusi terhadap performa deteksi yang lebih tinggi dibandingkan penggunaan YOLOv8n bawaan seperti pada penelitian ini. Penelitian lain yang menerapkan YOLOv8 pada deteksi cacat isolator [6] serta penelitian deteksi cacat produk masker menggunakan YOLOv5 [16] turut menegaskan bahwa performa algoritma keluarga YOLO pada kasus inspeksi cacat sangat bergantung pada karakteristik objek, kualitas anotasi, dan skala dataset yang digunakan, sehingga hasil pada satu domain industri tidak selalu dapat digeneralisasi secara langsung ke domain lain seperti cutting dies insole pada penelitian ini. Perbandingan ini menegaskan posisi penelitian ini sebagai purwarupa awal yang masih terbuka untuk perbaikan performa melalui penambahan data dan sumber daya komputasi.

4. Kesimpulan

Penelitian ini berhasil membangun sistem deteksi dan klasifikasi cacat produk berbasis web menggunakan algoritma YOLOv8 untuk mendukung proses quality control pada produk metal cutting dies insole di PT Jomu Studio Indonesia. Model YOLOv8n yang dilatih menggunakan dataset ± 200 gambar dengan transfer learning dari checkpoint publik MS COCO menghasilkan nilai Precision sebesar 51,5% pada epoch terbaik ke-31, Recall sebesar 44,3%, dan mAP@0.5 sebesar 39,3%, yang menunjukkan bahwa model mampu mengenali area cacat pada permukaan metal cutting dies insole meskipun performanya masih dipengaruhi oleh keterbatasan jumlah dataset dan lingkungan pelatihan berbasis CPU tanpa akselerasi GPU. Sistem berbasis web yang dibangun menggunakan Flask dan HTML berhasil mengklasifikasikan kondisi akhir produk ke dalam kategori defect dan non-defect menggunakan pendekatan confidence thresholding dengan ambang batas 50%, dan pengujian fungsionalitas menggunakan Black Box Testing menunjukkan seluruh skenario pengujian berjalan sesuai rencana dengan tingkat keberhasilan 100%. Penelitian selanjutnya disarankan memperluas jumlah dan variasi dataset, menambah jenis cacat yang dideteksi, menggunakan perangkat komputasi berbasis GPU, mengeksplorasi varian YOLOv8 yang lebih besar atau arsitektur deteksi objek lain, serta menguji berbagai nilai confidence threshold untuk meningkatkan akurasi sistem pada penerapan industri yang sesungguhnya. Secara keseluruhan, penelitian ini menunjukkan bahwa kombinasi antara model object detection yang ringan dan antarmuka berbasis web yang sederhana dapat menjadi titik awal yang layak untuk mendorong adopsi inspeksi kualitas berbasis kecerdasan buatan pada industri manufaktur skala kecil hingga menengah di Indonesia, meskipun diperlukan penyempurnaan lebih lanjut sebelum diterapkan secara penuh pada lingkungan produksi nyata. Dokumentasi tahapan penelitian ini, mulai dari pengumpulan data primer hingga pengujian sistem, juga diharapkan dapat menjadi rujukan metodologis bagi mahasiswa maupun praktisi lain yang hendak menerapkan pendekatan serupa pada kasus inspeksi visual di industri sejenis.

Referensi

1. A. A. Khan, A. A. Laghari, and S. A. Awan, "Machine learning in computer vision: A review," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 8, no. 32, pp. 1-11, 2021. <https://doi.org/10.4108/eai.21-4-2021.169418>
2. N. Lilansa, R. Aji Pratama, R. Tanadi, and R. N. Nindi Kara, "Pengujian algoritma YOLO untuk deteksi cacat pada produk hasil aluminium casting pada industri otomotif," *J. Teknol. Terapan*, vol. 10, no. 2, 2024. <https://doi.org/10.31884/jtt.v10i2.664>

DOI: <https://doi.org/10.31004/riggs.v5i2.12724>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

3. B. Tang, L. Chen, W. Sun, and Z. Lin, "Review of surface defect detection of steel products based on machine vision," *IET Image Process.*, vol. 17, no. 2, pp. 303-322, 2023. <https://doi.org/10.1049/ipr2.12647>
4. M. Hussain, "YOLOv1 to v8: Unveiling each variant-a comprehensive review of YOLO," *IEEE Access*, vol. 12, pp. 42816-42833, 2024. <https://doi.org/10.1109/ACCESS.2024.3378568>
5. X. Song, S. Cao, J. Zhang, and Z. Hou, "Steel surface defect detection algorithm based on YOLOv8," *Electronics*, vol. 13, no. 5, 2024. <https://doi.org/10.3390/electronics13050988>
6. L. Zhang, B. Li, Y. Cui, Y. Lai, and J. Gao, "Research on improved YOLOv8 algorithm for insulator defect detection," 2023. <https://doi.org/10.21203/rs.3.rs-3337929/v1>
7. N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object tracking menggunakan algoritma You Only Look Once (YOLO)v8 untuk menghitung kendaraan," *KOMPUTA J. Ilm. Komput. Inform.*, vol. 12, no. 2, 2023. <https://doi.org/10.34010/komputa.v12i2.10654>
8. H. Kong and Y. C., "Improved steel surface defect detection algorithm based on YOLOv8," *IEEE Access*, 2024. <https://doi.org/10.1109/ACCESS.2024.10695239>
9. J. E. Gallagher and E. J. Oughton, "Surveying You Only Look Once (YOLO) multispectral object detection advancements, applications and challenges," 2024. <https://doi.org/10.48550/arXiv.2409.12977>
10. I. N. Simbolon, D. F. Lumbanraja, and K. Tampubolon, "Analysis and implementation of YOLOv7 in detecting pin del in real-time," *J. Tek. Inform. (Jutif)*, vol. 5, no. 2, pp. 579-587, 2024. <https://doi.org/10.52436/1.jutif.2024.5.2.1286>
11. E. Jovanovska, M. Kotevski, B. Kotevski, and S. Koceski, "Automated door state detection using deep learning: A computer vision approach with Roboflow platform," *Balkan J. Appl. Math. Inform.*, vol. 8, no. 1, 2025. <https://doi.org/10.46763/BJAMI>
12. I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Comput. Sci.*, vol. 2, no. 6, 2021. <https://doi.org/10.1007/s42979-021-00815-1>
13. A. J. Dhruv, R. Patel, and N. Doshi, "Python: The most advanced programming language for computer science applications," 2022. <https://doi.org/10.5220/0010307900003051>
14. C. Wijayanto and Y. A. Susetyo, "Implementasi Flask framework pada pembangunan aplikasi sistem informasi helpdesk (SIH)," 2022.
15. J. Shadiq, A. Safei, and R. W. R. Loly, "Pengujian aplikasi peminjaman kendaraan operasional kantor menggunakan blackbox testing," *Inf. Manag. Educ. Prof.*, vol. 5, no. 2, pp. 97-110, 2021.
16. Y. Arvio, D. T. Kusuma, and I. B. Sangadji, "Pendekatan algoritma YOLO V5 untuk mendeteksi cacat produk masker," 2024.
17. C. Nisa', A. Wijaya, and F. Rizal, "Teori UML dan implementasi praktek: panduan untuk pengembangan perangkat lunak," *CV Bravo Press Indonesia*, 2024.