

## PENGUJIAN ALGORITMA *YOLO* UNTUK DETEKSI CACAT PADA PRODUK HASIL *ALUMINIUM CASTING* PADA INDUSTRI OTOMOTIF

Noval Lilansa, Dipl.Ing(FH)<sup>1</sup>, Rizqi Aji Pratama<sup>2</sup>, Richard Tanadi<sup>3</sup>,  
Renold Nindi Kara N.<sup>4</sup>

<sup>1,2,4</sup> Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Negeri Bandung

<sup>3</sup>Praktisi Industri, Data Labs Analytics Indonesia

Email: <sup>1</sup>noval@ae.polman-bandung.ac.id, <sup>2</sup>rizqi@ae.polman-bandung.ac.id, <sup>3</sup>richard.tanadi@datalabs.id,

<sup>4</sup>223442909@mhs.polman-bandung.ac.id

### Abstrak

Penelitian ini membahas tantangan yang dihadapi oleh salah satu pabrik pembuat *parts* otomotif di Indonesia, dalam mendeteksi kebocoran pada komponen mobil yang diproduksi melalui *die casting aluminium*. Metode pemantauan manual yang ada saat ini memakan waktu, rentan terhadap kesalahan manusia, dan menimbulkan risiko terhadap kualitas produk dan keselamatan operasional. Untuk mengatasi tantangan ini, penelitian ini mengusulkan penerapan *machine learning*, khususnya teknik *computer vision*, seperti *Object Detection* untuk mengidentifikasi dan melokalisasi gelembung gas menggunakan *Leak Tester Machine* yang dilengkapi sensor kamera yang dilengkapi algoritma *YOLO* untuk meningkatkan efisiensi proses deteksi. Seiring perkembangan teknologi terjadi pergeseran dari pendekatan manual ke sistem otomatis yang lebih efisien, dengan memanfaatkan algoritma *computer vision*, hardware seperti *Nvidia Jetson* dan *Digital Camera*. Algoritma *YOLO* menunjukkan akurasi yang cukup baik dalam mengatasi deteksi cacat pada produk hasil *aluminium casting*, mencapai *precision* sebesar 0.94, *recall* sebesar 0.82, dan *mAP@0.5* sebesar 0.90. Performa algoritma *YOLO* dalam mendeteksi *bubble*/gelembung gas yaitu 100 – 120 ms per frame dari *streaming camera* sehingga didapatkan performa fps-nya yaitu 10 fps (*frame per second*).

**Kata Kunci:** *Quality Control, Defect Products, Computer Vision, Object Detection, Leak Tester*

### Abstract

*This research addresses the challenges faced by one of the automotive parts manufacturing plants in Indonesia, in detecting leaks in car components manufactured through aluminum die casting. Existing manual monitoring methods are time-consuming, prone to human error, and pose risks to product quality and operational safety. To overcome these challenges, this research proposes the application of machine learning, specifically computer vision techniques, such as Object Detection to identify and localize gas bubbles using a Leak Tester Machine equipped with a camera sensor equipped with the YOLO algorithm to improve the efficiency of the detection process. As technology develops, there is a shift from manual approaches to more efficient automated systems, utilizing computer vision algorithms, hardware such as Nvidia Jetson and Digital Camera. The YOLO algorithm shows good accuracy in overcoming defect detection in aluminum casting products, achieving a precision of 0.94, recall of 0.82, and mAP@0.5 of 0.90. The performance of the YOLO algorithm in detecting gas bubbles is 100 - 120 ms per frame from the streaming camera so that the fps performance is 10 fps (frames per second).*

**Keywords:** *Quality Control, Defect Products, Computer Vision, Object Detection, Leak Tester*

**I. PENDAHULUAN**

Dalam ranah industri manufaktur, kontrol kualitas (QC) merupakan salah satu aspek penting yang perlu diperhatikan untuk memastikan kualitas produk yang dihasilkan sesuai dengan standar yang ditetapkan (Putri & Primananda, 2021). Salah satu risiko yang perlu diwaspadai dalam QC adalah risiko kebocoran pada produk. Salah satu pabrik pembuatan *parts* otomotif di Indonesia mengalami tantangan serius dalam mendeteksi kebocoran pada produk suku cadang mobil yang diproduksi menggunakan metode pengecoran aluminium (die casting). Saat ini, metode deteksi yang bergantung pada pemantauan manusia telah menunjukkan keterbatasan yang signifikan. Proses ini rentan terhadap subjektivitas, ketidakpastian, dan efisiensi rendah, yang secara langsung mengakibatkan risiko tinggi terkait kualitas produk dan keselamatan operasional (Gladkoff et al., 2021).

Berdasarkan hasil wawancara dengan karyawan di salah satu pabrik pembuatan *parts* otomotif, terdapat beberapa masalah dalam pemantauan manusia untuk deteksi kebocoran pada produk, terutama lamanya waktu pemantauan dan potensi kesalahan manusia. Karyawan harus memeriksa setiap produk secara manual, yang membutuhkan waktu lama dan rentan terhadap kelelahan atau kelalaian, sehingga tidak dapat mendeteksi kebocoran dengan akurat. Jika tidak segera diatasi, salah satu pabrik pembuatan *parts* otomotif ini berpotensi menghadapi kerugian serius terkait dengan standar keamanan dan kualitas produk, termasuk kegagalan produk, bahaya operasional, dan dampak finansial akibat *recall* produk atau klaim konsumen. Menurut (Aust & Pons, 2022), penggunaan mesin seperti computer vision dapat meningkatkan kualitas produk dan efisiensi proses produksi dengan mengatasi keterbatasan pemantauan manusia. Algoritma pembelajaran mesin, khususnya dalam bidang Object Detection, telah muncul sebagai alat yang ampuh untuk mendeteksi kebocoran pada produk suku cadang mobil (Lindsay, 2021).

Adopsi metode pembelajaran mesin seperti Object Detection (Lindsay, 2021), merupakan tonggak penting dalam meningkatkan sistem deteksi kebocoran. Teknologi ini memungkinkan mesin Penguji Kebocoran Gelembung untuk secara otomatis mengidentifikasi dan melokalisasi gelembung gas (Yang et al., 2020) yang mengindikasikan kebocoran pada produk suku cadang mobil. Penggunaan convolutional neural network (CNN) (Lindsay, 2021) memungkinkan mesin memahami pola kompleks dalam gambar dan

mendapatkan wawasan mendalam tentang struktur produk suku cadang mobil.

Factor	Weight	Human		IP Software		AI Software		3D Scanning	
		Score	Weighted Score	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Inspection Accuracy	12.2	3	36.5	2	24.4	2	24.4	5	60.9
Inspection Consistency	10.9	4	43.6	5	54.5	5	54.5	5	54.5
Inspection Time	11.5	2	23.1	4	46.2	5	57.7	1	11.5
Investment Cost	7.1	2	14.1	4	28.2	2	14.1	1	7.1
Operating Cost	5.1	3	15.4	2	10.3	2	10.3	4	20.5
H/TRL	8.3	3	25.0	1	8.3	3	25.0	5	41.7
Agility	7.7	5	38.5	2	15.4	3	23.1	4	30.8
Flexibility	9.6	4	38.5	2	19.2	3	28.8	4	38.5
Automation	3.8	2	7.7	2	7.7	2	7.7	3	11.5
Standardisation	5.8	1	5.8	1	5.8	2	11.5	3	17.3
Interoperability	3.2	4	12.8	2	6.4	2	6.4	4	12.8
Documentation	5.1	2	10.3	1	5.1	5	25.6	5	25.6
Compliance	9.6	5	48.1	1	9.6	4	38.5	5	48.1
Sum	100		319.2		241.0		327.6		380.8

Gambar 1. *Weighted Factor Analysis for different inspection agents (Aust & Pons, 2022).*

Penelitian sebelumnya telah menunjukkan potensi besar teknologi deteksi kebocoran menggunakan metode modern. Penelitian yang ditulis oleh (Gao et al., 2020) mengembangkan metode deteksi kebocoran untuk bejana tekan yang menggabungkan analisis tekanan dan teknik sensor untuk mendeteksi kebocoran secara akurat. Selain itu, penelitian yang dilakukan oleh (Pauline et al., 2020) mengeksplorasi analisis citra termografik untuk mendeteksi kebocoran air dalam kondisi laboratorium, yang terbukti efektif dibandingkan metode akustik tradisional. Penelitian yang dilakukan (Yang et al., 2020) melakukan survei komprehensif tentang metode deep learning dalam mendeteksi cacat pada produk manufaktur, mengategorikan jenis cacat dan mendiskusikan teknik-teknik utama yang digunakan.

Berdasarkan permasalahan dan penelitian terdahulu tersebut, beberapa teknologi terkini telah diadopsi untuk meningkatkan deteksi kebocoran, salah satunya adalah Mesin Leak Tester yang dilengkapi dengan computer vision. Solusi inovatif ini memanfaatkan teknologi computer vision untuk melakukan deteksi otomatis gelembung gas yang mengindikasikan kebocoran (Gao et al., 2020), memberikan keunggulan tingkat akurasi yang tinggi dengan menggunakan angin sebagai media uji. Penelitian ini diharapkan dapat meningkatkan otomatisasi, akurasi, dan reliabilitas hasil deteksi kebocoran, membantu industri manufaktur untuk mengoptimalkan proses pengujian, meningkatkan efisiensi produksi, dan mengurangi risiko terkait kualitas produk serta keselamatan operasional.

**II. METODE**

*Die Casting Process*

Proses *die-casting* adalah metode pengecoran logam yang menggunakan tekanan tinggi untuk

memasukkan logam cair ke dalam rongga cetakan yang presisi. Metode ini umumnya menggunakan logam dengan titik leleh rendah seperti seng, aluminium, magnesium, tembaga, timah, dan timbal. Terdapat dua jenis proses *die-casting*, yaitu *hot chamber* dan *cold chamber*, yang dipilih berdasarkan titik leleh logam. Keunggulan utama dari *die-casting* adalah kecepatan produksi tinggi dan biaya rendah, membuatnya ideal untuk produksi massal. Produk yang dihasilkan memiliki kualitas permukaan dan sifat mekanik yang baik, meskipun proses ini memiliki kelemahan seperti biaya cetakan tinggi, batasan ukuran dan berat produk, serta risiko cacat seperti porositas dan retak (Qi et al., 2021).

Pada *die-casting* aluminium, batangan aluminium dipanaskan dan dilebur pada suhu tinggi sebelum dimasukkan ke dalam rongga cetakan untuk membentuk bagian logam yang diinginkan. Meskipun proses ini efisien, porositas tetap menjadi masalah utama karena dapat menyebabkan kebocoran setelah proses machining. Pin hole yang sulit dideteksi secara visual dapat menyebabkan masalah serius, terutama jika bagian yang cacat terpasang pada kendaraan. Oleh karena itu, kontrol dan penanganan porositas sangat penting untuk menjaga kualitas dan integritas produk aluminium die-casting (Qi et al., 2021).

### **Leak Tester Machine**

Mesin *Leak Tester* adalah perangkat penting dalam industri manufaktur yang digunakan untuk mendeteksi kebocoran pada komponen seperti *casting*, *forgings*, dan *weldings*. Mesin ini memastikan kualitas dan keamanan produk dengan menggunakan media air dan udara. Produk diisi dengan udara bertekanan dan direndam dalam tangki air. Jika terdapat kebocoran, udara bertekanan akan keluar dan membentuk gelembung yang dapat diamati sebagai indikasi kebocoran (Gowardipe et al., 2008). Gambar 2 merupakan gambar dari Mesin *Leak Teser*.



Gambar 2. *Leak Tester Machine*.

Komponen utama dari mesin ini meliputi kompresor udara untuk memasukkan udara

bertekanan, tangki air untuk merendam produk, serta sensor gelembung atau kamera untuk mendeteksi gelembung udara. Sistem kontrol dan monitoring juga krusial, mengatur tekanan udara serta durasi pengujian, dan memantau munculnya gelembung dengan sensor atau kamera. Beberapa mesin dilengkapi dengan perangkat lunak analitik yang menganalisis data dan memberikan laporan hasil pengujian secara otomatis. Dengan kombinasi komponen ini, mesin *Leak Tester* memastikan integritas produk dengan efisiensi dan akurasi tinggi, mengurangi risiko kebocoran yang dapat menyebabkan kerugian lebih lanjut (Gowardipe et al., 2008).

### **Computer Vision**

Computer vision adalah bidang dalam kecerdasan buatan yang memungkinkan komputer dan sistem untuk mengekstrak informasi bermakna dari gambar digital, video, dan input visual lainnya, serta mengambil tindakan atau membuat rekomendasi berdasarkan informasi tersebut. Tujuannya adalah untuk mengembangkan teknik yang memungkinkan komputer untuk memahami dan menginterpretasikan konten visual dengan cara yang mirip dengan bagaimana manusia melihat dan memahami dunia. Ini mencakup berbagai tugas seperti pengenalan wajah, deteksi objek, segmentasi gambar, dan analisis video, serta diterapkan dalam berbagai domain termasuk keamanan, kesehatan, otomotif, dan manufaktur.

Classification	Method	Backbone	AP <sub>[0.5,0.95]</sub> (%)	FPS
Anchor-based	Faster R-CNN	VGG-16	21.9	7
	R-FCN	ResNet-101	29.9	9
	Mask R-CNN	ResNeXt-101	39.8	11
	YOLO v2	VGG-16	21.6	40
	YOLO v3	DarkNet-53	31.0	35
	YOLO v4	CSPDarkNet-53	43.5	23
	YOLO v7	E-ELAN	51.4	161
	SSD	VGG-16	28.8	19.3
	DSSD	ResNet-101	33.2	5.5
	ASSD	ResNet101	34.5	6.1
	PSSD	VGG-16	33.8	45

Gambar 3. *Performance comparison of representative 2D detection models on the MS COCO test* (Chen et al., 2023).

Berdasarkan Gambar 3, algoritma YOLO v7 paling cocok untuk kasus yang diteliti karena memiliki performa yang tinggi dalam hal AP (*Average Precision*) dan FPS (*Frames Per Second*). Kombinasi ini menjadikannya pilihan unggul untuk aplikasi yang membutuhkan deteksi objek dengan akurasi tinggi dan kecepatan pemrosesan yang cepat.

### Dukungan Data

Untuk *dataset* pelatihan, *dataset* diambil dengan mereplikasi kondisi di salah satu pabrik pembuatan *parts* otomotif. Pengambilan gambar dilakukan dengan kamera yang diposisikan pada jarak antara 200 - 250 mm dari objek, dengan fokus pada gelembung yang berdiameter lebih dari 10 mm. *Dataset* yang terkumpul memiliki resolusi 1280 x 720 piksel, setiap gambar menggambarkan gelembung dalam tangki berisi air yang telah disesuaikan dengan kondisi operasional sebenarnya di pabrik. Upaya ini memastikan bahwa data yang digunakan dalam pelatihan mencerminkan situasi nyata, sehingga analisis dapat memberikan wawasan yang relevan dan dapat diterapkan di salah satu pabrik pembuatan *parts* otomotif.

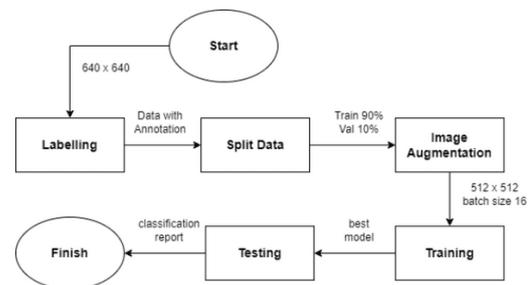
Setelah *dataset* terkumpul, langkah selanjutnya adalah memberikan anotasi atau *bounding box* menggunakan alat anotasi *Computer Vision Annotation Tool (CVAT)* (Boris Sekachev, 2023). Proses anotasi ini penting untuk menandai dan mengidentifikasi posisi serta ukuran gelembung pada setiap gambar. Setelah semua gambar berhasil dianotasi, data tersebut kemudian dibagi menjadi dua set, yaitu *data train* dan *data validation*, untuk memastikan bahwa model yang dilatih dapat diuji dengan data yang belum pernah dilihat sebelumnya. Pembagian data dilakukan dengan rasio 90% untuk data train dan 10% untuk data validation (Mrinal Walia, 2022). Langkah-langkah ini bertujuan untuk memaksimalkan akurasi dan generalisasi model dalam mendeteksi gelembung pada kondisi operasional yang sebenarnya di salah satu pabrik pembuatan *parts* otomotif.

### Pengembangan Model

Algoritma deteksi objek yang dipilih atau dikembangkan, seperti *YOLO (You Only Look Once)* (Wang et al., 2024), menggunakan *framework deep learning* seperti *PyTorch*. Integrasi dengan *OpenCV* mendukung prapemrosesan citra dan ekstraksi fitur yang diperlukan untuk deteksi *bubble*. *Python* dipilih sebagai bahasa pemrograman utama untuk memastikan kohesivitas dalam pengembangan dan integrasi komponen-komponen sistem.

*Flowchart* pada gambar 4 menggambarkan tahapan perancangan untuk mengembangkan algoritma *computer vision* yang berfokus pada deteksi objek, khususnya untuk mendeteksi *bubble* pada mesin *leak tester*. Proses dimulai pada tahap "Start", di mana data latih berupa gambar dengan ukuran 640 x 640 piksel dilabeli dengan anotasi pada tahap "Labelling". Setelah itu, data yang telah dilabeli dibagi menjadi dua bagian: 90% untuk

pelatihan dan 10% untuk validasi pada tahap "Split Data". Selanjutnya, dilakukan augmentasi gambar pada tahap "Image Augmentation" sebelum memasuki tahap "Training" dengan ukuran citra 512 x 512 piksel dan batch size 16. Algoritma terbaik dari pelatihan digunakan untuk pengujian pada tahap "Testing", yang menghasilkan laporan klasifikasi sebagai output akhir pada tahap "Finish". Dengan mengikuti *flowchart* yang terstruktur ini, pengembangan algoritma deteksi objek yang handal dapat dilakukan, memastikan akurasi tinggi dan keandalan dalam mendeteksi *bubble* yang penting untuk kinerja optimal mesin *leak tester*.

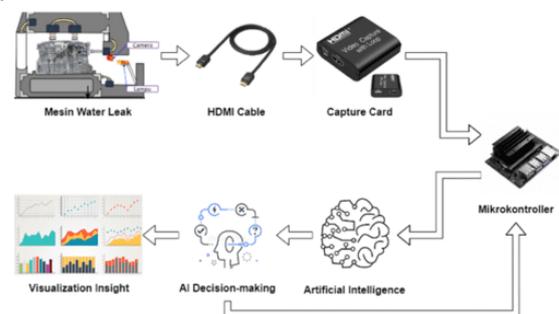


Gambar 4. *Flowchart* Pengembangan Model

### Pengembangan Hardware

Dengan mengimplementasikan teknik *computer vision* yang canggih, sistem deteksi *bubble* dapat memanfaatkan algoritma yang telah terlatih untuk mengenali dan memahami pola *bubble* pada produk aluminium casting yang ditesting pada mesin *leak tester*.

Perancangan *hardware* sistem deteksi *bubble* pada mesin *leak tester* melibatkan integrasi antara komponen perangkat keras dan perangkat lunak. Di inti sistem ini terdapat *Nvidia Jetson* (Bokovoy et al., 2019; Jeon et al., 2021), yang dipilih karena kemampuannya dalam pemrosesan yang kuat, ditempatkan secara strategis di dekat mesin untuk mengelola data dan menjalankan algoritma *Object Detection*. Kamera digital (Suder, 2022), sebagai komponen utama perangkat keras, dipasang pada *Nvidia Jetson* dan dikonfigurasi untuk mengambil gambar atau *video* dari area yang relevan pada mesin *leak tester*, seperti yang diilustrasikan pada gambar 5.



Gambar 5. Arsitektur sistem deteksi *bubble*

### Skenario Pengujian

Skenario pengujian algoritma pada mesin *leak tester* yang dilengkapi *computer vision* ini yaitu melakukan pengujian terhadap datasets yang diambil secara langsung dari salah satu pabrik pembuatan *parts* otomotif. Pengujian ini bertujuan untuk memilih algoritma terbaik yang akan digunakan dan diimplementasikan, algoritma yang digunakan untuk pengujian yaitu *YOLO*. *Output* yang dihasilkan berupa *Precision*, *Recall*, *mAP@0.5*, waktu deteksi, *fps* (*frame per second*) yang didapat dari masing masing algoritma. *Confusion matrix* (Hidayatullah, 2021) merupakan metode perhitungan statistik yang umum digunakan untuk mengukur kinerja algoritma.

Tabel 1. *Confusion Matrix* (Hidayatullah, 2021)

	Hasil Prediksi	
	0	1
Realita 0	TN	FP
Realita 1	FN	TP

Dari tabel *confusion matrix* tersebut dapat digunakan untuk menentukan *precision*, *recall* (Bunyamin, 2020; Hidayatullah, 2021) menggunakan persamaan dibawah ini:

$$precision = \frac{TP}{TP+FP} = \frac{TP}{DB} \quad (1)$$

$$recall = \frac{TP}{TP+FN} = \frac{TP}{GT} \quad (2)$$

### III. HASIL DAN PEMBAHASAN

#### *Datasets*

Proses pengumpulan data dilakukan dengan menggunakan akuarium yang memiliki ukuran dan kondisi serupa dengan yang ada di salah satu pabrik pembuatan *parts* otomotif. Akuarium ini berfungsi sebagai simulasi tangki di pabrik untuk memastikan bahwa data yang dihasilkan relevan dan representatif. Penggunaan akuarium ini memungkinkan penulis untuk mengontrol dan mereplikasi berbagai kondisi operasional yang ada di pabrik.

Produk yang digunakan dalam pengambilan data adalah hasil dari proses aluminium casting (Qi et al., 2021), yang sering kali menghasilkan gelembung gas ketika ada kebocoran yang menjadi

objek utama penelitian ini. Pengambilan gambar dilakukan dalam berbagai kondisi untuk mencerminkan situasi nyata di pabrik, seperti kondisi gelap untuk mensimulasikan lingkungan pabrik dengan pencahayaan minim, yang ditunjukkan pada gambar 6. Selain itu, gambar diambil saat gelembung terkena cahaya untuk mengamati efek refleksi dan refraksi, yang ditunjukkan pada gambar 7. Kondisi lain yang diobservasi adalah air bergelombang atau tidak tenang untuk mensimulasikan kondisi operasional yang dinamis, seperti yang ditunjukkan pada gambar 8. Pendekatan ini memastikan bahwa dataset yang terkumpul mencakup berbagai skenario yang mungkin terjadi di pabrik, sehingga analisis yang dilakukan dapat memberikan wawasan yang lebih akurat dan dapat diterapkan secara langsung.



Gambar 6. Foto *datasets* dengan pencahayaan minim



Gambar 7. Foto *datasets* gelembung terkena efek refleksi dan refraksi



Gambar 8. Foto *datasets* dengan air yang bergelombang

*Dataset* yang terkumpul terdiri dari 5280 gambar dengan total ukuran sebesar 4.2 GB (*Gigabyte*). Gambar-gambar ini memiliki resolusi 1280 x 720 piksel dan berhasil menangkap berbagai kondisi yang mencerminkan situasi nyata di pabrik. Setelah pengumpulan data selesai, langkah berikutnya adalah melakukan anotasi pada setiap gambar. Proses anotasi ini dilakukan menggunakan *Computer Vision Annotation Tool (CVAT)* (Boris Sekachev, 2023). Anotasi dilakukan untuk menandai lokasi dan ukuran gelembung pada setiap gambar, yang akan digunakan dalam proses pelatihan algoritma. Proses anotasi ini ditunjukkan pada gambar 9.



Gambar 9. Proses Anotasi datasets menggunakan CVAT

Untuk keperluan pelatihan algoritma, *dataset* ini dibagi menjadi dua set yaitu *data train* dan *data validation*. Pembagian dilakukan dengan rasio 9:1, di mana 90% dari total gambar digunakan untuk melatih algoritma, dan 10% sisanya digunakan untuk memvalidasi performa algoritma (Mrinal Walia, 2022). Rasio ini ditunjukkan pada Tabel 2.

Tabel 2. Tabel Data *Splitting*

Data	Rasio	Jumlah Gambar
Data Training	9:1	4761
Data Validation	1:9	529
Total Data	-	5290

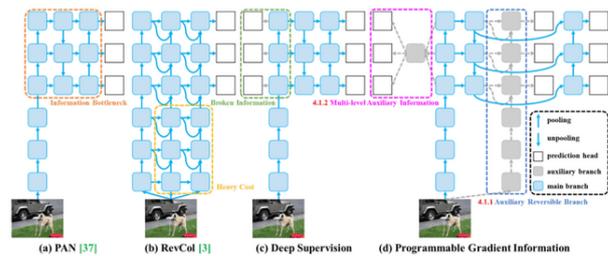
### Hasil Training YOLO (*You Only Look Once*)

Proses *training* merupakan langkah penting dalam mengoptimalkan algoritma menggunakan jaringan saraf tiruan (*neural network*). Sebelum dapat memulai proses training, langkah awal yang harus dilakukan adalah merancang arsitektur algoritma YOLO. YOLO (*You Only Look Once*) adalah algoritma yang sudah dilatih sebelumnya dengan *dataset* besar seperti COCO atau ImageNet.

YOLOv9 memperkenalkan sejumlah peningkatan signifikan dibandingkan YOLOv7,

terutama melalui implementasi arsitektur *Generalized Efficient Layer Aggregation Network (GELAN)* dan mekanisme *Programmable Gradient Information (PGI)*. GELAN dirancang untuk mengatasi masalah hilangnya informasi selama proses *feedforward* dengan memanfaatkan jalur gradien yang dapat diatur, sehingga memungkinkan *model* untuk mempertahankan dan memanfaatkan informasi penting secara lebih efektif. Sementara YOLOv7 sudah menunjukkan kinerja yang solid, YOLOv9 meningkatkan efisiensi *parameter* dan kompleksitas komputasi, mengurangi jumlah parameter hingga 42% dan kebutuhan kalkulasi hingga 22% dibandingkan *model* sebelumnya, sambil tetap mempertahankan dan meningkatkan akurasi deteksi objek. Peningkatan ini membuat YOLOv9 lebih efisien dan akurat, serta lebih cocok untuk aplikasi *real-time* yang membutuhkan kecepatan dan keandalan tinggi.

Dengan memanfaatkan arsitektur YOLOv9 yang sudah di training dengan data yang besar ini dapat menghemat waktu dan tenaga dalam perancangan dan pelatihan algoritma dari awal, sehingga lebih fokus pada penyyetelan parameter dan optimasi yang diperlukan untuk pengembangan aplikasi pendeteksian objek (Verma et al., 2024).



Gambar 10. PGI, related networks architectures and methods

Arsitektur YOLOv9 seperti yang ditunjukkan pada gambar 10, memperkenalkan inovasi seperti *Programmable Gradient Information (PGI)* dan *Generalized Efficient Layer Aggregation Network (GELAN)*. PGI membantu mengatasi *bottleneck* informasi pada jaringan *neural* dengan menghasilkan *gradient* yang andal di seluruh lapisan jaringan, memastikan konvergensi dan performa yang lebih baik. Sementara itu, GELAN meningkatkan efisiensi komputasi dengan mengintegrasikan blok komputasi dari arsitektur CSPNet dan ELAN, menyederhanakan modul *downsampling*, dan mengoptimalkan lapisan prediksi, sehingga meningkatkan pemanfaatan *parameter* dan efisiensi komputasi (Wang et al., 2024).

Proses training algoritma YOLO ini dilakukan menggunakan mesin GPU seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Kinerja Hasil *Training Model YOLOv9*

Model	Input Size	Epoch	Precision	Recall	mAP@0.5
YOLO-S	640	100	0.95	0.92	0.965

### Integrasi Hardware

Proses integrasi *hardware* sistem deteksi *bubble* pada mesin *leak tester* telah berhasil dilakukan, sebagaimana ditunjukkan pada gambar 11. Dengan menggunakan teknik *computer vision* canggih, sistem ini memanfaatkan algoritma yang telah terlatih untuk mengenali dan memahami pola *bubble* pada produk *aluminium casting* yang diuji. Perancangan *hardware* melibatkan integrasi komponen perangkat keras dan perangkat lunak dengan *Nvidia Jetson* sebagai inti sistem, *nvidia jetson* (Bokovoy et al., 2019; Jeon et al., 2021) ditempatkan di dekat mesin untuk mengelola data dari kamera digital (Suder, 2022) dan menjalankan algoritma *Object Detection* secara efisien.

Gambar 11. Integrasi *Hardware* pada Mesin *Leak Tester*

Kamera digital, sebagai komponen utama, dipasang dan dikonfigurasi pada *Nvidia Jetson* untuk mengambil gambar atau *video* dari area yang relevan pada mesin *leak tester*. Gambar yang dihasilkan kemudian dianalisis oleh algoritma untuk mendeteksi adanya *bubble*. Integrasi ini telah diuji dan terbukti bekerja dengan baik, memberikan visualisasi yang jelas dan mendetail mengenai kondisi produk *aluminium casting* selama proses pengujian, seperti yang ditunjukkan pada gambar 12. Integrasi ini memungkinkan proses deteksi *bubble* menjadi lebih efisien dan akurat, mendukung peningkatan kualitas produk secara keseluruhan.

Gambar 12. *Camera* yang dipasang pada Mesin *Leak Tester*

### Pengujian

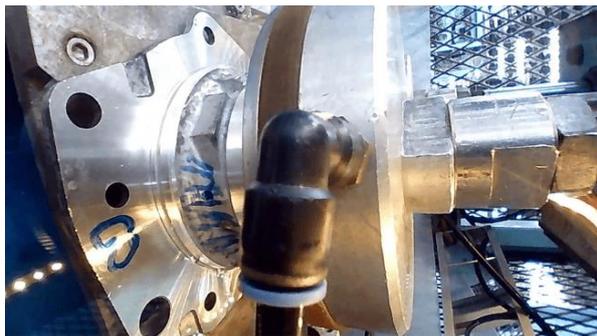
Pengujian terhadap sistem deteksi *bubble* pada mesin *leak tester* menggunakan metode *computer vision* dilakukan dengan tujuan untuk memilih algoritma terbaik yang akan digunakan dan diimplementasikan. Pengujian ini mengukur beberapa metrik seperti *Precision*, *Recall*, *mAP@0.5*, waktu deteksi, dan *fps (frame per second)* (Hidayatullah, 2021) dari algoritma YOLO. Selain itu, *confusion matrix* digunakan untuk mengukur kinerja masing-masing algoritma.

Data pengujian diambil langsung dari salah satu pabrik pembuatan *parts* otomotif saat mesin digunakan untuk produksi seperti yang dapat ditunjukkan pada gambar 13. Untuk memastikan algoritma dapat beradaptasi dengan kondisi baru, data pengujian tidak hanya diambil dari satu sudut. Pengujian dilakukan dengan mengambil data dari berbagai sudut berbeda untuk menilai kemampuan algoritma dalam mendeteksi *bubble* pada berbagai perspektif. Gambar 14 menunjukkan kamera yang dipasang di bagian lain mesin, dan gambar 15 menunjukkan gambar yang di-*capture* oleh kamera.

Gambar 13. *Dataset* untuk Pengujian



Gambar 14. Camera yang dipasang pada Mesin Leak Tester dengan posisi berbeda



Gambar 15. Dataset untuk pengujian dengan camera yang dipasang diposisi berbeda

Data pengujian dilabeli secara manual menggunakan CVAT (Boris Sekachev, 2023) untuk memastikan *ground truth* atau posisi sebenarnya dari *bubble*. Pengujian dilakukan dengan menghitung *confusion matrix* serta metrik-metrik lainnya seperti *precision* dan *recall*. Kode evaluasi algoritma digunakan untuk menghitung dan menampilkan metrik-metrik ini seperti yang ditunjukkan pada gambar 16.

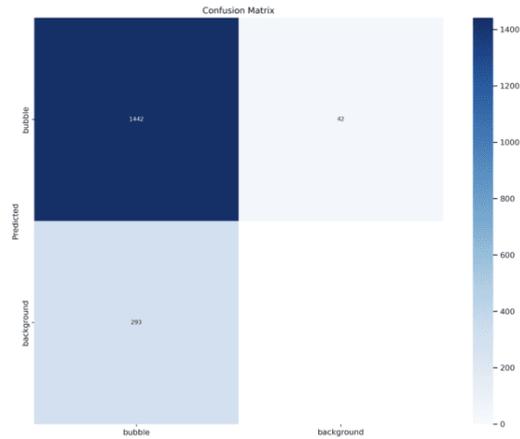
```
# Customize validation settings
validation_results = model.val(data="dataset.yaml", imgs=640, batch=16, conf=0.5, iou=0.5, device="0", plots=True)

Model summary (fused): 188 layers, 1152573 parameters, 8 gradients, 28.4 GMem
val: Scanning /content/datasets/datauji.cache... 1567 images, 222 backgrounds, 0 corrupt: 100% ██████████ 1567/1567 [00:00]
/usr/lib/python3.10/multiprocessing/popen_fork.py:166: RuntimeWarning: os.fork() was called. os.fork() is incompatible w
set: pid = os.fork()
Class Images Instances Box(P R mAP50 mAP50-95) 100% ██████████ 98/98 [00:11]
all 1567 1725 0.94 0.816 0.893 0.477
Speed: 0.1ms preprocess, 2.9ms inference, 0.0ms loss, 0.7ms postprocess per image
Results saved to runs/detect/val3
```

Gambar 16. Kode yang digunakan untuk menghitung metrik

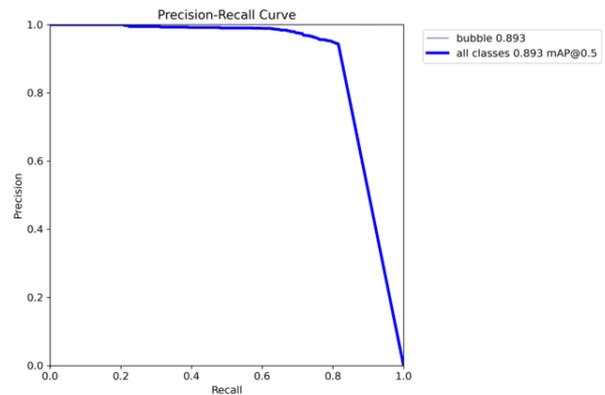
Dalam pengujian algoritma *YOLO* untuk deteksi *bubble*/gelembung gas menunjukkan bahwa algoritma ini memiliki performa yang sangat baik seperti yang ditunjukkan pada gambar 17, dengan 1442 prediksi benar untuk *bubble*, 42 prediksi salah untuk *bubble* dan 293 prediksi salah untuk *background*. *Confusion Matrix* ini mencerminkan jumlah prediksi yang benar dan salah untuk masing-masing kelas dengan sangat baik, menunjukkan

kemampuan deteksi yang tinggi dan jumlah kesalahan yang sangat kecil



Gambar 17. Confusion Matrix YOLO

Algoritma *YOLO* menunjukkan *mAP@0.50* sebesar 0.893, yang diilustrasikan pada gambar 18 dengan *Precision-Recall Curve* yang stabil mendekati 1 sepanjang hampir keseluruhan rentang *recall*, menandakan *precision* dan *recall* yang tinggi.



Gambar 18. PR Curve YOLO

Metrik pengujian yang lengkap untuk algoritma *YOLOv9* ditunjukkan pada tabel 4.

Tabel 4. Kinerja algoritma YOLO terhadap data uji

Metrik	YOLO
Precision	0.94
Recall	0.82
mAP@0.5	0.90
Waktu	100-120ms
Deteksi (ms)	
Frame per Second (fps)	10 fps

Berdasarkan perbandingan metrik kinerja algoritma terhadap data uji yang ditampilkan dalam Tabel IV, bahwa *YOLO* adalah algoritma yang cocok untuk digunakan dalam deteksi *bubble/gelembung* gas di lingkungan manufaktur. *YOLO* memiliki *precision* sebesar 0.94, *recall* sebesar 0.82 dan *mAP@0.5* sebesar 0.90, mengindikasikan akurasi yang baik dari algoritma dalam memprediksi data pengujian. Performa algoritma *YOLO* dalam mendeteksi *bubble/gelembung* yaitu 100 – 120 ms (*milliseconds*) per *frame* dari *streaming camera* sehingga didapatkan performa *fps* nya yaitu 10 *fps* (*frame per second*). Keunggulan dalam akurasi dan kecepatan yang dimiliki *YOLO* menjadikannya algoritma yang cocok untuk deteksi cacat pada produk hasil *aluminium casting* pada industri otomotif.

#### IV. PENUTUP

##### Kesimpulan

Penelitian ini berhasil mengimplementasikan teknologi *computer vision*, khususnya algoritma *Object Detection* seperti *YOLO* dalam upaya meningkatkan deteksi kebocoran pada produk suku cadang mobil di salah satu pabrik pembuatan *parts* otomotif. Berdasarkan hasil pengujian, algoritma *YOLO* menunjukkan kinerja yang baik dengan nilai *precision* sebesar 0.94, *recall* sebesar 0.82, dan *mAP@0.5* sebesar 0.90. Selain itu penggunaan teknologi ini juga berhasil mengatasi beberapa kelemahan dari metode manual, seperti subjektivitas dan ketidakpastian yang sering terjadi.

Implementasi mesin *Leak Tester* yang dilengkapi dengan teknologi *computer vision* memungkinkan proses deteksi menjadi lebih otomatis dan efisien, mengurangi potensi kesalahan manusia dan meningkatkan reliabilitas sistem pemantauan. Performa algoritma *YOLO* dalam mendeteksi *bubble/gelembung* gas yaitu 100 – 120 ms per *frame* dari *streaming camera* sehingga didapatkan performa *fps*-nya yaitu 10 *fps* (*frame per second*). Keunggulan dalam akurasi dan kecepatan yang dimiliki *YOLO* menjadikannya algoritma yang cocok untuk deteksi cacat pada produk hasil *aluminium casting*. Hal ini diharapkan dapat meningkatkan kualitas produk, efisiensi produksi, serta mengurangi risiko operasional dan kerugian finansial yang disebabkan oleh produk cacat.

##### Saran

Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi dan mengintegrasikan teknologi terbaru dalam bidang *deep learning* dan *computer*

*vision*, seperti algoritma deteksi objek yang menggunakan arsitektur *transformers*. Selain itu, pengujian lebih lanjut dengan berbagai kondisi operasional dan jenis produk yang berbeda dapat membantu meningkatkan generalisasi dan robustitas sistem. Penelitian juga dapat diarahkan untuk mengoptimalkan efisiensi *computational* dan *resource management* dari perangkat keras yang digunakan, seperti *Nvidia Jetson*, guna memastikan sistem dapat beroperasi dalam lingkungan produksi yang lebih luas dan beragam.

##### Dampak Ilmiah

Penelitian ini menunjukkan bahwa algoritma *YOLO* dapat diimplementasikan pada mesin *leak tester* untuk deteksi cacat pada produk hasil *aluminium casting* pada industri otomotif. Sehingga industri otomotif dapat beralih dari deteksi cacat pada produk hasil *aluminium casting* secara manual dengan pemantauan manusia menjadi pemantauan otomatis menggunakan metode *computer vision* yaitu algoritma *YOLO*.

Penelitian ini juga menunjukkan performa dan akurasi yang tinggi dalam mendeteksi *bubble / gelembung* gas pada produk hasil *aluminium casting* yang cacat. Algoritma *YOLO* dapat menjadi rekomendasi untuk industri otomotif dalam pengimplementasian pengecekan cacat dengan cara mempercepat waktu deteksi dan meningkatkan akurasi yang tinggi. Algoritma *YOLO* ini dapat meminimalisir kesalahan yang ditimbulkan oleh pemantauan manual yang menyebabkan *recall* produk atau klaim produk, meningkatkan kualitas produk dan standar keamanan, bahaya operasional dan dampak finansial bagi perusahaan.

#### V. DAFTAR PUSTAKA

- Aust, J., & Pons, D. (2022). Comparative Analysis of Human Operators and Advanced Technologies in the Visual Inspection of Aero Engine Blades. *Applied Sciences (Switzerland)*, 12(4). <https://doi.org/10.3390/app12042250>
- Bokovoy, A., Muravyev, K., & Yakovlev, K. (2019). *Real-time Vision-based Depth Reconstruction with NVidia Jetson*.
- Boris Sekachev. (2023). *Computer Vision Annotation Tool (CVAT) (2.8.2)*. Github. <https://doi.org/10.5281/zenodo.4009388>.
- Bunyamin, H. (2020, August 1). *Micro-average == Weighted macro-average*. [https://Hbunyamin.Github.Io/MI-2/Micro\\_and\\_Weighted\\_Macro\\_Averages/](https://Hbunyamin.Github.Io/MI-2/Micro_and_Weighted_Macro_Averages/).

- Chen, W., Li, Y., Tian, Z., & Zhang, F. (2023). 2D and 3D object detection algorithms from images: A Survey. In *Array* (Vol. 19). Elsevier B.V. <https://doi.org/10.1016/j.array.2023.100305>
- Gao, F., Lin, J., Ge, Y., Lu, S., & Zhang, Y. (2020). A Mechanism and Method of Leak Detection for Pressure Vessel: Whether, When, and How. *IEEE Transactions on Instrumentation and Measurement*, 69(9), 6004–6015. <https://doi.org/10.1109/TIM.2020.2969300>
- Gladkoff, S., Sorokina, I., Han, L., & Alekseeva, A. (2021). *Measuring Uncertainty in Translation Quality Evaluation (TQE)*. <http://arxiv.org/abs/2111.07699>
- Gowardipe, A. N., Deshmukh, A., Dimple, P., Deshmukh, A., & Garje, N. (2008). Leakage Testing Machine. *International Research Journal of Engineering and Technology*, 3957. [www.irjet.net](http://www.irjet.net)
- Hidayatullah, P. (2021). *Buku Sakti Deep Learning Computer Vision Menggunakan YOLO untuk Pemula*.
- Jeon, J., Jung, S., Lee, E., Choi, D., & Myung, H. (2021). Run Your Visual-Inertial Odometry on NVIDIA Jetson: Benchmark Tests on a Micro Aerial Vehicle. *IEEE Robotics and Automation Letters*, 6(3), 5332–5339. <https://doi.org/10.1109/LRA.2021.3075141>
- Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of Cognitive Neuroscience*, 33(10), 2017–2031. [https://doi.org/10.1162/jocn\\_a\\_01544](https://doi.org/10.1162/jocn_a_01544)
- Mrinal Walia. (2022, October 31). *Overfitting in Machine Learning and Computer Vision*. Roboflow Blog. <https://blog.roboflow.com/overfitting-machine-learning-computer-vision/>
- Pauline, E., Carreño-Alvarado, & Meza, G. R. (2020). *Water Leak Detection by Thermographic Image Analysis, In Laboratory Tests*. 15. <https://doi.org/10.3390/ecws-4-06440>
- Putri, A. S., & Primananda, F. (2021). Quality Control on Minimizing Defect Product on 20 OE Yarn. *Jurnal Ilmiah Teknik Industri*, 20(1), 81–88. <https://doi.org/10.23917/jiti.v20i1.12443>
- Qi, M., Xu, Y., Li, J., Kang, Y., & Wulabieke, Z. (2021). Microstructure refinement and corrosion resistance improvement mechanisms of a novel Al-Si-Fe-Mg-Cu-Zn alloy prepared by ultrasonic vibration-assisted rheological die-casting process. *Corrosion Science*, 180. <https://doi.org/10.1016/j.corsci.2020.109180>
- Suder, J. (2022). Parameters evaluation of cameras in embedded systems. *Przegląd Elektrotechniczny*, 98(9), 216–219. <https://doi.org/10.15199/48.2022.09.50>
- Verma, U., Kalia, A., & Sood, S. (2024). A Comparative Analysis of Deep Learning Based Object Detection Models. In *International Journal of Scientific Research & Engineering Trends* (Vol. 10, Issue 3).
- Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*. <http://arxiv.org/abs/2402.13616>
- Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., & Tang, S. (2020). Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. In *Materials* (Vol. 13, Issue 24, pp. 1–23). MDPI AG. <https://doi.org/10.3390/ma13245755>