

**DETEKSI ANOMALI PADA BOTOL PLASTIK
MENGUNAKAN *U-NET AUTOENCODER* DAN
*CONVOLUTIONAL NEURAL NETWORK***

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



Disusun Oleh:

MUSLIKAH

NIM 32602100094

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT

***ANOMALY DETECTION IN PLASTIC BOTTLES USING U-NET
AUTOENCODER AND CONVOLUTIONAL NEURAL NETWORK***

*Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University*



Arranged By:

**MUSLIKAH
NIM 32602100094**

***MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG***

2025

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**DETEKSI ANOMALI PADA BOTOL PLASTIK MENGGUNAKAN U-NET
AUTOENCODER DAN CONVOLUTIONAL NEURAL NETWORK**

**MUSLIKAH
NIM 32602100094**

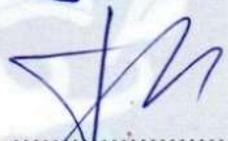
Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 3 Juni 2025

TIM PENGUJI UJIAN SARJANA :

Ghufron, ST., M.Kom
NIK. 210622056
(Ketua Penguji)

 5 - 6 - 2025

Moch Taufik, S.T., MIT
NIK. 210604034
(Anggota Penguji)

 5 - 6 - 2025

Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D
NIK. 210600017
(Pembimbing)

 5 - 6 - 2025

Semarang, 3 Juni 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung



Moch Taufik, ST., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Muslikah

NIM : 32602100094

Judul Tugas Akhir : Deteksi Anomali Pada Botol Plastik Menggunakan
U-Net Autoencoder Dan Convolutional Neural Network

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 3 Juni 2025

Yang Menyatakan,



Muslikah

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muslikah
NIM : 32602100094
Program Studi : Teknik Informatika
Fakultas : Teknologi industri
Alamat Asal : Kendal, Jawa Tengah

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **“Deteksi Anomali Pada Botol Plastik Menggunakan U-Net Autoencoder Dan Convolutional Neural Network”** Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 3 Juni 2025

Yang menyatakan,



Muslikah

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Deteksi Anomali Pada Botol Plastik Menggunakan *U-Net Autoencoder* Dan *Convolutional Neural Network*” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya dukungan dari berbagai pihak, materi maupun teknis, oleh karena itu penulis mengucapkan terima kasih kepada:

1. Rektor Universitas Islam Sultan Agung Bapak Prof, Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN.Eng.
3. Kepala program studi Teknik Informatika Bapak Moch. Taufik, ST., MIT
4. Dosen Pembimbing Bapak Imam Much Ibnu Subroto, ST., M.Sc., Ph.D yang telah memberikan banyak nasehat dan saran
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini serta memberikan dukungan dan doa agar tugas akhir ini dapat berjalan dengan lancar.
6. Dan pada semua pihak yang tidak bisa saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun ilmu pengetahuan dalam penyusunan laporan sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 3 Juni 2025



Muslikah

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN TUGAS AKHIR	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
ABSTRAK	xi
BAB I PENDAHULUAN.....	2
1.1 Latar Belakang.....	2
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.2.1 Deteksi Anomali	7
2.2.2 <i>Computer Vision</i>	8
2.2.3 <i>Unsupervised learning</i>	8
2.2.4 <i>U-Net Autoencoder</i>	9
2.2.5 <i>Deep learning</i>	11
2.2.6 <i>Convolutional Neural Network (CNN)</i>	11
2.2.7 <i>Thresholding</i>	12
2.2.8 <i>Python</i>	13
2.2.9 <i>TensorFlow</i>	14
BAB III METODE PENELITIAN	15

3.1	Metode Penelitian	15
3.2	Studi Literatur	15
3.3	Alur Perancangan Sistem	16
3.4	Pengumpulan <i>Dataset</i>	18
3.5	<i>Preprocessing</i>	20
3.6	Pembagian <i>Dataset</i>	23
3.7	<i>Library</i> yang di gunakan	24
3.8	Parameter Model	25
3.9	Pelatihan Model	26
3.10	Evaluasi model	27
BAB IV HASIL DAN ANALISIS PENELITIAN		29
4.1	Hasil dan Analisis	29
4.1.1	Pengumpulan data	29
4.1.2	<i>Preprocessing</i>	29
4.1.3	<i>Training Model</i>	33
4.2	Hasil Evaluasi Model	40
4.2.1	<i>Confusion Matrix</i>	41
4.2.2	Hasil Deteksi	44
4.3	<i>Deployment</i>	45
4.3.1	<i>Streamlit</i>	45
BAB V KESIMPULAN DAN SARAN		51
5.1	Kesimpulan	51
5.2	Saran	51
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

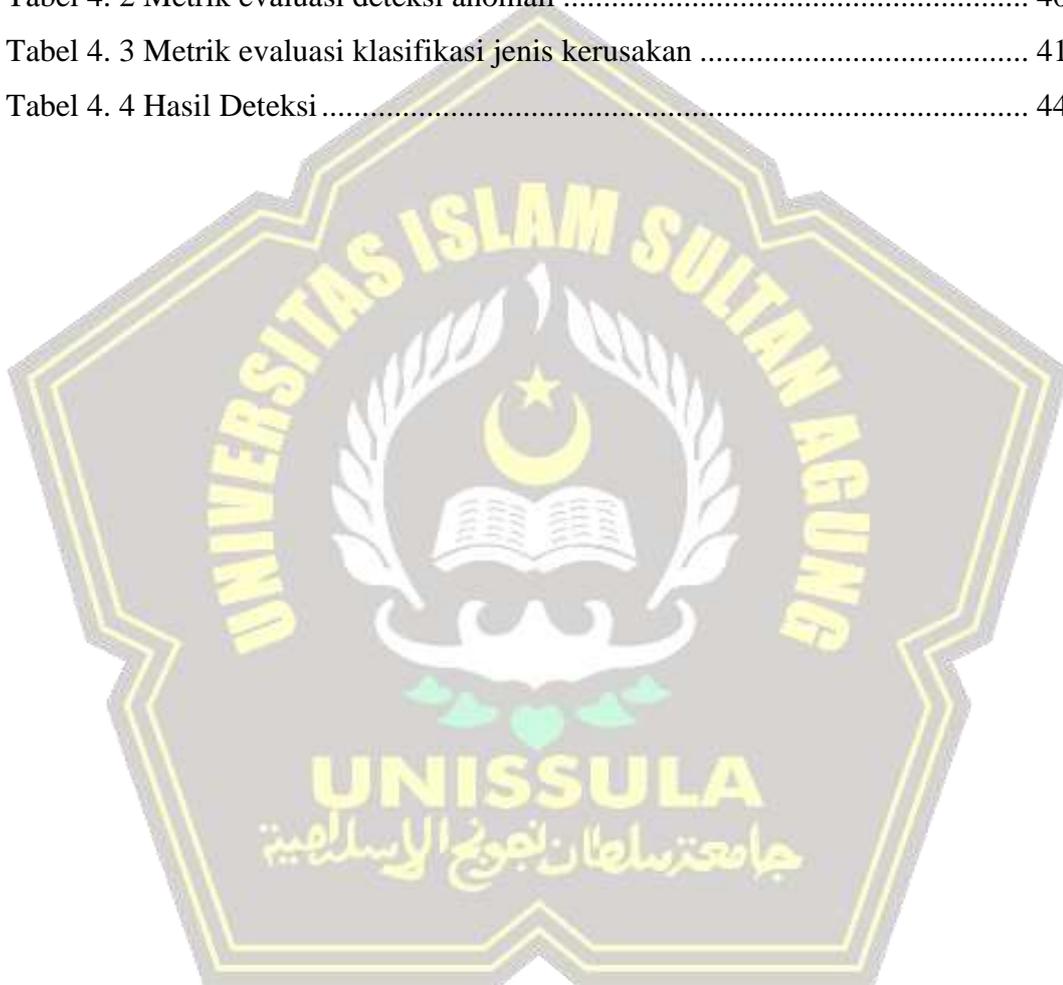
Gambar 2. 1 <i>Arsitektur U-Net Autoencoder</i> (Sharma, 2023).....	10
Gambar 2. 2 Algoritma CNN	12
Gambar 2. 3 <i>Python</i>	13
Gambar 2. 4 <i>Framework TensorFlow</i>	14
Gambar 3. 1 <i>Flowchart</i> alur perancangan sistem.....	16
Gambar 3. 2 <i>Flowchart</i> alur kerja sistem.....	17
Gambar 3. 3 Botol Rusak Kecil	18
Gambar 3. 4 Botol Rusak Besar	19
Gambar 3. 5 Botol Terkontaminasi.....	19
Gambar 3. 6 Botol Bagus	20
Gambar 4. 1 Konversi gambar ke <i>grayscale</i>	29
Gambar 4. 2 <i>Resize</i> gambar.....	30
Gambar 4. 3 Normalisasi piksel	30
Gambar 4. 4 <i>Optimizer</i> dan <i>parameter Training</i>	34
Gambar 4. 5 <i>Optimizer parameter cnn classifier</i>	35
Gambar 4. 6 Grafik <i>Training and validation loss</i> pada <i>U-Net Autoencoder</i>	36
Gambar 4. 7 Distribusi rekonstruksi <i>error</i>	37
Gambar 4. 8 <i>Classifier Accuracy</i>	38
Gambar 4. 9 <i>Classifier loss</i>	39
Gambar 4. 10 <i>Confusion matrix anomaly detection</i>	42
Gambar 4. 11 <i>Confusion matrix classifier</i>	43
Gambar 4. 12 Authtoken	45
Gambar 4. 13 <i>Ngrok Agents</i>	46
Gambar 4. 14 Install <i>library streamlit pyngrok</i>	46
Gambar 4. 15 Instalasi <i>Streamlit</i> Pyngrok serta Konfigurasi Authtoken Ngrok... 46	
Gambar 4. 16 <i>Output</i> Konfirmasi <i>Streamlit</i> dan Alamat Akses Ngrok	47
Gambar 4. 17 Dashboard <i>web</i> deteksi anomali	47
Gambar 4. 18 Mengunggah gambar	48
Gambar 4. 19 Hasil deteksi gambar botol yang terdeteksi normal	49

Gambar 4. 20 Hasil deteksi gambar botol yang terdeteksi kontaminasi..... 50



DAFTAR TABEL

Tabel 3. 1 Pembagian <i>Dataset</i> CNN	23
Tabel 3. 2 Pembagian <i>Dataset</i> CNN	23
Tabel 3. 3 <i>Library</i> yang digunakan untuk Sistem Deteksi.....	24
Tabel 4. 1 Augmentasi gambar	31
Tabel 4. 2 Metrik evaluasi deteksi anomali	40
Tabel 4. 3 Metrik evaluasi klasifikasi jenis kerusakan	41
Tabel 4. 4 Hasil Deteksi	44



ABSTRAK

Dalam industri manufaktur botol plastik, cacat seperti retakan, kontaminasi, dan deformasi dapat mengganggu fungsi kemasan serta menghambat proses produksi. Inspeksi visual manual yang umum digunakan cenderung tidak konsisten, lambat, dan kurang akurat untuk skala produksi tinggi. Oleh karena itu, diperlukan sistem otomatis berbasis teknologi yang dapat mendeteksi cacat secara cepat dan andal. Penelitian ini mengembangkan sistem deteksi dan klasifikasi cacat botol plastik menggunakan metode *deep learning*, yaitu *U-Net Autoencoder* untuk mendeteksi anomali dan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan jenis cacat. *U-Net Autoencoder* dilatih untuk merekonstruksi gambar botol normal dan mengenali anomali berdasarkan perbedaan citra asli dengan hasil rekonstruksi. Citra anomali kemudian diklasifikasikan menjadi tiga jenis cacat: *broken large*, *broken small*, dan *Contamination*. Hasil evaluasi menunjukkan bahwa sistem deteksi anomali dengan *U-Net Autoencoder* mencapai akurasi sebesar 99,67%, *precision* 99,65%, *recall* 100%, dan *F1-score* 99,83%. Model CNN untuk klasifikasi mencapai akurasi 85%. Hasil ini menunjukkan bahwa pendekatan gabungan yang dirancang mampu mendeteksi dan mengklasifikasikan cacat secara otomatis, sehingga dapat mendukung peningkatan mutu pengawasan produk dalam proses produksi botol plastik

Kata Kunci: Deteksi Anomali, Botol Plastik, *U-Net Autoencoder*, CNN, *Deep learning*, Klasifikasi Cacat

ABSTRACT

In the plastic bottle manufacturing industry, defects such as cracks, contamination, and deformation can disrupt packaging functions and hinder the production process. Manual visual inspection, which is still commonly used, tends to be inconsistent, time-consuming, and inaccurate for high-scale production. Therefore, a technology-based automated system is needed to detect defects quickly and reliably. This study developed an automatic defect detection and classification system for plastic bottles using deep learning methods: U-Net Autoencoder for anomaly detection and Convolutional Neural Network (CNN) for defect type classification. The U-Net Autoencoder was Trained to reconstruct normal bottle images and detect anomalies based on the differences between the original and reconstructed images. Detected anomalies were then classified into three defect categories: broken large, broken small, and Contamination. Evaluation results showed that the anomaly detection system achieved an Accuracy of 99.67%, precision 99.65%, recall 100%, and F1-score 99.83%. The CNN classifier achieved an Accuracy of 85%. These results indicate that the combined approach developed in this study is capable of automatically detecting and classifying defects, thereby supporting improved product quality monitoring in the plastic bottle manufacturing process.

Keywords: Anomaly Detection, Plastic Bottles, *U-Net Autoencoder*, CNN, *Deep learning*, Defect Classification

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam industri manufaktur plastik, khususnya pada lini produksi botol plastik, kualitas produk merupakan aspek yang sangat krusial. Kualitas kemasan tidak hanya memengaruhi daya tahan produk dan efisiensi distribusi, tetapi juga berpengaruh besar terhadap kepuasan konsumen. Botol plastik yang mengalami cacat seperti goresan, retakan, noda, atau deformasi sering kali dianggap tidak layak jual dan berpotensi menyebabkan kerugian finansial serta menurunkan reputasi perusahaan. Berdasarkan penelitian di salah satu perusahaan minuman botol plastik, tingkat cacat botol plastik mencapai 2% dari total produksi bulanan, yaitu sebanyak 6.421 botol dari 287.897 botol. Jenis cacat yang paling dominan adalah kotor material (30%), kotor oli (18%), dan cacat melipat (15%), yang umumnya disebabkan oleh faktor manusia, mesin, dan material (Wahid dkk., 2023).

Hingga kini, proses inspeksi visual secara manual masih menjadi metode utama dalam kontrol kualitas produk. Namun, pendekatan ini memiliki berbagai kelemahan, seperti ketergantungan pada persepsi manusia yang bersifat subjektif, kurang konsisten, dan tidak efisien dalam skala produksi besar atau sistem *real-time*. Oleh karena itu, diperlukan solusi berbasis teknologi yang mampu mendeteksi cacat secara otomatis, akurat, dan efisien. Perkembangan teknologi *Computer Vision* dan *deep learning* menawarkan solusi dalam mengatasi permasalahan ini. Salah satu pendekatan yang sering digunakan adalah algoritma *Convolutional Neural Network* (CNN), yang mampu mengenali pola citra dan mengklasifikasikan objek secara otomatis. Namun, tantangan muncul ketika *Dataset* tidak seimbang, di mana jumlah data produk cacat jauh lebih sedikit dibanding produk normal. Selain itu, jenis cacat yang bervariasi dan tidak selalu diketahui sebelumnya menjadi kendala dalam pendekatan *supervised learning* (Chen dkk., 2021). Untuk menjawab tantangan tersebut, metode

Unsupervised learning seperti *Autoencoder* menjadi solusi yang tepat. *Autoencoder* bekerja dengan mempelajari pola dari data normal dan mendeteksi *anomali* berdasarkan penyimpangan dari pola tersebut (Sabri, 2022).

Seiring berkembangnya teknologi kecerdasan buatan, pendekatan berbasis *deep learning* mulai banyak diterapkan untuk membantu proses deteksi cacat secara otomatis. Salah satu metode yang efektif untuk mendeteksi anomali adalah *Autoencoder*, terutama arsitektur *U-Net* yang mampu merekonstruksi gambar *input* dan mengidentifikasi perbedaan antara citra asli dan hasil rekonstruksi sebagai indikasi adanya cacat. Sementara itu, untuk mengidentifikasi jenis cacat yang terdeteksi, model *Convolutional Neural Network* (CNN) digunakan karena kemampuannya yang tinggi dalam klasifikasi citra. Kombinasi dari *U-Net Autoencoder* untuk deteksi anomali dan CNN untuk klasifikasi jenis cacat memberikan pendekatan yang komprehensif dan efisien dalam mendukung sistem inspeksi otomatis berbasis citra (Singh dkk., 2023).

Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem deteksi dan klasifikasi cacat pada botol plastik menggunakan metode *deep learning*, sehingga diharapkan dapat meningkatkan efisiensi, konsistensi, dan akurasi proses *quality control* dalam industri manufaktur.

1.2 Perumusan Masalah

1. Bagaimana membangun model *U-Net Autoencoder* untuk mendeteksi anomali atau cacat pada citra botol plastik secara otomatis.
2. Bagaimana menentukan *threshold* rekonstruksi *error* yang optimal untuk membedakan botol normal dan cacat.
3. Bagaimana menerapkan model CNN untuk mengklasifikasikan jenis cacat pada botol plastik yang telah terdeteksi sebagai *anomaly*.

1.3 Batasan Masalah

1. Penelitian ini tidak mencakup klasifikasi berdasarkan jenis botol plastik. Sistem yang dikembangkan difokuskan untuk mendeteksi dan mengklasifikasikan cacat pada botol plastik secara umum, tanpa membedakan bentuk, ukuran, atau merek botol. Gambar selain botol plastik tidak termasuk dalam ruang lingkup sistem.
2. Model hanya mendeteksi tiga jenis cacat: *Broken large*, *Broken small*, dan *Contamination*.
3. Deteksi anomali dilakukan hanya berdasarkan perbedaan antara gambar asli dan hasil rekonstruksi *Autoencoder*.
4. Klasifikasi cacat dilakukan hanya setelah anomali terdeteksi, menggunakan model CNN.

1.4 Tujuan

Penelitian ini bertujuan untuk membangun sistem otomatis yang dapat mendeteksi anomali dan mengklasifikasikan cacat pada botol plastik menggunakan arsitektur *U-Net Autoencoder* dan CNN untuk meningkatkan akurasi dan efisiensi dalam proses kontrol kualitas produk, khususnya pada deteksi cacat botol plastik tanpa memperhitungkan jenis atau variasi bentuk botol.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan solusi teknologi yang efektif bagi industri manufaktur plastik dalam mengotomatiskan proses inspeksi visual, sehingga dapat mengurangi kesalahan manusia, meningkatkan kualitas produk, dan menekan jumlah produk cacat.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang akan di pakai penulis dalam pembuatan laporan tugas akhir ini sebagai berikut:

BABI : PENDAHULUAN

Pada bab pendahuluan menyajikan permasalahan yang dibahas pada penelitian ini yaitu latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini disajikan informasi mengenai tinjauan Pustaka dari penelitian sebelumnya dan dasar teori yang digunakan.

BAB III : METODE PENELITIAN

Pada bab ini disajikan tentang metologi penelitian yang digunakan dan rancangan sistem deteksi *anomali* pada botol plastik menggunakan *U-Net Autoencoder* dan *Convolutional Neural Network* yang akan dibuat.

BAB IV : HASIL DAN ANALISA

Pada bab ini akan disajikan tentang hasil dan analisa dari sistem deteksi *anomali* pada botol plastik menggunakan *U-Net Autoencoder* dan *Convolutional Neural Network* yang telah dibuat.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini disajikan tentang kesimpulan dan saran dari seluruh rangkaian penelitian yang dilakukan.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam industri manufaktur, khususnya pada produksi botol plastik, deteksi cacat merupakan aspek krusial untuk memastikan kualitas produk. Proses deteksi manual seringkali tidak konsisten dan rawan kesalahan akibat kelelahan operator dan keterbatasan visual, terutama saat volume produksi tinggi. Oleh karena itu, implementasi sistem otomatis berbasis visi komputer menjadi solusi yang kian diminati karena dapat meningkatkan efisiensi dan akurasi deteksi cacat secara *real-time* (Soesilo dkk., 2024).

Perkembangan teknologi *deep learning* memberikan kemajuan signifikan dalam bidang visi komputer, terutama dalam pemrosesan citra digital. *Deep learning* memungkinkan mesin untuk secara otomatis mempelajari representasi fitur dari data mentah melalui jaringan saraf tiruan berlapis-lapis, tanpa memerlukan proses ekstraksi fitur manual yang kompleks. Dalam konteks deteksi cacat, pendekatan *deep learning* mampu mengenali pola visual halus dan kompleks yang tidak dapat diidentifikasi oleh metode *konvensional* (Mohandas dkk., 2024).

Salah satu arsitektur *deep learning* yang populer untuk deteksi anomali adalah *Autoencoder*. *Autoencoder* bekerja dengan cara merekonstruksi *input*-nya sendiri setelah melalui proses kompresi dan dekompresi, sehingga dapat digunakan untuk mengukur deviasi antara data *input* dan hasil rekonstruksi. Pada sistem deteksi cacat, *Autoencoder* dilatih menggunakan data *non-cacat* agar mampu mempelajari representasi “normal”. Ketika model diberikan *input* berupa citra cacat, hasil rekonstruksi biasanya buruk dan menghasilkan *error* yang tinggi, yang dapat dijadikan indikator adanya anomali.

Sebuah studi oleh (Ganesh dkk., 2020) mengusulkan strategi baru untuk transformasi data *input* menjadi representasi yang lebih bermakna untuk deteksi anomali, dengan memanfaatkan *Autoencoder* untuk mengkodekan data *input* dan menggunakan tiga faktor: representasi tersembunyi, vektor residu rekonstruksi, dan *error* rekonstruksi, sebagai representasi baru untuk data *input* tersebut. Untuk memperkuat kemampuan rekonstruksi, penelitian ini menggunakan *U-Net Autoencoder*. *U-Net* merupakan arsitektur *Convolutional Neural Network* (CNN) yang awalnya dikembangkan untuk segmentasi gambar medis, tetapi telah terbukti efektif juga dalam berbagai aplikasi pemrosesan citra industri karena kemampuannya dalam mempertahankan detail spasial melalui *skip connections*. Struktur *encoder-decoder* simetris dari *U-Net* memungkinkan model untuk merekonstruksi citra *input* dengan lebih akurat, menjadikannya sangat cocok untuk mendeteksi deviasi kecil pada gambar botol plastik. Dalam konteks deteksi cacat manufaktur, (Singh dkk., 2023) mengaplikasikan *U-Net* untuk segmentasi otomatis cacat pada manufaktur aditif, menunjukkan bahwa arsitektur ini efektif dalam mendeteksi porositas dan retakan dengan akurasi tinggi (Arif dkk., 2025).

Selain mendeteksi keberadaan cacat, sistem juga dirancang untuk mengklasifikasikan jenis cacat yang ditemukan. Untuk tugas klasifikasi ini digunakan CNN (*Convolutional neural network*) karena kemampuannya dalam mengekstraksi fitur spasial dari gambar dan mengenali pola visual secara hierarkis. CNN telah menjadi pendekatan utama dalam tugas-tugas klasifikasi citra karena struktur berlapisnya yang terdiri dari konvolusi, *pooling*, dan lapisan dense. Dalam penelitian ini, CNN digunakan untuk membedakan jenis cacat pada botol, seperti *broken large*, *broken small*, dan *Contamination*, mengembangkan sistem inspeksi cacat permukaan otomatis berdasarkan *Autoencoder* dan *fully convolutional neural networks*, menunjukkan bahwa kombinasi ini efektif dalam mendeteksi berbagai jenis cacat permukaan secara otomatis (Darsin dkk., 2020).

Agar performa model lebih stabil dan generalisasi meningkat, dilakukan proses augmentasi data, terutama karena data cacat cenderung lebih sedikit dibandingkan data normal. Augmentasi dilakukan melalui transformasi geometris dan penambahan *noise*, seperti *Gaussian noise* dan *salt-and-pepper noise*. Teknik ini terbukti dapat membantu mengurangi *overfitting* dan memperkaya variasi data pelatihan. Dalam studi oleh (Kasnanda Bintang dkk., 2024), penggunaan *transfer learning* pada arsitektur *U-Net* menunjukkan peningkatan performa dalam deteksi cacat permukaan baja, terutama ketika data pelatihan terbatas.

Evaluasi sistem dilakukan menggunakan metrik klasifikasi seperti *Accuracy*, *precision*, *recall*, dan *F1-score*, yang merupakan metrik umum dalam mengevaluasi performa model klasifikasi dan deteksi anomali. Selain itu, *Confusion matrix* digunakan untuk memvisualisasikan performa model secara lebih rinci terhadap prediksi benar dan salah pada tiap kelas. Pemilihan *threshold* untuk deteksi anomali berdasarkan distribusi *error* rekonstruksi dari data normal, umumnya menggunakan persentil ke-95 sebagai batas deteksi. (Akçay dkk., 2022) mengusulkan metode DASVDD yang menggabungkan *Autoencoder* dengan *Support Vector Data Descriptor* untuk deteksi anomali, menunjukkan bahwa kombinasi ini dapat meningkatkan akurasi deteksi anomali dengan meminimalkan volume *hypersphere* yang melingkupi representasi laten data normal.

2.2 Dasar Teori

2.2.1 Deteksi Anomali

Deteksi anomali merupakan proses identifikasi pola atau sampel yang tidak sesuai dengan perilaku mayoritas data. Dalam dunia industri, hal ini berarti mendeteksi produk cacat yang menyimpang dari spesifikasi standar. Teknik ini sangat penting ketika distribusi data cacat sangat tidak seimbang atau ketika jenis cacat tidak diketahui sebelumnya. Dalam penelitian ini, deteksi anomali dilakukan dengan pendekatan *unsupervised* menggunakan *U-Net Autoencoder*. Citra normal direkonstruksi dengan *error* rendah, sementara citra cacat akan menghasilkan *error* rekonstruksi yang tinggi.

Perbedaan ini diukur menggunakan metrik seperti *Mean Squared Error* (MSE), yang kemudian dibandingkan dengan nilai ambang batas (*threshold*). Citra dianggap anomali jika *error*-nya melebihi *threshold* tersebut. Pendekatan ini sangat berguna untuk aplikasi industri karena tidak memerlukan label data cacat yang banyak, serta mampu mendeteksi variasi cacat baru (Tang dkk., 2023).

2.2.2 *Computer Vision*

Computer Vision adalah bidang dalam kecerdasan buatan yang berfokus pada bagaimana komputer dapat “melihat” dan memahami informasi dari gambar atau video secara otomatis, menyerupai cara kerja penglihatan manusia. Teknologi ini memungkinkan mesin untuk melakukan analisis visual seperti deteksi objek, klasifikasi citra, segmentasi, dan pelacakan pergerakan objek dalam lingkungan nyata. Dalam dunia industri, terutama pada lini produksi, *Computer Vision* dimanfaatkan secara luas dalam sistem inspeksi otomatis untuk mengidentifikasi cacat produk dengan cepat dan konsisten, tanpa perlu intervensi manusia. Sistem berbasis *Computer Vision* mengandalkan teknik pengolahan citra digital dan algoritma pembelajaran mesin, terutama *deep learning*, yang memungkinkan ekstraksi fitur secara otomatis dan akurat dari citra *input*. Dalam penelitian ini, pendekatan *Computer Vision* digunakan untuk memproses citra botol plastik sebagai *input*, yang kemudian dianalisis menggunakan *U-Net Autoencoder* untuk mendeteksi anomali serta *Convolutional Neural Network* (CNN) untuk mengklasifikasikan jenis cacat. Kombinasi teknik ini menunjukkan bagaimana *Computer Vision* dapat diintegrasikan dalam sistem *quality control* yang efisien dan adaptif terhadap variasi visual yang kompleks pada produk manufaktur (Mohandas dkk., 2024).

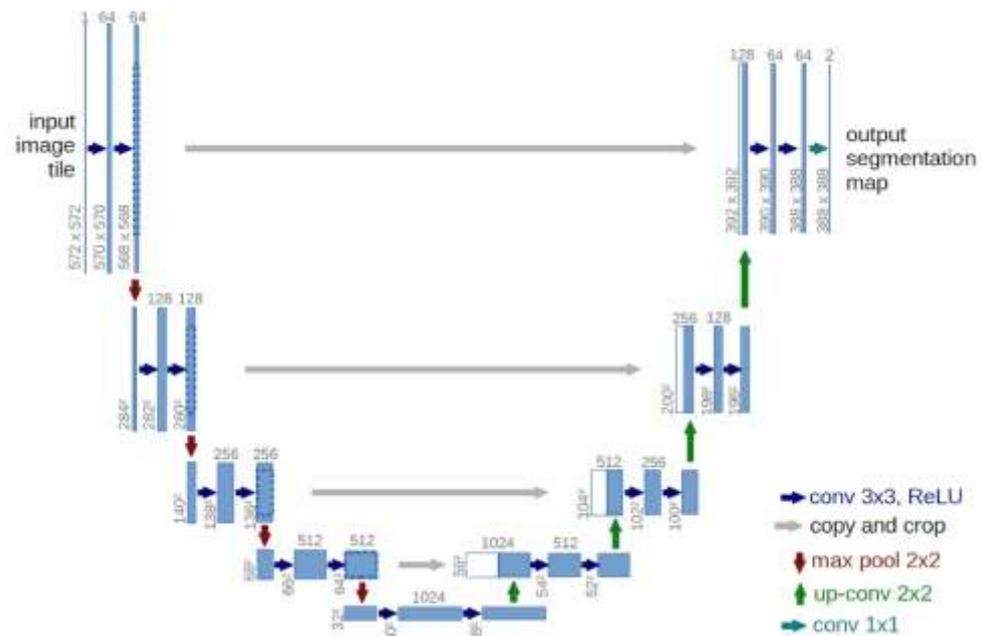
2.2.3 *Unsupervised learning*

Unsupervised learning adalah salah satu pendekatan dalam *machine learning* di mana model dilatih menggunakan data tanpa label atau anotasi. Tidak seperti *supervised learning* yang membutuhkan pasangan *input-output* selama proses pelatihan, *Unsupervised learning* hanya mengandalkan

struktur atau pola intrinsik dalam data untuk menemukan informasi tersembunyi. Tujuan utama dari pendekatan ini adalah untuk mengidentifikasi pola, kelompok, atau *outlier* dalam data tanpa pengetahuan eksplisit tentang kategori target. Dalam konteks deteksi cacat produk, seperti botol plastik, *Unsupervised learning* sangat berguna karena sering kali data cacat jauh lebih sedikit atau tidak tersedia dalam jumlah yang cukup untuk membangun model klasifikasi yang andal. Salah satu metode populer dalam *Unsupervised learning* adalah *Autoencoder*, yang bekerja dengan merekonstruksi *input* dan menghitung selisih antara citra asli dan citra hasil rekonstruksi. Jika selisih tersebut melebihi ambang batas tertentu, maka *input* dianggap sebagai anomali. Pendekatan ini cocok untuk sistem inspeksi otomatis karena memungkinkan pendeteksian cacat tanpa perlu melabeli setiap jenis cacat secara manual, sehingga lebih efisien dan fleksibel dalam implementasinya (Naeem dkk., 2023).

2.2.4 U-Net Autoencoder

Autoencoder adalah model jaringan saraf tiruan yang dilatih untuk merekonstruksi *input*-nya sendiri. Arsitekturnya terdiri dari dua bagian utama: *encoder* yang memampatkan *input* ke representasi laten, dan *decoder* yang merekonstruksi *input* dari representasi tersebut. *Autoencoder* sangat efektif dalam mendeteksi anomali karena model dilatih hanya dengan data normal, sehingga gagal dalam merekonstruksi data yang menyimpang (cacat). Dalam penelitian ini, digunakan *U-Net Autoencoder*, yaitu varian *Autoencoder* dengan struktur simetris berbentuk "U" yang menggabungkan *skip connections* antara lapisan *encoder* dan *decoder*. *Skip connections* ini memungkinkan pelestarian informasi spasial yang sangat penting dalam pengolahan citra, sehingga rekonstruksi menjadi lebih akurat. *U-Net* awalnya dikembangkan untuk segmentasi citra medis, namun strukturnya terbukti efisien dalam tugas deteksi anomali karena dapat mengungkap detail visual kecil yang tidak sesuai (Kalla, 2025).



Gambar 2. 1 Arsitektur *U-Net Autoencoder* (Sharma, 2023)

Pada gambar 2.1 merupakan arsitektur *U-Net Autoencoder* yang digunakan dalam sistem deteksi anomali botol plastik. *U-Net* merupakan jaringan konvolusional berbasis *encoder-decoder* yang dirancang khusus untuk tugas segmentasi citra, namun juga efektif dalam deteksi anomali melalui proses rekonstruksi. Bagian *encoder* (di sisi kiri) berfungsi untuk mengekstraksi fitur penting dari citra masukan dengan menerapkan operasi konvolusi 3x3 dengan fungsi aktivasi ReLU diikuti oleh *max pooling* 2x2 untuk mengurangi dimensi spasial dan memperdalam representasi fitur. Setelah mencapai bagian terdalam jaringan (*bottleneck*), proses dilanjutkan ke bagian *decoder* (di sisi kanan), di mana citra direkonstruksi kembali melalui operasi *up-convolution* (*up-conv*) 2x2 dan digabungkan dengan fitur dari tahap *encoder* yang selevel menggunakan teknik *copy and crop* (Kalla, 2025). Hal ini memungkinkan jaringan memanfaatkan konteks lokal dan global secara bersamaan.

Pada tahap akhir, digunakan konvolusi 1x1 untuk menghasilkan peta segmentasi atau hasil rekonstruksi. Dalam konteks deteksi anomali, hasil rekonstruksi ini dibandingkan dengan citra *input* asli untuk menghitung

Reconstruction Error. Error yang tinggi menunjukkan adanya perbedaan signifikan yang mengindikasikan kemungkinan cacat pada botol plastik, sedangkan *error* yang rendah mengindikasikan citra botol normal. Struktur simetris dan *skip connection* pada *U-Net* memungkinkan rekonstruksi citra yang lebih akurat, sehingga sangat efektif untuk mendeteksi kerusakan kecil sekalipun pada permukaan botol (Singh dkk., 2023).

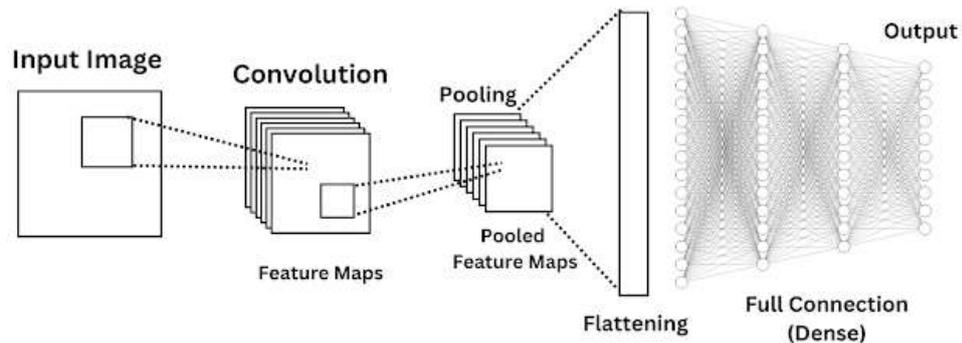
2.2.5 Deep learning

Deep learning adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan (*artificial neural networks*) dengan banyak lapisan tersembunyi (*hidden layers*) untuk mempelajari pola kompleks dari data berukuran besar. Model *deep learning* mampu melakukan ekstraksi fitur secara otomatis tanpa intervensi manusia, sehingga sangat cocok untuk pengolahan citra dalam industri. Pendekatan ini telah banyak diterapkan dalam sistem inspeksi visual otomatis untuk mendeteksi cacat produk dengan presisi tinggi. Dalam penelitian ini, digunakan model *deep learning* berbasis *Autoencoder* dan *Convolutional Neural Network* (CNN) karena kedua model tersebut telah terbukti efektif dalam tugas deteksi anomali dan klasifikasi citra industri. *Deep learning* juga menawarkan keunggulan dalam hal skalabilitas, generalisasi model, dan kemampuan adaptasi terhadap variasi data *input* yang tinggi (Kasnanda Bintang dkk., 2024).

2.2.6 Convolutional Neural Network (CNN)

CNN merupakan arsitektur *deep learning* yang paling banyak digunakan dalam pengolahan citra karena kemampuannya dalam mengekstraksi fitur spasial dari gambar. CNN terdiri dari beberapa jenis lapisan seperti *convolutional layer*, *pooling layer*, dan *fully connected layer*. Lapisan konvolusi mengekstraksi fitur penting seperti tepi, tekstur, dan pola, sedangkan lapisan *pooling* mengurangi dimensi data untuk mempercepat proses dan menghindari *overfitting*. *Fully connected layer* bertindak sebagai pengklasifikasi berdasarkan fitur yang telah dipelajari sebelumnya. Dalam sistem ini, CNN digunakan untuk mengklasifikasikan jenis cacat botol

plastik menjadi tiga kelas: *Broken large*, *Broken small*, dan *Contamination*. CNN dipilih karena telah terbukti efektif dalam tugas klasifikasi citra industri dengan akurasi tinggi dan kemampuan generalisasi yang baik terhadap berbagai jenis cacat visual (Choi & Lee, 2024).



Gambar 2. 2 Algoritma CNN

Pada gambar 2.2 menunjukkan arsitektur dasar dari *Convolutional Neural Network* (CNN) yang digunakan untuk pemrosesan citra. CNN bekerja dalam beberapa tahap, dimulai dari *input image*, yaitu citra yang dimasukkan ke jaringan. Selanjutnya, pada tahap *convolution*, *filter* diterapkan untuk mengekstraksi fitur penting seperti tepi atau pola, menghasilkan *feature maps*. Kemudian, *pooling* dilakukan untuk mereduksi dimensi dan mempertahankan informasi utama, menghasilkan *pooled feature maps*. Hasil *pooling* ini diubah menjadi vektor melalui proses *flattening*, lalu diteruskan ke *fully connected (dense) layer* untuk proses klasifikasi. Akhirnya, CNN menghasilkan *output* berupa prediksi berdasarkan fitur yang telah dipelajari (Wang dkk., 2022).

2.2.7 Thresholding

Thresholding adalah teknik penting dalam sistem deteksi anomali yang digunakan untuk memisahkan antara data normal dan anomali berdasarkan nilai numerik tertentu. Dalam konteks penelitian ini, *threshold* ditentukan dari distribusi *error* rekonstruksi yang dihasilkan oleh *U-Net Autoencoder* pada data validasi normal. *Threshold* ini mewakili batas toleransi sistem terhadap kesalahan rekonstruksi. Jika *error* citra uji

melebihi nilai *threshold*, maka citra dianggap sebagai anomali. Sebaliknya, jika *error* berada di bawah *threshold*, citra dikategorikan sebagai normal. Penentuan *threshold* yang akurat sangat krusial karena terlalu rendah akan menyebabkan banyak *false positive* (normal dianggap cacat), sedangkan terlalu tinggi dapat menyebabkan *false negative* (cacat tidak terdeteksi). Nilai *threshold* disimpan dalam file eksternal agar dapat digunakan kembali tanpa perlu melatih ulang model, sehingga meningkatkan efisiensi sistem (Zhong dkk., 2025).

2.2.8 Python

Python merupakan bahasa pemrograman tingkat tinggi yang banyak digunakan dalam pengembangan sistem berbasis kecerdasan buatan karena kemudahan sintaks, fleksibilitas, dan dukungan ekosistem pustaka yang luas seperti NumPy, OpenCV, dan Matplotlib. Dalam pengolahan citra dan *deep learning*, *Python* menjadi bahasa utama karena mampu menangani proses komputasi numerik dan visualisasi data dengan efisien.



Gambar 2. 3 Python

Pada gambar 2.3 menunjukkan logo *Python*, sebuah bahasa pemrograman tingkat tinggi yang dirancang dengan sintaks sederhana dan mudah dipahami, sehingga cocok digunakan oleh pemula maupun profesional. Bahasa ini bersifat *open source* dan dapat digunakan untuk berbagai keperluan, seperti pengembangan *web*, analisis data, kecerdasan buatan, hingga pengolahan citra. *Python* juga memiliki banyak pustaka atau *library* bawaan yang sangat membantu dalam proses pengembangan program. Karena fleksibel, mudah digunakan, dan didukung oleh komunitas yang besar, *Python* menjadi salah satu bahasa pemrograman yang paling

populer dan banyak digunakan di dunia saat ini (Maulida Surbakti dkk., 2024).

2.2.9 TensorFlow

TensorFlow adalah *framework* yang dirancang oleh Google untuk membangun dan melatih model *machine learning* dan *deep learning*. *Framework* ini mendukung eksekusi paralel, *GPU acceleration*, dan integrasi dengan pustaka tingkat tinggi seperti Keras, sehingga cocok untuk proyek-proyek yang memerlukan komputasi intensif seperti deteksi anomali citra.



Gambar 2. 4 *Framework TensorFlow*

Pada gambar 2.4 menunjukkan logo *TensorFlow*, dengan Keras API digunakan sebagai *framework* utama dalam sistem deteksi anomali dan klasifikasi botol plastik ini. *framework* ini menyediakan berbagai komponen seperti *Conv2D*, *MaxPooling2D*, *UpSampling2D*, dan *Dense* yang digunakan untuk membangun arsitektur *U-Net Autoencoder* dan *CNN*. Selain itu, *TensorFlow* mendukung *preprocessing* citra, pelatihan model di *GPU/CPU*, serta memudahkan pengembangan sistem secara efisien dan fleksibel, sehingga sangat cocok untuk aplikasi deteksi cacat berbasis *deep learning* (Abadi dkk., 2021).

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

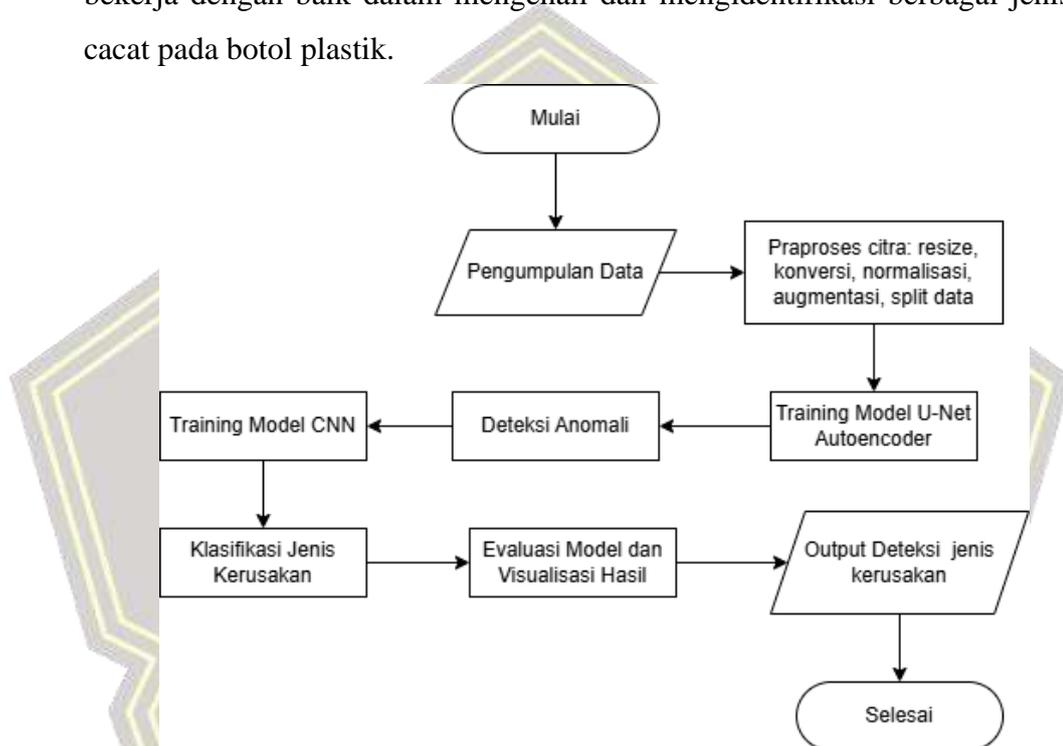
Penelitian ini menggunakan metode atau algoritma *Deep learning* berbasis arsitektur *U-Net Autoencoder* dan *Convolutional Neural Network* (CNN) dalam prosesnya. Penggunaan *U-Net Autoencoder* bertujuan untuk mendeteksi anomali pada citra botol plastik melalui proses rekonstruksi citra, sedangkan CNN digunakan untuk mengklasifikasikan jenis cacat yang terdeteksi. Penerapan kedua arsitektur ini dilakukan agar model dapat memperoleh akurasi yang baik dalam proses deteksi dan klasifikasi. Untuk mendukung proses pelatihan dan evaluasi, digunakan *framework TensorFlow* dan pustaka Keras. Penelitian ini melalui beberapa tahapan yang terstruktur untuk membangun sistem pendeteksi cacat botol plastik secara otomatis.

3.2 Studi Literatur

Teori yang dipelajari dalam melakukan penelitian ini adalah mengenai *Computer Vision*, *deep learning*, *Autoencoder*, *anomaly detection*, dan *Convolutional Neural Network* (CNN). Selain itu, literatur yang digunakan juga mencakup pembelajaran tentang struktur *U-Net* dan penerapannya dalam rekonstruksi citra, serta bagaimana CNN bekerja dalam klasifikasi gambar industri. Studi literatur dilakukan dengan meninjau berbagai sumber seperti artikel ilmiah, jurnal, dan situs *web* akademik terpercaya dari Google Scholar. Tujuan dari tinjauan ini adalah untuk memperoleh pemahaman yang kuat tentang metode dan arsitektur model yang relevan sehingga dapat diterapkan secara optimal pada sistem deteksi dan klasifikasi cacat botol plastik.

3.3 Alur Perancangan Sistem

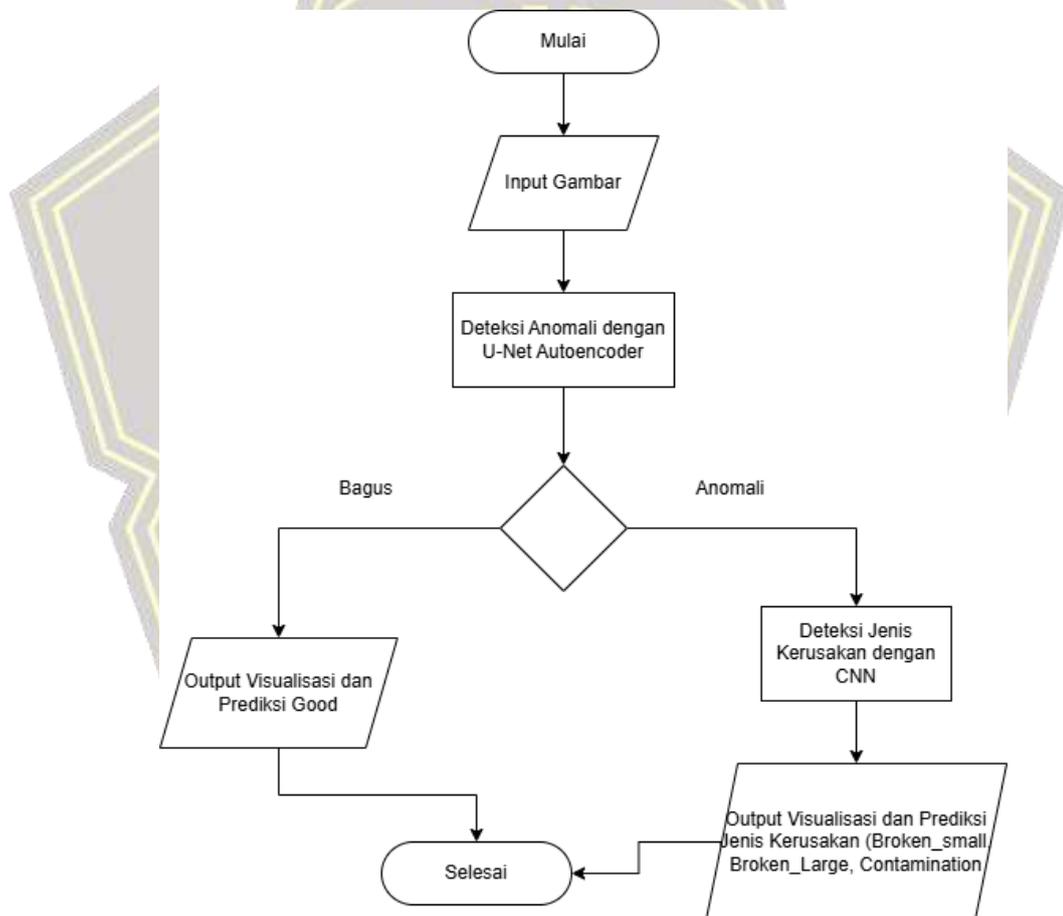
Untuk membangun sistem deteksi dan klasifikasi cacat botol plastik, diperlukan alur perancangan yang terstruktur. Gambar 3.1 berikut menunjukkan tahapan utama dalam pengembangan sistem berbasis deep learning yang terdiri dari dua tahap, yaitu deteksi anomali dan klasifikasi jenis cacat. Setiap tahapan dirancang untuk memastikan model dapat bekerja dengan baik dalam mengenali dan mengidentifikasi berbagai jenis cacat pada botol plastik.



Gambar 3.1 Flowchart alur perancangan sistem

Pada gambar 3.1 merupakan *flowchart* proses perancangan alur sistem deteksi dan klasifikasi cacat pada botol plastik menggunakan pendekatan dua tahap berbasis *deep learning*. Proses diawali dengan pengumpulan *dataset* citra botol plastik yang kemudian dilakukan tahap pra-pemrosesan, seperti pengubahan ukuran, konversi ke *grayscale*, normalisasi, dan augmentasi data. Setelah itu, data dibagi menjadi tiga subset, yaitu data latih, data validasi, dan data uji untuk memastikan proses pelatihan model berlangsung optimal serta menghindari *overfitting*. Tahap selanjutnya adalah pelatihan model deteksi anomali menggunakan arsitektur *U-Net Autoencoder*. Model ini bertugas membedakan antara citra normal dan citra

yang mengandung cacat berdasarkan selisih hasil rekonstruksi. Citra yang terdeteksi mengandung anomali kemudian masuk ke tahap klasifikasi jenis cacat menggunakan model CNN *classifier*. Model klasifikasi ini dilatih untuk membedakan beberapa jenis kerusakan, seperti *broken large*, *broken small*, dan *contamination*. Hasil dari kedua model dievaluasi menggunakan metrik performa seperti akurasi, *f1-score*, serta divisualisasikan melalui *Confusion matrix* dan grafik evaluasi. Diagram ini menggambarkan alur kerja sistem secara sistematis dari awal pengolahan data hingga menghasilkan *output* berupa informasi jenis cacat yang terdeteksi pada botol plastik.



Gambar 3. 2 *Flowchart* alur kerja sistem

Pada gambar 3.2 merupakan *flowchart* alur kerja sistem deteksi anomali pada botol plastik menggunakan *U-Net Autoencoder* dan CNN.

Proses dimulai dari *input* gambar botol plastik yang kemudian dianalisis menggunakan model *U-Net Autoencoder* untuk mendeteksi adanya anomali. Jika gambar terdeteksi normal, sistem langsung menghasilkan *output* visualisasi dan prediksi bahwa botol dalam kondisi bagus. Namun jika terdeteksi anomali, gambar akan diproses lebih lanjut oleh model CNN untuk mengklasifikasikan jenis kerusakan menjadi salah satu dari tiga kategori: *Broken small*, *Broken large*, atau *Contamination*. Hasil klasifikasi ini divisualisasikan sebagai *output* akhir sistem.

3.4 Pengumpulan Dataset

Pada penelitian ini, digunakan *Dataset* berupa citra botol plastik dengan total sebanyak 79 gambar. *Dataset* tersebut dikategorikan ke dalam empat kelas, yaitu *Broken small*, *Broken large*, *Contamination*, dan *Good*. Pengambilan *dataset* merujuk pada referensi dari *dataset* MVTEC AD yang dapat diakses melalui tautan:

<https://www.mvtec.com/company/research/Datasets/mvtec-ad>.

Berikut ini merupakan contoh gambar dari masing-masing kategori yang digunakan dalam penelitian:

1. Rusak kecil



Gambar 3. 3 Botol Rusak Kecil

Pada gambar 3.3 merupakan gambar botol rusak kecil (*broken small*). Botol plastik dalam kategori ini memiliki kerusakan ringan yang

mungkin tidak terlalu terlihat secara kasat mata tetapi tetap memengaruhi kualitas produk. Contoh kerusakan kecil mencakup goresan halus, cacat permukaan kecil, atau deformasi ringan yang tidak terlalu mencolok namun tetap dianggap tidak memenuhi standar kualitas.

2. Rusak besar



Gambar 3. 4 Botol Rusak Besar

Pada gambar 3.4 merupakan gambar botol rusak besar (*broken large*). Kategori ini mencakup botol plastik dengan kerusakan yang lebih signifikan dan mudah dikenali. Kerusakan besar dapat berupa retakan, lubang, atau deformasi yang mencolok, seperti bagian botol yang penyok atau bentuknya tidak sesuai standar.

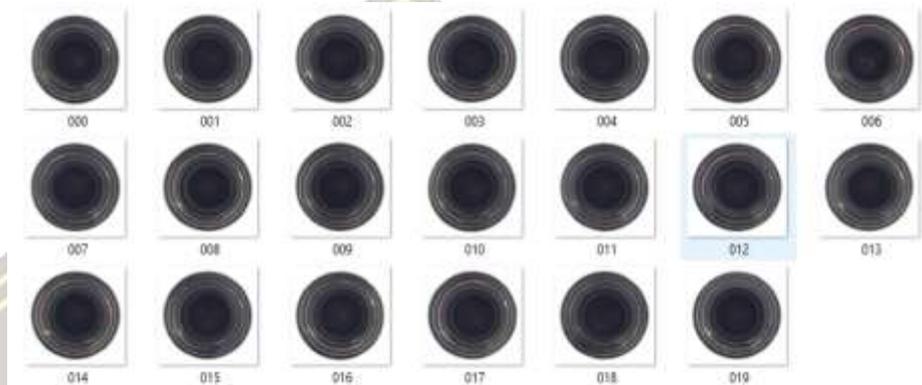
3. Terkontaminasi



Gambar 3. 5 Botol Terkontaminasi

Pada gambar 3.5 merupakan gambar botol yang terkontaminasi (*Contamination*). Botol dalam kategori ini menunjukkan adanya kontaminasi seperti noda minyak, debu, atau material asing yang menempel pada permukaan botol. Kontaminasi tersebut dapat berasal dari proses produksi, penyimpanan, atau distribusi, yang berpotensi membahayakan kualitas dan keamanan produk.

4. Bagus



Gambar 3. 6 Botol Bagus

Pada gambar 3.6 merupakan gambar botol bagus (*good*). Kategori ini terdiri dari botol plastik yang memenuhi standar kualitas dengan kondisi sempurna, tanpa kerusakan maupun kontaminasi. Botol dalam kategori ini memiliki bentuk, permukaan, dan kebersihan yang sesuai untuk didistribusikan ke pasar. Gambar dalam kategori ini digunakan sebagai data acuan untuk mendeteksi anomali pada botol yang rusak.

3.5 Preprocessing

Tahapan praproses citra merupakan langkah penting dalam *pipeline* deteksi anomali dan klasifikasi jenis cacat botol, karena bertujuan untuk mempersiapkan data mentah agar dapat digunakan secara efektif oleh model *deep learning*. Dalam sistem ini, praproses mencakup beberapa komponen utama, yaitu:

1. Konversi dan Standarisasi Ukuran Gambar

Tahap pertama dalam proses praproses data adalah melakukan konversi format dan standarisasi ukuran gambar. Seluruh citra yang diambil dari *Dataset* awal memiliki ukuran dan format yang bervariasi,

sehingga perlu diubah agar konsisten. Pada tahap ini, setiap gambar dikonversi ke dalam format skala keabuan (*grayscale*) menggunakan parameter `color_mode='grayscale'`. Konversi ini bertujuan untuk menyederhanakan informasi visual dari gambar, mengurangi jumlah channel warna dari tiga (RGB) menjadi satu, sehingga mempercepat proses pelatihan model tanpa kehilangan informasi penting yang berkaitan dengan bentuk dan pola cacat. Setelah itu, setiap gambar diubah ukurannya menjadi 128x128 piksel, sehingga seluruh gambar memiliki dimensi *input* yang seragam. Ukuran ini dipilih karena cukup kecil untuk efisiensi komputasi, namun masih cukup besar untuk menangkap detail penting dari cacat pada permukaan botol.

2. Normalisasi Nilai Piksel

Setelah gambar dikonversi ke dalam format *grayscale* dan disesuaikan ukurannya, langkah berikutnya adalah normalisasi nilai piksel. Normalisasi dilakukan dengan mengubah rentang nilai piksel dari 0 hingga 255 menjadi 0 hingga 1. Proses ini dilakukan dengan membagi seluruh nilai piksel dengan angka 255 menggunakan `astype('float32') / 255`. Tujuan dari normalisasi ini adalah agar data memiliki distribusi nilai yang seragam dan berskala kecil. Dengan begitu, model *deep learning* yang digunakan menjadi lebih stabil saat dilatih dan proses konvergensi menjadi lebih cepat. Normalisasi juga membantu mencegah nilai piksel besar mendominasi proses pembelajaran sehingga meningkatkan kinerja dan akurasi model.

3. Augmentasi Data Cacat (*Defect Images*)

Dataset citra botol cacat (*defect images*) umumnya memiliki jumlah yang jauh lebih sedikit dibandingkan dengan gambar botol normal. Oleh karena itu, dilakukan augmentasi data untuk meningkatkan jumlah dan keragaman gambar cacat. Augmentasi ini dilakukan menggunakan teknik *Image Data Generator*, yang menghasilkan variasi baru dari gambar asli dengan melakukan transformasi acak seperti rotasi sebesar 20 derajat, pergeseran

horizontal dan *vertikal* sebesar 10%, *zoom* hingga 20%, *flipping horizontal* (membalik gambar), serta variasi kecerahan dalam kisaran 0.8 hingga 1.2. Dengan augmentasi ini, model dapat belajar mengenali pola cacat dalam berbagai kondisi visual dan orientasi, sehingga meningkatkan generalisasi model terhadap data yang belum pernah dilihat.

4. Penambahan *Noise* (*Gaussian salt and pepper*)

Untuk membuat model lebih tangguh terhadap gangguan atau *noise visual* yang sering terjadi dalam pengambilan gambar di dunia nyata, dilakukan penambahan *noise* buatan pada citra. Dua jenis *noise* digunakan dalam tahap ini, yaitu *Gaussian noise* dan *salt-and-pepper noise*. *Gaussian noise* ditambahkan dengan menghasilkan distribusi acak yang mengikuti pola normal dengan *mean* nol dan standar deviasi 0.1, kemudian dijumlahkan dengan nilai piksel asli gambar. Ini merepresentasikan *noise* alami yang biasanya muncul dari sensor kamera. Sementara itu, *salt-and-pepper noise* dibuat dengan mengacak sebagian kecil piksel dalam gambar agar bernilai hitam (0) atau putih (1), yang menyerupai gangguan digital akibat kerusakan data atau sinyal. Penambahan *noise* ini membuat model mampu mengenali cacat meskipun terdapat gangguan visual yang tidak ideal pada gambar.

5. Penggabungan Gambar Augmentasi

Langkah akhir dalam praproses adalah menggabungkan data gambar asli dengan hasil augmentasi dan penambahan *noise*. Data hasil augmentasi dari gambar cacat digabungkan kembali dengan data aslinya, menghasilkan *dataset* cacat yang lebih besar dan beragam. Setiap kelas cacat seperti *broken large*, *broken small*, dan *contamination* mengalami proses ini secara terpisah, lalu seluruh data dikombinasikan menjadi satu *set* data pelatihan untuk model klasifikasi. Proses ini penting untuk mengurangi ketimpangan jumlah data antar kelas dan memastikan bahwa model CNN *classifier* dapat belajar secara seimbang dari ketiga jenis cacat. Dengan *dataset* yang lebih beragam

dan seimbang, model memiliki kemampuan generalisasi yang lebih baik saat menghadapi data cacat baru.

3.6 Pembagian *Dataset*

Untuk mendukung proses pelatihan model deteksi dan klasifikasi, *dataset* yang digunakan dalam penelitian ini dibagi berdasarkan kebutuhan masing-masing tahap. Pembagian ini mencakup dua pendekatan, yaitu untuk deteksi anomali menggunakan *U-Net Autoencoder* dan untuk klasifikasi jenis cacat menggunakan *CNN classifier*. Rincian pembagian *dataset* disajikan pada Tabel 3.1 dan Tabel 3.2 berikut:

Tabel 3. 1 Pembagian *Dataset* CNN

Split	Jumlah Gambar	Presentase
<i>Train</i>	166	57,9%
<i>Validation</i>	42	14,6%
<i>Test (good+bad)</i>	79 (19+60)	27,5%
Total	287	100%

Tabel 3. 2 Pembagian *Dataset* CNN

Kelas	Jumlah Gambar	Presentase
<i>Train</i>	1612	80%
<i>Validation</i>	404	20%
Total	2016	100%

Pada Tabel 3.1 adalah pembagian *dataset* yang dilakukan dengan dua pendekatan, yaitu untuk proses deteksi anomali menggunakan *U-Net Autoencoder* dan untuk proses klasifikasi jenis cacat menggunakan *CNN classifier*. *Dataset* utama terdiri dari gambar botol plastik yang dikelompokkan menjadi dua kategori utama, yaitu botol normal (*good*) dan botol cacat (*bad*).

Untuk pendekatan deteksi anomali dengan *U-Net*, model hanya dilatih menggunakan gambar botol normal agar mampu merekonstruksi gambar tersebut dengan akurat. Gambar cacat tidak disertakan dalam proses

pelatihan, melainkan hanya digunakan pada tahap pengujian untuk mengukur kemampuan model dalam mendeteksi anomali melalui *error* rekonstruksi. Pembagian data dilakukan menjadi data latih (57,9%), data validasi (14,6%), dan data uji (27,5%). Data uji terdiri dari 19 gambar normal dan 60 gambar cacat, sehingga total keseluruhan adalah 287 gambar.

Sementara itu, Tabel 3.2 menjelaskan pembagian *dataset* khusus untuk pelatihan model klasifikasi CNN. *Dataset* ini hanya terdiri dari gambar botol cacat yang terbagi ke dalam tiga kelas utama, yaitu *broken large*, *broken small*, dan *contamination*. Setiap gambar cacat telah melalui proses augmentasi sebanyak lima kali untuk memperbesar jumlah data dan meningkatkan kemampuan generalisasi model. Jumlah total gambar setelah augmentasi adalah 2016 gambar, yang kemudian dibagi menjadi data latih sebanyak 1612 gambar (80%) dan data validasi sebanyak 404 gambar (20%).

3.7 Library yang di gunakan

Tabel 3. 3 *Library* yang digunakan untuk Sistem Deteksi

Kategori	Library	Fungsi
Deep Learning Framework	tensorflow	Framework utama untuk membangun dan melatih model <i>deep learning</i> .
	keras	API tingkat tinggi di atas TensorFlow untuk mempermudah pembuatan model <i>neural network</i> .
Numerik & Matriks	numpy	Operasi <i>array</i> , manipulasi data numerik.
Visualisasi	matplotlib	Untuk <i>plotting</i> visualisasi <i>loss</i> , hasil rekonstruksi, <i>confusion matrix</i> , dll.
	seaborn	Visualisasi <i>heatmap</i> atau evaluasi hasil yang lebih estetik.
Pengolahan Citra	opencv-python (cv2)	Membaca, memotong, <i>resize</i> , dan manipulasi gambar.
	Pillow (PIL)	Alternatif untuk memproses gambar secara

		sederhana.
<i>Preprocessing & Utilities</i>	sklearn (scikit-learn)	Normalisasi data, splitting data, evaluasi metrik (<i>accuracy, recall, precision, confusion matrix</i>).
	os, glob	Untuk membaca file dari direktori secara terstruktur.
Evaluasi Anomali	scikit-learn.metrics	Perhitungan metrik seperti MSE, ROC AUC, F1-Score, dll.
GPU Acceleration (<i>opsional</i>)	tensorflow-gpu	Jika menggunakan GPU untuk mempercepat pelatihan model.

3.8 Parameter Model

Sistem ini menggunakan dua model pembelajaran mendalam utama, yaitu *U-Net Autoencoder* dan *Convolutional Neural Network (CNN)*, yang masing-masing memiliki parameter dan arsitektur khusus sesuai fungsinya. Model *U-Net Autoencoder* mengusung arsitektur *encoder-decoder* simetris dengan empat blok konvolusi di bagian *encoder*. Setiap blok terdiri dari dua lapisan Conv2D yang menggunakan kernel 3×3 dan fungsi aktivasi ReLU untuk menangkap fitur spasial yang kompleks. Penggunaan *MaxPooling2D* dengan ukuran 2×2 pada setiap blok bertujuan menurunkan dimensi spasial sekaligus mempertahankan informasi penting agar jaringan dapat fokus pada pola-pola signifikan. Pada bagian decoder, model menggunakan empat blok *UpSampling2D* yang bertugas mengembalikan dimensi gambar ke ukuran awal, dengan *skip connection* yang menghubungkan bagian *encoder* dan *decoder* untuk meminimalkan kehilangan detail selama rekonstruksi. Lapisan akhir model berupa Conv2D dengan kernel 1×1 dan aktivasi *sigmoid* menghasilkan citra rekonstruksi dengan piksel bernilai antara 0 sampai 1. *Optimizer* Adam dipilih untuk mempercepat dan menstabilkan proses pelatihan, sementara fungsi *loss* yang digunakan adalah *Mean Squared Error (MSE)* untuk mengukur kesalahan antara citra input dan rekonstruksi. Model dievaluasi berdasarkan nilai rata-rata MSE pada citra

normal sebagai indikasi seberapa baik model dapat merekonstruksi data tanpa cacat.

Model CNN *classifier* memiliki arsitektur yang lebih sederhana namun efektif untuk tugas klasifikasi tiga kelas cacat, yaitu *broken large*, *broken small*, dan *contamination*. Terdiri dari tiga blok Conv2D dengan filter masing-masing 32, 64, dan 128, model ini menggunakan kernel yang sama berukuran 3×3 dan fungsi aktivasi ReLU untuk mengekstrak fitur visual dari citra cacat. MaxPooling2D digunakan setelah setiap blok konvolusi untuk mengurangi dimensi spasial dan mengurangi kompleksitas komputasi. Setelah tahap ekstraksi fitur, lapisan Flatten mengubah data fitur menjadi bentuk vektor satu dimensi yang selanjutnya diolah oleh Dense layer berisi 128 *unit* dengan aktivasi ReLU. Dropout sebesar 0.5 diterapkan untuk mencegah *overfitting* dengan menonaktifkan sebagian neuron secara acak selama pelatihan. *Output* model menggunakan *Dense layer* dengan tiga unit dan fungsi aktivasi softmax untuk menghasilkan probabilitas klasifikasi ke masing-masing kelas. *Optimizer Adam* dan fungsi *loss Sparse Categorical Crossentropy* dipakai untuk mengoptimalkan pelatihan dengan target label yang berupa *integer* kelas, dan performa model diukur dengan metrik akurasi.

3.9 Pelatihan Model

Pelatihan model dilakukan secara terpisah untuk setiap model sesuai dengan sifat data dan metode pembelajaran yang digunakan. Model *U-Net Autoencoder* dilatih secara tidak terawasi (*unsupervised*) hanya dengan data citra botol plastik tanpa cacat dari folder *Train/good*. Tujuannya adalah agar model dapat mempelajari representasi distribusi visual botol normal sehingga ketika diberi citra cacat, hasil rekonstruksi akan menunjukkan perbedaan yang signifikan. Proses pelatihan berlangsung selama 15 *epoch* dengan batch size 4, dan dilakukan validasi sebesar 20% untuk memantau kemampuan generalisasi model selama pelatihan. Setelah pelatihan, model diuji pada data uji yang terdiri dari citra normal dan cacat. Nilai *Mean Squared Error* (MSE) dihitung antara citra asli dan hasil rekonstruksi.

Untuk menentukan ambang batas (*threshold*) yang membedakan citra normal dan cacat, digunakan nilai persentil ke-95 dari distribusi *error* rekonstruksi citra normal. Artinya, *threshold* ini merepresentasikan nilai di mana 95% citra normal berada di bawahnya, dan citra dengan error melebihi *threshold* dianggap mengandung anomali. Penggunaan *threshold* berbasis persentil ini membantu sistem menghindari deteksi *false positive* yang berlebihan, sekaligus menjaga sensitivitas terhadap cacat nyata. Citra yang terdeteksi anomali kemudian diteruskan ke model CNN *classifier* untuk klasifikasi jenis cacat, sedangkan citra yang berada di bawah *threshold* dianggap normal dan tidak diproses lebih lanjut.

Sementara itu, model CNN *classifier* dilatih secara terawasi (*supervised*) menggunakan data citra cacat yang telah berlabel dari folder *Train_cnn*. *Dataset* ini dibagi menjadi data latih dan validasi dengan perbandingan 80:20 menggunakan metode stratifikasi agar distribusi kelas tetap seimbang. Agar model mampu mengenali variasi cacat yang lebih luas dan tidak mudah *overfitting*, dilakukan augmentasi data berupa *rotation*, *zoom*, translasi, *flip horizontal*, pengaturan kecerahan, serta penambahan *noise Gaussian* dan *salt & pepper*. Proses pelatihan dilakukan selama 50 *epoch* dengan *batch size* 4. Performa model dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*. Selain itu, *Confusion matrix* digunakan untuk menampilkan performa klasifikasi pada tiap kelas secara rinci, sehingga memudahkan analisis kesalahan dan peningkatan model. Dengan demikian, pelatihan ini bertujuan menghasilkan model klasifikasi yang *robust* dan akurat dalam mengenali jenis cacat botol plastik.

3.10 Evaluasi model

Setelah proses deteksi anomali dilakukan, hasilnya dievaluasi menggunakan berbagai metrik evaluasi untuk mengukur kinerja model secara menyeluruh. Metrik yang digunakan antara lain:

1. *Accuracy*: Mengukur tingkat ketepatan model dalam mengklasifikasikan seluruh data dengan benar (baik data normal maupun anomali).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2. *Precision*: Mengukur proporsi deteksi yang diklasifikasikan sebagai anomali yang benar-benar anomali (menghindari *false positive*).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3. *Recall (Sensitivity)*: Mengukur kemampuan model dalam mendeteksi seluruh data anomali yang ada (menghindari *false negative*).

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4. *F1-Score*: Rata-rata harmonik dari *precision* dan *recall*, berguna ketika ada ketidakseimbangan antara data normal dan anomali.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Selain metrik-metrik di atas, dilakukan pula:

1. Analisis *ROC Curve (Receiver Operating Characteristic)*: Untuk mengevaluasi *trade-off* antara *true positive rate* (TPR) dan *false positive rate* (FPR) dalam berbagai *threshold*.
2. *AUC (Area Under Curve)*: Luas di bawah kurva ROC yang mencerminkan kemampuan model dalam membedakan antara kelas normal dan anomali.
3. *Confusion Matrix*: Digunakan untuk melihat distribusi hasil prediksi antara kelas normal dan anomali (TP, TN, FP, FN) secara rinci.

Hasil deteksi divisualisasikan untuk mempermudah pemahaman, memungkinkan identifikasi titik-titik anomali, serta memberikan gambaran yang lebih jelas terhadap performa model dalam konteks nyata.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil dan Analisis

4.1.1 Pengumpulan data

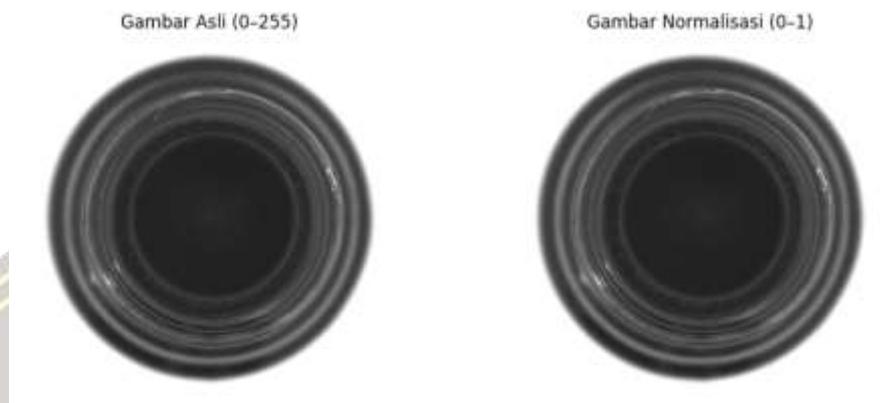
Pada penelitian ini, *dataset* yang digunakan terdiri dari gambar botol plastik yang dibagi menjadi dua bagian utama, yaitu data *Train* sebanyak 208 gambar botol yang dikategorikan sebagai botol normal (*good*), dan data test sebanyak 79 gambar yang terdiri dari empat kelas, yaitu *broken small* sebanyak 21 gambar, *broken large* sebanyak 19 gambar, *contamination* sebanyak 20 gambar, dan *good* sebanyak 19 gambar. Data *Train* digunakan untuk melatih model *U-Net Autoencoder* agar dapat mengenali pola dari botol yang tidak cacat dan mendeteksi penyimpangan sebagai anomali. Sementara itu, data test digunakan untuk menguji kemampuan model dalam mendeteksi anomali dan mengklasifikasikan jenis cacat menggunakan *CNN classifier*. Pendekatan ini memungkinkan sistem untuk tidak hanya mendeteksi adanya cacat pada botol, tetapi juga menentukan jenis cacat yang terjadi.

4.1.2 Preprocessing

Sebelum gambar digunakan dalam pelatihan model, dilakukan tahap preprocessing untuk menyederhanakan informasi visual, menyeragamkan dimensi, dan meningkatkan konsistensi data. Proses ini mencakup beberapa langkah utama yang ditampilkan pada Gambar 4.1 hingga Gambar 4.3.



Gambar 4. 1 Konversi gambar ke *grayscale*

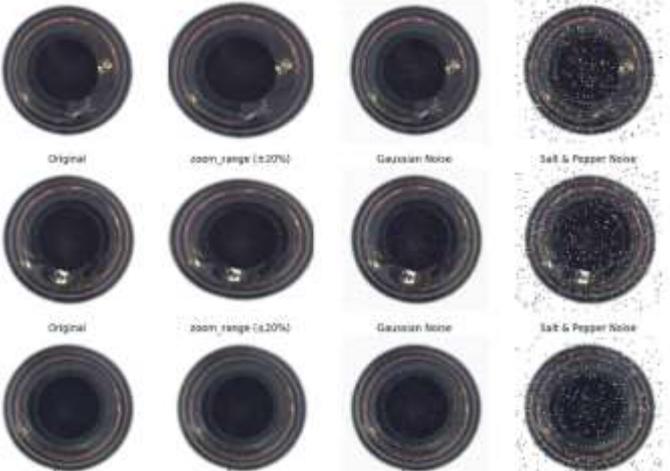
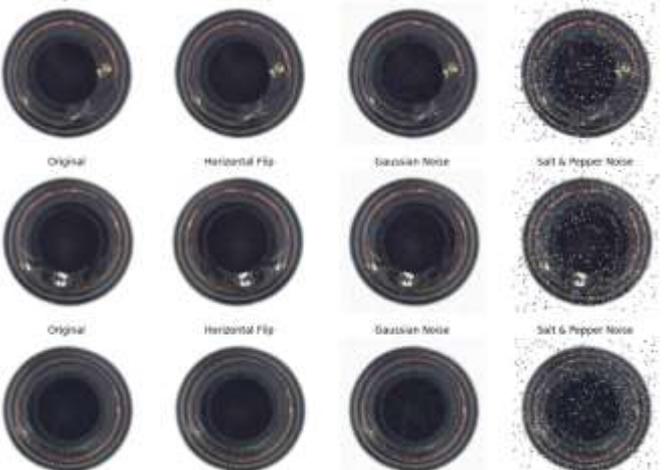
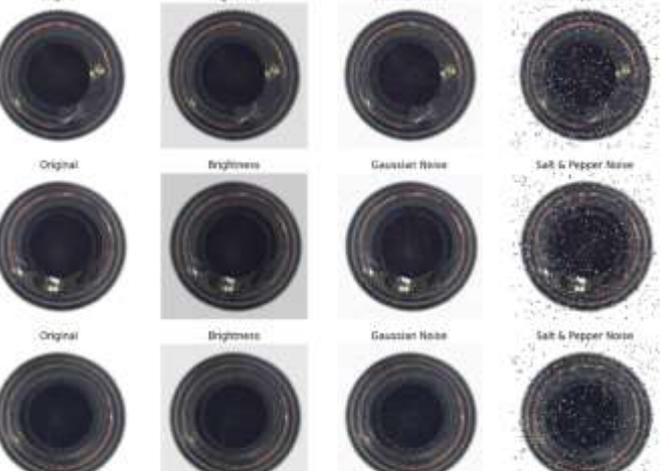
Gambar 4. 2 *Resize* gambar

Gambar 4. 3 Normalisasi piksel

Pada Gambar 4.1 menunjukkan proses konversi gambar dari format RGB ke *grayscale* untuk mengurangi kompleksitas data. Gambar 4.2 memperlihatkan proses perubahan ukuran gambar menjadi dimensi yang seragam agar sesuai dengan *input* model. Selanjutnya, Gambar 4.3 menggambarkan tahap normalisasi, di mana nilai piksel dikonversi ke skala 0–1 agar mempercepat dan menstabilkan proses pelatihan. Setelah tahap awal *preprocessing*, augmentasi data diterapkan untuk menambah variasi gambar dan meningkatkan kemampuan generalisasi model. Tabel 4.1 menyajikan berbagai kombinasi teknik augmentasi yang digunakan

Tabel 4. 1 Augmentasi gambar

1.	<i>Rotated + Gaussian Noise + salt & pepper noise</i>	Original	Rotation $\pm 20^\circ$	Gaussian Noise	Salt & Pepper Noise
					
		Original	Rotation $\pm 20^\circ$	Gaussian Noise	Salt & Pepper Noise
					
		Original	Rotation $\pm 20^\circ$	Gaussian Noise	Salt & Pepper Noise
					
2.	<i>Width_shift + Gaussian Noise + salt & pepper noise</i>	Original	width_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					
		Original	width_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					
		Original	width_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					
3.	<i>Height_shift + Gaussian Noise + salt & pepper noise</i>	Original	height_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					
		Original	height_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					
		Original	height_shift_range ($\pm 10\%$)	Gaussian Noise	Salt & Pepper Noise
					

4.	Zoom + Gaussian Noise + salt & pepper noise	 <p>The grid shows three rows of images. Each row contains four images: 'Original', 'zoom_range (±20%)', 'Gaussian Noise', and 'Salt & Pepper Noise'. The images show a coin being zoomed in and out, and then having different types of noise added to it.</p>
5.	Horizontal flip + Gaussian Noise + salt & pepper noise	 <p>The grid shows three rows of images. Each row contains four images: 'Original', 'Horizontal Flip', 'Gaussian Noise', and 'Salt & Pepper Noise'. The images show a coin being flipped horizontally and then having different types of noise added to it.</p>
6.	Brighness [0.8, 1.2] + Gaussian Noise + salt & pepper noise	 <p>The grid shows three rows of images. Each row contains four images: 'Original', 'Brighness', 'Gaussian Noise', and 'Salt & Pepper Noise'. The images show a coin with its brightness adjusted and then having different types of noise added to it.</p>

Pada tahap *preprocessing*, gambar pertama-tama dikonversi ke format grayscale seperti yang ditunjukkan pada Gambar 4.1. Proses ini bertujuan untuk mengurangi kompleksitas data dengan menghilangkan informasi

warna, sehingga fokus hanya pada intensitas piksel yang lebih relevan untuk mendeteksi cacat pada botol plastik. Selanjutnya, gambar diubah ukurannya (*resize*) menjadi dimensi yang konsisten seperti pada Gambar 4.2. *Resize* ini penting agar seluruh gambar memiliki ukuran yang sama sehingga dapat diproses secara efisien oleh model, sekaligus mempertahankan proporsi visual yang cukup untuk analisis. Setelah itu, dilakukan normalisasi piksel (Gambar 4.3) dengan mengubah nilai piksel ke rentang antara 0 hingga 1. Normalisasi ini membantu mempercepat konvergensi pelatihan model dan meningkatkan stabilitas numerik selama proses pembelajaran.

Selain itu, untuk memperkaya data dan membuat model lebih tahan terhadap variasi nyata pada kondisi pengambilan gambar, dilakukan augmentasi gambar seperti yang tertera pada Tabel 4.1. Berbagai teknik augmentasi diterapkan, termasuk rotasi (*rotation*), pergeseran lebar (*width shift*), pergeseran tinggi (*height shift*), *zoom*, dan pembalikan *horizontal* (*horizontal flip*). Setiap transformasi ini dikombinasikan dengan penambahan *noise* berupa *Gaussian noise* dan *salt & pepper noise*, serta penyesuaian kecerahan (*brighness*) dalam rentang 0.8 hingga 1.2. Kombinasi augmentasi ini bertujuan untuk mensimulasikan variasi nyata dalam data sehingga model dapat belajar mengenali cacat pada botol plastik dalam berbagai kondisi pencahayaan, posisi, dan gangguan visual yang mungkin terjadi. Dengan demikian, *preprocessing* ini tidak hanya menyiapkan data agar sesuai dengan kebutuhan model, tetapi juga meningkatkan kualitas dan keragaman data pelatihan sehingga model dapat bekerja lebih efektif dan akurat.

4.1.3 Training Model

Pada tahap ini, dilakukan proses pelatihan model *U-Net Autoencoder* yang berfungsi untuk mendeteksi cacat pada botol plastik. Proses *Training* dimulai dengan kompilasi model, di mana dipilih *optimizer* dan fungsi *loss* yang sesuai untuk mengoptimalkan performa model. Setelah itu, model dilatih menggunakan *dataset* citra botol plastik dengan parameter *Training*

yang sudah ditentukan. Selain itu, untuk menghindari *overfitting* dan memantau performa model secara berkala, sebagian data disisihkan sebagai data validasi. Proses pelatihan ini bertujuan untuk meminimalkan kesalahan rekonstruksi pada citra input sehingga model dapat mengenali pola cacat secara efektif.

```
# === Train Autoencoder ===
UNET_AUTOENCODER = build_unet_autoencoder()
HISTORY = UNET_AUTOENCODER.fit(train_images, train_images, epochs=15, batch_size=4, validation_split=0.2, shuffle=True)

# === Simpan model Autoencoder ===
MODEL_PATH = "/content/drive/MyDrive/bottle_models/"
os.makedirs(MODEL_PATH, exist_ok=True)
UNET_AUTOENCODER.save(os.path.join(MODEL_PATH, "unet_autoencoder_model.keras"))
```

Gambar 4. 4 Optimizer dan parameter Training

Pada gambar 4.4 menunjukkan proses kompilasi, pelatihan, dan penyimpanan model *U-Net Autoencoder* untuk mendeteksi cacat pada botol plastik. Pada bagian pertama, model *Autoencoder* dikompilasi menggunakan *optimizer Adam* dan fungsi *loss Mean Squared Error (MSE)*. *Optimizer Adam* dipilih karena kemampuannya dalam melakukan penyesuaian *learning rate* secara dinamis, sehingga dapat mempercepat proses konvergensi. Sementara itu, *MSE* digunakan untuk mengukur *error* rekonstruksi antara citra *input* dan hasil rekonstruksi citra oleh *Autoencoder*. Pada bagian kedua, model *Autoencoder* dilatih menggunakan *dataset* citra botol plastik (*Train_images*) dengan parameter *epochs=15*, *batch_size=4*, dan *validation_split=0.2*. Sebanyak 80% data digunakan untuk pelatihan dan 20% data digunakan untuk validasi. Data dilatih secara acak (*shuffle=True*) untuk menghindari *overfitting*. Pada bagian terakhir, model *Autoencoder* yang telah selesai dilatih disimpan ke Google Drive pada folder yang telah ditentukan (*/content/drive/MyDrive/bottle_models/*). Direktori tersebut dibuat menggunakan perintah *os.makedirs()* jika belum ada. Model disimpan dalam format *.keras* menggunakan metode *save()* untuk kemudahan akses saat proses inferensi atau pengujian model di masa mendatang.

```

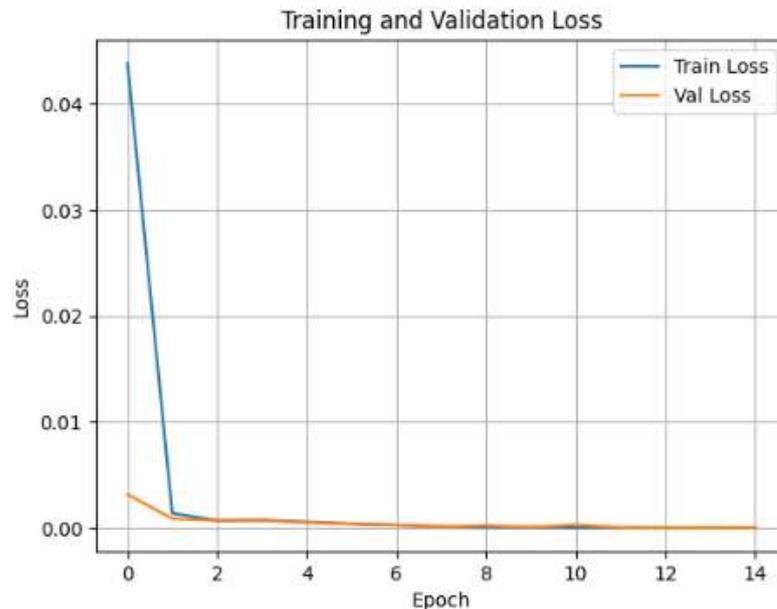
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history_cls = classifier.fit(anomaly_images, anomaly_labels, epochs=50, batch_size=8, validation_split=0.2)

# === Simpan model Classifier ===
classifier.save(os.path.join(MODEL_PATH, "classifier_model.keras"))

```

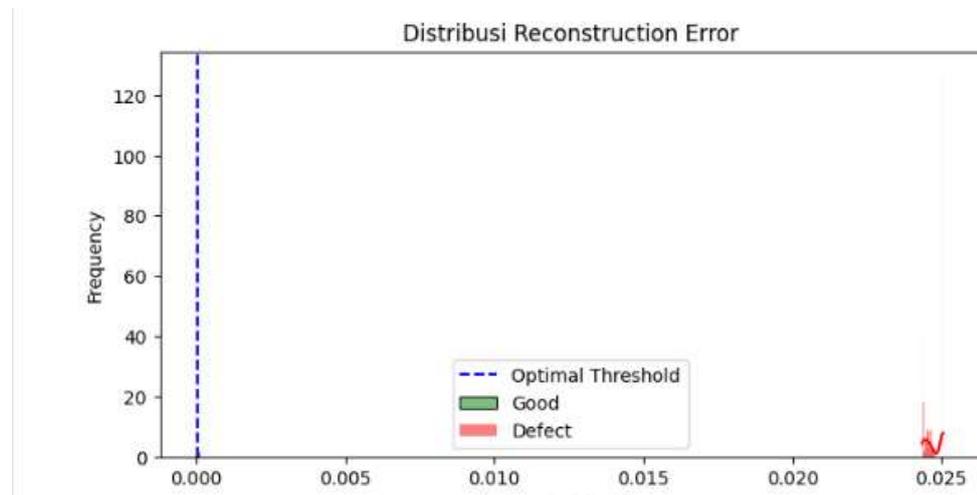
Gambar 4. 5 Optimizer parameter cnn classifier

Pada gambar 4.5 menunjukkan proses kompilasi, pelatihan, dan penyimpanan model klasifikasi untuk mendeteksi jenis cacat pada botol plastik. Pada tahap kompilasi, model dikonfigurasi menggunakan *optimizer Adam*, fungsi *loss Sparse Categorical Crossentropy*, dan metrik *Accuracy*. *Optimizer Adam* dipilih karena kemampuannya dalam melakukan penyesuaian *learning rate* secara adaptif, sehingga proses pelatihan menjadi lebih efisien. Fungsi *loss Sparse Categorical Crossentropy* digunakan karena data label pada *dataset* disajikan dalam format *integer*, bukan *one-hot encoding*. Sementara itu, metrik akurasi (*Accuracy*) dipilih untuk mengevaluasi performa klasifikasi model selama pelatihan. Pada tahap pelatihan, model dilatih menggunakan *dataset* citra anomali (*anomaly_images*) beserta labelnya (*anomaly_labels*). Parameter pelatihan yang digunakan adalah *epochs=50* dan *batch_size=4*, dengan 20% data digunakan sebagai data validasi (*validation_split=0.2*). Setelah proses pelatihan selesai, model klasifikasi disimpan dalam format keras di direktori Google Drive yang telah ditentukan (*MODEL_PATH*). Model tersebut akan digunakan pada tahap inferensi untuk mendeteksi jenis cacat pada botol plastik berdasarkan hasil rekonstruksi *Autoencoder*.



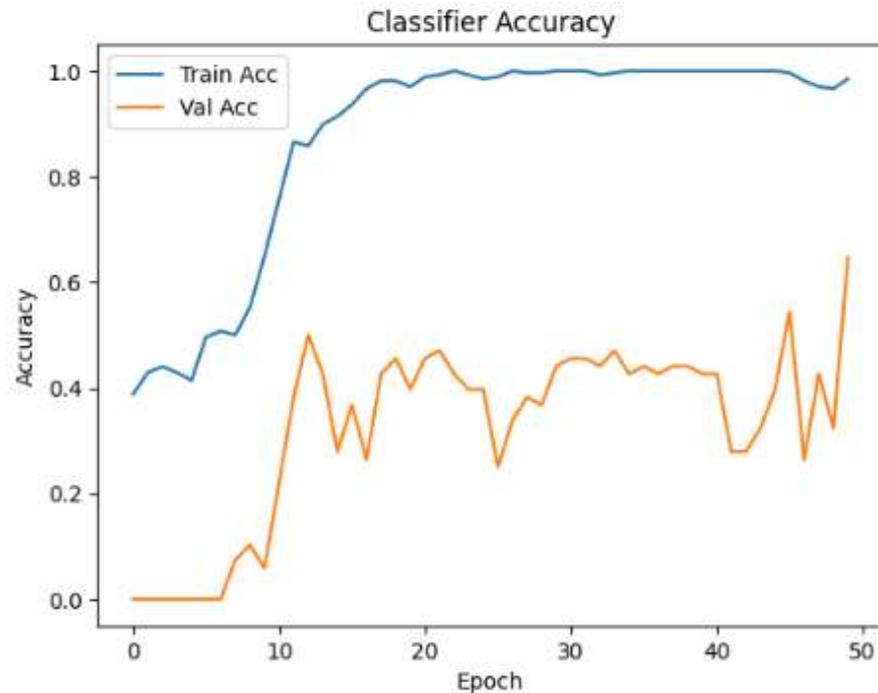
Gambar 4. 6 Grafik *Training and validation loss* pada *U-Net Autoencoder*

Pada gambar 4.6 menunjukkan grafik perbandingan antara nilai *loss* pada data pelatihan (*Train Loss*) dan data validasi (*Val Loss*) selama proses pelatihan model *U-Net Autoencoder*. Sumbu *horizontal* (*Epoch*) menunjukkan jumlah *epoch* pelatihan yang dilakukan, sedangkan sumbu *vertikal* (*Loss*) menunjukkan nilai *loss*. Pada awal pelatihan, nilai *loss* pada data pelatihan dan validasi relatif tinggi, namun terjadi penurunan yang signifikan pada *Epoch* pertama hingga kedua. Setelah itu, nilai *loss* baik pada data pelatihan maupun validasi cenderung stabil mendekati nol, menunjukkan bahwa model berhasil mempelajari pola dari data pelatihan dengan baik. Perbedaan antara *Train Loss* dan *Val Loss* juga relatif kecil, mengindikasikan bahwa model tidak mengalami *overfitting* dan dapat menghasilkan rekonstruksi citra yang akurat. Hasil ini menunjukkan bahwa model *U-Net Autoencoder* telah terlatih dengan baik untuk mendeteksi anomali pada botol plastik.



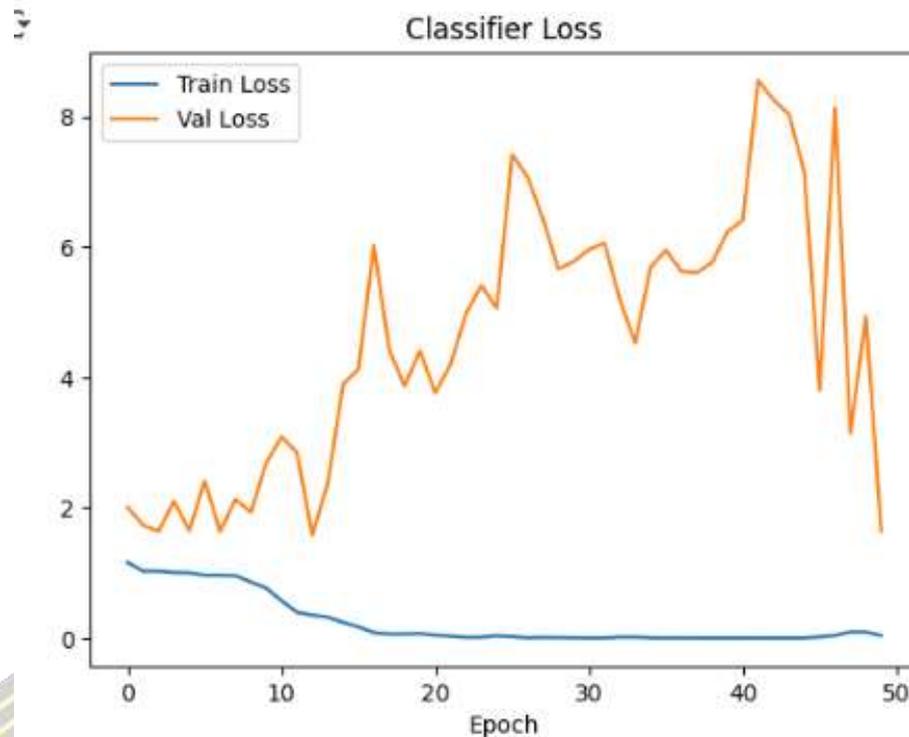
Gambar 4. 7 Distribusi rekonstruksi *error*

Pada gambar 4.7 menunjukkan distribusi rekonstruksi *error* pada model *U-Net Autoencoder* yang digunakan untuk mendeteksi cacat pada botol plastik. Sumbu *horizontal* (*Reconstruction Error*) menunjukkan nilai *error* yang dihitung sebagai selisih antara citra *input* dan citra hasil rekonstruksi oleh *Autoencoder*. Sumbu *vertikal* (*Density*) menunjukkan jumlah sampel pada masing-masing rentang nilai *error*. Pada grafik tersebut, garis putus-putus biru menunjukkan nilai ambang batas (*threshold*) yang ditentukan pada persentil ke-95 dari distribusi *error*. Area hijau (*Good*) merepresentasikan sampel dengan *error* di bawah ambang batas yang diklasifikasikan sebagai botol tanpa cacat. Sementara itu, area merah (*Defective*) menunjukkan sampel dengan *error* di atas ambang batas, yang diidentifikasi sebagai botol cacat. Grafik ini menunjukkan bahwa sebagian besar data memiliki *error* rekonstruksi yang sangat rendah (berkumpul di dekat nol), namun terdapat beberapa sampel dengan *error* tinggi yang mengindikasikan adanya cacat pada botol plastik tersebut. Dengan menetapkan *threshold* ini, model *Autoencoder* dapat secara efektif memisahkan botol cacat dari botol normal berdasarkan nilai rekonstruksi *error*.



Gambar 4. 8 *Classifier Accuracy*

Pada gambar 4.8 menunjukkan grafik akurasi selama proses pelatihan (*Train Acc*) dan validasi (*Val Acc*) model klasifikasi jenis cacat pada botol plastik. Sumbu *horizontal* (*Epoch*) menunjukkan jumlah *epoch* pelatihan, sedangkan sumbu vertikal (*Accuracy*) menunjukkan nilai akurasi. Grafik *Train Acc* (berwarna biru) memperlihatkan peningkatan akurasi yang signifikan dan mencapai nilai mendekati 1.0, menunjukkan bahwa model sangat baik dalam mengenali pola pada data pelatihan. Namun, grafik *Val Acc* (berwarna oranye) menunjukkan akurasi yang jauh lebih rendah dan tidak stabil, bahkan mengalami fluktuasi yang cukup tajam sepanjang proses pelatihan. Kondisi ini kembali mengindikasikan terjadinya *overfitting*, di mana model hanya belajar dari data pelatihan tanpa mampu melakukan generalisasi dengan baik terhadap data validasi. Untuk mengurangi *overfitting*, dapat dilakukan beberapa pendekatan seperti peningkatan variasi data melalui augmentasi, memperbesar jumlah data validasi, serta menerapkan regularisasi atau *drop out* pada arsitektur model.



Gambar 4.9 Classifier loss

Gambar 4.6 menunjukkan grafik *loss* dari proses pelatihan model klasifikasi jenis cacat botol plastik menggunakan CNN. Grafik ini membandingkan nilai *loss* pada data pelatihan (*Train Loss*) dan data validasi (*Val Loss*) selama 50 *epoch*. Terlihat bahwa nilai *Train Loss* mengalami penurunan secara konsisten, menandakan bahwa model belajar dengan baik pada data pelatihan. Namun, *Val Loss* menunjukkan fluktuasi yang signifikan dan cenderung meningkat seiring bertambahnya *epoch*. Pola ini mengindikasikan terjadinya *overfitting*, di mana model terlalu menyesuaikan diri terhadap data pelatihan dan kehilangan kemampuan generalisasi terhadap data yang belum pernah dilihat sebelumnya (data validasi). Untuk mengatasi hal ini, pendekatan seperti *early stopping*, regularisasi, atau penambahan data pelatihan dapat dipertimbangkan.

4.2 Hasil Evaluasi Model

Setelah proses pelatihan selesai, dilakukan evaluasi terhadap performa model baik pada tahap deteksi anomali maupun klasifikasi jenis cacat. Evaluasi ini menggunakan beberapa metrik standar seperti *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur kemampuan model dalam mengenali dan mengklasifikasikan cacat pada botol plastik.

Tabel 4. 2 Metrik evaluasi deteksi anomali

Metrix Evaluasi	Skor (%)
<i>Accuracy</i>	99.68%
<i>Precision</i>	99.66%
<i>Recall</i>	100%
<i>F-1 Score</i>	99.83%

Tabel 4.2 menunjukkan hasil evaluasi dari model deteksi anomali menggunakan *U-Net Autoencoder* dengan ambang batas (*threshold*) yang ditentukan pada persentil ke-95 dari data kategori *good*. Nilai *threshold* yang digunakan adalah 0.000049. Berdasarkan hasil evaluasi tersebut, diperoleh akurasi sebesar 0.9968, *precision* sebesar 0.9966, *recall* sebesar 1.0000, dan *f1-score* sebesar 0.9983. Nilai *recall* yang sempurna (1.0000) menandakan bahwa seluruh citra cacat berhasil dideteksi oleh model tanpa ada yang terlewat (*false negative*). Nilai *precision* yang tinggi menunjukkan bahwa hampir semua citra yang dideteksi sebagai cacat memang benar-benar cacat (minim *false positive*). Secara keseluruhan, metrik ini mencerminkan bahwa model deteksi anomali memiliki performa yang sangat baik dalam membedakan antara botol yang cacat dan tidak cacat.

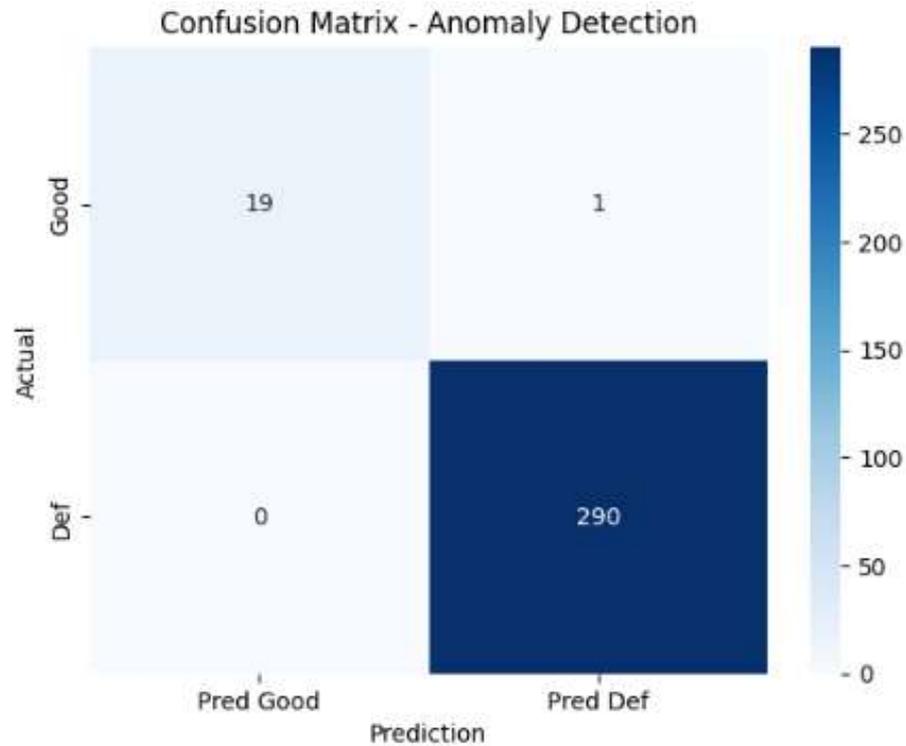
Tabel 4. 3 Metrik evaluasi klasifikasi jenis kerusakan

Metrix Evaluasi	Precision	Recall	F-1 Score
<i>Broken large</i>	88%	99%	93%
<i>Broken small</i>	93%	97%	95%
<i>Contamination</i>	96%	78%	86%
Accuracy	92%		

Tabel 4.3 menunjukkan hasil metrik evaluasi dari model klasifikasi jenis cacat botol plastik menggunakan CNN. Dari tabel tersebut, diperoleh nilai akurasi keseluruhan sebesar 92%, yang menunjukkan bahwa model mampu mengklasifikasikan data dengan cukup baik. Untuk kelas *broken large*, model memiliki *precision* sebesar 0.88 dan *recall* yang sangat tinggi yaitu 0.99, menghasilkan *f1-score* sebesar 0.93. Kelas *broken small* menunjukkan kinerja yang paling baik dengan *f1-score* tertinggi sebesar 0.95, didukung oleh *precision* 0.93 dan *recall* 0.97. Sementara itu, kelas *contamination* memiliki *precision* tertinggi sebesar 0.96 namun *recall* yang lebih rendah yaitu 0.78, menghasilkan *f1-score* 0.86. Hal ini menunjukkan bahwa model sedikit kesulitan dalam mengenali seluruh data *contamination* secara menyeluruh. Secara keseluruhan, nilai *macro average* dan *weighted average* untuk *precision*, *recall*, dan *f1-score* berada pada kisaran 0.91–0.92, mengindikasikan bahwa model memiliki performa yang konsisten di antara ketiga kelas cacat.

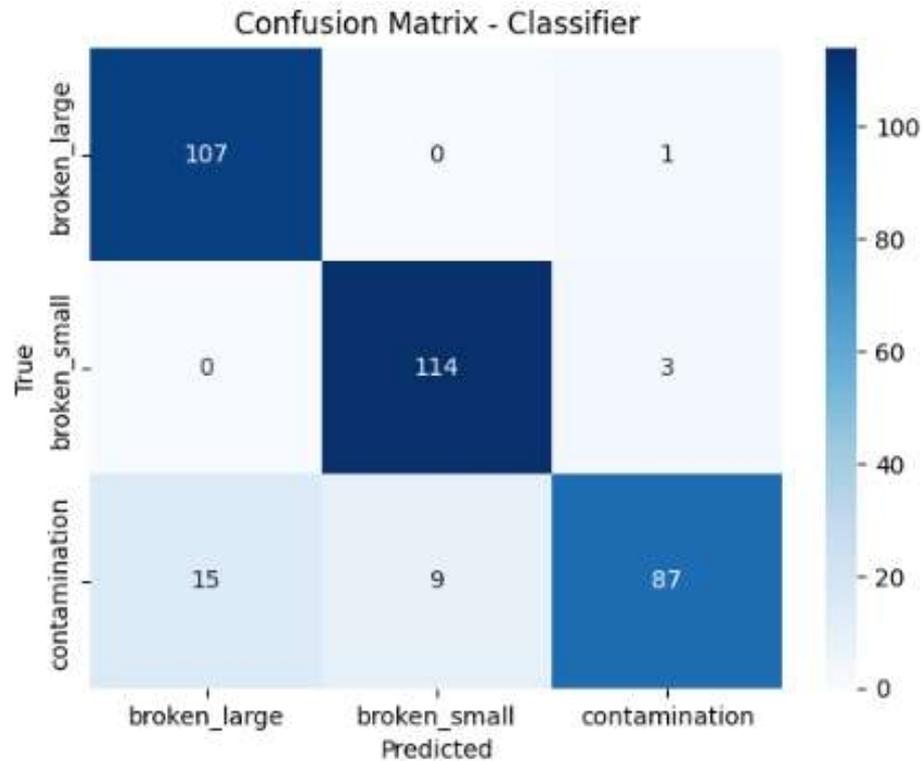
4.2.1 Confusion Matrix

Confusion matrix merupakan salah satu metode evaluasi yang digunakan untuk mengukur performa model klasifikasi dengan menampilkan jumlah prediksi benar dan salah pada masing-masing kelas. Dalam penelitian ini, *Confusion matrix* digunakan untuk mengevaluasi dua tahap sistem: pertama, performa model *U-Net Autoencoder* dalam mendeteksi adanya anomali (cacat) pada citra botol plastik; dan kedua, kinerja model CNN dalam mengklasifikasikan jenis cacat yang terdeteksi.



Gambar 4. 10 *Confusion matrix anomay detection*

Pada gambar 4.10 merupakan *Confusion matrix* hasil evaluasi model deteksi anomali menggunakan *U-Net Autoencoder* pada citra botol plastik. Tabel ini menunjukkan kinerja model dalam mengklasifikasikan citra sebagai "Good" (tidak rusak) atau "Def" (cacat). Dari total 307 data uji, model berhasil mengklasifikasikan 19 citra botol bagus dengan benar dan hanya salah mengklasifikasikan 1 citra bagus sebagai cacat (*false positive*). Sementara itu, seluruh 287 citra botol cacat berhasil dikenali dengan benar (*true positive*), tanpa adanya kesalahan klasifikasi cacat sebagai bagus (*false negative*). Hasil ini menunjukkan bahwa model *anomaly detection* yang digunakan memiliki performa sangat baik dalam membedakan antara botol baik dan cacat, dengan tingkat akurasi, presisi, dan *recall* yang sangat tinggi, terutama dalam mendeteksi botol cacat yang merupakan fokus utama dalam aplikasi deteksi cacat produk.



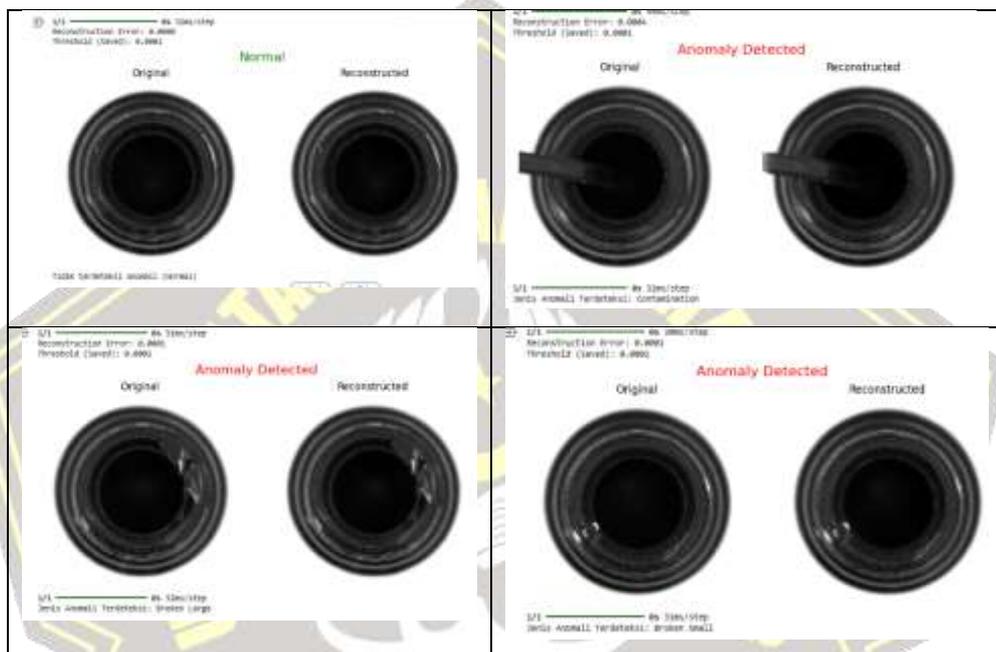
Gambar 4. 11 *Confussion matrix classifier*

Pada Gambar 4.11 ditampilkan *Confussion matrix* dari hasil klasifikasi jenis cacat pada botol plastik menggunakan CNN classifier. Matriks ini menunjukkan kinerja model dalam membedakan tiga jenis cacat, yaitu *broken large*, *broken small*, dan *contamination*. Dari hasil evaluasi, model berhasil mengklasifikasikan 107 gambar *broken large* secara benar, dengan hanya 1 kesalahan klasifikasi ke kelas *contamination*. Untuk kelas *broken small*, model mencatat kinerja sangat baik dengan 114 prediksi yang benar dan hanya 3 gambar yang salah diklasifikasikan sebagai *contamination*. Sementara itu, pada kelas *contamination*, sebanyak 87 gambar diklasifikasikan dengan benar, namun terdapat 15 gambar yang salah diklasifikasikan sebagai *broken large* dan 9 sebagai *broken small*. Hasil ini menunjukkan bahwa meskipun akurasi keseluruhan cukup tinggi, masih terdapat tantangan dalam membedakan antara kelas *contamination* dan jenis cacat lainnya, yang dapat disebabkan oleh kemiripan visual antar kelas.

4.2.2 Hasil Deteksi

Untuk mengevaluasi kemampuan model dalam mendeteksi cacat secara visual, dilakukan analisis hasil deteksi pada beberapa sampel citra dari *dataset* uji. Analisis ini bertujuan untuk melihat sejauh mana model *U-Net Autoencoder* dapat membedakan antara botol yang cacat dan tidak cacat berdasarkan nilai *Reconstruction Error*.

Tabel 4. 4 Hasil Deteksi



Pada tabel 4.4 menunjukkan hasil visualisasi deteksi anomali pada beberapa citra botol plastik, yang mencakup empat kategori hasil deteksi: botol normal (*Good*), *broken large*, *broken small*, dan *Contamination*. Pada masing-masing hasil ditampilkan citra asli (*original*) dan hasil rekonstruksi (*reconstructed*) dari model *U-Net Autoencoder*. Deteksi dilakukan berdasarkan nilai *Reconstruction Error* yang dibandingkan dengan ambang batas (*threshold*) sebesar 0.006. Citra botol yang memiliki *Reconstruction Error* di bawah *threshold*, seperti pada botol normal (*error*: 0.0024), diklasifikasikan sebagai tidak cacat. Sebaliknya, botol dengan *Reconstruction Error* di atas ambang batas terdeteksi sebagai anomali, misalnya: *broken large* (*error*: 0.0065), *Contamination* (*error*: 0.0068), dan *broken small* (*error*: 0.0067). Perbedaan antara citra asli dan hasil

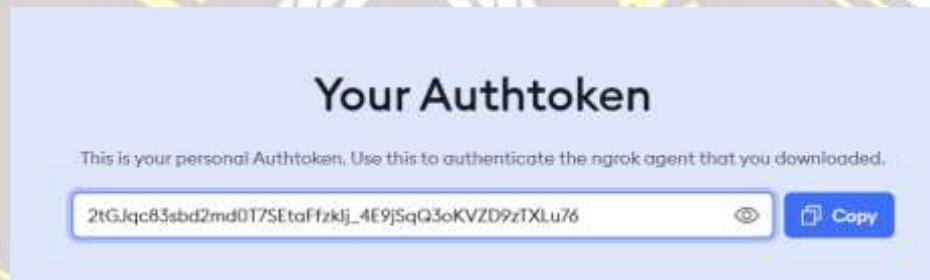
rekonstruksinya cukup jelas terlihat pada bagian yang mengalami cacat, yang memperkuat keandalan metode deteksi ini dalam membedakan antara produk cacat dan tidak cacat berdasarkan tingkat kesalahan rekonstruksi.

4.3 Deployment

Pada sistem deteksi anomali pada botol plastik untuk memudahkan pengguna dalam mengakses sistem deteksi anomali tanpa perlu instalasi lokal, sistem ini di *deploy* dalam bentuk aplikasi *web* menggunakan *Streamlit*, yang dijalankan melalui Google Colab dan dibagikan secara publik menggunakan *pyngrok*. Pendekatan ini memanfaatkan infrastruktur *cloud* gratis serta memungkinkan pengujian dan demonstrasi secara interaktif.

4.3.1 Streamlit

1. Dashboard Ngrok



Gambar 4. 12 Authtoken

Pada gambar 4.12 merupakan tampilan dari *dashboard Ngrok* yang menampilkan *AuthToken* pribadi pengguna. *AuthToken* ini berfungsi sebagai kredensial autentikasi yang digunakan untuk menghubungkan agen Ngrok lokal dengan akun pengguna di server Ngrok. Token ini perlu disalin dan dikonfigurasi pada agen Ngrok yang telah diunduh dan dijalankan secara lokal, biasanya melalui perintah terminal seperti `ngrok config add-authtoken <token>`. Dengan memasukkan token tersebut, Ngrok dapat mengaitkan sesi *tunneling* dengan akun pengguna, memungkinkan penggunaan fitur tambahan seperti *custom domain*, *dashboard* trafik, dan konfigurasi akses yang lebih aman.

Penting untuk menjaga kerahasiaan token ini agar tidak disalahgunakan oleh pihak lain.



Gambar 4. 13 Ngrok Agents

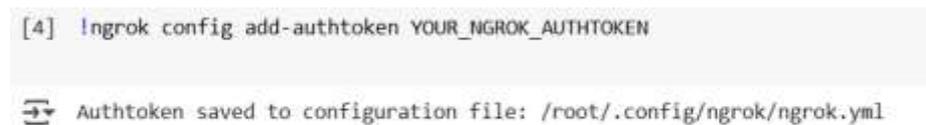
Pada gambar 4.13 menampilkan halaman *Agents* pada *dashboard* Ngrok, yang berfungsi untuk memantau dan mengelola instansi agen Ngrok yang sedang berjalan. Dalam tampilan tersebut, terdapat alamat IP publik 34.168.120.74. Agen ini menggunakan versi ngrok-agent/3.22.1 dan saat ini memiliki satu tunnel yang aktif atau *online*. Agen tersebut dijalankan oleh pengguna dengan email. Informasi ini berguna bagi pengguna untuk mengetahui status koneksi, lokasi server, serta aktivitas agen secara *real-time* melalui *dashboard* Ngrok.

2. *Installation library streamlit*



Gambar 4. 14 Install *library* streamlit pyngrok

Pada gambar 4.14 menunjukkan proses instalasi paket *Streamlit* dan Ngrok menggunakan perintah pip install. Instalasi ini diperlukan agar aplikasi web berbasis *Streamlit* bisa dijalankan dan diakses secara *online* melalui Ngrok.



Gambar 4. 15 Instalasi *Streamlit* Pyngrok serta Konfigurasi Authtoken Ngrok

Pada gambar 4.15 menunjukkan perintah pada ngrok authtoken yang digunakan untuk menyimpan token autentikasi Ngrok ke dalam file konfigurasi pada direktori `/root/.config/ngrok/ngrok.yml`. Token ini diperlukan agar pengguna dapat menjalankan dan mengakses aplikasi *Streamlit* secara *online* melalui terowongan aman yang dibuat oleh Ngrok.

3. *Link Streamlit*

```
Auth token saved to configuration file: /root/.config/ngrok/ngrok.yml
Streamlit Web App dapat diakses di: ngrokTunnel: "https://27b1-34-16-140-235.ngrok-free.app" -> "http://localhost:8501"
```

Gambar 4. 16 *Output* Konfirmasi *Streamlit* dan Alamat Akses Ngrok

Pada gambar diatas merupakan alamat sementara dari aplikasi berbasis *Streamlit* yang dijalankan secara lokal dan dibagikan ke publik menggunakan layanan Ngrok.

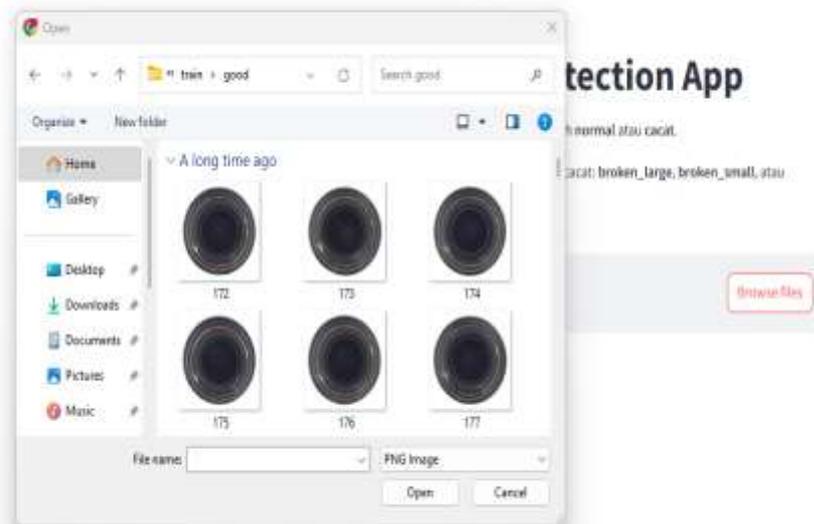
4. *Dashboard Streamlit*

Setelah *User* berhasil mengakses, akan menampilkan tampilan *web* seperti gambar 4.17.



Gambar 4. 17 *Dashboard web* deteksi anomali

Untuk mengunggah gambar yang akan diprediksi maka *User* dapat mengklik tombol *browse files* seperti pada gambar 4.17 sehingga dapat memunculkan gambar yang akan diunggah.



Gambar 4. 18 Mengunggah gambar

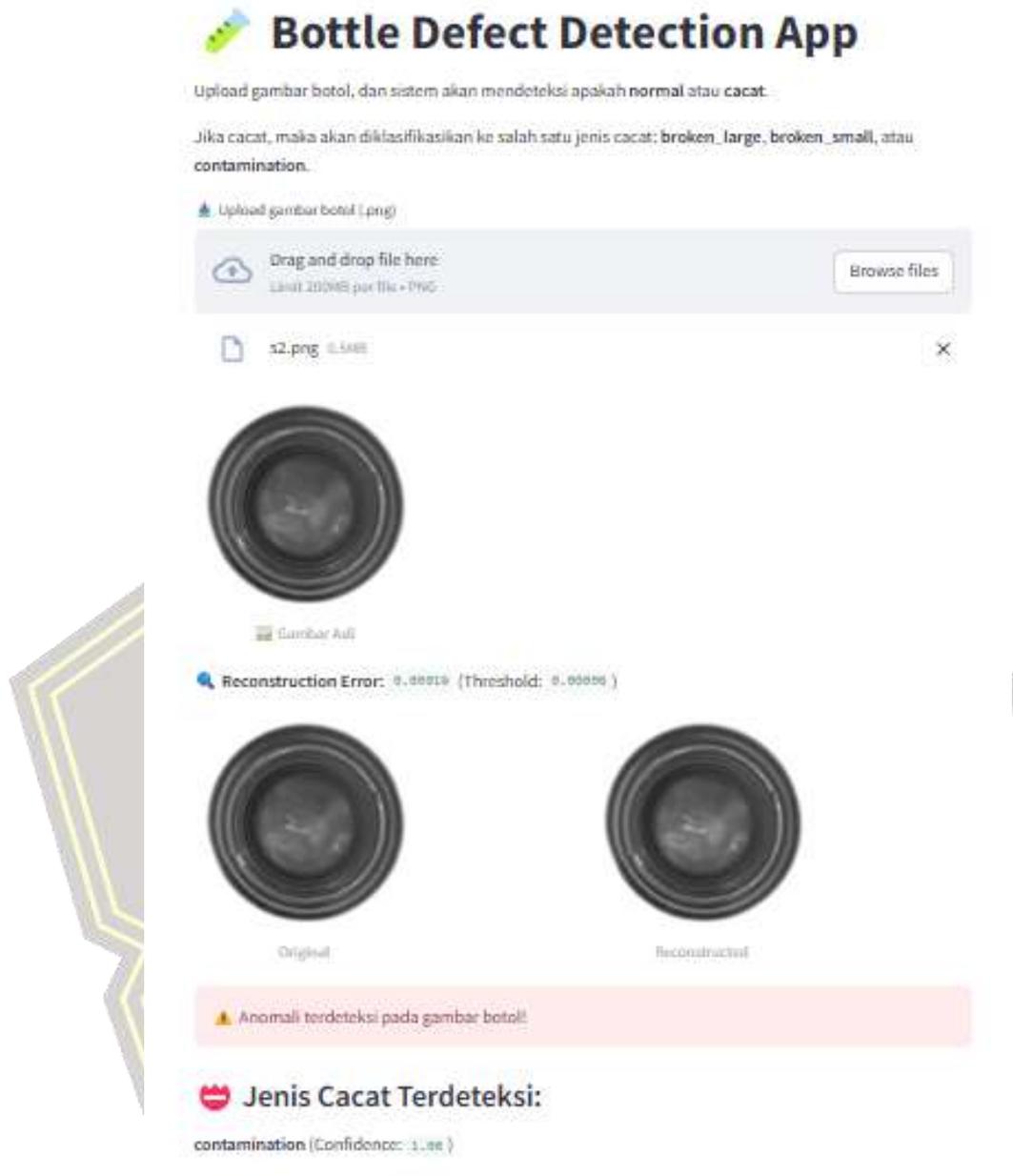
Pada gambar 4.18 adalah tampilan saat pengguna mengunggah gambar ke dalam sistem deteksi botol plastik. Jendela file *explorer* muncul untuk memilih file gambar dengan format JPG, PNG, atau JPEG dari penyimpanan lokal.

5. Hasil Prediksi

Setelah sistem deteksi dan klasifikasi cacat berhasil dibangun dan diintegrasikan ke dalam antarmuka aplikasi, dilakukan pengujian terhadap hasil prediksi secara langsung melalui input citra botol plastik. Proses ini mencakup unggahan citra oleh pengguna, pemrosesan oleh model deteksi anomali, dan jika terdeteksi cacat, dilanjutkan dengan klasifikasi jenis kerusakan.



Gambar 4. 19 Hasil deteksi gambar botol yang terdeteksi normal



Gambar 4. 20 Hasil deteksi gambar botol yang terdeteksi kontaminasi.

Setelah gambar berhasil di unggah, maka sistem akan memproses dan menampilkan hasil prediksi. Pada Gambar 4.19 menunjukkan hasil prediksi terhadap citra botol yang terdeteksi sebagai normal (tanpa cacat), sedangkan Gambar 4.20 menunjukkan hasil prediksi pada citra botol yang terdeteksi mengalami kerusakan dengan jenis kontaminasi. Visualisasi ini membuktikan bahwa sistem mampu memberikan umpan balik secara jelas dan informatif terhadap kondisi botol yang dianalisis.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil mengembangkan sistem otomatis dalam mendeteksi dan mengklasifikasikan cacat pada botol plastik menggunakan metode deep learning berbasis *U-Net Autoencoder* dan *Convolutional Neural Network (CNN)*. Sistem deteksi anomali menggunakan *U-Net Autoencoder* mampu mengenali cacat dengan membandingkan citra asli dan hasil rekonstruksi citra normal. Proses deteksi dilakukan dengan menetapkan ambang batas (*threshold*) sebesar 0.000049, yang diperoleh dari persentil ke-95 data kategori *good*. Dengan *threshold* ini, model berhasil mencapai akurasi deteksi sebesar 99,67%, *precision* 99,65%, *recall* 100%, dan *F1-score* 99,83%, yang menunjukkan kinerja sangat baik dalam mengidentifikasi anomali tanpa kesalahan deteksi (*false negative*). Pada tahap klasifikasi jenis cacat, model CNN digunakan untuk mengklasifikasikan cacat menjadi tiga kategori utama: *Broken Large*, *Broken Small*, dan *Contamination*, dengan akurasi klasifikasi keseluruhan mencapai 92%. Namun, performa model pada kategori *Contamination* masih kurang optimal dengan *recall* sebesar 78%, yang mengindikasikan masih adanya kesulitan model dalam mengenali seluruh data *contamination* secara konsisten. Hal ini kemungkinan disebabkan oleh kemiripan visual antar jenis cacat serta variasi kontaminasi yang belum sepenuhnya terwakili dalam *dataset*.

5.2 Saran

1. Mengembangkan model dengan memperluas jumlah data latih dan menambahkan lebih banyak variasi jenis cacat, termasuk deformasi struktural, warna abnormal, atau cacat permukaan lainnya untuk meningkatkan kemampuan generalisasi model.

2. Meningkatkan variasi *dataset* pada kelas *Contamination*, termasuk memperluas jenis kontaminasi seperti debu, minyak, dan noda besar/kecil serta menerapkan augmentasi spesifik untuk meningkatkan kemampuan generalisasi model pada kelas tersebut.
3. Mengembangkan sistem deteksi real-time yang terintegrasi dengan kamera produksi dan perangkat keras lainnya untuk penerapan sistem secara langsung di jalur produksi industri.
4. Melakukan uji coba ketahanan dan keandalan model dalam kondisi pencahayaan dan latar belakang yang bervariasi agar sistem tetap akurat dalam situasi produksi nyata.



DAFTAR PUSTAKA

- Abadi, M., Agarwal, A., & Barham, P. (2021). TensorFlow: A system for large-scale machine learning. *Communications of the ACM*, 64(9), 62–71. <https://doi.org/https://doi.org/10.1145/3454124>
- Akçay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N., & Genc, U. (2022). Anomalib: a Deep Learning Library for Anomaly Detection. *Proceedings - International Conference on Image Processing, ICIP*, 1706–1710. <https://doi.org/10.1109/ICIP46576.2022.9897283>
- Arif, N., Mercu, U., & Yogyakarta, B. (2025). *VARIATIONAL AUTOENCODER DAN EXTREME VALUE*. 13(2), 709–716.
- Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z., & Shao, L. (2021). PENGARUH PRODUK CACAT DAN PRODUK RUSAK TERHADAP HARGA POKOK PRODUKSI (Studi Pada Perusahaan Tekstil CV. Suho Garmino). *Applied Sciences (Switzerland)*, 11(16).
- Choi, Y. W., & Lee, S. W. (2024). Development of a Defect Diagnosis Algorithm for Blow-molded Transparent Plastic Bottles based on Convolutional Neural Networks (CNN). *International Journal of Precision Engineering and Manufacturing - Smart Technology*. <https://doi.org/https://doi.org/10.23919/IJPEM-ST.2024.00041>
- Darsin, M., Asrofi, M., Anggianto, J., & Soesatijono, S. (2020). Upaya mengatasi cacat produksi botol kemasan air 600 ml dengan metode Statistical Process Control. *Jurnal Litbang Industri*, 10(2), 129. <https://doi.org/10.24960/jli.v10i2.6139.129-137>
- Ganesh, M., Kumar, A., & Pattabiraman, V. (2020). Autoencoder based network anomaly detection. *Proceedings of 2020 IEEE International Conference on Technology, Engineering, Management for Societal Impact Using Marketing, Entrepreneurship and Talent, TEMSMET 2020, April*. <https://doi.org/10.1109/TEMSMET51618.2020.9557464>
- Kalla, A. (2025). Detecting Anomalies in Industrial Images with Neural Networks: How We Built a U-Net Autoencoder for Quality Control. *Medium*. <https://medium.com/@aparnakalla11/detecting-anomalies-in-industrial->

images-with-neural-networks-how-we-built-a-U-Net-Autoencoder-for-057c7aa37bf9

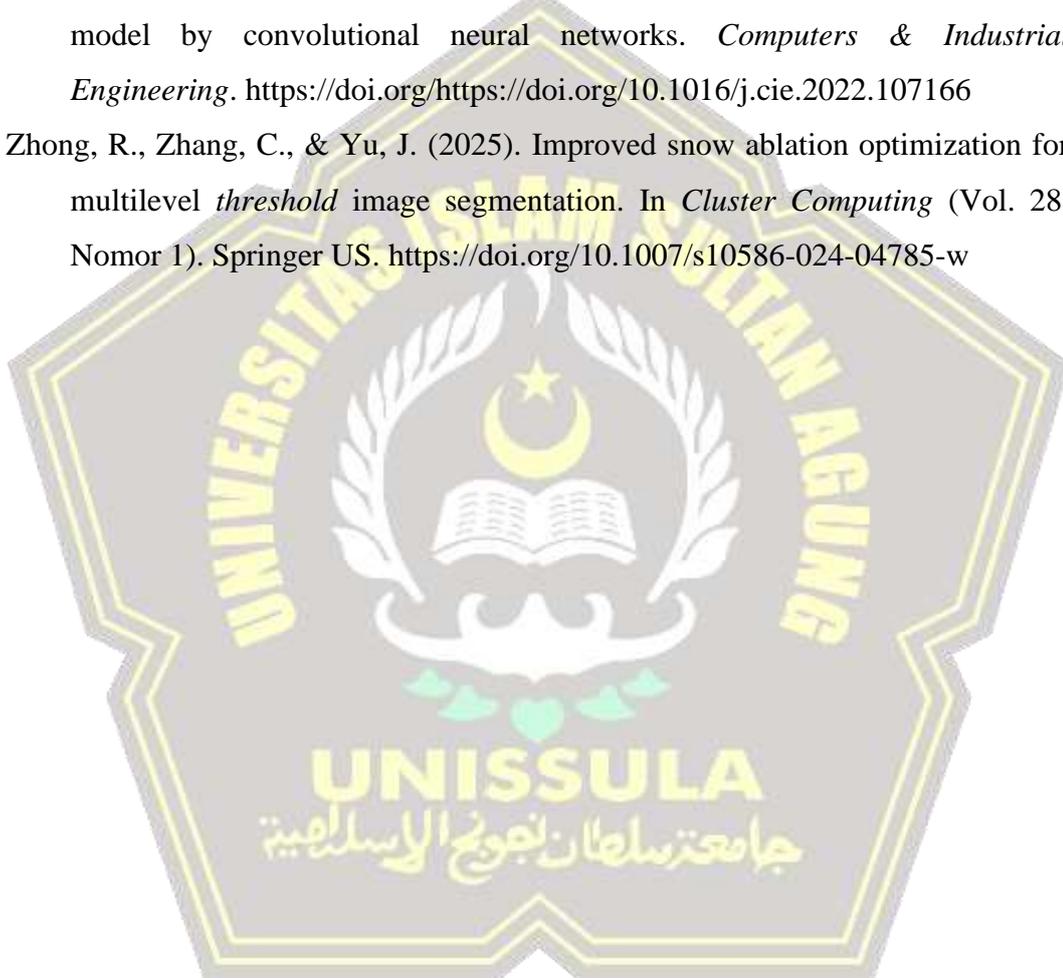
- Kasnanda Bintang, Y., Imaduddin, H., Kasnanda, Y., & Corresponding Author, B. (2024). *Pengembangan Model Deep Learning Untuk Deteksi Retinopati Diabetik Menggunakan Metode Transfer Learning*. 9(3), 1442–1455.
- Maulida Surbakti, N., Talia, A., Br Perangin-Angin, C., Olivia Nainggolan, D., Devi Friskaully, N., & Ruth Br Tumorang, S. (2024). Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel. *Kebumihan dan Angkasa*, 2(3), 98–107. <https://doi.org/10.62383/algorithm.v2i3.67>
- Mohandas, R., Southern, M., Fitzpatrick, C., & Hayes, M. (2024). Deep learning enabled computer vision in remanufacturing and refurbishment applications: defect detection and grading for smartphones. *Circular Economy and Sustainability*. <https://doi.org/https://doi.org/10.1007/s13243-024-00147-2>
- Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, 13(1), 911–921. <https://doi.org/10.12785/ijcds/130172>
- Sabri, A. (2022). Segmentasi Anomali Produk Dengan Ssim Autoencoder. *Prosiding Seminar SeNTIK*, 6.
- Sharma, A. (2023). *Image Segmentation with U-Net in PyTorch: The Grand Finale of the Autoencoder Series*. PyImageSearch. <https://pyimagesearch.com/2023/11/06/image-segmentation-with-U-Net-in-pytorch-the-grand-finale-of-the-Autoencoder-series/>
- Singh, J., Yadav, A., Siddhu, A., & Sharma, S. (2023). *Comparative Analysis of Autoencoders and U-Net-Based Image Steganography* (hal. 643–655). https://doi.org/10.1007/978-981-19-8865-3_57
- Soesilo, R., Valentin, A. D., Ramadhani, A., & Nirfison. (2024). *Proses Manufaktur Modern: Memahami Proses dan Metode Produksi* (Nurhaeni (ed.); 1 ed.). CV. Mega Press Nusantara.
- Tang, Y., Li, X., Zhao, Y., Zhang, X., & Wang, H. (2023). Industrial anomaly

detection with multiscale *Autoencoder* and deep feature extractor-based neural network. *IET Image Processing*, 17(5), 456–468. <https://doi.org/https://doi.org/10.1049/ipr2.12752>

Wahid, A., Nuriyanto, Munir, M., & Syarifuddin, A. (2023). Identifikasi Cacat Produk Botol Plastik 500 mL Dengan Pendekatan Metode FTA (Fault Tree Analysis) di PT. X Pasuruan. *Journal of Industrial View*, 05, 36–48.

Wang, K. J., Jiang, H. F., & Lee, Y. X. (2022). A multiple-stage defect detection model by convolutional neural networks. *Computers & Industrial Engineering*. <https://doi.org/https://doi.org/10.1016/j.cie.2022.107166>

Zhong, R., Zhang, C., & Yu, J. (2025). Improved snow ablation optimization for multilevel *threshold* image segmentation. In *Cluster Computing* (Vol. 28, Nomor 1). Springer US. <https://doi.org/10.1007/s10586-024-04785-w>



LAMPIRAN

