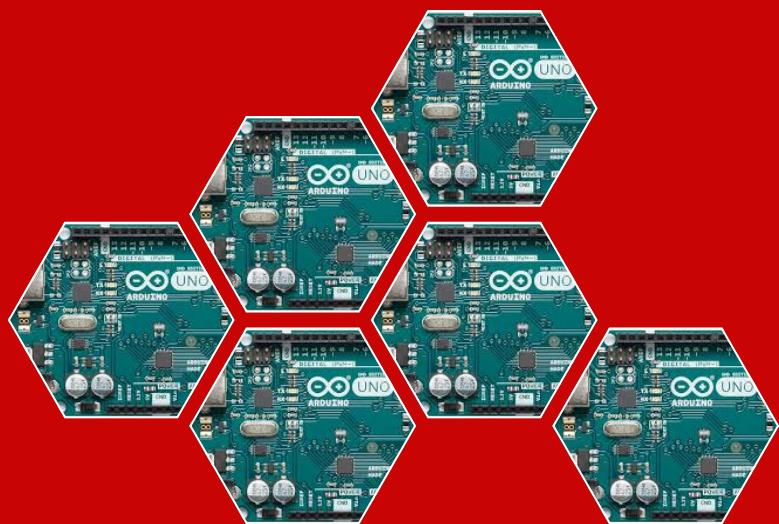


**PROTOTIPE SISTEM DETEKSI DAN PENANGANAN CACAT PADA
KONTAINER KIMIA UNTUK AREA INDUSTRI-TERISOLASI
BERBASIS ARTIFICIAL INTELLIGENCE DAN LENGAN ROBOT**



**ALRIDHO
H021211006**



**PROGRAM STUDI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2025**

**PROTOTIPE SISTEM DETEKSI DAN PENANGANAN CACAT PADA
KONTAINER KIMIA UNTUK AREA INDUSTRI-TERISOLASI
BERBASIS ARTIFICIAL INTELLIGENCE DAN LENGAN ROBOT**

**ALRIDHO
H021211006**



**PROGRAM STUDI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2025**

**PROTOTIPE SISTEM DETEKSI DAN PENANGANAN CACAT PADA
KONTAINER KIMIA UNTUK AREA INDUSTRI-TERISOLASI
BERBASIS ARTIFICIAL INTELLIGENCE DAN LENGAN ROBOT**

ALRIDHO
H021211006

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Fisika

pada

**PROGRAM STUDI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2025**

SKRIPSI**PROTOTIPE SISTEM DETEKSI DAN PENANGANAN CACAT PADA
KONTAINER KIMIA UNTUK AREA INDUSTRI-TERISOLASI
BERBASIS ARTIFICIAL INTELLIGENCE DAN LENGAN ROBOT**

ALRIDHO
H021211006



telah dipertahankan di depan Panitia Ujian Sarjana Xxxx pada tanggal bulan
tahun dan dinyatakan telah memenuhi syarat kelulusan
pada

Mengesahkan:
Pembimbing tugas akhir,

Prof. Dr. Arifin, M.T
NIP. 19670520 199403 1 002

Mengetahui:
Ketua Program Studi,

Prof. Dr. Arifin, M.T
NIP. 19670520 199403 1 002

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa, skripsi berjudul "**PROTOTIPE SISTEM DETEKSI DAN PENANGANAN CACAT PADA KONTAINER KIMIA UNTUK AREA INDUSTRI-TERISOLASI BERBASIS ARTIFICIAL INTELLIGENCE DAN LENGAN ROBOT**" adalah benar karya saya dengan arahan dari pembimbing (Prof. Dr. Arifin, M.T). Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apapun kepada perguruan tinggi mana pun. Sumber inf ormasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daf tar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, XX Juni 2025

Alridho
H021211006

UCAPAN TERIMA KASIH

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce malesuada non dui quis condimentum. Etiam dapibus ligula sapien:

1. Items are numbered automatically.
2. The numbers start at 1 with each use of the `enumerate` environment.
3. Another entry in the list.

Makassar, XX Juni 2025

Alridho
H021211006

ABSTRAK

Alridho. **JUDUL BAHASA INDONESIA** (dibimbing oleh Prof. Dr. Arifin, M.T.).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata Kunci: Prototipe; monitoring; IoT; sensor; telegram.

ABSTRACT

Alridho. **JUDUL BAHASA INGRIS** (dibimbing oleh Prof. Dr. Arifin, M.T.).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata Kunci: Prototipe; monitoring; IoT; sensor; telegram.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PENGESAHAN	iii
LEMBAR PERNYATAAN KEASLIAN SKRIPSI	iv
UCAPAN TERIMA KASIH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	3
1.3 Manfaat Penelitian	3
BAB 2 METODE PENELITIAN	4
2.1 Tempat dan Waktu Penelitian	4
2.2 Peralatan Penelitian	4
2.3 Metode Kerja	4
2.3.1 Perancangan <i>Hardware</i>	6
2.3.2 Perancangan <i>Software</i>	7
2.3.3 Bagan Alir Sistem Kerja Alat	9
BAB 3 HASIL DAN PEMBAHASAN	11
3.1 Hasil Perancangan Sistem	11
3.2 Hasil Perancangan Model YOLO	11
3.2.1 Dataset dan <i>Preprocessing</i>	11
3.2.2 Hasil Pelatihan Model	12
3.3 Hasil Perancangan Model Deteksi Kecacatan	16
3.3.1 Arsitektur Variational Autoencoder	16
3.3.2 Dataset dan <i>Preprocessing</i>	16
3.3.3 Hasil Pelatihan Model	17
3.3.4 Penentuan <i>Threshold</i> Kecacatan	19
3.4 Hasil Pengujian Sistem Lengan Robot	21
BAB 4 KESIMPULAN	26
4.1 Kesimpulan	26
4.2 Saran	26
DAFTAR PUSTAKA	26

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan pesatnya perkembangan *Artificial Intelligence* (AI), robotika, dan otomatisasi, penggunaan lengan robot otomatis semakin meluas di sektor manufaktur, logistik, dan layanan (Jhang et al., 2024). Otomatisasi bertujuan untuk menjalankan rangkaian tindakan sesuai dengan proses yang telah ditetapkan tanpa intervensi manusia, dengan mengendalikan perangkat mekanis secara otomatis (Oaki et al., 2023). Lengan robot mampu melakukan berbagai tugas, seperti perakitan, penanganan, dan pengemasan, sehingga manusia tidak perlu lagi melakukan pekerjaan yang berulang-ulang (Lin et al., 2024). Dalam industri manufaktur, lengan robot sering dipadukan dengan sensor kamera, algoritma visi komputer, dan teknologi otomatisasi guna mendeteksi cacat pada objek dengan tingkat presisi dan efisiensi yang tinggi.

Dalam konteks industri, terutama pada penanganan material berbahaya seperti kontainer kimia, aspek keamanan menjadi prioritas utama. Inspeksi visual otomatis menggunakan robot diperlukan untuk mengurangi risiko paparan zat berbahaya kepada manusia. Selain itu, isolasi dalam proses deteksi cacat pada kontainer kimia sangat penting, karena kerusakan atau kontaminasi pada kontainer dapat menimbulkan risiko keselamatan bagi konsumen. Truong et al. (2024) menyebut tingkat kesalahan inspeksi manual juga cukup tinggi, berkisar 20% hingga 30%. Beberapa faktor yang menyebabkan kesalahan ini, termasuk kelelahan, stres, kesendirian, pencahayaan yang tidak memadai, dan kurangnya pengalaman. Peningkatan penggunaan robot juga meningkatkan produktivitas tenaga kerja sebesar 0,36% per tahun, meningkatkan produktivitas total, dan menurunkan harga *output* (Graetz et al., 2018). Tsai et al. (2021) mengungkap, antara tahun 2005 dan 2011, jumlah robot per 1.000 pekerja meningkat sebesar 25%, yang menyebabkan penurunan 72.658 cedera kerja per tahun. Penurunan cedera ini diperkirakan menghemat \$1,67 miliar (sekitar Rp27,3 triliun) per tahun, atau total \$11,69 miliar (sekitar Rp191,2 triliun) selama periode tersebut. Oleh karena itu, penerapan metode otomatis yang tepat untuk mendeteksi cacat dan kontaminan sangat krusial guna menjamin keamanan, mencapai skalabilitas, dan meningkatkan efisiensi biaya.

Algoritma *deep learning* berupa *autoencoder* dapat dimanfaatkan untuk mendeteksi anomali dan cacat permukaan (Tsai et al., 2021). Model *autoencoder*, khususnya *convolutional autoencoder* (CAE), dapat dilatih hanya dengan sampel bebas cacat (Chen et al., 2021). CAE bekerja dengan meminimalkan kesalahan rekonstruksi sehingga fitur-fitur representatif dapat teridentifikasi secara optimal. Dengan melatih model pada data normal, setiap sampel yang mengandung cacat pada tahap evaluasi akan menunjukkan perbedaan yang signifikan dibandingkan dengan data bebas cacat yang telah dilatih (Liu et al., 2025). Pendekatan ini sangat

berguna ketika pengumpulan data cacat terbukti sulit atau mahal.

Berbagai studi telah memanfaatkan YOLO bersama lengan robot untuk beragam aplikasi. Wang et al. (2024) mengembangkan model kontrol lengan robot berbasis YOLO yang lebih efisien untuk menggenggam komponen logam secara akurat di lingkungan industri yang kompleks. Kato et al. (2022) mengusulkan metode deteksi objek parsial dan estimasi kedalaman menggunakan kombinasi YOLO dan CNN untuk mengontrol lengan robot, yang terbukti efektif dalam eksperimen pengambilan objek dengan robot 4-DOF. Kim et al. (2021) menerapkan YOLO dalam metode genggam robot untuk deteksi dan pemilahan sampah secara *real-time*, dengan pendekatan pembatasan area pasca deteksi objek. Jin et al. (2025) mengembangkan versi peningkatan dari YOLOv8n dengan integrasi modul *dilated re-parameterization*, *feature pyramid*, dan *Scylla-IoU loss* untuk meningkatkan akurasi dan adaptabilitas lengan robot pemetik apel dalam kondisi kebun yang kompleks.

Sementara itu, *autoencoder* adalah model jaringan saraf dalam yang dilatih untuk merekonstruksi data *input*, dan digunakan dalam deteksi cacat dengan membandingkan perbedaan signifikan antara *input* normal dan hasil rekonstruksi dari data yang mengandung cacat. Bionda et al. (2022) menerapkan *autoencoder* dengan fungsi *loss* berbasis CW-SSIM untuk mendeteksi anomali pada citra industri, yang terbukti lebih efisien dibandingkan jaringan saraf besar lainnya. Yun et al. (2023) mengusulkan pendekatan *autoencoder* untuk mendeteksi kerusakan pada lengan robot industri berdasarkan sinyal suara internal, dengan memanfaatkan citra spektrogram STFT. Jia et al. (2023) memperkenalkan model deteksi anomali dengan struktur *encoder-decoder-encoder* (EDE) dan pelatihan dua tahap, yang menggabungkan rekonstruksi dan pendekatan konfrontasi generatif. Kozamernik et al. (2025) mengembangkan FuseDecode *autoencoder*, yang menerapkan pembelajaran bertahap mulai dari tanpa supervisi, semi-supervisi, hingga supervisi campuran, dan menunjukkan keunggulan dalam deteksi cacat pada data industri nyata serta dataset MVTec AD. Ruediger-Flore et al. (2024) membandingkan klasifikasi biner dan *autoencoder* dalam deteksi anomali citra untuk proses perakitan rangka, khususnya pada kesalahan posisi dan rotasi komponen, dan menyimpulkan bahwa *autoencoder* lebih unggul dalam mendeteksi anomali halus serta fleksibel terhadap data terbatas.

Kebutuhan untuk meningkatkan efisiensi dan keamanan dalam inspeksi industri mendorong pemanfaatan AI dan robotika. Deteksi cacat secara otomatis, terutama dalam penanganan material berbahaya, sangat penting untuk memastikan kualitas produk dan keselamatan manusia. *Autoencoder* merupakan algoritma yang efektif karena kemampuannya untuk belajar dari data normal sekaligus mengidentifikasi anomali. Dengan demikian, penelitian ini bertujuan untuk mengeksplorasi dan mengoptimalkan penerapan *autoencoder* dalam mendeteksi cacat permukaan, sehingga dapat berkontribusi pada kemajuan manufaktur yang cerdas dan berkelanjutan.

1.2 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini yaitu:

1. Merancang dan melatih model deteksi objek berbasis YOLO yang mampu mengenali kontainer kimia secara akurat dalam berbagai kondisi.
2. Membangun model deteksi kecacatan menggunakan algoritma *convolutional autoencoder* yang efektif dalam membedakan antara kontainer cacat dan tidak cacat.
3. Mengintegrasikan model deteksi objek dan deteksi kecacatan ke dalam sistem berbasis lengan robot untuk proses identifikasi dan penyortiran secara otomatis.

1.3 Manfaat Penelitian

Manfaat yang ingin dicapai dalam penelitian ini yaitu:

1. Memberikan kontribusi terhadap pengembangan sistem deteksi objek yang cepat dan akurat di lingkungan industri, khususnya untuk aplikasi inspeksi visual kontainer kimia dalam kondisi nyata yang bervariasi.
2. Menyediakan solusi yang efektif dalam mendeteksi kecacatan secara otomatis, sehingga dapat menggantikan metode inspeksi manual yang memakan waktu dan rentan terhadap kesalahan manusia (*human error*).
3. Mendorong otomatisasi penuh dalam proses identifikasi dan penyortiran kontainer kimia, sehingga meningkatkan efisiensi produksi, menurunkan biaya operasional, dan memperkecil risiko kesalahan klasifikasi dalam sistem manufaktur.

BAB 2

METODE PENELITIAN

2.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan mulai dari bulan Februari 2025 hingga Juni 2025, bertempat di Laboratorium Elektronika dan Instrumentasi, Departemen Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin, Makassar.

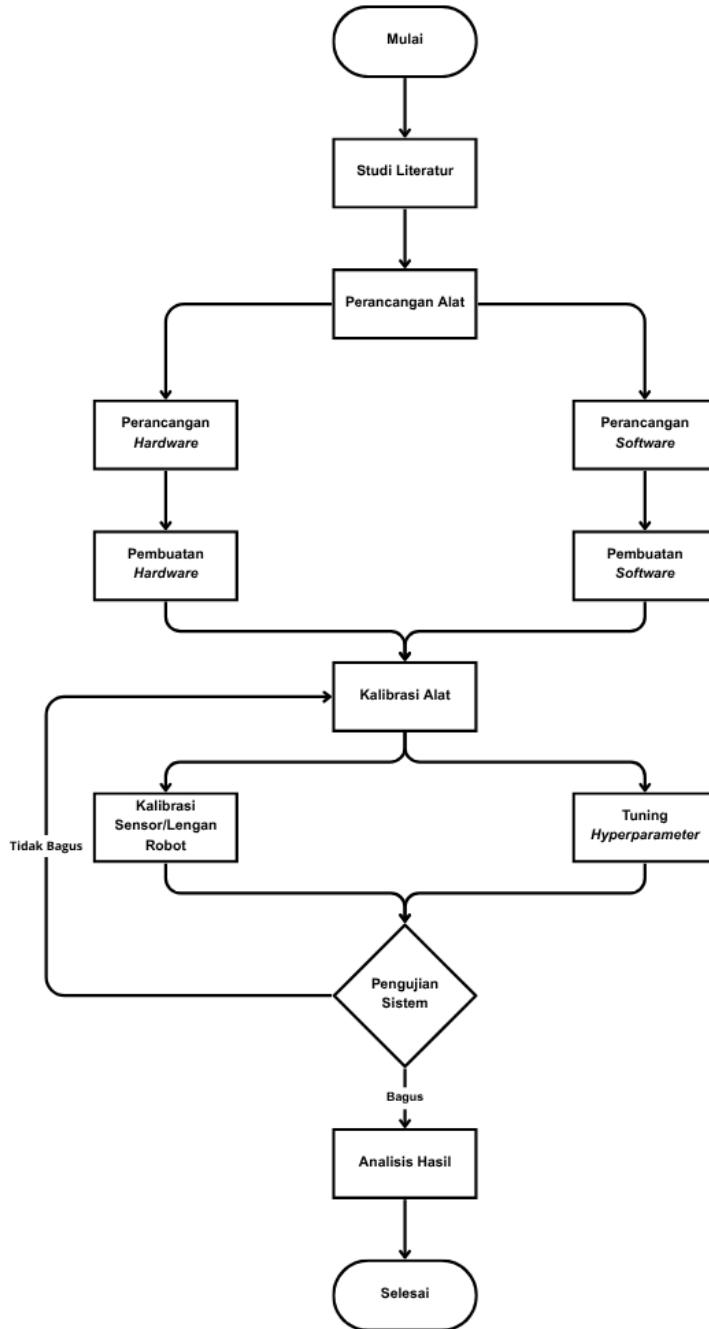
2.2 Peralatan Penelitian

Adapun peralatan yang digunakan pada penelitian ini adalah sebagai berikut:

1. Arduino Uno berfungsi sebagai mikrokontroler utama yang mengendalikan motor servo pada lengan robot serta menerima sinyal dari sensor.
2. Motor servo digunakan sebagai aktuator untuk menggerakkan bagian-bagian lengan robot sesuai perintah dari Arduino.
3. Lengan robot EEZYbotARM MK1 berfungsi sebagai struktur mekanik yang menjadi tempat pemasangan motor servo dan berperan sebagai sistem pergerakan robotik.
4. *Power supply* 5V berfungsi memberikan catu daya stabil untuk motor servo agar dapat beroperasi dengan baik.
5. *Sensor Passive Infrared Receiver* (PIR) HC-SR501 digunakan untuk mendeteksi gerakan dan membantu menghitung jumlah kontainer cacat dan non-cacat yang lewat.
6. Kamera digunakan untuk mengambil gambar kontainer kimia, yang kemudian diproses oleh model deteksi (YOLO) dan deteksi cacat (*autoencoder*).
7. Laptop/komputer digunakan untuk mengunggah program ke Arduino Uno, serta menjalankan model deteksi berbasis YOLO dan *autoencoder* untuk analisis visual.
8. Kabel *jumper* berfungsi menghubungkan berbagai komponen elektronik seperti sensor dan aktuator ke papan rangkaian dan Arduino.
9. Papan rangkaian berfungsi untuk menyediakan jalur koneksi antar komponen.

2.3 Metode Kerja

Dalam penelitian ini terdapat beberapa tahapan yang harus dilakukan. Tahapan penelitian secara lengkap dapat dilihat pada *flowchart* yang tertera pada Gambar 1. Penelitian ini dibatasi pada perancangan dan pembuatan prototipe sistem deteksi cacat otomatis.



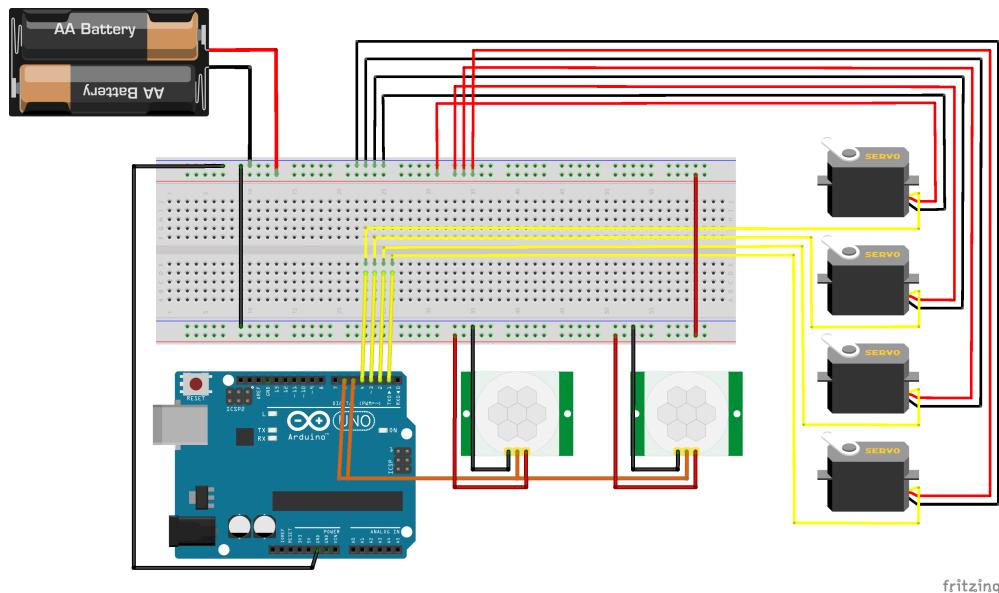
Gambar 1. Bagan alir penelitian

Tahapan dimulai dengan kajian mendalam terhadap teknologi robotik, algoritma *autoencoder* untuk deteksi anomali visual, serta metode deteksi objek seperti *You Only Look Once* (YOLO), guna memperoleh pemahaman komprehensif

tentang konsep dasar, format *dataset*, dan teknik perancangan model untuk mendeteksi cacat pada kontainer kimia. Setelah pemahaman awal diperoleh, dilakukan perancangan sistem yang mencakup perangkat keras (lengan robot dan sensor) serta perangkat lunak berupa algoritma *machine learning*. Selanjutnya, sistem robot dikalibrasi agar dapat bekerja secara optimal, termasuk proses *tuning hyperparameter* pada model pembelajaran mesin. Jika sistem telah berfungsi sesuai dengan yang diharapkan, maka dilanjutkan dengan proses pengambilan data sebagai langkah awal dalam pengujian dan validasi model.

2.3.1 Perancangan Hardware

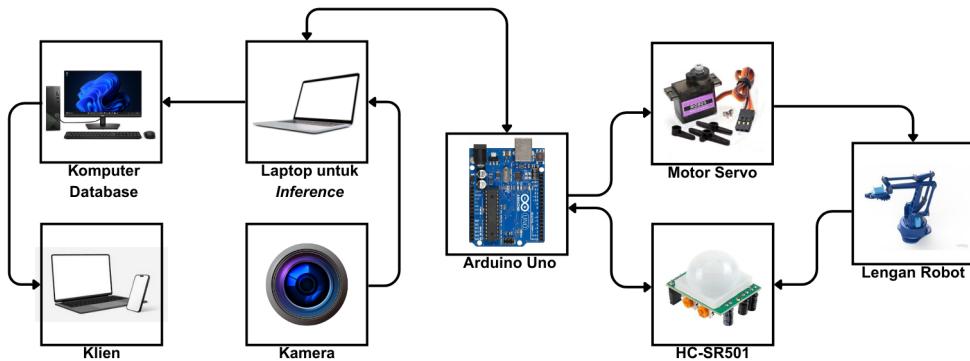
Penelitian ini dimulai dengan tahap perancangan *hardware*. Komponen *hardware* yang digunakan meliputi kamera untuk mengambil gambar kontainer kimia, laptop sebagai pusat pemrosesan dan eksekusi algoritma pembelajaran mesin, serta *motion sensor* untuk menghitung jumlah kontainer kimia, baik yang cacat maupun yang tidak. Adapun rancangan hardware dapat dilihat pada Gambar 2.



Gambar 2. Bagan alir penelitian

Sistem diawali saat kamera menangkap gambar kontainer kimia yang diletakkan di area pengambilan gambar. Gambar ini diproses oleh model deteksi objek (YOLO) untuk mengenali keberadaan kontainer sebelum tahap deteksi cacat. Selanjutnya, gambar yang telah dikenali dikirim ke model deteksi kecacatan berbasis *convolutional variational autoencoder* (CVAE) untuk menentukan apakah kontainer mengalami cacat atau tidak. Berdasarkan hasil analisis tersebut, sinyal dikirimkan ke mikrokontroler Arduino untuk menggerakkan servo sebagai respon terhadap kondisi kontainer. Lengan robot kemudian mengambil kontainer kimia dan memindahkannya ke wadah yang sesuai, tergantung pada hasil deteksi—apakah kontainer tersebut

cacat atau tidak. Untuk memantau dan menghitung jumlah kontainer yang telah dipindahkan, dua *motion sensor* dipasang pada masing-masing wadah (cacat dan noncacat). Data dari sensor ini dikirim ke *web server*, yang menyediakan API untuk dikonsumsi agar data jumlah kontainer dapat ditampilkan ke klien secara *real-time*. Secara keseluruhan, keterkaitan antar komponen perangkat keras dalam sistem ini dapat dilihat pada Gambar 3.



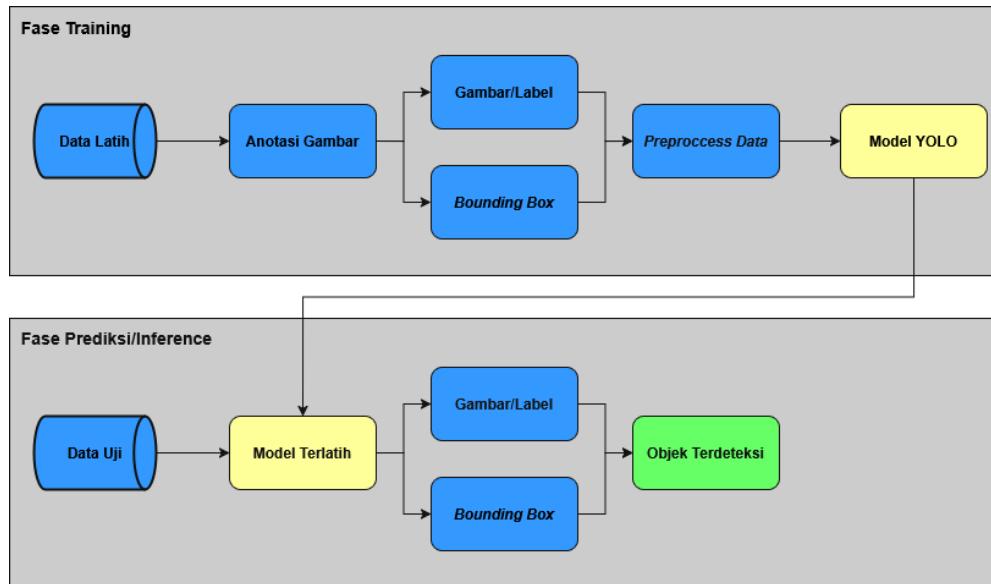
Gambar 3. Rancang sistem *hardware*

2.3.2 Perancangan Software

Perangkat lunak dalam penelitian ini mencakup perancangan beberapa algoritma inti. Pertama, dirancang model deteksi objek berbasis YOLO untuk mengenali kontainer kimia pada citra yang diambil oleh kamera. Kedua, digunakan CVAE sebagai model untuk mendeteksi cacat atau anomali visual pada kontainer. Selain itu, dirancang pula algoritma kontrol untuk mengatur pergerakan lengan robot dalam mengambil dan memindahkan kontainer berdasarkan hasil klasifikasi. Sistem ini juga terintegrasi dengan modul IoT untuk menampilkan data kontainer cacat dan non-cacat pada klien secara *real-time* melalui *website*.

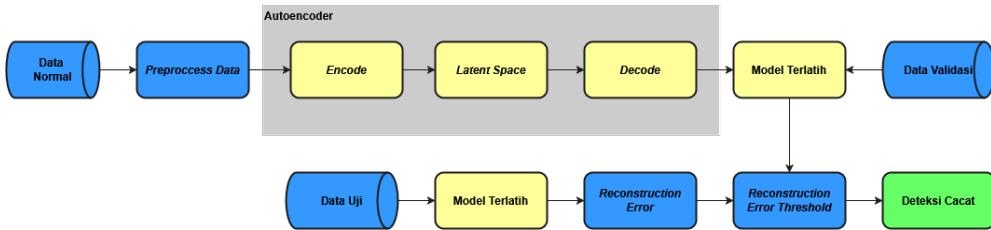
Tahap perancangan model deteksi objek dimulai dengan pengumpulan *dataset* berupa gambar kontainer kimia dari berbagai kondisi dan sudut pandang menggunakan kamera, yang nantinya dipasang bersama lengan robot. Setelah gambar terkumpul, dilakukan proses anotasi dengan memberikan label dan *bounding box* pada setiap kontainer sesuai dengan format yang dibutuhkan oleh algoritma YOLO. Dataset yang telah dianotasi kemudian digunakan untuk melatih model deteksi objek. Tujuannya agar model mampu mendeteksi kontainer kimia secara akurat dan cepat dalam berbagai situasi, misalnya ketika kontainer berada dalam posisi miring. Setelah proses pelatihan selesai, model dievaluasi menggunakan data uji yang belum pernah dilihat sebelumnya. Evaluasi dilakukan menggunakan beberapa metrik seperti *precision*, *recall*, dan *mean Average Precision* (mAP), guna memastikan bahwa model memiliki kemampuan generalisasi yang baik dan layak diterapkan di sistem robotik secara *real-time*. *Pipeline* perancangan model deteksi objek dapat

dilihat pada Gambar 4.



Gambar 4. Diagram *pipeline* pelatihan model YOLO

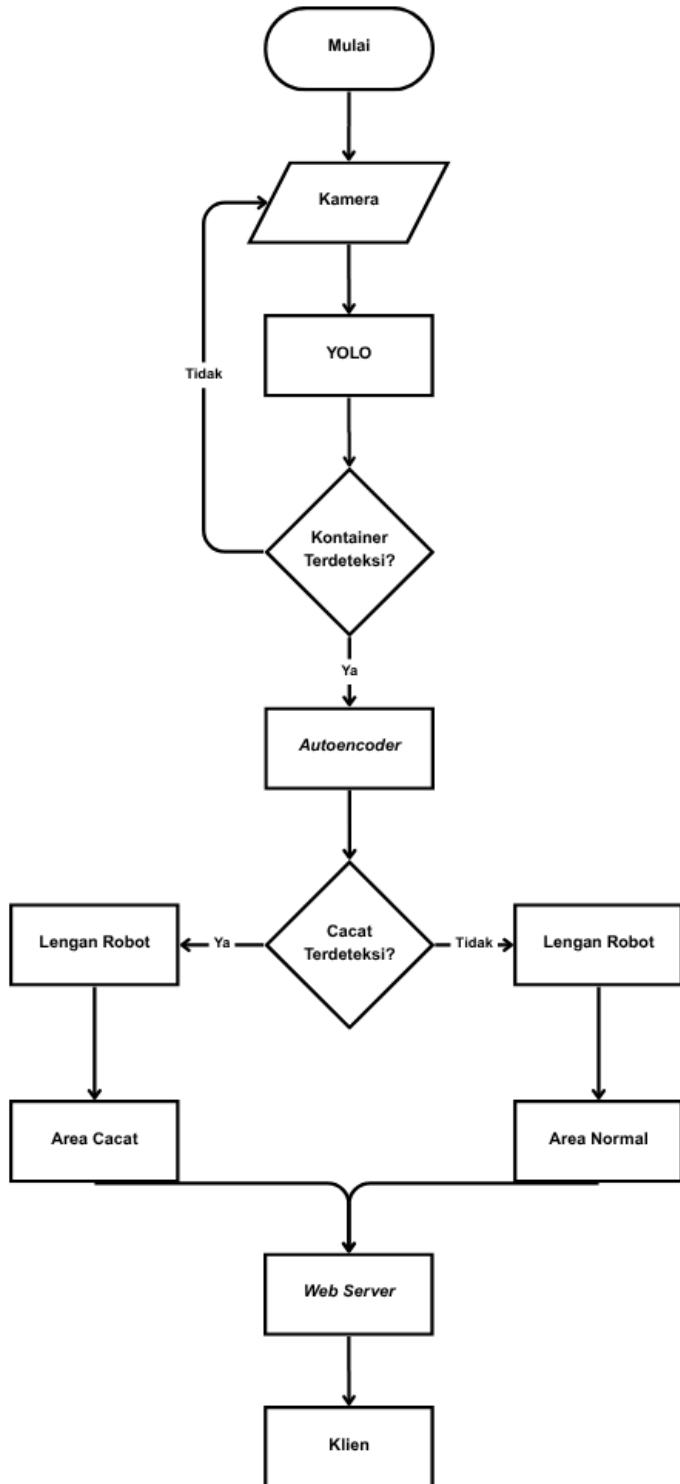
Berikutnya adalah tahap pembangunan model deteksi cacat menggunakan algoritma CVAE. Tahap ini menggunakan *dataset* yang sama seperti yang digunakan pada pelatihan model YOLO dengan beberapa tambahan, namun tanpa menggunakan anotasi *bounding box* karena sifat *unsupervised* dari *autoencoder*. Data diproses melalui tahap *preprocessing* seperti *resizing*, normalisasi, dan augmentasi (*rotasi*, *flipping*, pencahayaan) untuk meningkatkan variasi. Model dirancang dengan dua komponen utama: *encoder* untuk mengekstraksi fitur penting dan menghasilkan representasi berdimensi rendah (*latent space*), serta *decoder* untuk merekonstruksi gambar dari representasi tersebut. Setelah arsitektur selesai dan data siap, model dilatih untuk meminimalkan perbedaan antara gambar asli dan hasil rekonstruksi, sehingga mampu mengenali citra normal secara akurat. Evaluasi dilakukan dengan menghitung *reconstruction error*, yang digunakan untuk membedakan antara gambar normal dan cacat. Ambang batas deteksi ditentukan melalui analisis distribusi *error* pada data validasi. *Pipeline* perancangan model deteksi cacat dapat dilihat pada Gambar 5.



Gambar 5. Diagram *pipeline* pelatihan model deteksi cacat

2.3.3 Bagan Alir Sistem Kerja Alat

Bagan alir kerja sistem secara keseluruhan dimulai saat kamera menangkap citra, yang kemudian diproses oleh model YOLO untuk mendeteksi objek kontainer. Setelah terdeteksi, model *autoencoder* menganalisis citra tersebut untuk menentukan apakah terdapat kecacatan. Berdasarkan hasil klasifikasi ini, lengan robot akan secara otomatis menyortir kontainer ke lokasi yang telah ditentukan, yaitu ke "Area Cacat" jika teridentifikasi cacat atau ke "Area Normal" jika sebaliknya. Terakhir, informasi mengenai hasil penyortiran dikirimkan ke *web server* untuk keperluan pemantauan. Perancangan bagan alir sistem kerja alat secara keseluruhan ditunjukkan pada Gambar 6.



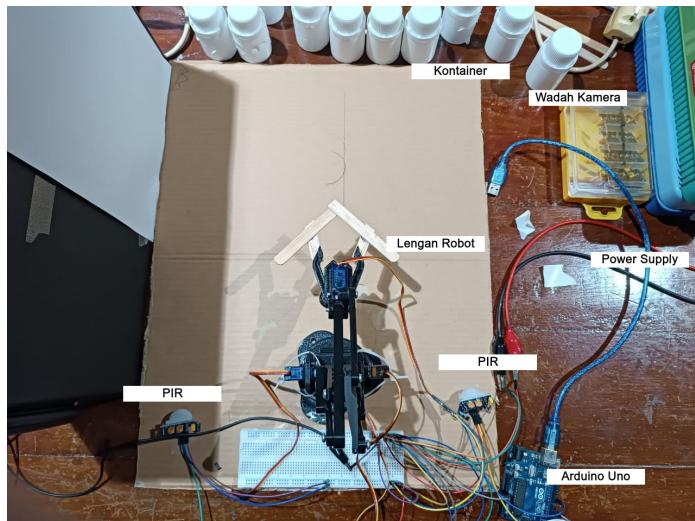
Gambar 6. Bagan alir sistem kerja alat

BAB 3

HASIL DAN PEMBAHASAN

3.1 Hasil Perancangan Sistem

Sistem deteksi kecacatan kontainer ini dirancang secara otomatis menggunakan kamera sebagai sensor utama dan lengan robot sebagai aktuator. Proses diawali dengan model YOLO yang mengidentifikasi keberadaan kontainer dari *input* kamera. Jika kontainer terdeteksi, citra akan dianalisis oleh model *autoencoder* untuk mengklasifikasikan adanya kecacatan. Berdasarkan hasil klasifikasi tersebut, lengan robot yang digerakkan *servo* akan menyortir kontainer ke dalam kategori cacat atau non-cacat. Secara terpisah, sensor PIR berfungsi menghitung jumlah total kontainer pada setiap kategori, di mana data tersebut kemudian dikirim (POST) ke *web-server* untuk visualisasi pada sisi klien melalui *website*. Desain sistem yang dirancang diilustrasikan pada Gambar 7.



Gambar 7. Hasil perancangan sistem

3.2 Hasil Perancangan Model YOLO

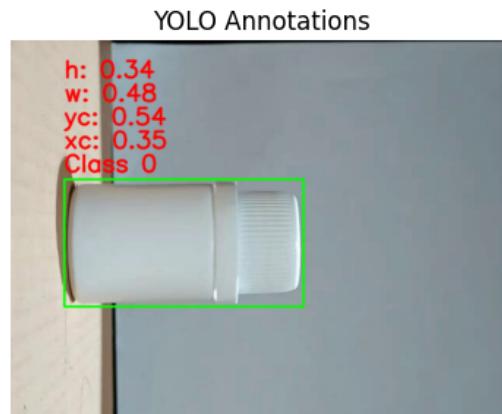
3.2.1 Dataset dan *Preprocessing*

Tahap awal dalam penelitian ini adalah pengumpulan data primer berupa citra kontainer menggunakan kamera. Proses akuisisi data dilakukan untuk membangun sebuah dataset kustom yang merepresentasikan objek target secara akurat. Total gambar mentah yang berhasil dikumpulkan adalah 463 citra. Pengambilan gambar dilakukan dengan melakukan variasi terhadap lokasi kontainer untuk memastikan model yang akan dilatih nantinya mampu mengenali objek dengan baik. Dataset kemudian dibagi menjadi 395 gambar untuk dataset latih dan 68 dataset validasi. Distribusi pembagian dataset disajikan pada Tabel 1.

Tabel 1. Distribusi pembagian dataset

Kategori	Jumlah Gambar	Percentase
Data Latih	395	85,3%
Data Validasi	68	14,7%
Total Data	463	100%

Setelah tahap akuisisi data, proses selanjutnya adalah anotasi gambar. Anotasi merupakan proses fundamental untuk menghasilkan dataset berlabel (*labeled dataset*) yang akan digunakan sebagai data pelatihan (*training data*) (Ragab et al., 2024). Dalam konteks penelitian ini, anotasi dilakukan dengan memberikan *bounding box* serta label kelas pada setiap objek di dalam gambar. Metode ini sangat penting karena arsitektur YOLO dirancang untuk memprediksi *bounding box* dan probabilitas kelas secara bersamaan, sehingga memerlukan data latih dengan format spesifik tersebut untuk dapat melakukan deteksi objek secara *real-time* (Hussain, 2024). Sebanyak 463 gambar kontainer telah dianotasi secara manual. Contoh hasil anotasi dapat dilihat pada Gambar 8.

**Gambar 8.** Salah satu data *train* dengan labelnya

3.2.2 Hasil Pelatihan Model

Pada penelitian ini, *mean Average Precision* (mAP) digunakan sebagai metrik utama untuk mengukur performa model YOLO dalam mendeteksi kontainer kimia. Akurasi deteksi ini sangat penting agar lengan robot dapat melakukan inspeksi kecacatan secara tepat. Metrik ini dihitung berdasarkan *Average Precision* (AP), yang merupakan kombinasi dari *precision* dan *recall* untuk setiap kelas secara individual. Nilai mAP sendiri adalah rata-rata AP untuk semua kelas, sehingga mampu memberikan gambaran performa model yang komprehensif (Padilla et al., 2021).

Untuk menentukan apakah sebuah prediksi dianggap benar (*True Positive*) atau salah (*False Positive*), digunakanlah metrik *Intersection over Union* (IoU). IoU merupakan rasio antara area tumpang tindih (*overlap*) dari *bounding box* hasil prediksi ($BB_{predict}$) dengan *bounding box ground truth* (BB_{ground}), dengan nilai berkisar antara 0 hingga 1 (Kaur dan Singh, 2023). Semakin mendekati 1, berarti prediksi semakin akurat dan sesuai dengan objek sebenarnya. IoU dihitung menggunakan persamaan:

$$\text{IoU} = \frac{|BB_{predict} \cap BB_{ground}|}{|BB_{predict} \cup BB_{ground}|} \quad (1)$$

Precision mengukur proporsi prediksi positif yang benar (*True Positive*) terhadap seluruh prediksi positif ($TP + False Positive$), sehingga menunjukkan kemampuan model dalam meminimalkan kesalahan deteksi. Sementara itu, *recall* atau *true positive rate* mengukur seberapa banyak *instance* positif sebenarnya yang berhasil dideteksi dengan benar, sehingga menggambarkan kemampuan model dalam menemukan semua objek yang ada (Casas et al., 2023). *Precision* dan *recall* dapat dihitung menggunakan persamaan:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

mAP adalah nilai rata-rata dari AP untuk semua kelas. Karena model YOLO yang digunakan dalam penelitian ini hanya memprediksi satu kelas yaitu kontainer kimia, maka nilai mAP setara dengan nilai AP. Nilai AP sendiri dihitung sebagai rata-rata *precision* di seluruh rentang nilai *recall* (0 hingga 1) (Cao et al., 2024), sebagaimana dirumuskan dalam Persamaan 3, di mana P adalah *precision* dan r adalah *recall*.

$$AP = \int_0^1 P(r) dr \quad (3)$$

Proses pelatihan model YOLO dilakukan menggunakan *platform Google Colaboratory*. Model dilatih secara iteratif selama 50 *epoch* dengan *batch size* 16 dan *optimizer* Adam, agar model dapat mempelajari fitur-fitur dari data latih secara optimal. Hasil pelatihan model YOLO setiap 10 *epoch* dapat dilihat pada Tabel 2.

Tabel 2. Proses *training* model YOLO

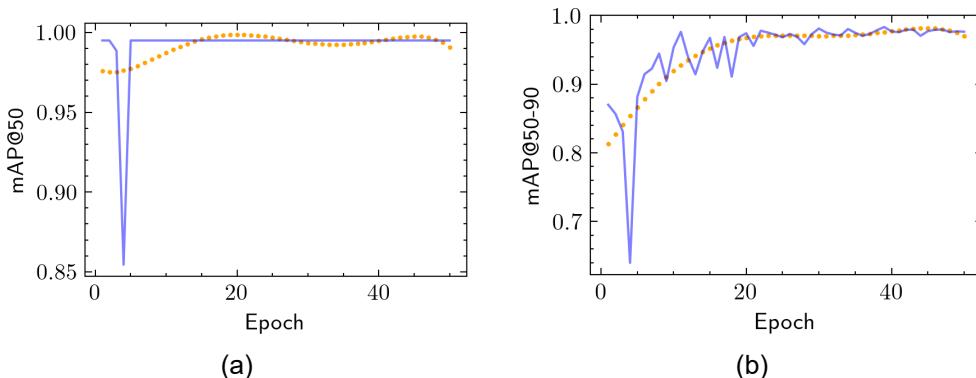
Epoch	Train/Box Loss	Train/Class Loss	Train/DFL Loss	Val/Box Loss	Val/Class Loss	Val/DFL Loss
0	0.59460	1.93744	0.97064	0.47086	2.50286	0.89899
10	0.43332	0.43561	0.87133	0.30517	0.37575	0.82624
20	0.32857	0.29349	0.85098	0.33417	0.21048	0.84483
30	0.28212	0.23284	0.83904	0.24181	0.15662	0.82436
40	0.22672	0.17762	0.80096	0.22091	0.13860	0.82009
50	0.20164	0.15458	0.79927	0.20030	0.11625	0.81561

Pada proses *training*, nilai *loss* pada data *train* dan validasi secara

umum menunjukkan tren penurunan seiring bertambahnya *epoch*. Penurunan ini menunjukkan bahwa model semakin mampu menyesuaikan parameter internalnya dengan data yang diberikan. Dengan demikian, model menjadi lebih baik dalam mempelajari pola yang relevan untuk mendeteksi kontainer.

Pada *Train/Box Loss*, terjadi penurunan signifikan dari 0.59460 pada *epoch* ke-0 menjadi 0.20164 pada *epoch* ke-50, yang mengindikasikan peningkatan kemampuan model dalam memprediksi posisi *bounding box* secara lebih akurat. Hal serupa juga terlihat pada *Train/Class Loss*, yang turun drastis dari 1.93744 menjadi 0.15458, menandakan model semakin tepat dalam melakukan klasifikasi objek. Sementara itu, *Train/DFL Loss (Distribution Focal Loss)* menurun dari 0.97064 menjadi 0.79927, yang berarti model semakin baik dalam memperbaiki distribusi prediksi *bounding box*. Pada data validasi, *Val/Box Loss* menurun dari 0.47086 ke 0.20030, *Val/Class Loss* dari 2.50286 ke 0.11625, serta *Val/DFL Loss* dari 0.89899 ke 0.81561. Penurunan yang konsisten pada semua komponen *loss* validasi menunjukkan bahwa model tidak hanya mampu belajar dengan baik pada data *train*, tetapi juga memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

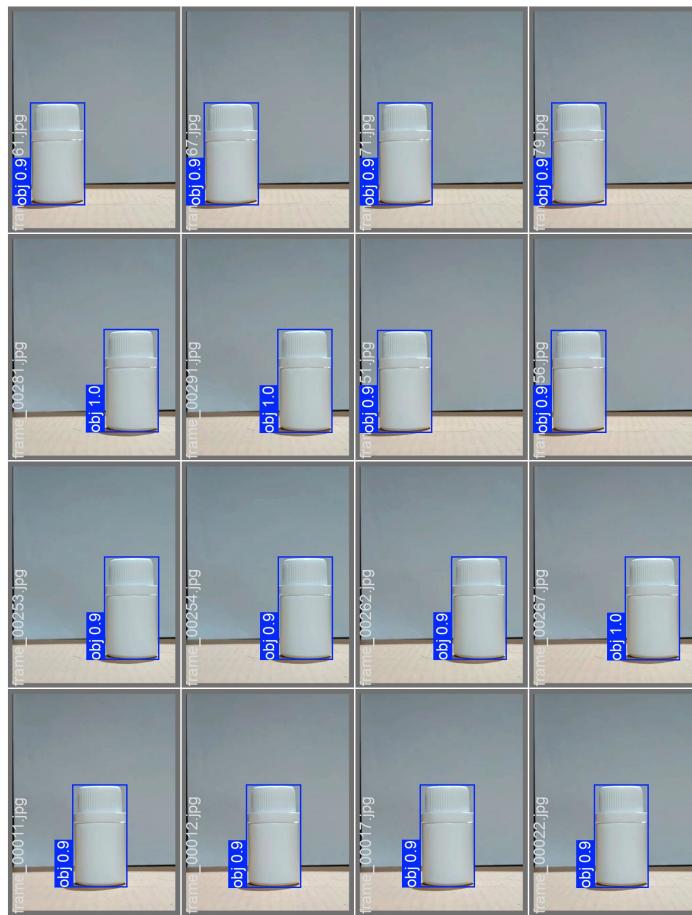
Namun, evaluasi kinerja model tidak bisa hanya mengandalkan nilai *loss*. Perlu digunakan metrik tambahan seperti mAP untuk menilai akurasi deteksi dan kualitas prediksi *bounding box* secara menyeluruh. Pada penelitian ini digunakan mAP@50 (*IoU threshold* 50%) untuk mengukur keberhasilan deteksi secara lebih longgar, serta mAP@95 (rata-rata mAP dari *IoU threshold* 50% hingga 95% dengan interval 5%) untuk menilai ketepatan prediksi *bounding box* secara lebih ketat dan detail. Hasil pengukuran mAP pada tiap *epoch* dapat dilihat pada Gambar 9.



Gambar 9. Grafik tren mAP: (a) mAP@50, (b) mAP@50-90

Grafik tersebut menyajikan evaluasi performa model deteksi objek menggunakan metrik mAP seiring berjalananya 50 *epoch* pelatihan. Grafik di sebelah kiri, mAP@50, yang menggunakan ambang batas IoU longgar (50%), menunjukkan bahwa model dengan sangat cepat mencapai performa puncak. Terlihat bahwa nilainya melonjak mendekati 1.00 hanya dalam 10-15 *epoch* pertama

lalu cenderung datar, menandakan model mampu mempelajari cara mendeteksi keberadaan objek secara umum dengan sangat cepat. Sebaliknya, grafik di sebelah kanan, mAP@50-95, yang mengukur performa pada rentang ambang batas IoU yang lebih ketat (50% hingga 95%), menunjukkan kurva pembelajaran yang lebih bertahap dan konsisten. Peningkatan yang lebih landai ini mengindikasikan bahwa model menggunakan mayoritas waktu pelatihannya untuk terus-menerus menyempurnakan presisi dan akurasi lokalisasi bounding box-nya. Secara keseluruhan, pencapaian nilai mAP yang sangat tinggi pada kedua metrik di akhir pelatihan menandakan model final yang dihasilkan tidak hanya mampu mendeteksi objek, tetapi juga sangat akurat dalam menentukan batas-batasnya. Hasil deteksi kontainer kimia pada data validasi dapat dilihat pada Gambar 10.

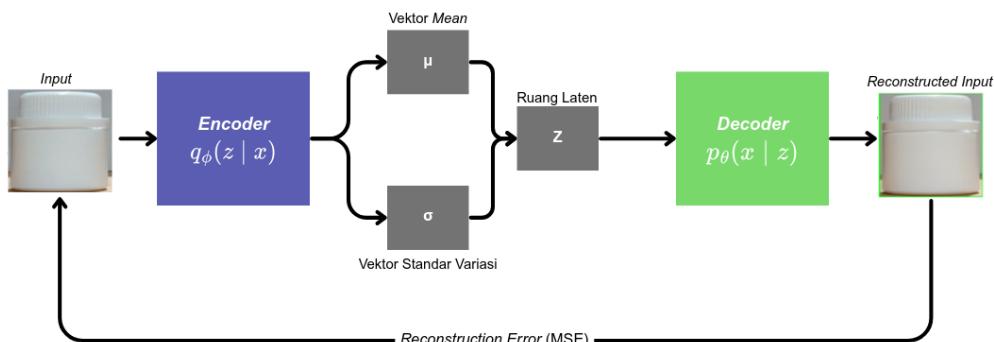


Gambar 10. Hasil prediksi YOLO pada data validasi

3.3 Hasil Perancangan Model Deteksi Kecacatan

3.3.1 Arsitektur Variational Autoencoder

Variational autoencoder (VAE) merupakan pengembangan dari *autoencoder*, yaitu sebuah arsitektur *neural network* yang berfungsi untuk mengekstrak fitur laten dari data. Perbedaan fundamental VAE dibandingkan pendahulunya terletak pada pendekatan Bayesian, di mana vektor laten (z) dipaksa mengikuti distribusi probabilitas terstruktur, umumnya distribusi Gaussian. Secara operasional, *encoder* VAE ($q_\phi(z | x)$) tidak hanya mengompres data *input* (x), melainkan juga memetakannya ke dalam parameter statistik: sebuah vektor *mean* dan vektor varians. Selanjutnya, sebuah sampel kode laten (z) diambil dari distribusi yang diwakili oleh kedua parameter ini, lalu diumpulkan ke *decoder* ($p_\theta(x | z)$) untuk merekonstruksi data *input* asli (Mansour et al., 2021). Arsitektur *variational autoencoder* yang digunakan pada penelitian ini dapat dilihat pada Gambar 11.



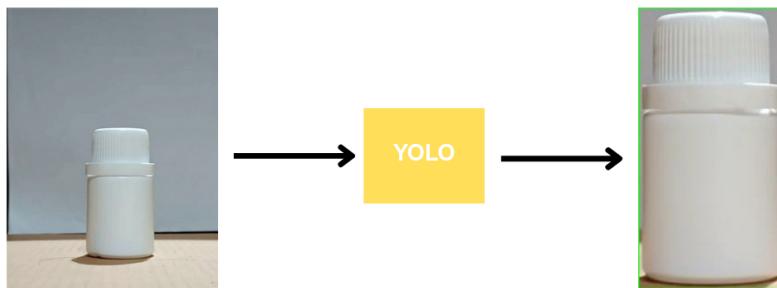
Gambar 11. Hasil prediksi YOLO pada data validasi

Arsitektur VAE yang diimplementasikan pada penelitian ini dirancang untuk memproses citra berukuran 128×128 dengan tiga kanal warna (RGB). Bagian *encoder* menggunakan lima lapis *convolutional* berurutan, masing-masing diikuti fungsi aktivasi ReLU, untuk secara progresif mengekstrak dan menurunkan resolusi fitur spasial, hingga menghasilkan representasi akhir berukuran (512, 4, 4). Representasi ini kemudian diproyeksikan menjadi dua vektor, yaitu vektor *mean* dan vektor *log-varians*, yang mendefinisikan distribusi Gaussian multivariat sebagai ruang laten. Setelah sampling vektor laten z , *decoder* memproyeksikan kembali z ke dimensi spasial melalui lapisan linear dan serangkaian lapisan *transpose convolutional* (dekonvolusi) yang secara bertahap memperbesar resolusi hingga kembali ke ukuran asli citra *input*. Fungsi aktivasi sigmoid pada lapisan output memastikan nilai piksel berada pada rentang [0, 1], sehingga hasil rekonstruksi dapat diinterpretasikan sebagai citra valid.

3.3.2 Dataset dan Preprocessing

Tahap ini adalah pengumpulan data primer berupa citra kontainer menggunakan kamera. Proses akuisisi data ini dilakukan untuk membangun

dataset kustom yang secara spesifik merepresentasikan objek target. Total gambar mentah yang berhasil dikumpulkan berjumlah 3005 citra, yang seluruhnya digunakan sebagai dataset latih. Sebelum digunakan untuk pelatihan *autoencoder*, seluruh citra diproses menggunakan model YOLO untuk mendeteksi serta melakukan *cropping* area kontainer secara otomatis. Dengan demikian, hanya bagian kontainer yang relevan yang masuk ke pipeline pemrosesan, sehingga model *autoencoder* dapat lebih fokus dalam mempelajari fitur visual kontainer secara mendalam. Contoh salah satu data dapat dilihat pada Gambar 12.



Gambar 12. Hasil prediksi YOLO pada data validasi

Karena *autoencoder* termasuk dalam kategori *unsupervised learning*, data yang digunakan pada penelitian ini tidak memerlukan label atau anotasi manual. Seluruh dataset hanya terdiri dari citra kontainer yang dalam kondisi baik (bebas kerusakan atau cacat visual), sehingga model *autoencoder* dapat belajar merepresentasikan distribusi citra kontainer normal secara optimal. Dengan demikian, *autoencoder* akan berfokus pada rekonstruksi citra input semirip mungkin, dan nantinya dapat digunakan untuk mendeteksi anomali atau kerusakan berdasarkan perbedaan signifikan antara citra asli dan hasil rekonstruksi.

3.3.3 Hasil Pelatihan Model

Fungsi *loss* VAE didasarkan pada *Evidence Lower Bound* (ELBO). Tujuannya adalah memaksimalkan *bound* ini agar dapat mengatasi masalah distribusi posterior yang tidak dapat dihitung secara langsung pada model probabilistik. Fungsi *loss* terdiri dari dua komponen utama. Komponen pertama adalah *reconstruction loss*, yang merupakan ekspektasi dari error rekonstruksi negatif. Komponen ini mengukur seberapa baik *decoder* probabilistik, $p_\theta(x|z)$, dapat merekonstruksi data *input* x dari representasi laten z . Komponen kedua adalah *regularization term*, yang dihitung sebagai Kullback-Leibler (KL) *divergence* antara distribusi posterior aproksimasi $q_\phi(z|x)$ dan distribusi prior $p_\theta(z)$. Komponen ini berfungsi sebagai penalti, supaya distribusi yang dihasilkan *encoder* tetap mendekati distribusi prior sederhana seperti distribusi Gaussian (Kingma dan Welling, 2013). Secara matematis, fungsi *loss* VAE dapat dituliskan sebagai:

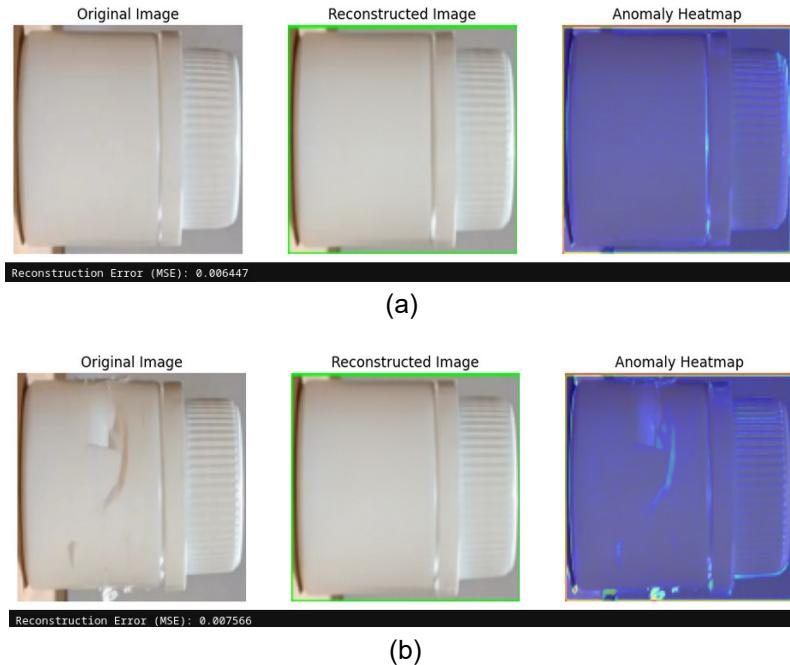
$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{\mathbf{q}(\phi)(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] - D_{KL}(\mathbf{q}_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z})) \quad (4)$$

Proses pelatihan model CVAE dilakukan menggunakan komputer dengan GPU NVIDIA GeForce RTX 3050 8GB VRAM untuk memanfaatkan kemampuan komputasi paralel CUDA. Model dilatih secara iteratif selama 100 *epoch* dengan *batch size* 64 dan *optimizer* Adam, agar model dapat mempelajari fitur-fitur dari data latih secara optimal. Hasil pelatihan model YOLO setiap 10 *epoch* dapat dilihat pada Tabel 3.

Tabel 3. Proses training model *convolutional variational autoencoder* (CVAE)

Epoch	Loss
1	837.7615
10	30.5440
20	22.6941
30	18.5220
40	15.0172
50	11.9098
60	9.8593
70	8.7772
80	7.3923
90	7.1070
100	6.8338

Pada proses *training*, nilai *loss* rekonstruksi secara umum menunjukkan tren penurunan seiring bertambahnya *epoch*. Penurunan ini mengindikasikan bahwa model *autoencoder* semakin mampu mempelajari representasi fitur penting dari data normal yang diberikan. Dengan demikian, model menjadi lebih baik dalam merekonstruksi kontainer kimia tanpa cacat, sehingga mempermudah deteksi perbedaan (anomali) ketika diaplikasikan pada data cacat. Hasil prediksi kecacatan *training* dapat dilihat pada Gambar 13.



Gambar 13. Hasil prediksi kecacatan: (a) Kontainer normal, (b) kontainer cacat

Gambar di atas menunjukkan rekonstruksi *error* model *autoencoder* pada kontainer normal dan cacat. Pada kontainer normal, *error* yang diperoleh adalah 0.006447 sedangkan pada kontainer cacat sebesar 0.007566. Dari hasil tersebut terlihat *error* pada kontainer cacat lebih besar dibandingkan *error* pada kontainer normal. Hal ini menunjukkan bahwa model sudah dapat membedakan kontainer normal dan cacat.

3.3.4 Penentuan *Threshold* Kecacatan

Penentuan *threshold* sangat penting agar sistem deteksi cacat dapat membedakan secara akurat antara kontainer kimia yang cacat dan yang normal. Tanpa *threshold* yang tepat, model berpotensi menghasilkan banyak kesalahan prediksi, sehingga mengurangi keandalan sistem. Dalam penelitian ini, metrik *error* yang digunakan untuk mengukur perbedaan antara citra asli dan hasil rekonstruksi adalah *Mean Squared Error* (MSE).

MSE merupakan metrik yang paling umum digunakan untuk mengukur kualitas citra secara kuantitatif. MSE menghitung rata-rata selisih kuadrat antara piksel citra asli (*input*) dengan piksel citra hasil rekonstruksi (*output*). Semakin kecil nilai MSE, semakin mirip citra hasil rekonstruksi dengan citra aslinya, yang berarti model berhasil meminimalkan distorsi (Najjar, 2024). MSE dirumuskan sebagai berikut:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (5)$$

Pengujian dilakukan pada 20 gambar kontainer kimia, terdiri dari 10 gambar kontainer normal dan 10 gambar kontainer cacat. Dari pengujian ini diperoleh distribusi nilai MSE masing-masing gambar untuk dianalisis lebih lanjut. Hasil perhitungan MSE pada 20 gambar uji tersebut dirangkum pada Tabel 4.

Tabel 4. Nilai *error* untuk setiap sampel

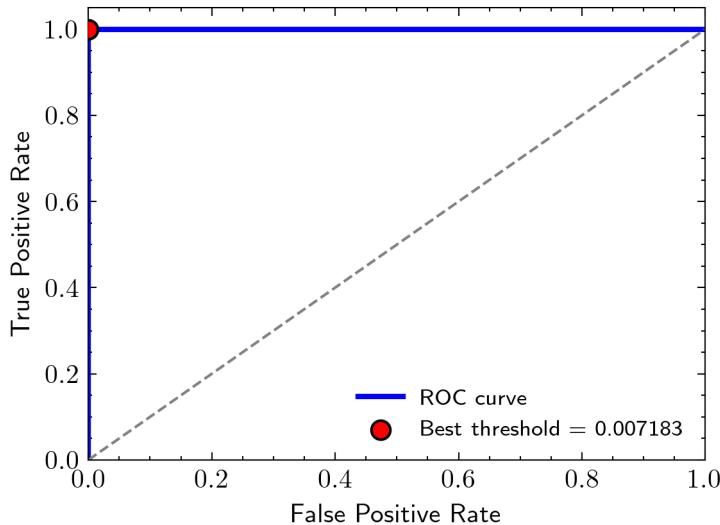
Index	Error	Kategori
1	0.006737	Normal
2	0.006828	Normal
3	0.007091	Normal
4	0.006997	Normal
5	0.006610	Normal
6	0.006795	Normal
7	0.006872	Normal
8	0.006725	Normal
9	0.006873	Normal
10	0.006817	Normal
11	0.007432	Cacat
12	0.007455	Cacat
13	0.008369	Cacat
14	0.007183	Cacat
15	0.007653	Cacat
16	0.008856	Cacat
17	0.007531	Cacat
18	0.007624	Cacat
19	0.007457	Cacat
20	0.008424	Cacat

Tabel 4 menunjukkan nilai *error* atau selisih rekonstruksi pada masing-masing sampel gambar, yang terdiri dari 10 kontainer normal dan 10 kontainer cacat. Terlihat bahwa sampel dengan kategori normal memiliki nilai error yang relatif kecil dan cukup seragam di sekitar 0.0067, menandakan model mampu merekonstruksi dengan baik. Sebaliknya, sampel dengan kategori cacat cenderung memiliki error lebih tinggi, menunjukkan adanya perbedaan yang signifikan pada area cacat yang gagal direkonstruksi dengan sempurna, sehingga mempermudah deteksi anomali.

Kurva *Receiver Operating Characteristic* (ROC) digunakan sebagai metode analisis untuk mengevaluasi performa model deteksi cacat pada kontainer kimia dalam penelitian ini. Kurva ROC dibuat dengan memplot nilai $1 - specificity$ (*false positive rate*) pada sumbu-x dan *recall* (*true positive rate*) pada sumbu-y untuk setiap nilai (*threshold*) yang diuji. *Specificity* sendiri adalah ukuran seberapa baik model dalam mengenali data negatif secara benar. Nilai *threshold* terbaik ditentukan dengan memaksimalkan statistik Youden J (Nahm, 2022). Kurva ROC dapat dilihat

pada Gambar 14. Persamaan statistik Youden J dituliskan sebagai berikut:

$$J = \text{True Positive Rate} - \text{False Positive Rate} \quad (6)$$

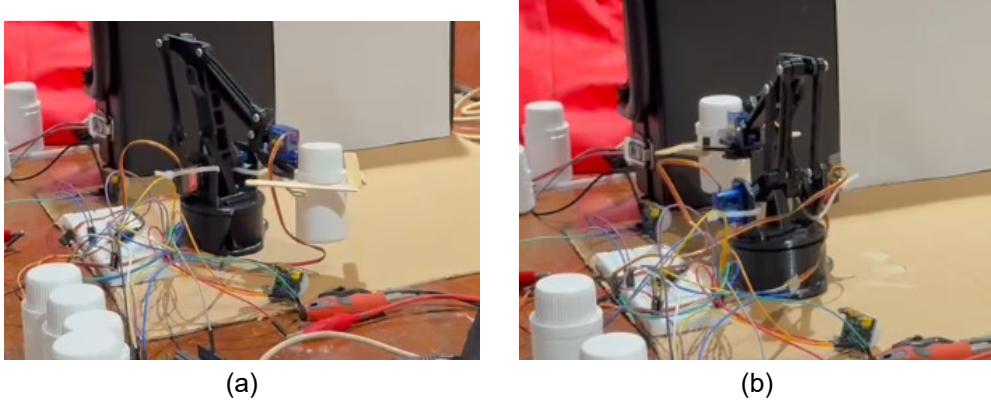


Gambar 14. Kurva ROC

Kurva di atas menunjukkan performa model dalam membedakan kontainer cacat dan tidak cacat dengan sangat baik, ditunjukkan oleh garis biru yang mendekati titik sudut kiri atas. Berdasarkan perhitungan, *threshold* optimal yang diperoleh adalah sebesar 0.007183, ditandai dengan lingkaran merah pada grafik. Nilai ini digunakan sebagai batas deteksi akhir, sehingga model dapat memaksimalkan sensitivitas sekaligus meminimalkan kesalahan positif dalam identifikasi cacat.

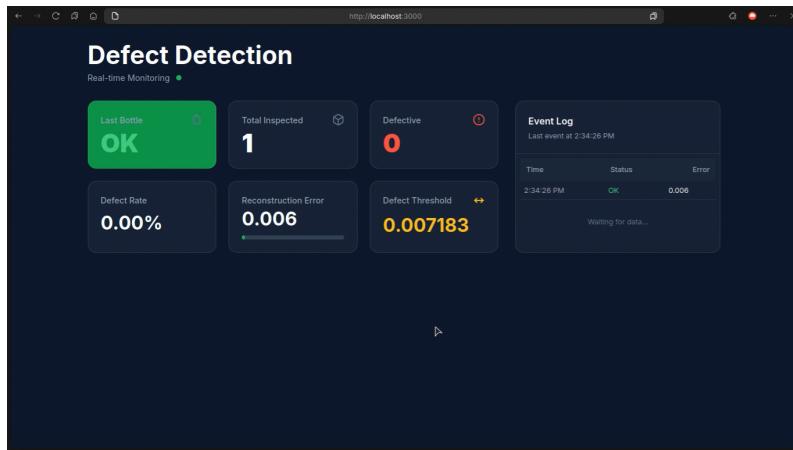
3.4 Hasil Pengujian Sistem Lengan Robot

Setelah model deteksi terlatih dan nilai *threshold* optimal ditentukan, tahap selanjutnya adalah integrasi sistem dengan lengan robot yang dikontrol oleh mikrokontroler Arduino. Berdasarkan hasil klasifikasi dari model, lengan robot akan secara otomatis memindahkan kontainer kimia. Objek akan diarahkan ke sebelah kiri jika sistem mengidentifikasinya sebagai cacat, dan ke sebelah kanan jika dinyatakan normal. Gambar 15 mengilustrasikan aksi lengan robot ketika sedang menyortir objek yang teridentifikasi cacat.

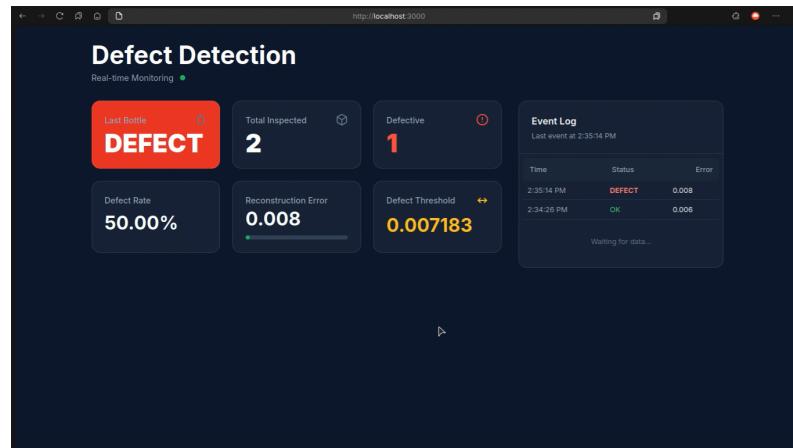


Gambar 15. Lengan robot menyortir kontainer: (a) normal, (b) cacat

Setelah lengan robot melepaskan kontainer di area penyortiran yang telah ditentukan (kiri untuk cacat, kanan untuk normal), sebuah sensor PIR yang terpasang di lokasi tersebut akan mendeteksi keberadaan objek. Deteksi dari sensor PIR ini berfungsi sebagai sinyal konfirmasi bahwa satu siklus penyortiran telah selesai. Sinyal ini kemudian memicu pengiriman data hasil klasifikasi, yang mencakup citra kontainer beserta label statusnya, ke server pusat. Untuk menampilkan informasi ini secara dinamis dan instan kepada klien, sistem memanfaatkan protokol *WebSocket*. Tampilan website dan hasil prediksi dapat dilihat pada Gambar 16.



(a)



(b)

Gambar 16. Tampilan hasil prediksi melalui website: (a) kontainer normal, (b) kontainer cacat

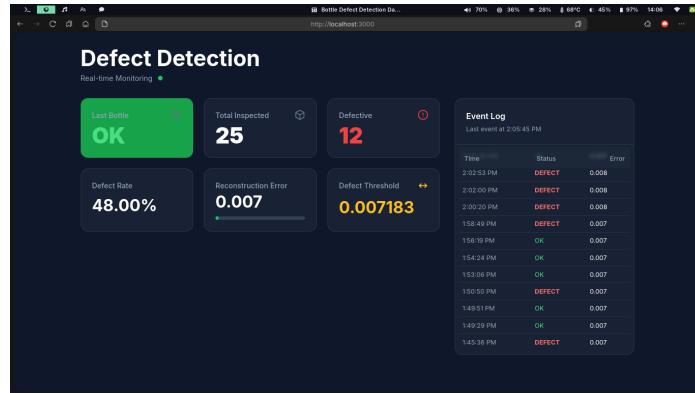
Untuk menguji keandalan dan konsistensi performa model, dilakukan serangkaian pengujian sebanyak 25 kali pada objek kontainer. Setiap pengujian menggunakan sampel dengan kondisi yang bervariasi antara normal dan cacat untuk mengevaluasi kemampuan generalisasi model. Hasil kuantitatif dari setiap pengujian ini dirangkum secara rinci pada Tabel 5.

Tabel 5. Nilai *error* untuk setiap sampel uji

No	Error	Kategori (Threshold < 0.007183)	Label Asli
1	0.006798	Normal	Normal
2	0.008298	Cacat	Cacat
3	0.006854	Normal	Normal
4	0.007066	Normal	Normal
5	0.006923	Normal	Normal
6	0.007607	Cacat	Cacat
7	0.007261	Cacat	Cacat
8	0.007393	Cacat	Cacat
9	0.006842	Normal	Normal
10	0.008099	Cacat	Cacat
11	0.006945	Normal	Normal
12	0.007381	Cacat	Cacat
13	0.006722	Normal	Normal
14	0.006865	Normal	Normal
15	0.007397	Cacat	Cacat
16	0.007099	Normal	Normal
17	0.006790	Normal	Normal
18	0.006760	Normal	Normal
19	0.007383	Cacat	Cacat
20	0.007693	Cacat	Cacat
21	0.008071	Cacat	Cacat
22	0.007977	Cacat	Cacat
23	0.006885	Normal	Normal
24	0.008007	Cacat	Cacat
25	0.006847	Normal	Normal
Tingkat Akurasi			100%

Tabel 5 menyajikan hasil pengujian performa model deteksi kecacatan pada 25 sampel kontainer yang berbeda untuk memvalidasi efektivitasnya. Keputusan klasifikasi untuk setiap sampel didasarkan pada perbandingan nilai *error* rekonstruksi terhadap *threshold* yang telah ditentukan sebesar 0.007183. Sesuai dengan mekanisme ini, objek dengan nilai *error* di bawah *threshold* dikategorikan sebagai "Normal", sementara objek dengan nilai *error* yang melebihinya dikategorikan sebagai "Cacat". Hasil pada tabel secara jelas menunjukkan bahwa untuk keseluruhan 25 data uji, kolom "Kategori" (label prediksi dari model) sepenuhnya konsisten dengan kolom "Label Asli" (*ground truth*). Sebagai contoh, pada data no. 2, nilai *error* yang tinggi (0.008298) berhasil diidentifikasi dengan benar sebagai "Cacat", sementara pada data no. 1, nilai *error* yang rendah (0.006798) juga tepat diklasifikasikan sebagai "Normal". Kesesuaian antara prediksi dan kondisi sebenarnya pada seluruh rangkaian pengujian ini membuktikan bahwa model, dengan *threshold* yang dipilih, mampu mencapai tingkat akurasi 100% dalam membedakan antara kontainer normal dan cacat. Gambar 17 menampilkan hasil website setelah melakukan prediksi

terhadap 25 sampel yang juga menunjukkan hasil yang sama dengan Tabel 5.



Gambar 17. Hasil prediksi kecacatan pada 25 sampel

BAB 4

KESIMPULAN

4.1 Kesimpulan

Berdasarkan hasil perancangan, analisis, dan implementasi alat yang telah digunakan, maka dapat diambil kesimpulan sebagai berikut:

1. Kesimpulan 1
2. Kesimpulan 2

4.2 Saran

Alat yang telah dibuat memiliki potensi untuk dikembangkan lebih lanjut. Berikut adalah beberapa saran untuk penelitian selanjutnya:

1. Saran 1
2. Saran 2

DAFTAR PUSTAKA

- Jhang, J. Y. & Lin, C. J. 2024. Jhang, Jyun-Yu, & Cheng-Jian Lin. Optimizing parameters of YOLO model through uniform experimental design for gripping tasks performed by an internet of things-based robotic arm. *Internet of Things* 27, 1-12. doi: 10.1016/j.iot.2024.101332.
- Oaki, J., Sugiyama, N., Ishihara, Y., Ooga, J., Kano, H. & Ohno, H., 2023. Micro-Defect Inspection on Curved Surface Using a 6-DOF Robot Arm with One-Shot BRDF Imaging. *IFAC-PapersOnLine* 56(2), 9354-9359. doi: 10.1016/j.ifacol.2023.10.224.
- Lin, C. J., Jhang, J. Y., Gao, Y. J. & Huang, H. M., 2024. Vision-based Robotic Arm in Defect Detection and Object Classification Applications. *Sensors & Materials* 36(2), 655-670. doi: 10.18494/SAM4683.
- Truong, A. M. & Luong, H. Q., 2024. A non-destructive, autoencoder-based approach to detecting defects and contamination in reusable food packaging. *Current Research in Food Science* 8, 1-12. doi: 10.1016/j.crf.2024.100758.
- Graetz, G. & Michaels, G., 2018. Robots at work. *Review of economics and statistics* 100(5), 753-768. doi: 10.1162/rest_a_00754.
- Gihleb, R., Giuntella, O., Stella, L. & Wang, T., 2022. Industrial robots, workers' safety, and health. *Labour economics* 78, 1-12. doi: 10.1016/j.labeco.2022.102205.
- Tsai, D. M. & Jen, P. H. 2021., Autoencoder-based anomaly detection for surface defect inspection. *Advanced Engineering Informatics* 48, 1-12. doi: 10.1016/j.aei.2021.101272.
- Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z. & Shao, L., 2021. Surface defect detection methods for industrial products: A review. *Applied Sciences* 11(16), 1-25. doi: 10.3390/app11167657.
- Liu, Y., Qiu, W., Fu, K., Chen, X., Wu, L. & Sun, M., 2025. An improved YOLOv8 model and mask convolutional autoencoder for multi-scale defect detection of ceramic tiles. *Measurement* 248, 1-11. doi: 10.1016/j.measurement.2025.116847.
- Wang, L., 2024. Robot arm grasping based on YOLOv5 in the perspective of automated production. *Engineering Research Express*, 6(4), 1-12. doi:10.1088/2631-8695/ad88dc.
- Kato, H., Nagata, F., Murakami, Y. & Koya, K., 2022. Partial depth estimation with single image using YOLO and CNN for robot arm control. *IEEE International Conference on Mechatronics and Automation (ICMA)*, 1727-1731. IEEE. doi: 10.1109/ICMA54519.2022.9856055.

- Kim, M. & Kim, S. 2021., YOLO-based robotic grasping. International Conference on Control, Automation and Systems (ICCAS) 21, 1120-1122. doi: 10.23919/ICCAS52745.2021.9649837.
- Jin, T., Han, X., Wang, P., Zhang, Z., Guo, J. & Ding, F., 2025. Enhanced deep learning model for apple detection, localization, and counting in complex orchards for robotic arm-based harvesting. Smart Agricultural Technology 10, 1-25. doi: 10.1016/j.atech.2025.100784.
- Bionda, A., Frittoli, L. & Boracchi, G., 2022. Deep autoencoders for anomaly detection in textured images using CW-SSIM. International Conference on Image Analysis and Processing, 669-680. doi: 10.1007/978-3-031-06430-2_56.
- Yun, H., Kim, H., Jeong, Y. H. & Jun, M. B., 2023. Autoencoder-based anomaly detection of industrial robot arm using stethoscope based internal sound sensor. Journal of Intelligent Manufacturing 34(3), 1427-1444. doi: 10.1016/j.jmssp.2004.10.013.
- Jia, H. & Liu, W., 2023. Anomaly detection in images with shared autoencoders. Frontiers in Neurorobotics 16, 1-11. doi: 10.3389/fnbot.2022.1046867.
- Kozamernik, N. & Bračun, D., 2025. A novel FuseDecode Autoencoder for industrial visual inspection: Incremental anomaly detection improvement with gradual transition from unsupervised to mixed-supervision learning with reduced human effort. Computers in Industry 164, 1-19. doi: 10.1016/j.compind.2024.104198.
- Ruediger-Flore, P., Klar, M., Hussong, M., Mukherjee, A., Glatt, M. & Aurich, J. C., 2024. Comparing Binary Classification and Autoencoders for Vision-Based Anomaly Detection in Material Flow. Procedia CIRP 121, 138-143. doi: 10.1016/j.procir.2023.09.241.
- Ragab, M. G., Abdulkadir, S. J., Muneer, A., Alqushaibi, A., Sumiea, E. H., Qureshi, R. et al., 2024. A comprehensive systematic review of YOLO for medical object detection (2018 to 2023). IEEE Access 12, 57815-57836. doi: 10.1109/ACCESS.2024.3386826.
- Hussain, M., 2024. Unveiling each variant—a comprehensive review of yolo. IEEE access 12, 42816-42833. doi: 10.1109/ACCESS.2024.3378568.
- Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & Da Silva, E. A., 2021. A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics 10(3), 1-28. doi: 10.3390/electronics10030279.
- Kaur, R., & Singh, S., 2023. A comprehensive review of object detection with deep learning. Digital Signal Processing 132, 1-17. doi: 10.1016/j.dsp.2022.103812.

- Casas, E., Ramos, L., Bendek, E., & Rivas-Echeverría, F., 2023. Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. *IEEE Access* 11, 96554-96583. doi: 10.1109/ACCESS.2023.3312217.
- Cao, L., Shen, Z., & Xu, S., 2024. Efficient forest fire detection based on an improved YOLO model. *Visual Intelligence* 2(20), 1-7. doi: 10.1007/s44267-024-00053-y.
- Mansour, R. F., Escoria-Gutierrez, J., Gamarra, M., Gupta, D., Castillo, O., & Kumar, S. 2021. Unsupervised deep learning based variational autoencoder model for COVID-19 diagnosis and classification. *Pattern Recognition Letters*, 151, 267-274.
- Kingma, D. P., & Welling, M. 2013. Auto-encoding variational bayes.
- Najjar, Y., A., 2024. Comparative analysis of image quality assessment metrics: MSE, PSNR, SSIM and FSIM. *International Journal of Science and Research* 13(3), 110-114. doi: orcid.org/0000-0002-3369-4999.
- Nahm, F. S., 2022. Receiver operating characteristic curve: overview and practical use for clinicians. *Korean journal of anesthesiology* 75(1), 25-36. doi: 10.4097/kja.21209.