

BAB III

METODE PENELITIAN

Bab ini menjelaskan pendekatan dan langkah-langkah yang digunakan dalam pelaksanaan penelitian. Penjelasan mencakup deskripsi umum, jenis penelitian, alur penelitian, perancangan *hardware*, perancangan model, perancangan integrasi sistem *hardware* dengan model, perancangan pengujian sistem dan evaluasi integrasi sistem identifikasi produk cacat.

3.1 Deskripsi Umum

Penelitian ini bertujuan untuk membangun sistem deteksi cacat pada *folding box* menggunakan metode yang sudah ditentukan berdasarkan hasil riset yaitu model YOLOv8m dengan varian model *instance segmentation* (YOLOv8m) berbasis teknologi *deep learning*. Sistem ini dirancang melalui sebuah prototipe yang menyerupai jalur produksi guna merepresentasikan kondisi nyata di lapangan. Proses penelitian diawali dengan studi literatur, penentuan spesifikasi komponen, rancang bangun *hardware*, *dataset*, model YOLOv8m, evaluasi model YOLOv8m, integrasi *hardware* dengan YOLOv8m, pengujian sistem *hardware* dengan YOLOv8m, dan evaluasi integrasi sistem *hardware* dan *model* YOLOv8m.

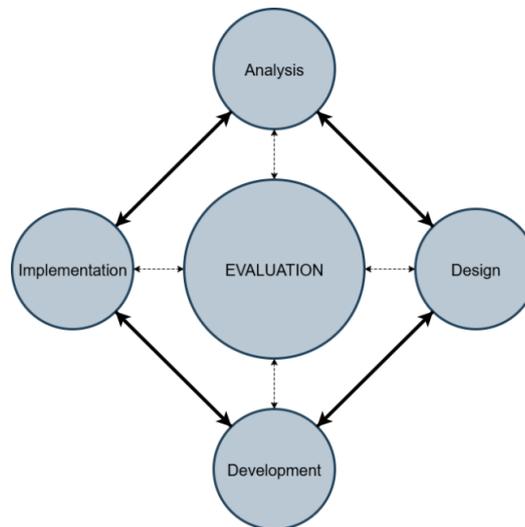
Hasil akhir penelitian ini menyajikan nilai-nilai dari metrik yang digunakan untuk mengevaluasi kinerja model, seperti akurasi (*accuracy*), presisi (*precision*), *recall*, serta efektivitas integrasi sistem YOLOv8m dengan *rejection area*. Diharapkan dapat membuktikan bahwa sistem deteksi cacat berbasis YOLOv8m yang dirancang mampu mendeteksi jenis cacat dengan akurat dan *real-time* serta dapat terintegrasi dengan lingkungan produksi nyata.

3.1.1 Jenis Penelitian

Jenis penelitian yang digunakan yaitu *Research and Development* (R&D). Metode R&D adalah metode penelitian yang bertujuan untuk menghasilkan suatu produk tertentu serta menguji keefektifan produk tersebut secara empiris (Hikmawati, 2020). Pada penelitian ini, metode R&D diterapkan untuk merancang dan membangun sistem identifikasi produk cacat pada *folding box* yang terintegrasi dengan *rejection area* berbasis algoritma YOLOv8. Dengan metode R&D ini

memungkinkan proses penelitian dilakukan secara empiris, sistematis, logis, dan relevan, serta merujuk pada teori yang sudah teruji secara ilmiah.

Penelitian ini menggunakan pendekatan penelitian ADDIE (*Analysis, Design, Development, Implementation, Evaluation*). Pendekatan penelitian ADDIE menawarkan keunggulannya dalam menghasilkan produk inovatif serta efektif yang sistematis dan terstruktur (Safitri dkk., 2019; Waruwu, 2024). Penggunaan pendekatan ADDIE dalam penelitian R&D tepat karena memberikan kerangka kerja sistematis dan terstruktur dalam mengembangkan produk, mulai dari analisis kebutuhan hingga evaluasi hasil. Design pendekatan ADDIE disajikan pada Gambar 3. 1.

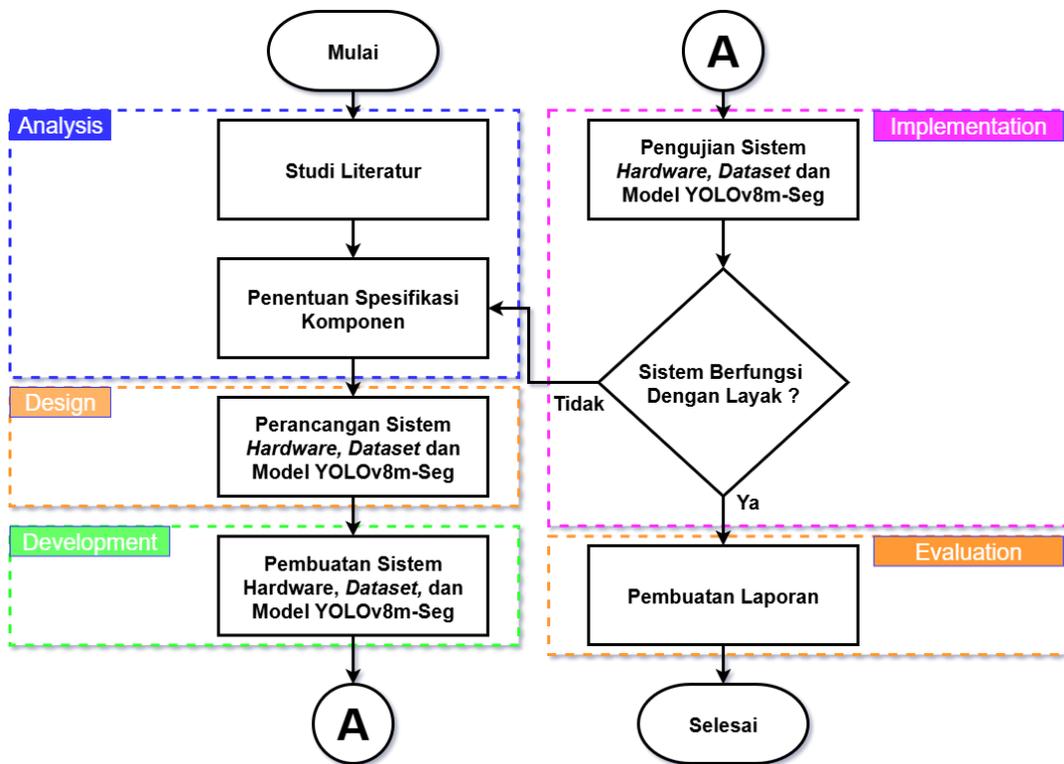


Gambar 3. 1 Desain Penelitian

3.1.2 Alur Penelitian

Alur penelitian mencakup seluruh proses, mulai dari penentuan topik hingga tahap analisis dan evaluasi sebagai bagian akhir dari kegiatan penelitian. Proses ini diawali dengan studi literatur, penentuan spesifikasi komponen, rancang bangun *hardware*, *dataset*, model YOLOv8m, evaluasi model YOLOv8m, integrasi *hardware* dengan YOLOv8m, pengujian sistem *hardware* dengan YOLOv8m dan evaluasi integrasi sistem *hardware* dan *model* YOLOv8m. Dalam penelitian ini digunakan pendekatan ADDIE sebagai kerangka pengembangan yang terdiri dari

tahapan *Analysis, Design, Development, Implementation, dan Evaluation*. Untuk gambaran diagram alur disajikan pada Gambar 3. 2.



Gambar 3. 2 Diagram Alur Penelitian

1. *Analysis*

Tahap analisis merupakan fase awal yang dilakukan untuk mengidentifikasi permasalahan secara menyeluruh sebagai landasan utama dalam perumusan fokus penelitian. Selain itu, analisis ini juga mencakup identifikasi kebutuhan operasional dan atribut sistem, baik dari aspek *hardware, dataset* maupun model deteksi. Analisis ini juga diarahkan untuk mengevaluasi kelayakan teknis, dan operasional sistem. Dalam konteks penelitian ini, proses analisis diawali dengan studi literatur dan kajian mendalam terhadap referensi ilmiah dan penelitian terdahulu yang relevan, guna merumuskan kebutuhan pengembangan sistem.

2. *Design*

Pada tahap ini mencakup proses perancangan *hardware, dataset* dan model deteksi. Perancangan *hardware* mencakup proses penataan tata letak komponen secara efisien, pembagian area berdasarkan fungsinya, serta pemetaan hubungan

antar komponen untuk memastikan integrasi sistem berjalan optimal. Kemudian, perancangan model mencakup proses pelatihan dan langkah-langkah pembuatan model deteksi berbasis algoritma YOLOv8m. Kemudian, perancangan *dataset* mencakup penyesuaian dimensi, warna, dan karakteristik desain yang relevan untuk mendukung kinerja model, perancangan. Sementara itu, perancangan integrasi *hardware* dengan YOLOv8m, perancangan pengujian sistem *hardware* dengan YOLOv8m, dan perancangan evaluasi sistem *hardware* dengan YOLOv8m.

3. *Development*

Tahap *development* dimulai dari proses pembuatan *hardware*, di mana sebuah kamera diintegrasikan dengan laptop untuk pengambilan gambar, setelah itu mikrokontroler diprogram untuk membaca data dari laptop dan menggerakkan aktuator. Kemudian, dilakukan pembuatan model YOLOv8m, dimulai dengan proses pengambilan data, pelabelan data (*annotation*), pemrosesan data (*pre-processing*), dan evaluasi model YOLOv8m. Evaluasi model menggunakan metrik performa seperti akurasi (*accuracy*), presisi (*precision*), *recall*, dan *F1-score*. *Dataset* pelatihan juga diperkaya melalui proses *augmentation data* untuk meningkatkan keragaman data dan performa model dalam kondisi variatif.

4. *Implementation*

Tahap *implementation* merupakan tahap integrasi sistem *hardware* dan model YOLOv8m, kemudian pengujian *hardware* dan model YOLOv8m dari perakitan yang telah disusun sebelumnya. Selanjutnya, evaluasi seluruh perancangan yang telah dibuat diimplementasikan secara menyeluruh menjadi sebuah sistem yang mampu mendeteksi objek menggunakan kamera dan model YOLOv8m. Pada tahap ini, aktuator mulai berfungsi sesuai dengan logika sistem, yaitu merespons hasil deteksi dan *folding box* diarahkan sesuai deteksi cacat. Seluruh komponen, baik *hardware* maupun model deteksi diuji dalam kondisi operasional sebenarnya untuk memastikan fungsi sistem berjalan sesuai dengan spesifikasi dan tujuan awal pengembangan.

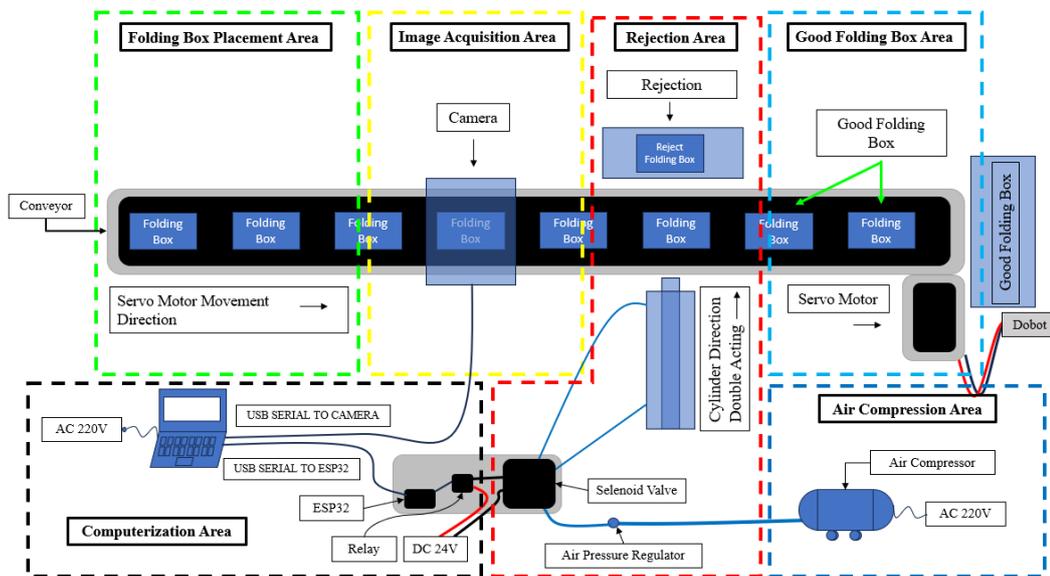
5. *Evaluation*

Pada tahap *evaluation*, dilakukan pengujian sistem untuk menilai sejauh mana model yang dikembangkan mampu berfungsi secara tepat dan sesuai dengan tujuan

penelitian, yaitu mendeteksi produk cacat menggunakan algoritma YOLOv8m serta secara otomatis mengeluarkan *folding box* yang cacat dari jalur *conveyor*. Evaluasi ini mencakup pengujian fungsionalitas serta efektivitas dalam pengambilan keputusan terhadap objek yang teridentifikasi cacat. Metode pengujian fungsionalitas menggunakan metode *blackbox testing* dan pengujian efektivitas menggunakan metode FAR (*false acceptance rate*) dan FRR (*false rejection rate*). Nilai-nilai dari metrik tersebut digunakan untuk memastikan bahwa sistem mampu bekerja secara *real-time* dan akurat dalam lingkungan operasional yang sesungguhnya.

3.2 Perancangan Sistem

Penelitian ini berfokus pada rancang bangun sistem berupa model deteksi berbasis YOLOv8m yang diintegrasikan dengan prototipe produksi area *conveyor* yang dilengkapi kamera sebagai *input* dan aktuator sebagai *output*. Pada penelitian ini terdapat enam area yang terdiri dari *folding box placement area*, *image acquisition area*, *rejection area*, *good folding box area*, *computerization area*, dan *air compression area*, yang masing-masing memiliki fungsi yang berbeda-beda dalam mendukung proses identifikasi dan pemisahan produk cacat secara *real-time*. Sistem yang ditampilkan pada Gambar 3.3 terdiri dari enam area utama yang saling terintegrasi untuk memastikan proses pengolahan *folding box* berjalan dengan efisien.



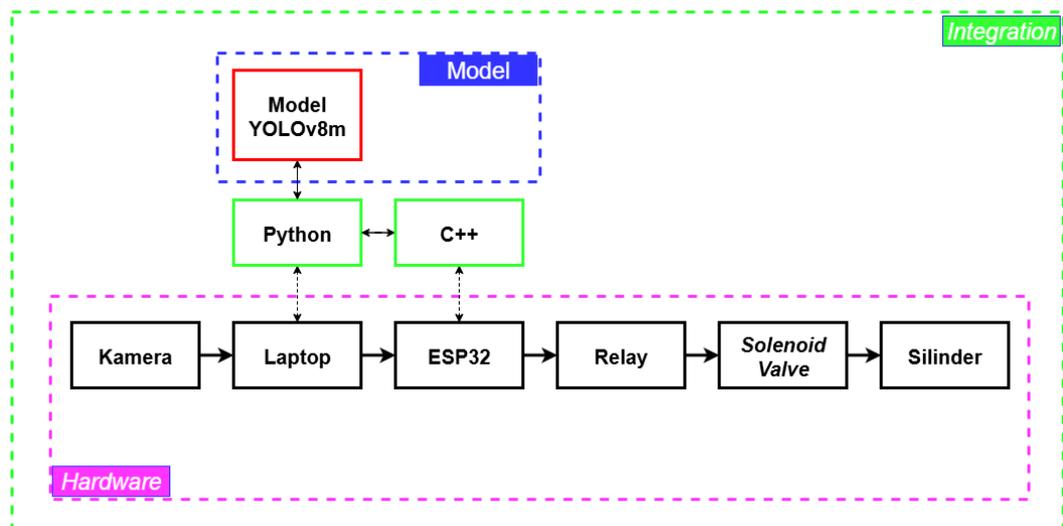
Gambar 3.3 Rancangan Sistem

Rancangan sistem pada gambar di atas berfungsi sebagai peta visual dan fungsional yang menggambarkan *mapping* tiap komponen dalam proses inspeksi produk. *Folding box placement area* berfungsi untuk menempatkan *folding box* pada *conveyor* yang digerakkan oleh *servo motor* untuk memindahkan *box* ke tahap berikutnya. *Image Acquisition Area* dilengkapi dengan kamera yang bertugas menangkap gambar *folding box*, yang kemudian dianalisis menggunakan model YOLOv8m untuk mendeteksi cacat. Setelah itu, *folding box* yang cacat diarahkan ke *rejection area*, di mana silinder *pneumatik* yang dikendalikan oleh *solenoid valve* memindahkan *folding box* cacat ke *rejection*. Sebaliknya, *folding box* yang lolos deteksi diteruskan ke *good folding box area*, yang merupakan tempat penampungan untuk produk yang memenuhi standar kualitas.

Keseluruhan proses dikendalikan melalui *computerization area*, yang terdiri dari laptop untuk memproses gambar dan mendeteksi cacat. ESP32 untuk mengontrol aktuator berupa *solenoid valve* dengan menggunakan *relay* sebagai perantara untuk menghubungkan dan mengaktifkan perangkat listrik secara aman dan terisolasi. Untuk mendukung pengoperasian silinder *pneumatik*, *air compression area* menyediakan udara bertekanan yang dihasilkan oleh *compressor* dan tekanannya diatur oleh *air pressure regulator* agar sesuai dengan kebutuhan

sistem. Setiap area memiliki fungsi khusus yang dirancang untuk bekerja secara sinkron, memastikan *folding box* yang tidak memenuhi standar dikeluarkan dari jalur produksi, sementara *folding box* yang baik diteruskan ke tahap berikutnya menggunakan *conveyor*.

Conveyor dalam sistem ini berperan untuk memindahkan *folding box* dari satu titik ke titik lainnya sepanjang jalur produksi. *Conveyor* yang digunakan merupakan jenis *conveyor* dari Dobot, yang mendapatkan *supply* daya 24V langsung dari *arm robot* Dobot itu sendiri. Dengan demikian, *arm robot* Dobot tidak berfungsi sebagai unit kendali utama, melainkan sebagai perangkat pendukung yang mengaktifkan pergerakan *conveyor* dalam sistem ini. Untuk memperjelas arsitektur sistem ditampilkan pada Gambar 3. 4.



Gambar 3. 4 Arsitektur Sistem

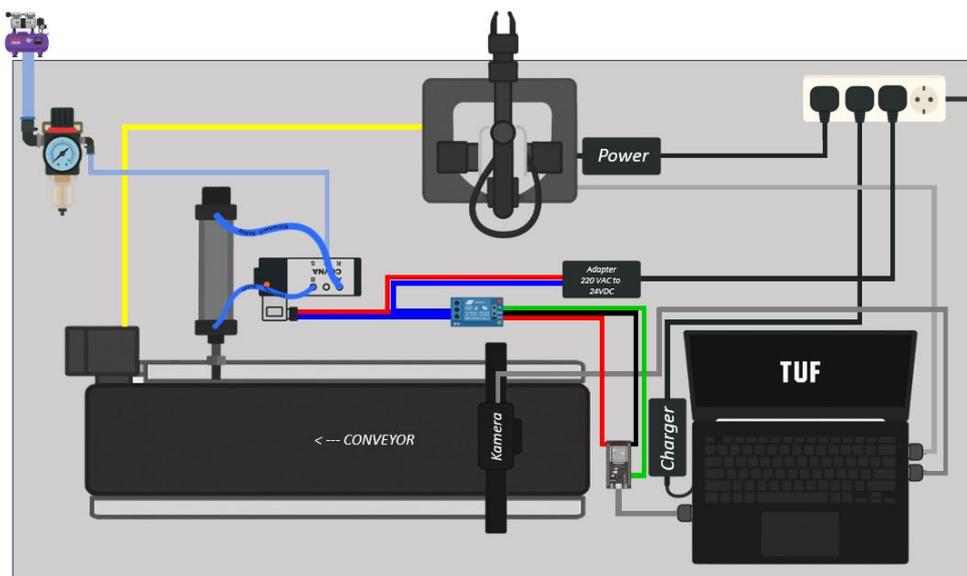
Arsitektur sistem secara keseluruhan dirancang untuk memberikan gambaran mengenai komponen-komponen utama dan interaksi antar komponen dalam sistem. Pada Gambar 3. 4 mengilustrasikan komponen utama sistem yang terdiri dari unit kamera, laptop, ESP32, *relay*, *solenoid valve* dan silinder. Masing-masing komponen memiliki peran spesifik dan saling berinteraksi dalam alur data yang terstruktur.

3.2.1 Perancangan *Hardware*

Pada tahap ini dilakukan proses perancangan *hardware* yang menjadi dasar dari sistem yang akan dibangun. Perancangan ini mencakup pembuatan skema rangkaian serta *flowchart* sistem *hardware* yang berfungsi sebagai panduan dalam proses implementasi dan memastikan seluruh komponen dapat bekerja sesuai fungsi yang diharapkan.

3.2.1.1 Skema Rangkaian

Pada tahap perancangan sistem deteksi produk cacat pada *folding box* berbasis YOLOv8m, diperlukan integrasi berbagai perangkat keras yang saling terhubung untuk mendukung proses otomatisasi. Skema rangkaian berfungsi untuk menggambarkan hubungan antar komponen *hardware* yang digunakan dalam sistem, seperti kamera, ESP32, aktuator, serta koneksi ke sumber daya. Skema rangkaian secara keseluruhan tampak atas ditampilkan pada Gambar 3. 5.

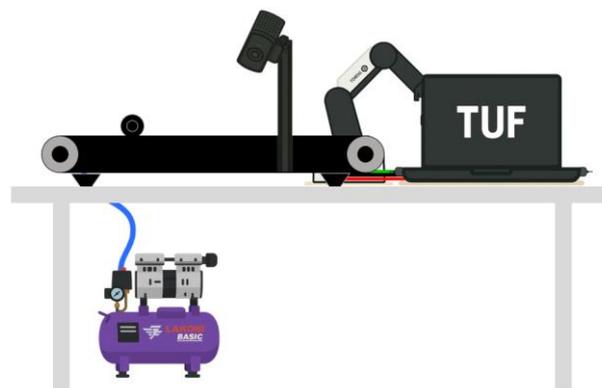


Gambar 3. 5 Skema Rangkaian Tampak Atas

Skema rangkaian di atas memperlihatkan alur koneksi antara beberapa komponen utama seperti laptop sebagai pusat kendali, kamera untuk pengambilan data gambar, ESP32, *solenoid valve*, aktuator berupa silinder *pneumatik*, serta perangkat pendukung lain seperti *relay*, adaptor, dan *power supply*. Skema rangkaian ini dirancang untuk memastikan seluruh proses inspeksi dan penanganan

folding box berjalan secara otomatis, mulai dari pengambilan gambar, pemrosesan data, hingga eksekusi aktuator yang bertugas memisahkan produk cacat. Setiap koneksi antar perangkat memperlihatkan alur sinyal dan distribusi daya yang mendukung sistem agar dapat bekerja secara terpadu sesuai fungsinya.

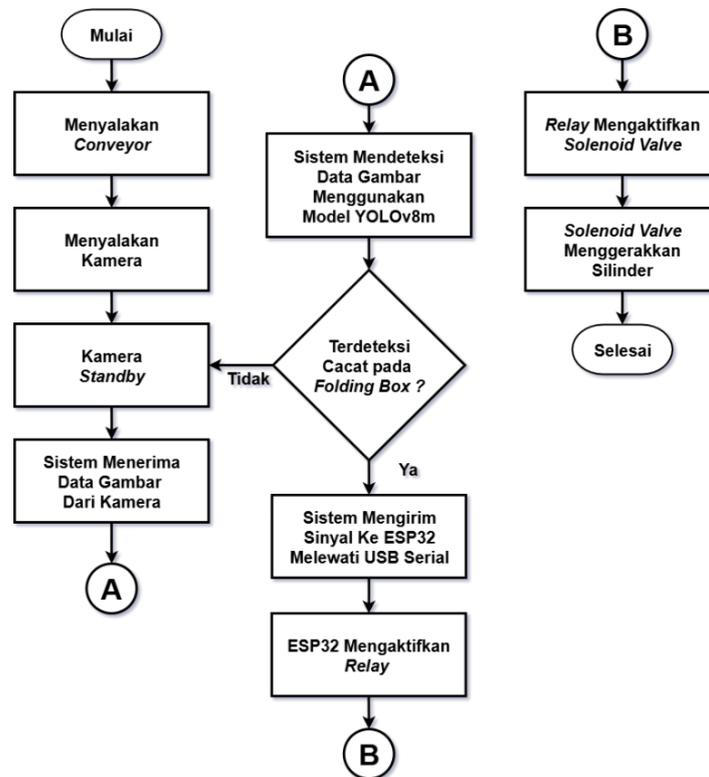
Selanjutnya pada Gambar 3. 6 diperlihatkan skema rangkaian tampak depan dari sistem yang akan dibangun. Ilustrasi ini memberikan gambaran visual mengenai desain sistem tampak depan, sehingga memudahkan pemahaman mengenai tata letak komponen utama serta memperjelas konsep perencanaan sistem yang akan direalisasikan.



Gambar 3. 6 Skema Rangkaian Tampak Depan

3.2.1.2 *Flowchart* Sistem *Hardware*

Flowchart sistem *hardware* merupakan diagram alir yang menggambarkan urutan proses kerja *hardware* secara terstruktur. Diagram ini memudahkan identifikasi hubungan antar komponen, aliran data, dan tahapan operasi sistem *hardware*. *Flowchart* sistem *hardware* disajikan pada Gambar 3. 7.

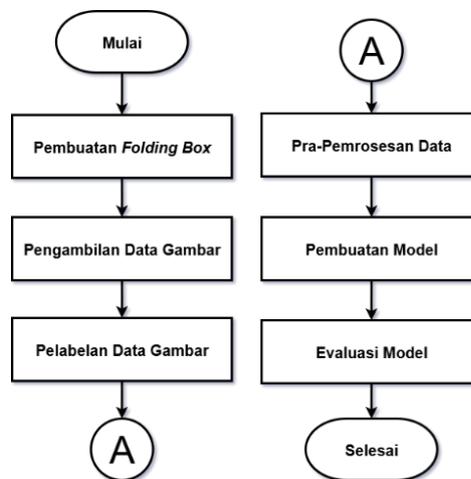


Gambar 3. 7 Flowchart Sistem Hardware

Berdasarkan Gambar 3. 7, sistem *hardware* bekerja secara berurutan mulai dari menyalakan *conveyor* dan kamera. Setelah kamera aktif, perangkat tersebut berada dalam kondisi *standby* untuk menunggu objek *folding box* yang akan diperiksa. Saat *folding box* bergerak di atas *conveyor*, kamera mengambil gambar dan mengirimkannya ke sistem untuk dianalisis menggunakan model YOLOv8m. Hasil analisis ini menentukan langkah selanjutnya. Jika tidak ditemukan cacat pada *folding box*, sistem akan kembali ke tahap awal untuk memantau objek berikutnya. Namun, apabila terdeteksi adanya cacat, sistem akan mengirimkan sinyal ke modul ESP32 melalui koneksi USB Serial. ESP32 kemudian mengaktifkan *relay*, yang selanjutnya menyalakan *solenoid valve*. *Solenoid valve* ini menggerakkan silinder *pneumatik* untuk mendorong atau memisahkan *folding box* yang cacat dari jalur utama *conveyor*. Setelah proses pemisahan selesai, sistem kembali ke kondisi awal untuk memproses *folding box* selanjutnya secara berulang.

3.2.2 Perancangan Pembuatan Model

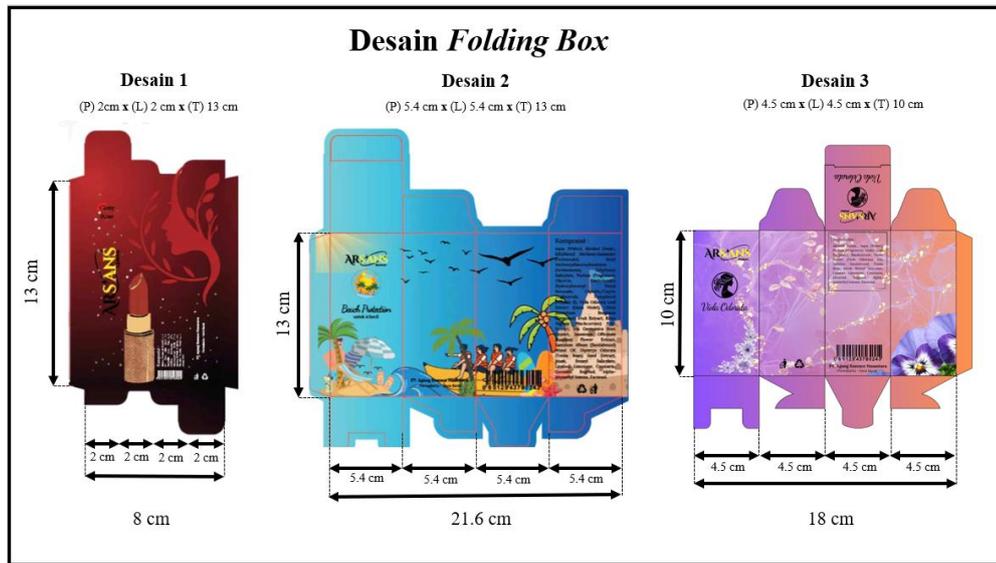
Pada tahap ini difokuskan untuk perancangan pembuatan model, mulai dari pembuatan *folding box* untuk dijadikan *dataset*, perancangan proses pengambilan data, pelabelan data, pembuatan model, dan evaluasi model. Hal ini tidak hanya memudahkan pemahaman mengenai fungsi masing-masing kerangka perancangan model, tetapi juga membantu dalam proses evaluasi dan *troubleshooting* apabila terjadi kendala selama implementasi. Secara keseluruhan *flowchart* pembuatan model disajikan pada Gambar 3. 8.



Gambar 3. 8 *Flowchart* Pembuatan Model

3.2.2.1 Perancangan Pembuatan *Folding Box*

Pada tahap pembuatan *folding box* di buat untuk memahami jenis data yang akan digunakan, objek dibuat secara manual dengan tujuan untuk memahami karakteristik dan jenis data yang akan digunakan dalam penelitian. *Folding box* yang digunakan memiliki tiga variasi dimensi dan warna yang berbeda, desain 1 memiliki dimensi 2 cm × 2 cm × 13 cm, desain 2 berukuran 5.4 cm × 5.4 cm × 13 cm, sedangkan desain 3 memiliki ukuran 4.5 cm × 4.5 cm × 10 cm. Untuk gambaran desain disajikan pada Gambar 3. 9.



Gambar 3. 9 Desain *Folding Box*

Variasi desain pada *dataset* ini ditujukan untuk meningkatkan kompleksitas data dan menguji kemampuan sistem dalam mengenali berbagai kondisi visual yang menyerupai situasi nyata. Dengan adanya perbedaan pada aspek warna, bentuk, dan elemen visual lainnya, sistem diuji agar mampu beradaptasi dengan berbagai skenario pencahayaan, latar belakang, serta karakteristik objek yang bervariasi. Selain itu, variasi ini juga bertujuan untuk memperkaya komposisi *dataset* sehingga model yang dikembangkan menjadi lebih adaptif, *robust*, dan akurat dalam mendeteksi serta mengklasifikasikan perbedaan bentuk dan warna pada objek fisik secara konsisten.

Dalam penerapannya, pemilihan skema warna pada desain pertama menggunakan kombinasi gradasi antara merah dan hitam yang ditujukan untuk menciptakan kontras visual yang cenderung gelap. Hal ini bertujuan agar sistem dapat mendeteksi objek berwarna hitam secara lebih efektif, meskipun warnanya memiliki kemiripan dengan permukaan *conveyor*. Sementara itu, pada desain kedua dan ketiga digunakan skema warna yang lebih terang untuk meningkatkan keterbacaan dan diferensiasi visual. Desain kedua memanfaatkan gradasi antara biru muda dan biru tua, sedangkan desain ketiga mengombinasikan gradasi ungu dan oranye. Penyesuaian tambahan juga dilakukan melalui penambahan aksent

warna putih, serta modifikasi elemen grafis yang menyerupai efek sobekan (*torn effect*) guna memberikan tantangan tambahan bagi model dalam proses deteksi.

3.2.2.2 Perancangan Proses Pengambilan Data

Pada tahap ini dilakukan proses pengambilan *dataset* yang digunakan untuk keperluan pelatihan dan pengujian model dalam sistem deteksi cacat pada *folding box*. Gambar diambil menggunakan kamera dengan resolusi 640×640 *pixel*, yang dipilih untuk memastikan kualitas gambar tetap memadai dalam menangkap detail objek. Pemilihan resolusi ini disesuaikan dengan kebutuhan proyek serta kompatibel dengan arsitektur model deteksi yang akan dibuat. Resolusi 640×640 dipilih karena menawarkan keseimbangan antara kecepatan pemrosesan dan akurasi deteksi yang masih dalam batas wajar. Meskipun resolusi lebih tinggi dapat meningkatkan akurasi, waktu pemrosesannya jauh lebih lama, sementara pada 640×640 penurunan akurasi masih tergolong kecil. Jumlah gambar yang diambil mengacu pada penelitian sebelumnya yang menggunakan 200 gambar untuk setiap kategori cacat dan tetap mampu mencapai nilai akurasi serta presisi yang tinggi.

Berdasarkan pertimbangan tersebut, pada penelitian ini dikumpulkan sebanyak 200 gambar untuk setiap kategori cacat dan kategori normal dari tiga desain yang dibuat sebagai *dataset*. Selain itu, ditambahkan kategori cacat gabungan, yaitu gambar yang mengandung dua jenis cacat sekaligus, yakni robek dan terlipat dalam satu gambar. Uji coba ini bertujuan untuk mengevaluasi kinerja model dalam mengenali berbagai jenis cacat secara akurat. Pengambilan gambar dilakukan dari dua posisi kamera dengan ketinggian 30 cm dan 40 cm.

3.2.2.3 Perancangan Proses Pelabelan Data (*Annotation*)

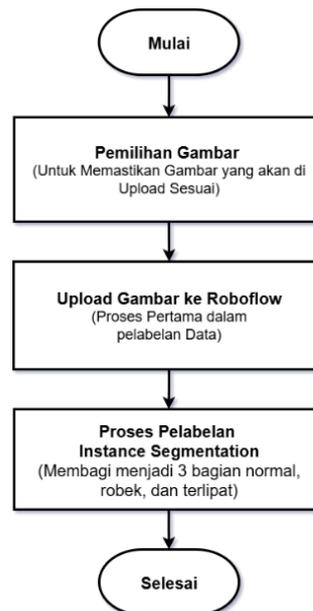
Setelah proses pengambilan data, dilakukan pelabelan (*annotation*) pada setiap gambar. Proses ini bertujuan agar sistem yang dikembangkan dapat mengenali dan mempelajari pola dari *dataset* yang akan digunakan untuk pelatihan. Pelabelan dilakukan secara manual menggunakan platform Roboflow dengan menandai jenis normal, robek dan terlipat pada objek *folding box* dengan menggunakan fungsi *segmentation*, sehingga model dapat membedakan antar objek secara individu secara akurat dan menyesuaikan dengan jenis cacat yang memiliki variasi bentuk. Proses ini mempertimbangkan berbagai faktor yang memengaruhi

Agung Satria Pamungkas, 2025

RANCANG BANGUN SISTEM IDENTIFIKASI PRODUK CACAT PADA FOLDING BOX BERBASIS YOLOv8m YANG TERINTEGRASI DENGAN REJECTION AREA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

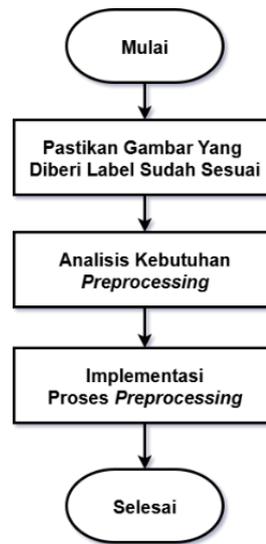
segmentation, seperti ukuran cacat, lokasi cacat dalam area *folding box*, dan bentuk cacat. Dengan pelabelan yang tepat, model mampu mempelajari representasi visual secara menyeluruh untuk mendeteksi dan mengklasifikasikan cacat secara lebih presisi. Untuk memberikan pemahaman yang lebih jelas mengenai proses pelabelan data di Roboflow, disusun sebuah *flowchart* yang ditampilkan pada Gambar 3. 10.



Gambar 3. 10 *Flowchart* Pelabelan Data

3.2.2.4 Perancangan Pra-pemrosesan Data (*Preprocessing*)

Setelah proses pelabelan data selesai, langkah selanjutnya adalah tahap pra-pemrosesan data (*preprocessing*). Pada tahap ini, diterapkan beberapa metode untuk meningkatkan keragaman data melalui berbagai modifikasi visual, sehingga model dapat dilatih dengan data yang memiliki variasi visual yang lebih luas. Hal ini penting untuk memperkuat kemampuan generalisasi model terhadap kondisi nyata di lapangan. *Flowchart* proses *preprocessing* data disajikan pada Gambar 3. 11.



Gambar 3. 11 Flowchart Proses *Preprocessing* Data

Pada Gambar 3. 1, proses *preprocessing* data dilakukan menggunakan platform Roboflow sebagai alat bantu untuk menerapkan tahapan *preprocessing* gambar secara otomatis. Setelah proses *annotation* gambar selesai, langkah selanjutnya adalah melakukan pembagian *dataset* (*splitting*) dengan proporsi 70% untuk data pelatihan (*training*), 20% untuk validasi (*validation*), dan 10% untuk pengujian (*testing*). Proporsi ini dipilih berdasarkan acuan dari beberapa penelitian sebelumnya yang menunjukkan bahwa konfigurasi tersebut merupakan praktik standar dalam pelatihan model.

Tujuannya adalah untuk memastikan bahwa model memiliki cukup data untuk belajar, sementara data validasi dan pengujian digunakan untuk mengukur kemampuan generalisasi dan mendeteksi kemungkinan *overfitting*. Meskipun terdapat perbedaan pada jumlah *epoch* yang digunakan karena keterbatasan sumber daya di Google Colab. Selanjutnya, data yang telah melalui tahapan *preprocessing* dievaluasi secara *internal* menggunakan platform Roboflow, dengan memanfaatkan model Roboflow 3.0 Instance Segmentation (*Fast*).

Evaluasi ini bertujuan untuk memverifikasi kesesuaian antara hasil pelabelan (*annotation*) dan proses *preprocessing* melalui pengujian performa awal. Hasil dari evaluasi *internal* ini berupa metrik performa seperti *mean average precision* (*mAP*), *precision*, dan *recall*. Untuk mendukung proses validasi yang lebih komprehensif,

disediakan juga metrik evaluasi model yang mencakup nilai mAP, mAP@50:95, serta grafik pelatihan (*training graph*) yang memuat informasi terkait *box loss*, *class loss*, dan *object loss*.

3.2.2.5 Perancangan Proses Pembuatan Model

Tahap ini merupakan bagian yang sangat penting dalam proses pembuatan model, karena pada tahap inilah model dilatih untuk mengenali objek sesuai dengan *dataset* yang telah disiapkan. Setelah proses *annotation*, *preprocessing*, dan *internal evaluation* di *platform* Roboflow selesai dilakukan dengan hasil yang sesuai dengan rancangan, maka pembuatan model dapat dilanjutkan. Model akan dilatih menggunakan dua metode, yaitu tanpa *augmentation* dan dengan *augmentation*, untuk membandingkan performa keduanya. Untuk *parameters* yang digunakan disajikan pada Tabel 3. 1 sebagai acuan utama. Detail pengaturan ini sangat berpengaruh dalam pembuatan model, baik pada *training* tanpa *augmentation* maupun dengan penerapan *augmentation*.

Tabel 3. 1 Training Parameters

<i>Hyperparameter</i>	<i>Value</i>
<i>Learning Rate</i>	0.001429
<i>Image Size</i>	640 x 640
<i>Optimizer</i>	<i>Auto</i>
<i>Batch Size</i>	16
<i>Epoch</i>	50

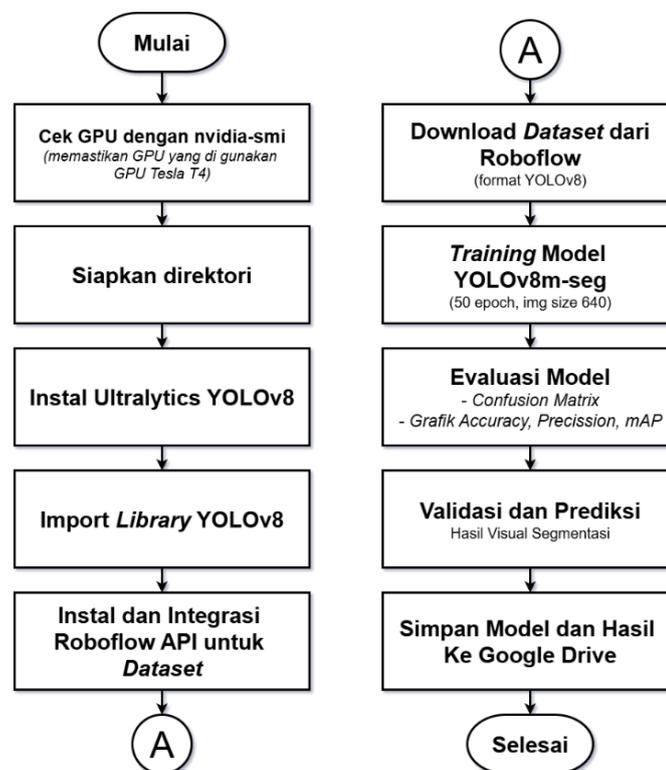
Berdasarkan tabel di atas pada tahap pelatihan, digunakan parameter bawaan (*default*) dari YOLO, yaitu *learning rate* sebesar 0.001429, *batch size* 16, serta resolusi gambar 640 x 640 *pixel*. Pemilihan parameter bawaan ini berdasarkan penelitian sebelumnya dan dilakukan agar proses pelatihan lebih stabil dan konsisten, sekaligus menghindari potensi *overfitting* akibat penyesuaian *parameter* yang tidak terkontrol. Selain itu, penggunaan *batch size* 16 untuk menyeimbangkan antara kapasitas GPU dan kecepatan proses pelatihan di Google Colab, sedangkan ukuran citra 640×640 dipilih karena memberikan kualitas deteksi yang memadai tanpa membebani memori secara berlebihan.

Agung Satria Pamungkas, 2025

RANCANG BANGUN SISTEM IDENTIFIKASI PRODUK CACAT PADA FOLDING BOX BERBASIS YOLOv8m YANG TERINTEGRASI DENGAN REJECTION AREA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Pada metode pertama, pelatihan dilakukan hanya dengan data asli, sedangkan pada metode kedua digunakan teknik *augmentation* agar data menjadi lebih beragam. Perbandingan ini bertujuan untuk mengetahui metode mana yang menghasilkan model paling optimal. Model dengan hasil terbaik nantinya akan dipilih sebagai model final. Proses pelatihan model dilakukan menggunakan *platform* Google Colab, yang mendukung pemrosesan komputasi GPU T4 dan integrasi langsung dengan berbagai *library* serta Roboflow. Adapun langkah-langkah pembuatan model disajikan secara sistematis melalui *flowchart* yang ditampilkan pada Gambar 3. 12.



Gambar 3. 12 *Flowchart* Pembuatan Model

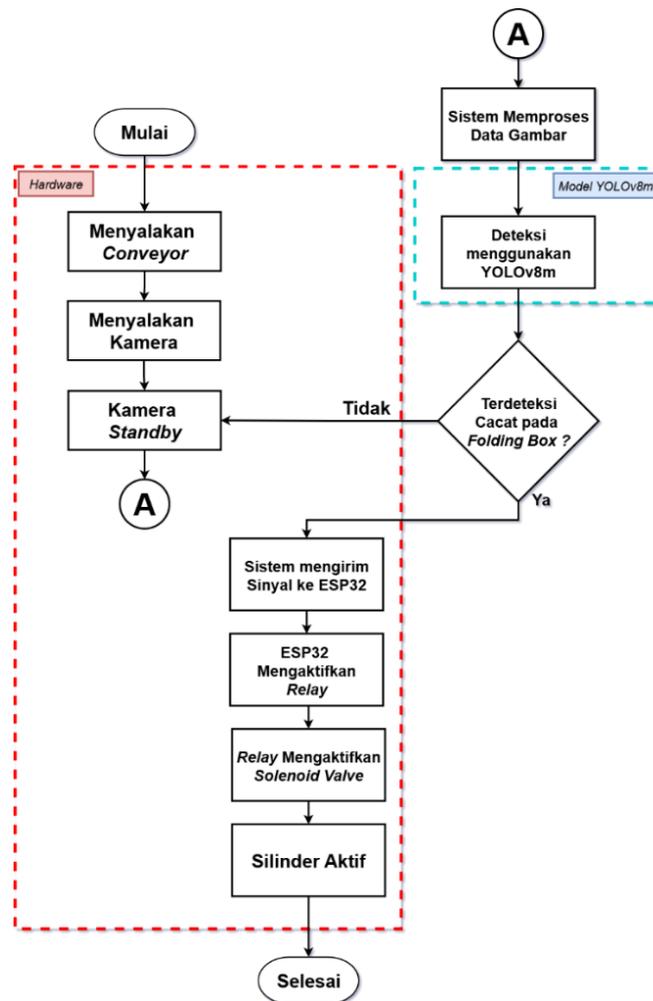
3.2.2.6 Perancangan Evaluasi Model

Setelah proses pelatihan model selesai, langkah selanjutnya adalah melakukan evaluasi untuk memastikan bahwa model memiliki performa yang baik dalam melakukan deteksi sesuai dengan tujuan perancangan. Evaluasi ini dilakukan dengan menggunakan *confusion matrix* sebagai dasar pengukuran performa, kemudian dari hasil *confusion matrix* selanjutnya dievaluasi menggunakan metrik

evaluasi seperti akurasi (*accuracy*), presisi (*precision*), *recall*, *mean Average Precision (mAP)*, *F1-score*, dan *loss*. Metrik-metrik ini memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan antara objek yang benar dan salah terdeteksi. Setiap metrik memiliki rumus matematis tersendiri yang digunakan untuk menghitung performa model secara kuantitatif, sehingga hasil evaluasi dapat dinilai secara objektif. Untuk persamaan metrik evaluasi disajikan pada Tabel 2. 3.

3.2.3 Perancangan Integrasi *Hardware* dan Model

Setelah model YOLOv8 selesai dibuat dan dievaluasi, tahap berikutnya adalah integrasi antara sistem *hardware* dan model. Proses ini dimulai dengan integrasi untuk menggerakkan *conveyor*, yaitu dengan memprogram sistem Dobot menggunakan kode *blocks* yang telah dibuat untuk mengoperasikan *conveyor*. Setelah *conveyor* berhasil dijalankan, proses selanjutnya menyalakan kamera dan menunggu hingga gambar siap diproses. Gambar yang diambil akan diproses oleh sistem menggunakan bahasa pemrograman Python dan model YOLOv8m untuk mendeteksi cacat pada *folding box*. Jika terdeteksi cacat, sistem mengirimkan sinyal ke ESP32 melewati kabel USB serial. Selanjutnya, ESP32 akan mengaktifkan *relay*, yang kemudian mengaktifkan *solenoid valve* untuk menggerakkan silinder *pneumatik* dan mengeluarkan produk cacat dari jalur *conveyor*. Untuk *flowchart* integrasi sistem *hardware* dan model disajikan pada Gambar 3. 13.



Gambar 3. 13 Flowchart Integrasi Hardware dan Model

3.2.4 Perancangan Pengujian Integrasi Hardware dan Model

Tahap ini merupakan perancangan pengujian integrasi, dilakukan pengujian integrasi *hardware* untuk memastikan seluruh komponen berfungsi sesuai rancangan. Pengujian mencakup aspek fungsionalitas, untuk memeriksa kesesuaian kerja sistem, dan efektivitas, untuk menilai kinerja serta efisiensinya.

3.2.4.1 Perancangan Pengujian Fungsionalitas Sistem

Pada tahap pengujian fungsionalitas sistem identifikasi produk cacat pada *folding box* yang telah diintegrasikan dengan *rejection area*, dilakukan pengujian *black box* untuk menguji sistem yang telah dirancang. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap fungsi utama dalam sistem, mulai dari proses *input* gambar melalui kamera, pemrosesan identifikasi menggunakan model

Agung Satria Pamungkas, 2025

RANCANG BANGUN SISTEM IDENTIFIKASI PRODUK CACAT PADA FOLDING BOX BERBASIS YOLOv8m YANG TERINTEGRASI DENGAN REJECTION AREA

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

YOLOv8m, pendeteksian kondisi *folding box* seperti normal, robek, dan terlipat, hingga proses pengeluaran *folding box* dari *conveyor*, dapat berjalan sesuai dengan spesifikasi yang diharapkan tanpa perlu mengetahui struktur internal sistem. Rancangan pengujian fungsionalitas sistem dirancang pada Tabel 3. 2.

Tabel 3. 2 Perancangan Pengujian Fungsionalitas Sistem

No.	Fitur	Fungsi	Hasil yang diharapkan	Hasil Pengujian	Status
1.	<i>Input Kamera</i>	Mengambil gambar <i>folding box</i> saat bergerak di <i>conveyor</i>	Gambar berhasil ditangkap secara <i>real-time</i>		
2.	Deteksi Cacat (YOLOv8)	Mendeteksi kondisi <i>folding box</i> normal, robek, atau terlipat	Model mendeteksi dan memberi label pada setiap <i>box</i> sesuai kondisi		
3.	<i>Output Deteksi</i>	Menampilkan hasil deteksi pada layar	Label kondisi <i>box</i> ditampilkan (normal, robek, terlipat)		
4.	Penyimpanan Data Deteksi	Menyimpan hasil deteksi cacat ke dalam penyimpanan lokal	Hasil deteksi cacat (robek/terlipat) tercatat penyimpanan lokal		
5.	Komunikasi Laptop-ESP32 (USB Serial)	Mengirimkan sinyal hasil deteksi dari sistem ke ESP32 melalui USB serial	ESP32 menerima sinyal instruksi untuk pengeluaran <i>folding box</i> cacat		
6.	Pengendalian <i>Relay</i> oleh ESP32	ESP32 mengatur logika kontrol untuk mengaktifkan <i>relay</i>	ESP32 mengaktifkan <i>output</i> digital menuju <i>relay</i> saat menerima perintah dari laptop		
7.	Silinder <i>Pneumatik</i> Aktif (via <i>Relay</i>)	<i>Relay</i> mengalirkan arus untuk mengaktifkan aktuator silinder	<i>Relay</i> aktif, silinder <i>pneumatik</i> bergerak untuk mendorong <i>folding box</i> keluar <i>conveyor</i>		
8.	Output pada <i>Folding Box</i> Normal	Tidak mengaktifkan <i>relay</i> jika <i>box</i> dalam kondisi normal	<i>Folding box</i> normal tetap berjalan di <i>conveyor</i> tanpa gangguan		

3.2.4.2 Perancangan Pengujian Efektivitas Deteksi

Pada tahap ini dilakukan perancangan pengujian terhadap model deteksi yang telah diintegrasikan dengan sistem *hardware*, sekaligus pengujian model sistem deteksi yang mendukung operasionalisasi perangkat tersebut. Pengujian efektivitas sistem dilakukan berdasarkan variasi jarak antar produk dan ketinggian kamera. Ketinggian kamera yang digunakan dalam pengujian ini adalah 30 cm dan 40 cm, sedangkan jarak antar *folding box* divariasikan dalam rentang 35 cm hingga 10 cm. Kemudian dilakukan pengujian terhadap nilai FPS (*Frame Per Second*) serta durasi pemrosesan dari *input* hingga *output* sistem. Parameter ini digunakan untuk mengukur kinerja waktu nyata (*real-time performance*) dari sistem dalam mendeteksi dan merespons objek pada ketinggian kamera 30 cm dan 40 cm. Format pengujian dan kombinasi pengujian disajikan secara rinci pada Tabel 3. 3.

Tabel 3. 3 Perancangan Pengujian Efektivitas Deteksi berdasarkan Jarak

<i>Folding box</i>	Jarak antar <i>folding box</i> (cm)	Kondisi <i>Folding Box</i>	FPS (detik)	Proses <i>Input-Output</i>
1	35			
	30			
	25			
	20			
	15			
	10			
...	35			
	30			
	25			
	20			
	15			
	10			
n+1	35			
	30			
	25			
	20			
	15			
	10			

Kemudian untuk metrik evaluasinya menggunakan FAR (*false acceptance rate*) dan FRR (*false rejection rate*) dengan rumus yang digunakan seperti persamaan (2.12) dan (2.13). Pada tahap ini ditambahkan *dataset* normal (*folding box* tanpa cacat) untuk menguji sistem dan mengukur frekuensi deteksi yang tidak

sesuai. Penambahan *dataset* normal penting sebagai *identification attempts* dalam perhitungan FAR dan FRR, membantu *confidence threshold*, mengurangi bias akibat ketidakseimbangan, serta menguji *robustness* model. Format untuk pengujian FAR dan FRR disajikan pada Tabel 3. 4.

Tabel 3. 4 Perancangan Pengujian Efektivitas FAR dan FRR

Folding Box	Kondisi Aktual	Kondisi Sistem	Kategori Metrik
1			
2			
...			
n+1			