BAB I

PENDAHULUAN

1.1 Latar Belakang

Kopi merupakan salah satu komoditas ekspor Indonesia dengan nilai ekspor pada tahun 2024 yang mencapai total 1.624.300.000 US$ yang berkisar 26 triliun rupiah. Angka tersebut menyumbang sebesar 31% dari total nilai ekspor hasil pertanian Indonesia pada tahun 2024. Sementara itu, berat bersih dari ekspor kopi di tahun yang sama, mencapai 312,900 ton (Indonesia, 2025.). Data-data tersebut menjadi bukti seberapa signifikan pasokan biji kopi terhadap negara Indonesia.

Sebelum biji kopi dapat dipasarkan, diperlukan serangkaian proses pasca-panen untuk mengolah buah dari tanaman kopi. Salah satu tahapan penting dalam proses tersebut adalah sortasi, yang berperan dalam menentukan kualitas serta nilai jual biji kopi yang dihasilkan. Proses sortasi bertujuan untuk memisahkan biji kopi cacat agar sesuai dengan Standar Nasional Indonesia (SNI) 01-2907-2008, yang umumnya dilakukan pada sistem conveyor belt. Proses ini sangat bergantung pada kemampuan manusia dalam memantau dan mengawasi biji kopi cacat (Pusat Penelitian Kopi dan Kakao Indonesia, 2023). Hal ini dilakukan, karena biji kopi cacat dapat memengaruhi kualitas seduhan kopi serta berpotensi membahayakan kesehatan akibat kandungan zat berbahaya di dalamnya, sehingga berdampak langsung pada nilai jual biji kopi (Chang & Liu, 2024). Umumnya, proses sortasi dilakukan dengan meletakkan biji kopi di atas konveyor, kemudian pekerja memisahkan biji cacat secara manual. Metode ini sangat bergantung pada kemampuan visual manusia dalam mendeteksi detail kecil pada biji kopi, yang memerlukan waktu lama serta tingkat ketelitian tinggi.

Terdapat penelitian yang menunjukkan keterbatasan metode manual setelah membandingkan penyortiran ukuran biji kopi secara manual dan menggunakan mesin sortasi. Hasilnya menunjukkan bahwa mesin sortasi mampu menyortir 200–300 kg/jam, sedangkan penyortiran manual hanya mampu menyortir sekitar 200 kg dalam waktu 4–5 jam dengan melibatkan 4–5 orang pekerja (Sidehabi dkk., 2022). Ketergantungan yang tinggi terhadap manusia meningkatkan risiko human error, yang dapat menghasilkan biji kopi berkualitas rendah akibat stres kerja dan kejenuhan (Febriana dkk., 2022). Oleh karena itu, salah satu solusi untuk menekan risiko tersebut adalah dengan mengurangi peran manusia dalam proses sortasi.

Dalam kasus penyortiran biji kopi yang mengandalkan indikator visual, pemanfaatan algoritma *computer vision* menjadi pendekatan yang tepat. *Computer vision* bekerja dengan mempelajari pola visual untuk mengidentifikasi objek dengan prinsip yang meniru penglihatan manusia (Matsuzaka dkk., 2023). Salah satu penerapan utama *computer vision* adalah *object detection*, yaitu proses lokalisasi dan klasifikasi objek yang tertangkap kamera. Prinsip ini selaras dengan proses sortasi biji kopi yang dilakukan oleh manusia, sehingga lebih mudah diadaptasikan ke sistem yang telah ada.

Object detection memiliki dua tugas utama, yaitu klasifikasi dan lokalisasi objek. Berbagai arsitektur deteksi objek telah dikembangkan dengan pendekatan yang berbeda-beda, namun umumnya memiliki komponen *backbone* yang bertugas mengekstraksi fitur menggunakan jaringan saraf tiruan. Penelitian sebelumnya menggunakan ResNet-50 untuk klasifikasi biji kopi di atas konveyor dan mencapai akurasi sebesar 93,3% (Micaraseth dkk., 2022). Penelitian lain memanfaatkan Mask R-CNN dengan *backbone* ResNet-50 untuk klasifikasi biji kopi pada tiga kecepatan konveyor berbeda. Namun, penelitian tersebut menunjukkan penurunan performa pada kecepatan tertinggi (70 RPM) dengan nilai *accuracy* 0,84, *precision* 0,56, dan *recall* 0,75, yang disebabkan oleh sensitivitas model terhadap *motion blur* (Talunga, 2025). Hal ini menunjukkan bahwa model deteksi yang digunakan belum mampu beradaptasi dengan variasi kecepatan konveyor. Terdapat penelitian yang melakukan analisis komparatif pada deteksi kendaraan lalu lintas menggunakan model YOLOv9, YOLOX, dan RT-DETR pada data CCTV, yang merupakan kasus dengan kompleksitas tinggi. Hasil penelitian menunjukkan bahwa RT-DETR unggul pada hampir seluruh metrik evaluasi, termasuk mAP50 sebesar 0,891 (Gladiensyah Bihanda dkk., 2024). Hal ini membuktikan bahwa RT-DETR memiliki performa yang baik pada kasus deteksi yang kompleks. Berdasarkan latar belakang tersebut, penelitian ini akan mengaplikasikan RT-DETR untuk mendeteksi dan mengklasifikasikan biji kopi cacat dan non-cacat pada sistem konveyor sesuai dengan SNI 01-2907-2008. Diharapkan model ini mampu memberikan kinerja yang stabil meskipun pada kondisi yang sulit, seperti variasi kecepatan konveyor. Penelitian ini juga diharapkan dapat mendorong kemajuan teknologi pengolahan komoditas Indonesia yang lebih efisien serta meningkatkan daya saing produk agrikultur di pasar internasional.

1.2 Landasan Teori

1.2.1 Kopi

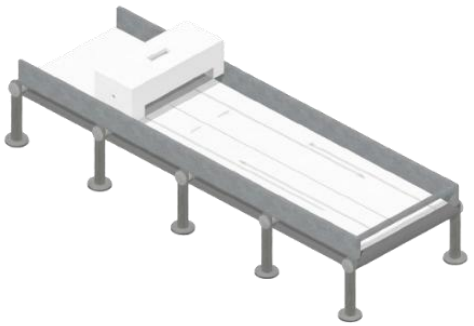
Gambar 1. Kopi

Kopi (*Coffea*) merupakan salah satu tanaman yang paling dikenal dan tersebar luas di seluruh dunia. Tanaman ini memiliki nilai ekonomi dan sosial yang tinggi, di mana produk biji kopi telah menjadi bagian dari kebiasaan masyarakat global. Bagi Indonesia, kopi merupakan sumber devisa penting dan berkontribusi signifikan terhadap Produk Domestik Bruto (PDB), sekaligus membuka lapangan kerja bagi jutaan petani di seluruh nusantara (Azhari dkk., 2025).

Proses penyortiran biji kopi merupakan tahapan krusial dalam pasca-panen yang bertujuan memisahkan biji berkualitas tinggi dari biji cacat. Menurut SNI 01-2907-2008, biji kopi cacat diklasifikasikan ke dalam beberapa jenis, seperti biji hitam, biji coklat, biji pecah, dan biji muda. Umumnya, penyortiran masih dilakukan secara manual menggunakan penglihatan manusia, namun metode ini dinilai kurang efisien karena bersifat padat karya, memerlukan waktu lama, serta rentan terhadap kelelahan dan stres pekerja (Febriana dkk., 2022).

1.2.2 Konveyor

Konveyor merupakan alat untuk memindahkan material secara efisien dari satu lokasi ke lokasi lain dan berperan penting dalam berbagai industri. Sistem konveyor umumnya terdiri dari sabuk yang digerakkan oleh dua katrol dan didukung oleh serangkaian rol untuk menjaga kestabilan. Kinerja konveyor sangat berpengaruh terhadap kapasitas produksi, di mana ketidakefisienan kecil sekalipun dapat menyebabkan penundaan operasional dan peningkatan biaya (Golwa dkk., 2024).



Gambar 2. Konveyor

1.2.3 *Computer Vision*

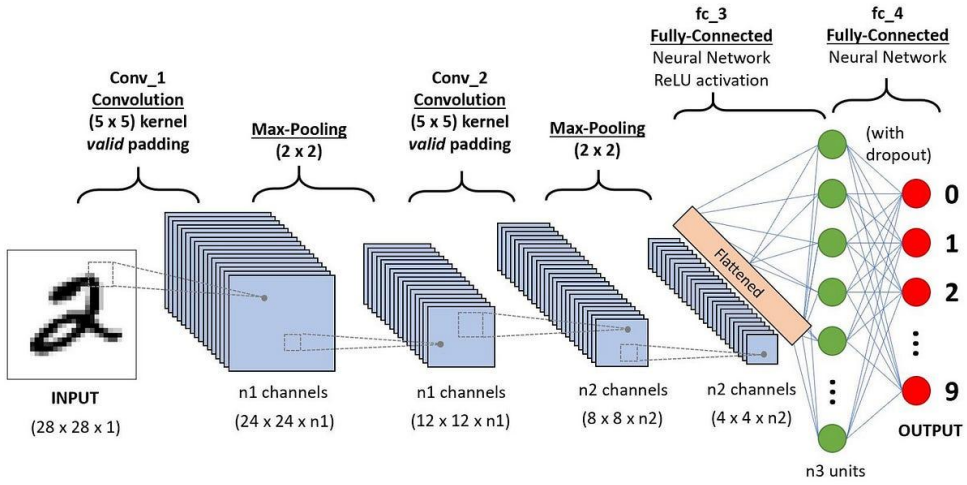
Computer vision adalah bidang kecerdasan buatan yang bertujuan memungkinkan mesin memahami citra digital atau video dengan meniru sistem visual manusia. Bidang ini bersifat interdisipliner dan memanfaatkan perkembangan *machine learning* untuk meningkatkan kemampuan analisis visual (Sinha dkk., 2023). Tiga tugas utama dalam *computer vision* adalah klasifikasi, deteksi, dan segmentasi. Deteksi objek tidak hanya mengidentifikasi jenis objek, tetapi juga menentukan lokasi objek melalui *bounding box* (Khadem dkk., 2026).

1.2.4 Deteksi Objek

Deteksi objek adalah tugas fundamental dalam visi komputer yang menggabungkan dua tugas, lokalisasi dan klasifikasi objek dalam citra atau video. Lokalisasi ditandai dengan menggambar kotak batas (*bounding box*) di sekitar setiap objek, sedangkan klasifikasi menetapkan label kategori untuk objek tersebut. Arsitektur untuk deteksi objek umumnya terdiri dari tiga komponen yaitu *backbone* untuk ekstraksi fitur, *neck* untuk penggabungan fitur multi-skala, dan *head* untuk memprediksi kotak batas beserta skor klasifikasi. Metode deteksi modern terbagi menjadi dua jenis yaitu, *two-stage detector* yang memisahkan langkah proposal wilayah dan klasifikasi (misalnya R-CNN dan penerusnya Faster R-CNN), kemudian jenis lain yaitu *one-stage detector* yang menggabungkan kedua tugas tersebut secara *end-to-end* (misalnya YOLO, dan SSD) (Zaidi dkk., 2022).

Sejak diperkenalkannya R-CNN pada 2014, penelitian deteksi objek telah mengalami evolusi signifikan. Keluarga *two-stage* menawarkan akurasi tertinggi dengan memanfaatkan proposal wilayah yang dihasilkan secara eksplisit, namun dengan biaya komputasi lebih tinggi. Sementara itu, *one-stage detectors* mengutamakan kecepatan inferensi, menjadikannya pilihan favorit untuk aplikasi *real-time* meskipun akurasi terkadang sedikit lebih rendah. Tren terkini meliputi penggunaan arsitektur *transformer* untuk meningkatkan konteks global, mekanisme *attention* untuk menyorot fitur krusial, serta desain model yang ringan untuk perangkat *edge*, sehingga deteksi objek dapat diimplementasikan di sistem terbatas sumber daya tanpa mengorbankan kinerja secara signifikan (Zou dkk., 2023).

1.2.5 *Convolutional Neural Network*

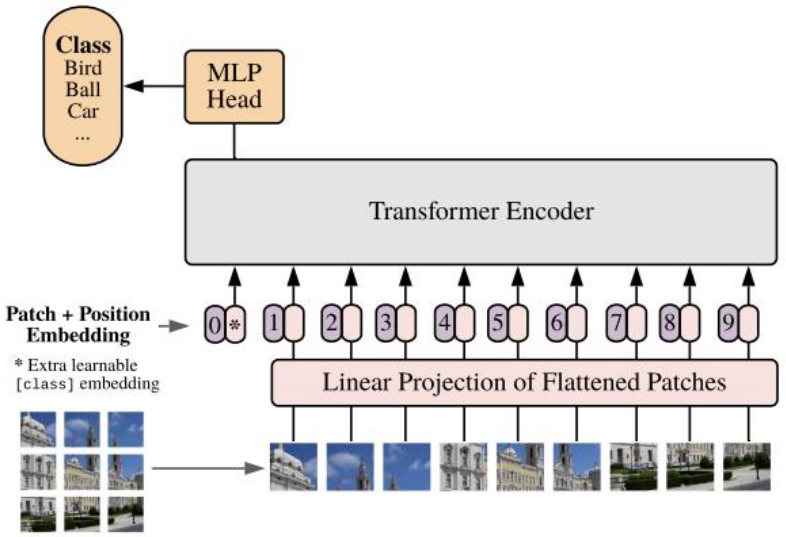


Gambar 3. Ilustrasi CNN

Convolutional Neural Network adalah metode pembelajaran mendalam yang mengambil gambar masukan dan menetapkan kepentingan (bias dan bobot yang

dapat dipelajari) ke berbagai objek dalam gambar, membedakan satu dari yang lain. Arsitektur CNN terdiri dari 3 bagian penting yang meliputi *convolution layer*, *pooling layer*, dan *fully connected layer*. Dengan *convolution layer*, fitur lokal akan diekstraksi dengan operasi *convolutional filter* pada citra input, kemudian peta fitur yang diperoleh akan ditransfer ke *convolution layer* berikutnya untuk mengekstrak fitur yang lebih tinggi. Selanjutnya, *pooling layer* akan berfungsi untuk mengurangi dimensi citra setelah *convolutional layer*. *Fully connected layer* yang terletak setelah *convolutional layer* terakhir, meratakan peta fitur menjadi vektor. Biasanya, fungsi *softmax* digunakan pada output layer untuk menormalkan output ke dalam bentuk probabilitas. Probabilitas ini menunjukkan bahwa citra diklasifikasikan ke dalam kategori tertentu (Chen dkk., 2025).

1.2.6 Vision Transformer



Gambar 4. Ilustrasi ViT

Vision Transformer (ViT) merupakan pengembangan dari arsitektur *transformer* yang awalnya digunakan dalam bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) dan kemudian diadaptasi untuk pengolahan citra. Berbeda dengan jaringan konvolusional (CNN) yang bekerja dengan mengekstraksi fitur lokal melalui operasi konvolusi, ViT menggunakan mekanisme *self-attention* untuk menangkap hubungan global antar bagian gambar. Dalam model ini, gambar dua dimensi dibagi menjadi potongan-potongan kecil yang disebut patch, yang kemudian diubah menjadi vektor berdimensi tetap melalui proyeksi linear. Setiap patch ini diperlakukan seperti “kata” dalam kalimat, sehingga seluruh gambar dapat diproses sebagai *sequence* oleh *encoder transformer*. Agar model memahami posisi spasial tiap *patch*, ViT menambahkan *positional embedding* setiap representasi *patch* (Han dkk., 2023).

1.2.7 RT-DETR

RT-DETR merupakan pengembangan dari arsitektur DETR (*Detection Transformer*) yang dirancang untuk meningkatkan efisiensi dan akurasi dalam tugas deteksi objek dengan memanfaatkan mekanisme *transformer* secara *end-to-end* tanpa memerlukan proses *Non-Maximum Suppression* (NMS). Model ini menggabungkan keunggulan *transformer* dalam menangkap hubungan global antar fitur gambar dengan rancangan *hybrid encoder* yang efisien. Selain itu, RT-DETR memperkenalkan metode *uncertainty-minimal query selection* untuk menghasilkan inisialisasi kueri yang lebih berkualitas dan meningkatkan performa keseluruhan model. Dengan rancangan ini, RT-DETR mampu mencapai keseimbangan antara efisiensi komputasi dan ketepatan deteksi (Zhao dkk., 2024).

Cara kerja RT-DETR dimulai dengan mengekstraksi fitur dari beberapa tahap akhir *backbone*, yang kemudian diproses oleh *efficient hybrid encoder*. Yang terdiri atas *Attention-based Intra-scale Feature Interaction* (AIFI) dan *CNN-based Cross-scale Feature Fusion* (CCFF). AIFI bertugas mempelajari hubungan antar fitur dalam satu skala, terutama pada fitur tingkat tinggi yang mengandung informasi konseptual penting, sementara CCFF menggabungkan informasi dari berbagai skala untuk memperkaya representasi dan memperkuat konteks objek. Hasil penggabungan ini merupakan urutan fitur gambar yang kemudian digunakan dalam tahap *uncertainty-minimal query selection*, di mana sejumlah fitur *encoder* dengan tingkat ketidakpastian terendah antara prediksi klasifikasi dan lokalisasi dipilih sebagai *object queries*. Tahap ini memastikan bahwa hanya fitur dengan kualitas tinggi yang digunakan untuk inisialisasi kueri pada *decoder*, sehingga proses optimisasi menjadi lebih stabil dan akurat. Selanjutnya, *decoder* menghasilkan prediksi akhir berupa kategori dan koordinat kotak pembatas objek (Zhao dkk., 2024).

1.2.8 *Confusion Matrix*

Confusion Matrix merupakan alat evaluasi yang memberikan gambaran menyeluruh mengenai performa suatu model klasifikasi. Matriks ini berbentuk tabel persegi berukuran N × N yang menampilkan hubungan antara hasil prediksi model dengan nilai aktual. Elemen-elemen penting dari *confusion matrix* mencakup *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN), yang masing-masing menggambarkan hasil klasifikasi benar maupun salah pada kelas positif dan negatif. Melalui informasi ini, peneliti dapat menilai sejauh mana model berhasil membedakan kelas secara benar serta mengidentifikasi jenis kesalahan yang paling sering terjadi (Sathyanarayanan & Tantri, 2024).

***Accuracy*.** Akurasi digunakan untuk mengukur proporsi total prediksi yang benar yang dibuat oleh model di antara semua kasus yang dievaluasi.

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

***Precision*.** Presisi adalah metrik yang berguna mencari tahu dari seberapa banyak prediksi positif yang nilainya benar-benar positif.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

***Recall*.** Sensitivitas adalah metrik yang berguna mencari tahu seberapa banyak kasus positif yang berhasil diprediksi positif.

$$recall = \frac{TP}{TP + FN} \tag{3}$$

1.2.9 *Mean Average Precision* (mAP)

Mean Average Precision (mAP) merupakan metrik evaluasi utama pada sistem deteksi objek yang mengukur seberapa baik model mengenali dan melokalisasi objek target. Nilai mAP dihitung sebagai rata-rata dari *Average Precision* (AP) setiap kelas, di mana AP adalah luas area di bawah kurva *Precision–Recall*. mAP50 menilai akurasi deteksi ketika tumpang tindih antara prediksi dan *ground truth* minimal 0.5, sedangkan mAP50-95 menghitung rata-rata mAP pada ambang IoU 0.5 hingga 0.95 dengan langkah 0.05. Semakin tinggi nilai mAP, semakin baik kinerja model secara keseluruhan dalam mengklasifikasikan dan melokalisasi objek (Li dkk., 2023).

$$mAP = \frac{1}{N}\sum_{i=1}^{n} AP_i \tag{4}$$

1.3 Rumusan Masalah

Penelitian ini memiliki rumusan masalah antara lain:

1. Bagaimana cara menerapkan RT-DETR untuk mendeteksi biji kopi cacat pada *conveyor belt*?
2. Bagaimana performa RT-DETR mendeteksi biji kopi cacat pada *conveyor belt*?

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Penelitian ini memiliki tujuan antara lain:

1. Menerapkan model RT-DETR untuk mendeteksi biji kopi cacat pada *conveyor belt*
2. Menganalisa performa model RT-DETR dalam mendeteksi biji kopi cacat pada *conveyor belt*

1.4.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Membantu perkembangan sistem *monitoring* otomatis atau teknologi pengolahan sumber daya alam, khususnya dalam proses penyortiran biji kopi sehingga mampu meningkatkan kualitas dan nilai jual komoditas
2. Menjadi referensi untuk penelitian *object detection* berbasis *neural network* dan *transformers*, terutama untuk objek bergerak.

1.5 Batasan Masalah

Penelitian ini memiliki batasan antara lain:

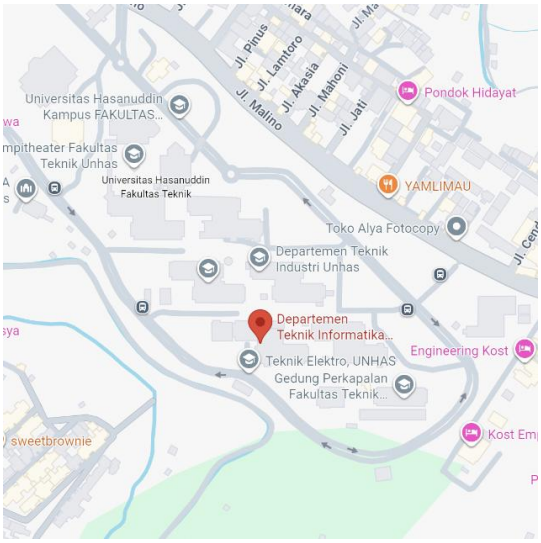
1. Kelas *defect* merupakan biji kopi yang menunjukkan ciri-ciri cacat, sementara *non_defect* merupakan biji kopi yang tidak menunjukkan adanya ciri-ciri cacat, sesuai ketentuan SNI 01-2907-2008.
2. Jarak antara kamera dan alas konveyor adalah 10 cm & 15 cm.
3. Variasi kecepatan konveyor yaitu 35 RPM, 50 RPM, dan 70 RPM.

BAB II

METODE PENELITIAN

2.1 Tempat dan Waktu

Penelitian dilakukan mulai sejak disetujuinya proposal penelitian pada Mei hingga November 2025. Penulis melakukan penelitian ini di Laboratorium *Artificial Intelligence*, Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.



Gambar 5. Lokasi Departemen Teknik Informatika

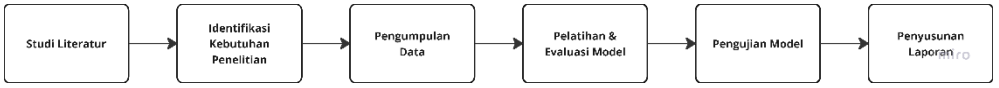
2.2 Benda Uji dan Alat

Benda dan alat yang digunakan dalam penelitian ini meliputi perangkat lunak dan perangkat keras berikut:

1. Perangkat lunak
   a. Windows 11 64-bit
   b. Google Colab
   c. Kaggle GPU P100
   d. Roboflow
   e. Microsoft Word
   f. Microsoft Powerpoint
   g. Microsoft Edge
   h. Zotero
   i. Canva
2. Perangkat keras
   a. ASUS Vivobook S14 OLED K3402ZA, Intel Core i5-12500H, Intel Iris Xe Graphics, RAM 12GB, SSD 477GB

   b. *Conveyor Belt*
   c. Brio 4K
3. Bahasa pemrograman
   a. Python v3.12.12
4. Library
   a. albumentations 2.0.8 berguna untuk augmentasi gambar dan pengalihan format.
   b. cv2 berguna untuk mengolah *frame* pada video.
   c. matplotlib 3.10.0 berguna untuk visualisasi data.
   d. numpy 2.0.2 berguna untuk melakukan kalkulasi *confusion matrix*.
   e. pandas 2.2.2 berguna untuk mengimpor atau mengekspor data tabular berbagai format.
   f. pillow 11.3.0 berguna untuk membuka file gambar.
   g. roboflow 1.2.11 berguna untuk mendapatkan dataset dari Roboflow.
   h. torch 2.8.0+cu126 berguna untuk menjalankan proses *training* dan *testing* pada GPU.
   i. torchmetrics 0.11.0 berguna untuk menghitung mAP validasi pada setiap *epoch training*.
   j. transformers 4.57.1 berguna untuk inisialisasi model deteksi objek.
   k. supervision 0.26.1 berguna untuk kalkulasi metrik model dan pengalihan format dataset.
   l. time berguna untuk mengukur FPS.

2.3 Tahapan Penelitian



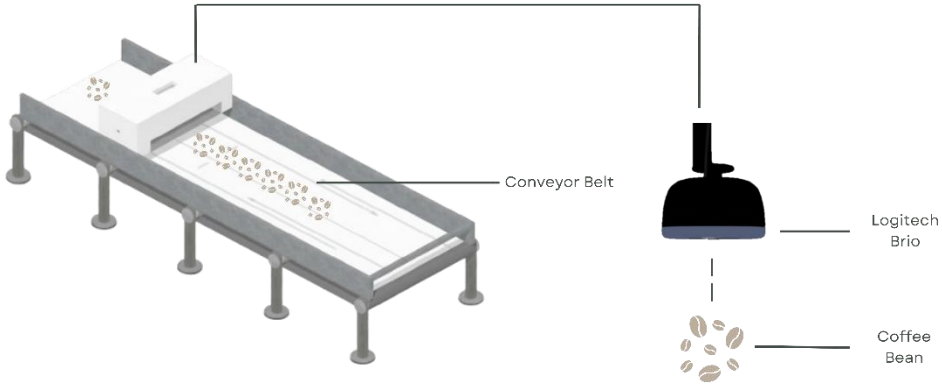
Gambar 6. Tahapan Penelitian

Tahapan penelitian ini ditunjukkan pada Gambar 6. Diawali studi literatur dengan mencari informasi relevan mengenai topik penelitian. Dari informasi yang dikumpulkan, kemudian dilakukan identifikasi kebutuhan penelitian untuk menentukan permasalahan, tujuan, serta data dan metode yang digunakan. Selanjutnya, tahap pengumpulan data dengan cara merekam biji kopi di atas konveyor yang sedang berjalan, kemudian ekstraksi *frame* serta anotasi lalu pembagian menjadi data *training*, validasi, dan *testing*. Kemudian, dilakukan pelatihan model dengan memanfaatkan data *training* dimanfaatkan untuk melatih model, serta data validasi untuk mencari tahu pengaruh nilai *hyperparameter* terhadap hasil deteksi, proses pelatihan dan evaluasi dilakukan berulang untuk mendapatkan nilai terbaik. Tahap selanjutnya yaitu pengujian, yang melibatkan data *testing* untuk membandingkan hasil deteksi pada berbagai skenario. Akhirnya, dari proses tersebut dapat dilakukan analisis serta pengambilan kesimpulan untuk penyusunan laporan.

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini sepenuhnya merupakan data primer, yang diperoleh melalui tahapan-tahapan berikut.

2.4.1 *Data collection*



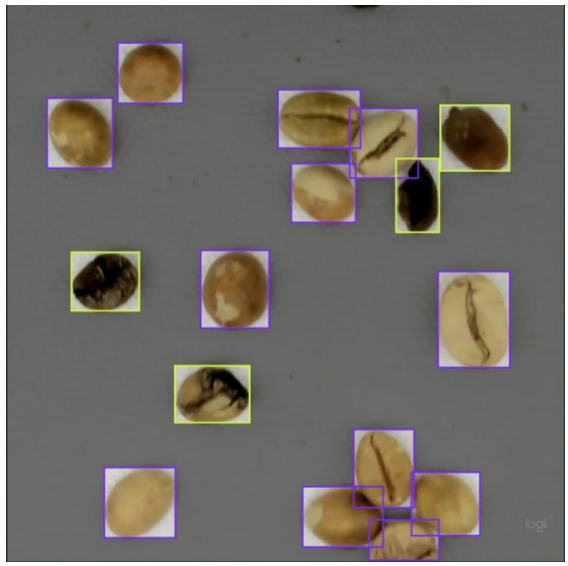
Gambar 7. Ilustrasi Pengambilan Data

Pengambilan data dilakukan dengan merekam biji kopi yang terletak di atas konveyor yang sedang berjalan. Perekaman video dilakukan dengan 2 variasi jarak antara *webcam* dengan alas konveyor, yaitu 10 cm dan 15 cm. Sementara itu, konveyor dijalankan dengan 3 variasi kecepatan, yaitu 35 RPM, 50 RPM dan 70 RPM. Hasil dari proses ini berupa video rekaman yang berformat .mp4 yang kemudian akan diambil beberapa *frame*. Berikut merupakan salah satu *frame* yang mewakili skenario dengan jarak 10 cm serta kecepatan 35 RPM.

Gambar 8. Frame pada jarak 10 cm dan kecepatan 35 RPM

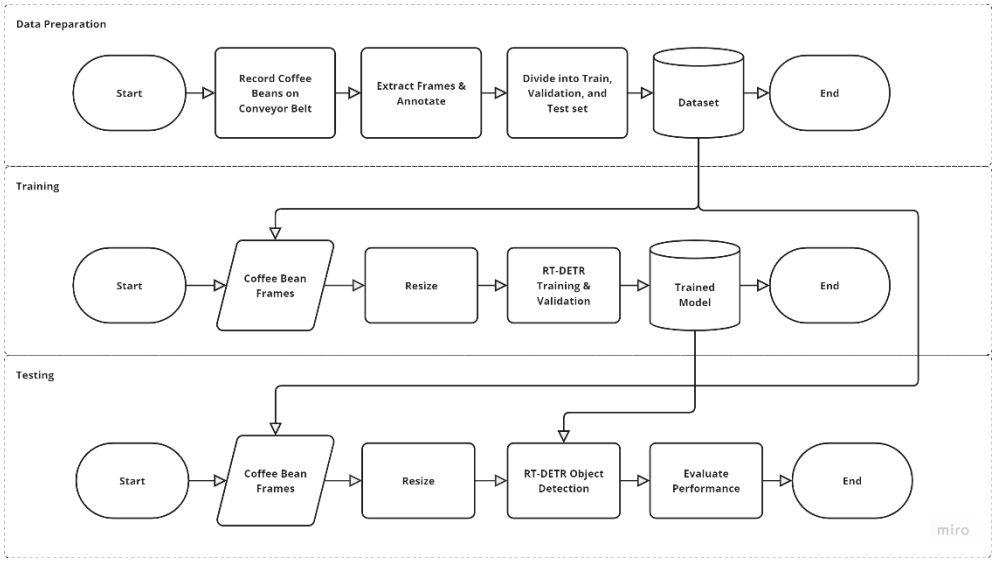
2.4.2 *Data annotation*

Frame yang telah dikumpulkan kemudian dilanjutkan proses anotasi biji-biji kopi ke dalam salah satu kelas. Kelas *defect* jika menunjukkan adanya ciri-ciri cacat dan *non_defect* tidak menunjukkan ciri-ciri cacat sesuai dengan ketentuan SNI 01-2907-2008. Berikut contoh frame yang telah ternaotasi.



Gambar 9. Frame yang telah teranotasi

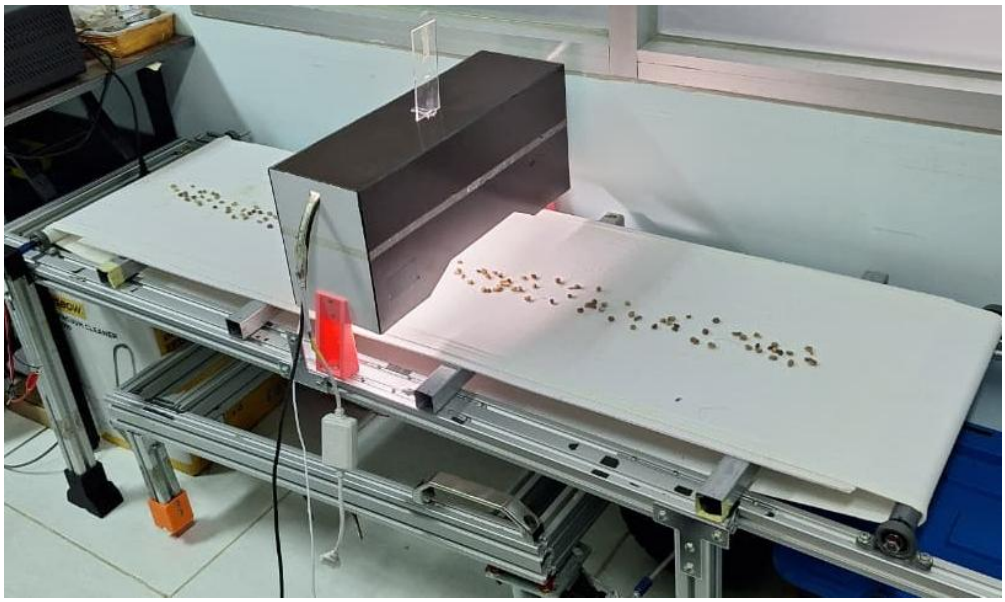
2.5 Perancangan dan Implementasi Sistem



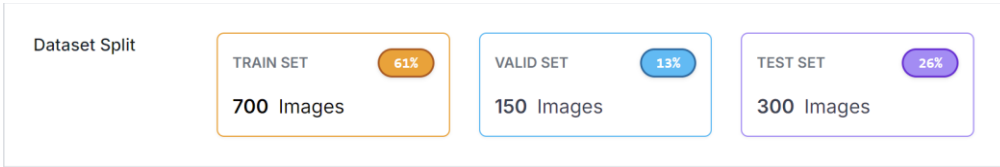
Gambar 10. Rancangan Sistem

2.5.1 *Data preparation*

Penelitian ini dimulai dari tahap *data preparation*, pertama dilakukan proses perekaman biji kopi di atas konveyor yang sedang berjalan, video rekaman juga akan terdiri dari berbagai skenario. Setelah itu, video rekaman kemudian diekstrak menjadi *frame* lalu dilakukan anotasi objek. Tahap ini diakhiri dengan pembagian *frame* ke dalam subdataset yaitu *training*, *validation*, dan *testing*.



Gambar 12. Dokumentasi pengambilan data



Gambar 11. Pembagian dataset

2.5.2 *Training*

1. *Coffee Bean Frames*

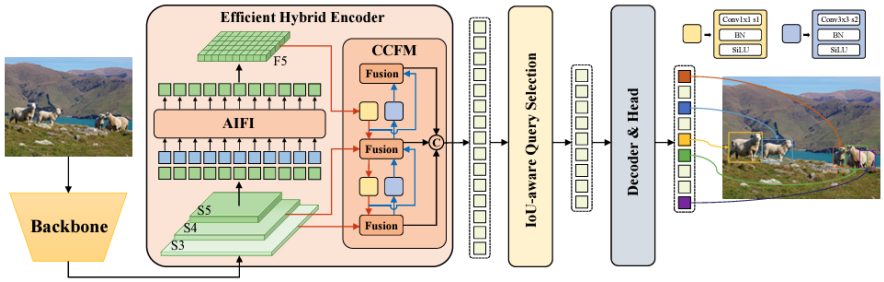
   Data *training* dan *validation* berperan dalam dalam proses *training*. Data *training* akan digunakan untuk mempelajari pola-pola objek, kemudian data *validation* digunakan untuk melihat performa model terhadap objek yang belum pernah terlihat oleh model pada setiap akhir *epoch*. Jumlah data *training* yaitu 700 frame dengan jumlah *validation* sebanyak 150 *frame*.

2. *Resize*

   Resize merupakan satu-satunya *preprocessing* pada penelitian ini. Sebelum *frame* dijadikan input ke dalam model, akan dilakukan proses *resize*. *Frame* yang berukuran 1080 × 1080 diperkecil menjadi 480 × 480. Pemilihan ukuran didasarkan atas dimensi citra yang mampu diterima oleh model deteksi, sementara *aspect ratio* 1:1 dipilih karena lebih fleksibel untuk mengikuti kebutuhan spesifikasi citra oleh model.

3. RT-DETR *Training* & *Validation*

   RT-DETR adalah model deteksi objek end-to-end yang merupakan pengembangan DETR (*Detection Transformer*). Model ini berbasis CNN dan juga Transformer di dalam arsitekturnya yang juga terdiri dari *backbone*, *encoder*, *query selection*, dan *decoder*.



Gambar 13. Arsitektur RT-DETR

Backbone. Model yang berperan sebagai backbone pada RT-DETR adalah ResNet. RT-DETR memiliki beberapa versi yang menggunakan ResNet yang berbeda-beda, penelitian ini menggunakan ResNet-18 karena ukurannya yang ringan, namun dengan akurasi yang tidak berbeda dari ResNet lainnya. Fungsi ResNet adalah untuk melakukan ekstraksi fitur. Dalam ResNet terdapat lima blok. Peta fitur yang dilanjutkan menuju tahap selanjutnya adalah S3, S4, dan S5 yang diambil dari blok ketiga, keempat, dan kelima.

Encoder. *Efficient Hybrid Encoder* pada model RT-DETR bekerja dengan cara meneruskan S5 ke dalam AIFI (*Attention-based Intra-scale Feature Interaction*) yang mencari informasi melalui hubungan antar fitur dari suatu skala, mengubah S5 menjadi F5. S5 dipilih karena memiliki fitur tingkat tinggi karena telah mengandung konsep penting dari citra. Selanjutnya, CCFF (CNN-*based Cross-scale Feature Fusion*) menggabungkan informasi yang terdapat pada F5 digabungkan dengan S3, dan S4 yang menghasilkan informasi yang lebih mendalam mengenai citra. Output yang dihasilkan dari komponen ini akan berupa sekuens fitur *encoder*.

Query selection. RT-DETR menggunakan *Uncertainty-minimal query selection* untuk meringkas ukuran sekuens. Setiap fitur akan diprediksi distribusi lokasi dan kelasnya yang kemudian dicari selisihnya di antara satu sama lain. Semakin kecil selisihnya semakin yakin pula model terhadap prediksinya, sehingga fitur yang diprioritaskan merupakan fitur-fitur dengan nilai selisih minimum.

Decoder. *Decoder* & *Head* bertugas untuk mengalihkan sekuens yang telah diringkas menjadi prediksi model terhadap gambar. Prediksi yang dimaksud berupa lokalisasi dan klasifikasi objek pada gambar. Komponen ini merupakan bagian terakhir dari RT-DETR.

Proses pelatihan dijalankan dengan memanfaatkan situs Kaggle yang menyediakan akses sumber daya GPU guna efisiensi *training* model. Adapun nilai *hyperparameter* yang digunakan yaitu sebagai berikut:

Tabel 1. *Hyperparameter*

Hyperparameter	**Nilai**
Resize	480 × 480
Epoch	150
Learning rate	0.0002
Batch size	32

Terdapat beberapa faktor yang menjadi alasan dari pemilihan nilai-nilai *hyperparameter* diatas. *Hyperparameter* yang pertama adalah *resize* dengan ukuran 480 x 480 yang didasari efisiensi sumber daya dalam proses pelatihan model serta karena ukuran tersebut memberikan performa yang tidak berbeda jika dibandingkan ukuran yang lebih besar yaitu 640 x 640. *Resize* merupakan satu-satunya *hyperparameter* yang berperan dalam proses *preprocessing* citra.

Hyperparameter kedua, yaitu *epoch* yang diberikan nilai 150 karena proses pelatihan menunjukkan performa yang setara terhadap data training dan validasi pada *epoch* tersebut. Model belum mengalami konvergensi pada *epoch* 100 dan mengalami overfitting pada *epoch* 200. Selanjutnya adalah lerning rate sebesar 0.0002 yang dipastikan memiliki nilai kecil untuk menjaga kestabilan proses pembelajaran model.

Hyperparameter yang memberikan pengaruh terbesar yaitu *batch size* yang diberikan nilai 32. Pemberian nilai lebih besar akan membuat model mempelajari pola objek pada gambar yang lebih banyak secara sekaligus. Dengan begitu model akan mengenali detail yang relevan dalam penentuan kelas karena mendapatkan variasi citra yang lebih baik dibanding ukuran yang lebih kecil seperti 16.

4. *Trained Model*

   Keluaran dari tahap ini adalah model RT-DETR yang telah mempelajari pola-pola pada objek yang disimpan dalam format .safetensors, selain itu juga file config.json yang menyimpan konfigurasi *preprocessing* citra. Model ini akan diteruskan menuju tahap *testing* untuk dilakukan evaluasi serta analisis performa.

2.5.3 *Testing*

1. *Coffee Bean Frames*

   Tahap testing menggunakan data yang berbeda dari data *training*. Pada tahap ini, data *testing* digunakan untuk mengukur performa deteksi oleh model terhadap biji kopi yang belum pernah terlihat di beberapa skenario yang berbeda dengan jumlah *frame* sebagai berikut:

Tabel 2. Skenario

Skenario ke-	Ketinggian	Kecepatan	Jumlah *frame*
1	10 cm	35 RPM	50
2	10 cm	50 RPM	50
3	10 cm	70 RPM	50
4	15 cm	35 RPM	50
5	15 cm	50 RPM	50
6	15 cm	70 RPM	50

2. *Resize*

   Resize pada tahap *testing* akan berpedoman pada proses *training*, yaitu dengan menurunkan ukuran *frame* dari yang mulanya 1080 × 1080 menjadi 480 × 480. Penurunan ukuran pada citra memungkinkan proses pengolahan citra oleh model menjadi lebih cepat serta hemat daya komputasi sehingga seluruh proses menjadi lebih efisien.

3. RT-DETR *Object Detection*

   Frame yang telah di-*resize* akan menjadi input RT-DETR. Berbeda dengan tahap *training*, kali ini model tidak akan mempelajari pola objek lagi melainkan akan mencari lokasi dan kelas dari objek yang terdapat di *frame*. Proses ini akan berjalan selama enam kali sesuai dengan jumlah skenario.

4. *Confidence Score Comparison*

   Pada tahap ini, dilakukan perbandingan performa model jika diberikan *threshold* dari *confidence score* pada *post-processing*. Penentuan threshold berguna untuk melakukan penyaringan hasil deteksi yang tidak relevan. Berikut merupakan tabel performa model jika diberikan berbagai *threshold*:

Tabel 3. Perbandingan performa terhadap threshold untuk confidence score

Threshold	**mAP**	***Accuracy***
0.1	0.96	0.91
0.2	0.94	0.94
0.3	0.93	0.95
0.4	0.92	0.96
0.5	0.90	0.97
0.6	0.86	0.98
0.7	0.75	0.98
0.8	0.44	0.99
0.9	0.04	1.00

Tabel di atas menunjukkan pengaruh *threshold* terhadap mAP dan juga *accuracy*. Terlihat bahwa semakin tinggi *threshold*, maka nilai *accuracy* juga ikut meningkat, namun dengan nilai mAP yang semakin menurun. Hal ini dikarenakan hanya deteksi yang setara atau berada di atas *threshold* yang dianggap sebagai objek, jadi peningkatan *threshold* juga akan mengurangi jumlah deteksi. Karena hanya deteksi dengan *confidence score* tinggi yang dianggap sebagai objek, membuat nilai misklasifikasi menurun sehingga meningkatkan nilai *accuracy*. Maka pada penelitian ini akan menerapkan *threshold* pada 0.3 agar model memiliki performa klasifikasi yang lebih tinggi, dengan mAP yang cukup baik.

5. *Evaluate performance*

Tahap ini merupakan tahap terakhir dari fase *testing*, dimana telah diperoleh metrik evaluasi yang dihasilkan oleh percobaan deteksi objek oleh model. Metrik yang akan menjadi bahan evaluasi yaitu *accuracy*, *precision*, *recall*, serta mAP50, dengan setiap skenario akan memiliki nilainya masing-masing. Nilai-nilai tersebut berguna untuk proses evaluasi model serta pengambilan kesimpulan penelitian.