

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian mengenai deteksi dan klasifikasi kualitas biji kopi berbasis citra digital terus berkembang seiring kemajuan teknologi *deep learning* dan *computer vision*. Berbagai penelitian dilakukan untuk mengatasi keterbatasan proses inspeksi manual yang masih bersifat subjektif, membutuhkan waktu lama, dan menghasilkan penilaian yang tidak konsisten.

Penelitian oleh Chang dan Liu [13] membahas permasalahan deteksi cacat biji kopi yang masih sulit dilakukan secara akurat, terutama pada objek berukuran kecil dan *multi-defect*. Penelitian tersebut menggunakan metode *Multiscale Defect Extraction Neural Network* berbasis *CNN* dengan teknik *multi-scale* feature fusion. Hasil penelitian menunjukkan bahwa model mampu memperoleh akurasi hingga 96% dan memberikan performa lebih baik dibandingkan *CNN* konvensional dalam mendeteksi cacat biji kopi.

Penelitian lain oleh Heryanto dan Nugraha [14] mengangkat permasalahan inspeksi cacat biji kopi yang masih dilakukan secara manual dan subjektif. Penelitian tersebut menerapkan metode *Mask R-CNN* untuk melakukan *objek deteksi* dan *instance segmentation* pada biji kopi. Hasil penelitian menunjukkan bahwa *Mask R-CNN* mampu mendeteksi dan mengklasifikasikan cacat biji kopi dengan performa deteksi yang baik, selanjutnya, Diloy et al. [15] meneliti proses otomatisasi deteksi biji kopi untuk meningkatkan efisiensi *quality control* industri kopi. Penelitian tersebut menggunakan metode *Mask R-CNN* dengan pendekatan *instance segmentation* untuk memisahkan objek biji kopi dalam satu citra. Hasil penelitian menunjukkan bahwa model mampu melakukan segmentasi dan deteksi objek secara akurat pada kondisi citra yang kompleks, penelitian oleh Santoso et al. [16] membahas permasalahan identifikasi cacat morfologi biji kopi yang masih tidak konsisten apabila dilakukan secara manual. Penelitian ini menggunakan

metode *Convolutional Neural Network (CNN)* untuk melakukan klasifikasi cacat morfologi berdasarkan citra digital. Hasil penelitian menunjukkan bahwa *CNN* mampu mengenali pola cacat biji kopi dengan tingkat klasifikasi yang baik.

Pada penelitian oleh Ji, Xu, dan Yan [17], dibahas bahwa model objek deteksi sebelumnya masih kurang optimal dalam mendeteksi *defect* kecil pada *green coffee bean*. Penelitian tersebut menggunakan metode Improved *YOLOv8* untuk meningkatkan performa deteksi objek. Hasil penelitian menunjukkan adanya peningkatan nilai *precision*, *recall*, dan *mean Average Precision (mAP)* dibandingkan model *YOLOv8* standar, penelitian lain oleh Adiwijaya et al. [18] membahas tantangan deteksi *defect* kecil pada biji kopi secara *real-time*. Penelitian ini menggabungkan metode *YOLO* dengan *framework SAHI (Slicing Aided Hyper Inference)* untuk meningkatkan kemampuan deteksi objek kecil. Hasil penelitian menunjukkan bahwa kombinasi *YOLO* dan *SAHI* mampu meningkatkan akurasi deteksi *defect* dengan performa *real-time* yang baik.

Talunga et al. [19] melakukan penelitian mengenai deteksi cacat biji kopi pada mesin conveyor untuk mendukung proses sortasi otomatis di industri kopi. Penelitian tersebut menggunakan metode *Mask R-CNN* untuk melakukan deteksi objek secara *real-time*. Hasil penelitian menunjukkan bahwa sistem mampu mendeteksi cacat biji kopi dengan akurasi yang baik pada lingkungan *conveyor*, selain itu, Adiwijaya, Sarno, dan Wijaya [20] membahas permasalahan inspeksi *manual green coffee bean* yang kurang efisien pada produksi skala besar. Penelitian tersebut menggunakan arsitektur *YOLO* berbasis *objek deteksi* untuk mendeteksi *defect* pada citra biji kopi. Hasil penelitian menunjukkan bahwa model *YOLO* mampu melakukan deteksi *defect* secara cepat dan akurat.

Penelitian oleh Muchtar [21] membahas keterbatasan sistem *AI* berbasis *cloud* yang masih memiliki *latency* tinggi dan ketergantungan terhadap koneksi internet. Penelitian tersebut menggunakan pendekatan *Edge AI* berbasis *deep learning* dan *Streamlit* untuk membangun sistem deteksi *defect* biji kopi secara ringan dan *real-time*. Hasil penelitian menunjukkan bahwa sistem mampu melakukan deteksi objek dengan performa yang baik dan antarmuka yang interaktif, sementara itu, penelitian

oleh Febriana et al. [22] membahas keterbatasan *dataset* multi-class biji kopi untuk kebutuhan *deep learning*. Penelitian tersebut mengembangkan *USK-Coffee Dataset* yang terdiri dari beberapa kelas biji kopi Arabika seperti *premium*, *peaberry*, *longberry*, dan *defect*. Hasil penelitian menunjukkan bahwa *dataset* tersebut dapat digunakan sebagai *benchmark* untuk penelitian klasifikasi dan *objek deteksi* pada biji kopi.

Tabel 2.1 Tabel Penelitian Terdahulu

No	Peneliti	Tahun	Masalah Penelitian	Metode	Hasil Penelitian	Kelemahan Penelitian/ <i>Research gap</i>
1	Chang & Liu [13]	2024	Deteksi cacat biji kopi masih sulit dilakukan secara akurat, terutama pada objek kecil dan <i>multi-defect</i> .	<i>Multiscale Defect Extraction Neural Network</i> berbasis <i>CNN</i> dengan <i>multi-scale feature fusion</i> .	Model memperoleh akurasi hingga 96% dan mampu meningkatkan deteksi cacat dibanding metode <i>CNN</i> konvensional.	Berfokus pada klasifikasi berbasis <i>CNN</i> dan tidak membandingkan performa dengan model objek deteksi modern seperti <i>YOLO</i> , <i>Faster R-CNN</i> , dan <i>SSD</i> .
2	Heryanto & Nugraha [14]	2022	Proses identifikasi cacat biji kopi secara manual masih subjektif dan memerlukan waktu lama.	<i>Mask R-CNN</i> untuk <i>objek deteksi</i> dan <i>instance segmentation</i> .	<i>Mask R-CNN</i> mampu mendeteksi serta mengklasifikasi cacat biji kopi dengan performa deteksi yang baik.	Hanya menggunakan satu model sehingga tidak diketahui bagaimana performanya dibandingkan model <i>one-stage detector</i> yang lebih cepat.
3	Diloy et al. [15]	2024	Deteksi biji kopi otomatis diperlukan untuk meningkatkan efisiensi <i>quality control</i> industri kopi.	<i>Mask R-CNN</i> dengan pendekatan <i>instance segmentation</i> .	Model berhasil melakukan segmentasi dan deteksi objek biji kopi secara akurat pada kondisi citra kompleks.	Fokus pada segmentasi objek dan tidak mengevaluasi <i>trade-off</i> antara akurasi dan kecepatan inferensi dengan model lain.
4	Santoso et al. [16]	2025	Pemeriksaan cacat morfologi biji kopi secara manual masih tidak konsisten.	<i>Convolutional Neural Network (CNN)</i> untuk klasifikasi cacat morfologi.	<i>CNN</i> mampu mengenali pola cacat biji kopi dengan tingkat klasifikasi yang baik.	Menggunakan pendekatan klasifikasi citra, bukan objek deteksi, sehingga lokasi objek cacat tidak dapat diidentifikasi secara langsung.
5	Ji, Xu, & Yan [17]	2024	Model <i>objek deteksi</i> sebelumnya	<i>Improved YOLOv8</i> untuk <i>objek deteksi</i> .	Model meningkatkan nilai <i>precision</i> ,	Hanya berfokus pada pengembangan

No	Peneliti	Tahun	Masalah Penelitian	Metode	Hasil Penelitian	Kelemahan Penelitian/Research gap
			masih kurang optimal dalam mendeteksi <i>defect</i> kecil pada <i>green coffee bean</i> .		<i>recall</i> , dan <i>mAP</i> dibanding <i>YOLOv8</i> standar.	<i>YOLOv8</i> dan belum membandingkannya dengan arsitektur berbeda seperti <i>Faster R-CNN</i> dan <i>SSD</i>
6	Adiwijaya et al. [18]	2024	Deteksi <i>defect</i> kecil pada biji kopi <i>real-time</i> masih menjadi tantangan.	<i>YOLO</i> dikombinasikan dengan SAHI (Slicing Aided Hyper Inference).	Kombinasi <i>YOLO</i> dan <i>SAHI</i> meningkatkan kemampuan deteksi objek kecil dengan performa <i>real-time</i> yang baik.	Evaluasi terbatas pada keluarga <i>YOLO</i> sehingga belum diketahui efektivitasnya dibandingkan model objek deteksi lain.
7	Talunga et al. [19]	2024	Proses sortasi biji kopi pada conveyor membutuhkan deteksi otomatis yang cepat dan akurat.	<i>Mask R-CNN</i> pada conveyor machine.	Sistem mampu mendeteksi cacat biji kopi secara <i>real-time</i> dengan akurasi yang baik.	Tidak melakukan perbandingan dengan model yang memiliki kecepatan inferensi lebih tinggi seperti <i>YOLO</i> atau <i>SSD</i> .
8	Adiwijaya, Sarno, & Wijaya [20]	2024	Inspeksi manual <i>green coffee bean</i> kurang efisien untuk produksi skala besar.	<i>YOLO-based</i> objek deteksi menggunakan citra <i>green bean</i> .	Model <i>YOLO</i> efektif digunakan untuk deteksi <i>defect</i> secara cepat dan akurat.	Hanya mengevaluasi satu model <i>YOLO</i> sehingga belum memberikan gambaran model terbaik untuk deteksi kualitas biji kopi.
9	Muchtar et al. [21]	2025	Sistem <i>cloud AI</i> memiliki latency tinggi dan bergantung pada internet.	<i>Edge AI</i> berbasis <i>deep learning</i> dan <i>Streamlit</i> .	Sistem mampu melakukan deteksi <i>defect</i> secara <i>real-time</i> dengan antarmuka yang ringan dan interaktif.	Fokus pada implementasi sistem, bukan pada komparasi performa model objek deteksi.
10	Febriana et al. [22]	2022	<i>Dataset</i> multi-class biji kopi untuk <i>deep learning</i> masih terbatas.	Pengembangan USK-COFFEE <i>Dataset</i> .	<i>Dataset</i> berhasil menyediakan citra multi-kelas biji kopi Arabika untuk penelitian <i>deep learning</i> dan objek deteksi.	Penelitian hanya menyediakan <i>dataset</i> dan belum melakukan evaluasi komprehensif terhadap berbagai model objek deteksi.

Berdasarkan Tabel 2.1, dapat diketahui bahwa penelitian mengenai deteksi kualitas dan cacat biji kopi telah banyak dilakukan menggunakan berbagai pendekatan *deep learning*, seperti *CNN*, *Mask R-CNN*, dan *YOLO*. Penelitian

berbasis *CNN* [13][16] menunjukkan kemampuan yang baik dalam mengenali pola cacat, namun masih terbatas pada tugas klasifikasi sehingga tidak dapat menentukan lokasi objek secara langsung. Sementara itu, penelitian berbasis *Mask R-CNN* [14][15][19] mampu melakukan deteksi dan segmentasi objek dengan akurasi yang tinggi, tetapi umumnya memiliki kompleksitas komputasi yang lebih besar dan belum dibandingkan secara langsung dengan model *one-stage* detector.

Pada sisi lain, penelitian berbasis *YOLO* [17][18][20] menunjukkan keunggulan dalam kecepatan inferensi dan kemampuan deteksi *real-time*. Namun, penelitian tersebut hanya berfokus pada pengembangan atau modifikasi satu keluarga model *YOLO* tanpa membandingkannya dengan arsitektur objek deteksi lainnya. Selain itu, sebagian besar penelitian sebelumnya hanya mengevaluasi aspek akurasi deteksi tanpa menganalisis keseimbangan antara akurasi dan efisiensi komputasi.

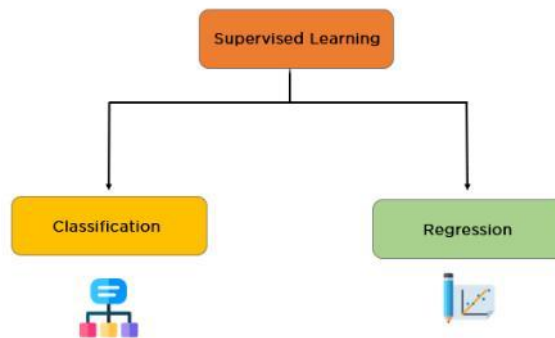
Berdasarkan hasil analisis tersebut, terdapat tiga research gap utama. Pertama, masih terbatas penelitian yang membandingkan model objek deteksi dari arsitektur yang berbeda pada kasus kualitas biji kopi. Kedua, belum ditemukan penelitian yang secara komprehensif membandingkan *YOLOv8*, *Faster R-CNN-ResNet50 FPN*, dan *SSD300 VGG16* menggunakan *dataset* yang sama. Ketiga, penelitian sebelumnya belum banyak mengevaluasi *trade-off* antara akurasi deteksi dan efisiensi komputasi sebagai dasar pemilihan model terbaik untuk sistem *grading* kualitas biji kopi otomatis. Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut melalui komparasi performa ketiga model tersebut menggunakan *dataset USK-COFFEE*.

2.2 Teori yang berkaitan

Teori yang berkaitan merupakan landasan konseptual yang digunakan untuk mendukung pelaksanaan penelitian. Teori-teori dibawah ini digunakan sebagai dasar dalam memahami proses deteksi dan klasifikasi kualitas biji kopi berbasis citra digital serta sebagai acuan dalam pengembangan dan evaluasi model yang diusulkan;

2.2.1 Machine Learning

Machine Learning merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan sistem komputer untuk mempelajari pola dari data dan membuat keputusan atau prediksi tanpa harus diprogram secara eksplisit. Dalam konteks pengolahan citra (*image processing*), *machine learning* berperan penting dalam mengenali pola visual seperti warna, bentuk, dan tekstur yang digunakan untuk mengklasifikasikan suatu objek.



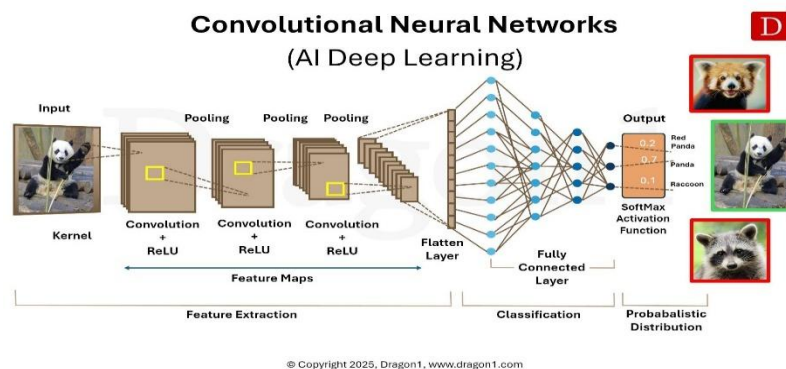
Gambar 2.1 Supervised Learning

Secara umum, algoritma *machine learning* dibagi menjadi tiga kategori utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Pada penelitian yang berfokus pada klasifikasi pada bagian gambar 2.1 dan deteksi kualitas biji kopi, pendekatan yang paling umum digunakan adalah *supervised learning*, di mana model dilatih menggunakan *dataset* yang telah memiliki *label* kelas tertentu, dalam pendekatan *supervised learning*, model mempelajari hubungan antara data input dan *label* output sehingga mampu melakukan prediksi terhadap data baru yang belum pernah dilihat sebelumnya, metode ini digunakan dalam berbagai aplikasi seperti klasifikasi citra, pengenalan wajah, deteksi objek, serta analisis kualitas produk pertanian.

Penelitian oleh Adiwijaya et al. [14] menunjukkan bahwa metode *machine learning* seperti *K-Nearest Neighbor (KNN)* dapat digunakan untuk mengklasifikasikan kualitas biji kopi berdasarkan fitur warna (*RGB*) yang diekstraksi dari citra digital. Hasil penelitian tersebut menunjukkan bahwa pendekatan berbasis pembelajaran mesin mampu meningkatkan efisiensi proses sortasi biji kopi yang sebelumnya dilakukan secara manual, dengan tingkat akurasi mencapai 83%, seiring perkembangan teknologi, pendekatan *machine learning* berbasis *deep learning*, khususnya *Convolutional Neural Network (CNN)*, semakin banyak digunakan dalam bidang *computer vision*. Salah satu implementasi penting adalah penggunaan algoritma *You Only Look Once (YOLO)* dalam tugas deteksi objek. Penelitian oleh Pawłowski et al. [15] menunjukkan bahwa model *YOLO* mampu mendeteksi dan mengukur objek biji secara otomatis dalam citra dengan akurasi yang tinggi, serta dapat mengidentifikasi posisi, ukuran, dan jenis objek secara simultan.

2.2.2 Deep Learning

Deep learning merupakan bagian dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) untuk mempelajari representasi data secara hierarkis, pendekatan ini sangat efektif dalam menangani data kompleks seperti citra, suara, dan teks karena mampu menangkap hubungan non-linear dan pola yang kompleks dalam data.



Gambar 2.2 Contoh Arsitektur Deep Learning

Sumber : [19]

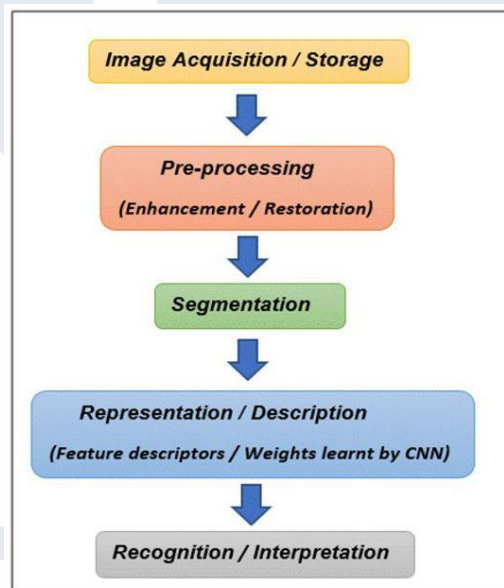
Keunggulan utama *deep learning* pada contoh arsitektur gambar 2.2 terletak pada kemampuannya dalam melakukan ekstraksi fitur secara otomatis (*automatic feature extraction*) tanpa memerlukan proses ekstraksi fitur manual. Dalam analisis citra, *deep learning* khususnya *Convolutional Neural Network (CNN)* [16] mampu mengenali berbagai pola visual seperti warna, tekstur, bentuk, serta struktur objek secara bertingkat, mulai dari fitur sederhana hingga fitur yang lebih kompleks.

Menurut Wang et al. [16], penerapan *deep learning* berbasis *CNN* pada sistem inspeksi biji kopi mampu meningkatkan akurasi deteksi cacat seperti jamur, kerusakan fisik, dan kontaminasi lainnya dibandingkan metode manual. Selain itu, pendekatan ini juga dapat mengurangi ketergantungan terhadap tenaga manusia serta meningkatkan efisiensi dan konsistensi dalam proses penilaian kualitas biji kopi, dalam implementasinya, *CNN* terdiri dari beberapa lapisan utama, yaitu *convolutional layer* untuk mengekstraksi fitur, *pooling layer* untuk mereduksi dimensi data, serta *fully connected layer* untuk proses klasifikasi. Kombinasi dari lapisan-lapisan ini memungkinkan model untuk memahami representasi visual secara mendalam dan menghasilkan prediksi yang akurat.

Penelitian lain oleh Korkmaz et al. [17] menunjukkan bahwa berbagai arsitektur *deep learning* seperti *InceptionV3*, *DenseNet*, dan *Xception* mampu menghasilkan performa klasifikasi biji kopi dengan tingkat akurasi yang tinggi, bahkan mencapai lebih dari 90% pada beberapa model, hal ini menunjukkan bahwa pemilihan arsitektur *deep learning* yang tepat sangat berpengaruh terhadap hasil klasifikasi, selain itu, penelitian oleh Jayakumari et al. [18] mengungkapkan bahwa model *deep learning* seperti *EfficientNet* dan *ResNet* dapat digunakan untuk proses *grading* biji kopi secara otomatis dengan akurasi yang sangat tinggi, bahkan mencapai hingga 100% dalam kondisi tertentu. Pendekatan ini sangat relevan dalam industri karena mampu bekerja secara cepat, konsisten, dan tidak merusak objek (*non-destructive*).

2.2.3 Computer Vision

Computer Vision merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan sistem komputer untuk memahami dan mengekstraksi informasi dari data visual seperti citra atau video. Dalam konteks penelitian ini, *computer vision* digunakan untuk menganalisis citra biji kopi guna mengidentifikasi kualitas, cacat, serta karakteristik visual lainnya secara otomatis, menurut Jiao et al. [20], teknologi machine vision memungkinkan ekstraksi berbagai fitur visual dari biji kopi, seperti bentuk, ukuran, warna, dan tekstur, yang kemudian digunakan dalam proses klasifikasi dan *grading* kualitas. Pendekatan ini menjadi solusi yang lebih efisien dibandingkan metode manual, karena mampu bekerja secara cepat, konsisten, dan tidak bergantung pada subjektivitas manusia



Gambar 2.3 Bagian Computer Vision

Sumber : [23]

Dalam implementasinya, sistem *computer vision* umumnya terdiri dari beberapa tahapan utama, yaitu image acquisition, *preprocessing*, feature extraction, dan *classification*. Pada tahap *preprocessing*, citra mengalami berbagai proses seperti konversi ke *grayscale*, *filtering* untuk mengurangi noise, peningkatan kontras, serta segmentasi objek dari latar belakang. Sebagai contoh, pada penelitian

Jiao et al. [20], citra biji kopi diproses menggunakan teknik seperti *Gaussian filtering*, transformasi *gamma*, deteksi tepi, dan *thresholding* untuk meningkatkan kualitas citra sebelum dilakukan analisis lebih lanjut.

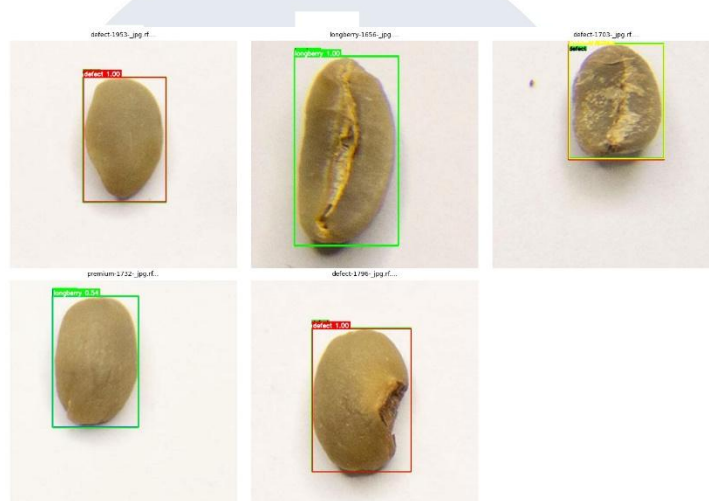
Selanjutnya, pada tahap ekstraksi fitur, model akan mengambil informasi penting dari citra yang merepresentasikan karakteristik objek. Dalam pendekatan modern, proses ini dilakukan secara otomatis menggunakan model *deep learning* seperti *CNN* atau *Transformer*. Penelitian oleh Izza dan Kusuma [21] menjelaskan bahwa dalam *computer vision* modern, citra dapat diubah menjadi representasi berbentuk *patch* yang kemudian dianalisis menggunakan mekanisme *self-attention* untuk memahami hubungan antar bagian citra secara lebih menyeluruh.

Selain itu, perkembangan terbaru dalam *computer vision* juga melibatkan integrasi dengan model *vision-language* untuk tugas yang lebih kompleks seperti pengenalan teks pada kemasan produk. Le et al. [22] menyatakan bahwa proses pengolahan citra pada kondisi dunia nyata sering menghadapi berbagai tantangan seperti variasi pencahayaan, blur, distorsi geometris, dan latar belakang yang kompleks, sehingga diperlukan tahap *preprocessing* yang kuat untuk meningkatkan kualitas citra sebelum dilakukan proses pengenalan.

2.2.4 Objek Deteksi

Objek Deteksi merupakan salah satu cabang penting dalam *computer vision* yang bertujuan untuk mengidentifikasi serta menentukan lokasi objek dalam suatu citra. Berbeda dengan klasifikasi citra yang hanya memberikan satu *label* pada keseluruhan gambar, *objek deteksi* mampu mendeteksi banyak objek sekaligus beserta posisi spasialnya dalam bentuk *bounding box*. Dalam konteks penelitian ini, *objek deteksi* digunakan untuk mendeteksi biji kopi secara lebih detail, termasuk jumlah, lokasi, serta kondisi objek dalam satu citra. Hal ini menjadi penting karena dalam satu gambar dapat terdapat banyak biji kopi dengan kualitas yang berbeda-beda, sehingga pendekatan klasifikasi saja tidak cukup.

Menurut Liu et al. [13], teknologi *objek deteksi* berbasis *deep learning* memberikan solusi otomatis dalam analisis visual, terutama pada sektor pertanian. Namun, terdapat berbagai tantangan dalam penerapannya, seperti ukuran objek yang kecil, *occlusion* (objek saling menutupi), serta variasi pencahayaan yang dapat mempengaruhi akurasi deteksi, permasalahan ini juga umum terjadi pada objek seperti buah kopi yang sering berkelompok dan memiliki kemiripan warna antar kelas kematangan.



Gambar 2.4 Contoh Penggunaan Objek Deteksi

Seperti Gambar 2.4 Model Objek Deteksi modern umumnya berbasis *deep learning*, khususnya *Convolutional Neural Network (CNN)* dan kombinasi dengan *Transformer*. Salah satu algoritma yang paling populer adalah *YOLO (You Only Look Once)*, yang dikenal karena kemampuannya dalam melakukan deteksi objek secara *real-time* dengan akurasi yang baik. *YOLO* bekerja dengan membagi citra menjadi grid dan secara langsung memprediksi *bounding box* serta kelas objek dalam satu tahap proses.

Penelitian Liu et al. [13] menunjukkan bahwa pengembangan model berbasis *YOLO* yang dikombinasikan dengan *Vision Transformer* mampu meningkatkan kemampuan deteksi, terutama dalam memahami konteks global dan menangani objek kecil, model yang diusulkan, yaitu *SSViT-YOLOv11*, berhasil mencapai nilai *mean Average Precision (mAP@50)* sebesar 84.54% dengan performa yang

seimbang antara akurasi dan kecepatan, selain *YOLO*, pendekatan objek deteksi lainnya juga berkembang, seperti *Faster R-CNN* dan *SSD*. Model *Faster R-CNN* menggunakan pendekatan dua tahap (*two-stage detector*), yaitu menghasilkan region proposal terlebih dahulu sebelum melakukan klasifikasi, [23] sehingga memiliki akurasi tinggi namun kecepatan lebih rendah. Sementara itu, *SSD (Single Shot Detector)* merupakan model satu tahap (*one-stage detector*) yang lebih cepat dibandingkan *Faster R-CNN*, namun biasanya memiliki *trade-off* pada akurasi, lebih lanjut, Hassan[24] menyebutkan bahwa integrasi *deep learning* dalam *computer vision*, termasuk pada *objek deteksi*, mampu meningkatkan performa sistem secara signifikan dibandingkan metode tradisional, terutama dalam menangani kompleksitas data visual seperti variasi bentuk, warna, dan tekstur pada objek.

2.3 Framework/Algoritma yang digunakan

Dalam penelitian ini digunakan beberapa algoritma *deep learning* berbasis objek deteksi, yaitu *YOLO (You Only Look Once)*, *Faster R-CNN*, dan *SSD (Single Shot Detector)*. Ketiga algoritma ini dipilih karena memiliki karakteristik yang berbeda dalam hal kecepatan dan akurasi deteksi objek.

2.3.1 YOLO (You Only Look Once)

YOLO (You Only Look Once) merupakan salah satu algoritma objek deteksi berbasis *deep learning* yang menggunakan pendekatan *single-stage detector*, di mana proses deteksi objek dilakukan hanya dalam satu kali *forward pass* pada jaringan, pendekatan ini memungkinkan model untuk secara langsung memprediksi lokasi objek dalam bentuk *bounding box* sekaligus menentukan kelas objek dalam satu proses yang terintegrasi. *YOLO* bekerja dengan membagi citra input menjadi beberapa grid, di mana setiap *grid* bertanggung jawab untuk mendeteksi objek yang berada di dalam wilayahnya. Untuk setiap *grid* tersebut, model akan menghasilkan prediksi berupa koordinat *bounding box*, nilai kepercayaan (*confidence score*), serta probabilitas kelas objek, sehingga memungkinkan deteksi multi-objek secara efisien dalam satu citra.

Dalam proses evaluasi deteksi objek, salah satu metrik yang digunakan adalah *Intersection over Union (IoU)*, yang berfungsi untuk mengukur tingkat kesesuaian antara *bounding box* hasil prediksi dengan *ground truth*. Nilai *IoU* dihitung sebagai perbandingan antara luas area irisan (*overlap*) dengan luas area gabungan (*union*), yang secara matematis dirumuskan sebagai berikut:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of } U}$$

Equation 2. 1 *IoU Yolo*

Selain itu, *YOLO* juga menggunakan *confidence score* untuk menunjukkan tingkat keyakinan model terhadap keberadaan objek dalam suatu *bounding box*, yang diperoleh dari hasil perkalian antara probabilitas keberadaan objek dengan nilai *IoU*, yang dapat dirumuskan sebagai berikut:

$$\text{Confidence} = P(\text{Object}) \times IoU$$

Equation 2. 2 Rumus *Confidence Yolo*

Untuk mengoptimalkan proses pelatihan, *YOLO* menggunakan fungsi *loss* gabungan yang mencakup kesalahan prediksi lokasi, ukuran *i*, probabilitas objek, serta klasifikasi kelas. Fungsi *loss* ini dirancang untuk meminimalkan perbedaan antara prediksi model dan nilai sebenarnya, yang secara umum dapat dinyatakan sebagai berikut:

$$\text{Loss} = \lambda_{\text{coord}} [(x - x')^2 + (y - y')^2] + \lambda_{\text{coord}} [(w - w')^2 + (h - h')^2] + (p - p')^2 + (c - c')^2$$

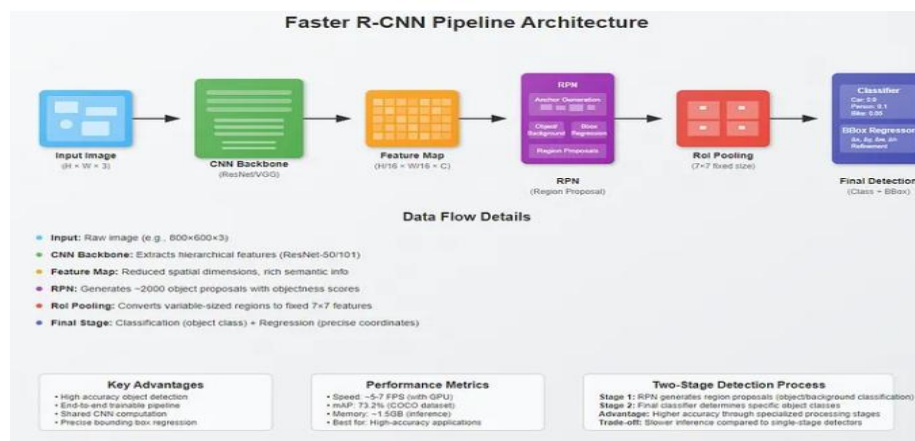
Equation 2. 3 Rumus *Loss Yolo*

Perkembangan terbaru dari *YOLO*, khususnya *YOLOv8*, menunjukkan peningkatan performa yang signifikan baik dari segi akurasi maupun efisiensi komputasi. Penelitian yang dilakukan oleh Zhang et al. menunjukkan bahwa pengembangan model berbasis *YOLOv8* mampu meningkatkan kemampuan deteksi

objek dalam kondisi kompleks dengan tetap mempertahankan kecepatan inferensi yang tinggi [25]. Selain itu, penelitian oleh You dan Kong juga menunjukkan bahwa optimasi arsitektur *YOLOv8* dapat meningkatkan performa deteksi pada objek dengan karakteristik yang sulit, seperti variasi tekstur dan bentuk yang kompleks [26]. Hal ini menunjukkan bahwa *YOLO* terus mengalami perkembangan dan tetap relevan untuk berbagai aplikasi objek deteksi di dunia nyata.

2.3.2 *Faster R-CNN*

Faster R-CNN merupakan salah satu algoritma objek deteksi berbasis *deep learning* yang termasuk dalam kategori *two-stage* detector. Pada pendekatan ini, proses deteksi objek dilakukan melalui dua tahapan utama, yaitu tahap pertama untuk menghasilkan kandidat area objek (*region proposal*) dan tahap kedua untuk melakukan klasifikasi serta penyempurnaan lokasi objek tersebut. Tahap awal dalam *Faster R-CNN* menggunakan *Region Proposal Network* (RPN) yang berfungsi untuk menghasilkan sejumlah kandidat lokasi objek yang berpotensi mengandung objek pada citra.



Gambar 2.5 Arsitektur *Faster R-CNN*

Sumber : [29]

Selanjutnya, pada tahap kedua seperti Gambar 2.5, setiap kandidat tersebut diproses lebih lanjut untuk menentukan kelas objek serta memperbaiki koordinat bounding box agar lebih akurat. Pendekatan dua tahap ini memungkinkan model

untuk lebih fokus dalam mendeteksi objek secara detail dibandingkan metode satu tahap, dibandingkan dengan metode objek deteksi lainnya, Faster R-CNN dikenal memiliki tingkat akurasi yang tinggi dalam mendeteksi objek, terutama pada objek berukuran kecil atau memiliki karakteristik visual yang kompleks. Hal ini disebabkan karena proses region proposal yang dilakukan secara eksplisit memungkinkan model untuk menyeleksi area penting dalam citra sebelum melakukan klasifikasi. Penelitian yang dilakukan oleh Zhang et al. menunjukkan bahwa pendekatan Faster R-CNN mampu memberikan hasil deteksi yang akurat dalam berbagai kasus pengenalan objek berbasis citra, termasuk pada objek dengan variasi bentuk dan tekstur yang beragam. Selain itu, penelitian oleh Zangana et al. juga menunjukkan bahwa integrasi Faster R-CNN dengan metode lain dapat meningkatkan robustitas sistem deteksi objek, terutama dalam kondisi citra yang kompleks atau memiliki gangguan visual. Hal ini menegaskan bahwa Faster R-CNN merupakan model yang kuat dalam menangani permasalahan objek deteksi yang membutuhkan presisi tinggi.

2.3.3 SSD (Single Shot Detector)

Single Shot Detector (SSD) merupakan salah satu algoritma *objek deteksi* berbasis *deep learning* yang termasuk dalam kategori *one-stage* detector. Berbeda dengan metode *two-stage* seperti *Faster R-CNN*, *SSD* melakukan proses deteksi objek secara langsung dalam satu tahap tanpa memerlukan proses region proposal terpisah. Model ini bekerja dengan cara memanfaatkan *multi-scale* feature maps dari beberapa layer konvolusi untuk mendeteksi objek dengan berbagai ukuran dalam satu kali proses inferensi. Pendekatan ini memungkinkan *SSD* untuk menghasilkan prediksi *bounding box* dan klasifikasi objek secara simultan dalam satu alur jaringan.

Dalam implementasinya, *SSD* menggunakan jaringan konvolusi sebagai *feature extractor* yang kemudian diikuti oleh beberapa layer tambahan untuk menghasilkan prediksi objek pada berbagai skala. Berdasarkan penjelasan pada diagram arsitektur, *SSD* memanfaatkan beberapa layer konvolusi dengan ukuran berbeda untuk mendeteksi objek kecil hingga besar secara efektif. Setiap layer

menghasilkan kandidat *bounding box* beserta probabilitas kelas objek, sehingga model mampu menangani variasi ukuran objek dalam citra secara lebih fleksibel, dari segi performa, *SSD* dikenal memiliki keseimbangan yang baik antara kecepatan dan akurasi. Penelitian oleh Wahab et al. [30] menunjukkan bahwa *SSD* mampu digunakan untuk sistem deteksi objek secara *real-time* dengan tingkat akurasi yang tinggi serta efisiensi yang baik pada perangkat dengan spesifikasi terbatas. Selain itu, *SSD* juga memiliki keunggulan dalam hal stabilitas performa dibandingkan beberapa metode *one-stage* detector lainnya.

Pemilihan model *YOLOv8*, *Faster R-CNN*, *ResNet50 FPN*, dan *SSD300 VGG16* didasarkan pada perbedaan karakteristik arsitektur yang mewakili pendekatan *objek deteksi* yang paling banyak digunakan dalam penelitian dan industri. *YOLOv8* dipilih karena merupakan salah satu model *one-stage* detector modern yang memiliki keunggulan pada kecepatan inferensi dan kemampuan deteksi *real-time*. *Faster R-CNN*, *ResNet50 FPN* dipilih karena merupakan representasi *two-stage* detector yang dikenal memiliki akurasi tinggi terutama untuk objek kecil dan kompleks. Sementara itu, *SSD300 VGG16* dipilih karena menawarkan keseimbangan antara kecepatan dan akurasi sehingga sering digunakan sebagai baseline dalam berbagai penelitian *objek deteksi*.

Meskipun telah tersedia model yang lebih baru seperti *YOLOv10*, *YOLOv11*, *EfficientDet*, *RT-DETR*, dan *DETR*, model-model tersebut belum banyak digunakan pada penelitian kualitas biji kopi sehingga hasil perbandingan yang diperoleh akan lebih sulit dibandingkan dengan penelitian terdahulu. Selain itu, *YOLOv8*, *Faster R-CNN*, dan *SSD* merupakan model yang telah matang (*mature architecture*), memiliki dokumentasi yang lengkap, implementasi yang stabil pada *framework PyTorch*, serta sering dijadikan *benchmark* dalam penelitian *objek deteksi*. Oleh karena itu, ketiga model tersebut dipilih untuk memberikan perbandingan yang representatif antara aspek akurasi dan efisiensi komputasi pada kasus deteksi kualitas biji kopi.

Pemilihan *SSD300-VGG16* sebagai salah satu model pembanding juga didukung oleh penelitian terdahulu yang menguji *SSD* secara langsung. Srivastava

dkk. [27] membandingkan SSD, Faster R-CNN, dan YOLO dalam satu lingkungan pengujian yang sama menggunakan *dataset* MS COCO dan menunjukkan bahwa SSD mampu memberikan keseimbangan yang baik antara akurasi dan kecepatan deteksi, sehingga layak dijadikan salah satu baseline dalam studi komparasi model objek deteksi. Hasil tersebut memperkuat alasan pemilihan SSD300-VGG16 sebagai representasi *one-stage* detector klasik (baseline) untuk dibandingkan secara langsung terhadap model *two-stage* (Faster R-CNN) maupun *one-stage* modern (YOLOv8) dalam satu kerangka pengujian yang sama. Temuan serupa juga dilaporkan oleh Wahab dkk. [28], yang menunjukkan bahwa SSD dapat diimplementasikan untuk sistem deteksi objek secara *real-time* dengan akurasi tinggi dan efisiensi yang baik bahkan pada perangkat dengan spesifikasi terbatas. Dengan demikian, pemilihan SSD300-VGG16 pada penelitian ini tidak hanya didasarkan pada keseimbangan kecepatan dan akurasinya, tetapi juga sebagai baseline standar yang umum digunakan dalam penelitian objek deteksi. Pemilihan ini sekaligus menjawab *research gap* yang telah diuraikan pada Subbab 2.1, yaitu belum adanya penelitian yang membandingkan YOLOv8, Faster R-CNN ResNet50-FPN, dan SSD300-VGG16 secara langsung menggunakan *dataset* biji kopi yang sama.

Tabel 2.2 Perbandingan Arsitektur Model Objek Deteksi

Aspek	YOLOv8	Faster R-CNN ResNet50-FPN	SSD300 VGG16
Tipe arsitektur	<i>One-stage detector</i>	<i>Two-stage detector</i>	<i>One-stage detector</i>
<i>Backbone</i>	CSPDarknet	ResNet50 + FPN	VGG16
Mekanisme deteksi	Prediksi langsung (anchor-free)	<i>Region Proposal Network</i> + klasifikasi	Multi-scale feature maps
Kelebihan	Cepat dan ringan untuk deteksi <i>real-time</i>	Akurasi tinggi, baik untuk objek kecil dan kompleks	Seimbang antara kecepatan dan akurasi
Kekurangan	Kurang presisi pada kelas berkemiripan visual	Komputasi berat dan inferensi lambat	Presisi di bawah <i>two-stage detector</i>
Ukuran input	640 x 640	Fleksibel	300 x 300

Tabel 2.2 merangkum perbandingan karakteristik ketiga arsitektur model objek deteksi yang dikaji pada penelitian ini. Ketiganya mewakili pendekatan yang berbeda, yaitu YOLOv8 dan SSD300 sebagai *one-stage detector* yang mengutamakan kecepatan, serta Faster R-CNN sebagai *two-stage detector* yang

mengutamakan akurasi. Perbedaan karakteristik tersebut menjadi dasar pemilihan ketiga model untuk dibandingkan secara komprehensif pada penelitian ini.

2.3.4 Metrik Evaluasi Objek Deteksi

Dalam penelitian ini, performa model *objek deteksi* dievaluasi menggunakan beberapa metrik yang umum digunakan dalam bidang *computer vision*, yaitu *Intersection over Union (IoU)*, *Precision*, *Recall*, dan *mean Average Precision (mAP)*. Metrik-metrik ini digunakan untuk mengukur seberapa baik model dalam mendeteksi dan mengklasifikasikan objek pada citra secara akurat.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of U}}$$

Equation 2. 4 Rumus *Intersection over Union* metrik evaluasi

Intersection over Union (IoU) merupakan metrik yang digunakan untuk mengukur tingkat kesesuaian antara bounding box hasil prediksi model dengan ground truth. Nilai IoU dihitung berdasarkan rasio antara area irisan (intersection) dan area gabungan (union) dari kedua bounding box tersebut. Semakin tinggi nilai IoU, maka semakin baik prediksi lokasi objek yang dihasilkan oleh model.

Selanjutnya, *Precision* digunakan untuk mengukur tingkat ketepatan model dalam mendeteksi objek, yaitu seberapa banyak prediksi yang benar dibandingkan dengan seluruh prediksi yang dilakukan. Nilai *precision* yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan deteksi (*false positive*) yang rendah.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Equation 2.5 Rumus *Precision* Objek Deteksi

Sementara itu, *Recall* digunakan untuk mengukur kemampuan model dalam menemukan seluruh objek yang seharusnya terdeteksi dalam suatu citra. Nilai *recall* yang tinggi menunjukkan bahwa model mampu meminimalkan jumlah objek yang terlewat (*false negative*).

$$\text{Recall} = \frac{TP}{TP+FN}$$

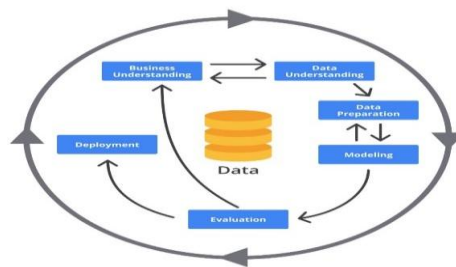
Equation 2.6 Rumus *Recall* Objek Deteksi

Sebagai metrik utama dalam evaluasi objek deteksi, *mean Average Precision (mAP)* digunakan untuk menggabungkan nilai *precision* dan *recall* dalam satu ukuran kinerja yang komprehensif. *mAP* dihitung dengan mengambil rata-rata nilai *Average Precision (AP)* pada setiap kelas objek, di mana *AP* diperoleh dari kurva *precision -recall*. Metrik ini banyak digunakan karena mampu memberikan gambaran menyeluruh terhadap performa model dalam berbagai kondisi deteksi objek [29][28]

2.3.5 *Framework* dan Metodologi dalam Pengembangan Objek Deteksi

2.3.5.1 *CRISP-DM (Cross-Industry Standard Process for Data Mining)*

Adalah metode yang dikembangkan untuk memberikan panduan langkah demi langkah yang terstruktur dalam proses pengumpulan data. Metode ini terdiri dari enam tahapan utama, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment* [30].



Gambar 2.6 Metode *CRISP-DM*

Sumber : [31]

CRISP-DM dirancang untuk membantu para profesional memahami konteks bisnis, menganalisis data, mempersiapkan data untuk analisis, membangun model, mengevaluasi hasil, dan menerapkan solusi yang dikembangkan. Pendekatan ini menyediakan kerangka kerja yang jelas, memastikan bahwa proses

pengumpulan data dilakukan secara sistematis dan efektif sesuai pada gambar 2.6 berikut :

a. Business Understanding

Pada tahap ini, dijelaskan secara rinci tujuan dan persyaratan yang perlu dicapai dalam ruang lingkup penelitian. Proses ini melibatkan analisis mendalam untuk mengidentifikasi tujuan akhir penelitian dan kebutuhan yang harus dipenuhi. Selain itu, tahap ini juga mencakup pemahaman tujuan penelitian dan batasan-batasan yang ada, yang kemudian dijabarkan dalam bentuk rumusan masalah yang terukur.

b. Data Understanding

Tahap ini dimulai dengan pengumpulan data yang relevan dengan tujuan penelitian. Proses pengumpulan mencakup serangkaian tindakan seperti pencarian, pengambilan, dan dokumentasi data yang dibutuhkan. Setelah data terkumpul, langkah berikutnya adalah melakukan analisis mendalam untuk mengidentifikasi variabel-variabel penting yang berpengaruh terhadap tujuan penelitian. Analisis ini sangat penting untuk memastikan bahwa data yang dikumpulkan dapat memberikan kontribusi yang signifikan pada tahap selanjutnya.

c. Data Preparation

Pada tahap ini, dilakukan persiapan awal data yang akan digunakan dalam penelitian. Proses ini mencakup pemilihan variabel yang relevan untuk analisis dan penyesuaian variabel tersebut jika diperlukan, sehingga data siap digunakan dalam tahap *modeling*.

d. Modeling

Pada tahap *modeling*, teknik pemodelan yang sesuai dipilih berdasarkan tujuan penelitian. Setelah pemilihan teknik, prosedur pemodelan diterapkan untuk mencapai hasil yang optimal. Jika diperlukan, proses ini bisa kembali ke tahap persiapan data untuk menyempurnakan data agar sesuai dengan kebutuhan pemodelan.

e. *Evaluation*

Selama tahap evaluasi, model-model yang telah dibangun dievaluasi untuk memastikan efektivitasnya sebelum diterapkan secara praktis. Hal ini mencakup analisis mendalam dan pengujian kinerja model untuk menentukan apakah model tersebut memenuhi kebutuhan spesifik dan karakteristik kasus yang sedang diteliti. Evaluasi memainkan peran penting dalam memastikan bahwa model yang diterapkan di dunia nyata memiliki kualitas yang tinggi dan dapat diandalkan.

f. *Deployment*

Pada tahap *deployment*, hasil yang dihasilkan oleh model dipresentasikan, presentasi ini mencakup berbagai hasil yang dicapai selama proses, serta wawasan yang diperoleh dari model. Tujuan dari tahap ini adalah untuk memastikan bahwa hasil penelitian mudah dipahami oleh pengguna dan memberikan manfaat praktis yang dapat diterapkan dalam konteks dunia nyata.

2.4 Tools atau *software* yang digunakan

Penelitian ini memanfaatkan berbagai tools dan perangkat lunak untuk mendukung proses pengolahan data, pelatihan model, serta evaluasi performa sistem *objek deteksi*. Pemilihan tools didasarkan pada kemampuannya dalam mendukung pengembangan model berbasis *deep learning* secara efisien dan terintegrasi.

2.4.1 *Python*

Python merupakan bahasa pemrograman tingkat tinggi yang banyak dimanfaatkan dalam pengembangan aplikasi data science, *machine learning*, dan *computer vision*. Bahasa pemrograman ini populer karena memiliki sintaks yang sederhana serta mudah dipahami, sehingga mendukung proses pengembangan sistem secara lebih efisien. Selain itu, *Python* didukung oleh berbagai *library* seperti *NumPy*, *Pandas*, *Matplotlib*, dan *PyTorch* yang mempermudah proses pengolahan data, visualisasi, hingga pengembangan model *deep learning* [31].

Dalam bidang objek deteksi, *Python* sering digunakan sebagai bahasa utama karena mampu mengintegrasikan berbagai *framework* dan *library machine learning* secara fleksibel. Penggunaan *Python* memungkinkan proses pembangunan sistem dilakukan secara terstruktur mulai dari *preprocessing* data, pelatihan model, evaluasi performa, hingga implementasi sistem secara efisien [32].

2.4.2 OpenCV

OpenCV (*Open Source Computer Vision Library*) merupakan *library* yang digunakan untuk pengolahan citra dan video secara *real-time*. *Library* ini menyediakan berbagai fungsi seperti deteksi objek, segmentasi citra, ekstraksi fitur, dan transformasi gambar, menurut penelitian terkini, *OpenCV* sering digunakan sebagai komponen pendukung dalam sistem objek deteksi untuk tahap *preprocessing*, seperti *resizing*, normalisasi citra, serta augmentasi data. Selain itu, *OpenCV* juga digunakan untuk menampilkan hasil deteksi dalam bentuk *bounding box* pada citra atau video secara langsung [28].

2.4.3 Roboflow

Roboflow merupakan platform berbasis *web* yang digunakan untuk mendukung proses pengelolaan *dataset* pada bidang *computer vision*. Platform ini menyediakan berbagai fitur seperti anotasi objek, *preprocessing* data, augmentasi citra, serta konversi format *dataset* yang mempermudah proses pengembangan model *deep learning*. Dengan adanya fitur tersebut, proses persiapan *dataset* dapat dilakukan secara lebih terstruktur dan efisien [33], dalam penelitian berbasis objek deteksi, *Roboflow* banyak dimanfaatkan untuk meningkatkan kualitas *dataset* melalui teknik augmentasi data seperti rotasi, *flipping*, *scaling*, dan penyesuaian pencahayaan. Penggunaan augmentasi tersebut membantu menghasilkan variasi data yang lebih beragam sehingga model *deep learning* mampu melakukan generalisasi dengan lebih baik terhadap kondisi citra yang berbeda-beda. Selain itu, *Roboflow* juga mendukung proses ekspor *dataset* ke berbagai format seperti *YOLO* dan *COCO* yang kompatibel dengan *framework deep learning* modern [35].

2.4.4 *Ultralytics YOLO*

Ultralytics YOLO merupakan implementasi modern dari algoritma *YOLO* (*You Only Look Once*) yang digunakan untuk objek deteksi secara *real-time*. Model ini termasuk dalam kategori *one-stage* detector yang mampu melakukan deteksi objek dalam satu kali proses tanpa tahap *region proposal*, penelitian terbaru menunjukkan bahwa *YOLOv8* yang dikembangkan oleh *Ultralytics* memiliki peningkatan performa baik dari segi akurasi maupun kecepatan dibandingkan versi sebelumnya. Model ini mampu mendeteksi objek dengan presisi tinggi serta waktu inferensi yang cepat, sehingga cocok digunakan dalam aplikasi *real-time* seperti pengawasan, industri, dan pertanian [25], [26].

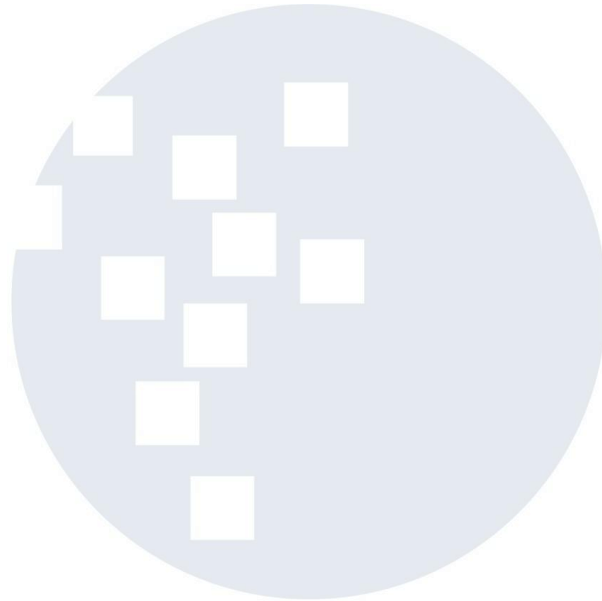
2.4.5 *PyTorch*

PyTorch merupakan *framework deep learning* yang banyak digunakan untuk membangun dan melatih model jaringan saraf tiruan. *Framework* ini menyediakan fleksibilitas tinggi melalui konsep *dynamic computation graph*, sehingga memudahkan proses eksperimen dan pengembangan model, dalam penelitian *computer vision*, *PyTorch* sering digunakan untuk implementasi model berbasis *Convolutional Neural Network (CNN)*, termasuk untuk tugas objek deteksi. Selain itu, *PyTorch* juga mendukung akselerasi *GPU* yang memungkinkan proses pelatihan model menjadi lebih cepat dan efisien. Penelitian menunjukkan bahwa *PyTorch* menjadi salah satu *framework* yang paling banyak digunakan dalam pengembangan model berbasis citra karena kemudahan penggunaannya dan performa yang tinggi [30].

2.4.6 *Google Colaboratory*

Google Colaboratory merupakan platform berbasis *cloud* yang memungkinkan pengguna menjalankan kode *Python* secara *online* dengan dukungan *GPU* dan *TPU*. Platform ini banyak digunakan dalam penelitian *machine learning* karena menyediakan sumber daya komputasi yang cukup tanpa memerlukan perangkat keras lokal yang tinggi.

Dalam studi terbaru, *Google Colab* digunakan sebagai lingkungan pengembangan untuk melatih model *deep learning* karena kemudahan akses, integrasi dengan *Google Drive*, serta dukungan *library machine learning* yang lengkap. Hal ini memungkinkan proses eksperimen dan pelatihan model dilakukan secara lebih fleksibel dan efisien [34].



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA