

BAB 2

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Deteksi Tanaman Melon

Berdasarkan penelitian terdahulu, telah diimplementasikan identifikasi penyakit *Powdery Mildew* pada buah melon menggunakan metode YOLO yang dibantu oleh *Deep Learning, Single Shot Multibox Detection*. Penelitian ini hanya mendeteksi abnormalitas daun melon; identifikasi penyakit daun melon pada *green house* menggunakan model *Pruned-YOLO v5s+Shuffle* (PYSS) yang merupakan gabungan dari YOLOv5s dan *ShuffleNet V2* dan dilakukan secara *real-time*, sehingga memiliki akurasi dan kecepatan lebih tinggi dibandingkan model YOLOv3, *Faster R-CNN*, dan YOLOv5s. Namun, belum ada sistem yang mengimplementasikan identifikasi untuk penyakit *Powdery Mildew, Cucumber Mosaic Virus*, dan *Alternaria* pada tanaman buah melon berbasis YOLOv8.

2.1.2 *Alternaria*

Alternaria adalah salah satu genus jamur patogen yang sering menyerang tanaman, termasuk tanaman melon (*Cucumis melo*), yang menyebabkan penyakit yang dikenal dengan nama *Alternaria leaf spot* atau *Alternaria blight*. Penyakit ini dapat menurunkan kualitas dan hasil produksi tanaman melon, mengakibatkan kerugian ekonomi yang signifikan pada sektor pertanian.

A Penyebab *Alternaria* pada Tanaman Melon

Penyakit *Alternaria* pada tanaman melon disebabkan oleh jamur *Alternaria alternata*, yang merupakan spesies patogen utama dalam genus *Alternaria*. Jamur ini menghasilkan konidia (spora aseksual) yang dapat tersebar melalui angin, air hujan, atau kontak langsung antar tanaman [3]. Spora yang jatuh ke permukaan daun melon akan berkecambah dan menembus jaringan tanaman, menyebabkan infeksi.

B Gejala *Alternaria* pada Tanaman Melon

Gejala serangan *Alternaria* pada tanaman melon terutama terlihat pada daun. Pada tahap awal infeksi, muncul bercak kecil berwarna coklat atau kehitaman dengan batas yang kabur. Seiring berjalannya waktu, bercak ini akan membesar dan membentuk cincin konsentris yang menyerupai pola target. Jika infeksi berlangsung lama, daun yang terinfeksi akan menguning, mati, dan gugur. Pada tahap yang lebih parah, infeksi dapat menyebabkan kematian tanaman [4].

2.1.3 *Cucumber Mosaic Virus*

Cucumber Mosaic Virus (CMV) adalah salah satu virus yang sering menyerang tanaman melon (*Cucumis melo*) dan berbagai tanaman lainnya. CMV termasuk dalam kelompok virus RNA dari keluarga *Cucumoviridae* dan dapat menyebabkan kerusakan yang signifikan pada pertumbuhan tanaman, serta mengurangi hasil produksi melon. Infeksi CMV pada melon umumnya mempengaruhi kualitas buah dan daun, serta menyebabkan gejala yang beragam tergantung pada tingkat keparahan infeksi dan kondisi lingkungan.

A Penyebab *Cucumber Mosaic Virus* (CMV) pada Tanaman Melon

Cucumber Mosaic Virus disebabkan oleh virus RNA yang termasuk dalam *Cucumovirus*, dengan *Cucumber mosaic virus* sebagai spesies utama. CMV menyebar melalui vektor serangga, terutama kutu daun dari keluarga *Aphididae* [5]. Virus ini dapat dengan mudah ditularkan dari tanaman yang terinfeksi ke tanaman sehat melalui gigitan kutu daun yang membawa partikel virus. Selain itu, CMV juga dapat ditularkan melalui kontak langsung dengan tanaman yang terinfeksi atau peralatan pertanian yang terkontaminasi.

B Gejala *Cucumber Mosaic Virus* pada Tanaman Melon

Gejala CMV pada tanaman melon bervariasi tergantung pada tingkat infeksi dan ketahanan tanaman terhadap virus. Gejala yang paling umum terlihat adalah perubahan warna daun yang tidak merata, mulai dari bercak kuning atau hijau pucat, hingga pola mosaik hijau terang dan gelap [6]. Selain itu, daun juga dapat mengalami penggumpalan atau deformasi, dan pada tahap yang lebih parah, pertumbuhan tanaman akan terhambat, buah yang dihasilkan menjadi kecil,

terdistorsi, atau bahkan tidak berkembang sama sekali. Dalam beberapa kasus, tanaman yang terinfeksi *CMV* dapat mati lebih cepat, terutama jika infeksi terjadi pada tahap awal pertumbuhan tanaman.

2.1.4 Downy Mildew

Downy mildew adalah penyakit tanaman yang disebabkan oleh jamur yang bernama *Pseudoperonospora cubensis*. Penyakit ini terutama menyerang tanaman dari famili *Cucurbitaceae*, seperti melon, mentimun, dan semangka. Infeksi *downy mildew* terjadi pada bagian daun, dengan gejala khas berupa bercak kuning (klorosis) yang berkembang menjadi nekrosis. Penyakit ini menyebar dengan cepat melalui spora yang terbawa angin, terutama pada kondisi lingkungan yang lembap. *Downy mildew* dapat menurunkan kapasitas fotosintesis tanaman secara signifikan dan menyebabkan penurunan hasil panen, bahkan kegagalan produksi apabila tidak dikendalikan dengan baik.

A Penyebab Downy Mildew pada Tanaman Melon

Penyakit ini disebabkan jamur *Pseudoperonospora cubensis*, yang berkembang biak melalui spora yang tersebar lewat udara. Spora ini menyerang bagian daun tanaman dan dapat menyebabkan kerusakan jaringan secara luas. Infeksi umumnya terjadi saat kelembapan daun tinggi, karena kondisi basah mendukung perkecambahan spora dan pembentukan struktur infeksi awal. Tidak seperti *powdery mildew*, *downy mildew* sangat bergantung pada kelembapan tinggi untuk memulai infeksi.

B Gejala Downy Mildew pada Tanaman Melon

Gejala awal ditandai dengan munculnya bercak klorosis berwarna kuning muda pada permukaan daun yang tidak terbatas oleh tulang daun dan berbentuk tidak beraturan. Seiring waktu, bercak ini membesar dan berubah warna menjadi kuning terang, kemudian berkembang menjadi nekrosis berwarna coklat keoranyean hingga coklat tua. Infeksi tingkat lanjut dapat menyebabkan kerusakan parah pada daun dan menurunkan kemampuan fotosintesis tanaman, sehingga tanaman tampak kerdil dan tidak tumbuh optimal.

2.1.5 Powdery Mildew

Powdery mildew adalah penyakit yang disebabkan oleh berbagai jenis jamur patogen yang menyerang tanaman melon (*Cucumis melo*). Penyakit ini dapat menyebabkan kerusakan signifikan pada pertumbuhan dan hasil tanaman melon. *Powdery mildew* pada tanaman melon terutama disebabkan oleh spesies *Erysiphe cichoracearum* dan *Sphaerotheca fuliginea*, yang termasuk dalam ordo *Erysiphales*. Infeksi ini menurunkan kualitas dan kuantitas hasil panen melon, serta dapat mengurangi nilai ekonominya.

A Penyebab Powdery Mildew pada Tanaman Melon

Powdery mildew pada melon terutama disebabkan oleh jamur *Erysiphe cichoracearum* dan *Sphaerotheca fuliginea*. Jamur ini berkembang biak melalui spora yang terbawa angin dan dapat menginfeksi bagian tanaman seperti daun, batang, dan tunas. Spora ini biasanya tumbuh dan berkecambah pada permukaan tanaman yang memiliki kelembapan yang cukup, tetapi tidak basah, sehingga menjadikannya berbeda dari penyakit jamur lainnya yang lebih berkembang pada kondisi lembap [7].

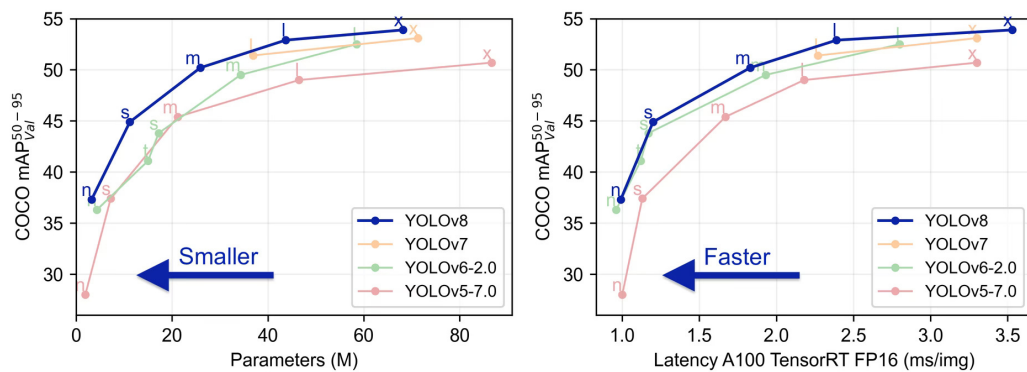
B Gejala Powdery Mildew pada Tanaman Melon

Gejala awal serangan *powdery mildew* pada tanaman melon adalah munculnya bercak putih seperti bubuk di permukaan daun, terutama di bagian atas dan bawah. Bercak ini disebabkan oleh pertumbuhan miselium jamur dan konidia. Seiring waktu, bercak tersebut dapat menyebar, membuat daun menguning dan kering. Serangan parah dapat menyebabkan penurunan fotosintesis, menghambat pertumbuhan tanaman, dan mengurangi kualitas buah melon [8]. Selain itu, pada beberapa kasus, infeksi yang berat bisa menyebabkan kematian tanaman melon.

2.1.6 YOLOv8

YOLO, yang merupakan singkatan dari *You Only Look Once*, telah berkembang pesat sejak pertama kali diperkenalkan pada tahun 2015. Puncak dari evolusi YOLO tercapai dengan dirilisnya YOLOv8 pada Januari 2023. Model YOLO lebih mengutamakan kecepatan *real-time* dan akurasi klasifikasi yang tinggi, namun tetap efisien dalam penggunaan sumber daya komputasi [9].

Setiap iterasi model YOLO semakin menekankan kinerja tinggi dan kemampuan deteksi yang cepat, menjadikannya solusi yang optimal untuk berbagai aplikasi, seperti inspeksi kualitas otomatis dalam industri deteksi cacat. Rilis terbaru, YOLOv8, dipublikasikan oleh Ultralytics pada Januari 2023, yang juga merilis YOLOv5. Dalam perbandingan dengan YOLOv5 dan YOLOv6 yang dilatih pada resolusi gambar 640, YOLOv8 menunjukkan *throughput* lebih baik dengan jumlah *parameter* yang setara, yang menandakan perubahan besar dalam arsitektur yang lebih hemat sumber daya perangkat keras. Dengan hasil *benchmarking* awal yang dikeluarkan oleh Ultralytics, YOLOv8 diyakini akan menjadi model unggul dalam sistem dengan keterbatasan perangkat keras, mengutamakan kecepatan *inferensi* tinggi [10].



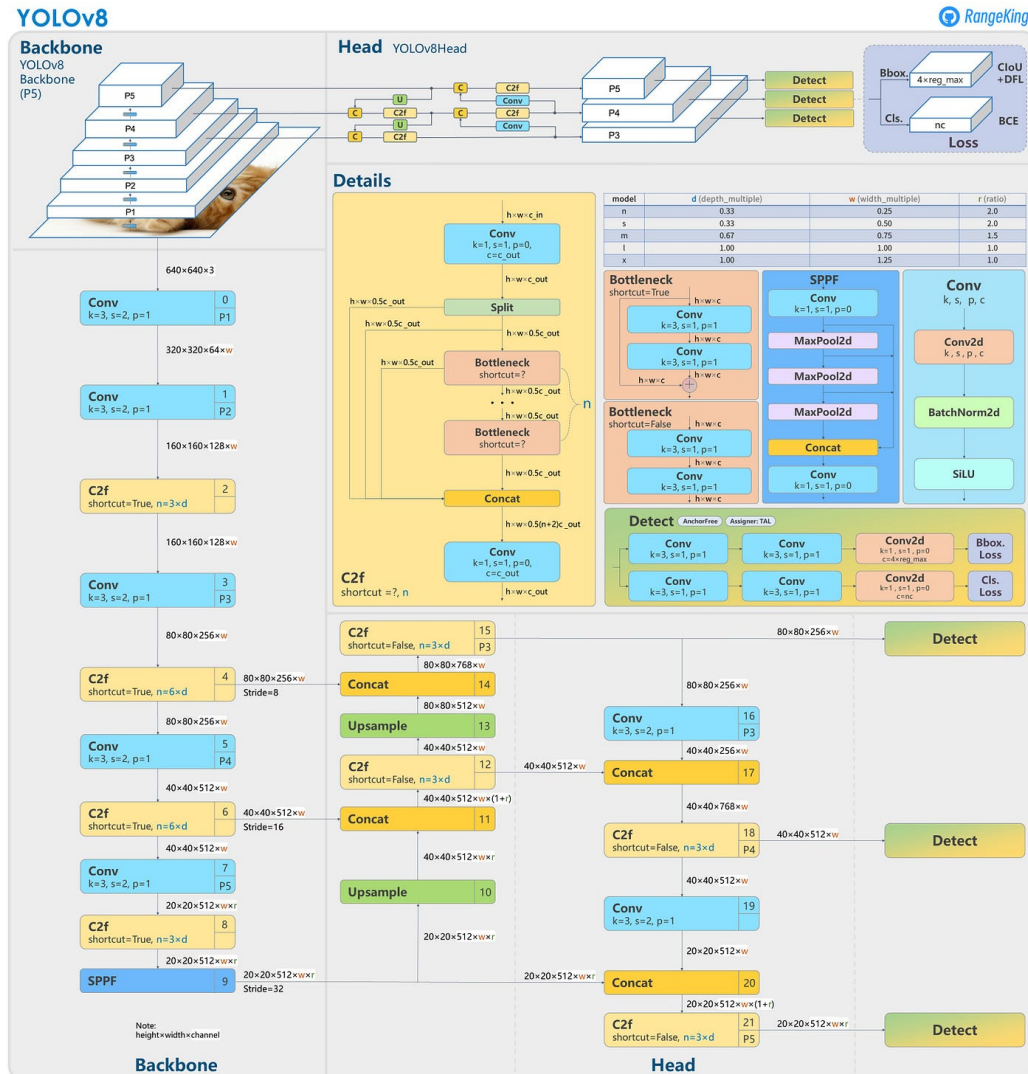
Gambar 2.1. Grafik Evolusi YOLO

Model YOLO telah mengalami serangkaian peningkatan signifikan sejak versi pertamanya pada tahun 2016. Dimulai dengan YOLOv1 yang memperkenalkan konsep deteksi objek secara *real-time* dengan satu kali proses lewat jaringan saraf, setiap versi berikutnya (seperti YOLOv2, v3, dan v4) terus meningkatkan keakuratan dan kecepatan deteksi menggunakan teknik seperti *batch normalization*, jaringan *feature pyramid*, dan koneksi *Cross-Stage Partial*. YOLOv5 menambahkan penjadwalan laju pembelajaran otomatis, pelatihan presisi campuran, serta teknik pelatihan yang membuat implementasi lebih mudah. YOLOv6 dan v7 semakin memperbaiki fungsionalitas *activation function* dan strategi *data augmentation*, meningkatkan efisiensi serta akurasi deteksi objek. Versi terakhir, YOLOv8, lebih diarahkan pada aplikasi industri, khususnya untuk deteksi cacat, dengan fokus menjaga kecepatan *inferensi* yang tinggi serta meningkatkan akurasi, sesuai dengan kebutuhan inspeksi kualitas otomatis di sektor manufaktur.

Arsitektur YOLOv8 masih menggunakan *backbone* yang serupa dengan YOLOv5, tetapi ada perubahan pada *CSPLayer* yang kini disebut *modul C2f*. Modul C2f (*cross-stage partial bottleneck*) mengoptimalkan fitur tingkat tinggi dengan informasi kontekstual untuk meningkatkan akurasi deteksi [11]. *Backbone* ini menghasilkan *feature maps* dengan resolusi yang bervariasi, sementara YOLOv8 mengimplementasikan model tanpa *anchor* dan menggunakan kepala terpisah untuk menangani tugas keberadaan objek, klasifikasi, dan *regresi* secara independen. Desain ini memungkinkan setiap cabang untuk berfokus pada tugasnya masing-masing, meningkatkan akurasi keseluruhan model. Pada lapisan *output*, YOLOv8 menggunakan fungsi aktivasi *sigmoid* untuk skor keberadaan objek dan *softmax* untuk probabilitas kelas, mewakili kemungkinan kelas yang dimiliki oleh setiap objek.

YOLOv8 mengoptimalkan *loss function* dengan menggunakan *CIoU* dan *DFL* untuk *regresi* kotak pembatas, serta *binary cross-entropy* untuk klasifikasi. *CIoU* (*Complete IoU*) adalah metode untuk *regresi* kotak pembatas yang mempertimbangkan tiga ukuran geometris: area tumpang tindih, jarak titik tengah, dan rasio aspek. *DFL* (*Distribution Focal Loss*) adalah varian dari *Generalized Focal Loss* (*GFL*), yang memperluas *Focal Loss* asli ke versi kontinu. *GFL* memungkinkan solusi optimasi global untuk menargetkan nilai kontinu yang diinginkan, bukan hanya nilai diskrit. *DFL*, sebagai bagian dari *GFL*, membuat jaringan lebih fokus pada pembelajaran probabilitas di sekitar lokasi kotak pembatas target dalam distribusi yang lebih fleksibel dan acak. Fungsi-fungsi kerugian ini meningkatkan kinerja deteksi objek, terutama pada objek yang lebih kecil. Rangkaian arsitektur YOLOv8 dapat dilihat pada gambar berikut.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.2. Arsitektur YOLOv8

Pada arsitektur YOLOv8, *backbone* adalah tahap pertama yang bertugas mengekstrak fitur dari gambar *input*. YOLOv8 menggunakan model CNN (*Convolutional Neural Network*) yaitu CSP Darknet 53. Model CSP ini berfungsi untuk membagi *feature maps* menjadi dua bagian: satu untuk lapisan konvolusi, dan lainnya untuk digabungkan dengan hasil konvolusi. *Backbone* menghasilkan *feature maps* dengan resolusi yang bervariasi, yang kemudian digunakan dalam tahap berikutnya. *Neck* bertanggung jawab untuk menggabungkan *feature maps* dari *backbone* dengan resolusi yang berbeda. Sedangkan bagian *head* bertugas mendeteksi objek, terbagi dalam tiga cabang: satu untuk objek kecil, satu lagi untuk objek berukuran sedang, dan terakhir untuk objek besar.

2.1.7 Confusion Matrix

Confusion matrix merupakan alat penting dalam evaluasi model klasifikasi pada pembelajaran mesin, yang digunakan untuk menilai kinerja dan efektivitas model dalam membuat prediksi. Biasanya, *confusion matrix* digambarkan dalam bentuk matriks persegi yang terdiri dari baris dan kolom. Baris mewakili kelas aktual, sedangkan kolom mewakili kelas yang diprediksi oleh model. Untuk klasifikasi biner, *confusion matrix* berbentuk matriks 2×2 , sementara untuk klasifikasi multi-kelas, matriks ini berkembang menjadi ukuran $k \times k$, di mana k menunjukkan jumlah kelas yang ada. Matriks ini digunakan untuk menilai sejauh mana model dapat mengklasifikasikan data dengan benar. Di bidang rekayasa perangkat lunak, Font et al. menerapkan *confusion matrix* untuk membandingkan antara nilai yang diprediksi dan nilai aktual yang dihasilkan oleh elemen-elemen dalam model. *Confusion matrix* ini secara khusus menggunakan empat metrik utama: *true positive* (TP), *false positive* (FP), *true negative* (TN), dan *false negative* (FN). Dari nilai-nilai ini, kita dapat menghasilkan beberapa indikator kinerja, seperti *precision*, *recall*, dan *F-score*.

Precision merupakan ukuran akurasi model untuk kelas tertentu dan dapat dihitung dengan menggunakan Persamaan 2.1.

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Recall mengukur sejauh mana model dapat memilih *instance* dari kelas tertentu dari seluruh data yang ada, dan dihitung dengan Persamaan 2.2.

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

F1-score adalah nilai rata-rata harmonis antara *precision* dan *recall*, yang dapat dilihat pada Persamaan 2.3.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.3)$$

Tabel 2.1. *Confusion Matrix*

	Predicted Class		
		P	N
Actual Class	P	<i>True Positive</i>	<i>True Negative</i>
	N	<i>False Positive</i>	<i>False Negative</i>

2.1.8 *mAP (mean Average Precision)*

mAP (mean Average Precision) adalah metrik yang digunakan untuk menilai keakuratan detektor objek di seluruh kelas dalam suatu dataset. *mAP* dihitung dengan mengambil rata-rata nilai *Average Precision (AP)* yang dihitung untuk setiap kelas secara individu. *Average Precision (AP)* mengukur area di bawah kurva *precision-recall* untuk kelas tertentu, yang dihitung menggunakan Persamaan 2.4.

$$AP_{\text{all}} = \sum_n (R_{n+1} - R_n) \cdot P_{\text{interp}}(R_{n+1}) \quad (2.4)$$

Dimana n adalah indeks untuk iterasi saat ini, R_{n+1} adalah nilai *recall* pada iterasi berikutnya, R_n adalah nilai *recall* pada iterasi saat ini, dan $P_{\text{interp}}(R_{n+1})$ adalah *precision* interpolasi yang dihitung pada Persamaan 2.5.

$$P_{\text{interp}}(R_{n+1}) = \max_{\tilde{R} \geq R_{n+1}} P(\tilde{R}) \quad (2.5)$$

Dimana R adalah nilai *recall*, $R \geq R_{n+1}$ adalah kondisi di mana nilai *recall* lebih besar atau sama dengan nilai *recall* iterasi berikutnya, dan $P(R)$ adalah nilai *precision* pada level *recall* tertentu. Oleh karena itu, *mean Average Precision (mAP)* dihitung sebagai rata-rata dari *AP* di semua kelas, seperti yang ditunjukkan pada Persamaan 2.6.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.6)$$

Dimana AP_i adalah nilai *AP* untuk iterasi ke- i , dan N adalah jumlah total kelas yang dievaluasi. Dalam YOLOv8, terdapat dua jenis *mAP* yang sering digunakan, yaitu *mAP@0.5* dan *mAP@0.5:0.95*. *mAP@0.5* dihitung dengan

ambang batas *IoU* sebesar 0.5, yang digunakan untuk mengevaluasi performa deteksi secara keseluruhan. Prediksi dianggap benar ketika nilai *mAP* mencapai atau melebihi 0.5. Sedangkan *mAP@0.5:0.95* mengukur *mAP* dengan rentang *IoU* antara 0.5 hingga 0.95 dengan langkah 0.05. *IoU* (*Intersection over Union*) adalah metrik yang digunakan dalam deteksi objek untuk menilai true positives dan false positives dalam kumpulan prediksi. Pemilihan nilai threshold *IoU* yang tepat sangat penting untuk evaluasi yang akurat.

2.1.9 Evaluasi Model

Evaluasi model dilakukan dengan memperhatikan beberapa metrik seperti *Losses*, *Precision*, *Recall*, *F1-score*, dan *mAP* (*mean Average Precision*) yang dihasilkan dari proses pelatihan menggunakan YOLOv8. Evaluasi ini dapat dilakukan melalui matriks visualisasi dan statistik yang diperoleh selama proses pelatihan. Matriks dan statistik yang diperoleh mencakup:

- ***Losses***: Menunjukkan kerugian selama pelatihan yang membantu menilai seberapa efektif model dalam belajar. Penurunan kerugian menunjukkan peningkatan kinerja model dalam mendeteksi dan mengklasifikasikan objek dengan akurasi.
- ***Confusion Matrix***: Digunakan untuk mengevaluasi akurasi model dengan membandingkan nilai yang diprediksi dengan nilai yang sebenarnya.
- ***Precision*, *Recall*, dan *F1-score***: Metrik ini memberikan gambaran yang lebih mendalam tentang kinerja model pada setiap kelas. *Precision* yang tinggi menunjukkan lebih sedikit false positives, *Recall* yang tinggi menunjukkan lebih sedikit false negatives, dan nilai *F1* yang tinggi menandakan keseimbangan yang baik antara *Precision* dan *Recall*.
- ***mAP* (*mean Average Precision*)**: *mAP* adalah metrik yang banyak digunakan dalam deteksi objek untuk memberikan gambaran umum tentang kinerja model di seluruh kelas dan sangat berguna untuk membandingkan model atau melacak kinerja model dari waktu ke waktu.