

## DETEKSI PENYAKIT DAUN MANGGA MENGUNAKAN YOLOV8 BERBASIS DEEP LEARNING

Nathanael Putrapratama Purnawan, Nengah Widya Utami, Jauzaa Maylia Suhendro

Sistem Informasi, Universitas Primakara

JL. Tukad Badung No. 135 Renon, Denpasar-Bali, 80226, Indonesia

nael.purnawan@gmail.com

### ABSTRAK

Mangga (*Mangifera indica* L.) merupakan komoditas hortikultura penting di Indonesia, namun produktivitasnya sering terganggu oleh penyakit dan hama yang umumnya menyerang bagian daun sebagai indikator awal kesehatan tanaman. Proses identifikasi penyakit daun mangga yang masih dilakukan secara manual memiliki keterbatasan dari segi waktu, konsistensi, dan akurasi, sehingga berpotensi menimbulkan kesalahan diagnosis. Oleh karena itu, diperlukan sistem deteksi berbasis citra digital yang mampu mendukung identifikasi penyakit secara cepat dan objektif. Penelitian ini mengembangkan sistem deteksi penyakit dan hama daun mangga menggunakan algoritma YOLOv8 berbasis *deep learning*. Dataset yang digunakan terdiri dari 4000 citra daun mangga dengan delapan kelas, yaitu tujuh kelas penyakit dan hama serta satu kelas daun sehat. Pengolahan data mengikuti metodologi CRISP-DM yang mencakup tahap pemahaman data, persiapan data, pemodelan, evaluasi, dan implementasi. Evaluasi model dilakukan menggunakan *confusion matrix*, F1-score, dan *mean Average Precision* (mAP). Hasil pengujian menunjukkan bahwa model YOLOv8 memiliki performa deteksi yang sangat baik dengan nilai mAP50 mencapai 0,99. Model kemudian diimplementasikan ke dalam sistem berbasis web menggunakan *framework* Flask untuk mendukung deteksi penyakit daun mangga secara praktis.

**Kata kunci :** YOLOv8, Deteksi Penyakit Daun Mangga, Deep Learning, Computer Vision, Flask, Roboflow

### 1. PENDAHULUAN

Mangga (*Mangifera indica* L.) merupakan komoditas buah tropis yang banyak dibudidayakan di Indonesia karena kesesuaian kondisi iklim dan lingkungannya [1], [2]. Namun, produktivitas tanaman mangga sering mengalami penurunan akibat serangan penyakit dan hama yang umumnya menyerang bagian daun sebagai indikator awal kesehatan tanaman, seperti antraknosa dan powdery mildew [3]. Serangan penyakit pada daun dapat menghambat proses fotosintesis dan berdampak pada penurunan produktivitas tanaman dalam jangka panjang. Hingga saat ini, proses identifikasi penyakit daun mangga masih banyak dilakukan secara manual, yang memiliki keterbatasan dari segi waktu, konsistensi, dan akurasi, terutama karena kemiripan gejala visual antar penyakit. Kondisi ini berpotensi menimbulkan kesalahan diagnosis dan keterlambatan dalam penanganan, sehingga diperlukan pendekatan yang lebih objektif dan efisien melalui pemanfaatan teknologi berbasis data dan sistem otomatis [4].

Perkembangan teknologi Machine Learning (ML) dan Deep Learning (DL) dalam beberapa tahun terakhir telah mendorong pemanfaatan Computer Vision pada sektor pertanian, khususnya dalam deteksi penyakit tanaman berbasis citra. Convolutional Neural Network (CNN) menjadi arsitektur utama dalam pengolahan citra, sementara algoritma You Only Look Once (YOLO) dikenal unggul dalam kecepatan dan akurasi deteksi objek [5]. Beberapa penelitian sebelumnya telah menunjukkan efektivitas YOLO dalam mendeteksi penyakit tanaman, seperti penerapan YOLOv5s pada deteksi penyakit dan hama tanaman jagung [6]. Meskipun demikian, pendekatan

tersebut masih memiliki keterbatasan dalam mendeteksi objek berukuran kecil serta belum banyak dikembangkan dalam bentuk sistem berbasis web yang dapat diakses secara praktis. Oleh karena itu, penelitian ini menggunakan YOLOv8 yang menawarkan peningkatan akurasi, efisiensi inferensi, serta kemampuan deteksi objek kecil dengan kebutuhan komputasi yang lebih ringan, sehingga lebih sesuai untuk dioptimalkan dan diimplementasikan pada sistem berbasis web [7], [8], [9].

Sebagai upaya penyelesaian permasalahan tersebut, penelitian ini mengembangkan sistem deteksi penyakit dan hama daun mangga berbasis citra digital dengan memanfaatkan algoritma YOLOv8. Pengolahan dan manajemen dataset dilakukan menggunakan platform Roboflow yang menyediakan fitur anotasi, augmentasi, dan ekspor data dalam format yang sesuai dengan YOLOv8 [8]. Dataset citra daun mangga diproses melalui tahap preprocessing untuk memperkaya variasi data latih, dengan objek berupa daun yang difoto pada latar bersih dan pencahayaan yang konsisten. Model yang telah dilatih kemudian diimplementasikan ke dalam sistem berbasis web menggunakan *framework* Flask karena sifatnya yang ringan dan mudah diintegrasikan dengan model machine learning. Sistem ini memproses citra statis yang diambil langsung melalui kamera perangkat tanpa mendukung deteksi real-time dan tanpa memerlukan unggah gambar ke server. Selain menampilkan hasil deteksi, sistem juga menyediakan informasi penyebab penyakit serta rekomendasi penanganan sederhana. Pengembangan sistem mengikuti metodologi CRISP-DM yang mencakup

tahapan business understanding, data understanding, data preparation, modeling, evaluation, dan deployment untuk memastikan alur penelitian yang terstruktur [10]. Penelitian ini difokuskan pada deteksi lima jenis penyakit dan dua jenis hama utama pada daun mangga dengan tujuan menghasilkan sistem deteksi berbasis web yang praktis, mudah diakses, dan dapat digunakan secara langsung oleh petani maupun masyarakat umum sebagai pendukung pengambilan keputusan penanganan di lapangan.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Aufa Faiz Setyawan [11] membahas deteksi penyakit daun mangga menggunakan algoritma YOLOv8 berbasis deep learning. Penelitian ini menunjukkan bahwa peningkatan keberagaman dataset memberikan dampak signifikan terhadap performa model dalam mendeteksi penyakit daun mangga.

Hasil penelitian menunjukkan bahwa akurasi model meningkat dari 48% menjadi 71% dan nilai F1-score naik dari 65% menjadi 83% setelah penambahan dataset yang lebih variatif. Temuan ini menegaskan bahwa kualitas dan variasi data latih berperan penting dalam meningkatkan kinerja algoritma YOLOv8 pada kasus deteksi penyakit daun mangga.

### 2.2. Tanaman Mangga dan Penyakitnya

Mangga (*Mangifera indica* L.) merupakan komoditas hortikultura penting di Indonesia dengan produktivitas yang sangat dipengaruhi oleh kesehatan daun [1]. Daun mangga rentan terhadap berbagai penyakit dan hama seperti antraknosa, *powdery mildew*, *bacterial canker*, *die back*, dan *sooty mould* yang umumnya ditandai dengan perubahan warna, bercak, atau deformasi permukaan daun [3]. Gejala visual tersebut dapat dimanfaatkan sebagai dasar deteksi penyakit berbasis citra digital [2].

### 2.3. Deep Learning

*Deep Learning* merupakan pendekatan *Machine Learning* yang mampu mengekstraksi fitur secara otomatis dari data tidak terstruktur seperti citra [12]. Dalam bidang visi komputer, pendekatan ini banyak diterapkan pada deteksi penyakit tanaman menggunakan arsitektur *Convolutional Neural Network* (CNN) [13].

### 2.4. YOLOv8 (You Only Look Once version 8)

YOLO (*You Only Look Once*) merupakan algoritma deteksi objek berbasis CNN yang melakukan prediksi *bounding box* dan kelas objek dalam satu tahap inferensi [8]. YOLOv8 merupakan versi terbaru yang mengadopsi pendekatan *anchor-free* serta optimasi arsitektur *backbone* dan *head*, sehingga mampu meningkatkan akurasi deteksi terutama pada objek berukuran kecil [14]. Dibandingkan versi YOLO sebelumnya, YOLOv8 menawarkan keseimbangan antara performa, efisiensi

komputasi, dan kemudahan implementasi, sehingga sesuai untuk sistem deteksi berbasis web [9].

### 2.5. Confusion Matrix

*Confusion matrix* merupakan metode evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan label sebenarnya. Pada penelitian ini, *confusion matrix* disajikan dalam bentuk *normalized confusion matrix*, di mana nilai pada setiap sel menunjukkan proporsi prediksi dalam rentang 0 hingga 1 [15]. Normalisasi ini memudahkan analisis performa model pada masing-masing kelas, khususnya pada kasus klasifikasi multi-class dengan distribusi data yang tidak seimbang.

Tabel 1. Confusion matrix

|                        | <i>Predicted Positive</i>     | <i>Predicted Negative</i>     |
|------------------------|-------------------------------|-------------------------------|
| <i>Actual Positive</i> | <i>True Positive</i><br>(TP)  | <i>False Negative</i><br>(FN) |
| <i>Actual Negative</i> | <i>False Positive</i><br>(FP) | <i>True Negative</i><br>(TN)  |

Confusion matrix merepresentasikan hubungan antara hasil prediksi model dan kondisi sebenarnya melalui empat komponen utama, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Dalam deteksi penyakit daun mangga, confusion matrix digunakan untuk menilai kemampuan model dalam membedakan antar kelas penyakit dan kondisi daun sehat, di mana nilai diagonal menunjukkan prediksi yang benar dan nilai di luar diagonal menunjukkan kesalahan klasifikasi. Confusion matrix menjadi dasar perhitungan metrik evaluasi lanjutan seperti *precision*, *recall*, *F1-score*, dan akurasi untuk memberikan gambaran kinerja model secara kuantitatif dan menyeluruh [16].

### 2.6. F1-Score

F1-score merupakan metrik evaluasi yang digunakan untuk mengukur keseimbangan antara *precision* dan *recall* dalam model klasifikasi, sehingga sangat sesuai diterapkan pada permasalahan multi-class dengan distribusi data yang tidak seimbang [17]. Nilai *precision* dan *recall* diperoleh berdasarkan *confusion matrix* dan dijabarkan pada persamaan berikut:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

F1-score dihitung sebagai rata-rata harmonik dari *precision* dan *recall* dengan persamaan:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Nilai F1-score berada pada rentang 0 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan performa model yang semakin baik. Pada kasus klasifikasi multi-class, nilai F1-score dihitung untuk setiap kelas kemudian dirata-ratakan untuk

merepresentasikan kinerja model secara keseluruhan [18].

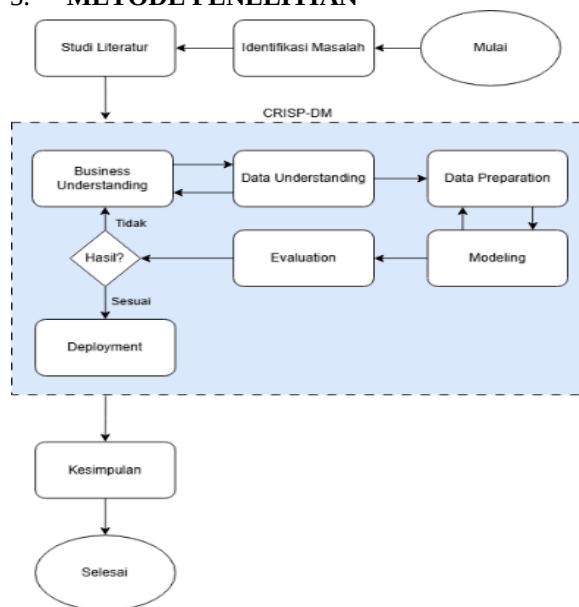
## 2.7. Roboflow

Roboflow adalah *platform* web yang berfungsi untuk mengelola data set secara komprehensif, dari pengumpulan hingga optimasi data untuk pengembangan model *computer vision*. Roboflow menyediakan berbagai alat yang mempermudah proses *preprocessing*, seperti augmentasi data, mengubah ukuran gambar, anotasi, dan mengelola *dataset* secara kolaboratif [19].

## 2.8. Flask

Flask merupakan *framework* Python yang bersifat ringan dan fleksibel untuk pengembangan aplikasi berbasis web. *Framework* ini termasuk dalam kategori *micro-framework* karena tidak memerlukan dependensi tambahan yang kompleks dalam penggunaannya [20]. Dengan sintaks yang sederhana dan mudah dipahami, Flask memberikan keleluasaan bagi pengembang dalam membangun aplikasi web secara cepat dan efisien, serta didukung oleh komunitas yang aktif sehingga memudahkan proses pengembangan dan pemecahan masalah [21], [22].

## 3. METODE PENELITIAN



Gambar 1. Alur penelitian

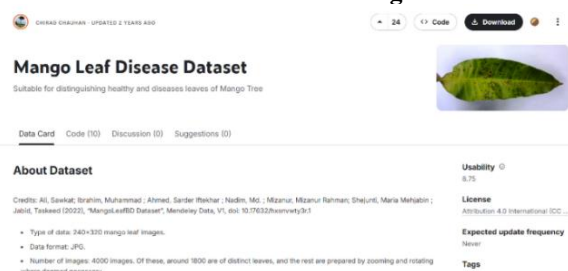
Penelitian ini menggunakan metodologi CRISP-DM (Cross Industry Standard Process for Data Mining) yang terdiri dari enam tahapan, yaitu business understanding, data understanding, data preparation, modeling, evaluation, dan deployment [10]. Tahap awal dilakukan melalui identifikasi permasalahan dan studi literatur untuk menentukan tujuan penelitian serta research gap terkait deteksi penyakit daun mangga. Dataset yang digunakan berasal dari “Mango Leaf Disease Dataset” dari Kaggle, terdiri dari 4000 citra daun mangga dengan delapan kelas (lima

penyakit, dua hama, dan satu kelas sehat), yang dibagi menjadi 80% data pelatihan dan 20% data pengujian. Eksplorasi dan analisis dataset dilakukan untuk memahami distribusi kelas dan kualitas data sebelum memasuki tahap pemodelan.

Tahap data preparation mencakup proses anotasi, penyesuaian ukuran citra, dan augmentasi data menggunakan platform Roboflow agar model mampu melakukan generalisasi dengan baik. Selanjutnya, algoritma YOLOv8 diterapkan pada tahap modeling untuk membangun model deteksi penyakit daun mangga dengan konfigurasi parameter tertentu. Evaluasi model dilakukan menggunakan Confusion Matrix dan F1-Score guna menilai akurasi dan keseimbangan prediksi antar kelas. Model terbaik kemudian diimplementasikan ke dalam sistem berbasis web menggunakan framework Flask, yang memproses citra statis dari kamera perangkat tanpa deteksi real-time. Sistem dilengkapi dengan confidence threshold sebesar 0,70 dan mekanisme validasi tambahan untuk memastikan hasil deteksi yang relevan dan dapat digunakan secara praktis oleh pengguna.

## 4. HASIL DAN PEMBAHASAN

### Data Understanding

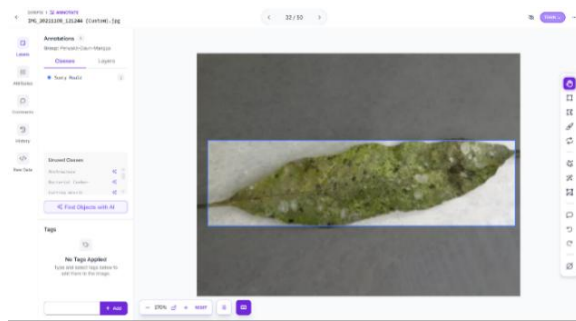


Gambar 2. Sumber dataset

Penelitian ini menggunakan pendekatan *supervised learning* untuk melakukan deteksi penyakit pada daun mangga, di mana setiap citra diberi label sesuai kelasnya agar model dapat mempelajari fitur visual secara efektif. Dataset yang digunakan berasal dari platform Kaggle dan terdiri dari tujuh kelas penyakit daun mangga serta satu kelas daun sehat [23]. Seluruh citra berformat *.jpg* dengan resolusi  $240 \times 320$  piksel dan telah melalui proses augmentasi berupa *zoom* dan rotasi untuk meningkatkan variasi data latih.

### 4.1. Data Annotation

Salah satu langkah sebelum dilakukannya training terhadap model yaitu melakukan anotasi data sesuai dengan *class*, anotasi data dilakukan dengan cara manual menggunakan platform Roboflow. Adapun tahap anotasi data dilakukan sesuai dengan *class* dari data yang sudah didapatkan melalui platform kaggle dengan Jumlah 7 *class* penyakit daun mangga di antaranya (*Anthracoese*, *Bacterial Canker*, *Cutting Weevil*, *Die Back*, *Gall Midge*, *Powder Mildew*, *Sooty Mould*) dan 1 *class* daun sehat.



Gambar 3. Proses data annotation dengan “bounding box”

## 4.2. Data Preprocessing

Tahap berikutnya adalah pemilihan *preprocessing* data. *Preprocessing* data adalah tahapan selanjutnya dalam memproses data mentah menjadi data yang selanjutnya akan siap digunakan. Dalam penelitian ini *preprocessing* dilakukan untuk transformasi gambar menjadi data yang siap digunakan, dengan menggunakan *tools* yang terdapat di dalam platform Roboflow bernama *auto-orient* yang berfungsi untuk memperbaiki orientasi gambar secara otomatis agar model dapat memahami *dataset* sehingga mendapatkan hasil deteksi yang akurat.

## 4.3. Data Augmentation

Data *augmentation* adalah teknik pengolahan data pada *image classification* yang berfungsi untuk meningkatkan jumlah data, dan variasi *dataset* dengan cara membuat salinan data asli yang sedikit dimodifikasi. Dalam platform roboflow augmentasi yang digunakan dalam penelitian ini yaitu:

- Flip:** Yaitu membalik gambar dari kiri, kanan, atas, dan bawah, di mana ini dapat membantu model dalam mengenali data dari sudut dan posisi yang berbeda beda.
- Rotation:** Yaitu memutar gambar secara acak, ini dapat membantu model tetap akurat dalam mengenali daun meskipun objek miring, atau tidak rata pada data yang nyata. Pemilihan *rotation* pada penelitian ini yaitu di antara  $-15^\circ$  sampai  $+15^\circ$ .
- Hue Adjustment:** Yaitu mengubah sedikit warna pada gambar, dengan mengubah sedikit warna pada gambar dapat membantu model dalam menghadapi variasi kondisi warna yang berbeda dan dapat membantu model memahami pola yang berbeda karena ada sedikit perbedaan warna. Penyesuaian *hue* yang dipilih dalam penelitian ini yaitu  $-15\%$  sampai  $+15\%$ .
- Saturation Adjustment:** Yaitu mengatur tingkat kecerahan warna, ini dapat membantu model dalam mengenali daun meskipun warna terlihat lebih pudar maupun lebih pekat. Penyesuaian *saturation* yang dipilih dalam penelitian ini yaitu  $-25\%$  sampai  $+25\%$ .

- Blur:** yaitu menambahkan efek *blur* ringan, ini berfungsi agar model tetap bisa mengenali daun walaupun kondisi kamera di lapangan kurang fokus ataupun *blur*. Penambahan *blur* yang dilakukan dalam persiapan *dataset* pada penelitian ini yaitu  $1.5\text{px blur}$ .
- Noise:** Yaitu menambah *noise* acak pada gambar, yang dapat membantu model untuk mengenali daun meskipun terdapat *noise* yang disebabkan oleh faktor eksternal misalnya kamera hp. Jumlah *noise* yang ditambahkan ke dalam *dataset* penelitian ini mencapai hingga  $0.22\%$ .

## 4.4. Data Split

Tahap selanjutnya yaitu melakukan distribusi data pada setiap *class*. Pada tahap ini, *dataset* memiliki jumlah sebanyak 9577 gambar yang akan dibagi untuk *training* model dan pengujian model. Proses pemisahan data dilakukan dengan membagi data menjadi tiga bagian sesuai dengan kebutuhan untuk pengembangan model sebagai berikut:

- Data Training:** Data *training* memiliki pembagian dengan alokasi yang terbesar di antara lainnya yaitu sebanyak  $88\%$  dari total *dataset* dengan jumlah 8397 gambar, Hal ini berguna agar model dapat mempelajari data yang akan digunakan.
- Data Validation:** Data digunakan untuk mengevaluasi performa model yang dijalankan selama proses *training* dan membantu untuk melakukan validasi prediksi. Data *validation* menggunakan  $6\%$  dari total *dataset* yaitu sebanyak 599 gambar.
- Data Testing:** Data yang akan digunakan untuk mengukur performa akhir dari hasil deteksi model. Data ini tidak digunakan dalam kedua proses sebelumnya, sehingga dapat memberikan evaluasi yang objektif tentang kemampuan model untuk mengatasi data baru selanjutnya. Data Testing ditetapkan sejumlah  $6\%$  dari total *dataset* yaitu sebanyak 581 gambar.

Pada penelitian ini, pembagian data dilakukan dengan platform Roboflow. Selanjutnya *file dataset* yang sudah dilakukan *labelling* dan distribusi akan diunduh dengan *image* dan *annotation* format menggunakan YOLOv8 untuk selanjutnya dilakukan *training* model.

## 4.5. Modelling

Tahap selanjutnya *modelling*, yaitu tahap merancang model untuk dilakukan proses *training* hingga mendapatkan akurasi terbaik.

## 4.6. Library Pendukung

Dalam implementasi pengembangan model *machine learning*, beberapa *library* pendukung digunakan untuk proses pengembangan, pelatihan. Dan evaluasi dalam model YOLO. Beberapa *library* yang digunakan akan dijelaskan pada tabel berikut.

Tabel 2. Library pendukung

| No | Nama Library | Penjelasan                                                                                                                       |
|----|--------------|----------------------------------------------------------------------------------------------------------------------------------|
| 1  | Ultralytics  | Library yang digunakan untuk memanggil model yang akan digunakan dalam penelitian ini yaitu YOLO.                                |
| 2  | Shutil       | Library standar dari python yang fungsinya untuk memindahkan, menyalin, atau menghapus direktori.                                |
| 3  | Os           | Library bawaan python untuk berinteraksi dengan sistem operasi, seperti membuat folder, mengakses file dan menavigasi direktori. |
| 4  | Cv2          | Library yang populer untuk memproses dan menyimpan hasil gambar yang telah diberi bounding box.                                  |
| 5  | Flask        | Library python yang berfungsi sebagai framework untuk pembuatan website.                                                         |

#### 4.7. Hyperparameter dan Parameter Model

Dalam penelitian ini, model YOLOv8 dirancang dan dilatih untuk melakukan tugas deteksi penyakit pada daun mangga dengan menggunakan parameter utama untuk mencapai performa model yang maksimal. Proses penyesuaian metrik parameter model melibatkan berulang kali percobaan hingga model dapat berjalan secara optimal. Setelah dilakukan beberapa percobaan awal dengan konfigurasi parameter bawaan YOLOv8, dilakukan penyesuaian parameter pelatihan (*fine-tuning*) untuk meningkatkan stabilitas dan performa model.

Penyesuaian ini dilakukan berdasarkan hasil observasi dari tahap *training* awal menggunakan parameter bawaan YOLOv8 yang menunjukkan kebutuhan optimasi pada *learning rate*, *batch size*, serta menonaktifkan fitur augmentasi lanjutan. Parameter akhir yang digunakan dalam pelatihan model akan di tunjukan pada tabel 3.

Tabel 1. Hyperparameter dan parameter model

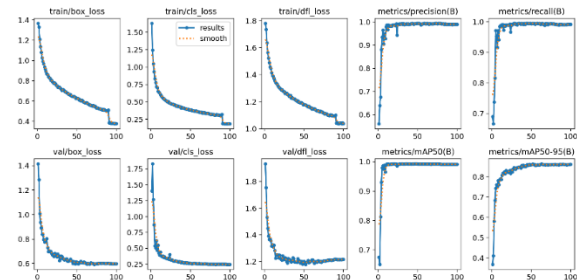
| Parameter           | Nilai Akhir | Keterangan                                                                                                         |
|---------------------|-------------|--------------------------------------------------------------------------------------------------------------------|
| Epoch               | 15          | Jumlah iterasi pelatihan untuk seluruh dataset                                                                     |
| Image Size (imgsz)  | 768         | Resolusi gambar yang digunakan pada saat training                                                                  |
| Batch Size          | 12          | Jumlah gambar yang diproses setiap iterasi                                                                         |
| Learning rate (lr0) | 0.00018     | Mengatur kecepatan pembaruan bobot model                                                                           |
| Optimizer           | AdamW       | Optimizer adaptif yang memberikan kestabilan                                                                       |
| Cosine LR           | True        | Menurunkan <i>learning rate</i> secara bertahap dengan skema <i>cosine annealing</i>                               |
| Freeze              | 10          | Melakukan <i>freeze</i> di bagian <i>backbone</i> agar <i>fine tuning</i> terakhir berjalan untuk <i>head</i> saja |

| Parameter           | Nilai Akhir | Keterangan                                                                                     |
|---------------------|-------------|------------------------------------------------------------------------------------------------|
| Mosaic Augmentation | 0.0         | Mematikan fitur augmentasi untuk menjaga konsistensi data                                      |
| MixUp               | 0.0         | Menonaktifkan teknik augmentasi pencampuran citra yang dilakukan saat <i>tuning</i> sebelumnya |

Berdasarkan konfigurasi di atas, penurunan *learning rate* dan penggunaan *optimizer* AdamW terbukti dapat menjaga stabilitas model, serta dengan *fine tuning* menggunakan parameter di atas dapat memperkuat *head* YOLOv8 dalam melakukan deteksi pada daun mangga.

#### 4.8. Train Model

Proses pelatihan model dilakukan setelah seluruh tahapan persiapan data selesai, dengan pendekatan bertahap mulai dari pelatihan model dasar hingga tahap *fine-tuning* akhir. Pada tahap awal, model YOLOv8 dilatih menggunakan konfigurasi dasar dengan 100 epoch, ukuran gambar 640 piksel, batch size 16, dan *learning rate* 0.002 menggunakan *optimizer* AdamW. Strategi *cosine learning rate* dan *mosaic augmentation* sebesar 0.3 diterapkan untuk meningkatkan kemampuan generalisasi model.



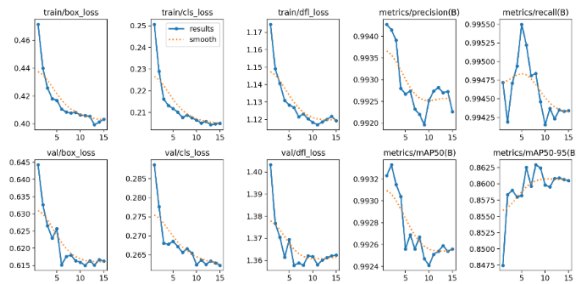
Gambar 4. Hasil evaluasi training base model

Hasil pelatihan awal menunjukkan penurunan *training loss* dan *validation loss* yang stabil hingga akhir pelatihan. Nilai *precision* dan *recall* berada di atas 0.99 dengan mAP50 mencapai 0.99, yang menunjukkan bahwa model mampu mempelajari fitur citra dengan baik dan memiliki kemampuan generalisasi yang tinggi. Meskipun demikian, penyempurnaan tetap diperlukan untuk meningkatkan stabilitas dan efisiensi model.

Tahap selanjutnya adalah Fine-Tuning Fase C (Final) yang bertujuan meningkatkan presisi hasil deteksi. Pada tahap ini dilakukan penyesuaian parameter utama, meliputi epoch 15, ukuran gambar 768, batch size 12, dan *learning rate* 0.00018. Selain itu, sepuluh lapisan awal backbone dibekukan untuk memusatkan proses pelatihan pada bagian head deteksi.

Hasil *fine-tuning* menunjukkan peningkatan kestabilan kurva loss serta performa deteksi yang konsisten. Nilai *precision* dan *recall* tetap berada di atas 0.99 dengan mAP50-95 yang stabil, menandakan

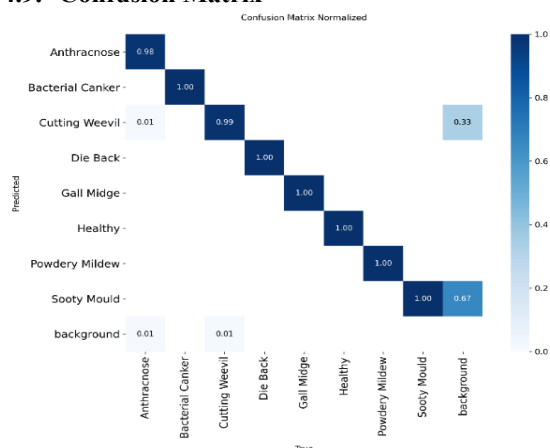
konvergensi model yang optimal. Model hasil fine-tuning ini kemudian dipilih sebagai model akhir karena memiliki keseimbangan terbaik antara akurasi dan efisiensi.



Gambar 5. Hasil evaluasi fine tuning fase C (final)

Hasil visualisasi *results.png* menunjukkan penurunan *loss* yang lebih stabil disertai peningkatan metrik performa model. Nilai *box\_loss* mencapai 0.40 dan *cls\_loss* menurun hingga 0.21, sementara *precision* dan *recall* bertahan di atas 0.994. Nilai mAP50 tercatat sebesar 0.993 dengan mAP50-95 yang stabil di sekitar 0.86, menunjukkan konvergensi model yang optimal tanpa indikasi *overfitting* maupun *underfitting*. Penggunaan *learning rate* yang lebih rendah serta pembekuan sebagian lapisan terbukti efektif dalam meningkatkan stabilitas proses pelatihan. Berdasarkan perbandingan dengan pelatihan dasar, model hasil fine-tuning phase C dipilih sebagai model akhir karena memiliki keseimbangan terbaik antara akurasi, efisiensi, dan kestabilan performa. Model akhir ini selanjutnya digunakan sebagai dasar evaluasi kinerja deteksi pada setiap kelas penyakit daun mangga.

#### 4.9. Confusion Matrix

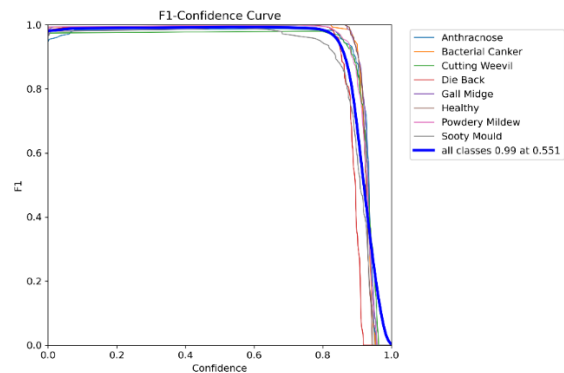


Gambar 6. Confusion matrix

*Confusion Matrix* digunakan untuk mengevaluasi performa klasifikasi model dengan membandingkan hasil prediksi dan label sebenarnya. Berdasarkan *Confusion Matrix Normalized* pada data validasi, model menunjukkan tingkat akurasi yang sangat tinggi pada sebagian besar kelas. Kelas *Bacterial Canker*, *Die Back*, *Gall Midge*, *Healthy*, dan *Powdery Mildew* berhasil diklasifikasikan dengan akurasi 1.00.

Sementara itu, kelas *Anthracnose* memperoleh nilai akurasi 0.98 yang menunjukkan adanya kesalahan klasifikasi dalam jumlah yang sangat kecil. Pada kelas *Sooty Mould*, sebagian citra masih terdeteksi sebagai latar belakang akibat dominasi warna gelap yang menyerupai bayangan atau *noise*. Secara keseluruhan, hasil *Confusion Matrix* menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dengan tingkat kesalahan yang rendah pada hampir seluruh kelas penyakit daun mangga.

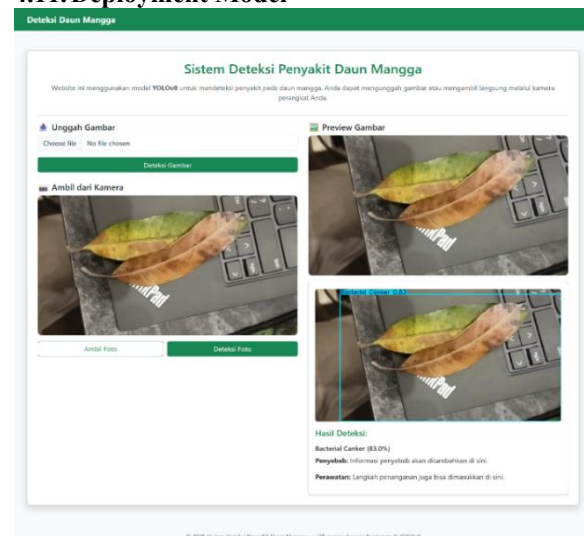
#### 4.10. F1-Confidence Curve



Gambar 7. F1-confidence curve

Dari grafik terlihat bahwa hampir seluruh kelas penyakit memiliki kurva yang konsisten berada di sekitar nilai maksimum 1.0, yang berarti performa deteksi model sangat stabil pada berbagai tingkat *Confidence*. Hanya terdapat sedikit fluktuasi pada beberapa kelas, namun tidak signifikan dan tidak memengaruhi performa keseluruhan. Dengan demikian, Model YOLOv8 dapat dikatakan memiliki tingkat keandalan tinggi serta mampu melakukan klasifikasi penyakit daun mangga dengan presisi yang hampir sempurna.

#### 4.11. Deployment Model



Gambar 8. User Interface Deployment Website

Model YOLOv8 diintegrasikan ke dalam sistem berbasis web menggunakan framework Flask sebagai penghubung antara frontend dan backend. Sistem memproses citra daun mangga yang diunggah atau diambil melalui kamera perangkat, menampilkan hasil deteksi berupa label penyakit dan nilai *confidence*, serta menerapkan ambang batas 0,70 dan validasi tambahan untuk meminimalkan kesalahan deteksi. Antarmuka pengguna dirancang sederhana dan informatif, dilengkapi hasil visual berupa *bounding box* dan informasi penyakit, dengan waktu pemrosesan di bawah 2 detik per gambar sehingga sistem responsif dan layak digunakan.

## 5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sistem deteksi penyakit dan hama daun mangga berbasis citra digital menggunakan algoritma YOLOv8 dengan performa yang sangat baik. Model mampu mengidentifikasi delapan kelas kondisi daun mangga dan menunjukkan akurasi tinggi berdasarkan evaluasi *confusion matrix*, *F1-score*, dan *mean Average Precision* (mAP) dengan nilai mAP50 mencapai 0,99. Sistem diimplementasikan ke dalam aplikasi berbasis web menggunakan *framework* Flask sehingga dapat digunakan secara praktis untuk mendukung identifikasi dini penyakit daun mangga. Pengembangan selanjutnya disarankan untuk memperkaya *dataset* dan mengoptimalkan parameter model guna meningkatkan kemampuan generalisasi, serta mengembangkan sistem ke dalam aplikasi *mobile* dengan kemampuan berjalan secara offline agar dapat digunakan secara lebih luas, khususnya di wilayah dengan keterbatasan akses internet.

## DAFTAR PUSTAKA

- [1] R. Adenia, A. E. Minarno, dan Y. Azhar, "Implementasi Convolutional Neural Network Untuk Ekstraksi Fitur Citra Daun Dalam Kasus Deteksi Penyakit Pada Tanaman Mangga Menggunakan Random Forest," *REPOSITOR*, vol. 4, no. 4, hlm. 473–482, 2022.
- [2] F. K. Attar dan M. U. Kartikawati, "DATA: 10 Daerah Penghasil Mangga Terbesar di Indonesia," *inilah.com*.
- [3] Prof. Ir. I. W. Susila, "Serangga yang berasosiasi pada puru daun mangga (*Mangifera indica* L.) DI BALI," *e-perpus unud*.
- [4] M. Rajuddin, S. Mulya Ulfa, I. Y. Harahap, S. L. Moulana, dan M. Ula, "Sistem Deteksi penyakit pada daun mangga menggunakan metode deteksi tepo sobel dan canny," 2024.
- [5] M. P. Mathew dan T. Y. Mahesh, "Leaf-based disease detection in bell pepper plant using YOLO v5," *Signal Image Video Process.*, vol. 16, no. 3, hlm. 841–847, 2022, doi: 10.1007/s11760-021-02024-y.
- [6] L. Firgia dan S. Thomas, "Deteksi Jenis Penyakit Dan Hama Pada Tanaman Jagung Menggunakan Arsitektur Spatial Pyramid Pooling Pada YOLOv5s," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 8, no. 2, hlm. 452–459, 2023, [Daring]. Tersedia pada: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- [7] I. G. K. A. W. Beratha, N. P. Sutramiani, dan N. K. A. Wirdiani, "Network Reduction Strategy on YOLOv8 Model for Mango Leaf Disease Detection," 2025.
- [8] A. D. Lestari, "Aplikasi deteksi hama dan penyakit pada buah Kakao menggunakan YOLOV8 berbasis website," Universitas PGRI Semarang, 2024.
- [9] M. A. K. Majid dan D. Ariatmanto, "Analisis Perbandingan Kinerja Model YOLO11 dan YOLOv8 dalam Identifikasi Penyakit pada Daun Tomat," *Jurnal Bangkit Indonesia*, vol. 14, no. 2, hlm. 63–72, Okt 2025, doi: 10.52771/bangkitindonesia.v14i2.459.
- [10] N. C. Sastya dan I. Nugraha, "Penerapan Metode CRISP-DM dalam Menganalisis Data untuk Menentukan Customer Behavior di MeatSolution," *Jurnal Pendidikan dan Aplikasi Industri*, hlm. 103–115, 2023, [Daring]. Tersedia pada: <http://ejournal.unis.ac.id/index.php/UNISTEK>
- [11] D. Guna, M. Persyaratan, M. Gelar, dan S. Teknik, "Tugas Akhir Identifikasi Penyakit Pada Daun Mangga menggunakan Algoritma YOLOv8."
- [12] M. R. sirfatullah Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, dan M. Elgar, "Penggunaan python sebagai bahasa pemrograman untuk Machine Learning dan Deep Learning," *Karimah Tauhid*, vol. 2, no. 1, 2023.
- [13] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, dan M. Arif Fauzi Idris, "RESOLUSI: Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, hlm. 509, 2024, [Daring]. Tersedia pada: <https://djournals.com/resolusi>
- [14] I. Munadhif, D. Hasbi Fathoni, dan Mohammad Abu Jamiin, T. Otomasi, dan J. Teknik Kimia, "Pengendalian CCTV menggunakan YOLO ONLY LOOK ONCE (YOLO)," *Seminar Nasional Terapan Riset Inovatif (SENTRINOV) Ke-6 ISAS Publishing Series: Engineering and Science*, vol. 6, no. 1, 2020.
- [15] M. Heydarian, T. E. Doyle, dan R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, vol. 10, hlm. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.
- [16] I. Markoulidakis dan G. Markoulidakis, "Probabilistic Confusion Matrix: A Novel Method for Machine Learning Algorithm Generalized Performance Analysis,"

- Technologies (Basel)*, vol. 12, no. 7, Jul 2024, doi: 10.3390/technologies12070113.
- [17] R. Diallo, C. Edalo, dan O. O. Awe, "Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score," 2025, hlm. 283–312. doi: 10.1007/978-3-031-72215-8\_12.
- [18] K. Takahashi, K. Yamamoto, A. Kuchiba, dan T. Koyama, "Confidence interval for micro-averaged F 1 and macro-averaged F 1 scores," *Applied Intelligence*, vol. 52, no. 5, hlm. 4961–4972, Mar 2022, doi: 10.1007/s10489-021-02635-5.
- [19] N. Sujana, M. Malik Mutoffar, dan A. Azzam Haryanto, "Analisis kinerja YOLOV8 optimalisasi Roboflow untuk deteksi ekspresi wajah emosional dengan Machine Learning," vol. 06, 2024.
- [20] A. Chandra, "Pengembangan aplikasi berbasis web dengan python flask untuk klasifikasi data," 2023.
- [21] D. F. Ningtyas dan N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 1, hlm. 19–34, Apr 2021, doi: 10.25008/janitra.v1i1.120.
- [22] U. Syach dan W. Martyas Edi, "IT-EXPLORE Perancangan Aplikasi web manajemen data produk bisnis perhiasan berbasis flask dan mongoDB," 2024.
- [23] S. Ali dkk., "MangoLeafBD Dataset," <https://www.kaggle.com/datasets/warcoder/man-go-leaf-disease-dataset>. Diakses: 22 Oktober 2025. [Daring]. Tersedia pada: <https://www.kaggle.com/datasets/warcoder/man-go-leaf-disease-dataset>