

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Desain Penelitian

Penelitian ini bertujuan untuk mengidentifikasi penyakit *Alternaria*, *Cucumber Mosaic Virus* (CMV), *Downy Mildew*, dan *Powdery Mildew* pada tanaman melon dengan berfokus pada kondisi daun yang terinfeksi menggunakan algoritma YOLOv8. Metode yang akan digunakan dalam penelitian ini meliputi tahapan berikut:

#### 1. Studi Literatur

Tahap awal ini bertujuan untuk mengumpulkan informasi dan teori yang relevan sebagai dasar penelitian. Kegiatan ini mencakup identifikasi penelitian terdahulu terkait deteksi penyakit tanaman, pemahaman gejala visual dari penyakit *Alternaria*, *CMV*, *Downy Mildew*, dan *Powdery Mildew*, serta studi mengenai algoritma YOLOv8 dalam konteks deteksi objek pada citra tanaman.

#### 2. Pengumpulan Data

Pada tahap ini, dilakukan pengumpulan *dataset* berupa gambar daun melon yang menunjukkan gejala penyakit maupun daun sehat. Sumber data dapat berasal dari pengambilan langsung di lapangan, *dataset* publik, atau kerja sama dengan institusi pertanian. Setiap gambar dilabeli sesuai dengan jenis penyakit oleh ahli atau berdasarkan referensi terpercaya.

#### 3. Pre-processing Data

Gambar yang telah dikumpulkan kemudian diproses agar siap digunakan dalam pelatihan model. Tahapan ini mencakup *resize* gambar ke ukuran standar (misalnya 640x640 piksel), *augmentasi* data (seperti rotasi, *flipping*, penyesuaian pencahayaan), serta proses anotasi menggunakan *tools* seperti *LabelImg* atau *Roboflow* untuk menghasilkan *bounding box* sesuai format YOLO.

#### 4. Pelatihan Model

Model YOLOv8 dilatih menggunakan *dataset* yang telah diproses. *Dataset* dibagi menjadi data pelatihan dan validasi. Parameter pelatihan seperti

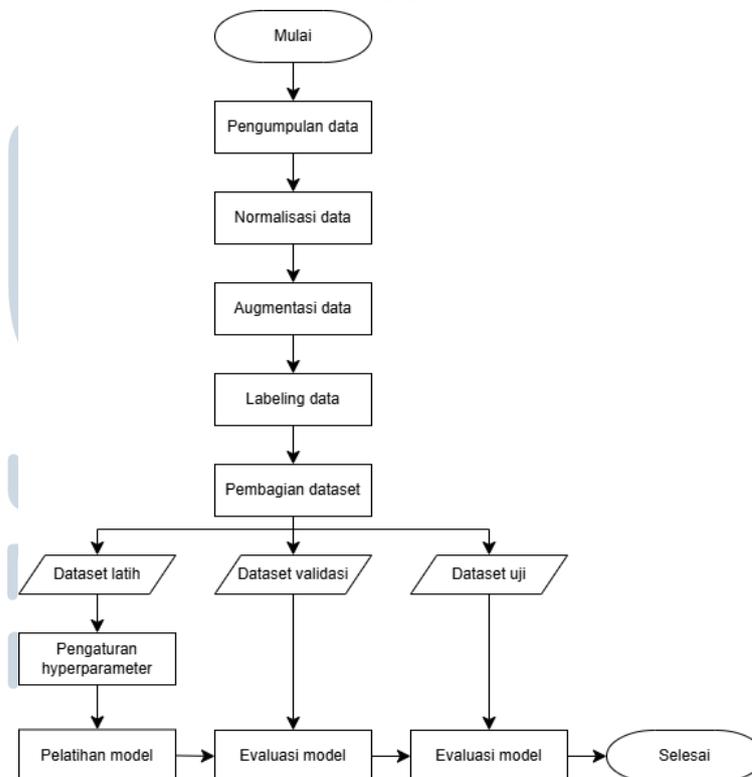
*learning rate*, *batch size*, dan jumlah *epoch* diatur untuk mendapatkan performa optimal. Model dilatih untuk mendeteksi dan mengklasifikasikan area pada daun yang menunjukkan gejala penyakit secara otomatis.

### 5. Uji Coba dan Evaluasi Model

Setelah pelatihan selesai, model diuji menggunakan data uji yang belum pernah dilihat sebelumnya. Evaluasi dilakukan menggunakan metrik seperti *mean Average Precision (mAP)*, *precision*, *recall*, dan *confusion matrix*. Hasil evaluasi digunakan untuk menilai efektivitas model dalam mengenali masing-masing penyakit dan menjadi dasar untuk perbaikan lebih lanjut jika diperlukan.

## 3.2 Alur Penelitian

Alur kerja penelitian ini divisualisasikan dalam Gambar 3.1, yang menggambarkan tahapan-tahapan teknis mulai dari pengumpulan data hingga evaluasi model. Tahapan awal seperti studi literatur tidak ditampilkan dalam diagram, karena bersifat konseptual dan mendahului proses implementasi.



Gambar 3.1. Alur Penelitian

### 3.3 Pengumpulan dan *Pre-processing* Data

Dataset yang digunakan dalam penelitian ini diperoleh dari *Roboflow Universe*, yang merupakan kumpulan dataset publik untuk *computer vision*. Dataset yang digabungkan terdiri dari tiga proyek, yaitu “*123 Computer Vision Project*” oleh *project-wbfl*, “*Melon Computer Vision Project*” oleh *AI*, dan “*Melon Computer Vision Project*” oleh *Ratchanon*. Ketiga dataset tersebut berisi gambar daun melon dengan berbagai kondisi penyakit yang menjadi fokus penelitian, yaitu *Alternaria*, *CMV*, *Downy Mildew*, *Powdery Mildew*, serta daun yang sehat (*Healthy*).

Setelah penggabungan, dilakukan proses seleksi dan revisi data, termasuk penghapusan beberapa gambar yang kurang relevan seperti foto buah melon, untuk memastikan dataset hanya berisi gambar daun yang sesuai dengan tujuan penelitian.

Proses anotasi awal dari ketiga dataset tidak seragam, di mana sebagian menggunakan *bounding box* yang terlalu luas, sebagian hanya mencakup sebagian daun, dan lainnya menggunakan segmentasi *polygon*. Untuk menjaga konsistensi dan meningkatkan fokus model pada objek utama, seluruh anotasi diseragamkan dengan format segmentasi berupa *smart polygon* yang menandai seluruh area daun secara presisi. Penggunaan anotasi *smart polygon* ini juga bermanfaat di tahap *pre-processing* sehingga dapat menggunakan teknik *isolate object*, sehingga hanya area daun yang telah dianotasi yang akan dipotong dan *background* dihilangkan. Hal ini bertujuan untuk meningkatkan kualitas data *input* dan fokus model pada bagian penting.

#### 3.3.1 Pra-pemrosesan Data

Tahapan pra-pemrosesan dilakukan menggunakan *Roboflow* dengan konfigurasi sebagai berikut:

- ***Preprocessing***:
  - *Auto-Orient: Applied*
  - *Isolate Objects: Applied*
  - *Resize: Stretch to 640×640 piksel*
  
- ***Augmentasi***:
  - *Outputs per training example: 3*

- *Flip: Horizontal, Vertical*
- *Crop: 0% Minimum Zoom, 20% Maximum Zoom*
- *Rotation: -15° hingga +15°*
- *Saturation: -25% hingga +25%*
- *Brightness: -15% hingga +15%*
- *Blur: hingga 0.9 piksel*
- *Noise: hingga 0.1% piksel*

Strategi augmentasi ini diterapkan secara cukup agresif karena jumlah data asli tergolong terbatas, yaitu 1.217 gambar. Untuk meningkatkan keberagaman data dan mengurangi risiko *overfitting*, dilakukan augmentasi hingga tiga kali lipat (maksimum versi per gambar: 3×), sehingga jumlah total data menjadi 3.240 gambar. Beberapa kelas memiliki jumlah sampel yang relatif kecil, sehingga augmentasi ekstrem dianggap perlu untuk membantu model dalam mengenali pola yang bervariasi pada data minoritas.

Setelah proses augmentasi dan anotasi selesai, dataset dibagi ke dalam tiga *subset*: 70% untuk data pelatihan, 15% untuk data validasi, dan 15% untuk data pengujian. Dataset kemudian diekspor dalam format YOLO (*You Only Look Once*) yang kompatibel dengan arsitektur YOLOv8 untuk proses pelatihan selanjutnya.

### 3.4 Pelatihan Model YOLOv8

Pada tahap ini, model YOLOv8 digunakan untuk melakukan deteksi penyakit pada daun tanaman melon. Tiga variasi arsitektur digunakan, yaitu YOLOv8n (*nano*), YOLOv8s (*small*), dan YOLOv8m (*medium*), untuk membandingkan performa antar ukuran model dalam konteks akurasi dan efisiensi pelatihan.

Dataset yang telah melalui proses pra-pemrosesan dibagi menjadi tiga bagian, yaitu 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian.

Pelatihan dilakukan menggunakan pustaka *ultralytics* dengan perangkat *GPU* untuk mempercepat komputasi. Konfigurasi *hyperparameter* utama yang digunakan adalah sebagai berikut:

- **Jumlah epoch:** 300
- **Batch size:** 16

- Nilai *patience*: 50, 75, dan 100 (untuk mengamati pengaruh *early stopping* terhadap hasil model)
- Ukuran gambar input (*imgsz*): 640×640 piksel

Seluruh kombinasi model dan nilai *patience* diuji, menghasilkan total sembilan konfigurasi pelatihan. Evaluasi dilakukan secara otomatis oleh sistem pada setiap *epoch* menggunakan metrik utama seperti *mAP*, *precision*, dan *recall*, yang akan dianalisis pada bab berikutnya.

### 3.5 Evaluasi dan Analisis Model

Evaluasi model dilakukan untuk mengukur kemampuan YOLOv8 dalam mendeteksi dan mengklasifikasikan daun melon yang terinfeksi penyakit *Alternaria*, *CMV*, *Downy Mildew*, *Powdery Mildew*, serta daun sehat (*Healthy*). Evaluasi dilakukan terhadap semua kombinasi konfigurasi model (YOLOv8n, YOLOv8s, YOLOv8m) dan nilai *patience* (50, 75, 100), sehingga diperoleh pemahaman yang komprehensif mengenai pengaruh variasi tersebut terhadap performa sistem.

#### 3.5.1 Metrik Evaluasi

Beberapa metrik digunakan untuk mengevaluasi kinerja model, yaitu:

- **Confusion Matrix**: Menyajikan informasi mengenai jumlah prediksi yang benar dan salah untuk masing-masing kelas. Matriks ini membantu mengidentifikasi kelas mana yang sering tertukar atau sulit dikenali oleh model.
- **Precision, Recall, dan F1-score**:
  - **Precision**: Mengukur proporsi prediksi positif yang benar. Semakin tinggi *precision*, semakin sedikit kesalahan dalam prediksi positif.
  - **Recall**: Mengukur proporsi data positif yang berhasil dikenali oleh model. Semakin tinggi *recall*, semakin baik model mengenali seluruh kasus penyakit.
  - **F1-score**: Merupakan rata-rata harmonik dari *precision* dan *recall*. Cocok untuk menilai performa jika ada ketidakseimbangan jumlah data antar kelas.

- **Mean Average Precision (mAP):** Metrik utama dalam evaluasi *object detection* yang mengukur rata-rata akurasi prediksi model pada berbagai ambang nilai *Intersection over Union* (IoU). Dalam penelitian ini digunakan nilai *mAP@0.5* dan *mAP@0.5:0.95* sebagai acuan utama.

### 3.5.2 Analisis Performa Model

Analisis dilakukan dengan membandingkan hasil evaluasi dari sembilan konfigurasi pelatihan berdasarkan kombinasi model (YOLOv8n, YOLOv8s, dan YOLOv8m) dan nilai *patience* (50, 75, 100). Fokus utama dari analisis adalah:

- **Pengaruh ukuran model** terhadap akurasi dan efisiensi pelatihan.
- **Pengaruh nilai *patience*** terhadap stabilitas pelatihan dan potensi *overfitting/underfitting*.
- **Kinerja per kelas** — untuk mengetahui apakah ada kelas penyakit tertentu yang secara konsisten sulit dikenali oleh model.

### 3.5.3 Benchmark Performa Model

Selain evaluasi akurasi, dilakukan juga pengujian terhadap efisiensi performa model untuk masing-masing konfigurasi menggunakan fungsi `benchmark()` dari pustaka `ultralalytics`. *Benchmark* dilakukan pada perangkat *GPU* menggunakan *backend PyTorch* dan pustaka `torchvision`, dengan tujuan untuk menilai kecepatan *inference, throughput* (FPS), dan efisiensi model secara keseluruhan.

Pengujian dilakukan terhadap file `best.pt` hasil pelatihan dari setiap kombinasi konfigurasi model. Ukuran input (`imgsz`) yang digunakan adalah  $640 \times 640$  piksel, sesuai dengan resolusi standar selama pelatihan.

```

1 from ultralytics.utils.benchmarks import benchmark
2
3 benchmark(model="./runs/detect/train2/weights/best.pt",
4           data='./melonersski-6/data.yaml',
5           imgsz=640,
6           half=False,
7           device='cuda')
```

Kode 3.1: Contoh Sintaks Benchmark Model Menggunakan Ultralytics

*Benchmark* dilakukan hanya menggunakan *backend PyTorch* dan pustaka *torchvision* untuk menjaga konsistensi lingkungan evaluasi di seluruh model yang diuji.

Seluruh hasil evaluasi dan *benchmark* akan dianalisis lebih lanjut pada Bab 4 untuk menentukan konfigurasi model terbaik berdasarkan keseimbangan antara akurasi dan efisiensi.

### **3.6 Batasan Penelitian**

Dalam penelitian ini terdapat beberapa batasan yang perlu diperhatikan, antara lain:

#### **1. Jenis Penyakit**

Penelitian ini hanya memfokuskan pada deteksi empat jenis penyakit tanaman melon, yaitu *Alternaria*, *Cucumber mosaic virus (CMV)*, *Downy mildew*, dan *Powdery mildew*. Penyakit lain yang mungkin juga memengaruhi tanaman melon tidak termasuk dalam cakupan penelitian ini.

#### **2. Kualitas Gambar**

Kualitas dan kondisi gambar input sangat memengaruhi performa model deteksi. Gambar dengan resolusi rendah, pencahayaan buruk, atau noise tinggi dapat menyebabkan penurunan akurasi deteksi.

#### **3. Komposisi Dataset**

Dataset yang digunakan merupakan gabungan dari beberapa dataset publik dengan variasi anotasi awal yang tidak seragam. Meskipun telah dilakukan proses re-annotasi dan seleksi, keberagaman sumber data mungkin masih membawa perbedaan kualitas dan representasi objek.

#### **4. Kondisi Lingkungan Pengambilan Gambar**

Gambar yang digunakan dalam dataset diambil dari berbagai sumber dan kondisi lapangan yang berbeda-beda. Oleh karena itu, model mungkin kurang optimal ketika diaplikasikan pada kondisi nyata yang sangat berbeda dari dataset pelatihan.

#### **5. Model dan Arsitektur**

Penelitian ini hanya membandingkan beberapa varian arsitektur YOLOv8 (yaitu YOLOv8n, YOLOv8s, dan YOLOv8m) tanpa mengeksplorasi model deteksi lain atau teknik peningkatan performa seperti *ensemble learning* atau *fine-tuning* lebih lanjut.

## 6. Label Healthy (Daun Sehat)

Label untuk daun sehat juga dimasukkan, namun variasi kondisi daun sehat yang sangat beragam (misal: variasi warna alami, noda ringan, atau kerusakan mekanis) tidak diteliti secara mendalam dan bisa memengaruhi performa klasifikasi.

### 3.7 Alat dan Bahan

Dalam penelitian ini, perangkat keras dan perangkat lunak yang digunakan harus memenuhi kebutuhan untuk pelatihan dan evaluasi model YOLOv8.

#### 3.7.1 Perangkat Keras

Komputer dengan spesifikasi yang memadai, terutama dilengkapi dengan GPU yang mampu mempercepat proses pelatihan model deteksi objek.

#### 3.7.2 Perangkat Lunak

Berbagai perangkat lunak yang digunakan meliputi:

- Python (versi 3.x)
- Library YOLOv8 dari Ultralytics
- OpenCV untuk pemrosesan citra
- Library pendukung lainnya seperti Torch, torchvision, dan lain-lain.

Dengan metodologi ini, diharapkan penelitian dapat memberikan analisis yang komprehensif mengenai kemampuan berbagai konfigurasi model YOLOv8 dalam mendeteksi penyakit *Alternaria*, *Cucumber mosaic virus* (CMV), *Downy mildew*, dan *Powdery mildew* pada tanaman melon dengan akurasi yang tinggi. Hasil analisis ini diharapkan dapat menjadi referensi untuk pengembangan model deteksi penyakit tanaman yang lebih optimal di masa depan.