

## Deteksi Penyakit Daun Durian dengan Algoritma YOLO (*You Only Look Once*)

Muhammad Ibna Mauladany\*, Bagus Fatkhurrozi, Rheza Ari Wibowo  
Program Studi Teknik Elektro, Universitas Tidar, Indonesia

### Article Info

#### Article history:

Submitted January 16, 2024  
Accepted February 12, 2024  
Published February 16, 2024

#### Keywords:

Deteksi objek,  
penyakit daun durian,  
YOLO

*Object detection,*  
*durian leaf disease,*  
*YOLO*

### ABSTRACT

Permasalahan yang dialami petani durian salah satunya adalah serangan penyakit terhadap daun sehingga mengganggu proses produksi buah. Penyakit yang sering menyerang daun durian adalah bercak daun dan hawar daun. Penelitian ini memiliki tujuan untuk menerapkan teknologi kecerdasan buatan yang dapat membantu mengenali, mengamati serta mendeteksi penyakit daun durian secara efektif. Algoritma deteksi objek menggunakan YOLO (*You Only Look Once*) merupakan bagian dari sistem kecerdasan buatan digunakan dalam penelitian ini. Objek yang dideteksi dalam penelitian ini dibagi menjadi 3 kelas yaitu bercak daun, hawar daun, dan daun sehat. Proses penyusunan sistem memanfaatkan citra daun yang memiliki kelas tersebut dengan jumlah 300 gambar dan 25 gambar sebagai citra uji. Dari hasil *training dataset* menggunakan *Google Colab*, nilai mAP tertinggi didapat pada *epoch* 100 yaitu sebesar 0,815. Model kemudian diuji dan mendapatkan nilai akurasi 85 %, kepresisian 96 %, dan *recall* 86%.

*One of the problems experienced by durian farmers is disease attacks on the leaves which disrupt the fruit production process. Diseases that often attack durian leaves are leaf spot and leaf blight. This research aims to apply artificial intelligence technology that can help recognize, observe and detect durian leaf disease effectively. The object detection algorithm using YOLO (you only look once) is part of the artificial intelligence system used in this research. The objects detected in this research were divided into 3 classes, namely leaf spots, leaf blight, and healthy leaves. The process of compiling the system utilizes leaf images that belong to this class with a total of 300 images and 25 images as test images. From the results of the training dataset using Google Colab, the highest mAP value was obtained at epoch 100, namely 0.815. The model was then tested and obtained an accuracy value of 85%, precision 96%, and recall 86%.*



### Corresponding Author:

Muhammad Ibna Mauladany,  
Program Studi Teknik Elektro, Universitas Tidar, Indonesia,  
Jl. Kapten Suparman No. 39, Potrobangsari, Kota Magelang, Indonesia.  
Email: \*ibnamauladany@gmail.com

## 1. PENDAHULUAN

Indonesia merupakan negara yang memiliki iklim tropis dengan dua musim yaitu kemarau dan hujan. Beragam jenis tanaman dikembangkan oleh masyarakat dengan memanfaatkan iklim tersebut. Salah satu hasil pertanian yang memiliki nilai ekonomis di Indonesia adalah buah durian [1]. Permasalahan yang dihadapi petani durian adalah serangan penyakit terhadap pohon sehingga mengganggu proses pertumbuhan dan produksi buah.

Langkah awal yang bisa dilakukan untuk penyelesaian masalah tersebut adalah melakukan identifikasi terhadap jenis penyakit tanaman. Beberapa penyakit yang sering menyerang durian adalah bercak daun, hawar daun, jamur upas, kanker batang, busuk akar dan busuk buah [2]. Penyakit tersebut akan mengakibatkan penurunan kualitas buah jika tidak ditangani dengan tepat. Pemanfaatan teknologi kecerdasan buatan, seperti deteksi objek menggunakan algoritma seperti YOLO (*You Only Look Once*) akan membuat petani secara efisien dan akurat dalam mengidentifikasi penyakit pada pohon durian. Teknologi ini memungkinkan petani untuk mendeteksi dini adanya penyakit pada tanaman, sehingga tindakan pencegahan dan pengendalian dapat dilakukan dengan tepat waktu. Hal tersebut akan mengurangi kerugian yang disebabkan oleh serangan penyakit.

Penelitian yang dilakukan oleh Silvi, dkk. menghasilkan sistem pakar tentang penyakit pada tanaman durian [3]. Sistem ini masih memiliki kelemahan dibanding deteksi objek yang menggunakan *input* berupa

gambar. Penelitian tentang deteksi tanaman tebu pada lahan pertanian menggunakan metode CNN dilakukan oleh Jilmy menghasilkan akurasi sebesar 95% [4]. Penelitian lainnya tentang deteksi objek dilakukan oleh Fangtao untuk mengidentifikasi stomata pada daun jagung menggunakan CNN. Hasil yang didapatkan mencapai tingkat presisi sebesar 94,8% dan *recall* sebesar 98,7% dalam deteksi stomata pada *dataset* daun jagung [5].

Muhammad Fathul Alim Maulana meneliti algoritma CNN untuk identifikasi penyakit pada tanaman Tomat. Penyakit yang dideteksi pada tanaman Tomat hanya diambil dari citra daun. Penelitian tersebut menggunakan *dataset* sebanyak 22.930 gambar yang dibagi menjadi 10 kelas. CNN dilatih untuk mendeteksi 9 jenis penyakit pada daun tomat. Dari enam model CNN yang digunakan, hasil terbaik dicapai oleh model ResNet-50 yaitu presisi 97%, akurasi 96,16%, dan *recall* 96% [6].

Monalika Padma Reddy dalam penelitiannya melakukan deteksi penyakit pada daun Murbei secara *realtime*. *Dataset* yang digunakan berjumlah 1000 gambar dan proses *training* dilakukan sebanyak 1000 *epoch*. Algoritma YoloV4 dalam penelitian ini menghasilkan kecepatan deteksi sebanyak 45 *frame* per detik dan akurasi yang cukup tinggi [7].

Berdasarkan beberapa permasalahan yang ada maka penulis mengembangkan penelitian untuk mendeteksi objek pada penyakit daun durian dengan judul “Deteksi Penyakit Daun Durian Dengan Algoritma Yolo (*You Only Look Once*)”. Pada penelitian ini penulis menggunakan YOLOv5 sebagai algoritma *objek detection* dan membagi kelas menjadi 3 yaitu daun sehat, bercak daun, dan hawar daun dengan jumlah *dataset* 300 gambar. Dengan adanya penelitian ini diharapkan mampu menjadi pengembangan teknologi di bidang pertanian.

## 2. METODE PENELITIAN

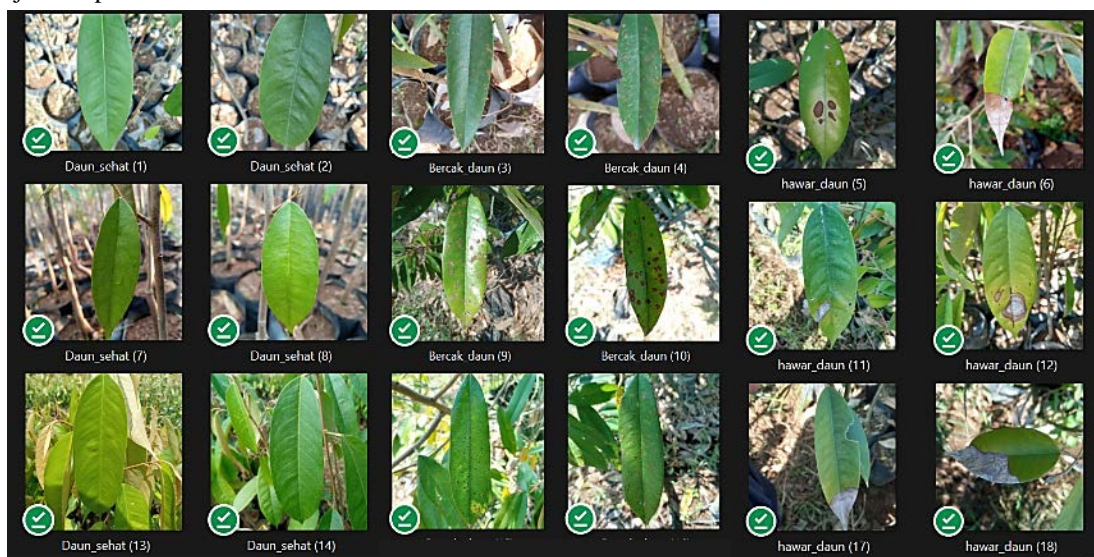
Penelitian ini dimulai dengan melakukan studi literatur terlebih dahulu. Tahap ini dilakukan untuk mencari referensi penelitian sekaligus memastikan bahwa penelitian yang akan dilakukan belum pernah dilakukan sebelumnya. Tahap selanjutnya adalah perancangan sistem deteksi tersebut meliputi perancangan diagram blok sistem dan penentuan komponen yang akan digunakan. Penulis menggunakan metode *Convolutional Neural Network* yang terdapat pada algoritma *You Only Look Once* (YOLO). Metode ini dipilih karena proses pendeteksian akan lebih cepat dan akurasi yang cukup tinggi [8].

### 2.1 Studi Literatur

Studi literatur merupakan serangkaian kegiatan yang berkenaan dengan metode pengumpulan data, membaca, mencatat dan mengolah bahan penelitian dari berbagai sumber. Adapun sumber yang dijadikan referensi berupa jurnal-jurnal terdahulu, buku, serta artikel lainnya yang didapat dari internet.

### 2.2 Pengumpulan Dataset

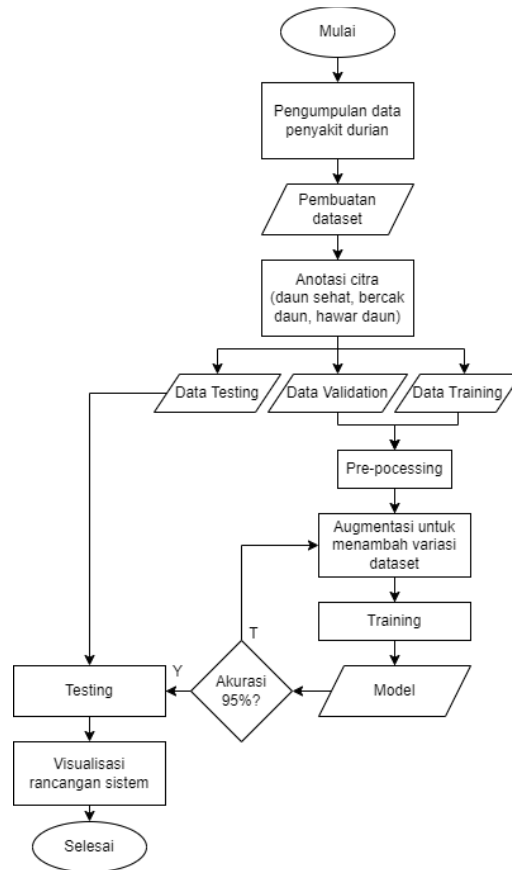
*Dataset* diperoleh dari petani bibit durian dan sumber lainnya dari internet. Gambar yang dibutuhkan dalam penelitian ini dibagi menjadi 3 jenis yaitu daun sehat, bercak daun dan hawar daun dengan jumlah masing-masing 100. Jumlah keseluruhan citra yang digunakan adalah 300 dengan format *file* jpg. Kumpulan citra ditunjukkan pada Gambar 1.



Gambar 1. *Dataset* daun durian

### 2.3 Perancangan Sistem

Proses perancangan dilakukan untuk membangun sistem agar bisa bekerja secara optimal. Tahapan dalam perancangan sistem bisa dilihat dari Gambar 2.



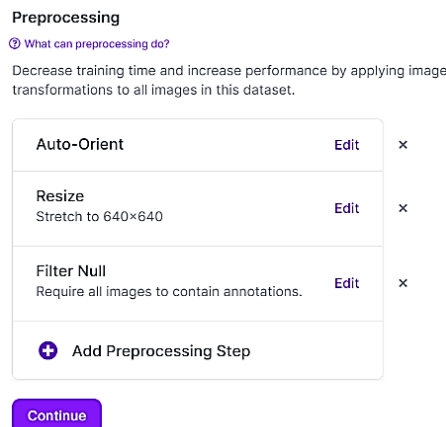
Gambar 2. Diagram alir perancangan sistem

### 2.3.1 Anotasi Data

Pada tahap ini, *dataset* akan diberikan label yang berisi informasi nama kelas, titik koordinat, serta ukuran objek. Proses ini dilakukan dengan memanfaatkan fitur yang ada pada *Roboflow*. Citra yang telah dilakukan anotasi kemudian dibagi menjadi 3 yaitu citra *training*, citra *validation set*, dan citra *test*. Pembagian ini dibuat dengan perbandingan 70%, 20%, 10% sehingga total datanya menjadi 210 untuk *train set*, 60 untuk *validation set*, dan 30 untuk *test set*.

### 2.3.2 Preprocessing

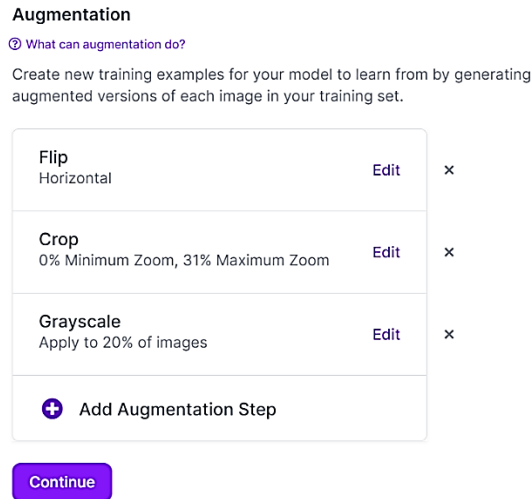
*Preprocessing dataset* di *Roboflow* bertujuan untuk mempersiapkan dan meningkatkan kualitas *dataset* sebelum dilakukan pelatihan model deteksi objek. *Auto-Orient* berguna untuk menghilangkan informasi EXIF dari data gambar, sehingga tampilan gambar sesuai dengan penyimpanan pada *disk*. Dimensi data disesuaikan menjadi 640 x 640 piksel, mengacu pada model YOLOv5 yang telah dilatih sebelumnya berukuran 640 piksel. *Filter null* diterapkan untuk menghapus data yang telah dicatat sebagai *null* atau gambar tanpa objek, sehingga data tersebut tidak akan dipertimbangkan dalam proses pelatihan model. Pra-pemrosesan dilakukan di *roboflow* sekaligus proses augmentasinya. Gambar 3 menunjukkan rincian *preprocessing*.



Gambar 3. *Preprocessing dataset*

### 2.3.3 Augmentasi

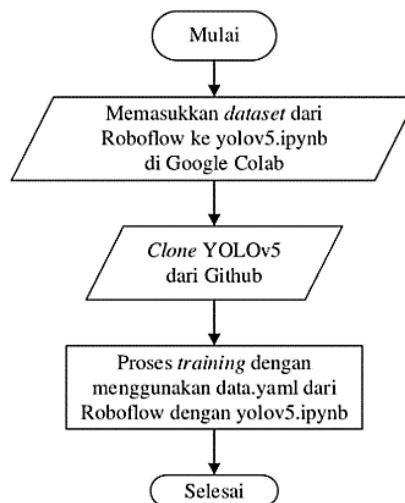
Augmentasi data digunakan untuk memperkaya *dataset* serta mengatasi permasalahan *overfitting* pada model. Tahap ini bertujuan menambah akurasi inferensi karena *dataset* yang dihasilkan semakin banyak dan bervariasi [9]. Teknik augmentasi yang digunakan adalah *horizontal flip*, *cropping* dan *grayscale*. Proses augmentasi ditunjukkan pada Gambar 4.



Gambar 4. Proses augmentasi

### 2.3.4 Modelling

Tahapan *modelling* dilakukan melalui *training dataset* yang sebelumnya telah diolah pada *roboflow*. Fitur ekspor *dataset* pada *roboflow* menghasilkan file bernama “data.yaml” dengan format yang didukung YOLOv5. File tersebut yang akan digunakan untuk proses *training* melalui platform *Google Colab*. Flowchart pada Gambar 5 mengilustrasikan proses *training*.



Gambar 5. Flowchart proses training

Proses *training dataset* dapat dijalankan dengan mengatur konfigurasi *image size*, *batch size*, *epoch*, *data source*, dan *weight*. Konfigurasi yang digunakan dalam proses *training* dapat dilihat pada Tabel 1.

Tabel 1. Konfigurasi Training

Parameter	Nilai
<i>Image Size</i>	640
<i>Batch Size</i>	10
<i>Epoch</i>	50, 100, 150, 200
<i>Data Source</i>	data.yaml
<i>Weight</i>	YOLOv5s.pt

### 2.3.5 Evaluasi

Evaluasi merupakan tahap penting yang harus dilakukan setelah proses *training*. Dalam melakukan evaluasi, performa model yang telah dilatih perlu dianalisis menggunakan metrik evaluasi berupa *confusion matrix* sehingga dapat dihitung nilai *precision*, *recall*, mAP, dan akurasi [10]. *Precision* melihat seberapa banyak yang diidentifikasi model benar, *recall* memberi gambaran nilai dari banyaknya objek yang seharusnya terdeteksi berhasil ditemukan, sementara akurasi memberi gambaran keseluruhan keberhasilan model [11]. Pada *Google Colab* terdapat *TensorBoard* yang dapat diakses langsung dan memiliki fitur untuk menampilkan hasil *training* sehingga proses evaluasi menjadi lebih mudah.

### 2.4 Pengujian

Pada tahap ini, dilakukan pengujian pada data testing menggunakan model deteksi objek yang telah dilatih, menghasilkan nilai mAP tertinggi. Pengujian ini melibatkan parameter evaluasi, termasuk *Precision*, *Recall*, dan Akurasi. *Precision* mengukur seberapa banyak objek yang diidentifikasi oleh model benar-benar relevan dapat dihitung dengan Persamaan (1), *Recall* mengukur seberapa banyak objek yang seharusnya diidentifikasi oleh model dapat diidentifikasi menggunakan Persamaan (2), dan Akurasi mencerminkan sejauh mana model dapat memberikan hasil yang benar secara keseluruhan dapat dihitung dengan Persamaan (3).

$$Precision = \frac{TP}{TP+TN} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Accuracy = \frac{TP+TN}{P+N} \tag{3}$$

### 2.5 Analisis dan Pembahasan

Pada tahap ini, dilakukan analisis terhadap hasil uji yang telah diperoleh. Proses ini bertujuan untuk mengidentifikasi faktor-faktor yang berkontribusi terhadap hasil uji yang telah dilakukan pada sub-bab 2.4. Analisis ini penting untuk memahami variabel-variabel yang mungkin memengaruhi performa atau *output* dari uji yang telah dilaksanakan.

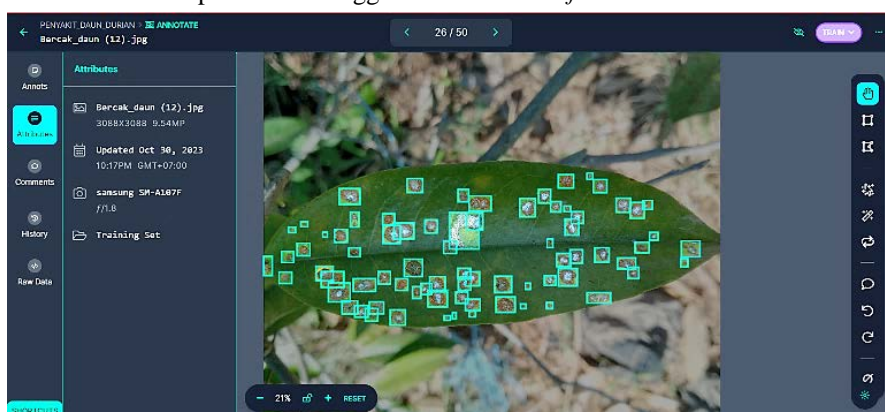
Dengan menganalisis faktor-faktor yang mempengaruhi hasil uji, peneliti dapat mendapatkan wawasan lebih mendalam tentang elemen-elemen yang perlu diperhatikan atau diperbaiki dalam pengaturan eksperimen atau implementasi model. Hasil analisis ini dapat menjadi dasar untuk pengambilan keputusan lebih lanjut, seperti perubahan parameter, penyesuaian teknis, atau strategi perbaikan lainnya untuk meningkatkan validitas dan reliabilitas hasil uji [12].

## 3. HASIL DAN PEMBAHASAN

Hasil yang diperoleh dari pengujian sistem deteksi penyakit daun durian dalam penelitian ini kemudian dibahas dan dapat diketahui tingkat keberhasilan sistem yang dibuat. Sistem deteksi ini disusun berdasarkan tahapan dalam metode penelitian.

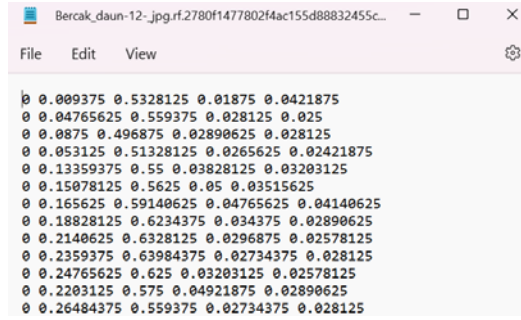
### 3.1 Hasil Anotasi Data

Pelabelan terdiri dari tiga nama kelas dengan masing-masing kelas berjumlah 100 gambar, sehingga jumlah total sebanyak 300 gambar dengan kelas daun sehat, bercak daun, dan hawar daun. Gambar 6 menunjukkan contoh citra hasil pelabelan menggunakan situs *roboflow*.



Gambar 6. Proses anotasi data

Setiap objek yang memiliki *bounding box* akan direpresentasikan dalam satu baris dengan lima kolom nilai. Kolom pertama menyatakan kelas objek, diikuti oleh koordinat x, koordinat y, lebar kotak, dan tinggi kotak *bounding box*. Kategori objek dinyatakan dalam bentuk kelas, di mana kelas 0 mencakup bercak daun, kelas 1 mencakup daun yang sehat, dan kelas 3 mencakup hawar daun. Gambar 7 menunjukkan *file* hasil anotasi citra dengan format YOLO TXT.

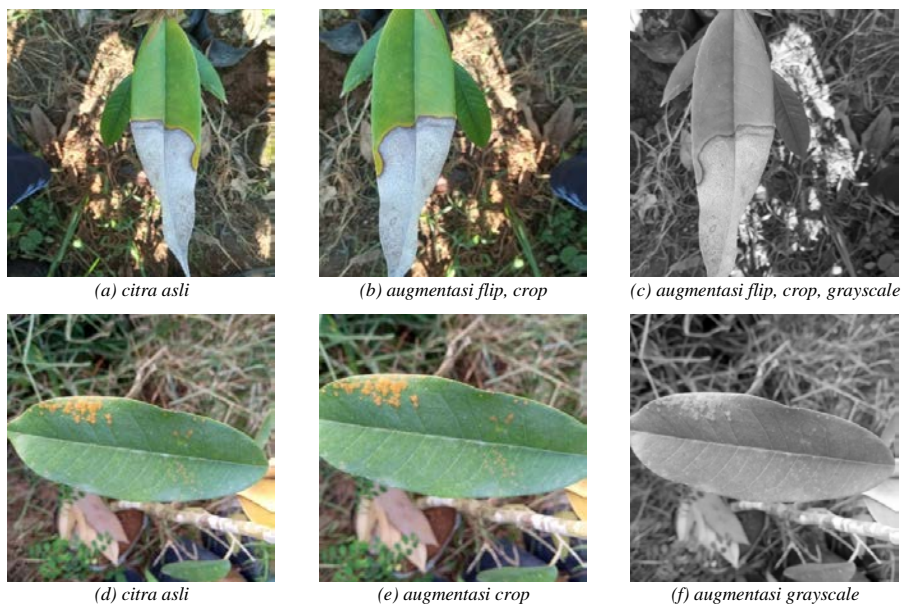


Gambar 7. File label dengan format YOLO TXT

### 3.2 Hasil Preprocessing dan Augmentasi

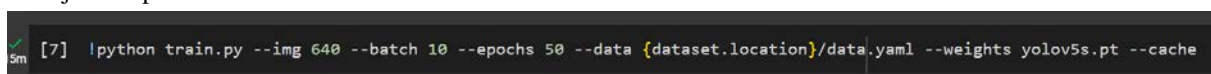
Pra-pemrosesan dilakukan dengan menyamakan ukuran citra menjadi 640 piksel mengacu pada model YOLOv5. Selain itu, *Auto-Orient* digunakan untuk menghilangkan informasi EXIF dari data gambar, sehingga tampilan gambar sesuai dengan penyimpanan pada *disk*. Pra-pemrosesan juga menggunakan fitur *filter null* agar data yang tidak diberi label terhapus, sehingga kualitas *dataset* semakin baik. Pra-pemrosesan diterapkan pada seluruh data yang berjumlah 300.

Proses *augmentasi* yang digunakan adalah variasi *horizontal-flip*, *grayscale*, dan *cropping*. *Flip* berfungsi agar mesin dapat membaca citra daun yang posisinya beragam. *Grayscale* akan menambah variasi *dataset* dengan mengubah citra daun menjadi warna abu-abu dengan tingkat keabuan 20%. Augmentasi selanjutnya dengan cara *cropping* atau memotong gambar agar mesin dapat mendeteksi variasi daun tidak terlihat secara penuh. Augmentasi hanya diterapkan pada data *train*, sehingga jumlah data *train* yang semula 210 bertambah menjadi 630 citra. Gambar 8 menunjukkan perbandingan *dataset* asli dan *dataset* hasil augmentasi.

Gambar 8. Perbandingan *dataset* asli dan *dataset* hasil augmentasi

### 3.3 Training Dataset

Proses pelatihan atau *training dataset* menggunakan *google colab* sebagai *jupyter notebook*. *Training* memerlukan *setup* beberapa parameter, di antaranya adalah mengatur nilai *batch* dan *epoch*. *Batch* adalah jumlah sampel yang diproses dalam satu kelompok, sedangkan *epoch* adalah jumlah pengulangan pada proses *training*. Pada penelitian proses *training* dilakukan 4 kali dengan variasi seperti Tabel 1. Pengaturan variasi *epoch* ditunjukkan pada Gambar 9.

Gambar 9. Konfigurasi parameter *training*

### 3.4 Evaluasi

*Dataset* yang telah dilatih sebanyak 4 kali dengan *epoch* 50, 100, 150, dan 200 kemudian dilakukan evaluasi. Proses evaluasi dilakukan dengan mengambil nilai *precision*, *recall*, *F1-score*, dan mAP. Perbandingan hasil *training* tersebut ditunjukkan pada Tabel 2.

Tabel 2. Perbandingan hasil *training*

<i>Epoch</i>	<i>Precision</i>	<i>Recall</i>	<i>mAP</i>	<i>F1-Score</i>
50	0.77	0.799	0.809	0.784
100	0.805	0.779	0.815	0.792
150	0.786	0.778	0.803	0.782
200	0.751	0.809	0.801	0.779

Hasil *training* yang terbaik didapatkan pada *training* dengan *epoch* 100 dilihat dari mAP yang dihasilkan sebesar 0.815, *precision* sebesar 0.77, *recall* sebesar 0.799, dan F1-score sebesar 0.784.

### 3.5 Hasil Pengujian Model

Pengujian dilakukan menggunakan 25 citra uji yang terdiri dari 10 citra daun durian sehat, 5 citra bercak daun, dan 10 citra hawar daun. Proses pengujian memerlukan *file weight* bernama “best.pt” yang diunduh dari *google colab* pada program *training*. File tersebut disimpan pada folder YOLOv5. Langkah selanjutnya adalah menjalankan perintah “detect.py” pada YOLOv5 melalui Visual Studio Code, namun sebelumnya perlu dilakukan konfigurasi parameter deteksi seperti *weight* dan lokasi *file* yang akan dideteksi. Gambar 10 menunjukkan parameter yang telah disesuaikan.

```
(yoloo) PS E:\Belajaryolo\Latihan1\yolov5> python detect.py --weights weight\best100.pt --source data\images\test2 --line-thickness 4 --save-txt --save-csv --save-conf --save-crop
detect: weights=['weight\best100.pt'], source=data\images\test2, data=data\coco128.yaml, imgsz=[640, 640], conf_thres=0.25, iou_thres=0.45, max_det=1000, device=, view_img=False, save_txt=True, save_csv=True, save_conf=True, save_crop=True, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=runs\detect, name=exp, exist_ok=False, line_thickness=4, hide_labels=False, hide_conf=False, half=False, dnn=False, vid_stride=1
```

Gambar 10. Konfigurasi pengujian model

Dari 25 citra yang dideteksi, didapatkan 258 objek bercak daun, 41 objek hawar daun, dan 10 objek daun sehat. Beberapa citra hasil pengujian dapat dilihat pada Gambar 11.



Gambar 11. Citra hasil pengujian

### 3.6 Analisis Hasil Pengujian

Berdasarkan proses pengujian menggunakan 25 citra didapatkan 10 objek daun sehat yang sesuai dengan jumlah aktual. Pada kelas bercak daun, jumlah aktual adalah 248 namun sistem mendeteksi bercak daun sejumlah 258. Kesalahan deteksi pada kelas bercak daun dibagi menjadi *False Positive*, dan *False Negative*. Objek bercak daun yang gagal terdeteksi sejumlah 12 merupakan *False Negative*, sedangkan pada *False Positive* terdapat 22 objek bukan bercak daun yang dideteksi sebagai bercak daun.

Pengujian pada kelas selanjutnya menghasilkan 41 objek terdeteksi hawar daun, sedangkan jumlah aktual kelas tersebut adalah 57. *False Positive* yang dihasilkan berjumlah 3 dan 19 objek dikategorikan sebagai *false negative*. Sejumlah *error* pada pengujian tersebut disebabkan *dataset* yang digunakan masih sedikit dan tidak *background* yang digunakan tidak variatif. Hasil pengujian dapat dilihat pada *confusion matrix* Tabel 3.

Tabel 3. *Confusion Matrix* hasil pengujian

	Nilai Sebenarnya	Daun sehat	Bercak Daun	Hawar Daun	<i>Background</i>
Nilai Prediksi	Daun Sehat	10	0	0	0
	Bercak Daun	0	236	7	15
	Hawar Daun	0	0	38	3
	<i>Background</i>	0	12	12	0

Berdasarkan *confusion matrix* tersebut, nilai akurasi dihitung dapat dihitung menggunakan dengan Persamaan (3), nilai *precision* menggunakan Persamaan (1), dan nilai *recall* menggunakan Persamaan (2).

$$\text{Akurasi} = \frac{10+236+38}{10+236+38+7+15+3+12+12} = 0,852$$

$$\text{Recall daun sehat} = \frac{10}{10} = 1$$

$$\text{Recall bercak daun} = \frac{236}{236+7+15} = 0,915$$

$$\text{Recall hawar daun} = \frac{38}{38+3} = 0,927$$

$$\text{Precision daun sehat} = \frac{10}{10} = 1$$

$$\text{Precision bercak daun} = \frac{236}{236+12} = 0.952$$

$$\text{Precision hawar daun} = \frac{38}{38+12+7} = 0.666$$

Dari *confusion matrix* tersebut dapat dihitung nilai akurasi, *precision*, *recall*, dan *error*. Nilai akurasi yang didapat adalah 85%, rata-rata *precision* 96%, dan rata-rata *recall* 86%. Berdasarkan diagram alir sistem, nilai yang dihasilkan belum mencapai target yaitu akurasi >95%. Nilai yang belum maksimal tersebut disebabkan oleh keterbatasan *dataset* yang tidak memiliki variasi *background*.

#### 4. KESIMPULAN

Berdasarkan pembahasan dalam penelitian tentang deteksi penyakit daun durian dengan YOLO, didapatkan kesimpulan bahwa performa model yang dibangun menggunakan algoritma YOLOv5 mendapat hasil yang baik. Model dibangun menggunakan 300 gambar sebagai *dataset* dan dilatih beberapa kali menggunakan *epoch* 50, 100, 150 dan 200. Performa model terbaik didapatkan pada *epoch* 100 dengan mAP (*mean Average Precision*) sebesar 0,815. Model terbaik tersebut digunakan untuk pengujian menggunakan 25 citra uji. Hasil pengujian mendapatkan nilai akurasi 85%, *precision* 96%, dan *recall* 86%. Keterbatasan *dataset* menyebabkan hasil penelitian belum mendapatkan nilai yang maksimal. Untuk meningkatkan nilai tersebut, penulis memberikan saran dengan menambah *dataset* dengan *background* yang bervariasi sehingga sistem dapat membaca dan mempelajari jenis penyakit daun durian dengan lebih baik.

#### REFERENSI

- [1] A. Furqon, N. I. Fadlilah, "Sistem Pakar Untuk Mendiagnosa Hama Dan Penyakit Pada Pohon Durian," *Konferensi Nasional Ilmu Sosial & Teknologi (KNiST)*, vol. 1, no. 1, hal. 105-109, 2016.
- [2] Napitupulu, Rodame Monitorir, "Bertanam Durian Unggul," Penebar Swadaya, Jakarta, 2015.
- [3] S. Indah Amelia, S. Eko, S. Achmad, "Sistem Pakar Diagnosis Hama dan Penyakit Pada Pohon Buah Durian Montong Menggunakan Metode Forward Chaining Dengan Php Native," *Jurnal Sistem Informasi dan Teknologi Peradaban (JSITP)*, vol. 1, no. 1, hal. 6-22, 2020.
- [4] Muhammad Alfin Jimly Asshiddiqie, Basuki Rahmat, Fetty Tri Anggraeny, "Deteksi Tanaman Tebu pada Lahan Pertanian Menggunakan Metode Convolutional Neural Network," *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 1, no. 1, hal. 229, 2020.
- [5] Ren, Fangtao, Zhang, Yawei-Liu, Xi-Zhang, Yingqi-Liu, Ying-Zhang, Fan, "Identification of Plant Stomata Based on YOLO v5 Deep Learning Model," *CSAI '21: Proceedings of the 2021 5th International Conference on Computer Science and Artificial Intelligence*, hal. 78-83, 2021. <https://doi.org/10.1145/3507548.3507560>
- [6] Alim, Maulana Muhammad Fathul, "Identifikasi Penyakit Tanaman Tomat Menggunakan Algoritma Convolutional Neural Network dan Pendekatan Transfer Learning," Universitas Negeri Semarang, Semarang, 2020.
- [7] Reddy, Monalika Padma, Deeksha, "Mulberry leaf disease detection using YOLO," *International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT)*, vol. 7, no. 3, hal. 1816, 2021.
- [8] Ardieka Fahmi Radhitya, Abdal Jabar, Muhammad Kemal Qodrat, Hasan Maulana, "Perbandingan Sistem Pendeteksian Kendaraan: Faster RCNN dengan YOLOV5 untuk Keselamatan Lalu Lintas," *Jurnal AI dan SPK: Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, vol. 1, no. 1, hal. 100-104, 2023.
- [9] Joseph Sanjaya, Mewati Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *JuTISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 6, no. 2, hal. 313, 2020. <https://doi.org/10.28932/jutisi.v6i2.2688>



- [10] Muhammad Nur Ihsan Muhlashin, Arnisa Stefanie, "Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8," *Jurnal Mahasiswa Teknik Informatika (JATI)*, vol. 7, no. 2, hal. 1363-1367, 2023. <https://doi.org/10.36040/jati.v7i2.6927>
- [11] Tomy Abuzairi, Nurdina Widanti, Arie Kusumaningrum, Yeni Rustina, "Implementasi Convolutional Neural Network Untuk Deteksi Nyeri Bayi Melalui Citra Wajah Dengan YOLO," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, hal. 624-630, 2021. <https://doi.org/10.29207/resti.v5i4.3184>
- [12] Annisa Auliya, Tresna Dewi, Yurni Oktarina, Mohammad Nawawi Noer, "Implementasi Pengolahan Citra Menggunakan Metode YOLO pada Security Robot di Bidang Pertanian," *Journal of Applied Smart Electrical Network and System (JASENS)*, vol. 3, no. 2, hal. 42-46, 2022. <https://doi.org/10.52158/jasens.v3i02.508>