

**IDENTIFIKASI HAMA DAN PENYAKIT PADA TANAMAN JATI
DENGAN METODE DEEP LEARNING YOLOV8**

Muna Sintya Elyza Putri

Universitas Muhammadiyah Ponorogo

E-mail: munasintyaep@gmail.com

Abstrak

Tanaman jati (*Tectona grandis*) adalah tanaman dengan kualitas kayu yang terkenal bermutu tinggi dan banyak dibudidayakan di Indonesia. Tanaman ini dikenal karena ketahanan-nya terhadap cuaca ekstrem, tetapi seringkali rentan terhadap hama dan penyakit yang merugikan produksi kayu berkualitas. Kemajuan teknologi di bidang agriculture, khususnya teknologi berbasis kecerdasan buatan, telah membantu para pembudidaya untuk mengatasi masalah dibidang pertanian. Penelitian ini bertujuan menerapkan teknologi kecerdasan buatan berupa object detection untuk mengenali dan mendeteksi penyakit serta hama pada tanaman jati secara efektif menggunakan metode deep learning, yaitu algoritma YOLOv8 (You Only Look Once). Objek yang dideteksi dalam penelitian ini dibagi menjadi sembilan kelas hama dan penyakit. Sistem dikembangkan menggunakan citra gambar daun dan batang jati dengan total 435 dataset gambar, yang kemudian dibagi menjadi 70% data latih, 20% data validasi, dan 10% data uji, sehingga diperoleh 381 data latih, 36 data validasi, dan 18 data uji. Pelatihan dataset dilakukan menggunakan Roboflow dan Google Colab. Model dilatih sebanyak 5 kali dengan menggunakan epoch 25, 10, 30, 100, 150 betch 16, telah menghasilkan model terbaik pada epoch 100 dengan nilai precision yang menghasilkan nilai sebesar 0.979, mAp 0.728, recall sebesar 0.264 dan F1-Score sebesar 0.628. Dari hasil training model yang sudah dilakukan pengujian pada model terbaik telah menghasilkan dataset dengan tingkat akurasi sebesar 87.6%, validation sebesar 8.3%, test 4.1%.

Kata Kunci — Objek deteksi, YOLO, Deep learning, Hama dan Penyakit.

Abstract

*The teak plant (*Tectona grandis*) is a plant with wood that is known for its high quality and is widely cultivated in Indonesia. This plant is known for its resistance to extreme weather, but is often susceptible to pests and diseases that are detrimental to quality wood production. Technological advances in the agricultural sector, especially artificial intelligence-based technology, have helped cultivators to overcome problems in the agricultural sector. This research aims to apply artificial intelligence technology in the form of object detection to recognize and detect diseases and pests in teak plants effectively using deep learning methods, namely the YOLOv8 (You Only Look Once) algorithm. The objects detected in this study were divided into nine classes of pests and diseases. The system was developed using images of teak leaves and stems with a total of 435 image datasets, which were then divided into 70% training data, 20% validation data, and 10% test data, resulting in 381 training data, 36 validation data, and 18 test data. Dataset training was carried out using Roboflow and Google Colab. The model was trained 5 times using epochs 25, 10, 30, 100, 150 bet 16, resulting in the best model at epoch 100 with a precision value of 0.979, mAp 0.328, recall of 0.264 and F1-Score of 0.428. From the results of the training model that has been tested on the best model, it has produced a dataset with an accuracy level of 87.6%, validation of 8.3%, test of 4.1%.*

Keywords — Object detection, YOLO, Deep learning, Pests and Diseases.

1. PENDAHULUAN

Pohon Jati (*Tectona grandis*) adalah jenis tanaman yang dikembangkan di Pulau Jawa kurang lebih seratus tahun yang lalu, Pohon jati adalah jenis tanaman yang memiliki kualitas batang kayu yang tinggi yang sangat terkenal dengan kualitas kayu yang kuat dan awet, sehingga sering digunakan untuk membuat furnitur atau konstruksi. Seiring perjalanan waktu kebutuhan manusia akan bahan baku kayu berkualitas tinggi seperti Pohon jati selalu meningkat dengan daur produksi dari tanaman ini yang lama setelah umur 60 tahun, sedangkan produksi optimum yang di butuhkan pada umur 80 tahun yang berakibat ketersediaan Jati mulai terbatas dalam produksinya [1].

Dalam perkembangan budidaya tanaman jati sering kali terdapat hambatan dalam proses pertumbuhannya, meskipun tanaman jati identik dengan tanaman yang memiliki ketahanan tinggi dalam beradaptasi terhadap serangan penyakit. Organisme pengganggu tanaman seperti hama dan penyakit dapat mengganggu ketersediaan bahan baku produksi [2]. Kerugian yang ditimbulkan oleh Organisme pengganggu tanaman menimbulkan dampak besar bagi budidaya tanaman jati, beberapa penyakit yang sering timbul pada tanaman jati yaitu penyakit batang *Corticium salmonicolor*, Penyakit pucuk daun *Phomopsi tectonae*, penyakit akar *Armillaria melea* [3]. Umumnya para pembudidaya tanaman jati dalam mendiagnosis penyakit secara manual, deteksi manual dapat memicu kesalahan dalam identifikasi hama dan penyakit yang berakibat pada kesalahan dalam penanganan atau pun pencegahan, selain deteksi manual menggunakan jasa konsultasi ilmuwan tanaman juga di lakukan tetapi dalam prosesnya membutuhkan biaya sangat mahal dalam operasionalnya.

Dengan kemajuan teknologi informasi bidang agriculture mengalami banyak inovasi terutama dalam kecerdasan buatan yang membuat model otomasi untuk mendiagnosis dengan presisi tinggi [4]. Deep learning merupakan teknik moderen baru pemrosesan gambar dan analisis data yang merupakan bagian dari machine learning. Machine learning adalah program komputer yang mampu belajar dari pengalaman, sehingga semakin banyak pengalaman yang diberikan maka performanya menjadi semakin baik. Dalam praktiknya deep learning dibuat untuk menyelesaikan masalah performa pada algoritma machine learning tradisional saat pertumbuhan data yang membesar [5].

Pemantauan pertumbuhan tanaman yang cermat dan baik dapat meningkatkan kualitas hasil tani yang berkualitas tinggi. Oleh karenanya perlunya mengidentifikasi pertumbuhan hama dan penyakit untuk tindakan pencegahan dan meminimalisir kerugian hasil tani. Proses identifikasi jenis hama dan penyakit pada tanaman jati dapat dilakukan dengan pengenalan citra dari jenis hama dan penyakit itu sendiri [6]. Dibutuhkan teknologi deteksi objek dan pengolahan citra yang efektif serta real-time dalam penerapannya, teknologi YOLOv8 (You Only Look Once version 8) adalah algoritma yang dapat mendeteksi objek secara real-time dengan peningkatan akurasi dan presisi [7]. Algoritma ini digunakan berdasarkan pendekatan deteksi objek berdasarkan Convolutional Neural Network (CNN) yang dapat mempelajari pola fitur data citra [8].

Penelitian dengan judul “Deteksi Penyakit dan Hama pada pohon Jati Dengan Metode Deep Learning Yolov8”. Bertujuan untuk deteksi jenis hama dan penyakit pada tanaman yang menerapkan metode deep learning pada pohon jati dengan algoritma YOLOv8 sebagai objek detection dengan membagi menjadi 9 kelas jenis penyakit dan hama yang sering menyerang tanaman Pohon jati, jumlah dataset yang digunakan sebanyak 435 gambar, penelitian ini diharapkan dapat mengembangkan model deteksi objek dengan segmentasi efektif dan akurat pada dataset gambar.

2. TINJAUAN PUSTAKA

Pelaksanaan penelitian ini mengacu dari kajian sebelumnya yang telah menghasilkan riset sebagai bahan penelitian saat ini, langkah ini dimaksudkan sebagai kontribusi substantif dari pengembangan lebih lanjut dari tema penelitian. Berikut adalah ringkasan mengenai penelitian sebelumnya yang sudah dilaksanakan yang mencakup pendekatan metodologi serta hasil yang diperoleh dari penelitian.

Penelitian yang berkaitan dengan objek deteksi menggunakan metode deep learning terdapat pada. Yasen dkk (2023) yang berjudul “Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode Deep Learning”. Dalam penelitian ini Tanaman cabai sering kali mengalami kegagalan panen akibat serangan hama dan penyakit yang berakibat pada kerugian. Kurangnya kesadaran petani dalam mengetahui jenis penyakit berakibat pada terlambat-nya diagnosis penyakit pada tanaman. Pengujian dilakukan dengan aplikasi berbasis web dibuat dengan framework flask. Hasil akurasi dari proses training model YOLO dengan epoch 150 yaitu 73%. Nilai precision, recall, dan mAP yang didapatkan yaitu 77,4%, 67,1%, dan 75,1%. Pengujian menghasilkan akurasi diatas 74% [9].

Penelitian sebelumnya pada penelitian yang dilakukan oleh Mira dkk (2023) dengan judul “Deteksi Jenis Penyakit Dan Hama Pada Tanaman Jagung Menggunakan Arsitektur Spatial Pyramid Pooling Pada YOLOv5s”. Pada penelitian ini, dilakukan deteksi pada jenis penyakit dan hama pada tanaman jagung menggunakan teknologi YOLO dengan arsitektur spatial pyramid pooling (SPP) sebagai bentuk inovasi kecerdasan pertanian tanaman jagung. Berdasarkan hasil training dan evaluasi model, diketahui bahwa nilai presisi dengan batch size 32 epoch 64 menghasilkan nilai presisi 0.65, recall, 0.76, f1 score 0.65 Map0.5 0.704 dan Map-.5:0.95 0.298. Sedangkan dengan batch size 64 epoch 100 nilai presisi 0.73, recall 0.77 f1 score 0.73 Map0.5 0.795 dan Map0.5:0.95 0.346 [10].

Penelitian yang mendeteksi jenis penyakit pada tanaman pohon jati terdapat pada Doli Indra Nasution dkk (2022) dengan judul “Sistem Pakar Mendiagnosis Penyakit Pada Pohon Jati Dengan Menggunakan Metode Teorema Bayes”. Pada PT Yasanda masih terdapat banyak kendala dalam proses mendiagnosa penyakit tanaman pohon kayu jati. Sistem pakar merupakan sistem yang menggunakan pengetahuan manusia ditangkap di komputer untuk memecahkan masalah yang biasanya membutuhkan keahlian manusia, maka bidang keilmuan yang digunakan adalah sistem pakar dengan menggunakan metode Teorema Bayes. Hasil penelitian ini untuk pengimplementasian sistem yang dilakukan menggunakan aplikasi mampu melakukan diagnosa Pohon Jati yang digunakan dengan baik, sehingga proses diagnosa tetap dapat dilakukan dengan baik [11].

Penelitian sebelumnya yang menggunakan sistem pakar diagnosis hama dan penyakit pada pohon jati ada pada, penelitian Agung Tri Laksono (2017) dengan judul “Sistem Pakar Diagnosis Hama dan Penyakit pada Pohon Jati Menggunakan Metode Certainty Factor Berbasis Android”. Pada penelitian ini dibuat sistem pakar menggunakan metode Certainty Factor yang memberi nilai fakta dan hipotesis untuk menentukan hasil solusi dari permasalahan yang dialami. Di dalam metode ini terdapat perhitungan nilai untuk memberikan suatu hasil serta metode Certainty Factor yang digunakan untuk mendiagnosa hama dan penyakit pada tanaman pohon jati serta dalam penelitian ini menggunakan Eclipse dalam pembuatan aplikasi sistem pakar berbasis android [3].

Penelitian sebelumnya yang menggunakan sistem deteksi daun menggunakan yolo terdapat pada penelitian Muhammad Ibna Mauladany dkk (2024) dengan judul “Deteksi Penyakit Daun Durian dengan Algoritma YOLO (You Only Look Once)”. 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 [11].

a. Algoritma Yolo

Merupakan algoritma pendeteksi objek berupa gambar atau video dengan metode deep learning secara real-tim dengan tingkat akurasi yang tinggi dan kecepatan pemrosesan yang cepat. Yolo bekerja dengan mendeteksi objek untuk mengatasi regresi tunggal melalui pixel gambar ke koordinat kotak pembatas dan probabilitas kelas. Yolo bekerja dengan membagi gambar menjadi $S \times S$ grid, pusat objek saling menimpa dengan sel grid maka sel grid akan mendeteksi objek tersebut. Setiap sel grid memprediksi bounding box B dan skor confidence di detail kotak berdasarkan kelas nya. Skor confidence berguna untuk memberikan seberapa akurat kotak yang berisi objek untuk di deteksi dan seberapa akurat hasil prediksi yang diberikan[12].

b. Deep Learning

Dalam pembelajaran mesin, deep learning adalah sekumpulan algoritma yang berusaha untuk belajar dalam berbagai tingkat, sesuai dengan tingkat abstraksi yang berbeda. Dalam model statistik yang dipelajari, tingkat konsep ditentukan oleh tingkat yang lebih rendah, dan tingkat yang lebih tinggi dapat membantu mendefinisikan banyak konsep.

c. Confusion Matrix

Teknik untuk mengukur kualitas keputusan dalam pembelajaran mesin confusion matrix bekerja dengan memvisualisasikan nilai tingkat kebingungan dari algoritma pada kelas yang berbeda, confusion matrix digunakan untuk menghitung akurasi dari konsep data mining. Tingkat akurasi, presisi, dan recall lalu akan menghasilkan evaluasi menggunakan model confusion matrix. Nilai akurasi adalah ketetapan record data yang sudah diklasifikasikan secara benar setelah dilakukan pengujian pada hasil kalsifikasi. Ketetapan situasi positif yang benar memberikan temuan positif pada data aktual yang dikenal dengan presisi atau confidence. Berikut merupakan tabel perhitungan confusion matrix.

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

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

$$F1\ score = 2 \times \frac{P \times R}{P+R}$$

Tabel 1. Confusion Matrix

| Prediksi | Keterangan |
|----------|----------------|
| TP | True Positive |
| FP | False Positive |
| FN | False Negative |
| TN | True Negative |
| P | Precision |
| R | Recall |

Keterangan pada persamaan (1) Rasio dari semua hal yang relevan yang dipilih dikenal dengan presisi. Recall adalah proporsi dari hal-hal relevan yang dipilih untuk semua item relevan yang dapat diakses pada rumus persamaan (2). Angka untuk mencetak rata-rata persisi dan recall yang kemudian menghasilkan F1 Score berada pada persamaan rumus (3) [12].

3. METODE PENELITIAN

Tahapan penelitian ini dimulai dengan proses pengumpulan data yang terdiri dari jenis hama dan penyakit pada tanaman jati. Selanjutnya dilakukan proses Labeling data dengan proses anotasi, membagi data menjadi data *train*, *test* dan *validasi* dan yang terakhir ialah membangun model pelatihan serta melakukan deteksi data. Tahapan penelitian dilakukan sebagai berikut:

Tabel 2. Perangkat Lunak Yang Digunakan

| No | Perangkat Keras | Kegunaan |
|----|---------------------|---|
| 1 | <i>Google Colab</i> | Menjalankan dan menguji pengembangan model. |
| 2 | <i>Roboflow</i> | Mengelola dataset serta membangun model pelatihan data. |

1. Pengumpulan dataset

Dataset diperoleh dari sumber internet yang berisi gambar-gambar hama dan penyakit pohon jati melalui Google, jumlah keseluruhan dataset yang diambil 181 dengan format file jpg. Gambar yang dibutuhkan dalam penelitian ini dibagi menjadi 9 jenis class yaitu:

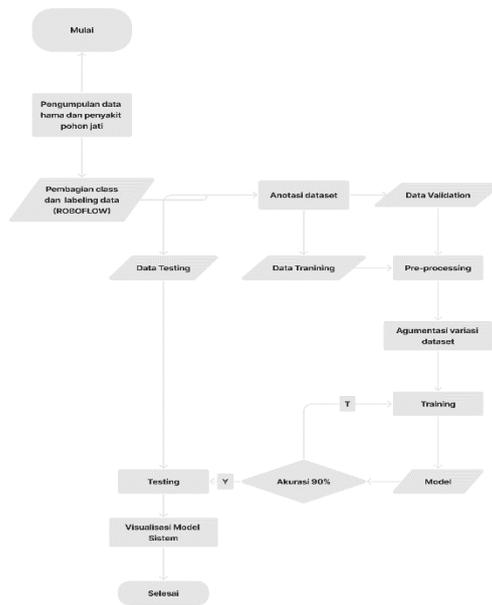
Tabel 3. Dataset Jenis Hama dan Penyakit Pohon Jati

| No | Nama hama dan penyakit | Gambar |
|----|---|--|
| 1 | Busuk akar dan pangkal batang |  |
| 2 | Hama penggerak batang (<i>Duomitus-ceramicus</i>) |  |
| 3 | Penyakit daun <i>Skeletoize</i> |  |

| | | |
|---|---|--|
| 4 | Penyakit bercak daun (<i>Hawar</i>) |  |
| 5 | Penyakit daun <i>Defoliasi</i> |  |
| 6 | Penyakit daun karat |  |
| 7 | Penyakit daun <i>Klorosis</i> |  |
| 8 | Kutu-putih (<i>Pseudococcus-Mealibug</i>) |  |
| 9 | Ulat-jati (<i>Hyblaea puera</i>) |  |

2. Perancangan Sistem

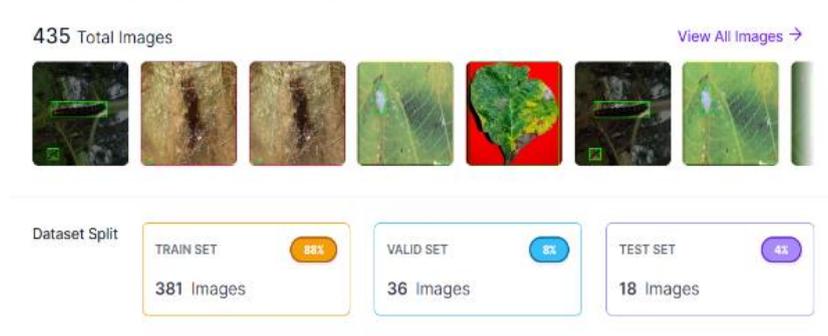
Proses ini dilakukan untuk membangun tahap selanjutnya dari pengumpulan dataset dan berlanjut dengan dilakukannya perancangan sistem yang bisa dilihat dari Gambar 1 berikut:



Gambar 1. Flowchart Perancangan sistem

a. Anotasi dataset

Tahap ini dilakukan untuk membuat dataset label yang berisi informasi nama kelas, titik koordinat serta ukuran objek, proses ini dilakukan dengan bantuan aplikasi *Roboflow*. Gambar yang telah melalui proses anotasi kemudian dibagi menjadi 3 data yaitu *training set*, *validation set*, dan *test set* pembagian ini dibuat perbandingan 70%, 20%, 10% sehingga data total yang diperoleh menjadi 381 *train set*, 36 *valid set*, 18 *test set*.



Gambar 1. Proses pembagian anotasi data split

b. Preprocessing

Bertujuan agar meningkatkan kualitas dataset sebelum dilakukan pelatihan model deteksi objek. *Auto-Orient* untuk menghilangkan informasi EXIF dari data gambar sehingga tampilan sesuai dengan penyimpanan pada disk[9]. Dimensi data menyesuaikan menjadi 450 x 450 piksel.



Gambar 2 Preprocessing dataset

c. Augmentasi

Digunakan untuk memperkaya dataset yang sudah ada, agar data yang didapat lebih

banyak dan beragam sehingga menghasilkan model yang lebih bervariasi [13].

```

Augmentations      Outputs per training example: 3
                   Grayscale: Apply to 10% of images
                   Hue: Between -38° and +38°
                   Saturation: Between -29% and +29%
                   Bounding Box: Blur: Up to 0.6px
    
```

Gambar 4. Proses Augmentasi dataset

d. Modelling

Modelling dilakukan setelah melewati proses *training dataset*, yang telah di olah pada *roboflow*. Selanjutnya *roboflow* akan memberikan API KEY seperti gambar 5. Setelah dataset gambar siap model pelatihan deteksi objek dan *instance segmentation* menggunakan arsitektur YOLOv8. Proses pelatihan menerapkan beberapa langkah seperti memilih dataset pelatihan dan validasi, hyperparameter serta melakukan pelatihan model yang seperti gambar 5.

```

!mkdir (HOME)/datasets
%cd (HOME)/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="edB0aCNGYo1yQFTSvhwM")
project = rf.workspace("penyakit-pada-daun-jati").project("hama-dan-penyakit-pohon-jati")
version = project.version(3)
dataset = version.download("yolo8")
    
```

Gambar 3 Proses impor Dataset dari Roboflow ke Google Colab

```

!yolo task=detect mode=train model=yolo8.pt data=(dataset.location)/data.yaml epoch=25 imgsz=200 plots=True

from 0 param: module augmentations
-1 1 328 ultralytics.nn.modules.conv Conv [7, 22, 3, 1]
1 1 18768 ultralytics.nn.modules.conv Conv [59, 64, 3, 2]
2 -1 22996 ultralytics.nn.modules.block C2F [64, 64, 1, True]
3 -1 1 73984 ultralytics.nn.modules.conv Conv [64, 128, 3, 2]
4 -1 2 187632 ultralytics.nn.modules.block C2F [128, 128, 3, True]
5 -1 1 295248 ultralytics.nn.modules.conv Conv [128, 256, 3, 2]
6 -1 2 788080 ultralytics.nn.modules.block C2F [256, 256, 3, True]
7 -1 1 1130872 ultralytics.nn.modules.conv Conv [256, 512, 3, 2]
8 -1 1 1438080 ultralytics.nn.modules.block C2F [512, 512, 3, True]
9 -1 1 884896 ultralytics.nn.modules.block SPPF [512, 512, 1]
10 -1 1 0 torch.nn.modules.upsampling.upsample [None, 2, 'nearest']
11 [-1, 0] 1 0 ultralytics.nn.modules.conv Convcat [3]
12 -1 1 591360 ultralytics.nn.modules.block C2F [768, 256, 1]
13 -1 1 0 torch.nn.modules.upsampling.upsample [None, 2, 'nearest']
14 [-1, 4] 1 0 ultralytics.nn.modules.conv Convcat [1]
15 -1 1 148224 ultralytics.nn.modules.block C2F [768, 128, 1]
16 -1 1 147712 ultralytics.nn.modules.conv Conv [128, 128, 3, 2]
17 [-1, 2] 1 0 ultralytics.nn.modules.conv Convcat [4]
18 -1 1 491856 ultralytics.nn.modules.block C2F [768, 256, 1]
19 -1 1 590336 ultralytics.nn.modules.conv Conv [256, 256, 3, 2]
20 [-1, 9] 1 0 ultralytics.nn.modules.conv Convcat [1]
21 -1 1 1899152 ultralytics.nn.modules.block C2F [768, 512, 1]
22 [15, 18, 21] 1 1129232 ultralytics.nn.modules.head.detect [9, [128, 256, 512]]
model summary: 225 layers, 1113863 parameters, 1113967 gradients, 28.7 GFLOPS
    
```

Gambar 4 Proses Pelatihan Model pada YOLOv8

e. Training Dataset

Proses ini dilakukan setelah tahap modelling selesai di jalankan dan selesai pada tahap pelatihan model YOLOv8. Proses training dilakukan menggunakan *data.yaml* dari Roboflow dengan file YOLOv8.ipynb, dengan mengatur konfigurasi *image size*, *batch size*, *epoch*, *data source* dan *weight*, *mAp*. Konfigurasi dalam proses training dapat dilihat pada tabel 3.

Tabel 4. Konfigurasi Training dataset

| Parameter | Nilai |
|-------------|-------------------|
| Image Size | 450 |
| Batch Size | 16 |
| Epoch | 25, 10,30,100,150 |
| Data Source | Data.yaml |
| Weight | YOLOv8.ipynb |
| imgsz | 800,200 |
| mAp | 50-95 |

f. Evaluasi

Setelah model dan pelatihan selesai dijalankan selanjutnya adalah proses evaluasi model untuk menganalisis dan mengevaluasi hasil tahapan yang sudah dilakukan. Untuk analisis model menggunakan metrik evaluasi berupa *confusion matrix* dengan menghitung nilai *precision*, *recall*, *mAp* dan *F1-Score* [14]. *Confusion matrix* digunakan untuk

mengevaluasi suatu model klasifikasi dengan membandingkan prediksi model dengan label sebenarnya [15].

3. Pengujian

Pada tahapan ini dilakukan uji data testing menggunakan model deteksi objek YOLOV8 yang sudah mengalami proses pelatihan dan menghasilkan nilai mAp tertinggi. Pengujian ini dilakukan dengan evaluasi yang mencakup *Precision*, *Recall* dan *f1 score* serta menghitung nilai loss dan acc persamaan indikator evaluasi model sebagai berikut:

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

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

$$F1\ score = 2 \times \frac{P \times R}{P+R} \quad (3)$$

Tabel 5. Confusion Matrix

| Prediksi | Keterangan |
|----------|----------------|
| TP | True Positive |
| FP | False Positive |
| FN | False Negative |
| TN | True Negative |
| P | Precision |
| R | Recall |

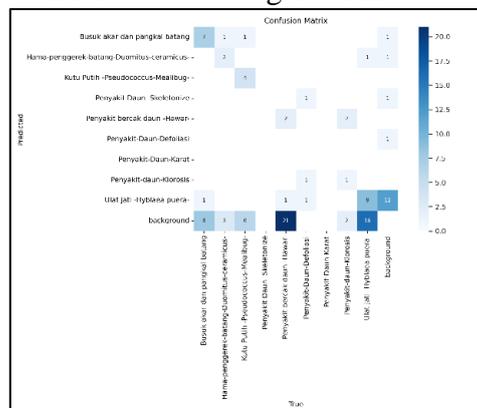
Data *loss* digunakan untuk memaksimalkan kinerja algoritma arsitektur yang terpilih saat melakukan proses *train*, sedangkan *accuracy* untuk mengukur *library* dan modul dari open CV dan YOLOv8. Prediksi teknologi YOLOv8 membagi gambar input menjadi *SxS* grid, prediksi pembatas B. Setiap titik memiliki skor *confidence* dan *conditional category probabilities*. *Loss function* menghitung jumlah posisi lebar, tinggi dan *error confidence* dari prediksi *box* menggunakan *mean square error* [6].

4. HASIL DAN PEMBAHASAN

Pada percobaan ini akan dilakukan model training dataset sebanyak 2 kali percobaan dengan epoch 25, 10, dari metode penelitian diatas akan menghasilkan sistem deteksi hama dan penyakit pada pohon jati menggunakan algoritma YOLOv8 yang memiliki akurasi model sebagai berikut:

a. Hasil Evaluasi Model

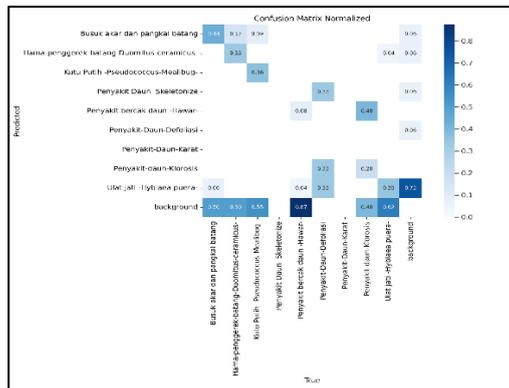
Hasil percobaan ini telah menghasilkan analisis model berupa confusion matrix serta nilai *precision*, *recall*, *mAp* dan *F1-Score* sebagai berikut :



Gambar 5. Confusion Matrix

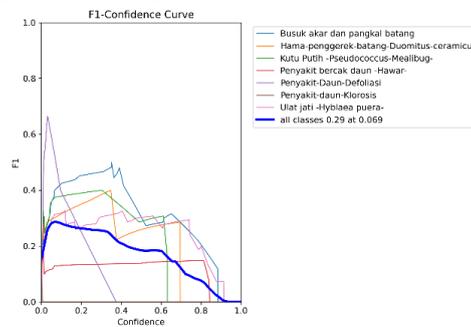
Confusion Matrix digunakan untuk evaluasi kinerja model klasifikasi dengan

membandingkan hasil prediksi dengan tabel sebenarnya dari objek. Pada gambar diatas training model menggunakan 435 gambar organisme pengganggu tanaman pada pohon jati yang terdiri dari 9 jenis penyakit dan hama, model training dilakukan 5 kali percobaan, dataset yang dilatih pada gambar telah menunjukkan bahwa model mampu memprediksi gambar sesuai dengan kelas label yang telah diberikan dan telah menghasilkan confusion metrik seperti gambar di atas.



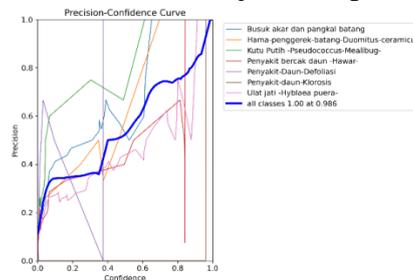
Gambar 6. Confusion Matrix Normalized

Confusion Matrix Normalized adalah matrix yang dinormalisasi menggambarkan proporsi atau persentase dari kesalahan klasifikasi dalam setiap kelas pada gambar yang di training pada Hal ini sangat membantu untuk memahami tentang kinerja model klasifikasi dalam berbagai kelas pada gambar yang di latih.



Gambar 7. F1 Confidence Curve

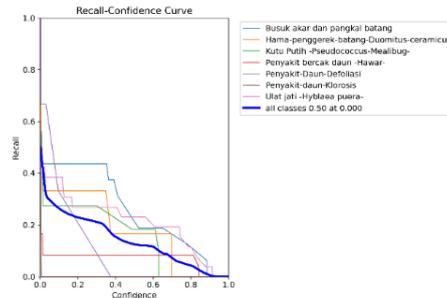
F1 Confidence Curve merupakan kurva yang menunjukkan tingkat kepercayaan semua kelas objek yang di deteksi oleh model deteksi YOLOv8. F1 score adalah matrix evaluasi untuk mengukur keseimbangan antara presisi dan recall pada model deteksi objek. Sedangkan *Confidence level* adalah tingkat keyakinan model dalam prediksi objek yang dinyatakan sebagai nilai dari probabilitas. Pada gambar 9 grafik skor Fi pada nilai *confidence* mencapai nilai rata-rata sebesar 0.29 terhadap nilai *confidence* 0.069, nilai ini menunjukkan bahwa model performa deteksi objek cukup baik.



Gambar 8. Precision-Confidence Curve

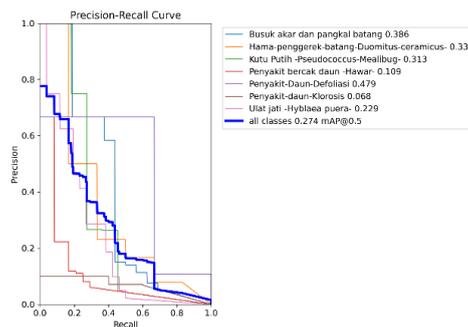
Pada gambar 10 grafik nilai *Precision* terhadap nilai *confidence all classes* 1.00 artinya semua kelas objek dideteksi oleh model YOLOv8 memiliki presisi sebesar 1.00,

nilai ini menunjukkan model yang dimiliki menunjukkan performa deteksi yang sangat akurat dan tidak membuat kesalahan dalam prediksi. Maka semua prediksi tersebut dinyatakan benar atau tidak ada kesalahan dalam prediksi yang telah dilakukan.



Gambar 9. Recall-Confidence Curve

Pada gambar 11 nilai *recall confidence curve* semua kelas objek sebesar 0.50, ini menunjukkan model mampu menemukan semua objek sebenarnya untuk setiap kelas objek yang ada, sedangkan 0.000 menunjukkan recall dihitung pada nilai *confidence threshold*.



Gambar 10. Precision-Recall Curve

Gambar 12 menunjukkan grafik nilai *Precision-Recall* terhadap nilai *confidence*, hasil training pada menghasilkan nilai rata-rata maksimal sebesar 0.274 terhadap nilai mAP 0.5. Semakin tinggi nilai mAP semakin baik performa yang didapatkan. Nilai dari kedua grafik tersebut dapat disimpulkan bahwa nilai *Precision-Recall* terhadap nilai *confidence* berjalan lebih baik.

Dataset telah dilatih 5 kali kemudian dievaluasi. Tabel 5 menunjukkan perbandingan hasil pelatihan setelah pengumpulan nilai precision, recall, F1-score, dan mAP.

Tabel 6. Perbandingan hasil *training*

| <i>Epoch</i> | <i>Precision</i> | <i>Recall</i> | <i>mAP</i> | <i>F1-Score</i> |
|--------------|------------------|---------------|------------|-----------------|
| 10 | 0.07 | 0.62 | 0.249 | 0.138 |
| 25 | 0.108 | 0.50 | 0.274 | 0.069 |
| 30 | 0.348 | 0.68 | 0.348 | 0.200 |
| 100 | 0.979 | 0.264 | 0.728 | 0.628 |
| 150 | 0.962 | 0.255 | 0.734 | 0.553 |

Pada hasil *training* model terbaik di dapat pada *epoch* 100 yang dapat dilihat dari *precision* yang menghasilkan nilai sebesar 0.979, mAP 0.728, *recall* sebesar 0.264 dan *F1-Score* sebesar 0.628.

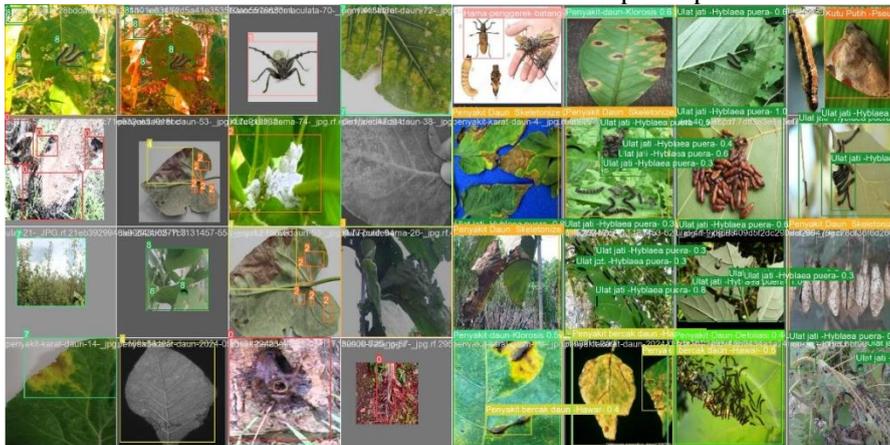
b. Hasil Pengujian Model

Train batch adalah parameter untuk mengatur data gambar yang di proses pada satu iterasi ketika melakukan pelatihan model. Secara spesifik train batch adalah model pelatihan YOLO yang menggunakan konsep deep learning untuk menghitung gradient dan pembaharuan bobot berdasarkan data latih, train batch juga mengontrol jumlah data gambar, semakin besar data gambar yang diproses dalam satu iterasi semakin cepat waktu

pelatihan yang di peroleh hasil deteksi dapat dilihat pada gambar 13-17. Val batch adalah parameter untuk mengatur jumlah data gambar yang diproses ketika melakukan evaluasi model pada data validasi. Ketika melakukan evaluasi data validasi tidak melakukan pembaharuan bobot sebaliknya hanya memperoleh prediksi data validasi untuk menghitung metrik seperti akurasi atau AP(Average Precision) hasil deteksi bisa dilihat pada gambar 13-17.



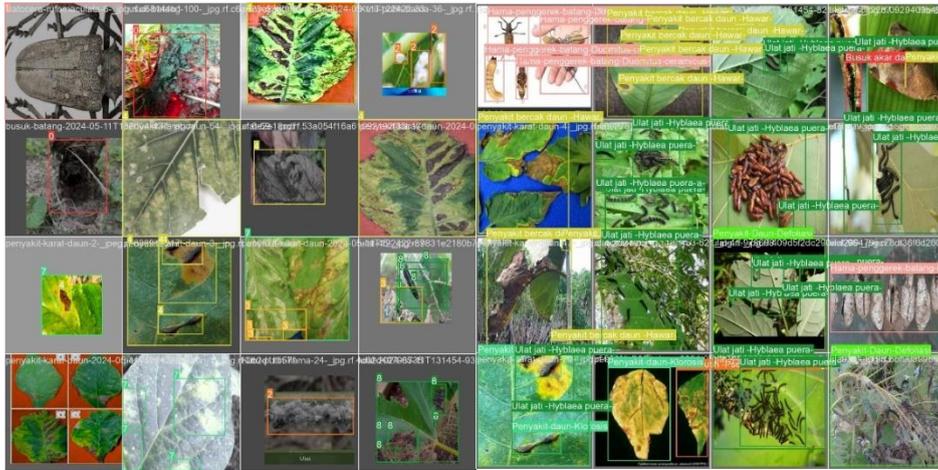
Gambar 11. Train batch 0 dan Val batch 0 pred epoch 25



Gambar 12. Train batch 0 dan Val batch 0 pred epoch 10



Gambar 15. Train batch 0 dan Val batch 0 pred epoch 30



Gambar 13. Val batch 0 pred epoch 100



Gambar 14. Train batch 0 dan Val batch 0 pred epoch 150

4. KESIMPULAN

Berdasarkan pembahasan dan hasil penelitian yang sudah dilakukan pada sistem deteksi hama dan penyakit pada pohon jati dengan algoritma YOLOv8, didapatkan kesimpulan penelitian ini menerapkan teknik Optimasi, segmentasi dan data argumentasi model deteksi objek untuk mengenali hama dan penyakit pada pohon jati yang telah menghasilkan model latih yang baik performanya. Pada penelitian ini dilakukan uji model sebanyak 5 kali dengan epoch 25, 10, 30, 100, 150 batch 16, telah menghasilkan model terbaik pada epoch 100 dengan nilai precision yang menghasilkan nilai sebesar 0.979, mAp 0.728, recall sebesar 0.264 dan F1-Score sebesar 0.628. Dari hasil training model yang sudah dilakukan pengujian pada model terbaik telah menghasilkan dataset dengan tingkat akurasi sebesar 87.6%, validation sebesar 8.3%, test 4.1%.

DAFTAR PUSTAKA

- A. T. L. Pamungkas, "Siyakit pada Pohon Jati Menggunakan Sistem Pakar Diagnosis Hama dan Metode Certainty Factor Berbasis Android," *J. Mhs. Tek. Inform.*, vol. 1, no. 1, pp. 73–80, 2017.
- D. I. Nasution, I. Zulkarnain, and S. Kusnari, "Sistem Pakar Mendiagnosis Penyakit Pada Pohon Jati Dengan Menggunakan Metode Teorema Bayes," *J. Sist. Inf. Triguna Dharma (JURSI TGD)*, vol. 1, no. 4, p. 507, 2022, doi: 10.53513/jursi.v1i4.5738.
- J. Brownlee, "Deep Learning for Computer Vision Image Classification, Object Detection, and Face Recognition in Python," *Deep Learn. Comput. Vis.*, p. 532, 2019, [Online]. Available: <https://machinelearningmastery.com/deep-learning-for-computer-vision/>

- J. Patty and C. Uruilal, "DIAGNOSIS JENIS PENYAKIT TANAMAN JATI (*Tectona Grandis*) PADA AREAL HUTAN TANAMAN DESA HATUSUA KECAMATAN KAIRATU KABUPATEN SERAM BAGIAN BARAT," *J. Hutan Pulau-Pulau Kecil*, vol. 1, no. 2, p. 136, 2016, doi: 10.30598/jhppk.2016.1.2.136.
- J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.
- K. A. Baihaqi and Y. Cahyana, "Application of Convolution Neural Network Algorithm for Rice Type Detection Using Yolo v3," *Systematics*, vol. 3, no. 2, pp. 272–280, 2021.
- L. Firgia and S. Thomas, "Deteksi Jenis Penyakit Dan Hama Pada Tanaman Jagung Menggunakan Arsitektur Spatial Pyramid Pooling Pada YOLOv5s," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 2, pp. 452–459, 2023, [Online]. Available: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- M. I. Mauladany, B. Fatkhurrozi, and R. A. Wibowo, "Deteksi Penyakit Daun Durian dengan Algoritma YOLO (You Only Look Once)," *Avitec*, vol. 6, no. 1, p. 73, 2024, doi: 10.28989/avitec.v6i1.2067.
- M. Ibrahim and U. Latifa, "Penerapan Algoritma Yolov8 Dalam Deteksi Waktu Panen Tanaman Pakcoy Berbasis Website," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 4, pp. 2489–2495, 2024, doi: 10.36040/jati.v7i4.7154.
- M. Pailus, D. Hatta Fudholi, and S. Hidayat, "Model Identifikasi Penyakit Pada Tumbuhan Padi Berbasis DenseNet," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 6, no. 2, pp. 615–625, 2022.
- Muhammad Nur Ihsan Muhlashin and A. Stefanie, "Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 2, pp. 1363–1368, 2023, doi: 10.36040/jati.v7i2.6927.
- N. M. Yasen, S. Rifka, R. Vitria, and Y. Yulindon, "Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode Deep Learning," *Elektron J. Ilm.*, vol. 15, pp. 63–71, 2023, doi: 10.30630/eji.0.0.397.
- R. F. Putra and D. I. Mulyana, "Optimasi Deteksi Objek Dengan Segmentasi dan Data Augmentasi Pada Hewan Siput Beracun Menggunakan Algoritma You Only Look Once (YOLO)," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 1, pp. 93–103, 2024, doi: 10.35870/jtik.v8i1.1391.
- S. Wulan Dari and J. Triloka, "Kajian Algoritme Mask Region-Based Convolutional Neural Network (Mask R-CNN) dan You Look Only Once (YOLO) Untuk Deteksi Penyakit Kulit Akibat Infeksi Jamur," *Semin. Nas. Has. Penelit. dan Pengabd. Masy.*, pp. 132–138, 2022.
- T. Pratiwi, K. Karmanah, and R. Gusmarianti, "Inventarisasi Hama Dan Penyakit Tanaman Jati Unggul Nusantara Di Kebun Percobaan Cogrek Bogor," *J. Sains Nat.*, vol. 2, no. 2, p. 123, 2017, doi: 10.31938/jsn.v2i2.42.