

IMPLEMENTATION OF YOU ONLY LOOK ONCE V8 ALGORITHM IN POTATO LEAF DISEASE DETECTION SYSTEM

Bagus Kurniawan Ekhsanto^{*1}, Bagus Adhi Kusuma², Adam Prayogo Kuncoro³

^{1,2,3}Informatics, Computer Science Faculty, Universitas Amikom Purwokerto, Indonesia

Email: ¹baguskurniawanpwt@gmail.com, ²bagus@amikompurwokerto.ac.id, ³adam@amikompurwokerto.ac.id

(Article received: May 17, 2024; Revision: May 20, 2024; published: July 29, 2024)

Abstract

Agriculture is an important foundation of the national economy, as effective development in this sector will support overall economic stability. Potato itself is one of the world's staple foods after rice, wheat and corn. This crop belongs to the category of horticulture which is widely planted and developed by people to meet their needs. On the farm of Bibit sida kangen Kalibening, Banjarnegara which is one of the farms that grow potatoes has constraints related to potato diseases which result in decreased productivity of crops. Therefore, the main purpose of this system is to provide fast and accurate disease detection capability on the farm of Bibit sida kangen Kalibening, Banjarnegara, so that it can help farmers in reducing losses caused by disease attacks on plants. By utilizing YOU ONLY LOOK ONCE V8 (YOLOv8) technology, this system can recognize and classify potato leaf disease types, including early_blight, late_blight, and healthy plants, with a high level of accuracy. Through evaluation using precision and recall matrices, the results show a significant success rate, with precision accuracy for early_blight of 87%, healthy plants of 81%, and late_blight of 97%, respectively. Meanwhile, the recall results for the three categories also reached 87%, 81%, and 97% respectively. With an overall accuracy of 88%, these findings confirm that the developed detection system is successful in identifying potato leaf diseases with high accuracy. This indicates the great potential of this system in assisting farmers in managing the condition of their potato crops, which in turn can improve farmers' productivity and welfare.

Keywords: agriculture, detection system, potato leaf diseases, YOLOv8.

IMPLEMENTASI ALGORITMA YOU ONLY LOOK ONCE V8 PADA SISTEM DETEKSI ENYAKIT DAUN KENTANG

Abstrak

Pertanian merupakan fondasi ekonomi nasional yang penting, karena pembangunan yang efektif di sektor ini akan mendukung stabilitas ekonomi secara keseluruhan. Kentang sendiri adalah salah satu makanan pokok dunia setelah beras, gandum, dan jagung. Tanaman ini termasuk dalam kategori hortikultura yang ditanam dan dikembangkan secara luas oleh masyarakat untuk memenuhi kebutuhan mereka. Pada pertanian Bibit sida kangen Kalibening, Banjarnegara yang merupakan salah satu pertanian yang menanam kentang memiliki kendala terakait penyakit kentang yang mengakibatkan menurunnya produktivitas hasil panen. Oleh karena itu, tujuan utama sistem ini adalah untuk memberikan kemampuan deteksi penyakit secara cepat dan akurat pada pertanian Bibit sida kangen Kalibening, Banjarnegara, supaya bisa membantu petani dalam mengurangi kerugian yang disebabkan oleh serangan penyakit pada tanaman. Dengan memanfaatkan teknologi YOU ONLY LOOK ONCE V8 (YOLOv8), sistem ini dapat mengenali dan mengklasifikasikan jenis penyakit daun kentang, termasuk early_blight, late_blight, dan tanaman sehat, dengan tingkat akurasi yang tinggi. Melalui evaluasi menggunakan matriks precision dan recall, hasilnya menunjukkan tingkat keberhasilan yang signifikan, dengan akurasi precision masing-masing untuk early_blight sebesar 87%, tanaman sehat 81%, dan late_blight 97%. Sementara itu, hasil recall untuk ketiga kategori tersebut juga mencapai 87%, 81%, dan 97% secara berurutan. Dengan akurasi keseluruhan mencapai 88%, temuan ini menegaskan bahwa sistem deteksi yang dikembangkan berhasil dalam mengidentifikasi penyakit daun kentang dengan akurasi tinggi. Ini mengindikasikan potensi besar sistem ini dalam membantu petani dalam mengelola kondisi tanaman kentang mereka, yang pada gilirannya dapat meningkatkan produktivitas dan kesejahteraan petani.

Kata kunci: penyakit daun kentang, pertanian, system deteksi, YOLOv8.

1. PENDAHULUAN

Pertanian merupakan fondasi ekonomi nasional yang penting, karena pembangunan yang efektif di sektor ini akan mendukung stabilitas ekonomi secara keseluruhan. Dalam konteks pertanian, kemajuan teknologi harus diarahkan untuk meningkatkan hasil dan kesejahteraan petani [1]. Dengan kemajuan teknologi dan penerapan *machine learning*, industri pertanian telah mengalami transformasi yang signifikan. Perubahan ini telah meningkatkan produktivitas dan keinginan industri [2]. Pada masa lalu, pertanian biasanya mengandalkan pengetahuan dan pengalaman langsung dari para petani. Namun, penerapan *machine learning* telah menjadi titik balik penting dalam mengangkat pertanian ke tingkat yang lebih maju. *Algoritma machine learning* mampu mengolah dan menganalisis data yang besar dan kompleks dengan cepat, memberikan pemahaman yang mendalam tentang berbagai tantangan yang terkait dengan pertanian [3]. Oleh karena itu, para petani dapat membuat keputusan yang lebih cerdas dan tepat waktu berkat adopsi teknologi ini, yang membantu mereka mengoptimalkan hasil panen dan mengurangi pemborosan sumber daya.

Kentang adalah salah satu makanan pokok dunia setelah beras, gandum, dan jagung. Tanaman ini termasuk dalam kategori *hortikultura* yang ditanam dan dikembangkan secara luas oleh masyarakat untuk memenuhi kebutuhan mereka. Kentang merupakan komoditas *hortikultura* yang dapat menjadi alternatif untuk beras karena umbinya kaya akan karbohidrat, vitamin, dan mineral yang penting [4]. Meskipun demikian, pertanian kentang tidak terlepas dari tantangan, di antaranya adalah masalah kesehatan tanaman, terutama yang berkaitan dengan penyakit daun kentang [5]. Penyakit daun kentang menjadi salah satu tantangan utama dalam pertanian di Indonesia, penyakit daun kentang yang biasa terjadi yaitu *Early blight* dan *late blight*, penyakit ini merupakan beberapa jenis penyakit daun kentang yang sering muncul dan memberikan dampak yang merugikan bagi para petani [6]. Penyakit daun pada tanaman kentang dapat menyebabkan kerusakan yang signifikan pada tanaman dan mengurangi hasil panen secara drastis.

Bibit Sida Kangen adalah nama sebuah pertanian milik pribadi yang terletak di Kecamatan Kalibening, Banjarnegara. Tempat ini dikenal sebagai tempat penjualan bibit dan hasil pertanian seperti kentang, tomat, dan lain-lain. Menurut penelitian yang dilakukan oleh [7], tanaman kentang paling baik tumbuh di daerah dataran tinggi yang memiliki ketinggian melebihi 1.300 mdpl. Lingkungan optimal bagi pertumbuhan kentang mencakup suhu udara antara 15°C hingga 20°C, kelembapan udara mencapai 80% hingga 90%, serta curah hujan berkisar antara 200 hingga 300 mm per bulan atau 1500 mm per tahun. Tanah yang sesuai untuk pertumbuhan kentang adalah tanah yang

gembur dan kaya bahan organik, seperti tanah andosol, latosol, dan grumosol, dengan tanah andosol menjadi pilihan utama karena kandungan unsur hara yang cukup tinggi. Dieng, Banjarnegara, dikenal sebagai salah satu daerah yang terkenal dengan produksi kentangnya. Namun, kurangnya pengetahuan mengenai penyakit daun kentang berpotensi menyebabkan penurunan hasil produksi dalam jangka panjang. Penyakit daun kentang dan serangan hama merupakan dua dari banyak faktor yang dapat menyebabkan kegagalan panen dalam produktivitas tanaman kentang [8]. Diperlukan upaya untuk mengurangi risiko kegagalan panen dengan melakukan deteksi penyakit pada daun kentang sejak dini [9].

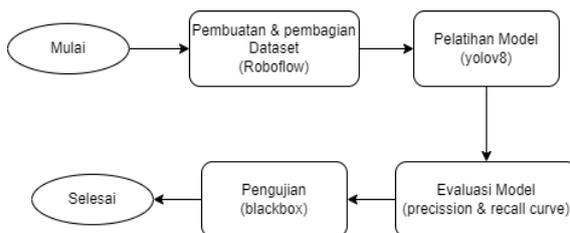
Untuk mendukung upaya tersebut, penggunaan teknologi modern dapat memberikan solusi yang efektif dalam mengatasi masalah ini dan membantu petani menghadapi tantangan di lapangan [10]. Dalam usaha untuk mengidentifikasi penyakit pada daun kentang, *YOLO* akan digunakan untuk membangun model *machine learning*. *YOLO* adalah sebuah algoritma yang telah mengubah paradigma pendeteksian objek dalam pengolahan gambar dan video. Tujuan utamanya adalah untuk mendeteksi objek secara instan dan *real-time* dengan kecepatan tinggi [11]. Model ini akan memberikan diagnosis yang lebih tepat dan cepat kepada petani melalui analisis visual gambar daun yang diambil. Perkembangan yang mengesankan telah terjadi dalam penelitian deteksi penyakit tanaman menggunakan *YOLO* [12]. *YOLO* memiliki beberapa versi, Setiap versi memiliki peningkatan kecepatan dalam membaca dan mendeteksi gambar serta video. Dalam penelitian ini, digunakan versi terbaru *YOLO* yaitu *V8* [13]. Menurut penelitian yang dilakukan oleh [14] terungkap bahwa dengan menggunakan arsitektur *YOLOv5*, telah berhasil dikembangkan sebuah sistem yang mampu mendeteksi penyakit yang mungkin menyerang daun paprika. Penggunaan *YOLOv5* ini menunjukkan tingkat akurasi yang tinggi dalam mengidentifikasi berbagai penyakit yang memengaruhi kondisi daun paprika, memungkinkan untuk intervensi tepat waktu dan pemantauan yang efisien terhadap kesehatan tanaman. Adapun Penelitian yang dilakukan oleh [15], dengan menggunakan *YOLOv5* juga menghasilkan temuan yang signifikan terkait sistem deteksi penyakit pada daun tomat. Dalam penelitiannya berhasil mencapai akurasi yang tinggi dalam mengidentifikasi berbagai jenis penyakit yang memengaruhi daun tomat. Selain itu ada juga penelitian yang dilakukan oleh Asif, Md. Khalid Rayhan Rahman, Md. Asfaqur Hena, dan Most. Hasna [16] menggunakan algoritma *Convolutional Neural Network (CNN)* untuk mengembangkan sistem deteksi penyakit pada daun kentang. Penelitian ini menghasilkan suatu sistem yang mampu mengenali dengan akurasi tinggi

berbagai jenis penyakit yang dapat menginfeksi daun kentang.

Menurut penelitian yang dilakukan oleh Nagaraj,dkk [17]nya, Arsitektur *YOLO* terkenal karena kecepatan yang lebih unggul dalam mendeteksi objek jika dibandingkan dengan *CNN*, memungkinkan penggunaan aplikasi *real-time* yang efisien dan responsive. Oleh karena itu, tujuan dari penelitian ini adalah membuat sebuah sistem deteksi penyakit pada daun kentang yang memanfaatkan keunggulan algoritma *YOLOv8*. Ini menunjukkan potensi besar dalam meningkatkan efisiensi dan akurasi dalam mengidentifikasi masalah kesehatan pada tanaman kentang, yang dapat memberikan kontribusi signifikan bagi industri pertanian dalam menghadapi tantangan penyakit tanaman.

2. METODE PENELITIAN

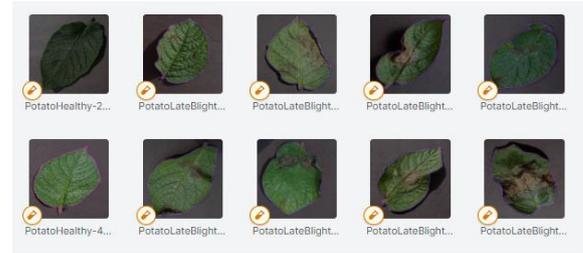
Penelitian ini memanfaatkan algoritma *YOLOv8* dengan metode *AI Project Cycle* yang melalui langkah-langkah sistematis, dimulai dari pembuatan *dataset* hingga *preprocessing* [18], yang mencakup pembagian data menjadi data pelatihan, pengujian, dan validasi. Fokus utama penelitian ini adalah mengimplementasikan model *YOLOv8* untuk mendeteksi penyakit pada daun kentang. Langkah-langkah prosedural meliputi prediksi dan evaluasi kinerja, dengan menggunakan metrik seperti *precision* dan *recall* untuk menilai model yang telah dilatih. Dalam pemrosesan data, peneliti menggunakan bahasa *Python* sebagai alat bantu untuk analisis dan pengolahan data, yang dilakukan di *Google Colab*. Semua langkah ini dijelaskan secara rinci dalam Gambar 1, yang memberikan gambaran grafis tentang prosedur dan hasil penelitian.



Gambar 1. Flowchart sistem deteksi

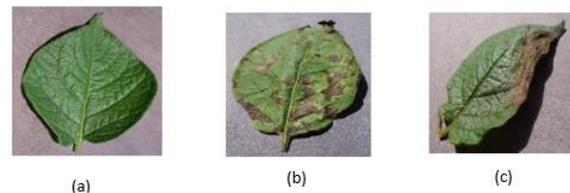
2.1. Pembuatan dan pembagian dataset

Pada tahap ini, penulis menyiapkan *dataset* yang diperoleh dari *Roboflow* yang dapat diakses pada link <https://universe.roboflow.com/plant-disease-slqvd/potato-leaf-disease/dataset/1>. *Roboflow* adalah situs web yang menyediakan berbagai jenis *dataset* untuk computer vision dan berfungsi sebagai platform berbagi *dataset* atau eksperimen dalam pengembangan metode untuk deteksi atau prediksi objek tertentu [19].



Gambar 2. Dataset

Pembuatan *dataset* melalui platform *Roboflow* adalah langkah penting dalam pengembangan berbagai aplikasi berbasis gambar seperti deteksi objek dan pengenalan pola. Platform ini memudahkan pengguna dalam mengumpulkan, mengelola, dan memproses sejumlah besar gambar [20], sehingga dapat menciptakan *dataset* yang sesuai dengan kebutuhan mereka. Dengan *dataset* berkualitas tinggi, kinerja model pembelajaran mesin dan akurasi analisis gambar dapat ditingkatkan. Pada penelitian menggunakan tiga kelas yaitu *healty*, *early blight*, dan *late blight*.



Gambar 3. (a) Healthy (b) Early blight (c) Late blight

Dalam penelitian ini, digunakan *dataset* yang terdiri dari 750 gambar yang dibagi ke dalam tiga kelas: *Healthy*, *early blight*, dan *late blight*. Terdapat 175 gambar *healthy*, 248 gambar *early blight*, dan 245 gambar *late blight*. Langkah selanjutnya yaitu Pembagian *dataset* yang dilakukan dengan alokasi 69% untuk set pelatihan, 20% untuk pengujian, dan 11% untuk validasi [21].

2.2. Pelatihan Model

Proses pelatihan model dilakukan menggunakan bahasa *Python* dengan bantuan *Google Colab* sebagai *Integrated Development Environment (IDE)* [22]. Untuk meningkatkan kualitas pemodelan, disarankan menggunakan *GPU* daripada *CPU* saat memproses kumpulan data gambar. *CPU* memiliki kegunaan yang lebih umum, sementara *GPU* dioptimalkan khusus untuk melatih model pembelajaran mendalam karena kemampuannya melakukan banyak penghitungan secara bersamaan. *GPU* memiliki jumlah inti yang besar, memungkinkan komputasi paralel yang efisien. Selain itu, karena komputasi *deep learning* seringkali melibatkan manipulasi data dalam jumlah besar, *bandwidth* memori *GPU* lebih cocok untuk tugas tersebut. Dalam pelatihan model ini, dilakukan beberapa tahapan seperti instalasi *library*, akuisisi data (*data acquisition*), dan pengaturan parameter.

```
[ ] !pip install ultralytics
⇄ Show hidden output

[ ] !pip install opencv-python
⇄ Show hidden output

[ ] !pip install roboflow
⇄ Show hidden output

[ ] from ultralytics import YOLO
import os

[ ] from IPython.display import Image
```

Gambar 4. Instalasi library

Untuk menjalankan proses pelatihan model ini, diperlukan beberapa *library* penting yang harus diinstal dan digunakan. *Library* tersebut meliputi *ultralytics*, *opencv-python*, *roboflow*, dan *YOLO*. *Ultralytics* menyediakan implementasi *YOLOv8*, yang merupakan model deteksi objek yang efisien dan banyak digunakan. *Opencv-python* adalah versi *Python* dari *OpenCV*, pustaka yang sangat berguna untuk pemrosesan citra dan video, serta berbagai aplikasi visi komputer. *Library Roboflow* digunakan untuk mengimport *dataset* dari *platform roboflow*, sehingga memudahkan integrasi *dataset* dengan model yang akan dilatih. Terakhir, *YOLO* adalah arsitektur jaringan saraf yang digunakan untuk deteksi objek dalam waktu nyata. Penggunaan *library-library* ini memungkinkan penulis untuk membangun, melatih, dan menguji model deteksi objek dengan lebih efisien dan efektif.

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="4YfrMTEub4CIVuPjte4w1")
project = rf.workspace("plant-disease-s1qvd").project("potato-leaf-disease")
version = project.version(1)
dataset = version.download("yolov8")
```

Gambar 5. Import dataset

Selanjutnya, dilakukan proses akuisisi data yang sangat penting terdiri dari *import* data dan memindahkan data *yaml*. Pada gambar 5 merupakan proses untuk mengimport *dataset* yang telah dibuat menggunakan *platform Roboflow* ke dalam lingkungan kerja *Google Colab*. Proses ini melibatkan beberapa langkah kritis, dimulai dari mengakses *dataset* yang telah disimpan di *Roboflow* hingga mengunduhnya dan memastikan bahwa data tersebut siap untuk digunakan dalam pelatihan model. *Roboflow* menyediakan berbagai fitur yang memudahkan pengelolaan dan anotasi *dataset*,

sehingga *dataset* yang diimpor ke *Google Colab* sudah terstruktur dengan baik dan siap untuk digunakan. Setelah *dataset* berhasil diimpor, data tersebut dapat langsung digunakan dalam berbagai operasi pemrosesan data dan pelatihan model di *Google Colab*. Proses ini memastikan bahwa *dataset* yang digunakan adalah *dataset* berkualitas tinggi, yang akan mendukung hasil yang lebih akurat dan efisien dalam pengembangan model pembelajaran mesin. Selain itu, penggunaan *Google Colab* memungkinkan pemanfaatan sumber daya komputasi yang lebih kuat, seperti *GPU*, yang dapat mempercepat proses pelatihan dan meningkatkan kinerja model. Dengan demikian, proses akuisisi data menjadi langkah kunci yang memastikan kelancaran dan keberhasilan proyek pemodelan yang dilakukan.

```
# pindah data.yaml
# Path to the source file
source_file = "/content/Potato-leaf-disease-1/data.yaml"

# Path to the destination directory
destination_directory = "/content/"

# Path to the desired destination file
destination_file = os.path.join(destination_directory, "data.yaml")

try:
    # Move the file
    os.rename(source_file, destination_file)
    print("File moved successfully!")
except Exception as e:
    print("An error occurred:", e)
```

Gambar 6. Memindahkan data yaml

Pada gambar 6 dijelaskan proses untuk memindahkan file *data.yaml* dari lokasi sumber ke lokasi tujuan. Pertama, path dari file sumber disimpan dalam variabel *source_file* dan path dari direktori tujuan disimpan dalam variabel *destination_directory*. Selanjutnya, path lengkap untuk file tujuan dibuat dengan menggunakan fungsi *os.path.join()* yang menggabungkan *destination_directory* dengan nama file. Kemudian, dilakukan percobaan pemindahan file menggunakan fungsi *os.rename()* yang mengubah nama atau memindahkan file dari *source_file* ke *destination_file*. Jika operasi pemindahan berhasil, pesan "*File moved successfully!*" akan ditampilkan; sebaliknya, jika terjadi kesalahan, pesan kesalahan yang spesifik akan ditampilkan bersama dengan informasi tentang kesalahan yang terjadi.

```
model.train(data='/content/data.yaml',
            task='detect',
            imgsz=640,
            epochs=50,
            batch=32,
            mode='train',
            name='yolov8n_v1_train')
```

Gambar 7. Pengaturan parameter

Langkah berikutnya dalam pemodelan adalah melakukan pelatihan *dataset* dengan menyesuaikan parameter seperti yang dijelaskan pada gambar 7.

Parameter 'data' merujuk pada lokasi file konfigurasi *dataset* yang akan digunakan dalam proses pelatihan model. File *data.yaml* berisi informasi penting seperti jalur ke *dataset* gambar, label kelas, dan konfigurasi lain yang dibutuhkan untuk pelatihan. Parameter 'task' menandakan jenis tugas yang akan dilakukan, yaitu '*detect*', yang menunjukkan bahwa model ini akan digunakan untuk mendeteksi objek. '*imgsz*' adalah ukuran gambar input yang digunakan selama pelatihan, di mana gambar akan diubah menjadi ukuran 640x640 piksel sebelum diproses oleh model. '*epochs*' menetapkan jumlah putaran pelatihan yang akan dilakukan oleh model, dalam hal ini sebanyak 50 kali putaran. '*batch*' menunjukkan ukuran *batch* data yang digunakan selama pelatihan, yaitu 32 gambar per *batch*. '*mode*' disetel sebagai '*train*', menunjukkan bahwa ini adalah tahap pelatihan model. '*name*' adalah nama yang diberikan untuk model yang sedang dilatih, yang digunakan untuk memberi nama dan mengidentifikasi berbagai versi model yang berbeda. Kode ini meneruskan parameter ke fungsi *model.train*, yang kemudian memulai proses pelatihan model YOLOv8 dengan konfigurasi yang telah ditentukan. Selama pelatihan, model akan belajar untuk mendeteksi objek dalam gambar berdasarkan *dataset* yang tersedia.

2.3. Evaluasi Model

Dalam penelitian ini, model dievaluasi menggunakan *precision* dan *recall* sebagai parameter utama untuk menilai kualitas hasil pelatihan [23]. *Precision* menilai keakuratan model dalam mengklasifikasikan data positif, sementara *recall* mengukur kemampuan model dalam mendeteksi dan mencakup seluruh data yang seharusnya positif. Kombinasi kedua parameter ini memberikan pemahaman menyeluruh tentang kinerja model dalam konteks klasifikasi. Analisis difokuskan pada mencapai keseimbangan optimal antara *precision* dan *recall*, dengan mempertimbangkan potensi *trade-off* di antara keduanya. Dengan menggunakan *precision* dan *recall*, studi ini memberikan wawasan mendalam tentang kemampuan model dalam memberikan prediksi yang akurat dan efisien.

Precision adalah metrik evaluasi yang mengukur ketepatan model dalam mengidentifikasi instans positif. Rumus *precision* dinyatakan sebagai rasio dari *True Positives* (instans positif yang diprediksi benar) terhadap total prediksi positif (*True Positives + False Positives*). Dengan kata lain, *precision* mengukur seberapa banyak dari kasus yang diprediksi positif oleh model benar-benar positif dan *Recall* adalah metrik evaluasi yang mengukur kemampuan model untuk menemukan semua instans positif yang sebenarnya. Rumus *recall* dihitung sebagai rasio dari *True Positives* terhadap total instans positif (*True Positives + False Negatives*) [24]. Untuk rumus matematika *precision* dan *recall* dapat dilihat pada persamaan 1 dan 2. Sedangkan

untuk rumus mean average *precision* dapat dilihat pada persamaan 3.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

2.4. Pengujian

Pada tahap selanjutnya, model akan diuji menggunakan metode *blackbox*. Metode *blackbox* adalah sebuah pendekatan evaluasi yang menguji model tanpa memperhatikan struktur internalnya, hanya melalui *input* dan *output* yang dihasilkan. Dengan kata lain, metode ini memperlakukan model sebagai sebuah "kotak hitam" di mana hanya *input* dan *output* yang diamati, tanpa memperhatikan detail implementasi internalnya. Dalam konteks ini, model akan diuji dengan menggunakan gambar daun sehat, *early blight*, dan *late blight* sebagai *input*, dan hasil klasifikasi yang dihasilkan oleh model akan diamati. Metode *blackbox* ini membantu untuk mengevaluasi kinerja model secara keseluruhan, tanpa perlu mengetahui detail implementasi atau struktur internal dari model tersebut.

3. HASIL DAN PEMBAHASAN

Hasil penelitian ini akan dilihat berdasarkan hasil dari beberapa percobaan pelatihan model yang dilakukan dan pengujian menggunakan metode *blackbox* pada model yang akan menghasilkan informasi yang penting untuk evaluasi kinerja. Melalui pengujian ini, peneliti dapat mengamati bagaimana model bertindak dalam kondisi dunia nyata, tanpa perlu memeriksa detail internalnya. Model berhasil mengklasifikasikan gambar daun *healthy*, *early blight*, dan *late blight* dengan tingkat akurasi yang dapat diandalkan. Untuk menganalisis kualitas hasil lebih mendalam, kami menggunakan matriks evaluasi berupa *precision* dan *recall curve*. *Precision* memberikan gambaran tentang seberapa akurat model dalam mengidentifikasi kelas positif, sedangkan *recall* mengukur seberapa baik model dalam mendeteksi seluruh sampel yang seharusnya positif. Dengan melihat kedua metrik ini, kami dapat memahami keseimbangan antara akurasi dan ketepatan identifikasi yang diberikan oleh model. Hasil dari pengujian dan evaluasi ini akan membantu dalam memvalidasi keefektifan model dalam deteksi penyakit daun kentang, serta memberikan wawasan yang berharga untuk pengembangan dan peningkatan model di masa mendatang.

3.1. Hasil Percobaan

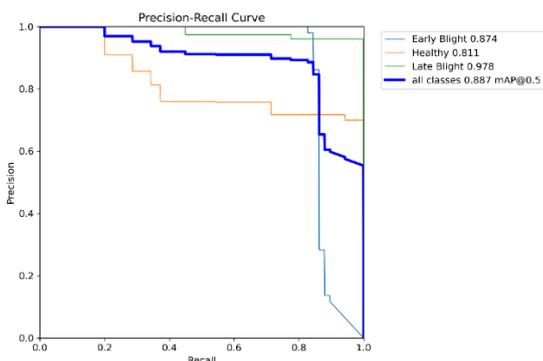
Tabel 1. Percobaan Pelatihan Model

case	Img size	batch	epoch	P&R	mAp
1	640	32	50	0.88	0.5
2	640	32	60	0.88	0.5
3	640	32	70	0.87	0.5
4	640	32	80	0.87	0.5
5	640	32	90	0.88	0.5

Pada tabel 1 menjelaskan hasil percobaan menunjukkan bahwa penggunaan *epoch* yang lebih banyak tidak selalu meningkatkan performa model secara signifikan. Dalam kasus ini, berbagai percobaan dilakukan dengan ukuran gambar 640, batch size 32, dan jumlah epoch yang berbeda, yaitu 50, 60, 70, 80, dan 90. Hasil evaluasi menggunakan *precision* and *recall* (*P&R*) dan *mean average precision* (*mAP*) menunjukkan bahwa nilai *P&R* tetap stabil di angka 0.88 pada epoch 50, 60, dan 90, namun sedikit menurun menjadi 0.87 pada *epoch* 70 dan 80. Nilai *mAP* juga konsisten di angka 0.5 untuk semua percobaan. Dari hasil ini, dapat disimpulkan bahwa penggunaan 50 *epoch* sudah memberikan hasil yang optimal dengan nilai *P&R* 0.88 dan *mAP* 0.5, sementara menambah jumlah *epoch* hanya meningkatkan waktu pelatihan tanpa memberikan peningkatan performa yang signifikan. Oleh karena itu, percobaan dengan 50 *epoch* adalah yang terbaik dalam konteks ini, karena memberikan keseimbangan optimal antara akurasi model dan efisiensi waktu.

3.2. Matrik Evaluasi

Pada gambar 8 menunjukkan diagram yang memperlihatkan hasil evaluasi menggunakan *precision* dan *recall* dari akurasi pelatihan, baik per kelas maupun untuk semua kelas secara keseluruhan. Dalam diagram tersebut, nilai akurasi masing-masing kelas ditampilkan dengan jelas, yaitu *early blight* dengan akurasi sebesar 87%, *healthy* dengan akurasi sebesar 81%, dan *late blight* dengan akurasi tertinggi yaitu 97%. Selain itu, diagram tersebut juga menampilkan akurasi gabungan untuk semua kelas yang mencapai 88%. Grafik ini memberikan visualisasi yang komprehensif mengenai kinerja model dalam mengklasifikasikan setiap kelas secara individual, serta secara keseluruhan, sehingga memudahkan analisis dan evaluasi terhadap efektivitas model yang telah dilatih.



Gambar 8. Precision-Recall Curve

3.3. Pengujian Model

Tabel 2. Pengujian

Daun	Gambar asli	Prediksi	Keterangan
healthy			Benar
Early blight			Benar
Late blight			Benar

Pada tabel 2, terdapat hasil pengujian yang dilakukan menggunakan tiga jenis daun kentang yang mewakili masing-masing kelas. Hasil dari pengujian ini memberikan gambaran yang jelas tentang kemampuan sistem dalam mendeteksi penyakit pada daun kentang. Setiap jenis daun kentang yang diuji berhasil dideteksi dengan akurasi yang tinggi, sesuai dengan kondisi daun tersebut. Ini mengonfirmasi kinerja sistem yang optimal dalam mengidentifikasi daun kentang yang sehat serta daun yang terinfeksi dengan penyakit *early blight* dan *late blight*. Melalui analisis yang mendalam terhadap hasil pengujian ini, dapat disimpulkan bahwa sistem telah berhasil dalam menjalankan tugasnya dengan baik dan memberikan hasil yang dapat diandalkan dalam mendeteksi penyakit pada tanaman kentang. Hasil ini menjadi landasan yang kuat untuk mengonfirmasi validitas dan kehandalan sistem yang dikembangkan. Selain itu, hasil pengujian ini juga memberikan wawasan yang berharga untuk pengembangan dan peningkatan sistem di masa depan, dengan fokus pada peningkatan akurasi dan efisiensi deteksi penyakit pada tanaman kentang secara keseluruhan.

Oleh karena itu pada penelitian ini berdasarkan hasil percobaan menunjukkan bahwa peningkatan jumlah *epoch* tidak selalu meningkatkan performa model secara signifikan. Pengujian menggunakan tiga jenis daun kentang menunjukkan bahwa sistem berhasil mendeteksi penyakit dengan akurasi tinggi, mengonfirmasi kinerja sistem yang optimal dan mendukung validitas serta keandalan sistem yang dikembangkan, memberikan wawasan untuk peningkatan di masa depan.

4. DISKUSI

Sistem deteksi penyakit daun kentang ini menggunakan algoritma *YOLOv8*. Algoritma ini merupakan suatu metode dalam bidang deteksi objek yang menggunakan teknik *deep learning*. Pendekatan ini adalah pengembangan dari model *YOLO* yang telah diperkenalkan sebelumnya. *YOLOv8* mengintegrasikan berbagai teknik dan pembaruan guna meningkatkan akurasi dan kinerja deteksi objek [25]. Pelatihan model menggunakan resolusi gambar 640x640 dan 50 *epoch* pelatihan, yang akhirnya

menghasilkan tingkat akurasi model sebesar 88%. Penelitian ini merupakan pengembangan dari studi sebelumnya yang menggunakan YOLOv5 sebagai modelnya yang digunakan untuk mengembangkan sistem deteksi penyakit tanaman [14]. Namun, berbeda dengan penelitian sebelumnya yang dilakukan oleh [16], sistem deteksi yang dihasilkan dari penelitian ini menunjukkan peningkatan kinerja yang signifikan, mencerminkan kemajuan dalam pemodelan deteksi penyakit daun kentang. Pendekatan yang digunakan dalam penelitian ini mampu memanfaatkan sejumlah besar data dengan menggunakan *dataset* yang terdiri dari 750 gambar, mencakup daun *healthy*, *early blight*, dan *late blight*. Dengan memperluas cakupan *dataset*, penelitian ini memberikan kerangka kerja yang lebih komprehensif dan akurat dalam melatih model deteksi penyakit, memungkinkan identifikasi yang lebih tepat dan responsif terhadap keberadaan penyakit pada tanaman kentang.

5. KESIMPULAN

Setelah melakukan penelitian komprehensif terhadap masalah deteksi penyakit daun kentang di pertanian Bibit Sida Kangen Kalibening, Banjarnegara, dapat disimpulkan bahwa penelitian ini berhasil mengembangkan sistem deteksi penyakit daun kentang dengan sukses. Tujuan dari penelitian ini, yaitu menciptakan sistem deteksi penyakit daun kentang menggunakan YOLOv8, berhasil tercapai dengan baik. Hasil pengujian sistem menunjukkan tingkat keberhasilan yang sangat baik dalam mengenali penyakit-penyakit pada daun kentang, termasuk *early blight*, *late blight*, dan kondisi daun yang sehat. Evaluasi model menggunakan metrik *precision* dan *recall* juga menghasilkan hasil yang mengesankan, dengan akurasi untuk *early blight* sebesar 87%, untuk daun *healthy* sebesar 81%, dan untuk *late blight* sebesar 97%. Secara keseluruhan, sistem berhasil mencapai tingkat akurasi sebesar 88%. Prestasi ini menegaskan bahwa sistem deteksi yang dikembangkan mampu dengan sangat akurat membedakan antara daun kentang yang sehat dan yang terinfeksi penyakit. Selain menjadi solusi teknologi yang efektif, penelitian ini juga bertujuan untuk meningkatkan produktivitas dalam hasil pertanian. Oleh karena itu, penelitian ini memberikan kontribusi yang signifikan dalam meningkatkan produktivitas di pertanian Bibit Sida Kangen Kalibening, Banjarnegara, sekaligus membantu petani dalam mendeteksi penyakit daun kentang secara cepat dan akurat.

DAFTAR PUSTAKA

- [1] R. D. Puspitasari, "Pertanian Berkelanjutan Berbasis Revolusi Industri 4.0," *J. Layanan Masy. (Journal Public Serv.*, vol. 3, no. 1, p. 26, 2020, doi: 10.20473/jlm.v3i1.2019.26-28.
- [2] M. Azis and E. A. Suryana, "KOMPARASI DAN IMPLEMENTASI KEBIJAKAN DIGITALISASI PERTANIAN: PELUANG DAN TANTANGAN," *Risal. Kebijak. Pertan. DAN Lingkung. Rumusan Kaji. Strateg. Bid. Pertan. dan Lingkung.*, vol. 10, no. 3, pp. 179–198, Dec. 2023, doi: 10.29244/jkebijakan.v10i3.51083.
- [3] M. Wali *et al.*, *Penerapan & Implementasi Big Data di Berbagai Sektor (Pembangunan Berkelanjutan Era Industri 4.0 dan Society 5.0)*. 2023.
- [4] N. M. S. Ayomi, B. M. Setiawan, and W. Roessali, "ANALISIS FLUKTUASI DAN ELASTISITAS TRANSMISI HARGA KENTANG DI KABUPATEN MAGELANG," *J. Litbang Provinsi Jawa Teng.*, vol. 18, no. 2, pp. 159–166, Dec. 2020, doi: 10.36762/jurnaljateng.v18i2.828.
- [5] A. M. Lesmana, R. P. Fadhillah, and C. Rozikin, "Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN)," *J. Sains dan Inform.*, vol. 8, no. 1, pp. 21–30, Jun. 2022, doi: 10.34128/jsi.v8i1.377.
- [6] L. Aumatullah, I. Ein, and M. M. Santoni, "Identifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur dan Warna Dengan Menggunakan Metode K-Nearest Neighbor," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, no. April, pp. 783–791, 2021.
- [7] Kompas, "Penurunan Produksi Kentang di Dieng." [Online]. Available: <https://www.kompasiana.com/15lusiariaxiimipa25402/6381a6a934b898219a257c54/penerunan-produksi-kentang-di-dieng.2022>
- [8] M. Kevin Santosa, M. Hanindia Prami Swari, and A. Nugroho Sihananto, "IMPLEMENTASI ARSITEKTUR ALEXNET DAN RESNET34 PADA KLASIFIKASI CITRA PENYAKIT DAUN KENTANG MENGGUNAKAN TRANSFER LEARNING," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 5, pp. 3293–3301, Jan. 2024, doi: 10.36040/jati.v7i5.7337.
- [9] K. I. Nauval and S. Lestari, "Implementasi Deteksi Objek Penyakit Daun Kentang dengan Metode Convolutional Neural Network," *J. Apl. Teknol. Inf. dan Manaj.*, vol. 3, no. 2, pp. 136–149, Nov. 2022, doi: 10.31102/jatim.v3i2.1576.
- [10] M. Masparudin, I. Fitri, and S. Sumijan, "Development of Apple Fruit Classification System using Convolutional Neural Network (CNN) MobileNet Architecture on Android Platform," *SISTEMASI*, vol. 13, no. 1, p. 230, Jan. 2024, doi: 10.32520/stmsi.v13i1.3533.
- [11] S. Abbasi, H. Abdi, and A. Ahmadi, "A Face-

- Mask Detection Approach based on YOLO Applied for a New Collected Dataset,” in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, IEEE, Mar. 2021, pp. 1–6. doi: 10.1109/CSICC52343.2021.9420599.
- [12] 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, Feb. 2024, doi: 10.28989/avitec.v6i1.2067.
- [13] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of Yolo Algorithm Developments,” *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022, doi: 10.1016/j.procs.2022.01.135.
- [14] M. P. Mathew and T. Y. Mahesh, “Leaf-based disease detection in bell pepper plant using YOLO v5,” *Signal, Image Video Process.*, vol. 16, no. 3, pp. 841–847, Apr. 2022, doi: 10.1007/s11760-021-02024-y.
- [15] R. Rajamohanam and B. C. Latha, “An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset,” *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 6, pp. 12033–12038, Dec. 2023, doi: 10.48084/etasr.6377.
- [16] M. K. R. Asif, M. A. Rahman, and M. H. Hena, “CNN based Disease Detection Approach on Potato Leaves,” in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, Dec. 2020, pp. 428–432. doi: 10.1109/ICISS49785.2020.9316021.
- [17] R. Raj, S. S. Nagaraj, S. Ritesh, T. A. Thushar, and V. M. Aparanji, “Fruit Classification Comparison Based on CNN and YOLO,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1187, no. 1, p. 012031, Sep. 2021, doi: 10.1088/1757-899X/1187/1/012031.
- [18] S. Shahriar, S. Allana, S. M. Hazratifard, and R. Dara, “A Survey of Privacy Risks and Mitigation Strategies in the Artificial Intelligence Life Cycle,” *IEEE Access*, vol. 11, pp. 61829–61854, 2023, doi: 10.1109/ACCESS.2023.3287195.
- [19] Suhartono, A. Ardilla, and Satria Gunawan, “Detection of Vehicle Type And License Plate With Convolutional Neural Network Model YOLOv7,” *J. Tek. Inform.*, vol. x, No. y, no. 2, pp. x-y, 2023.
- [20] D. S. Park *et al.*, “Improved Noisy Student Training for Automatic Speech Recognition,” in *Interspeech 2020*, ISCA: ISCA, Oct. 2020, pp. 2817–2821. doi: 10.21437/Interspeech.2020-1470.
- [21] S. G. E. Brucal, L. C. M. de Jesus, S. R. Peruda, L. A. Samaniego, and E. D. Yong, “Development of Tomato Leaf Disease Detection using YoloV8 Model via RoboFlow 2.0,” in *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)*, IEEE, Oct. 2023, pp. 692–694. doi: 10.1109/GCCE59613.2023.10315251.
- [22] R. Gelar Guntara, “Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 1, pp. 55–60, Feb. 2023, doi: 10.47233/jteksis.v5i1.750.
- [23] S. Khalid, M. Goldenberg, T. Grantcharov, B. Taati, and F. Rudzicz, “Evaluation of Deep Learning Models for Identifying Surgical Actions and Measuring Performance,” *JAMA Netw. Open*, vol. 3, no. 3, p. e201664, Mar. 2020, doi: 10.1001/jamanetworkopen.2020.1664.
- [24] A. Polyvyanyy, A. Solti, M. Weidlich, C. Di Ciccio, and J. Mendling, “Monotone Precision and Recall Measures for Comparing Executions and Specifications of Dynamic Systems,” *ACM Trans. Softw. Eng. Methodol.*, vol. 29, no. 3, pp. 1–41, Jul. 2020, doi: 10.1145/3387909.
- [25] A. Aboah, B. Wang, U. Bagci, and Y. Adu-Gyamfi, “Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8,” in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2023, pp. 5350–5358. doi: 10.1109/CVPRW59228.2023.00564.