

Detection of Potato Leaf Diseases Using the YOLOv8 Method

Deteksi Penyakit Daun Tanaman Kentang Menggunakan Metode YOLOv8

Fadel Raditya Latief¹, Budi Santosa²

^{1,2} Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

¹Fadelatief@gmail.com, ²Budisantosa@upnyk.ac.id,

*: Penulis korespondensi (corresponding author)

Informasi Artikel

Received: November 2025

Revised: December 2025

Accepted: January 2026

Published: February 2026

Abstract

The cultivation of potatoes faces significant challenges due to leaf diseases such as late blight and early blight, which adversely affect productivity. This study aims to evaluate the YOLOv8 model in detecting potato leaf diseases, particularly under varying object distances and different light intensity conditions. The research dataset consists of primary images collected from potato farms in Dieng and secondary data sourced from Kaggle. The model was tested at various distances (10 cm, 25 cm, 50 cm, and 75 cm) as well as under bright and dim lighting conditions. The test results showed that the model's performance tends to decline as the distance between the object and the camera increases, with the best results observed at a distance of 10 cm. For variations in light intensity, the model performed better under bright conditions compared to dim lighting, although the difference was not significant. The YOLOv8 model achieved an mAP50 score of 0.994, precision of 0.994, and recall of 0.997 during training with 50 epochs. These findings demonstrate the potential of YOLOv8 for automatic detection of potato leaf diseases with strong performance at close distances and across varying lighting conditions.

Abstrak

Keywords: ; Late Blight, Early Blight, YOLOv8, Leaf Disease Detection, Light Intensity, Object Distance

Kata kunci: Late Blight, Early Blight, YOLOv8, Deteksi Penyakit Daun, Intensitas Cahaya, Jarak Objek

Budidaya kentang menghadapi tantangan signifikan akibat penyakit daun seperti late blight dan early blight, yang memengaruhi produktivitas. Penelitian ini bertujuan untuk menguji model YOLOv8 dalam mendeteksi penyakit daun tanaman kentang, khususnya pada kondisi jarak yang bervariasi dan intensitas cahaya yang berbeda. Dataset penelitian terdiri dari citra primer yang dikumpulkan dari kebun kentang di Dieng dan data sekunder dari Kaggle. Model diuji dengan

berbagai jarak (10 cm, 25 cm, 50 cm, dan 75 cm) serta intensitas cahaya terang dan redup. Hasil pengujian menunjukkan bahwa performa model cenderung menurun seiring bertambahnya jarak objek dengan kamera, dengan hasil terbaik pada jarak 10 cm. Pada variasi

1. Pendahuluan

Budidaya tanaman kentang menghadapi tantangan yang signifikan, terutama dalam mengelola penyakit yang dapat secara serius memengaruhi hasil panen dan kualitasnya. Di antara penyakit tersebut, *late blight* (hawar daun) dan *early blight* (bercak kering) merupakan ancaman utama bagi tanaman kentang[4]. *Late blight* yang disebabkan oleh *Phytophthora infestans* ditandai dengan munculnya bercak nekrotik pada tepi dan ujung daun yang dapat menyebar dengan cepat dalam kondisi suhu sedang dan kelembapan tinggi. Sementara itu, *early blight* ditandai dengan bercak gelap konsentris yang secara bertahap membesar dan menyebabkan kerusakan pada daun[9].

Identifikasi penyakit secara tradisional masih sangat bergantung pada inspeksi manual oleh petani, yang memerlukan waktu lama, melelahkan, dan rawan kesalahan akibat banyaknya jumlah daun yang harus diperiksa[10]. Untuk mengatasi keterbatasan ini, kemajuan teknologi pembelajaran mesin, khususnya dalam deteksi dan klasifikasi citra, menawarkan solusi yang menjanjikan. Teknik seperti *Convolutional Neural Networks* (CNN), *Region-based CNN* (R-CNN), *Faster R-CNN*, dan *You Only Look Once* (YOLO) telah menunjukkan potensi dalam mengotomatisasi deteksi penyakit dengan akurasi tinggi[1].

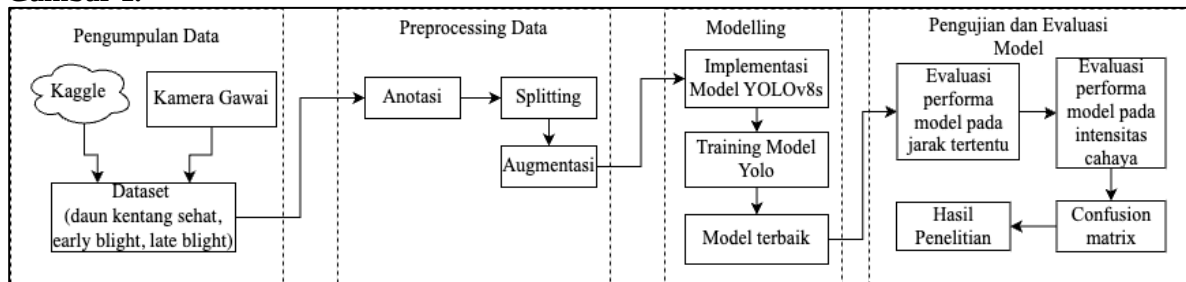
Beberapa penelitian sebelumnya telah mengeksplorasi berbagai arsitektur untuk meningkatkan kemampuan deteksi penyakit. Penelitian yang menggunakan ResNet-34, misalnya, mencapai akurasi klasifikasi sebesar 97% untuk penyakit daun kentang, tetapi menghadapi tantangan dalam mendeteksi objek yang lebih kecil pada kondisi lapangan. Penelitian lain yang memanfaatkan arsitektur MobileNet-V2, VGG16, dan ConvNeXtBase menunjukkan hasil yang baik di bawah pencahayaan terkontrol, namun terbatas oleh ukuran data validasi serta performa yang menurun pada kondisi pencahayaan rendah.

Meskipun penelitian sebelumnya telah menghasilkan hasil yang baik, masih terdapat kesenjangan dalam penerapan model-model tersebut di lingkungan nyata, terutama dalam mendeteksi penyakit dari jarak yang bervariasi dan di bawah kondisi pencahayaan yang berbeda. Untuk mengatasi tantangan ini, penelitian ini menggunakan model YOLOv8 yang dikenal memiliki keunggulan dalam mendeteksi dan mengklasifikasikan objek dengan akurasi tinggi, termasuk objek berskala kecil. Penelitian ini bertujuan untuk meningkatkan akurasi dan efisiensi deteksi penyakit daun kentang, sekaligus mengatasi keterbatasan pendekatan sebelumnya, seperti pengujian pada berbagai jarak, skala objek, dan variasi kondisi lingkungan.

2. Metodologi Penelitian

Metodologi penelitian menjelaskan rangkaian tahapan yang dilakukan selama proses penelitian. Di dalamnya mencakup seluruh prosedur yang diterapkan untuk merancang dan

mengembangkan sistem. Tahapan-tahapan tersebut dapat dilihat pada ilustrasi yang disajikan dalam Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Pengumpulan Data

Proses ini merupakan tahap pengumpulan data yang diperlukan untuk mendukung kelancaran penelitian. Data yang digunakan terdiri dari data primer dan data sekunder. Data primer dikumpulkan menggunakan kamera gawai di Perkebunan Kentang milik Pak Ghodin, yang terletak di Patakbanteng, Dieng. Sebanyak 450 citra daun kentang diperoleh, yang terdiri dari 350 citra daun kentang sehat, 50 citra daun dengan penyakit *early blight*, dan 50 citra daun dengan penyakit *late blight*.

Data sekunder diperoleh dari situs Kaggle melalui tautan <https://www.kaggle.com/datasets/emmarex/plantdisease/data>, dengan total 1152 citra daun kentang. Dataset ini mencakup 152 citra daun kentang sehat, 500 citra daun dengan penyakit *early blight*, dan 500 citra daun dengan penyakit *late blight*. Setelah data primer dan sekunder terkumpul, kedua sumber data tersebut digabungkan untuk membentuk dataset akhir.

2.2. Data Preprocessing

Tahapan preprocessing dilakukan untuk mengubah citra dari citra mentah menjadi citra yang siap digunakan untuk proses training. Tahapan preprocessing citra yang dilakukan pada penelitian ini diantaranya :

a) Annotating Data

Untuk mendeteksi objek, perlu menggunakan data yang telah diberi anotasi. Anotasi ini mencakup pemberian deskripsi objek dalam gambar serta pemberian bounding box pada gambar terkait. Bounding box adalah kotak persegi panjang yang menandai atau mengelilingi objek dalam citra, merepresentasikan lokasi dan ukuran objek yang akan dideteksi atau diklasifikasikan oleh model dalam tugas seperti deteksi objek. Dalam penelitian ini, roboflow digunakan sebagai alat untuk anotasi gambar. Setiap gambar daun kentang diberi bounding box dan diberi label sesuai dengan kelasnya: daun sehat, *early blight*, atau *late blight*.

b) Splitting Data

Setelah data selesai dianotasi, langkah selanjutnya adalah membagi dataset ke dalam tiga bagian utama: data pelatihan (*training*), data validasi (*validation*), dan data pengujian (*testing*). Pembagian dataset ini bertujuan untuk memastikan setiap bagian dapat digunakan secara efektif dalam mendukung proses pengembangan model, meliputi pelatihan, validasi, dan pengujian.

Dalam penelitian ini, dataset dibagi dengan proporsi 72% untuk data pelatihan, 19% untuk data validasi, dan 9% untuk data pengujian. Proses pembagian dilakukan secara acak untuk setiap kelas, dengan tujuan menjaga distribusi data yang seimbang di

seluruh kategori. Pembagian dataset dilakukan menggunakan platform Roboflow, yang memungkinkan pengaturan proporsi dataset secara intuitif. Pengguna hanya perlu menyesuaikan pengaturan proporsi dengan menggunakan fitur penggeser (*slider*) untuk menetapkan jumlah data pelatihan, validasi, dan pengujian sesuai kebutuhan.

c) *Augmentasi Data*

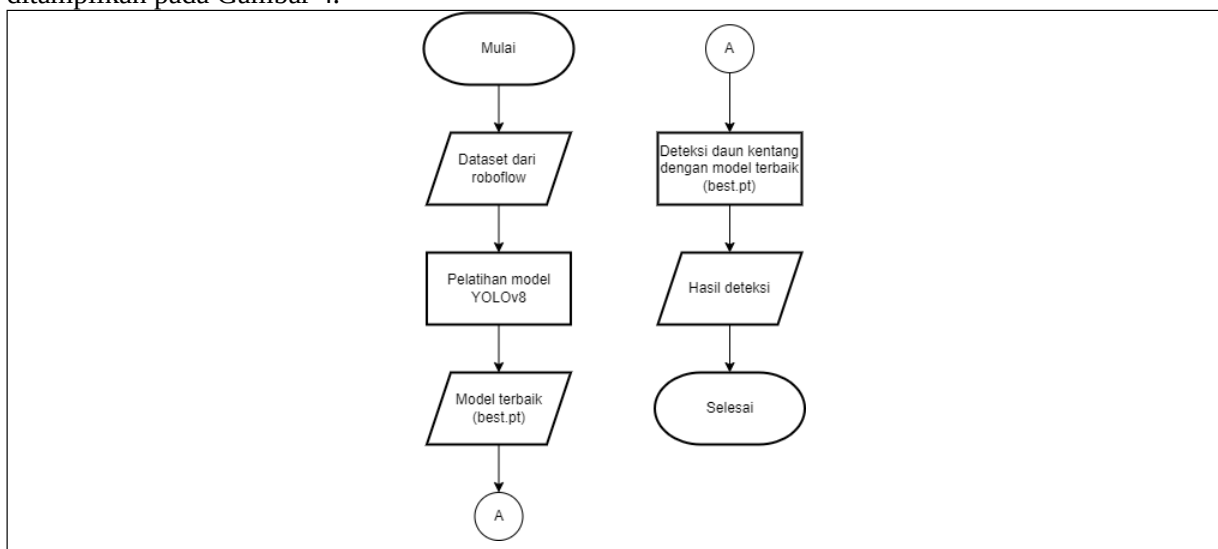
Proses ini bertujuan untuk meningkatkan keragaman data citra yang digunakan selama proses pelatihan. Dengan memperluas variasi data, model diharapkan mampu mengenali pola-pola yang lebih beragam, sehingga memperkuat kemampuan generalisasi terhadap data baru yang belum pernah ditemui sebelumnya. Teknik augmentasi yang diterapkan meliputi langkah-langkah berikut:

Tabel 1. Teknik Augmentasi

No.	Parameter	Nilai	Keterangan
1	Resize	640 x 640	mengubah ukuran gambar agar sesuai dengan input model, membantu model menangani gambar dengan berbagai dimensi dan resolusi
2	Rotation	15° ke kiri dan 15° ke kanan	memutar gambar sebesar 15 derajat ke kiri atau kanan untuk meningkatkan variasi dan ketahanan model terhadap perubahan orientasi.
3	Flip	Vertikal	membalik gambar secara vertikal (atas-bawah)
4	90° rotation	Clockwise, counter clockwise, upside down	memutar gambar 90 derajat searah jarum jam, berlawanan arah jarum jam, atau terbalik untuk meningkatkan variasi dan ketahanan model

2.3. Penerapan Model YOLOv8

Proses implementasi model YOLOv8 dimulai dengan pengolahan dan pelatihan dataset yang telah dipersiapkan untuk membangun model deteksi. Tahapan awal mencakup pengunduhan *library ultralytics* sebagai dasar pemodelan. Model yang digunakan dalam penelitian ini adalah YOLOv8n (*nano*). Alur lengkap proses implementasi model YOLOv8 dapat dilihat pada diagram alir yang ditampilkan pada Gambar 4.



Gambar 2 Flowchat Implementasi

Dataset yang telah melalui tahap pre-processing menggunakan Roboflow akan digunakan

untuk melatih model YOLOv8n. Model YOLOv8n memiliki 249 lapisan dengan total 2.690.793 parameter dan 2.690.777 gradien. Berikut ini adalah arsitektur YOLOv8n yang diterapkan dalam proses pelatihan.

Pada gambar berikut, dapat dilihat bahwa proses yang dilakukan oleh YOLOv8 terdiri dari 22 langkah. Pada baris pertama, terdapat lapisan konvolusi yang melibatkan 464 parameter yang dilatih. Argumen [3,16,3,2] menunjukkan bahwa terdapat 3 channel input (RGB), 16 channel output (feature map), ukuran kernel 3, dan stride sebesar 2. Setelah proses konvolusi, langkah-langkah selanjutnya meliputi C2f, SPPF, Upsample, concat, dan deteksi.

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	7360	ultralytics.nn.modules.block.C2f	[32, 32, 1, True]
3	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
4	-1	2	49664	ultralytics.nn.modules.block.C2f	[64, 64, 2, True]
5	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
6	-1	2	197632	ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	460288	ultralytics.nn.modules.block.C2f	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	1	148224	ultralytics.nn.modules.block.C2f	[384, 128, 1]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	1	37248	ultralytics.nn.modules.block.C2f	[192, 64, 1]
16	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	1	123648	ultralytics.nn.modules.block.C2f	[192, 128, 1]
19	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	1	493056	ultralytics.nn.modules.block.C2f	[384, 256, 1]
22	[15, 18, 21]	1	431257	ultralytics.nn.modules.head.Detect	[3, [64, 128, 256]]

Model summary: 249 layers, 2,690,793 parameters, 2,690,777 gradients, 6.9 GFLOPs

Gambar 3 Implementasi arsitektur YOLOv8n

Setiap tahapan memiliki rincian sebagai berikut :

a) Convolution

Proses convolution menggunakan operasi konvolusi 2D(Conv2d), setelah proses convolution aktivasi yang dihasilkan kemudian dinormalisasi menggunakan Batchnorm2D, yang menghitung rata-rata dan deviasi standar di seluruh batch untuk menstabilkan distribusi aktivasi. Setelah itu, fungsi aktivasi SiLU diterapkan.

b) C2f

C2f merupakan tahapan yang berisi 1 layer convolutional yang kemudian di split, masuk ke bottleneck, lalu masuk ke concat dan kembali ke dalam convolution layer.

c) SPPF

Pada tahap SPPF terdiri dari convolution layer lalu dilakukan maxPool2d sebanyak tiga kali, masuk ke tahap concat dan dikembalikan ke layer convolution.

d) Detect

Pada proses detect dibagi menjadi dua convolution layer dan dimasukkan ke dalam conv2d yang akan menghasilkan Bbox Loss dan Cls Loss.

2.4. Proses Training dan Klasifikasi YOLOv8

Pada penelitian ini, proses pelatihan model YOLOv8 dimulai dengan mengunduh pustaka *Ultralytics* sebagai platform utama untuk implementasi YOLOv8, diikuti dengan pemanggilan model YOLOv8n (*nano*), yang dikenal dengan efisiensi dan ukurannya yang ringan dalam mendeteksi objek. Dataset yang telah dianotasi sebelumnya diunduh dan dipersiapkan sebagai data pelatihan, dengan mencakup tiga kelas utama, yaitu daun kentang sehat, daun yang terinfeksi penyakit *early blight*, dan

daun yang terinfeksi penyakit *late blight*.

Hyperparameter pada proses pelatihan menggunakan konfigurasi default YOLOv8 untuk menjaga keseimbangan antara akurasi dan efisiensi. Jumlah epoch pelatihan ditetapkan sebanyak 50, dengan ukuran gambar yang digunakan adalah 640 x 640 piksel untuk setiap citra masukan. Ukuran ini memastikan resolusi yang cukup tinggi untuk menangkap detail penting dari citra daun kentang, namun tetap efisien untuk diproses oleh model YOLOv8n.

Selama pelatihan, model dilatih untuk mengenali dan mengklasifikasikan daun ke dalam tiga kelas tersebut dengan memperhatikan berbagai metrik kinerja, seperti *precision*, *recall*, mAP50, dan mAP50-95. Model ini dirancang untuk mengoptimalkan deteksi objek dengan mempertimbangkan kemampuan deteksi terhadap objek berukuran kecil dan pencahayaan bervariasi, yang sesuai dengan kondisi sebenarnya di lapangan.

2.5. Evaluasi Model Confusion Matrix

Confusion Matrix adalah alat evaluasi dalam Machine Learning yang menggambarkan performa suatu model klasifikasi dengan menyajikan hasil prediksi data yang benar dan yang salah. Terdiri dari empat nilai yang mewakili berbagai kombinasi hasil prediksi dan nilai sebenarnya, yaitu True Positive, False Positive, True Negative, dan False Negative. Sebagai ilustrasi, dalam sebuah skenario dengan dua kelas, P dan N, Evaluasi yang akan dihitung yaitu *accuracy*, adalah ukuran yang menggambarkan seberapa akurat sistem dalam mengklasifikasikan data dengan benar.

True Positive (TP) = Data positif yang terdeteksi benar, True

Negative (TN) = Data negative yang terdeteksi benar, False

Positive (FP) = Data negative yang terdeteksi positif, False

Negative (FN) = Data positif yang terdeteksi negative.

Tabel 2 Pengujian Model dengan Confusion Matrix

		True Class			Recall
		Healthy	Early Blight	Late Blight	
Predicted Value	Healthy				$TP = \frac{TP}{TP + FN}$
	Early Blight				
	Late Blight				
Precision		$TP = \frac{TP}{TP + FP}$			

2.6. Pengujian Model

Pengujian terhadap variasi jarak dilakukan untuk mengevaluasi performa model YOLOv8 berdasarkan hasil pelatihan yang telah dilakukan. Pengujian dirancang secara sistematis sesuai dengan tujuan evaluasi yang ingin dicapai terhadap model yang dikembangkan. Dalam penelitian ini, pengujian akan dilakukan dengan menggunakan beberapa jarak yang berbeda. Adapun rancangan kombinasi pengujiannya adalah sebagai berikut:

Tabel 3 Indikator keberhasilan pengujian jarak

Jarak	Precision	Recall	mAP50	mAP50-95
10 cm	0.92	0.89	0.94	0.76
25 cm	0.88	0.85	0.90	0.71
50 cm	0.81	0.78	0.83	0.62
75 cm	0.73	0.70	0.75	0.53

Menguji pengaruh variasi intensitas cahaya terhadap kinerja model dalam mendeteksi penyakit pada daun kentang. Pengujian ini bertujuan untuk menganalisis bagaimana perubahan intensitas cahaya memengaruhi kualitas citra yang dihasilkan dan dampaknya terhadap akurasi serta keandalan model dalam mengenali gejala penyakit. Indikator yang digunakan dalam pengujian intensitas cahaya adalah sebagai berikut:

Tabel 4 Indikator Pengujian Intensitas Cahaya

Intensitas	Precision	Recall	mAP50	mAP50-95
Terang	0.90	0.85	0.75	0.55
Redup	0.85	0.80	0.70	0.53

3. Hasil dan Pembahasan

3.1. Pengumpulan Data

Data sekunder dan data primer yang telah terkumpul melalui proses pengumpulan data selanjutnya digabungkan untuk membentuk satu set data yang komprehensif. Adapun rincian data yang diperoleh setelah proses integrasi adalah sebagai berikut:

Tabel 5 Penggabungan Data Sekunder dan Data Primer

Label	Data Primer	Data Sekunder	Jumlah
Healthy	350	152	502
Early Blight	50	500	550
Late Blight	50	500	550

Dataset primer dari area pertanian Dieng digunakan untuk menguji pengaruh jarak objek terhadap akurasi deteksi penyakit daun kentang. Variasi jarak menghasilkan ukuran objek yang berbeda dalam citra, memungkinkan evaluasi performa model. Berikut contoh dataset yang diuji:

Tabel 6 Contoh Gambar Dataset Pengujian Jarak

Jarak	Citra Dataset
10 cm	

25 cm	
50 cm	
75 cm	

Dataset yang digunakan untuk pengujian variasi intensitas cahaya diambil dari area pertanian kentang di Dieng. Dua kategori pencahayaan, yaitu terang dan redup, merepresentasikan variasi intensitas cahaya yang terjadi di lapangan. Contoh dataset pengujian intensitas cahaya dapat dilihat sebagai berikut :

Tabel 7 Contoh Gambar Dataset Pengujian Intensitas Cahaya

Kategori	Citra Dataset	Tingkat Kecerahan
Terang		151 - 200
Redup		51 - 100

3.2. Data Preprocessing

Data yang telah dikumpulkan melalui proses *preprocessing* meliputi anotasi data, pembagian dataset (*splitting*), dan augmentasi. Anotasi dilakukan untuk memberikan label pada citra, sementara pembagian data dilakukan untuk memisahkan data pelatihan, validasi, dan pengujian. Tahap akhir adalah augmentasi, yang bertujuan meningkatkan variasi data dengan menerapkan transformasi. Hasil akhir dari proses ini disajikan dalam tabel berikut.

Tabel 8 Persebaran Data Setelah Proses Data Preprocessing

Label	Data Training	Data Validation	Data Test
Healthy	1146	103	50
Early Blight	1149	99	50
Late Blight	1149	101	50

Tabel 8 menunjukkan distribusi data setelah proses *preprocessing*, yang mencakup anotasi, pembagian dataset, dan augmentasi. Data dibagi menjadi tiga kelas: *Healthy*, *Early Blight*, dan *Late Blight*, masing-masing dengan total data *training* menjadi 3.444 citra(88%), data *validation* 302(8%), dan data *test* sebanyak 150(4%). Pembagian ini dilakukan secara proporsional untuk memastikan representasi yang seimbang pada setiap kelas dalam pelatihan, validasi, dan pengujian model.

3.3. Pembuatan Model

Model yang digunakan dalam penelitian ini dikembangkan dengan menggunakan bahasa pemrograman Python. Proses pengembangan dimulai dengan menginstal dan mengimpor pustaka-pustaka yang relevan. Selanjutnya, dataset yang telah melalui tahap *preprocessing* diunduh untuk digunakan dalam pelatihan model. Tahapan berikutnya adalah melakukan inisialisasi arsitektur

YOLOv8n sebagai kerangka utama untuk proses pelatihan.

Setelah model YOLOv8 diinisialisasi menggunakan bobot pramuat `yolo8n.pt`, proses pelatihan dilakukan untuk menyesuaikan model dengan dataset. Pelatihan ini melibatkan pengaturan hyperparameter utama, seperti jumlah epoch dan ukuran gambar (*image size*), yang bertujuan untuk mengoptimalkan performa model dengan meminimalkan *loss*. File konfigurasi dataset `data.yaml` digunakan untuk memandu proses pelatihan, yang mencakup informasi dataset dan pengaturan pelatihan.

Algoritma Training Model

```
# Load a model
model = YOLO('yolo8n.pt') # load a pretrained model

# Train the model with 2 GPUs
results = model.train(data='/kaggle/input/dataset-gabungan/data.yaml', epochs=50,
imgsz=640, device=0)
```

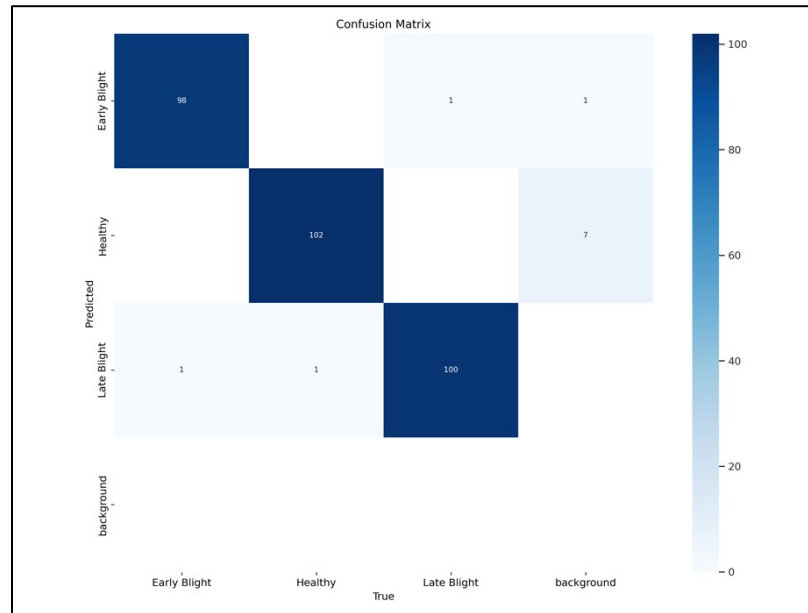
Kode ini memuat model YOLOv8n pralatih dari `yolo8n.pt` dan melatihnya menggunakan dataset yang diatur dalam `data.yaml`. Pelatihan dilakukan selama 50 epoch dengan resolusi gambar 640 piksel dan menggunakan GPU (`device=0`) untuk mempercepat proses. Parameter ini dirancang untuk mengoptimalkan deteksi penyakit pada daun kentang. Berdasarkan Algoritma 4.4 di atas, program melatih model YOLOv8 menggunakan data yang telah diinputkan. Proses pelatihan ini menghasilkan empat nilai utama, yaitu `box_loss`, `cls_loss`, `mAP50`, dan `mAP50-95`, yang disajikan dalam tabel berikut.

Tabel 9 Hasil Pelatihan Model

Epoch	Box_loss	Cls_loss	P	R	mAP50	mAP50-95
1	0.7324	1.553	0.93	0.84	0.96	0.803
2	0.6581	0.8636	0.537	0.667	0.644	0.379
3	0.6533	0.7274	0.946	0.972	0.992	0.846
...	...	...			...	...
48	0.1778	0.1338	0.994	0.997	0.994	0.961
49	0.1744	0.1254	0.993	0.997	0.993	0.955
50	0.169	0.1231	0.994	0.997	0.994	0.959

3.4. Evaluasi Model

Evaluasi model dilakukan untuk menilai efektivitas dan kinerja model dalam mendeteksi penyakit pada daun kentang, dengan menggunakan data validasi yang belum terlibat dalam proses pelatihan. Pengujian ini bertujuan untuk memastikan kemampuan model dalam menghasilkan deteksi yang akurat dan relevan. Berdasarkan hasil evaluasi, model menunjukkan performa yang sangat baik, dengan nilai mean Average Precision (mAP) sebesar 0.994, yang mengindikasikan tingkat ketepatan deteksi yang sangat tinggi. Selain itu, model juga menghasilkan mAP pada rentang 50-95% (mAP50-95) sebesar 0.962, yang menandakan kestabilan deteksi pada berbagai tingkat kesulitan objek yang terdeteksi. Precision yang dicapai oleh model adalah 0.994, yang menunjukkan bahwa hampir semua deteksi yang dilakukan oleh model adalah benar, dengan sedikit kesalahan deteksi atau false positive. Sementara itu, nilai recall yang mencapai 0.997 mengindikasikan bahwa model sangat sensitif dalam mendeteksi objek, dengan hanya sedikit objek yang terlewatkan (false negative). Secara keseluruhan, hasil evaluasi ini mengindikasikan bahwa model memiliki akurasi yang luar biasa serta kemampuan yang sangat baik dalam mengenali dan mengklasifikasikan penyakit daun kentang dengan tingkat kesalahan yang sangat rendah. Confussion Matrix pada evaluasi model ini ditampilkan pada gambar di bawah ini.



Gambar 4 Confussion Matrix Hasil Evaluasi Model

Berdasarkan confusion matrix pada gambar 4 dapat disimpulkan evaluasi model YOLOv8 menunjukkan performa yang baik dan mendapatkan nilai terbaik saat mendeteksi penyakit daun kentang *late blight*. Masih terjadi beberapa kesalahan deteksi karena masih mendeteksi objek menjadi *background*.

3.5. Pengujian Model

Pengujian model pada penelitian ini menggunakan dua pengujian yaitu pengujian jarak dan pengujian intensitas cahaya.

a. Pengujian pada berbagai jarak yang berbeda

Untuk menguji performa YOLOv8 dalam mendeteksi objek berukuran kecil, dilakukan pengujian dengan mendeteksi objek pada berbagai jarak tertentu. Model YOLOv8 akan diuji menggunakan dataset baru yang diperoleh dari berbagai jarak antara kamera dan objek. Pengujian ini bertujuan untuk membandingkan kinerja YOLOv8 dalam mendeteksi objek pada kondisi jarak yang berbeda.

Tabel 10 Hasil Pengujian Jarak

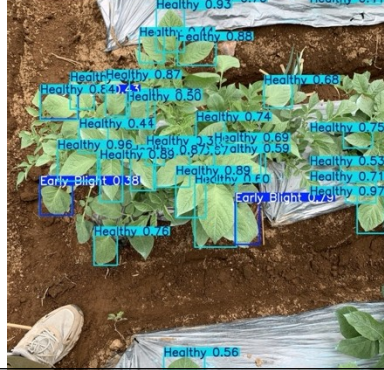
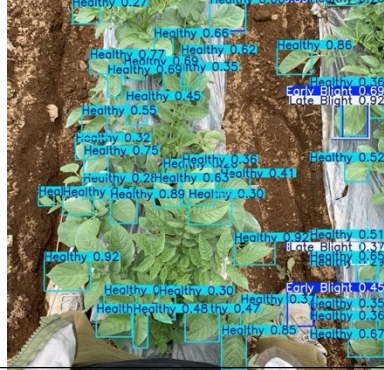
Jarak	Precision	Recall	mAP50	mAP50-95
10 cm	0.827	0.905	0.941	0.897
25 cm	0.843	0.438	0.452	0.401
50 cm	0.268	0.647	0.352	0.256
75 cm	0.128	0.541	0.268	0.153

Dari pengujian yang dilakukan dapat dilihat bahwa YOLOv8 dapat melakukan deteksi terbaik pada jarak 10 cm. Performa YOLOv8 dalam melakukan deteksi berkurang seiring bertambahnya jarak objek dengan kamera.

Penurunan precision dalam pengujian jarak pada penelitian YOLOv8 menunjukkan penurunan yang signifikan seiring bertambahnya jarak objek dengan kamera. Pada jarak 10 cm, precision tercatat sebesar 82.7%, dan sedikit meningkat menjadi 84.3% pada jarak 25 cm, menunjukkan bahwa model masih efektif mendeteksi objek pada jarak dekat. Namun, pada jarak 50 cm, precision menurun drastis menjadi 26.8%, dengan penurunan sebesar 57.5%, yang

menunjukkan kesulitan model dalam mendeteksi objek yang lebih kecil dan lebih jauh. Pada jarak 75 cm, precision kembali menurun menjadi 12.8%, mencatat penurunan sebesar 14% dari jarak 50 cm, yang mengindikasikan bahwa objek yang semakin kecil di gambar membuat model semakin sulit untuk mendeteksi objek dengan akurat.

Tabel 11 Hasil Sebelum dan Setelah Deteksi

Sebelum Deteksi	Setelah Deteksi	Jumlah Deteksi
		41
		33
		48

Berdasarkan pengujian pada jarak 75 cm, model YOLOv8 menunjukkan performa yang menurun dibandingkan jarak yang lebih dekat. Hal ini ditunjukkan dengan nilai precision sebesar 0.128, recall sebesar 0.541, mAP50 sebesar 0.268, dan mAP50-95 sebesar 0.153. Meskipun demikian, dengan nilai confidence 25%, model masih mampu mendeteksi objek dengan benar dalam kisaran 33 hingga 48 daun, tergantung pada jumlah daun yang terdapat dalam citra.

b. Pengujian pada tingkat cahaya yang berbeda

Untuk mengevaluasi performa YOLOv8 terhadap pengaruh intensitas cahaya, dilakukan pengujian dengan mendeteksi objek pada dua kondisi pencahayaan, yaitu intensitas terang dan redup. Model YOLOv8 diuji menggunakan dataset baru yang dihasilkan dari kedua kondisi pencahayaan tersebut. Tujuan dari pengujian ini adalah untuk membandingkan kinerja model YOLOv8 dalam mendeteksi objek pada variasi intensitas cahaya yang berbeda, sehingga dapat dianalisis sejauh mana perubahan pencahayaan memengaruhi akurasi dan efektivitas deteksi.

Tabel 12 Hasil Pengujian Intensitas Cahaya

Kategori	Precision	Recall	mAP50	mAP50-95
Terang	0.988	0.983	0.995	0.95
Redup	0.9	0.887	0.901	0.882

Berdasarkan hasil pengujian yang dilakukan, diperoleh data kuantitatif yang menunjukkan bahwa intensitas cahaya memiliki pengaruh terhadap performa deteksi model YOLOv8. Pada kondisi pencahayaan terang dengan skala tingkat kecerahan 151–200, model mencapai precision sebesar 98.8%, recall 98.3%, mAP50 99.5%, dan mAP50-95 95%, yang mencerminkan kinerja deteksi yang sangat optimal. Sementara itu, pada kondisi pencahayaan redup dengan skala tingkat kecerahan 51–100, terjadi penurunan performa dengan precision sebesar 90%, recall 88.7%, mAP50 90.1%, dan mAP50-95 88.2%.

Penurunan ini disebabkan oleh kualitas citra yang lebih rendah pada intensitas cahaya redup, sehingga model memiliki kesulitan dalam mengenali detail objek secara akurat. Namun, perbedaan performa antara kondisi terang dan redup tidak terlalu signifikan, yang menunjukkan bahwa model deteksi ini cukup robust dalam menghadapi variasi intensitas cahaya yang diuji. Variasi skala intensitas cahaya ini juga menunjukkan potensi model YOLOv8 untuk diterapkan dalam lingkungan nyata dengan pencahayaan yang bervariasi, meskipun faktor pencahayaan tetap perlu diperhatikan untuk memastikan hasil deteksi yang optimal.

4. Kesimpulan dan Saran

Berdasarkan hasil penelitian yang digunakan untuk implementasi YOLOv8 untuk mendeteksi penyakit daun tanaman kentang, maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem yang telah dibangun berhasil melakukan deteksi dengan baik terhadap penyakit daun tanaman kentang dan menghasilkan mAP sebesar 0.994, mAP50-95 sebesar 0,962, precision 0,994 dan recall 0,997 pada training menggunakan epoch bernilai 50.
2. Seiring dengan bertambahnya jarak antara objek dan kamera memiliki pengaruh signifikan terhadap performa model YOLOv8 dalam mendeteksi objek. Pada jarak 10 cm, model menunjukkan performa terbaik dengan nilai precision, recall, mAP50, dan mAP50-95 yang tinggi, menunjukkan kemampuan deteksi yang optimal pada jarak dekat. Namun, seiring bertambahnya jarak, performa model mengalami penurunan secara signifikan, terutama pada precision dan mAP50-95, yang mencerminkan tantangan dalam mendeteksi objek yang lebih kecil dan lebih jauh. Pada jarak 75 cm, model menunjukkan performa terendah, tetapi dengan nilai confidence 25%, model masih dapat mendeteksi objek dengan benar dalam jumlah 33 hingga 48 gambar daun tergantung pada jumlah daun yang terdapat dalam citra. Hal ini menunjukkan bahwa meskipun performa menurun pada jarak yang lebih jauh,

model tetap memiliki kemampuan deteksi yang baik.

3. Intensitas cahaya memang memiliki pengaruh terhadap akurasi deteksi penyakit pada daun kentang. Meskipun terdapat perbedaan antara kondisi pencahayaan terang yang berkisar memiliki nilai kecerahan sebesar 151 – 200 maupun redup dengan nilai kecerahan 51 - 100, pengaruh tersebut tidak terlalu signifikan terhadap kinerja deteksi yang dihasilkan. Hasil evaluasi menunjukkan bahwa pada kondisi pencahayaan terang, model menghasilkan precision, recall, mAP50, dan mAP50-95 yang lebih tinggi dibandingkan dengan kondisi redup. Namun, perbedaan ini tidak terlalu mencolok, yang mengindikasikan bahwa model deteksi yang digunakan cukup baik terhadap variasi intensitas cahaya.

Berdasarkan hasil yang didapat pada penelitian ini, saran yang perlu dilakukan untuk perbaikan sistem dan atau penelitian selanjutnya adalah penelitian selanjutnya diharapkan dapat di kembangkan ke dalam sebuah aplikasi *mobile* dan penelitian selanjutnya dapat mendeteksi penyakit dari jarak yang lebih jauh dengan nilai mAP yang lebih baik.

Daftar Pustaka

1. Angeline, N., Shabrina, N.H. and Indarti, S. (2023a) ‘Faster region-based convolutional neural network for plant-parasitic and non-parasitic nematode detection’, *Indonesian Journal of Electrical Engineering and Computer Science*, 30(1), pp. 316–324. Available at: <https://doi.org/10.11591/ijeecs.v30.i1.pp316-324>.
2. Angeline, N., Shabrina, N.H. and Indarti, S. (2023b) ‘Faster region-based convolutional neural network for plant-parasitic and non-parasitic nematode detection’, *Indonesian Journal of Electrical Engineering and Computer Science*, 30(1), pp. 316–324. Available at: <https://doi.org/10.11591/ijeecs.v30.i1.pp316-324>.
3. Aprilino, A. and Amin, I.H. Al (2022) ‘IMPLEMENTASI ALGORITMA YOLO DAN TESSERACT OCR PADA SISTEM DETEKSI PLAT NOMOR OTOMATIS’, *Jurnal TEKNOINFO*, 16, pp. 54–59.
4. Dicky Septian, A. and Suhendar, A. (2024) ‘IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK DETEKSI PENYAKIT DAUN KENTANG MENGGUNAKAN CITRA DIGITAL’, *Jurnal Informatika Teknologi dan Sains* [Preprint].
5. Dompeipen, T.A., Sompie, S.R.U.A. and Najoran, M.E.I. (2021) ‘jm_informatika, +31471-67034-1-ED2’, 16, pp. 65–76.
6. Hidayat, W.F., Asra, T. and Setiadi, A. (2022) ‘Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression’, *Indonesian Journal on Software Engineering (IJSE)*, 8(2), pp. 173–179. Available at: <http://ejournal.bsi.ac.id/ejurnal/index.php/ijse>.
7. Kurniawan Ekhsanto, B., Adhi Kusuma, B. and Kuncoro, A.P. (2024) ‘IMPLEMENTATION OF YOU ONLY LOOK ONCE V8 ALGORITHM IN

- POTATO LEAF DISEASE DETECTION SYSTEM', 5(4), pp. 125–132. Available at:
<https://doi.org/10.52436/1.jutif.2024.5.4.2104>.
8. Lestari, S. and Irfan Nauval, K. (2022) *IMPLEMENTASI DETEKSI OBJEK PENYAKIT DAUN KENTANG DENGAN METODE CONVOLUTIONAL NEUTRAL NETWORK*,
Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM).
 9. Pramanda, S. et al. (2019) *Sistem Diagnosis Penyakit Tanaman Kentang Menggunakan Metode K-Nearest Neighbor*. Available at: <http://j-ptiik.ub.ac.id>.
 10. Pranatha, M.D., Maricar, M.A. and Setiawan, G.H. (2024) 'Implementasi Arsitektural Resnet-34 Dalam Klasifikasi Gambar Penyakit Pada Daun Kentang', *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(3), pp. 575–580. Available at:
<https://doi.org/10.47233/jteksis.v6i3.1431>.