



# Klasifikasi Penyakit Daun Kentang Berbasis CNN MobileNetV2 dengan Optimasi Randomize Search

Jeason Lie, Abdul Rahman, Daniel Udjulawa\*

Fakultas Ilmu Komputer dan Rekayasa, Program Studi Informatika, Universitas Multi Data Palembang, Palembang, Indonesia

Email: <sup>1</sup>jeasonlie@gmail.com, <sup>2</sup>arahman@mdp.ac.id, <sup>3</sup>\*daniel@mdp.ac.id

Email Penulis Korespondensi: daniel@mdp.ac.id

**Abstrak**—Kentang merupakan komoditas pangan vital di Indonesia, namun produktivitasnya sering menurun secara signifikan akibat serangan berbagai penyakit daun yang menghambat pertumbuhan. Penelitian ini bertujuan untuk membangun model klasifikasi otomatis penyakit daun kentang yang efisien dan akurat menggunakan teknologi *Deep Learning*. Pendekatan yang digunakan dalam penelitian ini adalah arsitektur *Convolutional Neural Network (CNN) MobileNetV2* berbasis *transfer learning* yang dikenal memiliki efisiensi komputasi tinggi. Untuk mendapatkan performa model yang paling optimal, penelitian ini menerapkan strategi *hyperparameter tuning* otomatis menggunakan metode *Randomize Search* serta melakukan validasi model yang kuat melalui teknik *K-Fold Cross Validation* dengan 5 *fold*. Selain itu, teknik *balanced class weight* turut diterapkan guna mengatasi masalah ketidakseimbangan data pada delapan kelas penyakit yang diuji. Hasil eksperimen menunjukkan bahwa konfigurasi model terbaik dicapai pada iterasi ke-15 *fold* ke-5 dengan menggunakan kombinasi parameter *optimizer RMSprop*, *epoch 35*, dan *learning rate 0.001*. Evaluasi akhir pada data uji independen menghasilkan akurasi sebesar 78,45%, presisi 84,24%, dan *recall* 72,94%. Selisih yang sangat kecil antara akurasi validasi 79,30% dan akurasi uji mengindikasikan kemampuan generalisasi yang baik tanpa terjadi *overfitting*. Meskipun model mencapai hasil sempurna pada klasifikasi *Alternaria*, tantangan masih ditemukan pada identifikasi kelas *Fungi* yang memiliki kemiripan visual tinggi dengan *Phytophthora* dan *Pest*. Penelitian ini menyimpulkan bahwa integrasi *MobileNetV2* dengan optimasi *hyperparameter* mampu mengklasifikasikan penyakit daun kentang secara efektif.

**Kata Kunci:** Klasifikasi; Penyakit Daun Kentang; *Deep Learning*; *MobileNetV2*; *Transfer Learning*

**Abstract**—Potatoes are a vital food commodity in Indonesia, but their productivity often declines significantly due to attacks by various leaf diseases that inhibit growth. This study aims to build an efficient and accurate automatic classification model for potato leaf diseases using *Deep Learning* technology. The approach used in this study is the *MobileNetV2 Convolutional Neural Network (CNN)* architecture based on *transfer learning*, which is known to have high computational efficiency. To obtain the most optimal model performance, this study applies an automatic *hyperparameter tuning* strategy using the *Randomize Search* method and performs robust model validation using the *K-Fold Cross Validation* technique with 5 folds. In addition, the *balanced class weight* technique is also applied to overcome the problem of data imbalance in the eight disease classes tested. The experimental results show that the best model configuration is achieved at the 15th iteration of the 5th fold using a combination of *RMSprop optimizer* parameters, 35 epochs, and a learning rate of 0.001. The final evaluation on independent test data produces an accuracy of 78.45%, a precision of 84.24%, and a recall of 72.94%. The very small difference between the validation accuracy of 79.30% and the test accuracy indicates good generalization ability without *overfitting*. Although the model achieved excellent results in *Alternaria* classification, challenges remained in identifying the *Fungi* class, which has high visual similarity to *Phytophthora* and *Pest*. This study concludes that the integration of *MobileNetV2* with *hyperparameter optimization* is capable of effectively classifying potato leaf diseases.

**Keywords:** Classification; Potato leaf disease; *Deep Learning*; *MobileNetV2*; *Transfer Learning*

## 1. PENDAHULUAN

Kentang (*Solanum tuberosum L.*) merupakan tanaman pangan berumbi kaya akan karbohidrat yang tumbuh subur di dataran tinggi Indonesia. Selain berfungsi sebagai sumber energi utama, kentang juga mengandung serat pangan yang tinggi serta vitamin C yang sangat bermanfaat bagi metabolisme tubuh (Lesmana et al., 2022). Komoditas ini memegang peranan yang sangat vital dalam ekosistem industri pangan nasional, di mana pada tahun 2022 jumlah produksi kentang tercatat mencapai angka 1,5 juta ton. Namun, stabilitas produktivitas tersebut mengalami guncangan dengan penurunan drastis pada tahun 2023 menjadi 1,2 juta ton. Penurunan produksi yang mencapai angka 300 ribu ton ini sebagian besar, atau sekitar 80%, disebabkan oleh serangan *patogen* penyakit pada daun yang secara signifikan menghambat pertumbuhan tanaman dan merusak kualitas umbi (Ghandi et al., 2024). Kendala utama di lapangan saat ini adalah proses identifikasi penyakit yang masih sering dilakukan secara manual melalui pengamatan visual mata telanjang. Metode konvensional ini memiliki kelemahan besar, yaitu subjektivitas pengamat, kelelahan mata, dan ketidakakuratan dalam membedakan gejala penyakit yang memiliki kemiripan visual tinggi. Untuk mengatasi masalah penyakit daun kentang ini, salah satu solusi teknologi yang dapat digunakan adalah pengolahan citra digital (*image processing*). Pemanfaatan *image processing* memungkinkan klasifikasi penyakit pada daun kentang dilakukan secara otomatis dan presisi secara visual, sehingga pengelola pertanian dapat memberikan penanganan yang lebih efektif, cepat, dan efisien terhadap tanaman yang terindikasi sakit (Rozaqi et al., 2021).

Sebagai metode utama dalam pengolahan citra, *convolutional neural network (CNN)* merupakan model *deep learning* yang dikembangkan dari *multi-layer perceptron (MLP)* dan memiliki keunggulan dalam mengenali objek pada data citra karena struktur jaringan sarafnya yang dalam (Permana et al., 2022). Untuk meningkatkan efisiensi *CNN*, teknik *transfer learning* sering digunakan dengan memanfaatkan model *pre-trained*. Teknik ini memiliki kinerja jaringan saraf yang lebih baik karena telah dilatih menggunakan *dataset* besar, sehingga mampu menghemat waktu pelatihan dan bekerja optimal meski menggunakan dataset yang lebih sedikit (Aisyah & Prasetyadi, 2023). Salah satu

implementasi penting dalam *transfer learning* adalah *deep feature extraction*, yaitu proses mengekstraksi fitur citra menggunakan model *pre-trained* untuk menghasilkan representasi fitur yang informatif (Bansal et al., 2023). Dalam penelitian ini, arsitektur yang dipilih adalah *MobileNetV2*, sebuah pengembangan *google* yang menggunakan konsep *inverted residuals* dan *linear bottleneck*. Arsitektur ini dirancang untuk mencapai keseimbangan antara akurasi dan efisiensi, sehingga sangat cocok untuk pengelolaan data citra pada perangkat dengan sumber daya terbatas (Angeline & Kusniyati, 2024) (Agustiani et al., 2024).

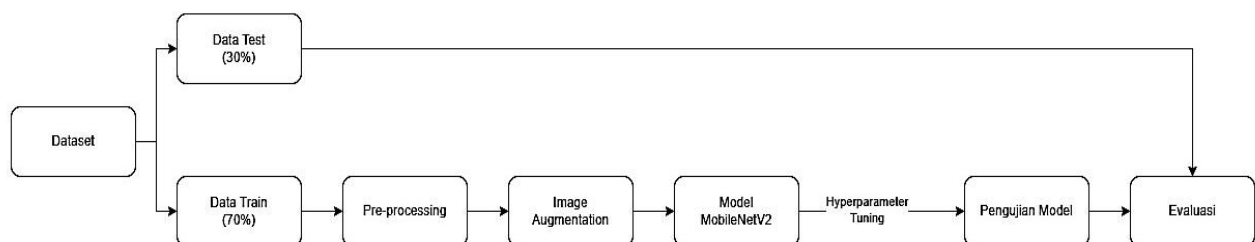
Berbagai penelitian terdahulu telah berupaya mengembangkan sistem klasifikasi penyakit daun kentang dengan beragam metode. Pada pendekatan *machine learning* konvensional, Amatullah et al. (2021) menggunakan ekstraksi fitur *GLCM* dengan algoritma *k-nearest neighbors (KNN)* pada 3 kelas penyakit dan menghasilkan akurasi terbaik 80% pada nilai  $k=3$ . Pendekatan serupa dilakukan oleh Hidayat et al. (2022) yang membandingkan *logistic regression* klasik dengan ekstraksi fitur *ResNet50*. Hasil penelitian tersebut menunjukkan variasi signifikan dengan penggunaan ekstraksi fitur *ResNet50* justru menghasilkan akurasi 80%, berbeda dengan metode klasik, yang diklaim mencapai 98% namun pada dataset yang sangat terbatas yaitu 405 citra. Hal ini mengindikasikan bahwa metode konvensional memiliki keterbatasan dalam konsistensi akurasi ketika dihadapkan pada variasi data citra.

Peralihan ke pendekatan *deep learning* menunjukkan peningkatan performa yang lebih menjanjikan. Asif et al. (2020) mengembangkan model *legitimate sequential CNN* untuk mendeteksi 3 kelas penyakit dengan 3000 data, yang menghasilkan akurasi 84,83%. Pengembangan model yang lebih kompleks dilakukan oleh Aisyah & Prasetyadi (2023) dengan menggabungkan *CNN* dan *RNN (VGG16-LSTM dan ResNet50-LSTM)* pada 5 kelas penyakit. Penelitian ini menemukan bahwa arsitektur *VGG16* dengan *dense layer 75* memberikan hasil terbaik dengan akurasi 86%. Lebih spesifik pada perbandingan arsitektur efisien, Fuadi & Suharso (2022) membandingkan *MobileNet* dan *NasNetMobile* unggul pada skema percobaan tertentu dengan akurasi 90,96%, serta menyarankan perlunya eksplorasi lebih lanjut pada teknik pengolahan data dan konfigurasi *fine tuning* untuk meningkatkan performa model.

Meskipun penelitian-penelitian sebelumnya telah menunjukkan hasil yang cukup baik, masih terdapat celah penelitian yang perlu dikembangkan. Mayoritas studi terdahulu, seperti yang dilakukan oleh Asif et al. (2020), Amatullah et al. (2021), dan Hidayat et al. (2022), hanya membatasi klasifikasi pada tiga kelas penyakit yaitu *early blight*, *late blight*, dan *healthy*, sehingga kurang merepresentasikan kompleksitas penyakit di lapangan. Selain itu, penentuan parameter model dilakukan secara manual tanpa proses optimasi yang mendalam. Oleh karena itu, penelitian ini bertujuan untuk mengisi kekurangan tersebut dengan menggunakan arsitektur *MobileNetV2* pada dataset penyakit yang lebih bervariasi. Kebaharuan utama dalam penelitian ini terletak pada penerapan teknik optimasi *hyperparameter* menggunakan *randomize search* untuk menemukan konfigurasi parameter terbaik secara otomatis, serta penggunaan metode validasi *k-fold cross validation* untuk memastikan model yang dihasilkan bersifat *robust* dan objektif, berbeda dengan pendekatan validasi statis pada penelitian sebelumnya.

## 2. METODOLOGI PENELITIAN

Penelitian ini menerapkan metodologi eksperimental untuk mengklasifikasi penyakit daun tanaman kentang menggunakan algoritma *CNN* dengan arsitektur *MobileNetV2*. Tahapan penelitian disusun secara sistematis untuk meminimalisir kesalahan dalam proses penelitian. Tahapan metodologi yang lengkap dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metodologi

Untuk tahapan awal dilakukan pengambilan dataset dilanjut dengan melakukan *splitting dataset*, setelah itu dilakukan *pre-processing* dan *image augmentation* pada data latih. Tahap selanjutnya dibuat model deep learning yang berbasis *MobileNetV2* dilanjut dengan proses pelatihan model, dan *hyperparameter tuning*. Setelah model dilatih, dilakukan pengujian menggunakan data latih. Setelah pengujian model selesai, dilakukan evaluasi model.

### 2.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah dataset publik yang diambil dari website *Mendeley Data* yang dibuat oleh (Shabrina et al., 2023) dan Kaggle. Dataset mencakup gejala penyakit pada daun tanaman kentang yang diambil dari beberapa angle dengan jarak 5-15 cm dan berbagai kondisi cuaca dengan rentang waktu jam 8 pagi sampai jam 3 petang. Resolusi yang digunakan untuk mengambil seluruh gambar pada dataset ini adalah 1500x1500. Terdapat penambahan kelas *Alternaria* dari kaggle untuk melengkapi 7 kelas lainnya yang berasal dari Mendeley Data. Rician distribusi jumlah data untuk setiap kelas penyakit disajikan secara lengkap pada Tabel 1.

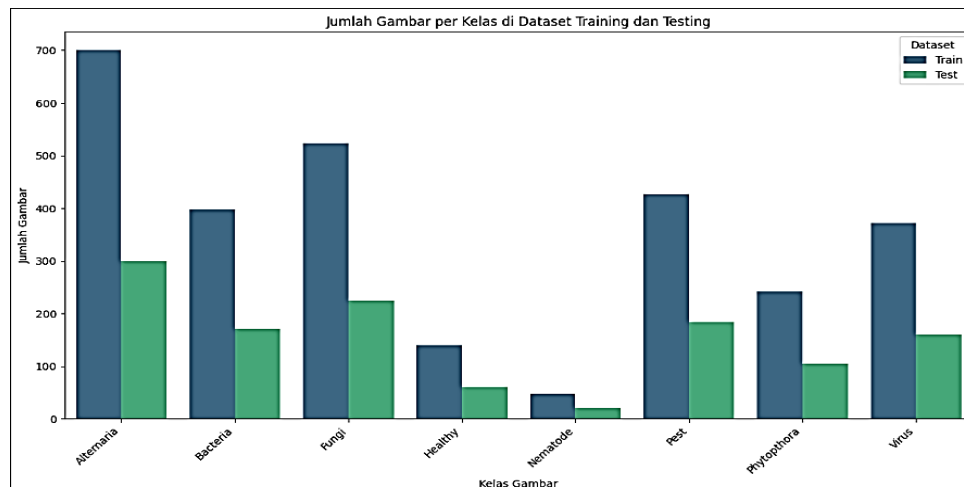
**Tabel 1.** Dataset yang digunakan

Nama Penyakit	Jumlah Data
<i>Alternaria</i>	1000
<i>Bacteria</i>	569
<i>Fungi</i>	748
<i>Healthy</i>	201
<i>Nematode</i>	68
<i>Pest</i>	611
<i>Phytophthora</i>	347
<i>Virus</i>	532

Dataset ini memiliki total 8 kelas penyakit dengan komposisi jumlah yang bervariasi. Kelas *Alternaria* memiliki jumlah data terbanyak yaitu 1000 citra, diikuti oleh *Fungi* dengan jumlah data 748 citra, dan *Pest* dengan jumlah data 611. Sementara itu, kelas *Nematode* menjadi kelas dengan jumlah data paling sedikit, yaitu hanya 68 citra. Data-data tersebut dikumpulkan dalam rentang usia tanaman 35-80 hari setelah penanaman.

## 2.2 Split Dataset

Setelah tahap pengumpulan data, dilakukan split data secara manual dengan perbandingan 70% data latih dan 30% data uji. Berikut hasil split data yang dapat dilihat pada Gambar 2.

**Gambar 2.** Hasil Split Dataset

Berdasarkan plot split data, setiap kelas memiliki 2 batang data, batang biru merepresentasikan jumlah data latih dan batang hijau merepresentasikan jumlah data uji. Terlihat bahwa proposi pembagian data diterapkan secara konsisten pada seluruh kelas, mulai dari *Alternaria* sampai *Virus*. Selain pembagian data statis tersebut, untuk data validasi akan dibagi menggunakan *k-fold cross validation* dengan jumlah  $k=5$ . Tujuan penggunaan *k-fold cross validation* adalah untuk mengurangi bias yang sering terjadi pada saat pengambilan sample data secara acak, sehingga hasil evaluasi model menjadi lebih objektif dan tidak bergantung pada keberuntungan satu kali pembagian data (Ridwansyah, 2022).

## 2.3 Pre-processing

Pada tahap ini, setiap data, baik data latih dan data uji akan dilakukan *pre-processing*. Teknik *pre-processing* yang digunakan adalah *resize* dengan tujuan untuk menyesuaikan ukuran citra dan ukuran *input* pada model yang digunakan (Shahi et al., 2022). Selain *resize*, *batch normalization* digunakan pada penelitian ini dengan tujuan agar mengatur nilai data ke rentang yang spesifik seperti -1,1 atau 0,1 sehingga proses pelatihan model lebih cepat (Alasadi & Bhaya, 2017).

## 2.4 Image Augmentation

Tahap selanjutnya, melakukan augmentasi pada data citra. Augmentasi hanya dilakukan pada data latih agar tidak terjadi *data leakage*. Teknik-teknik augmentasi yang digunakan pada penelitian ini adalah *random horizontal* dan *vertical flip*, *random rotation* dengan rentang rotasi 0.2, dan *random zoom* dengan rentang rotasi 0.2. Tujuan penggunaan *image augmentation* adalah untuk meningkatkan keragaman data latih, meningkatkan generalisasi serta *robustness* pada model, dan mampu mengurangi resiko *overfitting* (Yang et al., 2022).

## 2.5 Imbalance Data

Dalam menangani *imbalance data*, penelitian ini menggunakan *balanced class weight*. Tujuan dari penggunaan *class weight* adalah untuk mendorong model mempelajari pola kelas minoritas secara efektif tanpa menambah data sintesis



sehingga mengurangi terjadinya bias terhadap kelas mayoritas (Abdelhamid & Desai, 2024). Berikut nilai bobot yang digunakan disajikan pada Tabel 2.

**Tabel 2.** Nilai Class Weight

Class	Weight
Alternaria	0.5088
Bacteria	0.8948
Fungi	0.6809
Healthy	2.5438
Nematode	7.5771
Pest	0.8340
Phytophthora	1.4716
Virus	0.9573

Mengacu pada Tabel 2, terlihat pemberian bobot berbanding terbalik dengan jumlah data. Kelas Nematode memiliki nilai bobot tertinggi yaitu 7.571, yang menandakan bahwa ini kelas minoritas. Sebaliknya, kelas Alternaria mendapatkan bobot paling rendah sebesar 0.5088 karena merupakan kelas mayoritas. Pemberian bobot yang besar pada kelas Nematode dan Healthy bertujuan untuk menyeimbangkan penalti loss saat pelatihan, sehingga model tidak bias ke kelas yang dominan.

## 2.6 Hyperparameter Tuning

Pada penelitian ini, adanya penggunaan teknik *hyperparameter tuning*. Tujuan penggunaan *hyperparameter tuning* adalah memastikan model dapat belajar dengan baik sesuai dengan target *dataset*. Ada 4 parameter pengujian yang dituning yaitu *optimizer*, *epoch*, *batch size*, dan *learning rate*. *Optimizer* yang digunakan adalah *Adam*, *SGD*, dan *RMSprop*. *Epoch* yang digunakan adalah 25, 35, 50. *Batch size* yang digunakan adalah 16, 32, dan 64. *Learning rate* yang digunakan adalah 0.01, 0.001, dan 0.0001. Metode *hyperparameter tuning* yang digunakan pada penelitian ini adalah *Randomize Search* dengan jumlah iterasi sebanyak 15 kali. Alasan penggunaan *Randomize Search* pada penelitian ini adalah untuk menghemat komputasi pelatihan dengan cara melakukan kombinasi parameter secara acak (Rachmatsyah et al., 2024).

## 2.7 Perancangan Model

Perancangan model pada penelitian ini, dilakukan dengan pendekatan *transfer learning* dengan memanfaatkan *MobileNetV2* sebagai *feature extractor*. *MobileNetV2* dipilih sebagai *base model* karena ukurannya yang ringan dan efisiensi yang tinggi. *Base model* yang dibekukan dimodifikasi dengan penambahan *Global Average Pooling*, *Dense Layer* dengan jumlah 128 *neuron* serta aktivasi *ReLU*, dan *Dropout* dengan nilai 0,5 untuk melakukan klasifikasi. *Output layer* pada model ini menggunakan *Dense Layer* dengan jumlah 8 *neuron* dengan aktivasi *softmax*.

Untuk memastikan performa model yang objektif dan optimal, pelatihan model menggunakan teknik *k-fold cross validation* dengan jumlah  $k = 5$  yang dikombinasikan dengan *randomize search* serta penerapan *class weights*. *Loss function* yang digunakan pada penelitian ini adalah *categorical crossentropy* untuk mengklasifikasi multi class. Berikut model summary yang akan disajikan pada Tabel 3

**Tabel 3.** Model Summary

Layer	Output Shape	Param#
input_layer_1	(None, 224, 224, 3)	0
mobilenetv2_1.00_224	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d	(None, 1280)	0
dense	(None, 128)	163,968
dropout	(None, 128)	0
dense_1	(None, 8)	1,032
total params:	2,422,984	
trainable params:	165,000	
non-trainable params:	2,257,984	

Karena menerapkan transfer learning dengan membekukan base model, mayoritas parameter tersebut adalah 2,257,984 yang bersifat non-trainable. Hanya 165,000 parameter yang bersifat trainable yang berasal dari dense tambahan dengan jumlah 163,968 parameter dan lapisan output dengan jumlah 1,032 parameter. Efisiensi jumlah parameter yang dilatih diharapkan dapat mempercepat proses komputasi tanpa mengurangi akurasi secara signifikan.

## 2.8 Evaluasi Model

Evaluasi model yang digunakan pada penelitian ini adalah *confusion matrix* dan *classification report*. *Confusion matrix* merupakan tool analitik berupa tabel yang memiliki 4 representasi hasil dari proses klasifikasi, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* dengan tujuan untuk mengevaluasi kinerja pelatihan

model dengan memprediksi perbandingan nilai aktual dan nilai dari prediksi. Pengukuran kinerja diantaranya dapat menghasilkan nilai akurasi, presisi, *recall*, dan *f1-score* (Negoro et al., 2023). Setelah nilai-nilai tersebut didapat, maka akan diringkas dalam bentuk *classification report*. Berikut rumus yang digunakan untuk menghitung akurasi, presisi, *recall*, dan *f1-score* dapat dilihat pada Persamaan (1), (2), (3), dan (4):

$$Accuracy : \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision+Recall} \tag{4}$$

### 3. HASIL DAN PEMBAHASAN

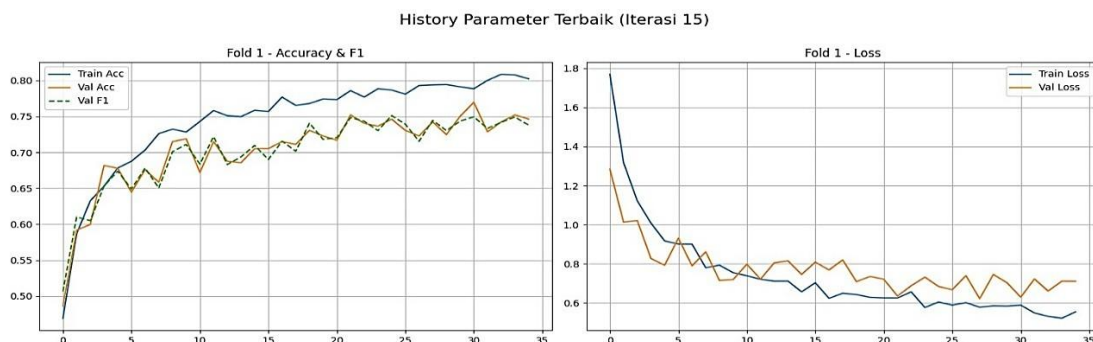
#### 3.1 Hasil Pelatihan Model

Pada saat proses pelatihan model, digunakan teknik *k-fold cross validation* serta menggunakan *randomize search* dengan jumlah 15 iterasi. Dengan banyak kombinasi parameter serta *fold* yang digunakan, diterapkan 2 fitur *callback* pada penelitian ini yaitu *early stopping* dan *checkpoint*. Tujuan *early stopping* dengan *patience* 10 digunakan agar dapat mencegah model mengalami *overfitting* dengan cara menghentikan pelatihan jika *validation loss* mengalami kenaikan. *Checkpoint* digunakan agar dapat menyimpan model terbaik berdasarkan *validation accuracy*. Berikut daftar percobaan yang dilakukan disajikan pada Tabel 4.

**Tabel 4.** Hasil Iterasi Randomize Search

Iterasi	Optimizer	Epoch	Batch Size	Learning Rate	Mean Accuracy
1	Adam	25	32	0.001	0.6408
2	Adam	25	16	0.01	0.6443
3	SGD	50	16	0.001	0.5936
4	SGD	25	32	0.001	0.5485
5	Adam	50	16	0.01	0.6457
6	SGD	25	32	0.01	0.6349
7	SGD	35	16	0.001	0.6614
8	SGD	35	64	0.0001	0.2344
9	SGD	25	16	0.001	0.5675
10	Adam	35	16	0.001	0.6479
11	SGD	50	32	0.001	0.6982
12	Adam	25	32	0.0001	0.6985
13	SGD	35	64	0.001	0.6918
14	RMSprop	25	32	0.0001	0.7136
15	RMSprop	35	64	0.001	0.7645

Berdasarkan hasil eksperimen yang tertera pada Tabel 4, terlihat adanya variasi performa yang signifikan antar skenario. Pencapaian terbaik diraih pada iterasi ke-15 dengan *mean* akurasi yaitu 76,45%. Konfigurasi optimal ini menggunakan kombinasi parameter *optimizer rmsprop*, *epoch* 35, *batch size* 64, dan *learning rate* 0.001. Sebaliknya, hasil terendah tercatat pada iterasi ke-8 dengan *mean* akurasi yaitu 23,44%. Rendahnya *mean* akurasi pada iterasi ke-8 disebabkan oleh penggunaan *learning rate* yang terlalu kecil yaitu 0.0001 pada *optimizer* SGD, yang menghambat model mencapai konvergensi optimal. Berikut penjelasan performa model dan plot *learning curve* setiap *fold* pada iterasi ke-15 bisa dilihat pada Gambar 3 sampai Gambar 7.

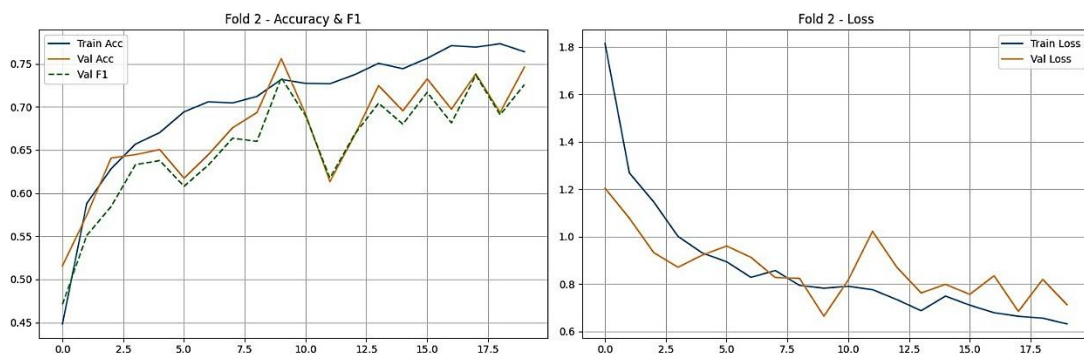


**Gambar 3.** Learning Curve Fold 1 Iterasi 15

Gambar 3 menampilkan *plot learning curve* dari model klasifikasi pada *fold 1*. *plot* ini merepresentasikan hasil dari konfigurasi parameter terbaik yang ditemukan pada iterasi ke-15. Proses pelatihan berlangsung selama 35 *epoch*, di mana kinerja model dipantau melalui metrik akurasi dan *f1 score* pada grafik kiri, serta nilai *loss* pada grafik kanan, baik untuk data latih maupun data validasi.

Pada *plot* sebelah kiri, terlihat bahwa model mampu mempelajari fitur dengan baik, ditunjukkan oleh garis *training accuracy* yang meningkat secara stabil dan mulus hingga mencapai angka sekitar 0.80 pada akhir pelatihan. Sejalan dengan itu, *validation accuracy* dan *validation f1 score* juga menunjukkan tren kenaikan hingga mencapai kisaran 0.75. Namun, pergerakan garis validasi terlihat lebih fluktuatif dibandingkan garis pelatihan, dan mulai terbentuk *gap* antara performa *training* dan validasi.

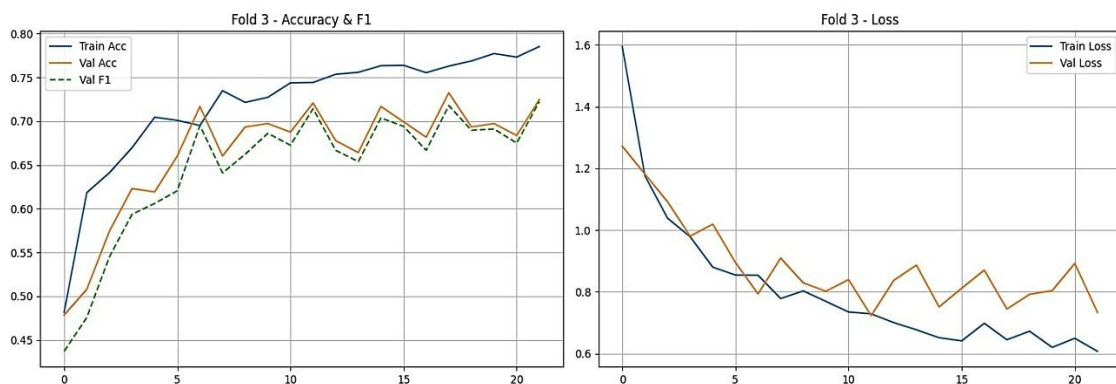
Analisis pada grafik *loss* di sebelah kanan mempertegas dinamika pelatihan tersebut. *training loss* menurun secara konsisten dan tajam dari awal hingga melandai di angka sekitar 0.55, menandakan model terus meminimalisir kesalahan pada data latih. Sebaliknya, *validation loss* menurun tajam hanya hingga *epoch* ke-5 atau ke-10. Setelah titik tersebut, nilai *loss* validasi cenderung stagnan di kisaran 0.7 dan mengalami fluktuasi yang cukup signifikan tanpa penurunan lebih lanjut yang berarti, yang mengindikasikan awal terjadinya *overfitting* di mana model lebih optimal pada data latih dibandingkan data validasi.



**Gambar 4.** Learning Curve Fold 2 Iterasi 15

Berbeda dengan *fold 1*, *plot* riwayat pelatihan pada *fold 2* yang divisualisasikan pada Gambar 4 menunjukkan dinamika yang lebih fluktuatif dan durasi pelatihan yang lebih singkat, berakhir pada *epoch* ke-20. Penghentian ini mengindikasikan bekerjanya mekanisme *early stopping* akibat tidak adanya perbaikan signifikan pada metrik validasi dalam beberapa iterasi terakhir. Pada grafik akurasi di bagian kiri, *training accuracy* meningkat stabil hingga 77%, namun *validation accuracy* menunjukkan ketidakstabilan yang tinggi dengan pola naik-turun yang tajam, termasuk penurunan performa yang drastis pada *epoch* ke-11.

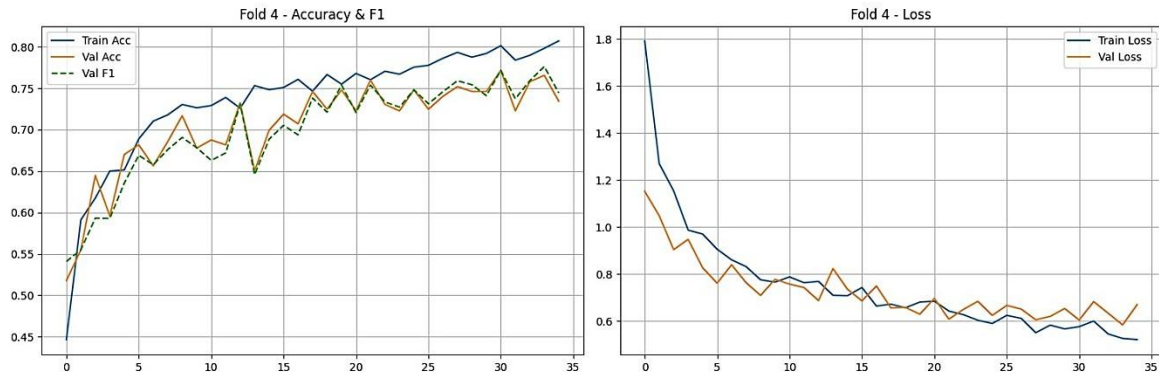
Pola serupa terlihat pada *plot loss* yang ada pada bagian kanan. Sementara *training loss* menurun dengan mulus, *validation loss* mencapai titik optimalnya di sekitar *epoch* ke-9 sebelum akhirnya mengalami tren kenaikan dan fluktuasi yang ekstrem. Ketidakstabilan pada *fold 2* ini, jika dibandingkan dengan *fold 1* yang lebih mulus, mengindikasikan bahwa *subset* data validasi pada *fold* ini kemungkinan memiliki karakteristik yang lebih sulit diprediksi atau mengandung *noise* yang lebih tinggi, sehingga model kesulitan mempertahankan performa generalisasi yang konsisten.



**Gambar 5.** Learning Curve Fold 3 Iterasi 15

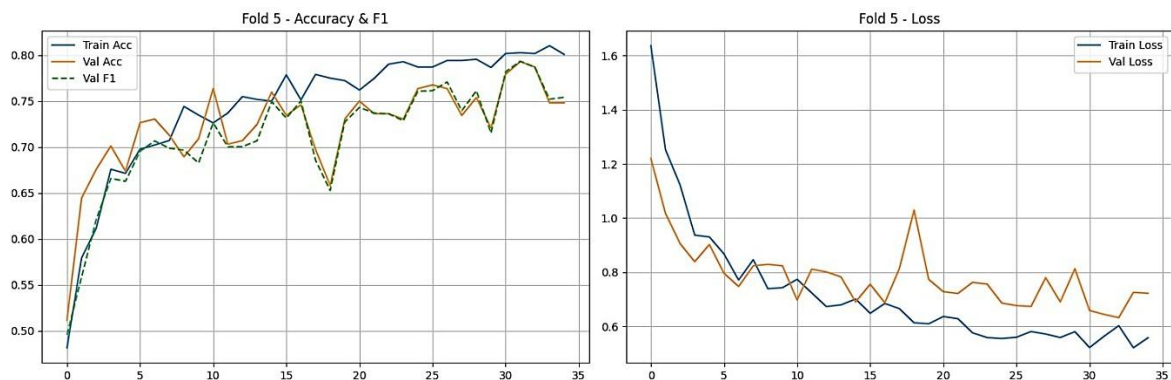
Visualisasi *Plot* riwayat pelatihan *fold 3* pada Gambar 5 menunjukkan pola *overfitting* yang lebih jelas dibandingkan *fold* sebelumnya. Pelatihan berakhir pada *epoch* ke-22. Pada grafik *loss* yang berada di kanan, terlihat adanya divergensi tren yang dimulai sejak dini, *training loss* menurun secara konsisten, sementara *validation loss* berhenti menurun di sekitar *epoch* ke-7 dan mulai menunjukkan tren peningkatan perlahan hingga akhir pelatihan. Meskipun demikian, akurasi validasi masih sempat mencapai puncaknya di sekitar *epoch* ke-16 sebelum akhirnya

menurun, yang menegaskan bahwa model mulai menghafal data latih dan kehilangan kemampuan generalisasi pada data validasi.



**Gambar 6.** Learning Curve Fold 4 Iterasi 15

Berbeda dengan *fold 2* dan *3*, hasil pelatihan pada *fold 4* yang divisualisasikan pada Gambar 6 menunjukkan stabilitas yang jauh lebih baik dan durasi pelatihan yang lebih panjang hingga 35 *epoch*. *Plot* akurasi dan *f1-score* pada bagian kiri gambar menunjukkan peningkatan yang selaras antara data latih dan validasi dengan *gap* yang wajar. Pada *plot loss* di bagian kanan gambar, meskipun terdapat fluktuasi kecil, *validation loss* cenderung menurun seiring waktu dan tidak menunjukkan divergensi yang ekstrem menjauhi *training loss*. Hal ini menandakan bahwa pada pembagian data *fold 4*, model mampu mencapai konvergensi yang optimal.



**Gambar 7** Learning Curve Fold 5 Iterasi 15

Visualisasi *plot fold 5* pada Gambar 7 menunjukkan, pelatihan juga berhasil diselesaikan hingga 35 *epoch*. Secara umum, tren pelatihan menunjukkan konvergensi yang positif. Namun, jika diperhatikan pada grafik *loss* di bagian kanan gambar, terdapat anomali signifikan di sekitar *epoch* ke-18, di mana terjadi lonjakan mendadak pada *validation loss* dan penurunan pada *validation accuracy*. Meskipun demikian, model menunjukkan kemampuan pemulihan yang baik. Setelah *epoch* tersebut, performa validasi kembali stabil dan meningkat hingga akhir sesi pelatihan, menyamai performa training dengan baik.

Secara keseluruhan, evaluasi menggunakan 5 *fold cross validation* mengungkap variabilitas kinerja model terhadap partisi data yang berbeda. *Fold 1*, *fold 4*, dan *fold 5* menunjukkan stabilitas dan konvergensi yang baik, membuktikan bahwa arsitektur model mampu belajar efektif pada sebagian besar data. Namun, ketidakstabilan dan *overfitting* dini yang terjadi pada *fold 2* dan *fold 3* menyoroti adanya *subset* data yang lebih kompleks atau *noisy*, yang memengaruhi kemampuan generalisasi model pada iterasi tertentu.

### 3.2 Hasil Pengujian Model

Berdasarkan rangkaian percobaan pelatihan yang telah dilakukan menggunakan metode *k-fold cross validation* yang dikombinasikan dengan teknik pencarian *hyperparameter randomize search*, sistem secara otomatis mengidentifikasi konfigurasi model yang paling optimal. Pemilihan ini didasarkan pada performa model dalam mengenali pola penyakit pada data validasi, di mana model dari *fold* ke-5 pada iterasi ke-15 terpilih sebagai model terbaik dengan capaian validasi akurasi sebesar 79.30%.

Konfigurasi *hyperparameter* yang membentuk model terbaik ini menggunakan *optimizer RMSprop* dengan *learning rate* sebesar 0.001. Proses pelatihan dilakukan selama 35 *epoch* dengan *batch size* sebanyak 64. Kombinasi parameter ini terbukti paling efektif dalam meminimalkan *loss* dan memaksimalkan akurasi dibandingkan kombinasi lainnya dalam ruang pencarian *Random Search*.

Tahap selanjutnya adalah pengujian vital menggunakan data uji (*test set* atau *hold-out set*). Data ini merupakan sekumpulan citra daun kentang yang benar-benar baru dan belum pernah "dilihat" oleh model selama proses pelatihan



maupun validasi. Pengujian ini bertujuan untuk mengukur kemampuan generalisasi model secara objektif. Berikut hasil dari pengujian menggunakan data uji, dilihat pada Tabel 5.

**Tabel 5.** Hasil Test Menggunakan Data Uji

Optimizer	Epoch	Batch Size	Learning Rate	Accuracy	Precision	Recall	Loss
RMSprop	35	64	0.001	0.7845	0.8424	0.7294	0.5713

Test model yang disajikan pada Tabel 5 menghasilkan tingkat akurasi sebesar 78.45% pada data uji. Angka ini merepresentasikan persentase total prediksi yang benar dari keseluruhan data uji. Temuan yang paling signifikan adalah kedekatan antara nilai akurasi validasi 79.30% dengan akurasi uji 78.45%. Selisih yang sangat kecil ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik. Model terbukti mampu mengenali pola penyakit pada data baru dengan performa yang setara dengan saat latihan, serta membuktikan bahwa model tidak mengalami *overfitting*.

Mengacu pada Tabel 5, nilai presisi yang diperoleh adalah 84.24%. Dalam konteks klasifikasi penyakit tanaman, angka ini menunjukkan tingkat keandalan model yang tinggi saat memberikan prediksi positif. Artinya, dari seluruh citra yang diprediksi sebagai kelas penyakit tertentu oleh model, 84,24% di antaranya adalah benar. Tingginya nilai presisi mengindikasikan bahwa model memiliki tingkat kesalahan *false positive* yang rendah, model tidak mudah "menuduh" daun sehat sebagai daun berpenyakit jika tidak memiliki keyakinan yang cukup kuat.

Sementara itu, nilai *recall* pada Tabel 5 tercatat sebesar 72.94%. Metrik ini mengukur kemampuan model dalam menemukan kembali seluruh kejadian penyakit yang ada di *dataset*. Meskipun cukup baik, nilai *recall* yang lebih rendah dibandingkan presisi menunjukkan bahwa model cenderung bersifat sedikit konservatif. Masih terdapat sekitar 27% kasus penyakit yang terlewat atau *false negative*.

Terakhir, nilai *loss* akhir pada data uji yang mengacu pada Tabel 5 berada pada angka 0.5713. Nilai ini merepresentasikan rata-rata kesalahan antara probabilitas prediksi model dengan label yang sebenarnya. Semakin rendah nilai ini, semakin dekat prediksi model dengan kebenaran.

Secara keseluruhan, model *MobileNetV2* yang dikembangkan telah menunjukkan performa yang stabil dan valid secara statistik, dengan keseimbangan yang baik antara kemampuan belajar dan kemampuan adaptasi pada data baru.

### 3.3 Evaluasi Model

Setelah pelatihan dan pengujian model, dilakukan evaluasi dan analisis lebih mendalam untuk mengetahui kualitas prediksi pada masing-masing kelas penyakit. Evaluasi ini menggunakan *classification report* yang merinci metrik *precision*, *recall*, dan *f1-score* untuk setiap label. Rincian performa klasifikasi untuk kedelapan kelas penyakit disajikan dalam bentuk tabel pada Tabel 6.

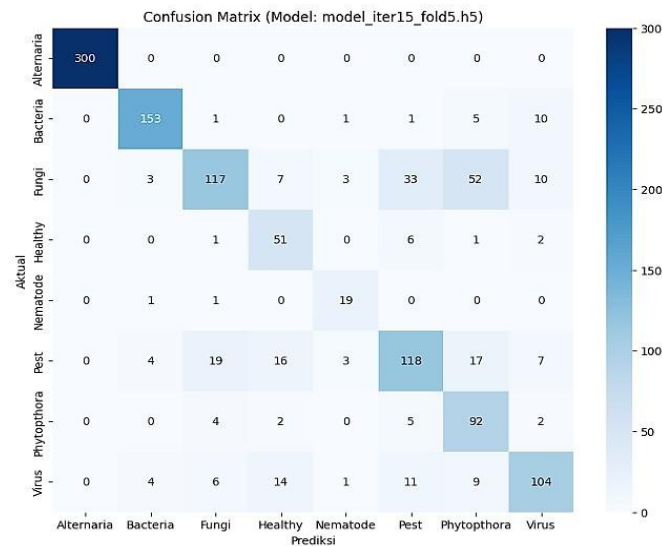
**Tabel 6.** Hasil Classification Report

	Precision	Recall	F1-Score	Support
Alternaria	1.00	1.00	1.00	300
Bacteria	0.93	0.89	0.91	171
Fungi	0.79	0.52	0.63	225
Healthy	0.57	0.84	0.68	61
Nematode	0.70	0.90	0.79	21
Pest	0.68	0.64	0.66	184
Phytophthora	0.52	0.88	0.65	105
Virus	0.77	0.70	0.73	149
Accuracy			0.78	1216
Macro avg	0.74	0.80	0.76	1216
Weighted avg	0.81	0.78	0.78	1216

Berdasarkan hasil *classification report* yang tersaji pada Tabel 6, model klasifikasi secara umum menunjukkan performa yang cukup baik dengan tingkat akurasi keseluruhan mencapai 78% dari total 1.216 sampel data uji. Angka ini merefleksikan kemampuan generalisasi model yang cukup stabil dalam mengenali berbagai jenis penyakit daun kentang. Secara spesifik, kinerja model terlihat sangat mengesankan dan superior pada kelas *Alternaria*, yang berhasil mencapai nilai sempurna untuk seluruh metrik evaluasi meliputi presisi, *recall*, dan *f1-score*. Hal ini mengindikasikan bahwa fitur visual dari penyakit *Alternaria* dapat diekstraksi dengan sangat optimal oleh arsitektur model. Selain itu, kelas *Bacteria* juga menunjukkan performa yang sangat kuat dengan perolehan *f1-score* sebesar 0.91, menegaskan bahwa model memiliki sensitivitas yang tinggi dalam membedakan penyakit bakteri dari kelas lainnya.

Meskipun demikian, evaluasi ini juga menyoroti adanya disparitas kinerja yang signifikan pada beberapa kelas tertentu. Model terindikasi memiliki kelemahan substansial pada kelas *Fungi*, yang ditunjukkan oleh rendahnya skor *recall* di angka 0.52. Rendahnya *recall* ini bermakna bahwa model sering gagal mendeteksi keberadaan jamur (*false negative*), yang dalam implementasi nyata dapat berisiko fatal karena penyakit tidak tertangani. Permasalahan lain

muncul pada kelas *Phytophthora* dan *Healthy*, di mana nilai presisi yang rendah menunjukkan tingginya tingkat kesalahan prediksi positif (*false positive*); artinya, model sering keliru melabeli daun berpenyakit lain sebagai daun sehat atau *Phytophthora*. Fenomena variasi performa ini juga tidak lepas dari faktor ketidakseimbangan distribusi data (*class imbalance*) pada *dataset*, di mana dominasi kelas mayoritas seperti *Alternaria* memberikan kontribusi besar pada akurasi global, sementara kelas minoritas seperti *Nematode* memiliki representasi yang minim, meskipun secara rata-rata tertimbang (*weighted average*), model masih mampu menjaga stabilitas performanya di angka 0.78.



**Gambar 8.** Confusion Matrix Model

Analisis lebih lanjut menggunakan *Confusion Matrix* yang telah divisualisasikan pada Gambar 8 memberikan rincian distribusi prediksi yang memperjelas letak kebingungan utama model. Matriks tersebut memvalidasi bahwa model bekerja tanpa cela pada kelas *Alternaria*, namun mengungkap pola kesalahan silang (*cross-classification error*) yang masif pada kelas *Fungi*. Kesalahan terbesar model terletak pada misklasifikasi *Fungi* yang diprediksi sebagai *Phytophthora* sebanyak 52 sampel dan sebagai *Pest* sebanyak 33 sampel. Sebaliknya, kelas *Pest* juga kerap tertukar dengan *Fungi* dan *Phytophthora*. Tingginya angka kebingungan antara *Fungi*, *Phytophthora*, dan *Pest* ini mengindikasikan adanya kemiripan fitur morfologi visual seperti bentuk bercak, warna lesi, atau tekstur kerusakan daun yang sangat signifikan di antara ketiga kelas tersebut. Hal ini menyebabkan *decision boundary* yang dibentuk oleh model menjadi kurang tegas dalam memisahkan ketiga penyakit ini, berbeda dengan kelas *Alternaria* atau *Nematode* yang memiliki ciri visual yang jauh lebih distingtif dan mudah dikenali.

#### 4. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan dan mengevaluasi kinerja algoritma *Convolutional Neural Network (CNN)* dengan arsitektur *MobileNetV2* untuk klasifikasi penyakit daun tanaman kentang, dengan fokus utama pada penerapan optimasi *Randomize Search* dan validasi model menggunakan *K-Fold Cross Validation*. Berdasarkan serangkaian eksperimen yang komprehensif, dapat disimpulkan bahwa pendekatan *transfer learning* menggunakan *MobileNetV2* mampu memberikan hasil yang efektif dan stabil dalam mengenali delapan kelas kondisi daun yang berbeda. Melalui proses pencarian *hyperparameter* secara otomatis, ditemukan bahwa kombinasi *optimizer RMSprop*, *learning rate* 0.001, dan *batch size* 64 pada iterasi ke-15 merupakan konfigurasi paling optimal, yang menghasilkan akurasi pada data uji sebesar 78,45%. Nilai ini sangat mendekati akurasi validasi terbaik sebesar 79,30%, yang membuktikan bahwa model memiliki kemampuan generalisasi yang sangat baik dan berhasil menghindari masalah *overfitting* yang sering terjadi pada dataset terbatas. Secara spesifik, model menunjukkan keunggulan luar biasa dalam mendeteksi penyakit *Alternaria* dengan tingkat keberhasilan sempurna pada seluruh metrik evaluasi. Namun, penelitian ini juga mengidentifikasi keterbatasan model dalam membedakan penyakit *Fungi*, yang memiliki nilai *recall* rendah sebesar 52% akibat tingginya tingkat kemiripan fitur visual morfologis dengan penyakit *Phytophthora* dan *Pest*, sebagaimana terlihat pada analisis *confusion matrix*. Penggunaan teknik *balanced class weight* terbukti membantu menyeimbangkan pelatihan namun belum sepenuhnya menghilangkan bias pada kelas dengan karakteristik visual yang serupa. Oleh karena itu disarankan untuk penelitian selanjutnya agar bisa memperbanyak iterasi dalam menggunakan *randomize search* untuk mencari kombinasi parameter yang mampu memberikan hasil yang lebih baik, menambahkan data pada *class* minoritas agar dapat mengurangi bias pada kelas minoritas serta model bisa membedakan fitur unik dari kelas minoritas lebih baik. Hasil penelitian ini menunjukkan potensi yang besar untuk dikembangkan pada perangkat dengan *resource* terbatas, seperti *smartphone* sehingga dapat mendukung kinerja yang efisien pada bidang industri pertanian.



## REFERENCES

- Abdelhamid, M., & Desai, A. (2024). Balancing the Scales: A Comprehensive Study on Tackling Class Imbalance in Binary Classification. *ArXiv Preprint ArXiv:2409.19751*. <https://doi.org/10.48550/arXiv.2409.19751>
- Agustiani, S., Aryanti, R., Wildah, S. K., Arifin, Y. T., Marlina, S., & Misriati, T. (2024). Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2. *Journal of Informatics Management and Information Technology*, 4(4), 150–157. <https://doi.org/10.47065/jimat.v4i4.411>
- Aisya, J. R., & Prasatiadi, A. (2023). Klasifikasi Penyakit Daun Kentang dengan Metode CNN dan RNN. *Jurnal Tekno Insentif*, 17(1), 1–10. <https://doi.org/10.36787/jti.v17i1.888>
- Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. In *Journal of Engineering and Applied Sciences* (Vol. 12, Issue 16, pp. 4102–4107). <https://doi.org/10.3923/jeasci.2017.4102.4107>
- Amatullah, L., Ein, I., & Santoni, M. M. (2021). Identifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur dan Warna Dengan Menggunakan Metode K-Nearest Neighbor. *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer Dan Aplikasinya*, 2(1), 783–791. <https://conference.upnvj.ac.id/index.php/senamika/article/view/1440>
- Angeline, A., & Kusniyati, H. (2024). Komparasi Performa VGG19, ResNet50, DenseNet121 dan MobileNetV2 Dalam Mendeteksi Gambar Deepfake. *CESS (Journal of Computer Engineering, System and Science)*, 9(2), 397. <https://doi.org/10.24114/cess.v9i2.58671>
- Asif, M. K. R., Rahman, M. A., & Hena, M. H. (2020). CNN based disease detection approach on potato leaves. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 428–432. <https://doi.org/10.1109/ICISS49785.2020.9316021>
- Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*, 14(4), 3609–3620. <https://doi.org/10.1007/s12652-021-03488-z>
- Fuadi, A., & Suharso, A. (2022). Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang. In *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7 (3). <https://doi.org/10.29100/jupi.v7i3.3026>
- Ghandi, S., Ma'ruf Nugroho, I., Raymond, Y., Informatika, R. T., Tinggi, S., Wastukencana, T., Cikopak, J., 53, N., Babakancikao, K., Purwakarta, K., & Barat, J. (2024). Penerapan Metode Convolutional Neural Network (CNN) Dalam Aplikasi Pendeteksi Penyakit Daun Tanaman Kentang Berbasis Android. *Jurnal Mahasiswa Teknik Informatika*, 8(5). <https://doi.org/https://doi.org/10.36040/jati.v8i5.10769>
- Hidayat, W. F., Asra, T., & Setiadi, A. (2022). Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression. *Indonesian Journal on Software Engineering (IJSE)*, 8(2), 173–179. <https://doi.org/10.31294/ijse.v8i2.14624>
- Lesmana, A. M., Fadhillah, R. P., & Rozikin, C. (2022). Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN). *Jurnal Sains Dan Informatika*, 8(1), 21–30. <https://doi.org/10.34128/jsi.v8i1.377>
- Negoro, N. K., Utami, E., & Yaqin, A. (2023). Klasifikasi Deteksi Penggunaan Masker Menggunakan Metode Convolutional Neural Network. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(2), 664–674. <https://doi.org/10.29100/jupi.v8i2.3748>
- Permana, R., Saldu, H., & Maulana, D. I. (2022). Optimasi Image Classification Pada Jenis Sampah Dengan Data Augmentation Dan Convolutional Neural Network. *Jurnal Sistem Informasi Dan Informatika (Simika)*, 5(2), 111–120. <https://doi.org/10.47080/simika.v5i2.1913>
- Rachmatsyah, A. D., Sugihartono, T., & Irfan, K. (2024). Perbandingan Teknik Optimasi Grid Search dan Randomized Search dalam Meningkatkan Akurasi Metode Klasifikasi SVM Pada Sentimen Ulasan Perbandingan Teknik Optimasi Grid Search dan Randomized Search dalam Meningkatkan Akurasi Metode Klasifikasi SVM Pada Sen. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, 8(January). <https://doi.org/10.36080/skanika.v8i1.3328>
- Ridwansyah, T. (2022). Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 2(5), 178–185. <https://doi.org/10.30865/klik.v2i5.362>
- Rozaqi, A. J., Sunyoto, A., & Arief, R. (2021). Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network Detection of PotaAto Leaves Disease Using Image Processing with Convolutional Neural Network Methods. *Creative Information Technology Journal (CITEC Journal)*, 8(1). <https://doi.org/https://doi.org/10.24076/citec.2021v8i1.263>
- Shabrina, N. H., Indarti, S., Maharani, R., Kristiyanti, D. A., Irmawati, I., Prastomo, N., & M, T. A. (2023). A novel dataset of potato leaf disease in uncontrolled environment. *Mendeley Data*. <https://doi.org/10.17632/ptz377bwb8.1>
- Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE*, 17(2 February), 1–21. <https://doi.org/10.1371/journal.pone.0264586>
- Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2022). *Image Data Augmentation for Deep Learning: A Survey*. *ArXiv Preprint*. <https://doi.org/10.13328/j.cnki.jos.007263>