**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

# Bayesian Optimized Pretrained CNNs for Mango Leaf Disease Classification: A Comparative Study

**Sri Rahayu*[1], Sayyid Faruk Romdoni[2]**

[1,2]Informatics, Institut Teknologi Garut, Indonesia

Email: [1]srirahayu@itg.ac.id

Received : Jun 30, 2025; Revised : Aug 23, 2025; Accepted : Aug 29, 2025; Published : Oct 16, 2025

**Abstract**

Mango leaf diseases pose a major threat to crop productivity, causing significant economic losses for farmers. Accurate and early detection is essential, yet manual diagnosis remains subjective and inefficient. This study aims to evaluate and compare the performance of five pretrained Convolutional Neural Network (CNN) architectures—DenseNet121, ResNet50V2, MobileNetV3 Small, MobileNetV3 Large, and InceptionV3—by systematically optimizing their hyperparameters to identify the most effective model for mango leaf disease classification. The public MangoLeafBD dataset, containing 4,000 images from eight balanced classes, was used. Bayesian Optimization was applied to fine-tune each model, and their performances were assessed before and after optimization. Results show that optimization substantially improved all models, with MobileNetV3 Large achieving the highest accuracy of 100% on the test set, followed by DenseNet121 (99.75%), ResNet50V2 (99.63%), MobileNetV3 Small (99.50%), and InceptionV3 (98.50%). The findings highlight that a well-tuned lightweight model can outperform more complex architectures, offering a practical and efficient solution for developing mobile-based diagnostic tools to support precision agriculture in resource-constrained settings.

***Keywords :*** *Bayesian Optimization, Convolutional Neural Networks, Deep Learning, Mango Leaf Disease, MobileNetV3, Transfer Learning.*

**This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License**

## 1. INTRODUCTION

Mangga (*Mangifera indica L.*) merupakan salah satu komoditas hortikultura unggulan di Indonesia dengan nilai ekonomi yang signifikan. Menurut data Badan Pusat Statistik (BPS), produksi mangga nasional pada tahun 2023 mencapai 33 juta kuintal, menjadikannya salah satu buah dengan produksi tertinggi di tanah air [1]. Secara global, Indonesia menempati posisi keempat sebagai produsen mangga terbesar, setelah India, Tiongkok, dan Thailand [2], [3], [4]. Namun, tingginya produksi ini tidak lepas dari berbagai faktor penghambat, di mana penyakit tanaman menjadi salah satu penyebabnya [5]. Identifikasi penyakit secara manual oleh petani seringkali tidak efektif karena bersifat subjektif, lambat, dan memerlukan keahlian khusus [6], [7], [8], [9], [10]. Ketidaktahuan dan kurangnya informasi mengenai penyakit dapat menyebabkan kerugian besar akibat kerusakan tanaman atau gagal panen [11]. Beberapa penyakit seperti *Anthracnose*, *Bacterial Canker*, dan *Gall Midge* telah dilaporkan menyerang tanaman mangga di Indonesia [12], [13], [14]. Penyakit-penyakit ini berpotensi menurunkan produktivitas dan kualitas buah. Oleh karena itu, pengembangan sistem identifikasi penyakit akurat menjadi sangat krusial.

Dalam satu dekade terakhir, pendekatan berbasis *Deep Learning*, khususnya *Convolutional Neural Network* (CNN), telah diadopsi secara luas dan menunjukkan kinerja yang luar biasa untuk deteksi penyakit tanaman [15], [16], [17]. Sejumlah penelitian telah mengaplikasikan metode ini pada daun mangga dengan berbagai tingkat keberhasilan dan pendekatan. Salah satu studi komprehensif dilakukan oleh Mahmud et al. [18] yang mengevaluasi performa berbagai arsitektur *pretrained* seperti

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

DenseNet121, DenseNet201, ResNet50, MobileNet, dan EfficientNetV2B3. Mereka menemukan bahwa ResNet50 menunjukkan akurasi tertinggi sebesar 99.37%, mengungguli model lain termasuk DenseNet121 (99.1%). Namun, kelemahan signifikan dari studi ini adalah proses optimasi *hyperparameter* yang hanya diterapkan secara manual pada model kustom (DenseNet78) yang mereka kembangkan, sementara arsitektur-arsitektur *pretrained* yang kuat dievaluasi dalam kondisi sub-optimal tanpa *tuning*.

Penelitian lain oleh Rinanda et al. [19] secara spesifik membandingkan tiga arsitektur, yaitu CNN konvensional, VGG16, dan InceptionV3. Hasilnya menunjukkan bahwa model VGG16 dan InceptionV3 mampu mencapai akurasi pelatihan masing-masing sebesar 96.87% dan 96.50%, jauh melampaui CNN konvensional. Meskipun memberikan perbandingan langsung, penelitian ini memiliki keterbatasan dalam hal cakupan model yang diuji dan tidak menyertakan proses *tuning hyperparameter* sama sekali, sehingga potensi maksimal dari setiap arsitektur belum tereksplorasi. Serupa dengan itu, studi oleh Likorawung & Wonohadidjojo [20] yang fokus pada implementasi DenseNet121 berhasil mencatatkan akurasi tinggi sebesar 99.5%, namun kesimpulannya terbatas karena hanya mengevaluasi satu arsitektur tanpa pembanding dan tanpa optimasi.

Beberapa peneliti telah mencoba menerapkan teknik optimasi. Pratap & Kumar [21], misalnya, berhasil meningkatkan akurasi model YOLOv8 hingga 98% menggunakan *African Buffalo Optimization* (ABO). Namun, perbandingannya menjadi kurang seimbang karena model-model *pretrained* lain seperti MobileNet dan VGG16 yang digunakan sebagai pembanding tidak mendapatkan perlakuan optimasi yang sama, sehingga hanya mencapai akurasi di kisaran 89-91%. Di sisi lain, Vijay & Pushpalatha [22] mengusulkan arsitektur gabungan (DV-PSO-Net) dari DenseNet121 dan VGG19 yang dioptimasi menggunakan *Particle Swarm Optimization* (PSO) dan mencapai akurasi 94.72%. Pendekatan ini menarik, namun fokusnya adalah pada validitas metode fusi dan optimasi PSO, bukan pada evaluasi sistematis terhadap performa individual dari berbagai arsitektur standar.

Dari tinjauan mendalam ini, sebuah pola yang menjadi kesenjangan penelitian utama (*research gap*) dapat diidentifikasi dengan jelas. Pertama, sebagian besar studi belum melakukan evaluasi komparatif yang menyeluruh terhadap berbagai model populer secara sistematis dalam kondisi yang setara. Kedua, dan yang paling krusial, adalah kurangnya optimasi sistematis pada arsitektur *pretrained* CNN, di mana strategi *tuning hyperparameter* seringkali diabaikan. Sebagai perbandingan spesifik, penelitian [19] yang membandingkan CNN konvensional, VGG16, dan InceptionV3, tidak menyertakan proses *hyperparameter tuning* sama sekali. Akibatnya, perbandingan yang mereka sajikan hanya sebatas performa *default* dan belum tentu mencerminkan potensi maksimal dari arsitektur tersebut. Pola serupa, di mana optimasi hanya dilakukan secara manual dan tidak sistematis atau bahkan diabaikan, juga terlihat pada studi oleh [18], [20]. Hal ini menyebabkan performa yang dilaporkan belum tentu mencerminkan potensi maksimal dari setiap arsitektur, karena parameter *default* mungkin tidak ideal untuk setiap masalah klasifikasi citra yang spesifik [23]. Selain itu, masih sangat minim eksplorasi pendekatan optimasi adaptif seperti *Bayesian Optimization*, yang secara teoretis terbukti lebih efisien dan sistematis dibandingkan *grid search* maupun *random search* untuk menemukan *hyperparameter* optimal pada model yang kompleks [24], [25].

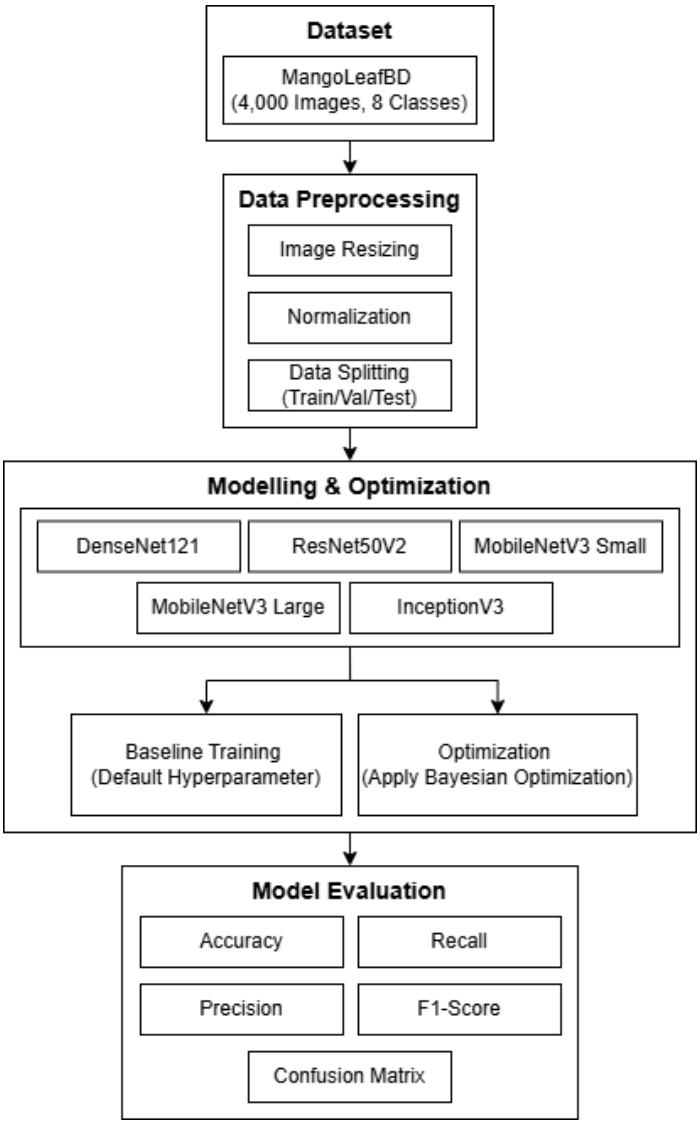
Untuk mengisi kesenjangan tersebut, penelitian ini akan melakukan studi komparatif secara sistematis terhadap lima model *pretrained* CNN (DenseNet121, ResNet50V2, MobileNetV3 Small, MobileNetV3 Large, dan InceptionV3), yang dipilih berdasarkan popularitas dan performa mereka di studi sebelumnya. Berbeda dengan studi sebelumnya, kontribusi utama penelitian ini terletak pada penerapan metode Bayesian Optimization secara independen pada kelima arsitektur untuk secara sistematis menemukan konfigurasi optimal yang menyeimbangkan antara akurasi klasifikasi dan efisiensi komputasi. Dengan pendekatan ini, penelitian diharapkan dapat memberikan evaluasi yang lebih adil dan komprehensif mengenai potensi maksimal dari masing-masing arsitektur. Dengan

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

demikian, penelitian ini bertujuan untuk membandingkan dan mengoptimasi lima pretrained CNN menggunakan Bayesian Optimization guna mengidentifikasi model terbaik untuk klasifikasi penyakit daun mangga, sehingga diperoleh solusi yang seimbang antara akurasi dan efisiensi komputasi.

## 2. METHOD

Penelitian ini dilakukan melalui serangkaian tahapan sistematis yang mencakup persiapan dataset, perancangan model, proses optimasi, dan evaluasi performa. Alur kerja penelitian secara umum diilustrasikan pada Gambar 1.

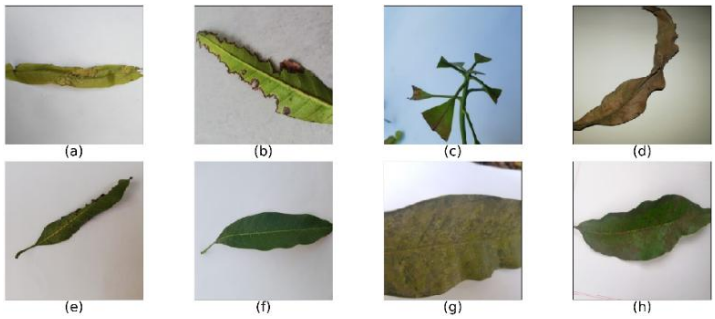



Gambar 1. Kerangka Penelitian

### 2.1. Dataset dan Pra-pemrosesan Citra

Penelitian ini memanfaatkan dataset publik MangoLeafBD yang tersedia di platform Kaggle [26]. Dataset ini terdiri dari 4.000 citra daun mangga yang terbagi rata ke dalam delapan kelas (masing-masing 500 citra): tujuh kelas penyakit (*Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Sooty Mould*) dan satu kelas daun sehat. Keseimbangan kelas ini ideal untuk mencegah potensi bias pada model selama pelatihan. Gambar 2 berikut menyajikan contoh citra dari tiap kelas, baik kategori daun sehat maupun yang mengandung penyakit.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967




Gambar 2. Sampel Data. (a) *Anthracnose* (b) *Bacterial Canker* (c) *Cutting Weevil* (d) Die Back (e) Gall Midge (f) Healthy (g) *Powdery Mildew* (h) *Sooty Mould*

Sebelum digunakan, dataset melalui beberapa tahap persiapan. Pertama, dataset dibagi menjadi data latih (80%) dan data uji (20%). Selanjutnya, 10% dari data latih dialokasikan sebagai data validasi untuk digunakan selama proses optimasi. Sehingga, untuk setiap kelas, distribusinya adalah 360 citra untuk pelatihan, 40 citra untuk validasi, dan 100 citra untuk pengujian. Tahap pra-pemrosesan selanjutnya adalah *resizing* dan normalisasi. Semua citra diseragamkan ukurannya agar sesuai dengan rekomendasi dimensi input untuk masing-masing arsitektur yang didapatkan dari dokumentasi Tensorflow, untuk ukuran gambar setiap arsitektur dirinci pada Tabel 1. Setelah itu, dilakukan normalisasi nilai piksel menggunakan fungsi preprocess_input dari TensorFlow untuk mengubah skala piksel ke rentang yang sesuai pada setiap arsitektur *pretrained* model.

Tabel 1. Ukuran Input Citra untuk Setiap Model

| Model | Size |
|---|---|
| DenseNet121 | (224, 224) |
| ResNet50V2 | (224, 224) |
| MobileNetV3 (Small & Large) | (224, 224) |
| InceptionV3 | (299, 299) |

## 2.2. Arsitektur Model dan Konfigurasi Pelatihan

Penelitian ini melakukan evaluasi komparatif terhadap lima arsitektur *pretrained* CNN. Setiap model diinisialisasi menggunakan bobot dari ImageNet (*transfer learning*) untuk memanfaatkan fitur-fitur yang telah dipelajari. Lapisan konvolusi dari setiap model dibekukan (*frozen*) dan hanya lapisan klasifikasi akhir yang dilatih ulang.

Salah satu arsitektur yang digunakan adalah DenseNet121, sebuah jaringan dengan 121 lapisan, termasuk 108 convolutional layer, yang dirancang untuk menghubungkan setiap lapisan dengan seluruh lapisan sebelumnya. Pola konektivitas ini mendukung *feature reuse* serta mengurangi jumlah parameter dalam model [27]. Dalam implementasinya, setiap layer dalam DenseNet menerima output dari semua layer sebelumnya, kecuali layer pertama yang menerima input gambar. Desain ini menghasilkan total L koneksi langsung untuk L layer, menciptakan efisiensi dan aliran informasi yang optimal dalam jaringan [28]. Model lain yang digunakan adalah ResNet50V2, versi lanjutan dari Residual Networks yang memanfaatkan residual block untuk mengatasi masalah vanishing atau exploding gradient. ResNet50V2 mengintegrasikan shortcut connection dan identity block yang membantu proses pembelajaran representasi fitur yang lebih dalam dan stabil [29]. Arsitektur ini terdiri dari 50 lapisan, dan dikenal dengan kemampuannya untuk mencapai konvergensi lebih cepat dan akurasi lebih tinggi dibandingkan CNN konvensional [30].

MobileNetV3, yang juga dievaluasi dalam penelitian ini, merupakan jaringan ringan yang dirancang khusus untuk efisiensi pada perangkat mobile. Arsitektur ini merupakan kombinasi dari

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

pendekatan *deep separable convolution* dan *inverted residual structure* dengan *linear bottleneck*, serta modul SE (Squeeze-and-Excitation) dan aktivasi non-linear h-swish [31]. MobileNetV3 hadir dalam dua varian: Small dan Large, yang memiliki struktur dasar serupa namun dengan kompleksitas berbeda. MobileNetV3-Small dirancang untuk perangkat dengan kemampuan komputasi rendah seperti perangkat embedded [32], [33]. Selain itu, penelitian ini juga menggunakan InceptionV3, arsitektur yang dikembangkan oleh Google dan merupakan kelanjutan dari InceptionV1 dan V2. Model ini menggunakan auxiliary classifier untuk mendistribusikan sinyal label ke bagian jaringan yang lebih dalam dan juga berfungsi sebagai regularisasi tambahan [34]. InceptionV3 dilatih pada dataset ImageNet dan memiliki kemampuan klasifikasi hingga 1000 kelas [35].

Seluruh eksperimen diimplementasikan menggunakan bahasa pemrograman Python di platform Google Colaboratory yang dilengkapi dengan T4 GPU. Untuk skenario pelatihan *baseline*, semua model dilatih selama 20 *epoch* menggunakan *Optimizer Stochastic Gradient Descent* (SGD). Mekanisme *EarlyStopping* dengan patience=5 juga diterapkan untuk menghentikan pelatihan jika tidak ada peningkatan akurasi pada data validasi, guna mencegah *overfitting*.

### 2.3. Optimasi Hyperparameter dengan Bayesian Optimization

Untuk menemukan konfigurasi *hyperparameter* yang optimal, penelitian ini menerapkan *Bayesian Optimization*. Metode ini dipilih karena efisiensinya dibandingkan pendekatan lain seperti *grid search* atau *random search* [24]. Bayesian Optimization (BO) dapat digunakan untuk memilih masukan yang menghasilkan keluaran terbaik dari sekumpulan kandidat masukan yang telah ditentukan sebelumnya [36]. Efektivitas BO, bergantung pada dua komponen utama:

1) *Surrogate Model*: Sebuah model probabilistik, umumnya *Gaussian Process* (GP), yang meniru fungsi objektif dan memberikan prediksi beserta estimasi ketidakpastiannya [37]. GP memodelkan fungsi objektif $f(x)$ sebagai:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \qquad (1)$$

di mana $m(x)$adalah *mean function* dan $k(x, x')$ adalah *kernel function* yang mengukur kemiripan antar titik hyperparameter.

2) *Acquisition Function*: Bayesian Optimization memanfaatkan *acquisition function* untuk mengevaluasi efektivitas suatu kombinasi *hyperparameter* terhadap *surrogate model* [38]. Penelitian ini menggunakan *acquisition function* Upper Confidence Bound (UCB) dengan rumus sebagai berikut:

$$a_{UCB}(x; \beta) = \mu(x) + \beta\sigma(x) \qquad (2)$$

di mana $\mu(x)$ dan $\sigma(x)$ masing-masing adalah *mean* dan *standard deviation* hasil prediksi GP. Parameter $\beta$ mengatur keseimbangan antara eksplorasi dan eksploitasi.

Proses optimasi pada penelitian ini dijalankan sebanyak 30 *trial* untuk setiap model. Ruang pencarian *hyperparameter* yang dieksplorasi dirangkum dalam Tabel 2.

Tabel 2. Ruang Pencarian Hyperparameter untuk Optimasi

| Hyperparameter | Range Nilai |
|---|---|
| Learning Rate | 1e-5 hingga 1e-2 (skala logaritmik) |
| Optimizer | RMSprop, SGD, Adam |
| Dropout Rate | 0.0 hingga 0.5 (step 0.1) |
| Dense Unit | 32 hingga 512 (step 32) |

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

### 2.4. Metrik Evaluasi

Kinerja akhir dari setiap model dievaluasi pada data uji. Metrik yang digunakan meliputi Akurasi, Presisi, *Recall*, dan F1-*Score*, yang dihitung berdasarkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Selain itu, *Confusion Matrix* digunakan untuk analisis performa per kelas secara mendalam. Berikut adalah penjelasan lebih lanjut mengenai masing-masing metrik yang digunakan dalam penelitian ini:

1) Akurasi (*Accuracy*): mengukur seberapa banyak prediksi model yang benar dibandingkan dengan total jumlah sampel. Rumusnya adalah:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

2) Presisi (*Precision*): menunjukkan persentase prediksi positif yang benar dibandingkan dengan total prediksi positif yang dihasilkan model. Presisi dihitung dengan rumus berikut:

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

3) *Recall* (Sensitivitas): mengukur seberapa banyak data positif yang diklasifikasikan dengan benar dibandingkan dengan total jumlah data positif yang sebenarnya. Nilai *recall* dihitung dengan:

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

4) *F1-score*: merupakan rata-rata harmonik antara *precision* dan *recall*, yang memberikan keseimbangan antara keduanya. Rumusnya adalah:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (6)$$

5) *Confusion matrix*: merupakan tabel yang menunjukkan jumlah prediksi yang benar dan salah berdasarkan kategori yang ada. *Confusion matrix* menyajikan ringkasan hasil prediksi model dengan membandingkan jumlah klasifikasi yang benar dan salah untuk setiap kategori. Dengan demikian, tidak hanya tingkat kesalahan yang dapat diketahui, tetapi juga jenis kesalahan yang terjadi dalam prediksi model.

## 3. RESULT

Pada bagian ini, disajikan secara objektif dan mendalam hasil dari serangkaian eksperimen yang telah dilakukan. Penyajian mencakup evaluasi performa kuantitatif, visualisasi proses pelatihan, serta analisis performa per kelas untuk setiap arsitektur pada dua skenario: *baseline* (sebelum optimasi) dan setelah optimasi *hyperparameter*.

### 3.1. Performa Model Baseline

Tahap ini bertujuan untuk mengukur performa dasar dari setiap arsitektur *pretrained* CNN sebelum dilakukan optimasi. Konfigurasi pelatihan yang digunakan mengacu pada pengaturan standar yang telah dijelaskan pada bagian Metodologi.

#### 3.1.1. DenseNet121

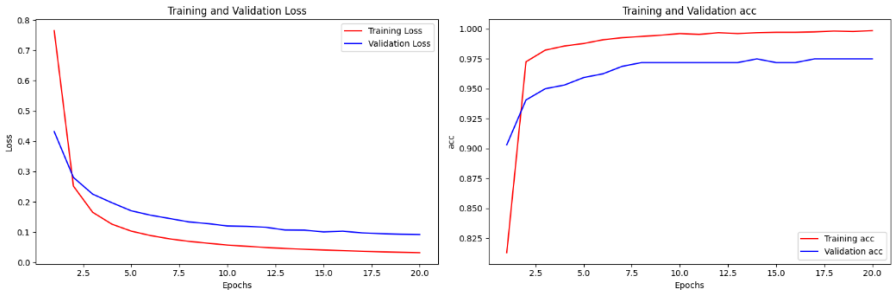
Evaluasi pada model DenseNet121 dengan konfigurasi *baseline* menghasilkan performa klasifikasi yang sangat baik, yang menjadi titik acuan penting untuk perbandingan. Ringkasan hasil performa disajikan pada Tabel 3.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Tabel 3. Ringkasan Evaluasi Model DenseNet121 (Baseline)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.99125 |
| Precision | 0.99143 |
| Recall | 0.99125 |
| F1-Score | 0.99127 |
| Training Time (Seconds) | 199.42 |
| Model Size (MB) | 28.14 |
| Accuracy | 0.99125 |

Berdasarkan Tabel 3, model DenseNet121 pada baseline mencapai akurasi sebesar 99.13% dengan F1-score 0.99127, menunjukkan kemampuan konsisten dalam mengenali pola penyakit. Waktu pelatihan relatif cepat (199.42 detik) dan ukuran model efisien (28.14 MB), menjadikannya kandidat potensial untuk klasifikasi penyakit daun mangga. Untuk menganalisis proses pelatihan, Gambar 3 menampilkan kurva *loss* dan akurasi. Pada grafik *loss* (kiri), terlihat bahwa nilai *training loss* (merah) dan *validation loss* (biru) sama-sama menurun secara stabil, dengan celah yang sangat kecil di antara keduanya. Hal ini mengindikasikan bahwa model belajar dengan baik dari data latih tanpa mengalami *overfitting* yang signifikan. Tren serupa juga terlihat pada kurva akurasi (kanan), di mana akurasi validasi mengikuti akurasi pelatihan hingga mencapai tingkat yang sangat tinggi dan stabil, yang mengonfirmasi kemampuan generalisasi model yang baik.




Gambar 3: Kurva Pembelajaran DenseNet121 Baseline

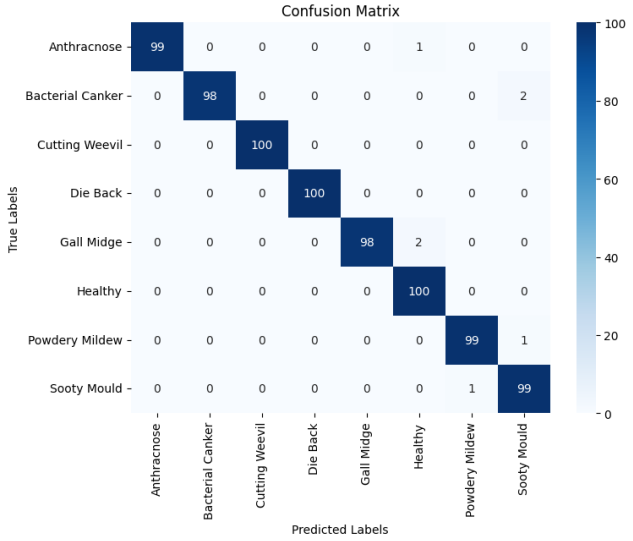
Hasil ini kemudian diperkuat oleh laporan klasifikasi (Gambar 4) dan *confusion matrix* (Gambar 5). Laporan klasifikasi menunjukkan bahwa model mencapai nilai presisi, recall, dan F1-score yang sangat tinggi (umumnya ≥ 0.98) untuk hampir semua kelas. Kelas 'Cutting Weevil' dan 'Die Back' bahkan berhasil diklasifikasikan dengan sempurna (1.00).

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 0.99 | 0.99 | 100 |
| Bacterial Canker | 1.00 | 0.98 | 0.99 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 1.00 | 1.00 | 1.00 | 100 |
| Gall Midge | 1.00 | 0.98 | 0.99 | 100 |
| Healthy | 0.97 | 1.00 | 0.99 | 100 |
| Powdery Mildew | 0.99 | 0.99 | 0.99 | 100 |
| Sooty Mould | 0.97 | 0.99 | 0.98 | 100 |
| accuracy | | | 0.99 | 800 |
| macro avg | 0.99 | 0.99 | 0.99 | 800 |
| weighted avg | 0.99 | 0.99 | 0.99 | 800 |

Gambar 4: Laporan Klasifikasi DenseNet121 Baseline

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

*Confusion matrix* pada Gambar 5 mengonfirmasi temuan ini. Diagonal utama menunjukkan jumlah prediksi benar yang sangat tinggi. Tercatat total hanya 7 misklasifikasi dari 800 sampel uji. Kesalahan klasifikasi minor teridentifikasi, seperti dua sampel 'Bacterial Canker' yang salah diklasifikasikan sebagai 'Sooty Mould' dan dua sampel 'Gall Midge' sebagai 'Healthy'. Jumlah kesalahan yang sangat kecil ini konsisten dengan metrik performa tinggi yang telah dibahas dan menegaskan bahwa model DenseNet121 *baseline* memiliki kemampuan diskriminatif yang sangat baik antar kelas.




Gambar 5: Confusion Matrix DenseNet121 Baseline

### 3.1.2. ResNet50V2

Tahap evaluasi selanjutnya dilakukan pada arsitektur ResNet50V2 dalam kondisi *baseline*. Hasil performa dasar model ini, yang akan menjadi titik referensi untuk perbandingan, disajikan pada Tabel 4.

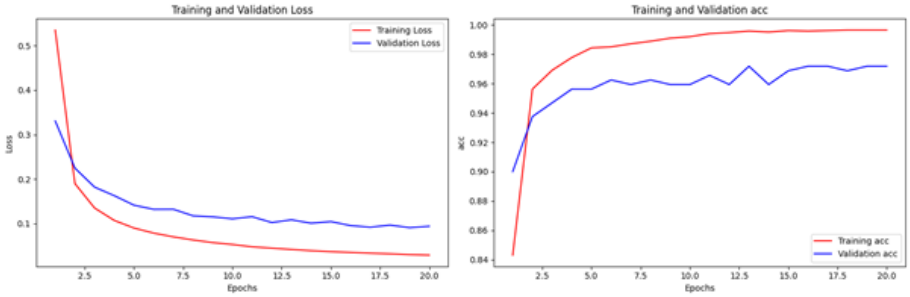
Tabel 4. Ringkasan Evaluasi Model ResNet50V2 (Baseline)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.98125 |
| Precision | 0.98126 |
| Recall | 0.98125 |
| F1-Score | 0.98124 |
| Training Time (Seconds) | 176.57 |
| Model Size (MB) | 90.51 |

Berdasarkan Tabel 4, metrik-metrik awal ini menunjukkan bahwa model ResNet50V2 telah mampu mencapai tingkat performa yang tinggi, dengan akurasi dan F1-score berada di atas 98%. Waktu pelatihan yang tercatat relatif cepat (176.57 detik), namun perlu dicatat bahwa ukuran modelnya (90.51 MB) secara signifikan lebih besar dibandingkan beberapa arsitektur lain.

Dinamika proses pelatihan divisualisasikan pada Gambar 6. Grafik *loss* (kiri) menunjukkan *training loss* (merah) yang menurun secara konsisten, menandakan proses pembelajaran yang efektif. *Validation loss* (biru) juga menunjukkan tren penurunan meskipun dengan sedikit fluktuasi. Pada grafik akurasi (kanan), terdapat sedikit celah antara akurasi pelatihan dan validasi, namun hal ini masih dalam batas wajar dan secara keseluruhan kurva menunjukkan bahwa model telah mencapai konvergensi yang baik tanpa *overfitting* yang parah.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967



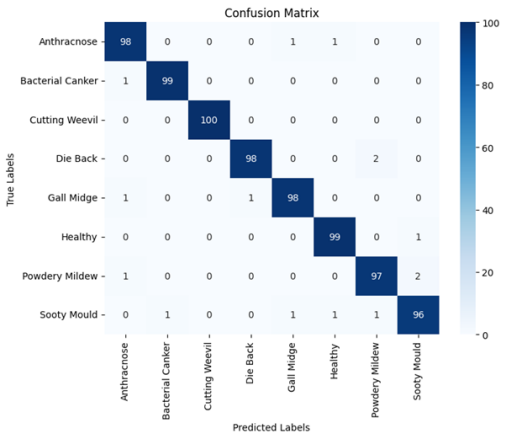

Gambar 6: Kurva Pembelajaran ResNet50V2 Baseline

Untuk analisis performa per kelas yang lebih mendalam, disajikan laporan klasifikasi pada Gambar 7. Secara umum, model menunjukkan performa yang sangat baik. Kelas 'Cutting Weevil' mencapai performa sempurna di semua metrik. Namun, beberapa kelas seperti 'Sooty Mould' menunjukkan F1-score yang sedikit lebih rendah (0.96), yang mengindikasikan adanya tantangan dalam mengklasifikasikan kelas tersebut.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 0.97 | 0.98 | 0.98 | 100 |
| Bacterial Canker | 0.99 | 0.99 | 0.99 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 0.99 | 0.98 | 0.98 | 100 |
| Gall Midge | 0.98 | 0.98 | 0.98 | 100 |
| Healthy | 0.98 | 0.99 | 0.99 | 100 |
| Powdery Mildew | 0.97 | 0.97 | 0.97 | 100 |
| Sooty Mould | 0.97 | 0.96 | 0.96 | 100 |
| accuracy | | | 0.98 | 800 |
| macro avg | 0.98 | 0.98 | 0.98 | 800 |
| weighted avg | 0.98 | 0.98 | 0.98 | 800 |

Gambar 7: Laporan Klasifikasi ResNet50V2 Baseline

*Confusion matrix* pada Gambar 8 memberikan visualisasi detail dari hasil klasifikasi. Meskipun sebagian besar sampel diklasifikasikan dengan benar, teridentifikasi total 15 misklasifikasi dari 800 sampel uji. Kesalahan klasifikasi ini tersebar di beberapa kelas, namun yang paling menonjol adalah pada kelas 'Sooty Mould', di mana 4 sampelnya salah diklasifikasikan ke beberapa kelas lain. Observasi ini sejalan dengan nilai F1-score yang lebih rendah untuk kelas tersebut pada laporan klasifikasi.




Gambar 8: Confusion Matrix ResNet50V2 Baseline

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

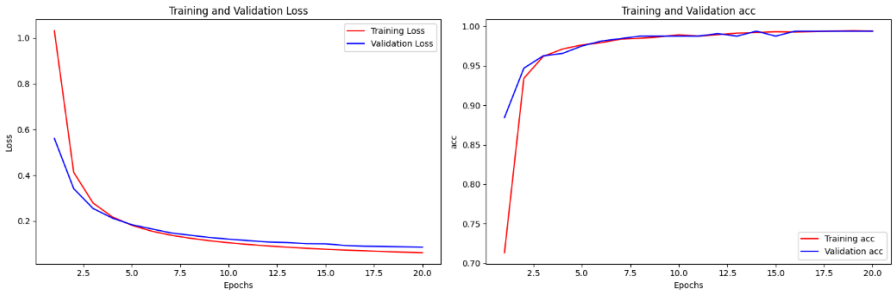
### 3.1.3. MobileNetV3 Small

Selanjutnya, evaluasi dilakukan pada arsitektur *lightweight* MobileNetV3 Small. Tabel 5 menyajikan performa model pada konfigurasi *baseline*.

Tabel 5. Ringkasan Evaluasi Model MobileNetV3 Small (Baseline)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.99125 |
| Precision | 0.99144 |
| Recall | 0.99125 |
| F1-Score | 0.99118 |
| Training Time (Seconds) | 113.29 |
| Model Size (MB) | 4.07 |

Berdasarkan Tabel 5, model ini berhasil mencapai akurasi 99.13%, setara dengan performa DenseNet121. Keunggulan signifikan dari model ini adalah efisiensi sumber dayanya, dengan waktu pelatihan yang sangat cepat (113.29 detik) dan ukuran model yang sangat kecil (4.07 MB), menjadikannya kandidat yang menarik untuk implementasi pada perangkat dengan keterbatasan komputasi. Kurva pembelajaran pada Gambar 9 menunjukkan proses pelatihan yang ideal. Baik kurva *loss* maupun akurasi untuk data latih dan validasi berjalan sangat berdekatan, yang mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik dan tidak mengalami *overfitting*.




Gambar 9: Kurva Pembelajaran MobileNetV3 Small Baseline

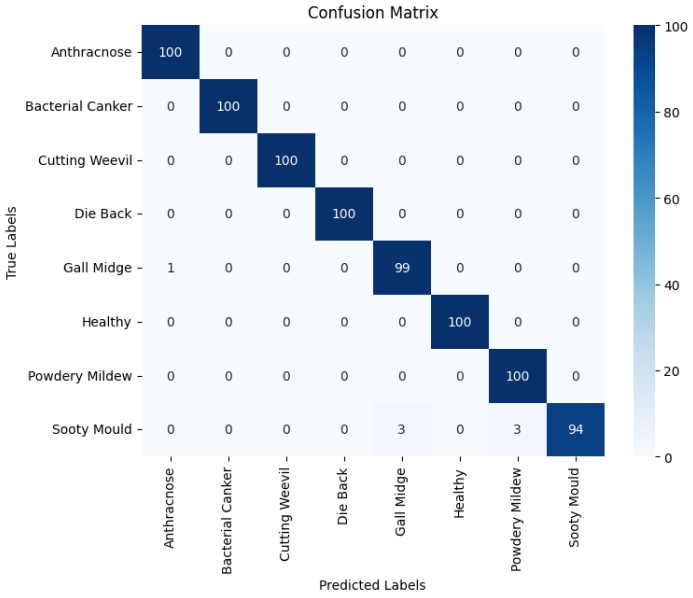
Laporan klasifikasi pada Gambar 10 merinci performa model untuk setiap kelas. Performa yang ditunjukkan sangat mengesankan, dengan sebagian besar kelas mencapai nilai F1-score 0.99 atau 1.00. Namun, terdapat sedikit penurunan pada metrik *recall* untuk kelas 'Sooty Mould' (0.94), yang menandakan adanya beberapa sampel dari kelas tersebut yang tidak berhasil diidentifikasi dengan benar oleh model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 0.99 | 1.00 | 1.00 | 100 |
| Bacterial Canker | 1.00 | 1.00 | 1.00 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 1.00 | 1.00 | 1.00 | 100 |
| Gall Midge | 0.97 | 0.99 | 0.98 | 100 |
| Healthy | 1.00 | 1.00 | 1.00 | 100 |
| Powdery Mildew | 0.97 | 1.00 | 0.99 | 100 |
| Sooty Mould | 1.00 | 0.94 | 0.97 | 100 |
| accuracy | | | 0.99 | 800 |
| macro avg | 0.99 | 0.99 | 0.99 | 800 |
| weighted avg | 0.99 | 0.99 | 0.99 | 800 |

Gambar 10: Laporan Klasifikasi MobileNetV3 Small Baseline

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Temuan ini diperkuat oleh *confusion matrix* pada Gambar 11, yang menyajikan visualisasi detail dari hasil klasifikasi. Tercatat total 7 misklasifikasi dari 800 sampel uji. Mayoritas kesalahan ini (6 dari 7) terjadi pada kelas 'Sooty Mould', yang salah diklasifikasikan sebagai 'Gall Midge' dan 'Powdery Mildew'. Hal ini konsisten dengan nilai *recall* yang lebih rendah untuk kelas tersebut dan menunjukkan adanya tantangan spesifik bagi model dalam membedakan 'Sooty Mould' dari penyakit lain dengan karakteristik visual serupa.




Gambar 11: Confusion Matrix MobileNetV3 Small Baseline

### 3.1.4. MobileNetV3 Large

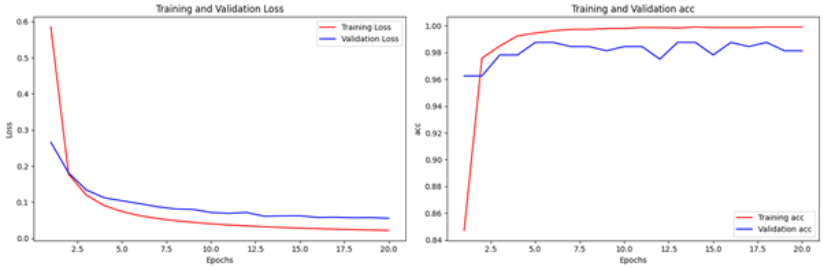
Evaluasi selanjutnya dilakukan pada varian yang lebih besar dari keluarga MobileNet, yaitu MobileNetV3 Large. Tabel 6 menyajikan performa model pada konfigurasi *baseline*.

Tabel 6. Ringkasan Evaluasi Model MobileNetV3 Large (Baseline)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.995 |
| Precision | 0.99502 |
| Recall | 0.995 |
| F1-Score | 0.99499 |
| Training Time (Seconds) | 129.94 |
| Model Size (MB) | 12.03 |

Hasil performa *baseline* dari MobileNetV3 Large sangatlah mengesankan. Berdasarkan Tabel 6, model ini berhasil mencapai akurasi 99.50%, yang merupakan performa tertinggi di antara semua model *baseline* yang diuji. Selain itu, model ini tetap mempertahankan efisiensi yang menjadi ciri khas keluarga MobileNet, dengan waktu pelatihan yang cepat (129.94 detik) dan ukuran model yang relatif kecil (12.03 MB). Dinamika proses pelatihan yang ditampilkan pada Gambar 12 menunjukkan proses yang sangat stabil. Kurva *loss* (kiri) untuk data latih dan validasi menurun secara konsisten dan konvergen pada nilai yang sangat rendah. Demikian pula, kurva akurasi (kanan) untuk kedua set data meningkat dengan cepat dan stabil pada tingkat yang sangat tinggi, dengan jarak yang minimal. Hal ini menegaskan kemampuan generalisasi model yang sangat baik.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967



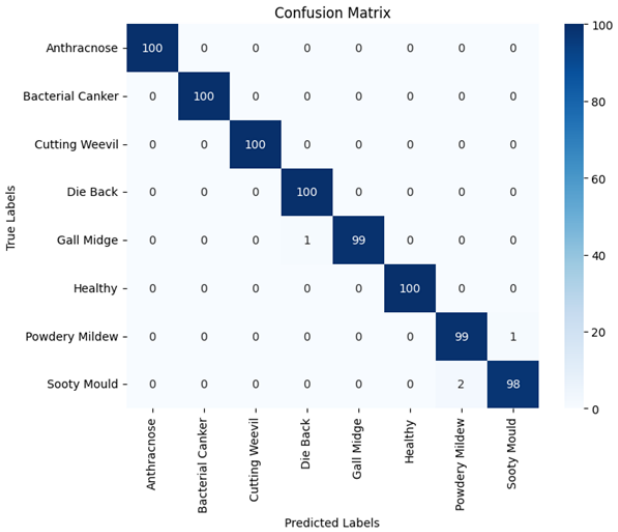

Gambar 12: Kurva Pembelajaran MobileNetV3 Large Baseline

Laporan klasifikasi pada Gambar 13 merinci performa model yang luar biasa untuk setiap kelas. Sebagian besar metrik, seperti presisi, recall, dan F1-score, mencapai nilai sempurna (1.00) atau mendekati sempurna (0.98-0.99), yang mengonfirmasi performa keseluruhan model yang sangat unggul dan seimbang.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 1.00 | 1.00 | 100 |
| Bacterial Canker | 1.00 | 1.00 | 1.00 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 0.99 | 1.00 | 1.00 | 100 |
| Gall Midge | 1.00 | 0.99 | 0.99 | 100 |
| Healthy | 1.00 | 1.00 | 1.00 | 100 |
| Powdery Mildew | 0.98 | 0.99 | 0.99 | 100 |
| Sooty Mould | 0.99 | 0.98 | 0.98 | 100 |
| accuracy | | | 0.99 | 800 |
| macro avg | 1.00 | 0.99 | 0.99 | 800 |
| weighted avg | 1.00 | 0.99 | 0.99 | 800 |

Gambar 13: Laporan Klasifikasi MobileNetV3 Large Baseline

*Confusion matrix* pada Gambar 14 secara visual mengonfirmasi performa tinggi dari model ini. Tercatat total hanya 4 misklasifikasi dari 800 sampel uji, jumlah kesalahan terendah dari semua model *baseline*. Kesalahan minor teridentifikasi antara kelas 'Powdery Mildew' dan 'Sooty Mould', yang mungkin mengindikasikan adanya kemiripan visual. Namun, jumlah kesalahan yang sangat rendah ini menegaskan kemampuan diskriminatif model yang luar biasa bahkan pada konfigurasi *baseline*.




Gambar 14: Confusion Matrix MobileNetV3 Large Baseline

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

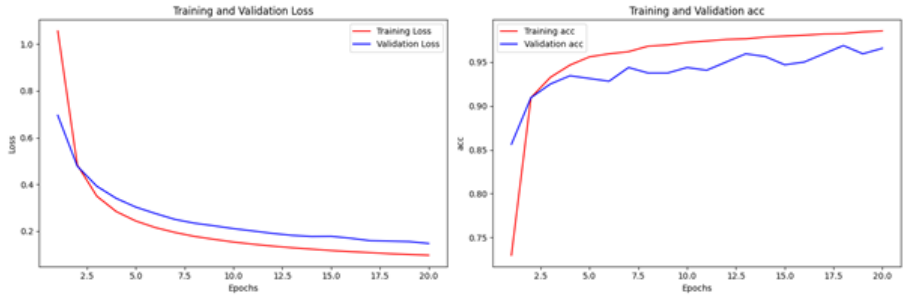
### 3.1.5. InceptionV3

Evaluasi terakhir pada tahap *baseline* dilakukan pada arsitektur InceptionV3. Hasil performa model disajikan pada Tabel 7.

Tabel 7. Ringkasan Evaluasi Model InceptionV3 (Baseline)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.97125 |
| Precision | 0.97126 |
| Recall | 0.97125 |
| F1-Score | 0.97124 |
| Training Time (Seconds) | 299.44 |
| Model Size (MB) | 84.17 |

Berdasarkan Tabel 7, model InceptionV3 pada konfigurasi *baseline* mencatatkan akurasi terendah di antara semua model yang diuji, yaitu 97.13%. Dari segi sumber daya, model ini juga memerlukan waktu pelatihan terlama (299.44 detik) dan memiliki ukuran yang cukup besar (84.17 MB). Kurva pembelajaran pada Gambar 15, meskipun menunjukkan tren penurunan *loss* yang konsisten, memperlihatkan celah yang lebih jelas antara akurasi pelatihan dan validasi dibandingkan model-model lain. Hal ini mengindikasikan kemampuan generalisasi model pada data validasi yang sedikit lebih lemah, meskipun tidak mengalami *overfitting* yang parah.




Gambar 15: Kurva Pembelajaran InceptionV3 Baseline

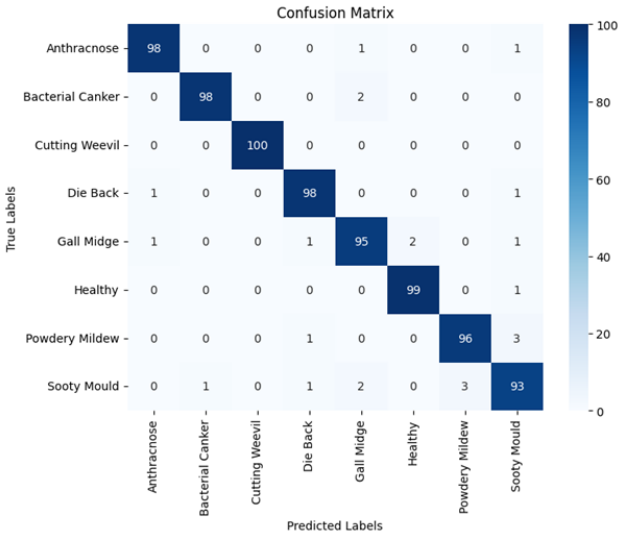
Laporan klasifikasi (Gambar 16) merinci performa model per kelas. Terlihat bahwa beberapa kelas seperti 'Sooty Mould' dan 'Gall Midge' memiliki F1-score yang lebih rendah (masing-masing 0.93 dan 0.95), menandakan tantangan yang lebih besar bagi model dalam mengklasifikasikan kedua kelas ini secara akurat.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 0.98 | 0.98 | 0.98 | 100 |
| Bacterial Canker | 0.99 | 0.98 | 0.98 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 0.97 | 0.98 | 0.98 | 100 |
| Gall Midge | 0.95 | 0.95 | 0.95 | 100 |
| Healthy | 0.98 | 0.99 | 0.99 | 100 |
| Powdery Mildew | 0.97 | 0.96 | 0.96 | 100 |
| Sooty Mould | 0.93 | 0.93 | 0.93 | 100 |
| accuracy | | | 0.97 | 800 |
| macro avg | 0.97 | 0.97 | 0.97 | 800 |
| weighted avg | 0.97 | 0.97 | 0.97 | 800 |

Gambar 16: Laporan Klasifikasi InceptionV3 Baseline

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Analisis pada *confusion matrix* pada Gambar 17 menguatkan temuan ini. Tercatat total 23 misklasifikasi, jumlah tertinggi dari semua model *baseline*. Kesalahan klasifikasi tersebar di beberapa kelas, menunjukkan bahwa model InceptionV3 pada konfigurasi dasarnya memiliki lebih banyak kesulitan dalam membedakan antar kelas dibandingkan arsitektur lain yang diuji.




Gambar 17: Confusion Matrix InceptionV3 Baseline

### 3.1.6. Rangkuman Performa Baseline

Untuk memudahkan perbandingan, Tabel 8 menyajikan rangkuman performa dari kelima model pada kondisi *baseline*. Dari data ini, terlihat jelas bahwa MobileNetV3 Large menunjukkan performa *baseline* terbaik secara keseluruhan, baik dari sisi akurasi maupun jumlah kesalahan yang minimal. Sementara itu, InceptionV3 menunjukkan performa terendah di hampir semua metrik.

Tabel 8. Perbandingan Performa Seluruh Model Baseline

| Model | Accuracy | Precision | Recall | F1-Score | Misclasified | Training Time (Seconds) | Model Size (MB) |
|---|---|---|---|---|---|---|---|
| DenseNet121 | 0.99125 | 0.99143 | 0.99125 | 0.99127 | 7 | 199.42 | 28.14 |
| ResNet50V2 | 0.98125 | 0.98126 | 0.98125 | 0.98124 | 15 | 176.57 | 90.51 |
| MobileNetV3 Small | 0.99125 | 0.99144 | 0.99125 | 0.99118 | 7 | 113.29 | 4.07 |
| MobileNetV3 Large | 0.995 | 0.99502 | 0.995 | 0.99499 | 4 | 129.94 | 12.03 |
| InceptionV3 | 0.97125 | 0.97126 | 0.97125 | 0.97124 | 23 | 299.44 | 84.17 |

## 3.2. Peningkatan Performa Setelah Optimasi Hyperparameter

Setelah performa dasar diukur, proses optimasi *hyperparameter* menggunakan *Bayesian Optimization* dijalankan selama 30 *trial* untuk setiap arsitektur. Dari proses ini, model dari *trial* dengan performa validasi terbaik langsung dipilih untuk dievaluasi pada data uji. Bagian ini menyajikan hasil dari model-model terbaik yang ditemukan selama proses *tuning* tersebut.

### 3.2.1. DenseNet121

Proses optimasi pada DenseNet121 berjalan selama 5300.66 detik. Konfigurasi *hyperparameter* terbaik ditemukan pada Trial ke-3, yang terdiri dari penggunaan *optimizer* rmsprop, *learning rate*

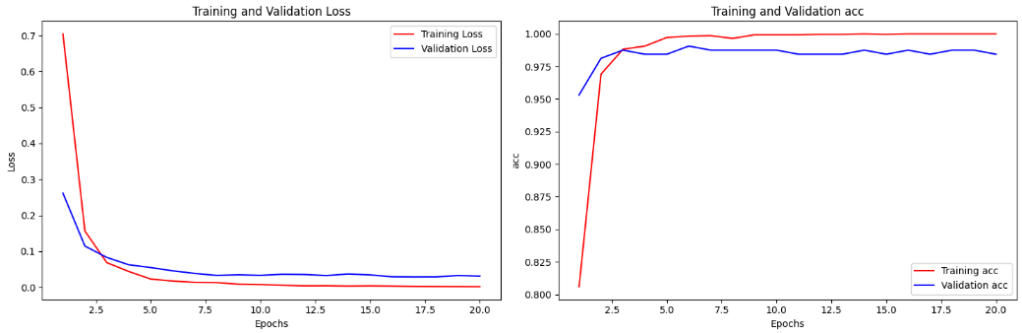
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

sebesar 1.58 x $10^{-4}$, lapisan *dense* dengan 384 unit, serta penerapan *dropout rate* sebesar 0.3. Model dari *trial* terbaik ini kemudian dievaluasi pada data uji dan menunjukkan peningkatan performa yang signifikan, seperti yang dirangkum pada Tabel 9.

Tabel 9. Ringkasan Evaluasi Model DenseNet121 (Setelah Tuning)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.9975 |
| Precision | 0.9975 |
| Recall | 0.9975 |
| F1-Score | 0.9975 |
| Training Time (Seconds) | 187.52 |
| Model Size (MB) | 29.18 |
| Durasi Total Proses Tuning (Seconds) | 5300.66 |

Berdasarkan Tabel 9, akurasi model meningkat menjadi 99.75%. Menariknya, waktu pelatihan untuk model yang telah dioptimasi ini justru sedikit lebih cepat dibandingkan versi *baseline*-nya, meskipun ukuran modelnya sedikit bertambah. Kurva pembelajaran dari *trial* terbaik ini disajikan pada Gambar 18. Grafik ini menampilkan konvergensi yang sangat baik, di mana kurva *loss* dan akurasi untuk data latih dan validasi berjalan sangat berdekatan dan stabil pada tingkat performa yang tinggi. Hal ini menegaskan bahwa model yang terpilih memiliki kemampuan generalisasi yang luar biasa dan tidak *overfitting*.




Gambar 18: Kurva Pembelajaran DenseNet121 dari Trial Terbaik

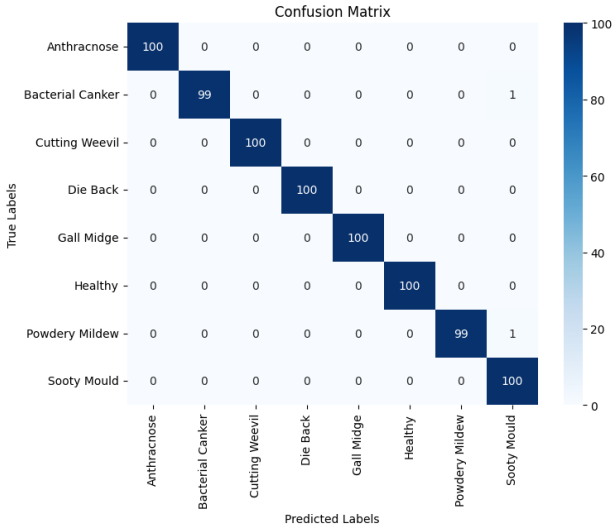
Analisis performa per kelas yang lebih detail pada Gambar 19 dan 20 menunjukkan hasil yang nyaris sempurna. Sebagian besar kelas penyakit berhasil diklasifikasikan dengan F1-score 1.00.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 1.00 | 1.00 | 100 |
| Bacterial Canker | 1.00 | 0.99 | 0.99 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 1.00 | 1.00 | 1.00 | 100 |
| Gall Midge | 1.00 | 1.00 | 1.00 | 100 |
| Healthy | 1.00 | 1.00 | 1.00 | 100 |
| Powdery Mildew | 1.00 | 0.99 | 0.99 | 100 |
| Sooty Mould | 0.98 | 1.00 | 0.99 | 100 |
| accuracy | | | 1.00 | 800 |
| macro avg | 1.00 | 1.00 | 1.00 | 800 |
| weighted avg | 1.00 | 1.00 | 1.00 | 800 |

Gambar 19: Laporan Klasifikasi DenseNet121 Setelah Tuning

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

*Confusion matrix* pada Gambar 20 secara visual mengonfirmasi peningkatan ini. Jumlah misklasifikasi berhasil dikurangi secara drastis dari 7 kasus pada model *baseline* menjadi hanya 2 kasus. Kedua kesalahan tersebut adalah satu sampel 'Bacterial Canker' dan satu sampel 'Powdery Mildew' yang salah diklasifikasikan sebagai 'Sooty Mould'.




Gambar 20: Confusion Matrix DenseNet121 Setelah Tuning

Untuk menyajikan perbandingan langsung, Tabel 11 merangkum perbedaan performa sebelum dan sesudah *tuning* untuk DenseNet121.

Tabel 11. Perbandingan Performa DenseNet121 (Baseline vs. Tuning)

| Metrik | Baseline | Tuning |
|---|---|---|
| Accuracy | 0.99125 | 0.9975 |
| Precision | 0.99143 | 0.99754 |
| Recall | 0.99125 | 0.9975 |
| F1-Score | 0.99127 | 0.9975 |
| Training Time (Seconds) | 199.42 | 187.52 |
| Model Size (MB) | 28.14 | 29.18 |

### 3.2.2. ResNet50V2

Proses optimasi pada ResNet50V2 berjalan selama 4002.92 detik (sekitar 67 menit). Konfigurasi *hyperparameter* terbaik ditemukan pada Trial ke-13, yang terdiri dari penggunaan *optimizer* adam, *learning rate* sebesar 8.98 x $10^{-4}$, lapisan *dense* dengan 224 unit, serta penerapan *dropout rate* sebesar 0.4. Model dari *trial* terbaik ini kemudian dievaluasi pada data uji, dengan hasil performa yang dirangkum pada Tabel 12.

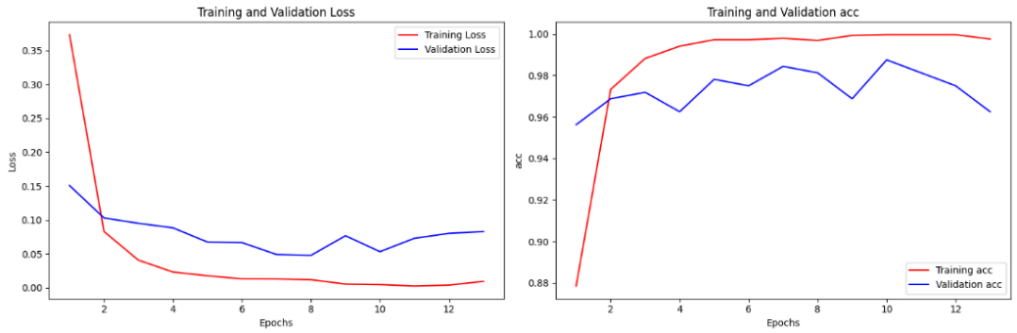
Tabel 12. Ringkasan Evaluasi Model ResNet50V2 (Setelah Tuning)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.99625 |
| Precision | 0.99629 |
| Recall | 0.99625 |
| F1-Score | 0.99624 |
| Training Time (Seconds) | 117.93 |
| Model Size (MB) | 92.02 |
| Durasi Total Proses Tuning (Seconds) | 4002.92 |

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Berdasarkan Tabel 12, akurasi model meningkat sangat signifikan menjadi 99.63%. Salah satu keuntungan terbesar dari proses optimasi pada model ini adalah penurunan waktu pelatihan yang drastis menjadi hanya 117.93 detik, atau 33.2% lebih cepat dari versi *baseline*-nya. Temuan ini menunjukkan bahwa optimasi tidak hanya meningkatkan performa prediktif, tetapi juga efisiensi komputasi model.

Kurva pembelajaran pada Gambar 26 menunjukkan proses pelatihan yang baik. Grafik *loss* (kiri) dan akurasi (kanan) menunjukkan konvergensi yang cepat. Meskipun terdapat sedikit fluktuasi pada kurva validasi di epoch-epoch akhir, mekanisme *EarlyStopping* memastikan model tidak mengalami *overfitting* dan tetap mempertahankan bobot terbaiknya.




Gambar 26: Kurva Pembelajaran ResNet50V2 dari Trial Terbaik

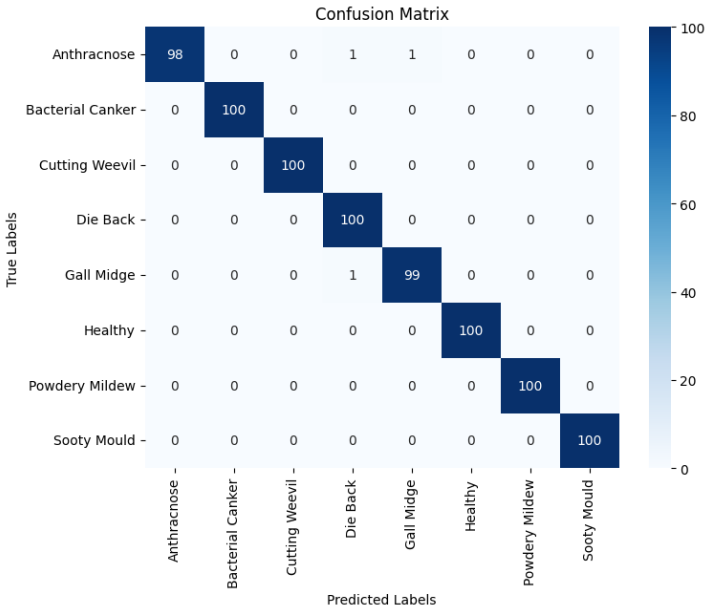
Laporan klasifikasi pada Gambar 27 menunjukkan performa yang nyaris sempurna di hampir semua kelas, dengan sebagian besar metrik mencapai 0.99 atau 1.00. Ini merupakan peningkatan besar dari hasil *baseline*.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 0.98 | 0.99 | 100 |
| Bacterial Canker | 1.00 | 1.00 | 1.00 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 0.98 | 1.00 | 0.99 | 100 |
| Gall Midge | 0.99 | 0.99 | 0.99 | 100 |
| Healthy | 1.00 | 1.00 | 1.00 | 100 |
| Powdery Mildew | 1.00 | 1.00 | 1.00 | 100 |
| Sooty Mould | 1.00 | 1.00 | 1.00 | 100 |
| accuracy | | | 1.00 | 800 |
| macro avg | 1.00 | 1.00 | 1.00 | 800 |
| weighted avg | 1.00 | 1.00 | 1.00 | 800 |

Gambar 27: Laporan Klasifikasi ResNet50V2 Setelah Tuning

Peningkatan paling dramatis terlihat pada *confusion matrix* (Gambar 28). Jumlah misklasifikasi berhasil ditekan secara masif, berkurang dari 15 kasus pada model *baseline* menjadi hanya 3 kasus. Kesalahan yang tersisa adalah satu sampel 'Anthracnose' yang salah diklasifikasikan sebagai 'Gall Midge', satu lagi sebagai 'Healthy', dan satu sampel 'Gall Midge' sebagai 'Die Back'. Ini menunjukkan kemampuan diskriminatif model yang jauh lebih superior setelah dioptimasi.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967




Gambar 28: Confusion Matrix ResNet50V2 Setelah Tuning

Tabel 13 menyajikan perbandingan langsung untuk menyoroti dampak dari proses *tuning* pada model ResNet50V2.

Tabel 13. Perbandingan Performa ResNet50V2 (Baseline vs. Tuning)

| Metrik | Baseline | Tuning |
|---|---|---|
| Accuracy | 0.98125 | 0.99625 |
| Precision | 0.98126 | 0.99629 |
| Recall | 0.98125 | 0.99625 |
| F1-Score | 0.98124 | 0.99624 |
| Training Time (Seconds) | 176.57 | 117.93 |
| Model Size (MB) | 90.51 | 92.02 |

### 3.2.3. MobileNetV3 Small

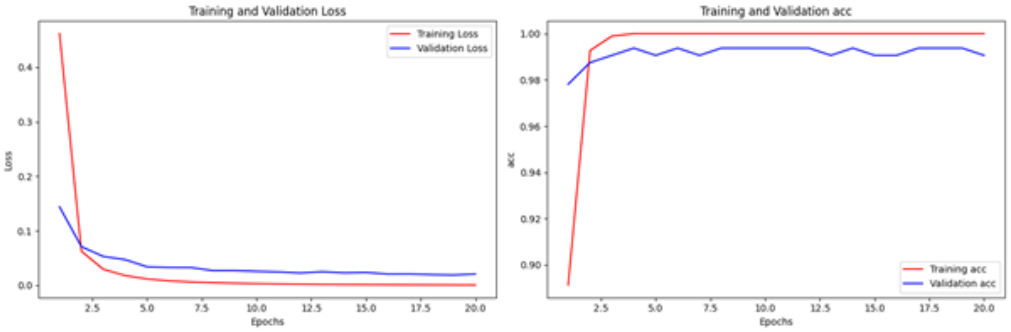
Proses optimasi pada MobileNetV3 Small berjalan selama 3986.97 detik (sekitar 66 menit). Konfigurasi *hyperparameter* terbaik ditemukan pada Trial ke-9, yang terdiri dari penggunaan *optimizer* adam, *learning rate* sebesar 3.11 x $10^{-4}$, lapisan *dense* dengan 480 unit, dan tanpa penggunaan *dropout* (*rate* 0.0). Model dari *trial* terbaik ini kemudian dievaluasi pada data uji, dengan hasil performa yang dirangkum pada Tabel 14.

Tabel 14. Ringkasan Evaluasi Model MobileNetV3 Small (Setelah Tuning)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.995 |
| Precision | 0.99507 |
| Recall | 0.995 |
| F1-Score | 0.995 |
| Training Time (Seconds) | 154.14 |
| Model Size (MB) | 4.97 |
| Durasi Total Proses Tuning (Seconds) | 3986.97 |

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Berdasarkan Tabel 14, akurasi model meningkat menjadi 99.50%. Meskipun performa *baseline* sudah sangat tinggi, proses *tuning* tetap memberikan perbaikan. Peningkatan performa ini diiringi sedikit kenaikan pada waktu pelatihan dan ukuran model jika dibandingkan dengan versi *baseline*-nya. Kurva pembelajaran dari *trial* terbaik pada Gambar 29 menampilkan proses konvergensi yang ideal. Grafik *loss* dan akurasi untuk data latih dan validasi berjalan sangat berdekatan, yang menegaskan bahwa model yang telah dioptimasi ini memiliki kemampuan generalisasi yang sangat baik dan tidak mengalami *overfitting*.




Gambar 29: Kurva Pembelajaran MobileNetV3 Small dari Trial Terbaik

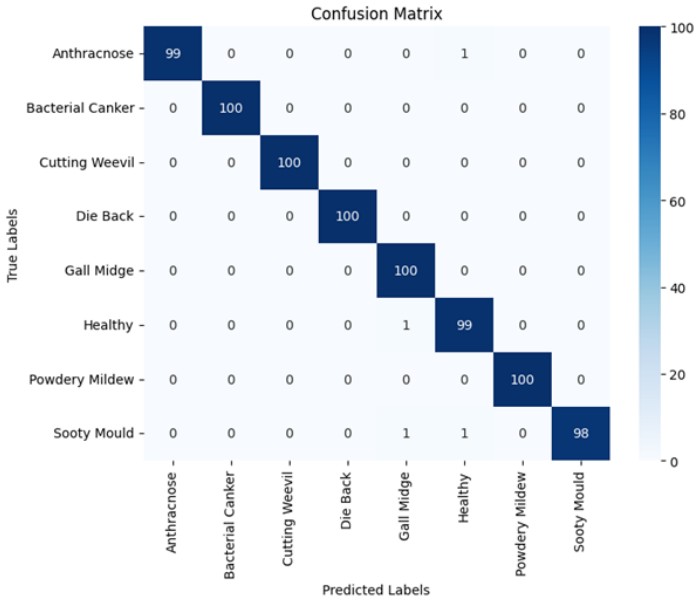
Laporan klasifikasi pada Gambar 30 menunjukkan performa yang luar biasa, di mana hampir semua kelas mencapai F1-score 0.99 atau 1.00. Ini menunjukkan perbaikan dari model *baseline*, terutama pada kelas-kelas yang sebelumnya memiliki sedikit kelemahan.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 0.99 | 0.99 | 100 |
| Bacterial Canker | 1.00 | 1.00 | 1.00 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 1.00 | 1.00 | 1.00 | 100 |
| Gall Midge | 0.98 | 1.00 | 0.99 | 100 |
| Healthy | 0.98 | 0.99 | 0.99 | 100 |
| Powdery Mildew | 1.00 | 1.00 | 1.00 | 100 |
| Sooty Mould | 1.00 | 0.98 | 0.99 | 100 |
| accuracy | | | 0.99 | 800 |
| macro avg | 1.00 | 0.99 | 1.00 | 800 |
| weighted avg | 1.00 | 0.99 | 1.00 | 800 |

Gambar 30: Laporan Klasifikasi MobileNetV3 Small Setelah Tuning

Peningkatan performa paling jelas terlihat pada *confusion matrix* (Gambar 31). Jumlah misklasifikasi berhasil dikurangi dari 7 kasus menjadi hanya 4 kasus. Kesalahan yang tersisa adalah satu sampel 'Anthracnose', satu sampel 'Healthy', dan dua sampel 'Sooty Mould' yang salah diklasifikasikan sebagai 'Gall Midge'.

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967




Gambar 31: Confusion Matrix MobileNetV3 Small Setelah Tuning

Tabel 15 menyajikan perbandingan langsung untuk menyoroti dampak dari proses *tuning* pada model MobileNetV3 Small.

Tabel 15. Perbandingan Performa MobileNetV3 Small (Baseline vs. Tuning)

| Metrik | Baseline | Tuning |
|---|---|---|
| Accuracy | 0.99125 | 0.995 |
| Precision | 0.99144 | 0.99507 |
| Recall | 0.99125 | 0.995 |
| F1-Score | 0.99118 | 0.995 |
| Training Time (Seconds) | 113.29 | 154.14 |
| Model Size (MB) | 4.07 | 4.97 |

### 3.2.4. MobileNetV3 Large

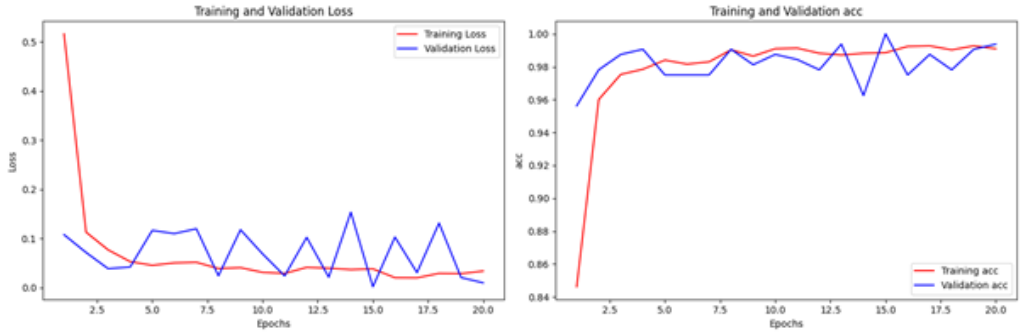
Proses optimasi pada MobileNetV3 Large berjalan selama 3335.59 detik (sekitar 56 menit). Konfigurasi *hyperparameter* terbaik yang menghasilkan performa sempurna ditemukan pada Trial ke-2, yang terdiri dari penggunaan *optimizer* rmsprop, *learning rate* yang relatif tinggi yaitu 0.00747, lapisan *dense* dengan 32 unit, dan *dropout rate* sebesar 0.2. Hasil evaluasi model dari *trial* terbaik ini pada data uji disajikan pada Tabel 16.

Tabel 16. Ringkasan Evaluasi Model MobileNetV3 Large (Setelah Tuning)

| Metrik | Nilai |
|---|---|
| Accuracy | 1.0 |
| Precision | 1.0 |
| Recall | 1.0 |
| F1-Score | 1.0 |
| Training Time (Seconds) | 135.04 |
| Model Size (MB) | 11.93 |
| Durasi Total Proses Tuning (Seconds) | 3335.59 |

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Seperti yang ditunjukkan pada Tabel 16, proses optimasi berhasil mendorong performa MobileNetV3 Large hingga mencapai nilai sempurna 1.0 untuk semua metrik evaluasi utama. Peningkatan performa ini dicapai dengan *trade-off* yang sangat minimal, di mana waktu pelatihan hanya sedikit bertambah dan ukuran model justru sedikit lebih kecil dibandingkan versi *baseline*. Kurva pembelajaran pada Gambar 32 menampilkan proses pelatihan dari *trial* terbaik. Meskipun kurva validasi (biru) menunjukkan adanya fluktuasi, mekanisme *EarlyStopping* memastikan bahwa bobot model dengan performa validasi tertinggilah yang disimpan. Efektivitas model akhir ini terbukti dengan pencapaian akurasi 100% pada data uji, yang menegaskan bahwa fluktuasi selama pelatihan tidak menghalangi pencapaian hasil yang optimal.




Gambar 32: Kurva Pembelajaran MobileNetV3 Large dari Trial Terbaik

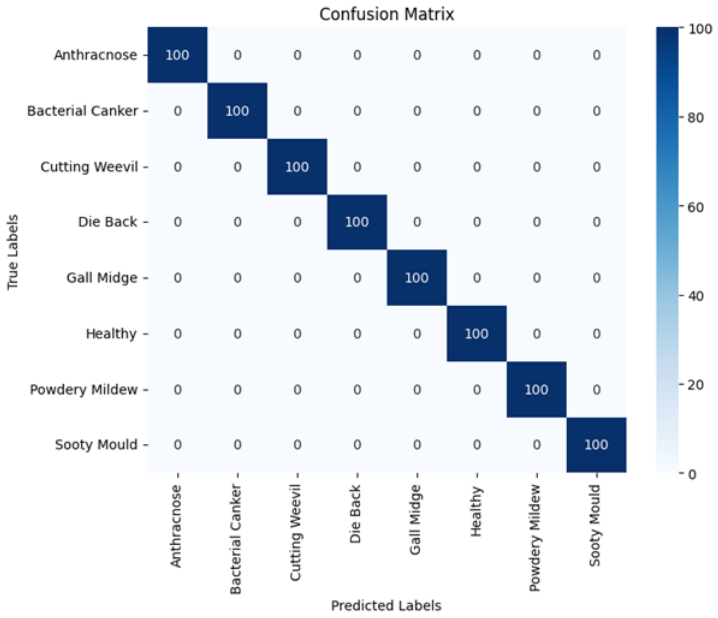
Laporan klasifikasi pada Gambar 33 menjadi bukti performa sempurna dari model ini. Semua metrik—presisi, *recall*, dan F1-score—mencapai nilai 1.00 untuk setiap kelas penyakit tanpa kecuali, menandakan tidak ada satupun kesalahan prediksi.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 1.00 | 1.00 | 1.00 | 100 |
| Bacterial Canker | 1.00 | 1.00 | 1.00 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 1.00 | 1.00 | 1.00 | 100 |
| Gall Midge | 1.00 | 1.00 | 1.00 | 100 |
| Healthy | 1.00 | 1.00 | 1.00 | 100 |
| Powdery Mildew | 1.00 | 1.00 | 1.00 | 100 |
| Sooty Mould | 1.00 | 1.00 | 1.00 | 100 |
| accuracy | | | 1.00 | 800 |
| macro avg | 1.00 | 1.00 | 1.00 | 800 |
| weighted avg | 1.00 | 1.00 | 1.00 | 800 |

Gambar 33: Laporan Klasifikasi MobileNetV3 Large Setelah Tuning

*Confusion matrix* pada Gambar 34 secara visual menyajikan hasil yang ideal ini. Semua 800 sampel data uji berada pada diagonal utama, yang berarti terdapat 0 misklasifikasi. Setiap sampel dari setiap kelas berhasil diklasifikasikan dengan benar, menunjukkan kemampuan diskriminatif yang absolut dari model hasil *tuning* pada dataset yang digunakan.

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967




Gambar 34: Confusion Matrix MobileNetV3 Large Setelah Tuning

Tabel 17 menyajikan perbandingan langsung untuk menyoroti dampak dari proses *tuning* pada model MobileNetV3 Large.

Tabel 17. Perbandingan Performa MobileNetV3 Large (Baseline vs. Tuning)

| Metrik | Baseline | Tuning |
|---|---|---|
| Accuracy | 0.995 | 1.0 |
| Precision | 0.99502 | 1.0 |
| Recall | 0.995 | 1.0 |
| F1-Score | 0.99499 | 1.0 |
| Training Time (Seconds) | 129.94 | 135.04 |
| Model Size (MB) | 12.03 | 11.93 |

### 3.2.5. InceptionV3

Proses optimasi pada InceptionV3 berjalan paling lama, yaitu selama 7146.33 detik (hampir 2 jam). Konfigurasi *hyperparameter* terbaik ditemukan pada Trial ke-11, yang terdiri dari penggunaan *optimizer* adam, *learning rate* sebesar 0.00561, lapisan *dense* dengan 288 unit, dan *dropout rate* sebesar 0.2. Model dari *trial* terbaik ini kemudian dievaluasi pada data uji, dengan hasil performa yang dirangkum pada Tabel 18.

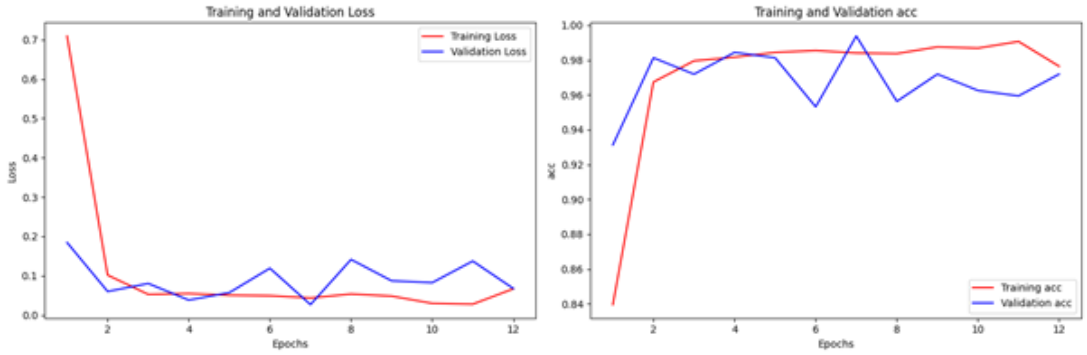
Tabel 18. Ringkasan Evaluasi Model InceptionV3 (Setelah Tuning)

| Metrik | Nilai |
|---|---|
| Accuracy | 0.985 |
| Precision | 0.98505 |
| Recall | 0.985 |
| F1-Score | 0.98498 |
| Training Time (Seconds) | 207.72 |
| Model Size (MB) | 86.03 |
| Durasi Total Proses Tuning (Seconds) | 7146.33 |

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

Berdasarkan Tabel 18, performa model InceptionV3 meningkat sangat signifikan setelah dioptimasi, dengan akurasi mencapai 98.50%. Ini merupakan salah satu peningkatan paling dramatis yang diamati dalam penelitian ini.

Kurva pembelajaran pada Gambar 35 menunjukkan proses pelatihan yang berhasil, namun dengan fluktuasi yang lebih terlihat pada kurva validasi dibandingkan model-model lain. Hal ini mengindikasikan bahwa arsitektur InceptionV3 mungkin sedikit kurang stabil selama proses pelatihan untuk dataset ini, meskipun mekanisme *EarlyStopping* berhasil mencegah *overfitting* yang parah.




Gambar 35: Kurva Pembelajaran InceptionV3 dari Trial Terbaik

Laporan klasifikasi pada Gambar 36 menunjukkan peningkatan yang signifikan di semua metrik per kelas, dengan sebagian besar F1-score berada di 0.98 atau 0.99.

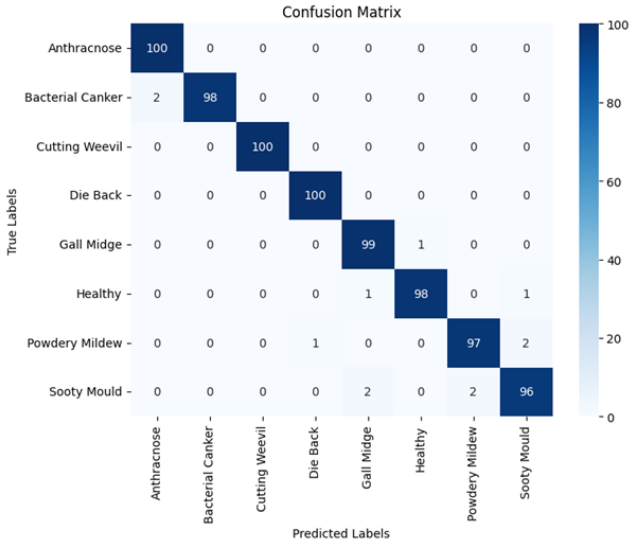
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Anthracnose | 0.98 | 1.00 | 0.99 | 100 |
| Bacterial Canker | 1.00 | 0.98 | 0.99 | 100 |
| Cutting Weevil | 1.00 | 1.00 | 1.00 | 100 |
| Die Back | 0.99 | 1.00 | 1.00 | 100 |
| Gall Midge | 0.97 | 0.99 | 0.98 | 100 |
| Healthy | 0.99 | 0.98 | 0.98 | 100 |
| Powdery Mildew | 0.98 | 0.97 | 0.97 | 100 |
| Sooty Mould | 0.97 | 0.96 | 0.96 | 100 |
| accuracy | | | 0.98 | 800 |
| macro avg | 0.99 | 0.98 | 0.98 | 800 |
| weighted avg | 0.99 | 0.98 | 0.98 | 800 |

Gambar 36: Laporan Klasifikasi InceptionV3 Setelah Tuning

Peningkatan paling signifikan terlihat pada *confusion matrix* (Gambar 37). Jumlah misklasifikasi berhasil berkurang hampir setengahnya, dari 23 kasus pada model *baseline* menjadi 12 kasus. Meskipun masih menjadi model dengan jumlah kesalahan tertinggi, perbaikan ini menunjukkan betapa besar dampak optimasi pada arsitektur ini.

Tabel 19 menyajikan perbandingan langsung untuk menyoroti dampak dari proses *tuning* pada model InceptionV3, yang merupakan salah satu yang paling signifikan.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967




Gambar 37: Confusion Matrix InceptionV3 Setelah Tuning

Tabel 19. Perbandingan Performa InceptionV3 (Baseline vs. Tuning)

| Metrik | Baseline | Tuning |
|---|---|---|
| Accuracy | 0.97125 | 0.985 |
| Precision | 0.97126 | 0.98505 |
| Recall | 0.97125 | 0.985 |
| F1-Score | 0.97124 | 0.98498 |
| Training Time (Seconds) | 299.44 | 207.72 |
| Model Size (MB) | 84.17 | 86.03 |

### 3.3. Rangkuman Komparatif Hasil Eksperimen

Untuk memberikan gambaran menyeluruh, Tabel 20 merangkum metrik performa utama dari kelima arsitektur, baik pada kondisi *baseline* maupun setelah melalui proses optimasi menggunakan *Bayesian Optimization*. Tabel ini menjadi dasar untuk analisis komparatif akhir.

Tabel 20. Rangkuman Komparatif Hasil Eksperimen Seluruh Model

| Model | Accuracy | Precision | Recall | F1-Score | Misclasified | Training Time (Seconds) | Model Size (MB) |
|---|---|---|---|---|---|---|---|
| DenseNet121 (B) | 0.99125 | 0.99143 | 0.99125 | 0.99127 | 7 | 199.42 | 28.14 |
| DenseNet121 (T) | 0.9975 | 0.99754 | 0.9975 | 0.99750 | 2 | 187.52 | 29.18 |
| ResNet50V2 (B) | 0.98125 | 0.98126 | 0.98125 | 0.98124 | 15 | 176.57 | 90.51 |
| ResNet50V2 (T) | 0.99625 | 0.99629 | 0.99625 | 0.99624 | 3 | 117.93 | 92.02 |
| MobileNetV3 Small (B) | 0.99125 | 0.99144 | 0.99125 | 0.99118 | 7 | 113.29 | 4.07 |
| MobileNetV3 Small (T) | 0.995 | 0.99507 | 0.995 | 0.99500 | 4 | 154.14 | 4.97 |
| MobileNetV3 Large (B) | 0.995 | 0.99502 | 0.995 | 0.99499 | 4 | 129.94 | 12.03 |
| MobileNetV3 Large (T) | 1.0 | 1.0 | 1.0 | 1.0 | 0 | 135.04 | 11.93 |
| InceptionV3 (B) | 0.97125 | 0.97126 | 0.97125 | 0.97124 | 23 | 299.44 | 84.17 |
| InceptionV3 (T) | 0.985 | 0.98505 | 0.985 | 0.98498 | 12 | 207.72 | 86.03 |

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

## 4. DISCUSSIONS

Hasil penelitian ini secara konsisten menunjukkan bahwa penerapan *Bayesian Optimization* untuk *tuning hyperparameter* memberikan dampak yang sangat positif terhadap performa kelima arsitektur *pretrained* CNN yang diuji. Dari analisis komparatif, dapat ditarik beberapa diskusi dan implikasi penting. Perspektif utama dari penelitian ini adalah bahwa strategi optimasi yang tepat seringkali lebih berpengaruh daripada pemilihan arsitektur yang kompleks semata. Hal ini dibuktikan dengan munculnya MobileNetV3 Large sebagai model paling unggul yang berhasil mencapai akurasi sempurna 100%, melampaui arsitektur yang lebih berat seperti DenseNet121 dan ResNet50V2.

Salah satu temuan menarik adalah bagaimana proses optimasi memberikan manfaat besar pada model-model yang memiliki performa *baseline* terendah, yaitu ResNet50V2 dan InceptionV3. Jumlah kesalahan klasifikasi pada ResNet50V2 berkurang drastis dari 15 menjadi 3, sementara pada InceptionV3 berkurang dari 23 menjadi 12. Ini mengindikasikan bahwa arsitektur yang lebih kompleks memiliki ruang pencarian *hyperparameter* yang sangat luas dan sensitif, sehingga sulit untuk mencapai potensi maksimalnya dengan konfigurasi standar. *Bayesian Optimization* terbukti mampu menavigasi ruang kompleks ini dan menemukan konfigurasi yang meningkatkan akurasi.

Temuan dari penelitian ini menunjukkan keselarasan dengan studi-studi sebelumnya, sekaligus memberikan perspektif baru terkait pentingnya optimasi. Tingginya akurasi yang dicapai (di atas 98.5% setelah optimasi) menegaskan pernyataan [15] bahwa pendekatan *deep learning* efektif untuk klasifikasi penyakit tanaman. Namun, penelitian ini memberikan kebaruan yang krusial. Berbeda dengan [18] yang melaporkan ResNet50 sebagai model terbaik dengan akurasi 99.37% tanpa optimasi sistematis, penelitian ini menemukan bahwa MobileNetV3 Large yang dioptimalkan tidak hanya mampu melampaui, tetapi mencapai akurasi sempurna 100%. Demikian pula, jika studi oleh [18], [20] melaporkan akurasi untuk DenseNet121 sebesar 99.5% dan 99.1%, pendekatan optimasi kami berhasil meningkatkannya menjadi 99.75%. Selain itu, jika [19] hanya mencapai akurasi InceptionV3 sebesar 96.5%, penelitian ini berhasil mendorongnya hingga 98.5%. Perbandingan kuantitatif ini menunjukkan superioritas tuning adaptif seperti Bayesian Optimization.

Analisis *trade-off* antara akurasi dan efisiensi juga memberikan wawasan penting untuk implementasi praktis. MobileNetV3 Large menunjukkan keseimbangan terbaik, dengan akurasi tertinggi (100%) namun tetap memiliki ukuran model yang sangat efisien (11.93 MB). Wawasan ini memiliki kontribusi langsung pada bidang informatika, khususnya dalam pengembangan AI yang efisien untuk *edge computing* di sektor pertanian. Dengan menawarkan performa tinggi namun dengan kebutuhan komputasi yang minimal, sehingga menjawab tantangan implementasi di lapangan. Di sisi lain, jika prioritas utama adalah ukuran model yang minimal untuk aplikasi pada perangkat *edge* atau seluler, maka MobileNetV3 Small (4.97 MB) menjadi pilihan yang sangat relevan dengan akurasi yang tetap tinggi (99.50%). Hal ini sangat relevan untuk konteks pertanian di Indonesia, di mana aksesibilitas teknologi melalui aplikasi *mobile* dapat menjadi kunci percepatan adopsi inovasi digital di lapangan.

Meskipun penelitian ini berhasil mencapai hasil yang sangat baik, terdapat beberapa keterbatasan yang perlu diakui. Penelitian ini menggunakan satu dataset publik (MangoLeafBD) yang citranya diambil dalam kondisi yang relatif terkontrol. Performa model mungkin dapat bervariasi jika diuji pada citra dari kondisi lapangan yang lebih beragam (misalnya, pencahayaan berbeda, latar belakang kompleks). Oleh karena itu, penelitian di masa depan dapat berfokus pada pengujian generalisasi model-model yang telah teroptimasi ini pada dataset yang lebih menantang. Selain itu, eksplorasi arsitektur yang lebih baru seperti *Vision Transformers* (ViT) dengan metode optimasi yang sama dapat menjadi arah riset yang menjanjikan.

## 5. CONCLUSION

MobileNetV3 Large mencapai 100% akurasi setelah optimasi menggunakan Bayesian Optimization, melampaui model lain dalam hal efisiensi sekaligus mempertahankan performa tertinggi. Penelitian ini meningkatkan pemahaman tentang optimasi Bayesian untuk CNN lightweight, berkontribusi pada pengembangan AI yang akurat, efisien, dan berkelanjutan di bidang pertanian informatika. Temuan ini membuktikan bahwa arsitektur ringan yang dioptimalkan dengan tepat, sehingga menawarkan solusi praktis bagi deteksi penyakit tanaman pada perangkat dengan sumber daya terbatas.

## CONFLICT OF INTEREST

Penulis menyatakan bahwa tidak ada konflik kepentingan antara para penulis maupun dengan objek penelitian dalam artikel ini.

## REFERENCES

[1] Badan Pusat Statistik Indonesia, "Produksi Buah–Buahan dan Sayuran Tahunan Menurut Jenis Tanaman." Accessed: Feb. 21, 2025. [Online]. Available: https://www.bps.go.id/id/statistics-table/3/WXpSVU5uUTBOSEl5WVhGQmVESTVSVnBSVlhWeVVUMDkjMw==/produksi-buah-buahan-dan-sayuran-tahunan-menurut-jenis-tanaman.html?year=2023

[2] A. M. Kiloes, D. Joyce, and A. Abdul Aziz, "Exploring the Challenges and Opportunities of Mango Export from Indonesia: Insights from Stakeholder Interviews," *The Qualitative Report*, vol. 29, no. 3, pp. 811–830, Mar. 2024, doi: 10.46743/2160-3715/2024.6343.

[3] A. M. Kiloes, Y. N. Muflikh, D. Joyce, and A. Abdul Aziz, "Understanding the complexity of the Indonesian fresh mango industry in delivering quality to markets: A systems thinking approach," *Food Policy*, vol. 118, p. 102497, Jul. 2023, doi: 10.1016/j.foodpol.2023.102497.

[4] G. V. Benatar, A. Wibowo, and Suryanti, "First report of Colletotrichum asianum associated with mango fruit anthracnose in Indonesia," *Crop Protection*, vol. 141, p. 105432, Mar. 2021, doi: 10.1016/j.cropro.2020.105432.

[5] S. T. Raza *et al.*, "A Review on White Mango Scale Biology, Ecology, Distribution and Management," *Agriculture*, vol. 13, no. 9, p. 1770, Sep. 2023, doi: 10.3390/agriculture13091770.

[6] S. Solikin, "Deteksi Penyakit Pada Tanaman Mangga Dengan Citra Digital : Tinjauan Literatur Sistematis (SLR)," *Bina Insani Ict Journal*, vol. 7, no. 1, p. 63, 2020, doi: 10.51211/biict.v7i1.1336.

[7] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," *IEEE Access*, vol. 9, no. Ccv, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[8] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, vol. 14, no. March, pp. 1–22, Mar. 2023, doi: 10.3389/fpls.2023.1158933.

[9] S. Hemalatha and J. J. B. Jayachandran, "A Multitask Learning-Based Vision Transformer for Plant Disease Localization and Classification," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 188, Jul. 2024, doi: 10.1007/s44196-024-00597-3.

[10] S. M. Hassan, M. Jasinski, Z. Leonowicz, E. Jasinska, and A. K. Maji, "Plant Disease Identification Using Shallow Convolutional Neural Network," *Agronomy*, vol. 11, no. 12, p. 2388, Nov. 2021, doi: 10.3390/agronomy11122388.

[11] S. Sulaiman, N. Nurhayati, and J. N. Sitompul, "SISTEM PAKAR MENDIAGNOSA PENYAKIT TANAMAN MANGGA ARUMANIS DENGAN METODE CERTAINTY FACTOR," *JURNAL TEKNISI*, vol. 1, no. 2, p. 61, Aug. 2021, doi: 10.54314/teknisi.v1i2.660.

[12] G. V. Benatar, Y. Nurhayati, and N. Febryani, "Identifikasi Colletotrichum asianum Penyebab Antraknosa Mangga Kultivar Golek di Indramayu," *Media Pertanian*, vol. 8, no. 1, pp. 1–13, May 2023, doi: 10.37058/mp.v8i1.6900.

[13] A. I. Tanzil, I. Sucipto, and W. Muhlison, "Inventory of Pest and Disease in Mango Plants (Mangifera indica)," *Jurnal Online Pertanian Tropik*, vol. 9, no. 2, pp. 098–105, 2022, doi:

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

10.32734/jopt.v9i2.8972.
[14] I. W. Susila *et al.*, "Abundance, distribution mapping and DNA barcoding of Procontarinia robusta (Diptera: Cecidomyiidae), a mango gall midge in Bali, Indonesia," *Biodiversitas Journal of Biological Diversity*, vol. 23, no. 12, pp. 6428–6436, Jan. 2023, doi: 10.13057/biodiv/d231241.
[15] D. Faye, I. Diop, and D. Dione, "Mango Diseases Classification Solutions Using Machine Learning or Deep Learning: A Review," *Journal of Computer and Communications*, vol. 10, no. 12, pp. 16–28, 2022, doi: 10.4236/jcc.2022.1012002.
[16] M. Jelali, "Deep learning networks-based tomato disease and pest detection: a first review of research studies using real field datasets," *Frontiers in Plant Science*, vol. 15, no. October, pp. 1–20, Oct. 2024, doi: 10.3389/fpls.2024.1493322.
[17] H. Nugroho, J. X. Chew, S. Eswaran, and F. S. Tay, "Resource-optimized cnns for real-time rice disease detection with ARM cortex-M microprocessors," *Plant Methods*, vol. 20, no. 1, p. 159, Oct. 2024, doi: 10.1186/s13007-024-01280-6.
[18] B. U. Mahmud, A. Al Mamun, M. J. Hossen, G. Y. Hong, and B. Jahan, "Light-Weight Deep Learning Model for Accelerating the Classification of Mango-Leaf Disease," *Emerging Science Journal*, vol. 8, no. 1, pp. 28–42, Feb. 2024, doi: 10.28991/ESJ-2024-08-01-03.
[19] P. D. Rinanda, D. N. Aini, T. A. Pertiwi, S. Suryani, and A. J. Prakash, "Implementation of Convolutional Neural Network (CNN) for Image Classification of Leaf Disease In Mango Plants Using Deep Learning Approach," *Public Research Journal of Engineering, Data Technology and Computer Science*, vol. 1, no. 2, pp. 56–61, Feb. 2024, doi: 10.57152/predatecs.v1i2.872.
[20] M. A. Likorawung and D. M. Wonohadidjojo, "Implementation of DenseNet Architecture With Transfer Learning to Classify Mango Leaf Diseases," vol. 06, no. 02, pp. 78–86, 2024, doi: 10.52985/insyst.v6i2.401.
[21] V. K. Pratap and N. S. Kumar, "Deep Learning based Mango Leaf Disease Detection for Classifying and Evaluating Mango Leaf Diseases," *Fusion: Practice and Applications*, vol. 15, no. 2, pp. 261–277, 2024, doi: 10.54216/FPA.150222.
[22] C. P. Vijay and K. Pushpalatha, "DV-PSO-Net: A novel deep mutual learning model with Heuristic search using Particle Swarm optimization for Mango leaf disease detection," *Journal of Integrated Science and Technology*, vol. 12, no. 5, p. 804, May 2024, doi: 10.62110/sciencein.jist.2024.v12.804.
[23] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, p. e26586, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.
[24] B. Khan *et al.*, "Bayesian optimized multimodal deep hybrid learning approach for tomato leaf disease classification," *Scientific Reports 2024 14:1*, vol. 14, no. 1, pp. 1–30, 2024, doi: 10.1038/s41598-024-72237-x.
[25] Q. A. Hidayaturrohman and E. Hanada, "A Comparative Analysis of Hyper-Parameter Optimization Methods for Predicting Heart Failure Outcomes," *Applied Sciences*, vol. 15, no. 6, p. 3393, Mar. 2025, doi: 10.3390/app15063393.
[26] S. I. Ahmed *et al.*, "MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves," *Data in Brief*, vol. 47, p. 108941, Apr. 2023, doi: 10.1016/j.dib.2023.108941.
[27] I. K. Seneng, P. D. W. Ayu, and R. R. Huizen, "Comparative Analysis of Augmentation and Filtering Methods in VGG19 and DenseNet121 for Breast Cancer Classification," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 3, pp. 1131–1146, 2025, doi: 10.52436/1.jutif.2025.6.3.4397.
[28] T. S. Arulananth, S. W. Prakash, R. K. Ayyasamy, V. P. Kavitha, P. G. Kuppusamy, and P. Chinnasamy, "Classification of Paediatric Pneumonia Using Modified DenseNet-121 Deep-Learning Model," *IEEE Access*, vol. 12, no. March, pp. 35716–35727, 2024, doi: 10.1109/ACCESS.2024.3371151.
[29] H. I. Fitriasari and M. Rizkinia, "Improvement of Xception-ResNet50V2 Concatenation for COVID-19 Detection on Chest X-Ray Images," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, IEEE, Apr. 2021, pp. 343–347. doi: 10.1109/EIConCIT50028.2021.9431916.
[30] D. Hindarto, "Use ResNet50V2 Deep Learning Model to Classify Five Animal Species," *Jurnal*

Vol. 6, No. 5, October 2025, Page. 3051-3078
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4967

*JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, no. 4, pp. 758–768, Dec. 2023, doi: 10.35870/jtik.v7i4.1845.

[31] J. Zhu, C. Zhang, and C. Zhang, "Papaver somniferum and Papaver rhoeas Classification Based on Visible Capsule Images Using a Modified MobileNetV3-Small Network with Transfer Learning," *Entropy*, vol. 25, no. 3, p. 447, Mar. 2023, doi: 10.3390/e25030447.

[32] A. R. Muslikh, D. R. I. M. Setiadi, and A. A. Ojugo, "RICE DISEASE RECOGNITION USING TRANSFER LEARNING XCEPTION CONVOLUTIONAL NEURAL NETWORK," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 6, pp. 1535–1540, Dec. 2023, doi: 10.52436/1.jutif.2023.4.6.1529.

[33] L. Zhao and L. Wang, "A new lightweight network based on MobileNetV3," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 1, pp. 1–15, Jan. 2022, doi: 10.3837/tiis.2022.01.001.

[34] Kaka Kamaludin, Woro Isti Rahayu, and M. Y. Helmi Setywan, "TRANSFER LEARNING TO PREDICT GENRE BASED ON ANIME POSTERS," *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 5, pp. 1041–1052, Oct. 2023, doi: 10.52436/1.jutif.2023.4.5.860.

[35] T. Hidayat, I. A. Astuti, A. Yaqin, A. P. Tjilen, and T. Arifianto, "Grouping of Image Patterns Using Inceptionv3 For Face Shape Classification," *JOIV : International Journal on Informatics Visualization*, vol. 7, no. 4, pp. 2336–2343, Dec. 2023, doi: 10.30630/joiv.7.4.1743.

[36] Y. Motoyama, R. Tamura, K. Yoshimi, K. Terayama, T. Ueno, and K. Tsuda, "Bayesian optimization package: PHYSBO," *Computer Physics Communications*, vol. 278, p. 108405, Sep. 2022, doi: 10.1016/j.cpc.2022.108405.

[37] A. Candelieri, "A Gentle Introduction to Bayesian Optimization," in *2021 Winter Simulation Conference (WSC)*, IEEE, Dec. 2021, pp. 1–16. doi: 10.1109/WSC52266.2021.9715413.

[38] M. A. B. Bhuiyan, H. M. Abdullah, S. E. Arman, S. Saminur Rahman, and K. Al Mahmud, "BananaSqueezeNet: A very fast, lightweight convolutional neural network for the diagnosis of three prominent banana leaf diseases," *Smart Agricultural Technology*, vol. 4, no. December 2022, p. 100214, Aug. 2023, doi: 10.1016/j.atech.2023.100214.