

IMPLEMENTASI ARSITEKTUR MOBILENETV2 UNTUK KLASIFIKASI PENYAKIT PADA DAUN CABAI BERBASIS CONVOLUTIONAL NEURAL NETWORK

Nabila¹, Rian Rahmanda², Della Oktaviany³

^{1,2,3}Program Studi Teknologi Informatika Multimedia Digital, Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya

Email: ¹ nabilanabila23103@gmail.com, ² rianrahmanda@polsri.ac.id, ³ della.oktaviany@polsri.ac.id

^{*}Penulis Korespondensi

Abstrak

Serangan hama dan penyakit merupakan penyebab utama risiko produksi pada tanaman cabai, yang dapat mengakibatkan penurunan kualitas dan kuantitas panen secara signifikan, bahkan hingga mengancam terjadinya gagal panen. Namun, identifikasi penyakit ini sebagian besar masih bergantung pada pengamatan visual yang subjektif dan lambat, sehingga menunda penanganan yang efektif. Penelitian ini mengimplementasikan *Convolutional Neural Network* dengan arsitektur MobileNetV2 menggunakan pendekatan *transfer learning* untuk klasifikasi enam jenis kondisi daun cabai. Model dilatih menggunakan dataset augmentasi sebanyak 11.998 citra dan dievaluasi dengan 1.856 citra uji. Proses pelatihan meliputi tahap *Feature Extraction* dan *Fine-Tuning* dengan *hyperparameter* yang dioptimalkan untuk stabilitas model. Model yang dikembangkan berhasil mencapai performa yang sangat tinggi pada data uji, dengan *Test Accuracy* sebesar 98.17% dan hasil *precision*, *recall*, dan *f1-score* yang mencapai rata-rata 98%. Hasil ini menunjukkan bahwa implementasi MobileNetV2 dengan metode *transfer learning* adalah pendekatan yang sangat efektif dan berpotensi besar untuk dikembangkan sebagai alat bantu diagnosis penyakit yang akurat, guna memfasilitasi tindakan penanganan secara dini.

Kata kunci: CNN, Transfer Learning, MobileNetV2, Deteksi Penyakit, Tanaman Cabai.

IMPLEMENTATION OF MOBILENETV2 ARCHITECTURE FOR DISEASE CLASSIFICATION ON CHILI LEAVES BASED ON CONVOLUTIONAL NEURAL NETWORK

Abstract

Pest and disease attacks are the primary cause of production risk in chili plants, which can result in a significant reduction in crop quality and quantity, even to the point of threatening crop failure. However, the identification of these diseases is still dependent on subjective, slow, and visual observations. Consequently, treatment is often delayed. This research implements a Convolutional Neural Network with MobileNetV2 architecture using the transfer learning approach for the classification of six types of chili leaf conditions. This model was trained using an augmentation dataset that contained 11,998 images, and it was evaluated using 1,856 test images. The training process has Feature Extraction and Fine-Tuning stages with hyperparameters optimized for model stability. The model achieved a high level of performance on the test data, with a Test Accuracy of 98.17% and precision, recall, and f1-score results that reached an average of 98%. The results indicate that the implementation of MobileNetV2 with the transfer learning method is a very effective approach and has great potential to be developed as an accurate disease diagnosis tool, to facilitate early treatment measures.

Keywords: CNN, Transfer Learning, MobileNetV2, Disease Detection, Chili Plant

1. INTRODUCTION

Tanaman cabai atau *Capsicum sp.* merupakan komoditas hortikultura yang berperan penting dalam sektor pertanian di Indonesia [1]. Komoditas ini memiliki nilai yang tinggi bagi perekonomian negara

karena jumlah permintaannya yang terus meningkat di setiap tahun. Menurut data Kementerian Pertanian, total produksi cabai di Indonesia yang mencapai 3,11 juta ton pada tahun 2023, yang menempatkan negara ini sebagai pemain potensial yang signifikan di pasar perdagangan global [2]. Namun, produktivitas

tanaman cabai seringkali terancam oleh serangan berbagai jenis penyakit, yang dapat mengakibatkan penurunan kualitas dan kuantitas hasil panen secara drastis, bahkan menyebabkan gagal panen. Hal ini berdampak langsung pada kerugian ekonomi yang signifikan bagi para petani [3].

Saat ini, identifikasi penyakit pada daun cabai umumnya masih dilakukan secara manual melalui pengamatan langsung oleh petani. Metode ini sejalan dengan salah satu prinsip Pengendalian Hama Terpadu (PHT), yaitu dengan melakukan pengamatan rutin untuk menentukan kapan tindakan kuratif diperlukan [4]. Namun, tanpa dukungan alat yang objektif, pendekatan konvensional ini memiliki kelemahan signifikan, yaitu tingkat subjektivitas yang tinggi dan ketergantungan pada keahlian individu, yang dapat menunda tindakan penanganan yang tepat dan efektif [5]. Oleh karena itu, deteksi penyakit yang cepat dan akurat dapat menjadi salah satu solusi untuk melakukan pencegahan dini dan meminimalkan terjadinya kerugian.

Perkembangan teknologi dalam bidang *computer vision* khususnya *deep learning*, menawarkan solusi yang inovatif untuk otomatisasi proses identifikasi. *Convolutional Neural Network* (CNN) merupakan salah satu algoritma dalam bidang pengenalan pola dan analisis citra. Algoritma ini menggunakan lapisan konvolusi untuk mengekstraksi fitur-fitur lokal yang relevan secara otomatis tanpa supervisi manusia. Selanjutnya, lapisan pooling digunakan untuk mereduksi dimensi data, yang bertujuan untuk mempercepat proses pelatihan dan mengatasi *overfitting* sambil mempertahankan informasi penting. Proses ekstraksi dan reduksi fitur secara bertahap inilah yang membuat CNN sangat efisien dan unggul dalam berbagai aplikasi praktis, termasuk deteksi objek [6].

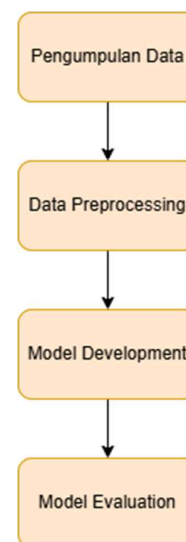
Penerapan *Convolutional Neural Network* dalam membangun model deteksi penyakit tanaman telah menunjukkan hasil yang sangat efektif pada berbagai penelitian terdahulu. Pada penelitian yang berjudul "*A CNN-based image detector for plant leaf diseases classification*" telah berhasil mengembangkan model CNN ringan yang dapat diimplementasikan pada perangkat keras dengan sumber daya terbatas untuk deteksi penyakit pada daun. Penelitian ini mampu mencapai hasil akurasi hingga 95.24% pada dataset *PlantVillage*, menunjukkan kelayakan penerapan teknologi ini di lapangan [7]. Sejalan dengan itu, pendekatan menggunakan arsitektur yang secara inheren efisien seperti MobileNetV2 juga telah menunjukkan keberhasilan. Penelitian deteksi penyakit pada daun tanaman telah diterapkan pada penelitian terdahulu dengan membangun model deteksi penyakit pada daun tebu dengan menggunakan arsitektur MobileNetV2 [8]. Adapun penelitian ini, menerapkan metode melatih model dengan metode *fine-tuning* dan menghasilkan akurasi sebesar 95,01% dan nilai loss 0,0899. Kedua penelitian ini memperkuat gagasan

bahwa algoritma CNN dan arsitektur MobileNetV2 sangat potensial untuk dikembangkan sebagai alat bantu diagnosis penyakit tanaman yang praktis dan akurat.

Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi performa arsitektur MobileNetV2 untuk tugas klasifikasi kondisi penyakit daun cabai dengan akurat dan efektif. Harapannya, sistem deteksi penyakit pada daun cabai ini mampu menjadi salah satu solusi teknologi inovatif yang memberikan dampak positif dan nyata bagi sektor pertanian hortikultura di Indonesia. Secara metodologis, penelitian ini akan menggunakan dataset citra daun cabai dengan enam kategori yang telah diaugmentasi. Arsitektur MobileNetV2 akan diimplementasikan dengan metode *transfer learning* untuk membangun model klasifikasi. Performa akhir dari model yang telah dilatih akan diukur dan dianalisis menggunakan metrik seperti akurasi, presisi, recall, dan F1-score dengan menggunakan confusion matrix.

2. RESEARCH METHOD

Dalam penelitian ini, akan mengacu pada perancangan model classifier dengan menggunakan arsitektur MobileNetV2 untuk mendeteksi penyakit pada daun cabai dan mengukur performa model classifier yang dibangun. Pada Gambar 1 dibawah merupakan tahapan penelitian yang akan dilakukan. Alur metode penelitian dimulai dari tahap melakukan pengumpulan data, *data preprocessing*, *model development*, dan *model evaluation*.



Gambar 1. Tahapan Penelitian

2.1 Tahap Pengumpulan Data

Tahap Pengumpulan data bertujuan untuk memenuhi kebutuhan pengembangan model *classifier*. Data yang dikumpulkan merupakan dataset

sekunder yang bersumber dari website Mendeley Data. Dataset dengan judul “*Chili Plant Leaf Disease and Growth Stage Dataset from Bangladesh*” menyediakan dataset gambar penyakit yang dapat diakses dan digunakan oleh kalangan peneliti, khususnya fokus di bidang *machine learning* [9]. Dataset ini memiliki dua jenis data, yaitu *Augmented Dataset* yang dialokasikan untuk proses training dan validasi model, sedangkan *Original Dataset* dialokasikan untuk tahap evaluation model.

Pada Tabel 1 di bawah merupakan jumlah data pada Original Dataset yang terdiri dari 6 kelas gambar daun cabai. Dataset ini terdiri dari kelas *Bacterial Spot*, *Curl Virus*, *Cercospora Leaf Spot*, *Nutrition Deficiency*, *White Spot*, dan *Healthy Leaves* dengan jumlah data per kelas yang berbeda.

Tabel 1. Jumlah data gambar pada Original Dataset

Index	Kelas	Data
1	<i>Bacterial Spot</i>	156
2	<i>Curl Virus</i>	423
3	<i>Cercospora Leaf Spot</i>	180
4	<i>Nutrition Deficiency</i>	444
5	<i>White Spot</i>	195
6	<i>Healthy Leaves</i>	458
Total		1856

Pada Tabel 2 di bawah merupakan jumlah data pada Augmented Dataset yang terdiri dari 6 kelas gambar daun cabai. Dataset ini terdiri dari kelas *Bacterial Spot*, *Curl Virus*, *Cercospora Leaf Spot*, *Nutrition Deficiency*, *White Spot*, dan *Healthy Leaves* dengan jumlah data sebanyak 2000 gambar per kelas.

Tabel 2. Jumlah data gambar pada Augmented Dataset

Index	Kelas	Data
1	<i>Bacterial Spot</i>	2000
2	<i>Curl Virus</i>	2000
3	<i>Cercospora Leaf Spot</i>	2000
4	<i>Nutrition Deficiency</i>	2000
5	<i>White Spot</i>	2000
6	<i>Healthy Leaves</i>	2000
Total		12000

2.2 Tahap Data Preprocessing

Pada tahap ini, dataset yang telah dikumpulkan akan di proses sebelum dilakukannya tahap *model development*. Tahap ini bertujuan untuk menghasilkan data yang lebih berkualitas dan distribusi data yang seimbang, sehingga model dapat memahami pola secara lebih efektif selama proses pelatihan data. Tahapan *data preprocessing* yang akan dilakukan meliputi:

- Melakukan pembagian dataset dengan proporsi yang sesuai. Dataset akan dibagi untuk proses *training* dan *validasi* saat pengembangan model dengan proporsi 80:20. Dataset akan dibagi dengan menggunakan pendekatan berbasis *prefix* nama file, di mana semua citra yang memiliki awalan nama *file (prefix)* yang sama diasumsikan berasal dari satu objek

daun yang sama. Hal ini bertujuan untuk menghindari terjadinya *data leakage*, yaitu kondisi ketika informasi dari luar dataset pelatihan (*validasi* atau *evaluasi set*) digunakan untuk membuat model. Kondisi ini menyebabkan model *classifier* menunjukkan performa yang sangat baik ketika pelatihan dan validasi, namun gagal bekerja secara efektif ketika dihadapkan pada data nyata yang benar-benar baru [10]. Berikut pembagian dataset dapat dilihat pada Tabel 3.

- Melakukan normalisasi pada citra gambar, yang sebelumnya berada dalam rentang [0, 255], diubah skalanya ke rentang [0, 1] dengan menggunakan $rescale=1./255$ pada set data *training*, *validasi*, dan *evaluasi* [11]. Langkah ini bertujuan untuk menstabilkan dan mempercepat proses *training* pada model. Selanjutnya, untuk meningkatkan kemampuan generalisasi model dan mengurangi resiko *overfitting*, pada set data *training* akan diterapkan teknik augmentasi *on-the-fly*. Teknik augmentasi yang digunakan meliputi rotasi, pergeseran horizontal dan vertikal, shear, zoom, horizontal flip, dan penyesuaian kecerahan.

Tabel 3. Pembagian dataset pada Training dan Validation

Jenis	Kelas	Jumlah
Training	<i>Bacterial Spot</i>	1555
	<i>Curl Virus</i>	1626
	<i>Cercospora Leaf Spot</i>	1608
	<i>Nutrition Deficiency</i>	1554
	<i>White Spot</i>	1560
	<i>Healthy Leaves</i>	1641
Validation	<i>Bacterial Spot</i>	443
	<i>Curl Virus</i>	374
	<i>Cercospora Leaf Spot</i>	392
	<i>Nutrition Deficiency</i>	446
	<i>White Spot</i>	440
	<i>Healthy Leaves</i>	359

- Melakukan penyeimbangan *Test Dataset* dengan menyamakan jumlah data gambar pada setiap kelas. Rasio pemisahan atau penyeimbangan sebuah kumpulan data seharusnya ditentukan berdasarkan ukurannya [12]. Pendekatan ini bertujuan untuk memastikan setiap kelas memiliki representasi yang adil, sehingga dapat mengurangi bias pada saat evaluasi kinerja model. *Original Dataset* dialokasikan sebagai *Test Dataset* untuk mensimulasikan pengujian model di dunia nyata, karena dataset ini berisi citra asli tanpa augmentasi. Dataset ini kemudian diseimbangkan dengan teknik *undersampling* dan *oversampling* untuk menghasilkan jumlah data yang seimbang. Data di setiap kelas diseimbangkan hingga masing-masing berisi 200 gambar, sehingga menghasilkan total 1200 gambar. Penetapan jumlah ini

didasarkan pada proporsi pengujian yang ideal, di mana 1200 gambar tersebut mewakili 10-20% dari keseluruhan dataset. Hasil penyeimbangan data pada Test Dataset dapat dilihat pada Tabel 4 di bawah.

Tabel 4. Penyeimbangan data pada Test Dataset

Index	Kelas	Data
1	<i>Bacterial Spot</i>	200
2	<i>Curl Virus</i>	200
3	<i>Cercospora Leaf Spot</i>	200
4	<i>Nutrition Deficiency</i>	200
5	<i>White Spot</i>	200
6	<i>Healthy Leaves</i>	200
Total		1200

2.3 Tahap Model Development

Untuk membangun model klasifikasi penyakit pada daun cabai, model dikembangkan dengan mengadopsi pendekatan *transfer learning*. Proses ini melibatkan penggunaan arsitektur *pre-trained* sebagai basis ekstraksi fitur, yang kemudian dilatih dalam dua tahap utama yaitu tahap *feature extraction* dan tahap *fine-tuning* yang bertujuan untuk mencapai performa optimal pada dataset spesifik penelitian ini.

2.3.1 Arsitektur Base Model

Model akan dibangun dengan menggunakan MobileNetV2 sebagai *base model* dengan melakukan *fine-tuning* model. Arsitektur ini merupakan pengembangan dan penyempurnaan dari MobileNetV1 oleh Sandler et al., yang dirancang untuk mengatasi kendala keterbatasan memori dan komputasi pada CNN tradisional. Arsitektur ini dioptimalkan untuk aplikasi *on-device*, yang menjadikannya pilihan praktis pada perangkat dengan sumber daya terbatas. Arsitektur ini memanfaatkan struktur *inverted residual* dan *linear bottleneck*, yang secara unik membalik urutan blok residual tradisional. Struktur ini secara efisien memperluas dimensi fitur menggunakan konvolusi 1x1, menerapkan proses filtering dengan konvolusi depthwise 3x3, dan kemudian memproyeksikannya kembali ke dimensi yang lebih rendah, sehingga secara signifikan mengurangi jumlah parameter dan beban komputasi [13].

2.3.2 Desain Arsitektur Model

Model klasifikasi dibangun dengan menggabungkan *base model* MobileNetV2 dengan sebuah *classifier head* kustom. Proses pelatihan ini dibagi menjadi dua tahap. Pada tahap *feature extraction*, seluruh layer pada base model MobileNetV2 dibekukan. Selanjutnya sebuah classifier head ditambahkan di atasnya, yang diawali dengan lapisan GlobalAveragePooling2D. Output

dari lapisan ini kemudian diproses oleh satu lapisan Dense dengan 256 neuron dan fungsi aktivasi ReLU. Untuk mencegah *overfitting*, teknik regularisasi Dropout dengan rate 0,4 dan 0,2 serta regularisasi L2 dengan faktor lambda 0,001 diterapkan sebelum menuju lapisan Dense output. Seperti yang ditunjukkan pada Gambar 2, arsitektur pada tahap ini memiliki 329.478 parameter yang dapat dilatih.

Selanjutnya pada tahap *fine-tuning*, sebagian lapisan teratas dari *base model* MobileNetV2 dilatih atau *unfreezing* untuk diperbarui bobotnya. Secara spesifik, proses *unfreezing* dimulai dari lapisan *block_15_expand* hingga lapisan terakhir dari base model, sementara semua lapisan sebelumnya tetap dibekukan. Pada Gambar 3, dapat dilihat jumlah parameter yang dapat dilatih dari 329.478 pada tahap feature extraction menjadi 1.535.558 pada tahapan fine-tuning.

Layer (type)	Output Shape	Param #
input_image (InputLayer)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_avg_pool (GlobalAveragePooling2D)	(None, 1280)	0
head_dropout_1 (Dropout)	(None, 1280)	0
head_dense_256 (Dense)	(None, 256)	327,936
head_dropout_2 (Dropout)	(None, 256)	0
output_softmax (Dense)	(None, 6)	1,542

Total params: 2,587,462 (9.87 MB)
 Trainable params: 329,478 (1.26 MB)
 Non-trainable params: 2,257,984 (8.61 MB)

Gambar 2. Model Summary pada tahap Feature Extraction

Layer (type)	Output Shape	Param #
input_image (InputLayer)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_avg_pool (GlobalAveragePooling2D)	(None, 1280)	0
head_dropout_1 (Dropout)	(None, 1280)	0
head_dense_256 (Dense)	(None, 256)	327,936
head_dropout_2 (Dropout)	(None, 256)	0
output_softmax (Dense)	(None, 6)	1,542

Total params: 2,587,462 (9.87 MB)
 Trainable params: 1,535,558 (5.86 MB)
 Non-trainable params: 1,051,904 (4.01 MB)

Gambar 3. Model Summary pada tahap Fine-Tuning

2.3.3 Konfigurasi Pelatihan Model

Proses pelatihan untuk kedua tahap dijalankan dengan ukuran *batch size* sebesar 32 dan konfigurasi *hyperparameter* lain yang terdefinisi pada Tabel 5. Optimizer Adam digunakan pada kedua tahap, namun dengan learning rate yang jauh lebih rendah pada tahap *fine-tuning* untuk mencegah *catastrophic forgetting*. Untuk mengatasi ketidakseimbangan kelas pada data *training*, *class_weight* dengan strategi *balanced* diterapkan. Proses training juga dilengkapi dengan callbacks EarlyStopping dan

ReduceLRonPlateau untuk efisiensi dan optimalisasi.

Tabel 5. Konfigurasi Hyperparameter untuk Pelatihan

Parameter	Tahap Feature Extraction	Tahap Fine-Tuning
Optimizer	Adam	Adam
Initial Learning Rate	5e-4 (0,0005)	5e-6 (0,000005)
Loss Function	Categorical Cross-Entropy (label smoothing =0.1)	Categorical Cross-Entropy (label smoothing =0.1)
Batch Size	32	32
Epochs	40	25
Patience (EarlyStopping)	7	4
Patience (ReduceLR)	5	2

2.4 Tahap Model Evaluation

Setelah tahap *model development*, performa dan kinerja model akan di evaluasi dengan menggunakan *Confusion Matrix* dan *Evaluation Metrics*. *Original Dataset* akan digunakan sebagai *Test Set* pada tahapan ini karena citra gambar harus merepresentasikan pengujian nyata di lapangan, yang berarti gambar yang digunakan adalah gambar asli tanpa adanya augmentasi seperti yang dilakukan saat *training* pada model. *Confusion Matrix* merupakan metode analisis yang merangkum informasi dengan menghitung frekuensi setiap kemungkinan hasil prediksi yang dibuat model terhadap dataset uji. Selain itu, metode ini dapat sekaligus menjadi landasan untuk menghitung metrik lainnya [14]. Pada tabel 4 di bawah merupakan contoh tabel confusion matrix, khususnya untuk klasifikasi biner.

Tabel 6. Confusion Matrix

		Prediction	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	TN

Berdasarkan tabel tersebut, berikut adalah empat kemungkinan hasil prediksi:

- True Positives* (TP) yaitu representasi jumlah kasus di mana data sebenarnya tergolong positif dan model juga berhasil memprediksinya sebagai positif.
- True Negatives* (TN) yaitu representasi jumlah kasus di mana data sebenarnya tergolong negatif dan model juga berhasil memprediksinya sebagai negatif.
- False Positives* (FP) yaitu representasi jumlah kasus kesalahan di mana data sebenarnya tergolong negatif, namun model memprediksinya sebagai kelas positif.
- False Negatives* (FN) yaitu representasi jumlah kasus kesalahan di mana data sebenarnya tergolong positif, namun model memprediksinya sebagai kelas negatif.

Sedangkan *evaluation metrics* digunakan untuk menghitung informasi yang tersaji dalam confusion matrix menjadi nilai-nilai numerik. Dalam penelitian ini, metrik yang akan digunakan meliputi akurasi, presisi, recall, dan F1-Score yang akan menjadi landasan untuk melakukan analisis komparatif terhadap kinerja dan performa model yang telah dikembangkan.

Akurasi (*accuracy*) merupakan metrik yang menghitung seberapa banyak prediksi yang benar secara keseluruhan pada model. Untuk mengukur akurasi, dapat digunakan rumus pada persamaan (1) berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Presisi (*precision*) digunakan untuk menghitung seberapa banyak prediksi positif yang benar positif pada model. Untuk mengukur presisi, dapat digunakan rumus pada persamaan (2) berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall digunakan untuk menghitung seberapa baik model dalam mengklasifikasikan seluruh kasus positif yang sebenarnya. Untuk mengukur *recall*, dapat digunakan rumus pada persamaan (3) berikut:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

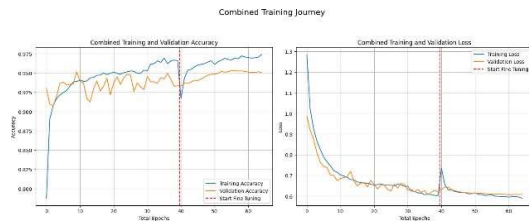
Terakhir yaitu metrik *F1-Score* yang menyeimbangkan nilai *precision* dan *recall* ke dalam satu skor tunggal melalui perhitungan rata-rata harmonik. Metrik ini berfungsi mengukur keseimbangan antara *false positives* (yang dipengaruhi oleh *precision*) dan *false negatives* (yang dipengaruhi oleh *recall*). Untuk mengukur *F1-Score*, dapat digunakan rumus pada persamaan (4) berikut:

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

3. RESULT AND ANALYSIS

3.1 Hasil Pelatihan Model

Proses pelatihan model klasifikasi penyakit pada daun cabai yang terdiri dari tahap *feature extraction* dan tahapan *fine-tuning*, dianalisis berdasarkan kurva akurasi dan loss terhadap epoch untuk memantau proses pembelajaran model. Dapat dilihat pada Gambar 4, secara keseluruhan grafik menunjukkan proses konvergensi yang sehat dan stabil, di mana kurva *training* dan *validation* bergerak secara selaras tanpa adanya indikasi *overfitting* yang signifikan.



Gambar 4. Combined Training Journey pada model MobileNetV2

Pada tahap *feature extraction*, pelatihan dimulai dari epoch 1 hingga epoch 40. Dengan melatih *classifier head*, model dengan cepat mampu mencapai performa yang sangat baik. Grafik pertama yang menunjukkan tren akurasi model, mencapai hasil akurasi sebesar 96,41% dengan validasi akurasi sebesar 93,32% di epoch ke 40. Sedangkan pada grafik kedua yang menunjukkan tren loss model, mencapai nilai training loss 0,6030 dengan validasi loss 0,6207. Puncak performa pada tahap *feature extraction* ini terjadi pada epoch ke 37 dengan validasi akurasi mencapai 94,95% dan nilai validasi loss sebesar 0,6057 sehingga bobot model dikembalikan pada epoch ini. Keberhasilan ini membuktikan kekuatan fitur pre-trained dari MobileNetV2 yang dapat beradaptasi secara efektif pada dataset penelitian.

Setelah tahap *feature extraction* selesai, tahap fine-tuning dilanjutkan untuk melakukan penyesuaian pada sebagian lapisan dari *base model*. Pelatihan pada tahap ini dimulai dari epoch ke 41 hingga epoch ke 65. Dapat dilihat pada Gambar 4, pada kedua grafik setelah penanda *Start Fine-Tuning*, tahap ini berhasil memberikan peningkatan performa lebih lanjut pada model. Tren akurasi model, training akurasi mencapai nilai sebesar 97,26% dengan validasi akurasi sebesar 95,03% di epoch ke 65. Sedangkan pada tren loss model, training loss mencapai nilai 0,5896 dengan validasi loss 0,6073. Puncak performa pada tahap *fine-tuning* ini terjadi pada epoch ke 61 dengan validasi akurasi mencapai 95,11% dan nilai validasi loss sebesar 0,6054 sehingga bobot model dikembalikan pada epoch ini. Secara keseluruhan, strategi pelatihan dua tahap ini terbukti sangat efektif. Tahap *feature extraction* berhasil membangun *base model* yang kuat dengan cepat, sementara tahap *fine-tuning* memberikan optimasi akhir yang berhasil menyempurnakan performa model pada data validasi sebelum dihentikan secara efisien oleh callback *EarlyStopping*.

3.2 Evaluasi Model

Setelah proses pelatihan selesai dan model terbaik dari tahap *fine-tuning* diperoleh, evaluasi akhir dilakukan untuk mengukur performa generalisasi model dengan menggunakan *test dataset*.

3.2.1 Hasil Test Model

Pengujian model dengan 1.200 citra data uji menunjukkan performa yang sangat tinggi. Pada Gambar 5 dibawah, dapat dilihat bahwa model berhasil mencapai akurasi pengujian sebesar 98,17% dengan nilai loss akhir 0,5536. Hasil ini mengonfirmasi bahwa model tidak hanya berkinerja baik pada data validasi, tetapi juga mampu menggeneralisasi polanya dengan sangat baik pada saat evaluasi model. Hal ini membuktikan robustitas model yang telah dikembangkan.

```
[ ] print("\nMemulai evaluasi model...")
test_loss, test_accuracy = model_fine_tuning.evaluate(eval_generator)
print(f"Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}")

Memulai evaluasi model...
38/38 ————— 453s 12s/step - accuracy: 0.9868 - loss: 0.5482
Test Loss: 0.5536, Test Accuracy: 0.9817
```

Gambar 5. Hasil test model dengan Test Dataset

3.2.2 Analisis Performa Model per Kelas

Agar mendapatkan pemahaman yang lebih detail mengenai kemampuan model dalam mengklasifikasikan setiap kelas, dilakukan analisis terhadap metrik *precision*, *recall*, dan *F1-score*. Secara keseluruhan, pada Gambar 6 dibawah menunjukkan bahwa model memiliki performa yang sangat seimbang dan tinggi di semua kelas, dengan nilai F1-score berkisar antara 0,97 hingga 1,00. Performa sempurna pada metrik *precision* dan *recall* berhasil dicapai oleh kelas *Curl Virus* dan *Nutrition Deficiency*, yang menandakan model mampu mengidentifikasi kedua kondisi ini tanpa kesalahan pada data uji. Sedangkan nilai *precision* paling rendah diperoleh pada kelas *Healthy Leaf* dengan nilai sebesar 0,94. Hal ini mengindikasikan adanya beberapa citra dari kelas lain yang keliru diklasifikasikan sebagai daun sehat. Sebaliknya, nilai *recall* pada kelas *White Spot* adalah 0,94, yang berarti terdapat sejumlah kecil citra *White Spot* yang gagal dikenali dan keliru diprediksi sebagai kelas lain.

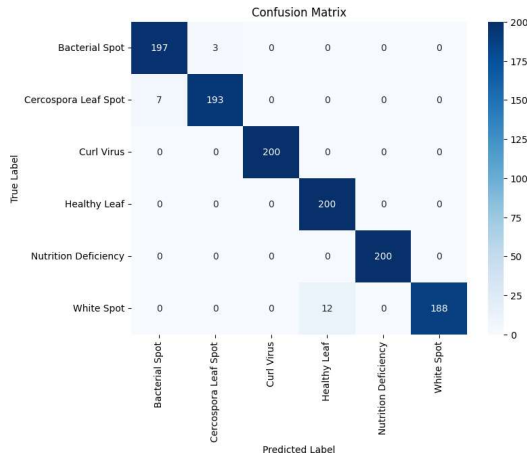
Classification Report:				
	precision	recall	f1-score	support
Bacterial Spot	0.97	0.98	0.98	200
Cercospora Leaf Spot	0.98	0.96	0.97	200
Curl Virus	1.00	1.00	1.00	200
Healthy Leaf	0.94	1.00	0.97	200
Nutrition Deficiency	1.00	1.00	1.00	200
White Spot	1.00	0.94	0.97	200
accuracy			0.98	1200
macro avg	0.98	0.98	0.98	1200
weighted avg	0.98	0.98	0.98	1200

Gambar 6. Classification Report model

3.2.3 Analisis Kesalahan (Error Analysis)

Pada tahap ini, Confusion Matrix digunakan untuk menganalisis kesalahan klasifikasi kelas lebih lanjut. Dapat dilihat pada Gambar 7 dibawah, model mampu memprediksi seluruh kelas dengan baik,

namun teridentifikasi terdapat beberapa kesalahan minor. Kesalahan klasifikasi terbanyak terjadi saat model salah mengklasifikasikan 12 citra *White Spot* sebagai *Healthy Leaf*. Selain itu, terdapat sedikit kebingungan antara kelas *Bacterial Spot* dan *Cercospora Leaf Spot*.



Gambar 7. Confusion Matrix

Untuk memahami penyebab kesalahan ini, berikut beberapa citra yang salah diklasifikasikan atau *misclassified* yang ditampilkan pada Gambar 8 dibawah. Berdasarkan observasi, dapat dihipotesiskan bahwa kesalahan klasifikasi terjadi karena beberapa faktor. Pada kasus *White Spot* yang diprediksi sebagai *Healthy Leaf*, terlihat bahwa gejala bintik putih pada beberapa citra sangat samar, berukuran kecil, atau tumpang tindih dengan refleksi cahaya, sehingga sulit dibedakan dari daun sehat normal. Demikian pula, kebingungan antara *Bacterial Spot* dan *Cercospora Leaf Spot* yang kemungkinan disebabkan oleh kemiripan ciri pada tekstur dan warna bercak kuning-kecoklatan pada tahap awal infeksi. Hal ini menunjukkan bahwa tantangan utama model terletak pada kasus-kasus dengan gejala yang ambigu atau kurang jelas.



Gambar 8. Misclassified Image saat evaluasi model

4. CONCLUSION

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa model deteksi penyakit pada daun cabai dengan menggunakan MobileNetV2 sebagai base model dan melatih model dengan pendekatan fine-tuning mampu menghasilkan model dengan performa yang sangat baik. Model ini mampu mengklasifikasikan 6 kondisi pada daun cabai dengan total akurasi 98,17% di ikuti dengan hasil precision, recall, dan f1-score mencapai 98%. Selain itu analisis eror saat melakukan evaluasi model, dapat ditarik kesimpulan bahwa model mengalami kebingungan dalam mengklasifikasikan beberapa kelas yang memiliki beberapa ciri yang serupa seperti pada *Bacterial Spot* dan *Cercospora Leaf Spot*, serta Image yang menampilkan gambar penyakit yang sedikit samar seperti pada kelas *White Spot*.

Berdasarkan hasil dan keterbatasan yang teridentifikasi, beberapa arah pengembangan dapat dieksplorasi untuk penelitian di masa depan. Untuk meningkatkan robustisitas model, disarankan untuk memperbanyak dataset, tidak hanya dengan menambah jumlah kelas penyakit, tetapi juga dengan menyertakan citra dengan variasi yang lebih tinggi seperti perbedaan kondisi pencahayaan dan latar belakang yang diambil langsung di lapangan. Selain itu, untuk mengatasi kesulitan model dalam membedakan kelas dengan ciri visual yang sangat mirip seperti antara *Bacterial Spot* dan *Cercospora Leaf Spot*, pendekatan *fine-grained visual classification* (FGVC) dapat dieksplorasi lebih lanjut. Kebutuhan pendekatan ini juga disorot pada penelitian terdahulu yang mengidentifikasi bahwa klasifikasi penyakit daun dengan gejala serupa yang memiliki masalah masalah *fine-grained image classification* [15]. Sebagai langkah implementasi praktis, pengembangan selanjutnya dapat difokuskan pada konversi model ke format yang dioptimalkan seperti TensorFlow Lite dan mengintegrasikannya ke dalam aplikasi mobile, sehingga dapat menjadi alat bantu diagnosis yang cepat dan mudah diakses oleh para petani.

5. ACKNOWLEDGEMENTS

Penulis mengucapkan terima kasih kepada Program Studi D4 Teknologi Informatika Multimedia Digital, Jurusan Teknik Komputer, dan Politeknik Negeri Sriwijaya atas segala dukungan dan fasilitas yang telah diberikan selama pelaksanaan penelitian ini hingga bisa terselesaikan dengan baik dan lancar.

6. REFERENCES

[1] S. Swastika, D. Pratama, T. Hidayat, and K. B. Andri, *TEKNOLOGI BUDIDAYA CABAI MERAH*. Riau: Universitas Riau UR PRESS, 2017.

- [2] Saefudin and S. Wahyuningsih, *ANALISIS KINERJA PERDAGANGAN CABAI MERAH*, vol. 14. Jakarta: Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian, 2024.
- [3] R. H. Misqi and T. Karyani, "ANALISIS RISIKO USAHATANI CABAI MERAH BESAR (*Capsicum annum* L.) DI DESA SUKALAKSANA KECAMATAN BANYURESMI KABUPATEN GARUT RISK ANALYSIS OF RED CHILI (*Capsicum annum* L.) FARMING IN SUKALAKSANA VILLAGE, BANYURESMI DISTRICT, GARUT REGENCY," *Jurnal Pemikiran Masyarakat Ilmiah Berwawasan Agribisnis*, vol. 6, no. 1, pp. 65–76, Jan. 2020, doi: <http://dx.doi.org/10.25157/ma.v6i1.2684>.
- [4] N. Sevirasari, C. Adileksana, and A. B. Pratama, *Modul Pembelajaran Praktik Terbaik Budi Daya cabai Merah*. Jakarta Selatan: Yayasan Edufarmers International, 2023.
- [5] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning - A Review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: [10.1109/ACCESS.2021.3069646](https://doi.org/10.1109/ACCESS.2021.3069646).
- [6] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: [10.1186/s40537-021-00444-8](https://doi.org/10.1186/s40537-021-00444-8).
- [7] L. Falaschetti, L. Manoni, D. Di Leo, D. Pau, V. Tomaselli, and C. Turchetti, "A CNN-based image detector for plant leaf diseases classification," *HardwareX*, vol. 12, p. e00363, 2022, doi: [10.17605/OSF.IO/UCM8D](https://doi.org/10.17605/OSF.IO/UCM8D).
- [8] S. Agustiani, R. Aryanti, S. Khotimatul Wildah, Y. T. Arifin, S. Marlina, and T. Misriati, "Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2," *Journal of Informatics Management and Information Technology*, vol. 4, no. 4, pp. 150–157, 2024, doi: [10.47065/jimat.v4i4.411](https://doi.org/10.47065/jimat.v4i4.411).
- [9] M. A. S. Nirob, A. K. M. F. K. SIAM, P. Bishshash, and M. Assaduzzaman, "Chili Plant Leaf Disease and Growth Stage Dataset from Bangladesh," vol. 1, 2025, doi: [10.17632/W9MR3VF56S.1](https://doi.org/10.17632/W9MR3VF56S.1).
- [10] Tim Mucci, "What is Data Leakage in Machine Learning? | IBM." Accessed: Jun. 02, 2025. [Online]. Available: <https://www.ibm.com/think/topics/data-leakage-machine-learning>
- [11] F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island: Manning Publications Co., 2021.
- [12] I. O. Muraina, "IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS," Feb. 2022. [Online]. Available: <https://www.researchgate.net/publication/358284895>
- [13] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif Intell Rev*, vol. 57, no. 4, Apr. 2024, doi: [10.1007/s10462-024-10721-6](https://doi.org/10.1007/s10462-024-10721-6).
- [14] J. D. Kelleher, B. Mac Namee, and A. D'Arcy, *FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS*. Cambridge: MIT Press, 2015.
- [15] P. Wang, T. Niu, Y. Mao, Z. Zhang, B. Liu, and D. He, "Identification of Apple Leaf Diseases by Improved Deep Convolutional Neural Networks With an Attention Mechanism," *Front Plant Sci*, vol. 12, Sep. 2021, doi: [10.3389/fpls.2021.723294](https://doi.org/10.3389/fpls.2021.723294).