

Perbandingan Performa Model *Baseline* dan Optimasi *Hyperparameter EfficientNetB0* dalam Identifikasi Penyakit Daun Cabai

^{1*}Firma Nahwa Firdaus, ²Risa Helilintar, ³Rony Heri Irawan

¹²³ Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹[*firmanaus11@gmail.com](mailto:firmanaus11@gmail.com), ²risa.helilintar@gmail.com, ³rony@unpkediri.ac.id

Penulis Korespondens : Firma Nahwa Firdaus

Abstrak—Penyakit daun cabai sering menyebabkan gagal panen, namun deteksi manual sering kali tidak akurat. Penelitian ini penting untuk menghadirkan solusi identifikasi otomatis berbasis kecerdasan buatan. Metode yang digunakan adalah arsitektur *deep learning* EfficientNet-B0 dengan membandingkan model *baseline* dan pengujian *fine-tuning*. Hasil penelitian menunjukkan model *baseline* mengalami *overfitting* dengan akurasi 62% akibat keterbatasan dataset lokal. Setelah optimasi *fine-tuning* dengan menurunkan laju belajar, akurasi global meningkat menjadi 65%. Model berhasil mengenali kelas virus keriting dengan sangat baik, meski kelas *cercospora* masih terkendala akibat kemiripan visual ekstrem dengan *bacterial spot*. Penelitian ini menyimpulkan bahwa modifikasi parameter mampu memperbaiki pengenalan bentuk daun keriting pada data kecil. Hasil ini penting sebagai fondasi pengembangan sistem deteksi penyakit tanaman lokal yang lebih adaptif.

Kata Kunci—Cabai, deteksi penyakit, *deep learning*, EfficientNet-B0, *fine-tuning*

Abstract—Chili leaf diseases often cause crop failure, while manual detection is frequently inaccurate. This research is significant as it aims to provide an artificial intelligence-based solution for automatic identification. The method employed is the EfficientNet-B0 deep learning architecture, comparing baseline models and fine-tuning experiments. The results show that the baseline model experienced overfitting with an accuracy of 62% due to the limitations of the local dataset. After fine-tuning optimization by lowering the learning rate, the global accuracy increased to 65%. The model was able to recognize the curly virus class very well, although the *cercospora* class remained challenging due to its extreme visual similarity to bacterial spots. This study concludes that parameter modification can improve the recognition of curly leaf patterns in small datasets. These results are important as a foundation for developing more adaptive local plant disease detection systems.

Keywords—Chili, disease detection, deep learning, EfficientNet-B0, fine-tuning.

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Banyak sekali komoditas hortikultura di Indonesia yang memiliki nilai jual tinggi, salah satu di antaranya adalah cabai[1]. Namun budidaya tanaman ini sering terkendala dengan adanya serangan virus dan bakteri yang dapat menyebabkan tanaman cabai mengalami kerusakan. Masalahnya ada di sini, para petani melakukan identifikasi penyakit dengan cara manual, yaitu secara visual. Pendekatan yang dilakukan ini memiliki kelemahan, salah satunya adalah rentan kesalahan dalam melakukan diagnosis oleh petani awam[2]. Hal ini dapat mengakibatkan

kesalahan dalam melakukan penanganan, sehingga dapat memicu kerugian karena hasil panen menurun drastis. Masalah ini menjadi dasar pentingnya pengembangan sistem komputasi yang cerdas yang mampu mengklasifikasikan penyakit tanaman secara objektif dan cepat.

Dalam bidang *computer vision*, arsitektur *Convolutional Neural Network* (CNN) mendominasi penelitian klasifikasi penyakit tanaman. Sebagai contoh, Zakaria dkk. (2022) berhasil mengklasifikasikan penyakit tanaman hias menggunakan CNN dengan akurasi 89,67%[3]. Selanjutnya, penerapan *transfer learning* pada MobileNetV2 oleh Tri B. mampu mendeteksi penyakit daun cabai dengan akurasi 98,15%[4]. Sementara itu, Fadhilatuzzahro dkk. (2025) memanfaatkan optimasi *fine-tuning* pada EfficientNet-B0 untuk mengklasifikasikan penyakit buah jeruk dengan akurasi mencapai 98,46%[5]. Sejalan dengan hal tersebut, Raihan dkk. (2026) membuktikan bahwa penambahan lapisan kustom pada EfficientNet-B0 berbasis *transfer learning* mampu menghasilkan akurasi pengujian hingga 100% pada penyakit mangga [6]. Pentingnya optimasi internal ini diperkuat oleh Aqil (2025) pada penyakit daun padi, di mana optimasi *hyperparameter* seperti *learning rate* dan *batch size* terbukti mendongkrak akurasi model secara signifikan dari 89,7% menjadi 94,5% [7].

Meskipun penelitian-penelitian tersebut telah membuktikan keunggulan berbagai arsitektur CNN dalam domain agrikultur, fokus utama pada studi terdahulu umumnya terletak pada pencapaian akurasi final tanpa melakukan komparasi mendalam terhadap konfigurasi parameter internal model. Oleh sebab itu, penelitian ini bertujuan melakukan analisis perbandingan performa antara model *baseline* EfficientNet-B0 dengan model yang telah melalui proses optimasi *tuning hyperparameter*. Penelitian ini bertujuan membuktikan bahwa penerapan optimasi *hyperparameter* pada arsitektur EfficientNet-B0 dapat meningkatkan akurasi klasifikasi penyakit tanaman cabai serta menghasilkan model dengan tingkat stabilitas yang lebih baik dibandingkan model *baseline*.

II. METODE

Penelitian ini dilakukan melalui beberapa tahapan sistematis, mulai dari pengumpulan data, tahapan pra-pemrosesan citra, perancangan arsitektur model, proses pelatihan dan *hyperparameter tuning*, hingga evaluasi performa model. Alur jalannya tahapan penelitian ini secara menyeluruh dibagi menjadi beberapa fase krusial yang saling berkesinambungan, seperti yang dijelaskan berikut ini.

A. Dataset dan Pra-pemrosesan Citra

Dataset penelitian ini menggunakan 240 citra digital daun tanaman cabai yang terbagi secara seimbang ke dalam 4 kelas kondisi: Daun Sehat, Bercak Daun, Virus Keriting, dan Bacterial Spot. Dataset tersebut kemudian dibagi menjadi data latih sebanyak 200 citra dan data validasi sebanyak 40 citra. Skema pembagian ini dipilih untuk memaksimalkan ketersediaan sampel pada proses pelatihan model deep learning di tengah keterbatasan volume dataset yang ada.

Data validasi digunakan secara independen untuk mengontrol jalannya proses pelatihan, mendeteksi gejala overfitting atau underfitting, serta menjadi acuan metrik evaluasi akhir untuk membandingkan performa model baseline dengan model hasil tuning hyperparameter. Sebelum memasuki tahap pelatihan, dilakukan pra-pemrosesan data (data preprocessing) untuk menyamakan karakteristik citra input, antara lain:

1. *Resizing*

Mengubah resolusi seluruh citra menjadidi 224×224 piksel sesuai dengan standar input arsitektur EfficientNet-B0, agar proses ekstraksi fitur berjalan secara konsisten.

2. *Normalization*

Mengubah nilai piksel citra ke dalam rentang 0 hingga 1 dengan maksud untuk mempercepat proses konvergensi saat pelatihan.

3. *Data Augmentation*

Proses ini bertujuan untuk memperbanyak variasi sampel dan menghindari *overfitting*, pada tahap ini, teknik augmentasi yang diterapkan berupa *horizontal flip*, *rotation*, *width shift*, dan *fill mode*.

B. Arsitektur Model *Baseline* EfficientNet-B0

Dalam penelitian ini, EfficientNet-B0 berbasis transfer learning dipilih sebagai arsitektur dasar. Model ini memanfaatkan metode *compound scaling* untuk menyeimbangkan dimensi kedalaman (*depth*), lebar (*width*), dan resolusi citra input secara bersamaan [8]. Pendekatan tersebut menghasilkan arsitektur ringan dengan parameter efisien dan tetap mempertahankan akurasi dan performa klasifikasi yang tinggi. Formulasi matematis untuk metode *compound scaling* ini dinyatakan melalui persamaan berikut:

$$Depth = \alpha^{\phi} \quad (1)$$

$$Width = \beta^{\phi} \quad (2)$$

$$Resolution = \gamma^{\phi} \quad (3)$$

Pada persamaan tersebut, α , β , dan γ masing-masing merepresentasikan faktor scaling untuk *depth*, *width*, dan *resolution*, sedangkan ϕ merupakan koefisien *scaling model*[9].

Proses implementasi model menggunakan bobot awal dari *pre-trained* ImageNet. Pada bagian akhir arsitektur *baseline*, lapisan *top layer* bawaan dihapus dan digantikan dengan lapisan kustom baru yang terdiri atas *Global Average Pooling*, *Dropout*, serta lapisan *Dense* dengan fungsi aktivasi *Softmax* untuk klasifikasi multi-kelas. Berikut mekanisme komputasi pada *top layer* baru. proses *forward propagation* dijalankan menggunakan sampel citra daun cabai berukuran $224 \times 224 \times 3$.

1. *Global Average Pooling* (GAP)

Tensor fitur X direduksi secara spasial menjadi vektor fitur v berdimensi 1×1280 melalui perhitungan rata-rata nilai piksel pada setiap matriks channel berukuran 7×7 (49 piksel). Sebagai ilustrasi, apabila jumlah total nilai piksel pada salah satu channel adalah 24,5. Maka nilai rata-rata komponen v_i :

$$v_i = \frac{1}{49} \sum_{k=1}^{49} x_k = \frac{24.5}{49} = 0.5$$

2. Dropout

Pada proses pelatihan dengan nilai rate 0,2. sebanyak 20% elemen vektor akan dinonaktifkan secara acak. Elemen yang tetap aktif (V_{drop}) dikalikan dengan faktor skala $\frac{1}{1-0,2} = 1,25$ guna mengurangi risiko overfitting:

$$V_{drop} = v_i \times 1.25 = 0.5 \times 1.25 = 0.625$$

3. Lapisan Dense

Vektor fitur hasil dropout diproses menggunakan perkalian matriks bobot (W) dan penambahan bias (b) menghasilkan nilai mentah prediksi (logits) sebanyak $K = 4$ kelas penyakit tanaman cabai:

$$z = [z_1, z_2, z_3, z_4] = [1,2; 0,5; 0,8; 2,5]$$

4. Fungsi Softmax

Nilai logits kemudian diubah menjadi $P(y = i|x)$ dengan proses normalisasi eksponensial, menjadi $\sum_{j=1}^4 e^{z_j} = e^{1,2} + e^{0,5} + e^{0,8} + e^{2,5} = 3,32 + 1,65 + 2,23 + 12,18 = 19,3$. Adapun proses perhitungannya adalah sebagai berikut:

$$\begin{aligned}\hat{y}_1(\text{Daun sehat}) &= \frac{e^{1,2}}{19,38} = \frac{3,32}{19,38} = 0,171 = 17,1\% \\ \hat{y}_2(\text{Bercak daun}) &= \frac{e^{0,5}}{19,38} = \frac{1,65}{19,38} = 0,085 = 8,5\% \\ \hat{y}_3(\text{Virus keriting}) &= \frac{e^{0,8}}{19,38} = \frac{2,23}{19,38} = 0,115 = 11,5\% \\ \hat{y}_4(\text{Bacterial Spot}) &= \frac{e^{2,5}}{19,38} = \frac{12,18}{19,38} = 0,629 = 62,9\%\end{aligned}$$

Berdasarkan probabilitas terbesar yang diperoleh ($\hat{y}_4 = 74,2\%$), model secara matematis mengategorikan citra masukan ke dalam kelas *Bacterial Spot*.

C. Optimasi Hyperparameter

Pengujian difokuskan pada perbandingan performa antara model baseline (parameter standar) dan model hasil optimasi. Variabel yang dimanipulasi dalam proses tuning hyperparameter meliputi penyesuaian nilai *learning rate*, ukuran *batch size*, serta jenis *optimizer* [10]. Skema pencarian kombinasi parameter terbaik dilakukan secara eksperimental melalui metode manual tuning, di mana variasi konfigurasi diuji secara bertahap untuk menemukan performa model yang paling optimal.

Tabel 1. Skenario *Hyperparameter*

Parameter	Nilai Model <i>Baseline</i>	Variasi Nilai <i>Tuning</i>
<i>Optimizer</i>	Adam	Adam
<i>Learning Rate</i>	1×10^{-3}	1×10^{-5}
<i>Batch Size</i>	32	32
<i>Epoch</i>	10	15 + <i>Early Stopping</i>

D. Analisis dan Metrik Evaluasi

Metrik evaluasi yang digunakan untuk mengukur kinerja dari masing-masing skenario pengujian adalah *Accuracy*, *Precision*, *Recall*, dan *F1-Score* [11]. Formula matematis yang digunakan untuk menghitung nilai metrik tersebut didasarkan pada komponen Confusion Matrix, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Sebagai skenario komputasi 100 sampel per kelas, total $N = 400$ dengan contoh kasus pada kelas *Bacterial Spot*, distribusi sampel ditentukan sebagai berikut, 90 sampel *Bacterial Spot* yang diprediksi benar (TP), 10 sampel *Bacterial Spot* yang salah diprediksi sebagai kelas lain (FN), 5 sampel dari kelas lain yang salah diprediksi sebagai *Bacterial Spot* (FP), serta 295 sampel non-*Bacterial Spot* yang berhasil diprediksi dengan benar (TN). Berdasarkan distribusi nilai tersebut, mekanisme perhitungan metrik evaluasi dirumuskan melalui persamaan berikut:

1. *Accuracy*

Mengukur rasio total prediksi benar (positif dan negatif) terhadap keseluruhan 400 sampel data uji.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Accuracy = \frac{90 + 295}{400} = \frac{385}{400} = 0,962 = 96,2\%$$

2. *Precision*

Mengukur tingkat ketepatan model dalam memprediksi kelas positif dari keseluruhan sampel yang dinyatakan sebagai *Bacterial Spot*:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Precision = \frac{90}{90 + 5} = \frac{90}{95} = 94,7\%$$

3. *Recall*

Mengukur keberhasilan model dalam menemukan kembali seluruh sampel positif aktual:

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$Recall = \frac{90}{90 + 10} = \frac{90}{100} = 0,900 = 90\%$$

4. *F1-Score*

Mengukur nilai rata-rata antara *Precision* dan *Recall* untuk menunjukkan keseimbangan performa model pada kelas tersebut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

$$mF1 - Score = 2 \times \frac{0,947 \times 0,900}{0,947 + 0,900} = 2 \times \frac{0,852}{1,847} = 0,922 = 92,2\%$$

Hasil pengukuran dari semua metrik di atas pada data uji akan dianalisis secara komparatif untuk menentukan tingkat stabilitas dan peningkatan performa yang dihasilkan oleh model setelah optimasi dibandingkan dengan model *baseline*.

E. Perangkat Penelitian

Pada proses pengujian model *deep learning* dalam penelitian ini, alat dan spesifikasi komputasi yang digunakan berupa laptop dengan prosesor Intel Core i3, RAM berkapasitas 4,00 GB, serta ditopang oleh sistem arsitektur berbasis 64-bit komputasi. Kemudian untuk perangkat lunak yang digunakan meliputi sistem operasi Windows 11, dengan menggunakan bahasa pemrograman Python, Tensorflow, dan Keras sebagai *framework deep learning* untuk membangun serta melatih model EfficientNet-B0, serta *jupyter Notebook* sebagai lingkungan dan eksekusi program penelitian.

III. HASIL DAN PEMBAHASAN

A. Karakteristik Dataset Daun Cabai

Dataset yang digunakan pada penelitian ini terdiri dari tiga kelas fase penyakit, yang terdiri dari bercak daun, virus keriting, dan *bacterial spot*, serta kelas daun sehat. Karakteristik visual dari masing-masing kelas memiliki tingkat kemiripan tekstur yang cukup tinggi, khususnya pada kemunculan gejala bercak yang terlihat secara visual pada permukaan daun. Tantangan utama dalam klasifikasi ini terletak pada kemiripan visual yang ekstrem antara bercak daun dan *Bacterial Spot* yang sama-sama menunjukkan gejala makroskopis berupa bercak cokelat kehitaman.



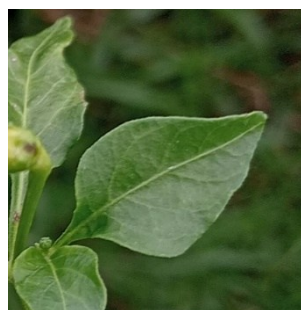
Gambar 2. *Bacterial Spot*



Gambar 3. *Cercospora*



Gambar 3. Virus Keriting



Gambar 4. Daun Sehat

B. Hasil Pengujian Model *Baseline*

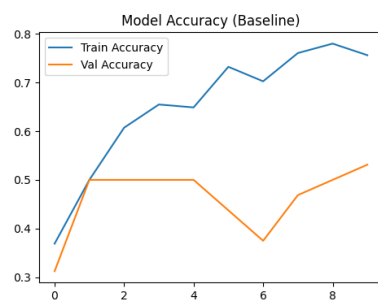
Pada tahap pertama, pengujian tahap pertama dilakukan dengan menggunakan model *baseline* EfficientNet-B0. Di tahap ini, semua kemampuan dasar model dalam mengenali gambar bawaan dari ImageNet dikunci total dengan tujuan mempertahankan bobot asli model agar tidak rusak. Pelatihan hanya difokuskan pada lapisan baru selama 10 epoch menggunakan *learning rate default* dari *optimizer* Adam, yaitu sebesar 1×10^{-3} . Setelah menyelesaikan seluruh siklus

pelatihan, model *baseline* diuji menggunakan data validasi yang belum pernah dilihat sebelumnya. Hasil pengujian menunjukkan bahwa model *baseline* baru mampu menghasilkan tingkat akurasi global sebesar 62% (0.62). Berikut hasil *classification report baseline*:

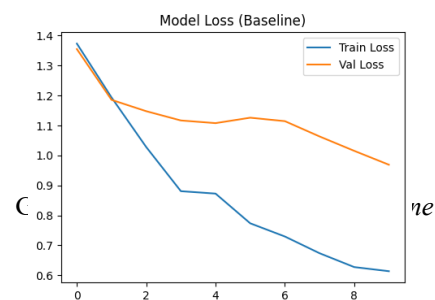
Tabel 2. *Classification Report Baseline*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Bacterial Spot</i>	0,50	0,80	0,62	10
Bercak daun/ <i>Cercospora</i>	1,00	0,10	0,18	10
Virus keriting	1,00	0,60	0,75	10
Daun sehat	0,59	1,00	0,74	10
<i>accuracy</i>			0,62	40
<i>macro avg</i>	0,77	0,62	0,57	40
<i>weighted avg</i>	0,77	0,62	0,57	40

Berdasarkan Tabel 2, model *baseline* menghadapi kendala serius dalam mengenali penyakit bercak daun, dengan nilai *recall* yang hanya mencapai 10%. Rendahnya nilai ini mengindikasikan bahwa dari 10 sampel aktual di lapangan, model hanya mampu mengklasifikasikan 1 citra secara tepat, sedangkan 9 citra lainnya terdeteksi keliru sebagai kelas lain. Sebaliknya, kelas daun sehat memperoleh *recall* sempurna (100%) namun dengan tingkat *precision* yang rendah, yaitu sebesar 59%. Angka ini mencerminkan adanya pola bias pada sistem, di mana model *baseline* cenderung mengambil keputusan aman dengan salah mengklasifikasikan daun yang sebenarnya sakit khususnya gejala ke dalam kategori daun sehat. Rendahnya performa pada tahap awal ini disebabkan oleh bobot *pre-trained* ImageNet yang belum adaptif terhadap fitur-fitur mikro serupa pada objek daun tanaman lokal.



Gambar 5. Model Accuracy



Gambar 6. Model Loss

Berdasarkan Gambar 5 dan Gambar 6, terdapat jarak (*gap*) yang lebar antara kurva latihan dan validasi. Akurasi latihan hampir mencapai 80% sedangkan akurasi validasi tertahan di 53%, dipertegas oleh penurunan *Train Loss* yang kontras dengan *Val Loss* yang stagnan di atas 0,9. Pola ini mengindikasikan model *baseline* mengalami *overfitting*. Masalah tersebut dipicu oleh penggunaan bobot umum ImageNet yang belum adaptif terhadap fitur spesifik daun cabai lokal, serta keterbatasan volume dataset yang menyebabkan arsitektur besar model cenderung menghafal data latih alih-alih membangun kemampuan generalisasi pada citra baru.

C. Hasil Optimasi *Hyperparameter*

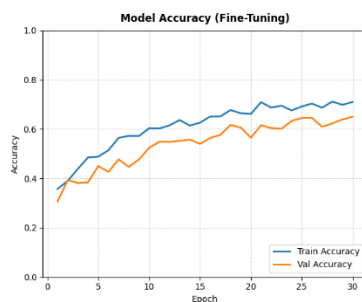
Tahap *fine-tuning* dilakukan dengan membuka seluruh lapisan arsitektur dan menurunkan *learning rate* menjadi 1×10^{-5} untuk memperbaiki masalah *overfitting*. Setelah diuji, akurasi global model meningkat menjadi 65%, persentase naik 3% dari *baseline*. Rincian performa akhir ini dapat dilihat pada Tabel 3.

Tabel 3. *Classification Report Model Fine-Tuning*

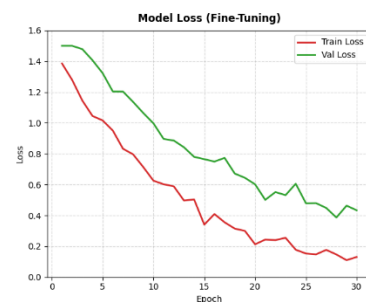
	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Bacterial Spot</i>	0,47	0,70	0,56	10
Bercak daun/ <i>Cercospora</i>	1,00	0,10	0,18	10
Virus keriting	1,00	0,80	0,89	10
Daun sehat	0,62	1,00	0,77	10
<i>accuracy</i>			0,65	40
<i>macro avg</i>	0,77	0,65	0,60	40
<i>weighted avg</i>	0,77	0,65	0,60	40

Berdasarkan tabel 3, optimasi ini membawa perubahan positif yang signifikan pada beberapa kelas. Peningkatan paling kontras terlihat pada kelas virus keriting, di mana nilai *recall* melonjak dari 60% menjadi 80% yang membuktikan bahwa model menjadi jauh lebih pintar dalam mengenali pola daun yang keriting. Selain itu, performa pada kelas daun sehat juga memuaskan karena nilai *recall* berhasil dipertahankan di angka sempurna yaitu 100%.

Namun, akurasi total model akhirnya tertahan di angka 65% karena kelas bercak daun/*cercospora* tetap stagnan dengan nilai *recall* sebesar 10%. Masalah ini disebabkan oleh dua faktor utama, yaitu kemiripan ekstrem antara karakteristik bercak *cercospora* dan *bacterial spot* sehingga sulit dibedakan oleh model dengan laju belajar rendah dalam waktu singkat, serta jumlah sampel (*support*) data uji yang sedikit, yaitu hanya 10 gambar sehingga membuat metrik penilaian menjadi sangat sensitif karena salah menebak beberapa gambar saja akan langsung menurunkan skor persentase secara drastis.



Gambar 7. Grafik Akurasi *Fine-Tuning*



Gambar 8. Grafik Loss *Fine-Tuning*

Model hasil *fine-tuning* pada Gambar 7 dan Gambar 8 menunjukkan pola pergerakan kurva yang konvergen dan sehat. Seiring bertambahnya *epoch*, kurva akurasi mengalami tren kenaikan dengan akurasi validasi akhir sebesar 65%, sementara kurva *loss* mengalami penurunan yang stabil hingga mencapai nilai 0,43 pada sisi validasi.

IV. KESIMPULAN

Penelitian ini berhasil menjawab tujuan utama mengenai penerapan arsitektur *deep learning* EfficientNet-B0 berbasis *transfer learning* untuk melakukan klasifikasi citra penyakit pada daun tanaman cabai lokal secara otomatis. Berdasarkan eksperimen yang telah dilakukan, pemanfaatan bobot *pre-trained* ImageNet terbukti memberikan kontribusi signifikan dalam membantu model melakukan generalisasi fitur visual dasar secara mandiri, sehingga mampu menghasilkan akurasi global sebesar 65% meskipun dihadapkan pada keterbatasan volume dataset lokal. Optimasi melalui tahapan *fine-tuning* dengan penurunan laju belajar (*learning rate*) secara terukur terbukti mampu meningkatkan kemampuan model dalam mengenali karakteristik morfologi penyakit yang kompleks, seperti yang ditunjukkan oleh lonjakan akurasi pengenalan pada kelas penyakit virus keriting secara signifikan. Meskipun demikian, peningkatan akurasi global ini belum sepenuhnya merata di seluruh kelas akibat kendala kemiripan fitur visual bercak daun yang ekstrem serta keterbatasan variasi sampel data di lapangan.

Sebagai saran untuk pengembangan penelitian selanjutnya di bidang sains komputer dan kecerdasan buatan, diperlukan penerapan teknik augmentasi data tingkat lanjut, seperti manipulasi warna atau pencahayaan khusus, guna mempertegas batas piksel halus pada penyakit bertekstur bercak yang serupa. Selain itu, perluasan volume dataset secara berkala serta penambahan durasi pelatihan dengan parameter adaptif sangat direkomendasikan agar model dapat mengeliminasi bias komputasi dan mencapai performa klasifikasi yang lebih tangguh dan menyeluruh sebelum diintegrasikan ke dalam sistem deteksi tanaman di sektor pertanian nyata.

DAFTAR PUSTAKA

- [1] J. Jayaputra, I. K. D. Jaya, and B. B. Santoso, "Pengembangan Hortikultura Lahan Kering dan Penerapan Teknologi Budidaya Ramah Lingkungan untuk Meningkatkan Pendapatan Petani Di Desa Sukadana Lombok Utara," *Jurnal Gema Ngabdi*, vol. 4, no. 3, pp. 262–272, Nov. 2022, doi: 10.29303/jgn.v4i3.291.
- [2] R. Ardianzah Pangka and B. Ali, "Sistem Pakar Diagnosis Penyakit Pada Tanaman Jagung Menggunakan Metode Forward Chaining Berbasis Website," 2024. doi: 10.53769/bandwidth.v2i1.594.
- [3] M. S. A. Zakaria, A. Faisol, and F. X. Ariwibisono, "Implementasi Convolutional Neural Network Untuk Deteksi Dan Klasifikasi Penyakit Tanaman Hias Berdasarkan Citra Daun," *Indonesian Journal of Humanities and Social Sciences*, vol. 5, Dec. 2024, doi: 10.33367/ijhass.v5i4.6397.
- [4] B. N. Tri, C. Ningrum, E. N. Ni'mah, M. P. Arifin, A. Dusea, and W. Dara, "Klasifikasi dan Pengenalan Pola Penyakit Cabai dengan Metode CNN (Convolution Neural Network)," 2024. doi: 10.29407/stains.v3i1.4137.
- [5] M. Fadhilatuzzahro, A. Fazrin, Z. I. Arishandy, T. A. Sukoco, and H. Maulana, "Optimalisasi Model CNN EfficientNet-B0 dengan Fine Tuning untuk Klasifikasi Penyakit Buah Jeruk," *Jurnal Ilmiah Teknologi Informasi dan Robotika*, 2025, doi: 10.33005/jifti.v7i1.162.
- [6] M. Raihan, C. Wulandari, and R. Kurniawan, "Klasifikasi Penyakit Tanaman Mangga Menggunakan Citra Daun Dengan Pendekatan Transfer Learning Efficientnet-B0," *Journal of Informatics Management and Information Technology*, vol. 6, no. 1, p. 82, Jan. 2026, doi: 10.47065/jimat.v6i1.904.

- [7] I. Aqil, “Pendeteksian Penyakit Daun Padi Menggunakan Model ResNet50 dengan Optimasi Hyperparameter,” *JCOSIS (Journal Computer Science and Information System)*, Oct. 2025, doi: doi.org/10.61567/jcosis.v2i2.257.
- [8] I. Dewa, A. Pradnya, P. Tentriajaya, G. Ngurah, and L. Wijayakusuma, “Eye Disease Classification Using EfficientNet-B0 Based on Transfer Learning,” Aug. 2025. doi: [10.30871/jaic.v9i4.9743](https://doi.org/10.30871/jaic.v9i4.9743).
- [9] A. Nur Fajrina *et al.*, “Penerapan Arsitektur EfficientNet-B0 Pada Klasifikasi Leukimia Tipe Acute Lymphoblastik Leukimia Application of the EfficientNet-B0 Architecture in the Classification of Acute Lymphoblastic Leukemia,” 2024. doi: [10.30595/jrre.v6i1.22090](https://doi.org/10.30595/jrre.v6i1.22090).
- [10] A. Julianto, A. Sunyoto, D. Ferry, and W. Wibowo, “Optimasi Hyperparameter Convolutional Neural Network untuk Klasifikasi Penyakit Tanaman Padi (Optimization of Convolutional Neural Network Hyperparameters for Classification of Rice Plant Diseases),” Dec. 2022. doi: [10.46764/teknimedia.v3i2.77](https://doi.org/10.46764/teknimedia.v3i2.77).
- [11] R. Ridwan, H. H. Handayani, S. A. P. Lestari, and Y. Cahyana, “Evaluasi Kinerja Algoritma Random Forest Dan Gradient Boosting Untuk Klasifikasi Penyakit Jantung,” *Jurnal Komtika (Komputasi dan Informatika)*, vol. 9, no. 1, pp. 112–124, Jun. 2025, doi: [10.31603/komtika.v9i1.13450](https://doi.org/10.31603/komtika.v9i1.13450).