

ANALISIS PERFORMA RESNET50 VGG16 DAN EFFICIENTNETB0 UNTUK KLASIFIKASI KONDISI VISUAL DAUN CABAI

Hasbi Assidiki, Aswan Supriyadi Sunge, Alhadi Saputra

Teknik Informatika, Universitas Pelita Bangsa

Jl. Inspeksi Kalimalang No.9, Cibat, Cikarang

Assidikihasbi40@gmail.com

ABSTRAK

Daun cabai merupakan bagian penting dalam fotosintesis dan perubahan visualnya dapat mengindikasikan gangguan tanaman. Kemiripan gejala antara *healthy*, *leaf spot*, *leaf curl*, *yellowish*, dan *whitefly* menyebabkan identifikasi manual berpotensi subjektif dan tidak konsisten. Penelitian ini bertujuan membandingkan performa ResNet50, VGG16, dan EfficientNetB0 berbasis *transfer learning* serta menentukan model yang paling seimbang. *Dataset* sekunder Kaggle awalnya terdiri atas 500 citra. Setelah audit duplikasi identik dan kemiripan visual, diperoleh 456 citra bersih yang dibagi terstratifikasi menjadi 364 data *train*, 46 *validation*, dan 46 *test*. Setiap arsitektur dilatih melalui *feature extraction* dan *fine-tuning* menggunakan *split* serta *classification head* yang sama dan diulang dengan *seed* 42, 123, dan 2026. Evaluasi menggunakan *confusion matrix*, *accuracy*, *macro precision*, *macro recall*, *macro F1-score*, waktu pelatihan, waktu *inference*, dan jumlah parameter. EfficientNetB0 memberikan hasil terbaik dengan rata-rata *accuracy* 81,88%, *macro precision* 83,52%, *macro recall* 82,03%, *macro F1-score* 82,06%, 4.055.976 parameter, dan waktu pelatihan rata-rata 9 menit 37 detik. Kesalahan utama terjadi antara *whitefly* dan *leaf spot* serta antara *leaf curl* dan *yellowish*.

Kata kunci : daun cabai, CNN, transfer learning, ResNet50, VGG16, EfficientNetB0

1. PENDAHULUAN

Cabai (*Capsicum annum* L.) merupakan komoditas hortikultura yang produktivitasnya dipengaruhi oleh kondisi kesehatan tanaman. Daun berperan langsung dalam fotosintesis, sehingga perubahan warna, bentuk, dan permukaannya dapat menjadi tanda awal gangguan. Pada data citra, kondisi *healthy*, *leaf spot*, *leaf curl*, *yellowish*, dan *whitefly* memperlihatkan ciri yang sebagian bertumpang tindih. *Leaf spot* menampilkan bercak lokal, *leaf curl* memperlihatkan deformasi atau gulungan, *yellowish* didominasi perubahan warna, sedangkan *whitefly* berkaitan dengan objek kecil atau dampak serangannya. Label tersebut diperlakukan sebagai kategori visual, bukan diagnosis etiologis.

Identifikasi manual dapat dipengaruhi pengalaman pengamat, pencahayaan, sudut pengambilan, latar, dan tingkat keparahan gejala. Daun menggulung dapat berkaitan dengan hama maupun gangguan lain, sedangkan warna kuning dapat muncul sebagai gejala berbagai kondisi. Kemiripan ini menuntut metode klasifikasi citra yang konsisten untuk penyaringan awal tanpa menggantikan pemeriksaan ahli.

CNN mempelajari representasi citra secara bertingkat, tetapi pelatihan dari awal pada data terbatas berisiko tidak stabil dan terlalu menyesuaikan diri dengan data latih. *Transfer learning* memanfaatkan bobot ImageNet, lalu menyesuaikannya pada citra daun melalui *feature extraction* dan *fine-tuning* [1]. Penelitian ini memilih ResNet50, VGG16, dan EfficientNetB0 karena ketiganya mewakili strategi berbeda: *residual connection* untuk jaringan dalam [2], blok konvolusi 3×3 sebagai baseline mapan [3], serta

compound scaling untuk keseimbangan kedalaman, lebar, dan resolusi [4].

Penelitian sebelumnya menunjukkan hasil yang tinggi tetapi sulit dibandingkan langsung. Setyadi dkk. memperoleh akurasi validasi 97,42% pada ResNet50 untuk lima kelas daun cabai [5]. Azhari dkk. melaporkan akurasi validasi 100% menggunakan EfficientNet pada empat kelas daun cabai [6], sedangkan Khoirunnisa dkk. memperoleh akurasi uji 98,62% untuk VGG16 pada tiga kelas cabai rawit [7]. Studi daun tomat dan tebu juga menghasilkan urutan model yang berbeda [8], [9]. Perbedaan objek, kelas, jumlah data, *preprocessing*, pembagian data, dan titik evaluasi membuat angka tersebut tidak setara.

Celah penelitian terletak pada belum tersedianya perbandingan ResNet50, VGG16, dan EfficientNetB0 secara mandiri untuk lima kondisi visual daun cabai menggunakan *dataset*, *split*, *augmentasi*, *classification head*, dan protokol evaluasi yang sama. Sejumlah studi hanya melaporkan *accuracy*, tidak mengunci data *test*, atau tidak menjelaskan variasi antar-run. Penelitian ini mengisi celah tersebut dengan audit duplikasi, data *test* terkunci, tiga *seed*, *confusion matrix*, *macro F1-score*, metrik per kelas, dan ukuran efisiensi. Tujuannya adalah merancang, menguji, serta menentukan model terbaik dengan dasar kualitas klasifikasi, kestabilan, pola kesalahan, dan sumber daya komputasi.

2. TINJAUAN PUSTAKA

2.1. CNN dan Transfer Learning

CNN memproses citra menggunakan konvolusi untuk membentuk *feature map*, fungsi aktivasi untuk memodelkan hubungan nonlinier, dan *pooling* atau *downsampling* untuk merangkum fitur [10]. Untuk lima kelas, keluaran akhir menggunakan *softmax*,

sedangkan *categorical cross-entropy* mengarahkan probabilitas ke kelas yang benar. Dalam *transfer learning*, *backbone* terlatih ImageNet digunakan sebagai pengekstrak fitur. *Classification head* baru memetakan fitur ke lima label. Pada tahap awal *backbone* dibekukan; selanjutnya sebagian lapisan atas dibuka menggunakan *learning rate* lebih kecil agar fitur tingkat tinggi menyesuaikan diri dengan domain daun [1].

2.2. Karakter Arsitektur

Tabel 1. Perbandingan arsitektur

Model	Kelebihan	Keterbatasan
ResNet50	<i>Residual connection</i> mendukung aliran gradien dan fitur kompleks.	Parameter besar; rentan <i>overfitting</i> pada data kecil.
VGG16	Blok 3×3 seragam dan mudah direplikasi sebagai baseline.	Memori dan waktu komputasi tinggi.
Efficient NetB0	<i>Compound scaling</i> memberi keseimbangan akurasi dan efisiensi.	Sensitif terhadap <i>preprocessing</i> dan strategi <i>fine-tuning</i> .

ResNet50 mempelajari pemetaan residual $F(x) + x$ sehingga jaringan memodelkan perubahan terhadap identitas [2]. VGG16 mengandalkan blok konvolusi berulang yang sederhana [3]. EfficientNetB0 menggunakan blok MBConv dan *compound scaling* [4]. Dokumentasi Keras menunjukkan bahwa model lengkap EfficientNetB0 relatif lebih ringkas dibandingkan ResNet50 dan VGG16 [11]. Namun, karakter teoritis tidak menentukan hasil pada domain baru; evaluasi tetap harus dilakukan pada protokol yang sama.

2.3. Sintesis Penelitian Terdahulu

Setyadi dkk. membandingkan beberapa arsitektur pada daun cabai, tetapi memaknai data validasi tanpa *test* terpisah [5]. Azhari dkk. mendukung relevansi EfficientNet pada cabai, namun hanya menguji satu keluarga arsitektur dan pembagian datanya tidak dinyatakan tegas [6]. Khoirunnisa dkk. membandingkan MobileNetV2 dan VGG16 pada tiga kelas [7]. Rozi dkk. menemukan VGG16 lebih baik daripada ResNet50 pada tomat [8], sedangkan Rofiq dan Noercholis menemukan EfficientNetB0 unggul pada tebu [9]. Laksono dan Fathurohman melaporkan akurasi tinggi dari kombinasi VGG16 dan ResNet34 pada data lintas tanaman [12]. Variasi hasil memperkuat perlunya perbandingan terkontrol pada objek cabai.

Perbedaan metode terdahulu dapat dijelaskan oleh tiga kelompok faktor. Pertama, tingkat kesulitan tugas berubah sesuai jumlah kelas dan kemiripan gejalanya. Kedua, data yang diperbesar melalui *augmentasi* tidak setara dengan penambahan citra independen; hasil dapat menjadi optimistis bila transformasi dari sumber yang sama tersebar ke subset berbeda. Ketiga, *validation accuracy* yang digunakan

berulang kali saat menyetel model tidak setara dengan *test accuracy* pada data terkunci. Karena itu, penelitian ini tidak memilih arsitektur dari angka publikasi tertinggi, melainkan menguji ulang ketiganya menggunakan kontrol eksperimen yang seragam.

3. METODE PENELITIAN

3.1. Desain dan Objek Penelitian

Penelitian menggunakan desain kuantitatif eksperimental-komparatif. Objeknya adalah citra daun cabai dari *dataset* publik *Chili plant disease* di Kaggle [13]. Lima label visual ialah *healthy*, *leaf curl*, *leaf spot*, *whitefly*, dan *yellowish*. Unit analisis berupa satu file citra; metadata tanaman, lokasi, varietas, dan kamera tidak tersedia. Oleh karena itu, simpulan dibatasi pada karakter data yang digunakan.

Data diperoleh sebagai data sekunder dengan mengunduh arsip dari halaman sumber, memeriksa lima *folder* kelas, mencocokkan nama label, menghitung berkas, dan mendekode seluruh citra. Struktur aktual dijadikan dasar bila berbeda dari deskripsi halaman. Penelitian tidak menambah citra dari sumber lain dan tidak melakukan pelabelan ulang tanpa ahli, sehingga jejak asal data tetap tunggal dan batas interpretasi label dapat dipertanggungjawabkan.



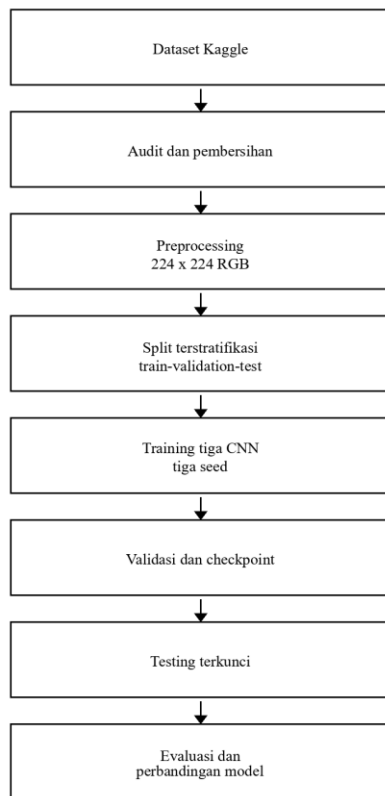
Gambar 1. Contoh lima kelas citra daun cabai

Dataset awal terdiri atas 500 citra seimbang. Audit dilakukan sebelum split dengan memeriksa keterbacaan, format, kanal warna, dimensi, SHA-256 untuk duplikat identik, dan *perceptual hash* dengan ambang jarak Hamming ≤ 5 untuk kandidat visual dekat. Satu representasi beresolusi tertinggi dipertahankan untuk duplikat satu kelas, sedangkan kelompok konflik label dikeluarkan. Audit ulang memastikan tidak ada irisan hash atau pasangan dekat yang tersisa antarsubset.

Tabel 2. Distribusi dataset bersih

Kelas	Train	Val.	Test	Total
<i>Healthy</i>	79	10	10	99
<i>Leaf curl</i>	69	9	9	87
<i>Leaf spot</i>	80	10	10	100
<i>Whitefly</i>	77	10	10	97
<i>Yellowish</i>	59	7	7	73
Total	364	46	46	456

3.2. Alur dan Pengolahan Data



Gambar 2. Kerangka penelitian

Seluruh citra diubah menjadi RGB dan di-resize ke 224×224 piksel. Fungsi *preprocessing* disesuaikan dengan *backbone*. *Augmentasi* berupa *horizontal flip*, rotasi 0,08, dan zoom 0,10 diterapkan hanya pada data *train* setelah split. *Augmentasi* warna tidak digunakan karena warna kuning menjadi ciri penting kelas *yellowish*. Manifest anggota subset disimpan dan dipakai oleh semua arsitektur untuk mengendalikan *data leakage*.

Data dibagi secara terstratifikasi mendekati rasio 80:10:10 menggunakan *seed* 42. *Holdout validation* digunakan karena tiga *backbone* dan dua tahap *training* telah menuntut biaya komputasi besar. Subset *validation* memantau generalisasi, mengaktifkan *early stopping*, dan memilih *checkpoint* berdasarkan *validation loss*. Subset *test* dikunci sampai seluruh keputusan model selesai. Setiap arsitektur diulang menggunakan *seed* 42, 123, dan 2026 pada split yang sama untuk mengukur variasi inisialisasi *classification head* dan urutan *augmentasi*.

3.3. Perancangan dan Pelatihan Model

Ketiga model memuat *backbone* berbobot ImageNet dengan `include_top=False`. Keluaran *backbone* diringkas melalui *GlobalAveragePooling2D*, diikuti *dropout* 0,30 dan Dense lima unit beraktivasi *softmax*. Pada *feature extraction*, *backbone* dibekukan dan *classification head* dilatih menggunakan Adam dengan *learning rate* 10^{-3} dan maksimum 30 *epoch*. Pada *fine-tuning*, 20% lapisan teratas dibuka, *batch normalization* tetap

dibekukan, *learning rate* diturunkan menjadi 10^{-5} , dan pelatihan dijalankan maksimum 15 *epoch*. *Batch size* ditetapkan 16 dan *early stopping* memakai *patience* lima *epoch*.

Tabel 3. Konfigurasi eksperimen

Komponen	Konfigurasi
Masukan	$224 \times 224 \times 3$; bobot ImageNet
Head	$GAP \rightarrow dropout\ 0,30 \rightarrow Dense(5, softmax)$
Tahap 1	Adam; <i>learning rate</i> 10^{-3} ; maksimum 30 <i>epoch</i> ; <i>backbone</i> beku
Tahap 2	20% lapisan dibuka; <i>learning rate</i> 10^{-5} ; maksimum 15 <i>epoch</i>
Kontrol	<i>Patience</i> 5; tiga <i>seed</i> ; split identik

3.4. Validasi dan Evaluasi

Confusion matrix C menempatkan kelas aktual pada baris dan prediksi pada kolom. Accuracy dihitung sebagai jumlah elemen diagonal dibagi seluruh citra. Untuk kelas k digunakan pendekatan *one-versus-rest*: $Precision_k = TP_k / (TP_k + FP_k)$, $Recall_k = TP_k / (TP_k + FN_k)$, dan $F1_k = 2(Precision_k \times Recall_k) / (Precision_k + Recall_k)$. *Macro precision*, *macro recall*, dan *macro F1-score* adalah rata-rata nilai lima kelas. *Macro F1-score* menjadi kriteria utama karena memberi bobot sama pada setiap kelas.

Setiap run mencatat metrik agregat dan per kelas, *confusion matrix*, jumlah parameter, waktu *training*, serta waktu *inference* per citra. Nilai tiga run disajikan sebagai rata-rata dan simpangan baku. Model terbaik dipilih berurutan berdasarkan rata-rata *macro F1-score*, simpangan baku yang lebih kecil, *minimum class recall* yang lebih tinggi, serta efisiensi parameter dan waktu. Aturan ini ditetapkan sebelum *test* agar pemilihan tidak bergantung pada satu run atau satu nilai accuracy.

Teknik holdout mempertahankan satu subset *validation* untuk pemilihan *checkpoint* dan satu subset *test* untuk estimasi akhir. Seluruh arsitektur menerima anggota subset yang identik. Pengulangan *seed* tidak mengubah komposisi data, tetapi mengubah inisialisasi kepala klasifikasi dan urutan *augmentasi*. Oleh karena itu, simpangan baku tiga *seed* ditafsirkan sebagai kestabilan proses *training* pada satu split, bukan sebagai ketidakpastian terhadap seluruh kemungkinan pembagian data.

3.5. Replikasi dan Implementasi

Implementasi disusun dalam modul audit data, pemuatan *dataset*, pembangunan *backbone*, *training* dua tahap, evaluasi, dan peringkasan hasil. Konfigurasi disimpan dalam berkas terstruktur yang memuat ukuran input, *batch size*, *learning rate*, *epoch*, *patience*, *seed*, serta proporsi lapisan *fine-tuning*. Manifest split, log per *epoch*, *checkpoint* terbaik, *confusion matrix*, prediksi *test*, dan metrik per kelas disimpan dengan nama model dan *seed* agar setiap keluaran dapat ditelusuri kembali ke satu run.

Penghitungan metrik menggunakan scikit-learn dengan `zero_division=0` untuk mencegah nilai tak

terdefinisi ketika suatu kelas tidak diprediksi. Jumlah parameter diambil langsung dari model, sedangkan waktu *training* diukur sejak awal *feature extraction* sampai akhir *fine-tuning*. Waktu *inference* dirata-ratakan per citra pada lingkungan yang sama setelah model dimuat. Praktik pencatatan ini mengikuti kebutuhan evaluasi CNN yang dapat direplikasi dan memungkinkan pemeriksaan konsistensi antara tabel agregat, file prediksi, dan *confusion matrix* [14].

4. HASIL DAN PEMBAHASAN

4.1. Audit Data dan Pelatihan

Audit menemukan 11 file duplikat identik dalam kelas sama, 16 file pada kelompok identik dengan konflik label, sembilan file duplikat visual dekat dalam kelas sama, dan delapan file pada kelompok visual dekat dengan konflik label. Setelah tindakan pembersihan, tersisa 456 citra. Audit ulang berstatus lulus: seluruh citra dapat dibaca dan tidak ditemukan duplikat identik maupun pasangan dengan jarak *perceptual hash* ≤ 5 . Pembagian akhir terdiri atas 364 citra *train*, 46 *validation*, dan 46 *test* sebagaimana Tabel 2.

Eksperimen dijalankan pada Windows 11, Python 3.12, dan TensorFlow/Keras 2.21 menggunakan CPU tanpa GPU. Sembilan *run* dihasilkan. Enam *run* memilih *checkpoint fine-tuning* dan tiga *run* memilih *feature extraction*. ResNet50 *seed* 42 dan 2026 serta VGG16 *seed* 2026 tidak memperoleh *validation loss* yang lebih baik pada tahap kedua. Temuan ini menunjukkan bahwa membuka lapisan atas tidak selalu meningkatkan generalisasi pada *dataset* terbatas; *early stopping* dan pemulihan *checkpoint* terbaik diperlukan untuk mengendalikan *overfitting*.

Tabel 4. Ringkasan proses pelatihan

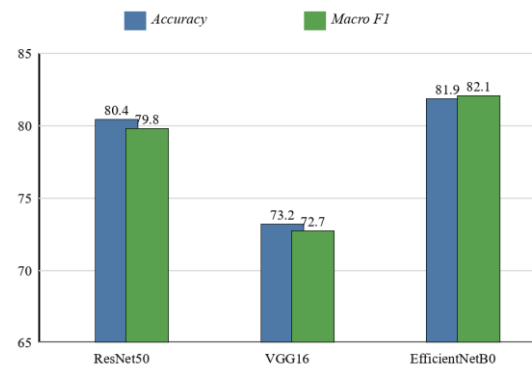
Model	Seed	FE/FT	Tahap	Loss
ResNet50	42	30/6	FE	0,331
ResNet50	123	15/14	FT	0,359
ResNet50	2026	17/14	FE	0,363
VGG16	42	24/8	FT	0,660
VGG16	123	30/6	FT	0,727
VGG16	2026	30/11	FE	0,604
EfficientNetB0	42	30/9	FT	0,447
EfficientNetB0	123	25/11	FT	0,462
EfficientNetB0	2026	30/9	FT	0,461

Kolom FE/FT menyatakan jumlah *epoch feature extraction* dan *fine-tuning* yang dijalankan. Tahap terpilih adalah *checkpoint* dengan *validation loss* terendah. Pola pemilihan menunjukkan bahwa *fine-tuning* bermanfaat pada sebagian besar *run*, tetapi efeknya bergantung pada arsitektur dan *seed*.

4.2. Perbandingan Kinerja

Tabel 5. Rata-rata hasil tiga seed

Model	Acc. (%)	F1 (%)	SD F1
ResNet50	80,43	79,81	2,21
VGG16	73,19	72,74	4,62
EfficientNetB0	81,88	82,06	1,34



Gambar 3. Perbandingan rata-rata accuracy dan macro F1-score

EfficientNetB0 memperoleh rata-rata *accuracy* 81,88%, *macro precision* 83,52%, *macro recall* 82,03%, dan *macro F1-score* 82,06%. Dibandingkan ResNet50, selisihnya 1,45 poin *persentase* pada *accuracy* dan 2,26 poin pada *macro F1-score*. Dibandingkan VGG16, selisihnya 8,70 dan 9,32 poin. Simpangan baku *macro F1-score* EfficientNetB0, yaitu 1,34%, merupakan yang terendah; VGG16 mencapai 4,62%. Akurasi tertinggi satu *run* adalah 82,61% pada ResNet50 *seed* 2026 serta EfficientNetB0 *seed* 123 dan 2026, tetapi rata-rata tiga *seed* memberi dasar yang lebih stabil.

4.3. Pola Kesalahan Model Terbaik

EfficientNetB0 - Akumulasi Tiga Seed

Kelas aktual \ Kelas prediksi	healthy	leaf curl	leaf spot	whitefly	yellowish
healthy	27	0	3	0	0
leaf curl	0	21	0	0	6
leaf spot	0	0	24	3	3
whitefly	0	1	6	23	0
yellowish	0	0	3	0	18

Gambar 4. Confusion matrix EfficientNetB0 akumulasi tiga seed

Confusion matrix pada Gambar 4 merupakan akumulasi tiga *seed* pada 46 citra *test* yang sama, sehingga 138 prediksi bukan 138 sampel independen. EfficientNetB0 mengenali 27 dari 30 prediksi *healthy*, 21 dari 27 *leaf curl*, 24 dari 30 *leaf spot*, 23 dari 30 *whitefly*, dan 18 dari 21 *yellowish*. Enam kesalahan *leaf curl* seluruhnya menuju *yellowish*; enam kesalahan *whitefly* menuju *leaf spot*. Pasangan tersebut merupakan sumber kesalahan utama.

Tabel 6. Metrik per kelas EfficientNetB0

Kelas	Prec. (%)	Recall (%)	F1 (%)
Healthy	100,00	90,00	94,74
Leaf curl	95,83	77,78	85,78
Leaf spot	66,67	80,00	72,73
Whitefly	88,43	76,67	82,07
Yellowish	66,67	85,71	75,00

Kelas *healthy* menghasilkan F1-score tertinggi 94,74% dan precision 100%, yang berarti tidak ada citra kelas lain masuk ke prediksi *healthy*. *Leaf spot* mempunyai F1-score terendah 72,73% karena precision 66,67%; masalahnya bukan terutama kehilangan citra *leaf spot*, melainkan masuknya prediksi dari *whitefly*, *yellowish*, dan *healthy*. *Whitefly* memiliki recall terendah 76,67%. Pola ini sejalan dengan kemiripan bercak, objek kecil, dan perubahan warna pada citra.

4.4. Efisiensi dan Penentuan Model

Tabel 7. Perbandingan efisiensi

Model	Parameter	Latih/run	Inferensi
ResNet50	23.597.957	21 m 15 d	124,88 ms
VGG16	14.717.253	51 m 16 d	191,72 ms
EfficientNetB0	4.055.976	9 m 37 d	175,23 ms

EfficientNetB0 menggunakan 4.055.976 parameter, sekitar 82,81% lebih sedikit daripada ResNet50 dan 72,44% lebih sedikit daripada VGG16. Ukuran berkas modelnya sekitar 33,80 MB dan waktu *training* rata-rata 9 menit 37 detik. ResNet50 mencatat *inference* tercepat, 124,88 ms per citra, sedangkan EfficientNetB0 175,23 ms dan VGG16 191,72 ms. Waktu tersebut spesifik terhadap CPU dan lingkungan pengujian. Berdasarkan *macro F1-score*, kestabilan, *minimum class recall* 75,19%, jumlah parameter, dan waktu *training*, EfficientNetB0 ditetapkan sebagai model terbaik.

4.5. Pembahasan dan Ancaman Validitas

Keunggulan EfficientNetB0 pada data ini konsisten dengan tujuan *compound scaling*, yaitu menyeimbangkan kedalaman, lebar, dan resolusi [4]. ResNet50 tetap kompetitif, tetapi mempunyai parameter paling besar dan dua dari tiga run-nya memilih *feature extraction*. VGG16 menjadi baseline yang jelas, tetapi menunjukkan rata-rata terendah, variasi tertinggi, dan waktu *training* terlama. Hasil tersebut tidak berarti EfficientNetB0 selalu unggul; studi tomat menemukan VGG16 lebih tinggi daripada ResNet50 [8], sedangkan studi tebu menempatkan EfficientNetB0 di atas ResNet50 [9]. Arsitektur terbaik bergantung pada objek, kelas, mutu data, dan protokol evaluasi.

Nilai penelitian ini lebih rendah daripada akurasi 97–100% yang dilaporkan pada beberapa studi cabai [5]–[7]. Perbandingan langsung tidak tepat karena studi tersebut berbeda dalam jumlah kelas, *augmentasi*, rasio split, dan penggunaan *validation* atau *test*. Penelitian ini menghapus duplikat, mengunci *test*, menerapkan *split* sama, serta melaporkan rata-rata

tiga *seed*. Kontrol tersebut membuat nilai lebih konservatif tetapi memperkuat keterlacakan perbandingan antarmodel.

Ancaman validitas utama adalah *test* hanya 46 citra; satu kesalahan mengubah accuracy sekitar 2,17 poin. Tiga *seed* mengukur variasi *training*, tetapi tidak menggantikan beberapa pembagian data atau *cross-validation*. Semua citra berasal dari satu sumber tanpa metadata tanaman, varietas, kamera, atau lokasi, sehingga generalisasi lapangan belum dapat disimpulkan. Label mengikuti *folder dataset* dan merupakan kondisi visual, bukan diagnosis laboratoris. Waktu *training* serta *inference* juga tidak dapat digeneralisasi ke GPU atau perangkat bergerak.

4.6. Implikasi Hasil

Secara metodologis, hasil memperlihatkan bahwa audit *dataset* dapat mengubah dasar evaluasi secara material. Penghapusan kelompok konflik label dan pencegahan *data leakage* menurunkan risiko metrik yang terlalu optimistis. Pendekatan tersebut penting pada klasifikasi penyakit tanaman karena variasi latar, kualitas citra, dan kemiripan gejala dapat lebih dominan daripada perbedaan arsitektur [15]. Dengan protokol yang sama, selisih performa lebih layak dikaitkan dengan karakter model daripada dengan komposisi subset.

Secara praktis, EfficientNetB0 dapat dijadikan kandidat awal untuk alat bantu penyaringan citra daun cabai. Namun, kelas *leaf spot* dan *whitefly* membutuhkan perhatian karena salah klasifikasi di antara keduanya paling sering terjadi. Sistem lapangan sebaiknya menampilkan probabilitas, peringatan saat keyakinan rendah, dan anjuran pemeriksaan lanjutan. Pendekatan pengenalan citra pada gangguan cabai telah menunjukkan potensi untuk deteksi awal [16], tetapi keluaran model tidak boleh diperlakukan sebagai diagnosis etiologis.

Jumlah parameter yang kecil memberi EfficientNetB0 peluang untuk optimasi lebih lanjut. Studi lintas-domain juga menunjukkan keluarga EfficientNet dapat mempertahankan kinerja dengan pemanfaatan representasi terlatih [17]. Sebelum diterapkan pada aplikasi bergerak seperti arah pengembangan penelitian cabai sebelumnya [18], model perlu divalidasi pada citra kebun independen, diuji terhadap perubahan pencahayaan dan latar, lalu dioptimasi melalui *quantization* atau *pruning* tanpa mengorbankan *recall* kelas yang berisiko.

5. KESIMPULAN DAN SARAN

Penelitian berhasil merancang dan menjalankan sembilan model CNN berbasis *transfer learning* untuk lima kondisi visual daun cabai dengan audit data, split identik, *augmentasi* khusus data *train*, *feature extraction*, *fine-tuning*, *holdout validation*, dan tiga *seed*. Audit mengurangi 500 citra menjadi 456 citra bersih. EfficientNetB0 menjadi model terbaik dengan rata-rata accuracy 81,88%, *macro precision* 83,52%, *macro recall* 82,03%, *macro F1-score* 82,06%,

simpangan baku *macro F1-score* 1,34%, serta 4.055.976 parameter. Kesalahan utama ialah *whitefly-leaf spot* dan *leaf curl-yellowish*. Kesimpulan berlaku pada *dataset* dan lingkungan penelitian ini.

Penelitian lanjutan perlu menambah citra lapangan dari varietas, lokasi, pencahayaan, dan perangkat berbeda; mencatat identitas tanaman agar split dapat dilakukan pada tingkat tanaman; melibatkan ahli untuk memverifikasi label; serta memakai *test* eksternal dan *repeated stratified k-fold* atau beberapa *holdout*. Optimasi *deployment* melalui *quantization* atau *pruning* sebaiknya dilakukan setelah validasi eksternal. Hasil klasifikasi harus tetap diposisikan sebagai penyaringan awal, bukan pengganti diagnosis ahli.

DAFTAR PUSTAKA

- [1] M. Iman, H. R. Arabnia, dan K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies*, vol. 11, no. 2, Art. no. 40, 2023, doi: 10.3390/technologies11020040.
- [2] K. He, X. Zhang, S. Ren, dan J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [3] K. Simonyan dan A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.
- [4] M. Tan dan Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th International Conference on Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019.
- [5] R. A. Setyadi, S. Rahman, D. Manurung, M. Hasanah, dan A. Indrawati, "Implementasi Transfer Learning untuk Klasifikasi Penyakit pada Daun Cabai Menggunakan CNN," *Djtechno: Jurnal Teknologi Informasi*, vol. 5, no. 2, pp. 304–315, 2024, doi: 10.46576/djtechno.v5i2.4642.
- [6] M. G. Azhari, M. A. Husain, dan M. Lutfi, "Transfer Learning Convolutional Neural Network dengan Arsitektur EfficientNet untuk Identifikasi Penyakit Daun Cabai," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 5, pp. 3662–3666, 2023, doi: 10.36040/jati.v7i5.7791.
- [7] I. I. Khoirunnisa, A. Fadlil, dan H. Yuliansyah, "Perbandingan Kinerja MobileNetV2 dan VGG16 dalam Klasifikasi Penyakit pada Citra Daun Tanaman Cabai," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 11, no. 1, pp. 70–82, 2026, doi: 10.14421/jiska.5075.
- [8] M. I. F. Rozi, N. O. Adiwijaya, dan D. I. Swasono, "Identifikasi Kinerja Arsitektur Transfer Learning VGG16, ResNet-50, dan Inception-V3 dalam Pengklasifikasian Citra Penyakit Daun Tomat," *Jurnal Riset Rekayasa Elektro*, vol. 5, no. 2, pp. 145–154, 2023, doi: 10.30595/jrre.v5i2.18050.
- [9] F. Rofiq dan A. Noercholis, "Analisis Komparatif Arsitektur CNN untuk Klasifikasi Penyakit Daun Tebu Berbasis Transfer Learning," *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 15, no. 2, pp. 666–678, 2026, doi: 10.35889/jutisi.v15i2.3431.
- [10] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, dan M. Parmar, "A Review of Convolutional Neural Networks in Computer Vision," *Artificial Intelligence Review*, vol. 57, no. 4, Art. no. 99, 2024, doi: 10.1007/s10462-024-10721-6.
- [11] Keras, "Keras Applications," Keras API Documentation. [Online]. Available: <https://keras.io/api/applications/>. [Accessed: Aug. 15, 2026].
- [12] F. B. Laksono dan A. Fathurohman, "Deteksi Penyakit Tanaman dengan Convolution Neural Network: Kombinasi Arsitektur VGG16 dan ResNet34 untuk Klasifikasi Daun," *Jurnal Komputer dan Teknologi Informasi*, vol. 2, no. 2, pp. 72–79, 2024, doi: 10.26714/jkti.v2i2.13932.
- [13] D. D. Prakoso, "Chili Plant Disease," Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/dhenyd/chili-plant-disease>. [Accessed: Aug. 15, 2026].
- [14] W. Hua, C. Li, dan X. Wang, "Review of Convolutional Neural Network Models and Image Classification," *Academic Journal of Science and Technology*, vol. 10, no. 3, pp. 178–184, 2024, doi: 10.54097/644jqv20.
- [15] D. A. Pangestu, O. Q. Aziz, dan C. Crysdiyan, "Klasifikasi Penyakit pada Tanaman Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 2, pp. 235–248, 2025, doi: 10.14421/jiska.2025.10.2.235-248.
- [16] S. F. Nazila, Y. Arman, D. Wahyuni, N. Nurhasanah, dan Y. S. Putra, "Deteksi Dini Serangan Hama Penyakit pada Cabai Rawit Menggunakan Metode Image Recognition," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 2, pp. 232–241, 2023, doi: 10.28932/jutisi.v9i2.6342.
- [17] M. R. Rasheed, S. Coleman, B. Gardiner, P. Vance, C. McAteer, dan K. Nguyen, "An EfficientNet-Based Transfer Learning System for Defect Classification in Manufacturing," in *Proc. IEEE 22nd International Conference on Industrial Informatics (INDIN)*, pp. 1–7, 2024, doi: 10.1109/INDIN58382.2024.10774443.
- [18] S. Winiarti dan I. I. Khoirunnisa, "Mobile Application Development for Chili Disease Detection with Convolutional Neural Network," *International Journal of Informatics and Computation*, vol. 6, no. 2, pp. 97–112, 2024, doi: 10.35842/ijicom.v6i2.93.