

KLASIFIKASI PENYAKIT DAUN TOMAT BERBASIS TRANSFER LEARNING EFFICIENTNETB0

Kadek Ganeis Jayarani Sapanca¹, Putu Hendra Suputra², Ni Wayan Marti³

^{1,2,3}Program Studi Ilmu Komputer, Universitas Pendidikan Ganesha

Jl. Udayana No.11, Singaraja, Kabupaten Buleleng, Bali 81116¹

¹ganeis@student.undiksha.ac.id, ²hendra.suputra@undiksha.ac.id, ³wayan.marti@undiksha.ac.id

Abstract

Tomato leaf diseases such as *Early Blight* and *Late Blight* pose serious threats to agricultural productivity, requiring early detection to prevent crop losses. This study develops a tomato leaf disease classification system using *EfficientNetB0* architecture with transfer learning method to identify three leaf condition categories: *Early Blight*, *Healthy*, and *Late Blight*. The dataset consists of 1,425 images divided using an 80:20 ratio and augmented to 4,557 training images through data augmentation techniques. Four experimental scenarios were conducted with variations in batch size (16 and 32) and number of epochs (25 and 50) to identify optimal hyperparameter configurations. Evaluation results show that the model with batch size 16 and 50 epochs achieves the best performance with test accuracy of 92.55%, precision of 92.82%, recall of 92.55%, and F1-score of 92.39%. Per-class performance reaches *Early Blight* (F1-score 92.46%), *Healthy* (F1-score 96.26% with perfect recall 100%), and *Late Blight* (F1-score 88.30%). Comparative analysis proves that batch size 16 consistently outperforms batch size 32 with an accuracy difference of 4.21%, and increasing epochs from 25 to 50 provides a 2.10% improvement, particularly in *Early Blight* detection. The developed model meets practical application standards with high accuracy and efficient training time, potentially assisting farmers in early disease detection to improve tomato crop productivity.

Keywords : *efficientnetb0*, tomato leaf disease classification, transfer learning, plant disease detection

Abstrak

Penyakit daun tomat seperti *Early Blight* dan *Late Blight* menjadi ancaman serius terhadap produktivitas pertanian yang memerlukan deteksi dini untuk mencegah kerugian hasil panen. Penelitian ini mengembangkan sistem klasifikasi penyakit daun tomat menggunakan arsitektur *EfficientNetB0* dengan metode transfer learning untuk mengidentifikasi tiga kategori kondisi daun *Early Blight*, *Healthy*, dan *Late Blight*. Dataset terdiri dari 1.425 gambar yang dibagi dengan rasio 80:20 dan ditingkatkan menjadi 4.557 gambar training melalui teknik augmentasi data. Empat skenario eksperimen dilakukan dengan variasi batch size (16 dan 32) serta jumlah epoch (25 dan 50) untuk mengidentifikasi konfigurasi *hyperparameter* optimal. Hasil evaluasi menunjukkan bahwa model dengan *batch size* 16 dan *epoch* 50 menghasilkan performa terbaik dengan *test accuracy* 92.55%, *precision* 92.82%, *recall* 92.55%, dan *F1-score* 92.39%. Performa per-kelas mencapai *Early Blight* (*F1-score* 92.46%), *Healthy* (*F1-score* 96.26% dengan *recall* sempurna 100%), dan *Late Blight* (*F1-score* 88.30%). Analisis komparatif membuktikan batch size 16 konsisten unggul dibanding *batch size* 32 dengan selisih akurasi 4.21%, dan peningkatan *epoch* dari 25 ke 50 memberikan kenaikan 2.10% terutama pada deteksi *Early Blight*. Model yang dikembangkan memenuhi standar aplikasi praktis dengan akurasi tinggi dan waktu training efisien, berpotensi membantu petani melakukan deteksi dini penyakit untuk meningkatkan produktivitas tanaman tomat.

Kata kunci : *efficientnetb0*, klasifikasi penyakit daun tomat, transfer learning, deteksi penyakit tanaman

1. PENDAHULUAN

Tomat (*Solanum lycopersium*) merupakan salah satu komoditas hortikultura strategis di Indonesia dengan produksi mencapai 1,09 juta

ton per tahun dan memiliki nilai ekonomi tinggi baik untuk konsumsi domestic maupun ekspor [1]. Sebagai tanaman semusim yang banyak dibudidayakan oleh petani Indonesia, tomat

memiliki peran penting dalam ketahanan pangan nasional dan menjadi sumber pendapatan utama bagi petani sayuran. Namun, produktivitas tanaman tomat sering mengalami penurunan signifikan akibat serangan berbagai penyakit daun, terutama *Early Blight* yang disebabkan oleh jamur *Alternaria solani* dan *Late Blight* yang disebabkan oleh *Phytophthora infestans*. Kedua penyakit ini dapat menyebar dengan cepat terutama pada kondisi kelembaban tinggi dan suhu optimal, ini menyebabkan penurunan hasil panen hingga 40-60% bahkan dapat mengakibatkan gagal panen total jika tidak ditangani secara cepat [2]. Penyakit pada tanaman merupakan suatu kondisi gangguan yang terjadi akibat adanya serangan patogen atau faktor penyebab tertentu yang berlangsung secara terus-menerus dalam periode waktu tertentu. Proses tersebut dapat mengganggu fungsi fisiologis tanaman sehingga tidak mampu berkembang secara optimal [3]. Deteksi dini dan akurat terhadap penyakit daun tomat sangat krusial untuk mencegah penyebaran penyakit dan meminimalkan kerugian ekonomi petani. Metode deteksi konvensional masih mengandalkan pemeriksaan visual oleh ahli, namun terbatas oleh ketersediaan tenaga, subjektivitas, cakupan area yang sempit, dan memakan waktu lama, terutama pada lahan berskala besar [4]. Selain itu, identifikasi visual pada tahap awal infeksi sering kali sulit dilakukan karena gejala yang belum terlihat jelas, sehingga penyakit baru terdeteksi setelah mencapai Tingkat keparahan yang tinggi. Perkembangan kecerdasan buatan *AI* telah banyak dimanfaatkan dalam berbagai permasalahan klasifikasi [5], telah membuka peluang besar untuk mengotomatisasi proses deteksi penyakit tanaman secara cepat, akurat, dan konsisten. Dalam konteks machine learning, pemilihan arsitektur serta pengaturan parameter model sangat berpengaruh terhadap performa klasifikasi yang dihasilkan. Oleh karena itu, diperlukan pendekatan yang tidak hanya akurat tetapi juga efisien dalam proses pelatihannya. Metode yang lebih canggih seperti deep learning sangat dibutuhkan untuk mengidentifikasi penyakit daun tomat. Arsitektur *deep learning* sendiri merupakan salah satu bentuk *machine learning* lanjutan yang dapat meningkatkan

performa komputasi dan tingkat akurasi melalui penambahan jumlah lapisan [6]. *EfficientNet* merupakan keluarga arsitektur CNN yang dikembangkan dengan pendekatan *compound scaling* yang secara seimbang menskalakan kedalaman, lebar, dan resolusi jaringan *neural network*. *EfficientNetB0* sebagai varian terkecil dari keluarga *EfficientNet*, menawarkan keseimbangan optimal antara akurasi tinggi dan efisiensi komputasi, menjadikannya sangat cocok untuk aplikasi real-time pada perangkat dengan sumber daya terbatas seperti smartphone atau edge devices [7]. Beberapa penelitian di Indonesia telah mengeksplorasi penggunaan berbagai arsitektur CNN untuk deteksi penyakit tomat, termasuk *VGG16*, *ResNet50*, dan *MobileNet* dengan tingkat akurasi yang bervariasi antara 85% hingga 95% [8]. Penelitian ini menawarkan pendekatan berbeda dibandingkan studi terdahulu dengan memadukan arsitektur *EfficientNetB0*, label smoothing, partial unfreezing, dan teknik augmentasi data secara komprehensif agar hasil klasifikasi lebih akurat dan generalizable. Penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit daun tomat yang tidak hanya akurat tetapi juga efisien dan robust menggunakan pendekatan transfer learning dengan arsitektur *EfficientNetB0*. Kontribusi utama penelitian ini meliputi implementasi *transfer learning* dengan *EfficientNetB0* untuk klasifikasi tiga kelas penyakit daun tomat (*Early Blight*, *Late Blight*, dan *Healthy*), pengembangan solusi efektif untuk mengatasi masalah model collapse melalui kombinasi teknik label smoothing (dengan *smoothing factor* 0,1) dan strategi partial unfreezing yang memungkinkan model untuk mempelajari representasi fitur yang lebih baik dan seimbang untuk semua kelas, perancangan dan implementasi strategi data *augmentation* yang komprehensif meliputi *rotasi*, *shifting*, *brightness adjustment*, *zoom*, dan *flipping* untuk meningkatkan kemampuan generalisasi model pada dataset terbatas, dan evaluasi performa model secara menyeluruh dengan metrik.

2. TINJAUAN PUSTAKA DAN TEORI

2.1. Penyakit Daun Tomat

Tanaman tomat (*Solanum lycopersicum*) merupakan salah satu komoditas hortikultura

penting yang rentan terhadap berbagai serangan penyakit daun yang dapat menurunkan produktivitas secara signifikan. Penyakit *Early Blight* yang disebabkan oleh jamur *Alternaria solani* ditandai dengan munculnya bercak konsentris berwarna coklat kehitaman pada daun, menyerupai pola target dengan diameter 1-2 cm, dan dapat menyebabkan kehilangan hasil panen [9]. Penyakit *Late Blight* yang disebabkan oleh *Phytophthora infestans* merupakan penyakit paling destruktif dengan gejala bercak basah berwarna hijau gelap hingga coklat kehitaman, berkembang sangat cepat pada kondisi kelembaban tinggi, dan dapat menyebabkan gagal panen total [10]. Sementara itu, daun tomat yang sehat memiliki karakteristik berwarna hijau segar dengan tekstur permukaan halus dan merata, tidak memiliki bercak atau perubahan warna abnormal.

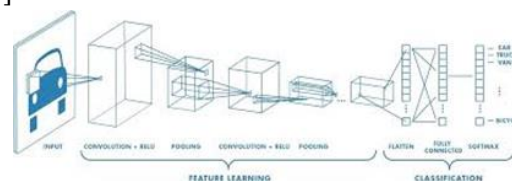
2.2. Deep Learning

Deep learning merupakan sub-bidang *machine learning* yang menggunakan arsitektur *artificial neural network* dengan banyak layer untuk mempelajari representasi data secara hierarkis dan otomatis [11]. Pendekatan ini berbeda dari *machine learning* konvensional yang memerlukan ekstraksi fitur manual, dimana *deep learning* mampu secara otomatis mempelajari representasi fitur dari data mentah melalui proses pembelajaran berlapis. Setiap layer dalam arsitektur *deep learning* mempelajari representasi yang semakin abstrak, dimulai dari fitur tingkat rendah seperti tepi dan tekstur pada layer awal, hingga fitur tingkat tinggi seperti bentuk objek kompleks pada layer yang lebih dalam.

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur *deep learning* yang secara khusus dirancang untuk pemrosesan data yang memiliki struktur grid seperti citra. CNN dibangun berdasarkan tiga komponen arsitektural utama yang bekerja secara hierarkis. Pertama, *Convolutional Layer* yang menggunakan operasi konvolusi untuk mengekstraksi fitur lokal dari citra melalui filter atau kernel yang dapat dipelajari, dimana setiap filter mendeteksi pola spesifik seperti tepi, sudut, atau tekstur tertentu.

Kedua, *Poolong* layer yang berfungsi untuk reduksi dimensi spasial sambil mempertahankan informasi penting, umumnya menggunakan *max pooling* atau *average pooling*, sehingga mengurangi parameter dan komputasi serta memberikan sifat *translation invariance*. Ketiga, *Fully Connected Layer* dibagian akhir arsitektur yang mengintegrasikan fitur-fitur yang telah diekstraksi untuk melakukan klasifikasi akhir [12].



Gambar 1. Arsitektur CNN

2.4. Transfer Learning

Transfer learning Adalah paradigma pembelajaran mesin yang memanfaatkan pengetahuan yang telah dipelajari dari satu domain sumber untuk meningkatkan pembelajaran pada domain target yang berbeda namun terkait [13]. Proses *transfer learning* melibatkan dua tahap utama, yang pertama, model di-pretrain pada dataset sumber yang besar untuk mempelajari representasi fitur umum, kedua, model tersebut di-*fine-tune* pada dataset target yang lebih kecil dan spesifik dengan menyesuaikan parameter model khususnya pada layer-layer teratas [14].

2.5. Arsitektur EfficientNetB0

EfficientNetB0 adalah varian dasar *EfficientNet* yang dirancang dengan *Neural Architecture Search (NAS)* dan *compound scaling*, sehingga menskalakan *depth*, *width*, dan *resolution* secara seimbang. Arsitektur ini memiliki 18 layer dalam 9 *stage*, menggabungkan *standard convolution* di awal, *MBCConv* sebagai blok utama, serta *global average pooling* dan *fully connected layer* untuk klasifikasi. *MBCConv* menggunakan *inverted residual*, *depthwise convolution*, *squeeze-and-excitation*, dan aktivasi *Swish*, serta *shortcut connection* dan *stochastic depth* sebagai regularisasi. Model ini efisien dengan 5,3 juta parameter dan 0,39 miliar *FLOPS*, cocok untuk perangkat dengan sumber daya terbatas. Gambar 2 merupakan Gambaran arsitektur *EfficientNetB0*.

menggunakan teknik partial unfreezing, yakni membekukan 50% layer awal serta melatih ulang 50% layer akhir agar model dapat beradaptasi terhadap pola khusus dataset penyakit daun tomat. Pada bagian atas model dasar, diterapkan kepala klasifikasi custom berupa *Global Average Pooling*, *Batch Normalization*, *Dropout (0,3)*, *Dense* dengan 256 unit aktivasi *ReLU*, *Batch Normalization* kedua, *Dropout (0,2)*, dan *Dense output* sebanyak 3 unit *softmax*. Model dikompilasi menggunakan *optimizer Adam* *learning rate 0,0001*, *loss function Categorical Crossentropy* dengan *label smoothing (0,1)*, serta metrik akurasi.

Pemilihan struktur dan *hyperparameter* tersebut dilakukan agar model memperoleh keseimbangan antara akurasi, efisiensi komputasi, serta kemampuan generalisasi. *Partial unfreezing* diterapkan karena lapisan awal *EfficientNetB0* telah mempelajari fitur generik dari dataset besar, sedangkan lapisan akhir perlu penyesuaian pada data domain khusus. Penggunaan *batch normalization*, *dropout*, serta teknik augmentasi data, selain untuk mempercepat konvergensi, juga efektif dalam meminimalkan risiko *overfitting*. *Learning rate* yang rendah dan *label smoothing* terbukti dapat menjaga kestabilan proses pelatihan dan meningkatkan generalisasi model [22], pendekatan terintegrasi ini telah direkomendasikan sebagai best practice pada penelitian klasifikasi penyakit tanaman berbasis citra.

3.4. Pelatihan Model

Proses pelatihan menguji empat skenario hyperparameter: batch size (16, 32) dan epoch (25, 50), dengan kombinasi batch 16/32 dan epoch 25/50. *Optimizer Adam* (*learning rate 0.0001*) dan *loss function categorical crossentropy* dengan *label smoothing 0,1* digunakan. *Class weights* dihitung otomatis untuk menangani ketidakseimbangan kelas. *Callback ModelCheckpoint* menyimpan model terbaik, *ReduceLROnPlateau* mengurangi *learning rate* saat *validation loss* stagnan selama 5 epoch, dan *EarlyStopping* menghentikan *training* jika tidak ada peningkatan *validation loss* dalam 10-15 epoch. Seluruh percobaan menggunakan data *training* (4.557 gambar),

validation (1.141 gambar), ukuran citra 224×224 piksel, dan random seed 42, serta dijalankan pada *GPU Google Colab*.

TABEL I. SKENARIO *HYPERPARAMETER* PELATIHAN

Skenario	Batch Size	Epoch
1	16	25
2		50
3	32	25
4		50

3.5. Evaluasi Model

Evaluasi performa model pada berbagai skenario menggunakan confusion matrix untuk menganalisis tingkat kesesuaian antara label prediksi dengan label sebenarnya pada data uji. Representasi matriks ini memungkinkan pengamatan terhadap distribusi hasil klasifikasi yang akurat dan tidak akurat, termasuk kategori *TP (True Positive)*, *TN (True Negative)*, *FP (False Positive)*, dan *FN (False Negative)* pada masing-masing kelas. Validasi dilakukan terhadap data yang tidak terlibat dalam proses training, memberikan gambaran objektif tentang kemampuan model mengenali pola penyakit daun tomat.

4. HASIL DAN PEMBAHASAN

4.1. Dataset

Dataset yang digunakan berjumlah 1.425 gambar daun tomat original yang terbagi dalam tiga kategori yaitu *Early Blight* (480 gambar), *Healthy* (482 gambar), dan *Late Blight* (463 gambar).



Gambar 5. Sample dataset penyakit daun tomat.

4.2. Preprocessing Data

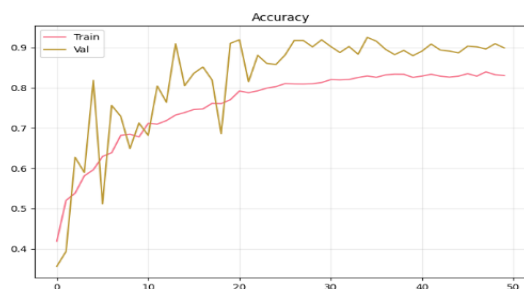
Tahap preprocessing akan dilakukan dalam tiga tahapan. Tahap pertama akan dilakukan split dataset menjadi 80% *training* dengan total 1.140 citra, dan 20% *testing* dengan total 285 citra. Setiap kelas berkontribusi proporsional dengan *Early Blight* (384 *training*, 96 *testing*), *Healthy* (386 *training*, 96 *testing*), dan *Late Blight* (370

training, 93 testing). Tahap kedua yaitu mengubah dimensi seluruh gambar menjadi 224x224 piksel. Standarisasi ukuran ini disesuaikan dengan spesifikasi input *EfficientNetB0* yang optimal. Tahap ketiga menerapkan augmentasi pada data *training* menggunakan Rotasi acak, *horizontal flip*, *vertical flip*, *transformasi shear*, *zoom*, dan *brightness*. Implementasi augmentasi meningkatkan variasi data training dari 1.140 menjadi 4.557 gambar. Normalisasi pixel dilakukan dengan rescaling faktor 1/255 untuk mengonversi rentang nilai 0-255 menjadi 0-1. Proses ini mempercepat konvergensi training dan meningkatkan stabilitas gradient descent.

4.3. Pelatihan Model EfficientNetB0

Pelatihan model dilakukan dengan menguji empat variasi *hyperparameter* untuk menemukan konfigurasi yang menghasilkan performa optimal. Hasil eksperimen menunjukkan bahwa konfigurasi terbaik dicapai pada Skenario 2 dengan *batch size* 16 dan *epoch* 50, yang menghasilkan *validation accuracy* tertinggi 92.55% dibandingkan ketiga skenario lainnya.

Proses training pada Skenario 2 memperlihatkan pola pembelajaran yang stabil dan konsisten sepanjang 50 *epoch*. Pada fase awal hingga *epoch* ke-15, model mengalami peningkatan akurasi yang cukup signifikan dari 72% menjadi 88%. Fase pembelajaran berlanjut secara bertahap pada *epoch* ke-16 hingga *epoch* ke-35, dimana *validation accuracy* terus meningkat hingga mencapai puncaknya di 92.55%. Setelah *epoch* ke-35, kurva akurasi menunjukkan stabilitas dengan fluktuasi minimal, mengindikasikan model telah mencapai konvergensi optimal.

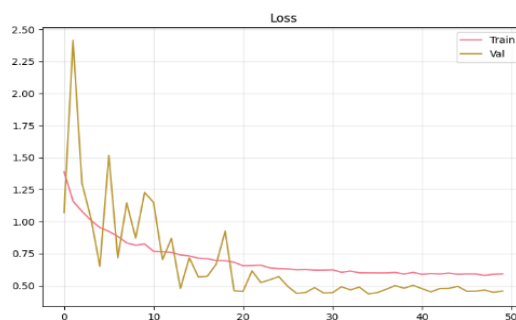


Gambar 6. Grafik Akurasi Skenario 2 (*Batch* 16, *Epoch* 50)

Visualisasi kurva *accuracy* pada Gambar 6 memperlihatkan bahwa training accuracy dan validation accuracy bergerak sejajar tanpa gap yang signifikan.

Training accuracy mencapai 94.12% sementara *validation accuracy* tercatat 92.55%, menghasilkan selisih hanya 1.57%. Gap yang kecil ini menunjukkan tidak adanya *overfitting*, dimana model mampu menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

Kurva loss pada Gambar 7 menampilkan tren penurunan yang konsisten dari awal hingga akhir training. Loss pelatihan menurun dari 1.08 pada *epoch* pertama menjadi 0.18 pada *epoch* terakhir. *Loss validasi* mengikuti pola serupa dengan nilai akhir 0.49, menunjukkan pembelajaran yang efektif tanpa indikasi memorisasi data training.



Gambar 7. Grafik Loss Skenario 2 (*Batch* 16, *Epoch* 50)

Stabilitas kedua kurva *loss* setelah *epoch* ke-30 mengonfirmasi bahwa model telah mencapai titik konvergensi yang baik.

TABEL II. PERBANDINGAN HASIL PENGUJIAN *EFFICIENTNETB0*

Skenario	Batch Size	Epoch	Accuracy (%)	Precision (%)	Recall (%)
1	16	25	90.62	90.79	90.62
2		50	92.55	92.82	92.55
3	32	25	86.24	86.55	86.24
4		50	48.73	48.91	48.73

Perbandingan hasil menunjukkan yang ditampilkan pada Tabel 2 menegaskan bahwa model *EfficientNetB0* mampu mengklasifikasikan penyakit daun tomat dengan tingkat akurasi dalam rentang 86.24% hingga 92.55% pada tiga skenario yang berhasil.

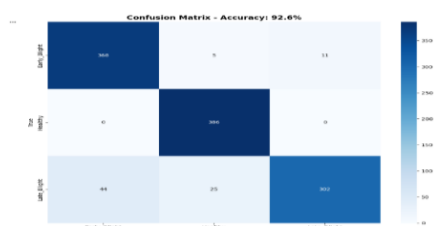
Performa tertinggi dicapai Skenario 2 yang mencatatkan nilai konsisten di seluruh metrik evaluasi dengan *Accuracy* 92.55%, *Precision* 92.82%, *Recall* 92.55%, dan *F1-Score* 92.39%. Sebaliknya, performa terendah pada skenario yang berhasil tercatat pada Skenario 3 dengan *accuracy* 86.24% dan *F1-Score* 85.89%.

Temuan ini mengindikasikan bahwa penggunaan batch size yang lebih kecil 16 memberikan hasil yang lebih baik dibandingkan batch size besar 32, bahkan pada durasi epoch yang identik. Skenario 4 mengalami kegagalan dengan *accuracy* hanya 48.73%, membuktikan bahwa kombinasi batch size besar dengan epoch tinggi tidak sesuai untuk karakteristik dataset ini.

4.4. Evaluasi Model

Evaluasi dilakukan pada test set yang terdiri dari 1.141 gambar menggunakan model dari keempat skenario. Skenario 2 (*batch* 16, *epoch* 50) terpilih sebagai konfigurasi terbaik dengan test *accuracy* 92.55% dan *F1-score weighted average* 92.39%. Kelas Healthy menunjukkan performa superior dengan *recall* sempurna 100%, mengindikasikan model tidak menghasilkan false negative pada daun sehat. *Precision* 92.79% menandakan tingkat kepercayaan tinggi dengan hanya 7.21% *false positive*.

Kelas *Early Blight* mencatat *recall* 95.83%, berarti model hanya melewatkan 4.17% kasus (16 dari 384 sampel). *Precision* 89.32% menunjukkan 10.68% prediksi adalah *false positive*, kemungkinan terjadi kebingungan dengan *Late Blight* pada stadium awal. Kelas *Late Blight* memiliki *precision* tertinggi 96.49%, menandakan ketika model memprediksi *Late Blight*, akurasi mencapai hampir 97%. *Recall* 81.40% mengindikasikan 18.60% kasus tidak terdeteksi, terutama pada gejala awal yang mirip dengan *Early Blight*.



Gambar 8. *Confusion Matrix* Skenario 2 (*Batch* 16, *Epoch* 50)

Confusion matrix pada Gambar 8 memperlihatkan konsentrasi nilai pada diagonal utama, mengonfirmasi akurasi tinggi. Kelas *Healthy* teridentifikasi sempurna (386/386), diikuti *Early Blight* (368/384) dan *Late Blight* (302/371). Misklasifikasi terbanyak terjadi antara *Early Blight* dan *Late Blight* dengan 14 kasus salah prediksi ke *Late Blight* dan 55 kasus sebaliknya, disebabkan kemiripan visual gejala kedua penyakit. Distribusi prediksi model menunjukkan *Early Blight* (412 prediksi, 36.11%), *Healthy* (416 prediksi, 36.46%), dan *Late Blight* (313 prediksi, 27.43%).

Keseimbangan ini mengonfirmasi tidak ada bias signifikan terhadap kelas tertentu, berbeda dengan Skenario 4 yang bias 81% ke *Healthy*. Perbandingan antar skenario mengungkapkan bahwa peningkatan epoch pada batch 16 memberikan *improvement* 2.10% *accuracy* dengan peningkatan *Early Blight recall* dari 85.16% menjadi 95.83%. Penggunaan batch 32 menurunkan *accuracy* 4.21% meskipun 62% lebih cepat, menunjukkan trade-off speed versus *accuracy* yang tidak favorable.

Kegagalan Skenario 4 dengan *accuracy* 48.73% memvalidasi bahwa batch size besar tidak kompatibel dengan training panjang pada dataset ini. Model optimal pada skenario 2 mencapai keseimbangan antara *accuracy* tinggi (92.55%), waktu training efisien (1 jam 20 menit), dan generalisasi robust pada data unseen. Hasil ini memenuhi requirement untuk aplikasi praktis deteksi penyakit daun tomat dengan *reliability* yang memadai.

4.5 Pengaruh Batch Size dan Jumlah Epoch Terhadap Performa Model

Variasi batch size dan jumlah epoch menunjukkan pengaruh signifikan terhadap performa model EfficientNetB0. Pengaruh jumlah epoch terlihat jelas pada batch size 16, dimana peningkatan dari 25 epoch menjadi 50 epoch menghasilkan *improvement* akurasi sebesar 2.10%, dari 90.45% menjadi 92.55%. Kenaikan paling signifikan terjadi pada kemampuan deteksi *Early Blight*, dengan *recall* meningkat drastis dari 85.16% menjadi 95.83%, menunjukkan penambahan 10.67 poin persentase. Model mencapai konvergensi

optimal pada epoch ke-35 dan mempertahankan stabilitas hingga epoch ke-50, membuktikan epoch tambahan tidak menyebabkan overfitting namun justru memperkuat generalisasi.

Sebaliknya, pembesaran batch size dari 16 menjadi 32 menghasilkan dampak negatif terhadap akurasi. Pada durasi epoch yang sama (25 epoch), batch size 32 (Skenario 3) mencatat akurasi 86.24%, turun 4.21% dibanding batch size 16 yang mencapai 90.45%. Penurunan performa ini disebabkan berkurangnya frekuensi gradient update dari 285 steps/epoch pada batch 16 menjadi hanya 143 steps/epoch pada batch 32, menghasilkan pembelajaran yang kurang detail.

Late Blight mengalami penurunan paling tajam dengan F1-score turun 7.48% (dari 86.37% menjadi 78.89%), mengindikasikan batch besar kesulitan menangkap variasi halus pada kelas dengan gejala yang kompleks.

Interaksi antara batch size besar dan epoch tinggi pada Skenario 4 menghasilkan kegagalan total dengan akurasi hanya 48.73%. Kombinasi batch 32 dengan 50 epoch menyebabkan overfitting ekstrem (gap 20.04%) dan bias prediksi dimana 81% output mengarah ke kelas Healthy. Model mengalami early stopping pada epoch 11, menunjukkan ketidakmampuan konvergen dengan konfigurasi ini. Analisis mengungkapkan bahwa learning rate 0.0001 yang optimal untuk batch 16 menjadi terlalu tinggi untuk batch 32, menyebabkan gradient descent overshoot dan tidak mencapai minimum optimal.

Hasil eksperimen membuktikan bahwa EfficientNetB0 lebih efektif melakukan pembelajaran ketika gradient update dilakukan secara frequent melalui batch kecil, dibandingkan akumulasi gradient pada batch besar. Batch size 16 dengan 50 epoch menjadi konfigurasi optimal yang menghasilkan keseimbangan terbaik antara akurasi (92.55%), stabilitas training, dan waktu komputasi yang reasonable.

4.6 Analisis dan Perbandingan dengan Penelitian Terdahulu

Analisis kritis terhadap hasil penelitian ini menunjukkan bahwa penerapan EfficientNetB0 dengan strategi transfer learning, partial

unfreezing, dan augmentasi data beragam menghasilkan akurasi 92,55%, lebih tinggi dibandingkan dengan penelitian yang mengaplikasikan CNN standar dengan akurasi 90,10% [23] dan yang menggunakan Inception-V3 dengan akurasi 91,70% [24] untuk klasifikasi penyakit daun tomat.

Keunggulan performa ini didukung oleh optimasi hyperparameter yang cermat serta penerapan label smoothing, sehingga model mampu mengatasi kemiripan fitur visual antara Early Blight dan Late Blight, yang sering menjadi tantangan pada penelitian sebelumnya. F1-score dan recall yang tinggi pada masing-masing kelas juga mengindikasikan generalisasi model yang lebih baik terhadap data uji.

Lebih lanjut, hasil ini konsisten dengan tren pada literatur internasional yang menunjukkan efisiensi serta keunggulan EfficientNetB0 dibandingkan arsitektur CNN generasi sebelumnya, terutama apabila dipadukan dengan teknik pelatihan lanjutan.

Secara keseluruhan, penelitian ini berhasil menunjukkan bahwa kombinasi pendekatan modern tidak hanya mampu memperbaiki performa klasifikasi penyakit daun tomat dibandingkan studi-studi nasional sebelumnya, tetapi juga dapat dijadikan rujukan pengembangan sistem deteksi penyakit tanaman berbasis citra secara lebih luas.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil mengembangkan sistem klasifikasi penyakit daun tomat menggunakan EfficientNetB0 yang mampu membedakan tiga kategori kondisi daun Early Blight, Healthy, dan Late Blight. Model optimal dengan konfigurasi batch size 16 dan epoch 50 mencapai akurasi 92.55% dengan F1-score 92.39% pada test set yang terdiri dari 1.141 gambar.

Hasil evaluasi menunjukkan performa per-kelas yang sangat baik, dimana kelas Healthy mencapai recall sempurna 100%, Early Blight mencapai F1-score 92.46%, dan Late Blight mencapai F1-score 88.30%. Analisis komparatif empat skenario eksperimen membuktikan bahwa batch size 16 konsisten unggul dibanding

batch size 32, dan penambahan epoch dari 25 ke 50 meningkatkan akurasi sebesar 2.10%.

5.2 Saran

Disarankan agar dataset diperluas dengan berbagai kondisi dan varietas tomat untuk meningkatkan generalisasi model. Pengujian lapangan dan integrasi ke aplikasi *mobile* perlu dilakukan agar sistem dapat langsung dimanfaatkan petani. Pengembangan model untuk klasifikasi multi-kelas dan pengoptimalan *hyperparameter* juga dianjurkan. Kolaborasi dengan pakar pertanian akan meningkatkan akurasi pelabelan dan validasi hasil, serta kepercayaan pengguna terhadap sistem ini.

6. UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bantuan selama proses penelitian ini berlangsung. Setiap bentuk perhatian dan bantuan yang diberikan sangat berarti dalam menyelesaikan penelitian ini.

Daftar Pustaka:

- [1] B. P. Statistik, "Statistik Tanaman Sayuran dan Buah-buahan Semusim Indonesia 2022."
- [2] D. Dan, K. Penyakit, and D. Tomat, "Jurnal Computer Science and Information Technology (CoSciTech) Detection And Classification Tomato Leaf Disease Using ResNet-50," vol. 6, no. 1, pp. 9–20, 2025.
- [3] I. Komputer, U. P. Ganesha, and K. Tabanan, "PENGARUH DATA AUGMENTASI PADA IDENTIFIKASI PENYAKIT DAUN TANAMAN OBAT MENGGUNAKAN K-NEAREST NEIGHBOR," vol. 9, no. 2, pp. 2094–2100, 2025.
- [4] A. C. Anwari, "Klasifikasi Penyakit Tanaman Tomat Melalui Citra Daun Menggunakan Metode Convolutional Neural Network," vol. 02, no. 03, pp. 639–647, 2024.
- [5] I. Adi, P. Pande, O. Andewi, and A. Ardivan, "Klasifikasi Tanaman Pinus Berbasis Forward Chaining Pada Sistem Pakar," vol. 13, no. 11, pp. 142–156, 2024.
- [6] D. Prima, H. Putri, N. Putu, N. Puspa, I. K. Purnamawan, and N. W. Marti, "Perbandingan Performansi Support Vector Machine (Svm) dan Backpropagation untuk Klasifikasi Studi," vol. 9, no. 3, pp. 492–501, 2023.
- [7] N. Putu *et al.*, "River Area Segmentation Using Sentinel-1 SAR Imagery with Deep-Learning Approach," vol. 19, no. 4, pp. 39–63, 2025.
- [8] J. Sistem, S. Komputer, A. J. Bastari, and A. Cherid, "CONVOLUTIONAL NEURAL NETWORK DAN IMPLEMENTASI MODEL H5 PADA APLIKASI DESKTOP," vol. 8, no. 2, pp. 199–207, 2023.
- [9] N. Istifadah, R. A. Putri, and S. Hartati, "The Abilities of Bacteria and Yeast Isolated from Vermicompost Water Extract to Inhibit Alternaria solani in vitro and Early Blight Disease on Tomato," vol. 4, no. 2, pp. 73–79, 2022.
- [10] and G. S. M. R. Widodo, S. H. Hidayat, "Epidemiologi dan Manajemen Penyakit Late Blight pada Tanaman Solanaceae di Indonesia," vol. 17, no. 04, pp. 1–28, 2024.
- [11] and G. H. Y. LeCun, Y. Bengio, "Deep Learning: Recent Advances and New Frontiers," *Commun. ACM*, vol. 64, no. 07, pp. 58–65, 2021.
- [12] M. Tan and Q. V. Le, "EfficientNetV2 : Smaller Models and Faster Training," *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021.
- [13] J. Rozaqi, A. Sunyoto, and R. Arief, "Implementasi Transfer Learning pada Algoritma Convolutional Neural Network Untuk Identifikasi Penyakit Daun Kentang Implementation of Transfer Learning in the Convolutional Neural Network Algorithm for Identification of Potato Leaf Disease," vol. 1, no. 1, 2021.
- [14] I. Arsitektur, A. Dan, and R. Pada, "Implementasi arsitektur alexnet dan resnet 34 pada klasifikasi citra penyakit daun kentang menggunakan transfer learning," vol. 7, no. 5, pp. 3293–3301, 2023.
- [15] R. Ronggo, B. Pratomo, and P. Palupingsih, "Analisis Perbandingan Performa Model Klasifikasi Kesehatan Daun Tomat menggunakan Arsitektur VGG, MobileNet, dan Inception V3 Analysis Tomato Leaf Health Classification Model Performance Comparison Using VGG, MobileNet, and Inception V3," vol. 10, pp. 98–110.
- [16] I. K. N. Ananda, N. Putu, N. Puspa, N. W. Marti, L. Joni, and E. Dewi, "JOURNAL OF APPLIED COMPUTER SCIENCE AND TECHNOLOGY (JACOST) Klasifikasi Multilabel Pada Gaya Belajar Siswa Sekolah Dasar Menggunakan Algoritma

- Machine Learning,” vol. 5, no. 2, pp. 144–154, 2024.
- [17] P. S. Paramita *et al.*, “MODEL DEEP LEARNING UNTUK KLASIFIKASI OBJEK PADA GAMBAR,” vol. 11, no. 3, 2024, doi: 10.25126/jtiik.938047.
- [18] P. Algoritme, N. Bayes, D. A. N. Knn, M. Riyyan, and H. Firdaus, “TERHADAP DATA PENERIMAAN BEASISWA (Studi Kasus Lembaga Beasiswa Baznas Jabar),” vol. 5, no. 1, pp. 1–10, 2022.
- [19] D. A. Pusparani *et al.*, “Identification of Little Tuna Species Using Convolutional Neural Networks (CNN) Method and ResNet-50 Architecture,” vol. 8, no. 1, pp. 86–93, 2025.
- [20] K. Ratna *et al.*, “ANALISIS HYPERPARAMETER PADA KLASIFIKASI JENIS TANAMAN MENGGUNAKAN ALGORITMA RESNET50 DAN MOBILENETV2,” vol. 9, no. 6, pp. 9921–9928, 2025.
- [21] H. P. Tulili and A. Septiarini, “METODE DEEP LEARNING DENGAN TEKNIK,” vol. 8, no. 2, pp. 365–375, 2025.
- [22] L. Li, S. Zhang, and B. I. N. Wang, “Plant Disease Detection and Classification by Deep Learning — A Review,” no. Ccv, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [23] P. Convolutional and N. Network, “Penerapan convolutional neural network (cnn) untuk klasifikasi penyakit daun tomat,” vol. 4221, pp. 58–67, 2025.
- [24] A. Nurdin, D. Satria, Y. Kartika, A. Rezha, and E. Najaf, “Klasifikasi Penyakit Daun Tomat Dengan Metode Convolutional Neural Network Menggunakan Arsitektur Inception-V3,” no. 1, pp. 1–6, 2024.