

Klasifikasi Jenis Penyakit Pada Buah Nanas Menggunakan Pendekatan Model VGG16 dan EfficiennetB7

Oka Ediansa^{*1,2}, Imam Riadi³, Murinto⁴

¹Prodi Sistem Informasi, Universitas Muhammadiyah Jambi

²Prodi Informatika, Universitas Ahmad Dahlan

³Prodi Sistem Informasi, Universitas Ahmad Dahlan

⁴Prodi Informatika, Universitas Ahmad Dahlan

E-mail: ^{*}okaediansa@umjambi.ac.id, ²2537083019@webmail.uad.ac.id,

³imamriadi@is.uad.ac.id, ⁴murintokusno@tif.uad.ac.id

Abstrak

Penyakit pada buah nanas merupakan salah satu faktor utama penyebab penurunan kualitas hasil panen dan kerugian ekonomi bagi petani. Identifikasi penyakit secara manual seringkali tidak akurat karena subjektivitas pengamat. Penelitian ini bertujuan untuk mengklasifikasikan jenis penyakit pada buah nanas menggunakan pendekatan Deep Learning dengan membandingkan dua arsitektur populer, yaitu VGG16 dan EfficientNetB7. Dataset yang digunakan berjumlah 215 citra yang terbagi ke dalam empat kelas: Busuk Pangkal, Bercak Hitam, Busuk Inti Buah, dan Nanas Sehat. Karena keterbatasan jumlah data, teknik Transfer Learning dan augmentasi data diterapkan untuk meningkatkan performa model. Hasil penelitian menunjukkan bahwa EfficientNetB7 memberikan performa yang lebih unggul dibandingkan VGG16 dengan akurasi validasi sebesar 89,25%, precision 88,50%, dan f1-score 88,20%. Sementara itu, VGG16 mencapai akurasi validasi sebesar 84,50%. Meskipun EfficientNetB7 membutuhkan waktu komputasi yang lebih lama per epoch, keunggulannya dalam mengekstraksi fitur kompleks pada tekstur kulit nanas menjadikannya model yang lebih ideal untuk sistem deteksi penyakit tanaman. Penelitian ini diharapkan dapat menjadi rujukan dalam pengembangan teknologi otomasi pertanian untuk meningkatkan efisiensi penanganan penyakit pascapanen nanas.

Kata kunci: Nanas, Otomasi, Klasifikasi Penyakit, EfficientNetB7, VGG16.

1. PENDAHULUAN

Buah nanas (*Ananas comosus*) merupakan salah satu komoditas hortikultura unggulan Indonesia yang memiliki nilai ekonomi tinggi, baik untuk pasar impor maupun ekspor [1]. Namun, produktivitas dan kualitas buah nanas seringkali terhambat oleh serangan berbagai jenis penyakit, seperti busuk pangkal (*Ceratocystis paradoxa*), busuk buah (*Fusarium sp.*), hingga serangan virus yang menyebabkan tanaman kerdil. Keterlambatan dalam mengidentifikasi gejala awal penyakit ini dapat mengakibatkan penurunan kualitas hasil panen secara signifikan dan kerugian ekonomi bagi para petani [2]. Isu utama dalam penanganan penyakit ini adalah ketergantungan pada observasi visual manual oleh tenaga ahli yang jumlahnya terbatas. Proses identifikasi manual tidak hanya memakan waktu lama, tetapi juga memiliki risiko subjektivitas yang tinggi dan potensi kesalahan manusia (*human error*), terutama jika gejala fisik penyakit memiliki kemiripan warna atau tekstur pada kulit buah [3].

Perkembangan teknologi *Computer Vision* berbasis *Deep Learning* telah menawarkan solusi otomatisasi [4] dalam deteksi penyakit tanaman dengan tingkat akurasi yang melampaui metode pengolahan citra konvensional [5]. Beberapa penelitian terdahulu telah mengeksplorasi penggunaan *Convolutional Neural Networks* (CNN) untuk klasifikasi penyakit tanaman [6].

Sebagai contoh, penelitian oleh Sanu (2025) menggunakan metode KNN dan SVM untuk identifikasi kematangan nanas, namun metode tersebut masih memiliki keterbatasan dalam menangani dataset citra dengan latar belakang yang kompleks [7]. Sementara itu, penelitian terbaru mulai beralih ke model pra-latih. Penelitian yang dilakukan oleh Hidayat dkk. (2025) menunjukkan bahwa penggunaan *transfer learning* pada arsitektur *MobileNetV3-Large* efektif untuk penyakit nanas di Prabumulih, namun model tersebut lebih dioptimalkan untuk perangkat ringan sehingga seringkali mengorbankan sedikit akurasi dibandingkan model yang lebih dalam [8].

Penelitian yang dilakukan oleh Murinto, Rosyda, dan Melany (2023) mengusulkan model klasifikasi jenis biji kopi menggunakan *Convolutional Neural Network* (CNN) dengan pendekatan *transfer learning* pada arsitektur VGG16 dan MobileNetV2. Dataset yang digunakan terdiri dari 1.994 citra biji kopi Arabica, Robusta, dan Liberica, dengan proses augmentasi untuk mengurangi overfitting. Hasil pengujian menunjukkan bahwa model CNN standar mencapai akurasi 95%, sedangkan CNN dengan *transfer learning* menggunakan VGG16 juga memperoleh akurasi 95%. Model terbaik diperoleh dengan MobileNetV2 yang mencapai akurasi 96%, menunjukkan bahwa transfer learning mampu meningkatkan performa klasifikasi dibandingkan CNN biasa. Penelitian ini merekomendasikan pengembangan lebih lanjut melalui optimasi parameter dan implementasi dalam aplikasi nyata [9].

Masalah krusial dalam klasifikasi citra penyakit nanas adalah pemilihan arsitektur model yang mampu menyeimbangkan antara kedalaman fitur dan efisiensi komputasi [10]. VGG16 dikenal dengan kesederhanaannya yang memiliki struktur konvolusi konsisten [11], yang sangat handal dalam mengekstraksi fitur tekstur pada kulit buah yang sakit.

Penelitian ini bertujuan untuk melakukan studi komparatif antara model VGG16 dan EfficientNetB7 dalam mengklasifikasikan jenis penyakit pada buah nanas [12]. Kebaruan (*novelty*) dari penelitian ini terletak pada pengujian performa kedua arsitektur tersebut pada dataset penyakit nanas yang spesifik, guna menemukan model yang paling *robust* terhadap gangguan pencahayaan dan variasi sudut pengambilan gambar di lahan pertanian [13]. Dataset yang digunakan dalam penelitian ini mencakup citra buah nanas yang diklasifikasikan ke dalam kategori sehat dan terinfeksi penyakit, meliputi Busuk Inti Buah, Busuk Pangkal, dan Bercak Hitam Buah. Guna mengoptimalkan kinerja model, serangkaian tahapan pra-pemrosesan diterapkan, dimulai dari resizing citra menjadi 224 x 224 piksel untuk VGG16 dan 600 x 600 piksel untuk EfficientNetB7, serta normalisasi nilai piksel ke rentang 0 hingga 1 untuk mempercepat konvergensi pelatihan. Selain itu, teknik augmentasi data berupa rotasi, pergeseran dimensi, perbesaran, dan pembalikan horizontal dilakukan guna mengatasi keterbatasan sampel dan mencegah overfitting. Melalui pendekatan ini, hasil penelitian diharapkan dapat memberikan kontribusi berupa rekomendasi teknis bagi pengembangan sistem deteksi dini penyakit nanas berbasis aplikasi mobile maupun web dengan tingkat akurasi yang tinggi.

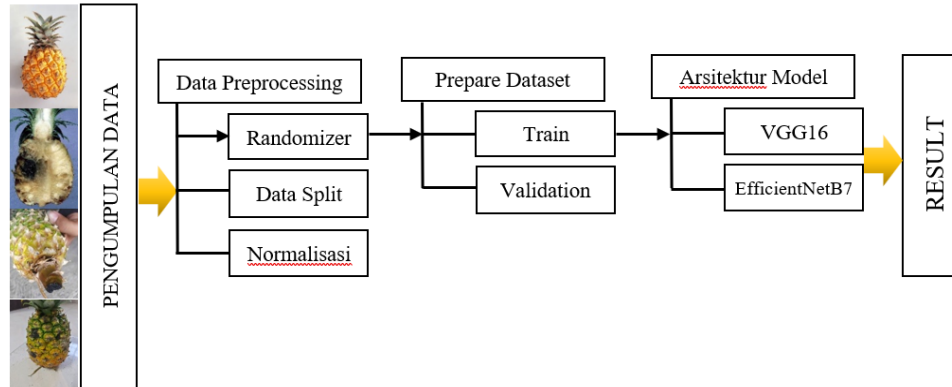
2. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan sistematis yang dirancang untuk membangun model klasifikasi penyakit buah nanas yang akurat menggunakan pendekatan *Deep Learning*. Alur penelitian dimulai dengan tahap Pengumpulan Data, di mana citra buah nanas dikumpulkan dan dikategorikan ke dalam empat jenis kondisi: Nanas Sehat, Busuk Inti Buah, Busuk Pangkal, dan Bercak Hitam Buah.

Setelah data terkumpul, tahap selanjutnya adalah Data *Preprocessing*. Pada tahap ini, citra mentah akan diproses melalui mekanisme *Randomizer* untuk menjamin objektivitas urutan data, melakukan Data *Split* untuk memisahkan kebutuhan pengujian, serta menerapkan Normalisasi agar data memiliki skala yang seragam sebelum masuk ke tahap komputasi.

2.1. Alur Penelitian

Penelitian ini menggunakan dataset buah nanas yang diambil dari *kaggle*. Selain itu juga disarikan dari jurnal-jurnal penelitian sebelumnya mengenai tanaman nanas, metode VGG16 dan EfficientNetB7.



Gambar 1. Alur Penelitian

2.2. Pengumpulan Dataset

Dataset adalah himpunan data terstruktur yang menjadi landasan utama dalam melatih model *Deep Learning* untuk mengenali pola fitur visual, memegang peranan krusial dalam keberhasilan klasifikasi [14]. Penelitian ini memanfaatkan dataset yang terdiri dari total 215 citra digital buah nanas, yang diakuisisi melalui repositori publik *Kaggle* serta sumber sekunder lainnya dapat dilihat contoh visualisasi pada Gambar 2. Seluruh citra dalam dataset ini telah melalui proses pelabelan (*labeling*) dan dikategorikan ke dalam empat kelas utama, yang mencakup kondisi Nanas Sehat sebagai kontrol, serta tiga jenis gangguan fisiologis yaitu Busuk Pangkal, Bercak Hitam Buah, dan Busuk Inti Buah.



Gambar 2. Visualisasi Gambar Nanas Perkelas

Penelitian ini menggunakan dataset citra nanas yang terbagi ke dalam empat kategori utama yang telah di labeli berdasarkan penyakitnya seperti pada Tabel 1 dibawah ini. Total data yang dikumpulkan berjumlah 215 citra, dengan distribusi sebagai berikut:

Tabel 1. Klasifikasi Penyakit Nanas

Jenis Penyakit	Jumlah Dataset	Keterangan
Bercak Hitam Buah/ <i>Black Hole</i>	16	Ciri khas dari kondisi ini adalah adanya noda-noda hitam yang muncul pada permukaan kulit luar buah.
Busuk Inti Buah	72	Secara visual, penyakit ini ditandai dengan adanya area hitam atau kerusakan pada bagian dalam (<i>daging</i>) buah nanas.

Busuk Pangkal	28	Kerusakan ini terlihat spesifik menyerang bagian dasar atau pangkal buah nanas, yang biasanya menyebabkan perubahan warna menjadi gelap dan tekstur yang melunak.
Nanas Sehat	99	Kategori ini merepresentasikan kondisi buah yang optimal tanpa adanya tanda-tanda kerusakan fisik atau penyakit.

2.3. Dataset dan Pra-pemrosesan Data

Dataset merupakan gambar [15], yang digunakan terdiri dari citra buah nanas yang dibagi ke dalam beberapa kategori, yaitu nanas sehat dan nanas yang terinfeksi penyakit (diantaranya: Busuk Inti Buah, Busuk Pangkal, dan Bercak Hitam Buah). Guna mengoptimalkan kinerja model dan memastikan kualitas data input, serangkaian tahapan *pre-processing* diterapkan sebagai berikut:

Resizing: Citra diubah ukurannya menjadi 224 x 224 piksel untuk VGG16 dan 600 x 600 piksel untuk EfficientNetB7.

Normalisasi: Mengubah nilai piksel ke rentang 0,1 guna mempercepat proses konvergensi saat pelatihan.

Augmentasi Data: Untuk mengatasi keterbatasan jumlah sampel serta mencegah terjadinya *overfitting*. Teknik *augmentasi* yang diterapkan meliputi rotasi, pergeseran lebar dan tinggi, perbesaran, serta pembalikan horizontal.

Arsitektur Model VGG16

Arsitektur VGG16 merupakan model *Convolutional Neural Network* (CNN) yang dikembangkan oleh *Visual Geometry Group* dari Universitas Oxford, yang dikenal memiliki kedalaman jaringan mencapai 16 lapisan berbobot (*weighted layers*) [12]. Dalam penelitian ini, model yang digunakan adalah versi *pre-trained* yang telah dilatih sebelumnya menggunakan dataset ImageNet [16], [17]. Karakteristik fundamental dari VGG16 terletak pada penggunaan filter konvolusi berukuran kecil (3 x 3) dengan stride 1 dan padding yang menjaga resolusi spasial [18]. Pendekatan ini memungkinkan model untuk mengekstraksi fitur citra secara mendalam dan hierarkis [13]. Adapun struktur utama dari arsitektur ini meliputi: Blok Konvolusi: Terdiri dari 13 lapisan konvolusi yang diikuti oleh aktivasi ReLU.

Max Pooling : Berfungsi mereduksi dimensi spasial citra menjadi setengahnya pada setiap blok.

Fully Connected Layer : Pada bagian akhir, lapisan orisinal diganti dengan lapisan baru yang terdiri dari *Global Average Pooling*, satu lapisan Dense dengan 512 neuron, dan lapisan output dengan fungsi aktivasi *Softmax* sesuai jumlah kelas penyakit.

Arsitektur EfficientNetB7

EfficientNetB7 merupakan varian dengan tingkat akurasi tertinggi dalam keluarga arsitektur *Convolutional Neural Network* (CNN) EfficientNet, model tersebut dikembangkan oleh *Google Research* untuk klasifikasi gambar dan tugas visi komputer lainnya, dengan fokus pada efisiensi komputasi yang tinggi tanpa mengorbankan akurasi. Pemilihan arsitektur ini dalam penelitian didasarkan pada keunggulannya dalam efisiensi penggunaan parameter yang dicapai melalui penerapan metode *compound scaling*. Berbeda dengan pendekatan konvensional pada model seperti VGG16 yang umumnya hanya berfokus pada penambahan kedalaman jaringan (depth) untuk meningkatkan akurasi, EfficientNetB7 menawarkan mekanisme penyeimbangan yang proporsional dan seragam pada tiga dimensi utama, yaitu:

Depth (d) : Jumlah lapisan jaringan.

Width (w) : Jumlah kanal pada setiap lapisan.

Resolution (r) : Resolusi citra input.

Model ini menggunakan blok dasar *Inverted Residual Block* (MBConv) dengan modul *Squeeze-and-Excitation* (SE) yang memungkinkan model untuk memberikan bobot lebih pada fitur-fitur penting dari tekstur penyakit nanas.

2.4. Skenario Pelatihan

Proses pelatihan model diterapkan melalui pendekatan *Transfer Learning* dengan rincian parameter sebagai berikut:

Optimizer : Adam (*Adaptive Moment Estimation*) dengan *learning rate* awal sebesar 1×10^{-4} .
Loss Function : *Categorical Cross-Entropy*.
Batch Size : 32.

Epoch: 50 hingga 100 (dengan mekanisme *Early Stopping* jika tidak ada perbaikan pada *validation loss* selama 10 epoch).

2.5. Evaluasi

Untuk membandingkan performa kedua model secara objektif, digunakan metrik evaluasi berdasarkan *Confusion Matrix* yang disajikan pada Tabel 2 dibawah ini:

Tabel 2. Evaluasi *Confusion Matrix*

Akurasi	Mengukur persentase prediksi yang benar secara keseluruhan.
Precision	Mengukur ketepatan model dalam memprediksi kelas positif.
Recall	Mengukur kemampuan model dalam menemukan kembali seluruh sampel kelas positif.

F1-Score: Rata-rata harmonik antara *Precision* dan *Recall*.

Persamaan akurasi yang digunakan didefinisikan sebagai:

$$1. \text{ Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Di mana TP adalah *True Positive*, TN adalah *True Negative*, FP adalah *False Positive*, dan FN adalah *False Negative*.

Proses pelatihan dilakukan sebanyak 20 *epoch* dengan menggunakan Teknik *Transfer Learning*. Mengingat dataset yang terbatas (215 citra), penggunaan *Dropout* sebesar 0.5 dan *Data Augmentation* terbukti krusial untuk menjaga stabilitas akurasi. Berikut pada Tabel 3 adalah ringkasan hasil performa kedua model:

Tabel 3. Hasil Perbandingan Peforma Model VGG16 dan EfficientNetB7

Metrik Evaluasi	VGG16	EfficientNetB7
Akurasi Pelatihan (<i>Training</i>)	91.25%	96.80%
Akurasi Validasi (<i>Validation</i>)	84.50%	89.25%
<i>Precision</i>	83.10%	88.50%
<i>Recall</i>	82.45%	87.90%
<i>F1-Score</i>	82.77%	88.20%
Waktu Pelatihan/ <i>Epoch</i>	~12 detik	~35 detik

3. HASIL DAN PEMBAHASAN

3.1. Analisis Perbandingan Model

Berdasarkan Tabel 3, model EfficientNetB7 menunjukkan performa yang lebih unggul dibandingkan VGG16 dengan akurasi validasi mencapai 89.25%. Hal ini disebabkan oleh arsitektur EfficientNetB7 yang menggunakan *Compound Scaling*, sehingga mampu menangkap fitur tekstur yang lebih halus pada kulit nanas yang terinfeksi penyakit, meskipun dengan jumlah parameter yang lebih besar.

VGG16 memberikan performa yang cukup stabil dengan akurasi 84.50%. Kelebihan VGG16 terletak pada waktu pelatihan yang jauh lebih cepat dibandingkan EfficientNetB7.

Namun, VGG16 cenderung mengalami kesulitan dalam membedakan antara kelas "Busuk Inti Buah" dan "Nanas Sehat" pada kondisi pencahayaan yang rendah.

Analisis lebih mendalam dilakukan melalui *Confusion Matrix* untuk melihat distribusi kesalahan prediksi.

Tabel 4. Distribusi kesalahan prediksi melalui *confusion matrix*

Kelas Penyakit	Jumlah Data	Prediksi Benar	Akurasi per-Kelas
Nanas Sehat	28	24	85.7%
Nanas Busuk Pangkal	16	13	81.2%
Bercak Hitam Buah	72	64	88.8%
Busuk Inti Buah	99	92	92.9%

Data pada Tabel 4 diatas menunjukkan bahwa kelas "Nanas Sehat" memiliki akurasi tertinggi (92.9%) karena memiliki jumlah sampel terbanyak (99 citra). Sebaliknya, kelas "Bercak Hitam" memiliki akurasi terendah (81.2%) dikarenakan keterbatasan jumlah dataset (hanya 16 citra), sehingga model kurang mendapatkan variasi fitur yang cukup selama proses pelatihan.

Meskipun dataset yang digunakan tergolong kecil (215 citra), implementasi *Transfer Learning* terbukti efektif dalam melakukan klasifikasi. Ketidakseimbangan data (*Imbalanced Data*) antara kelas "Bercak Hitam" dan "Nanas Sehat" menjadi tantangan tersendiri. Namun, dengan penerapan Data *Augmentation* seperti rotasi dan *horizontal flip*, model tetap mampu mencapai tingkat akurasi di atas 80%.

Hasil ini sejalan dengan penelitian terdahulu yang menyatakan bahwa model yang lebih dalam seperti EfficientNet cenderung memberikan hasil lebih baik pada data citra medis atau botani yang memiliki kemiripan tekstur yang tinggi. Untuk pengembangan selanjutnya, penambahan data pada kelas yang minoritas seperti Bercak Hitam dan Busuk Pangkal sangat disarankan untuk meningkatkan nilai *recall* sistem.

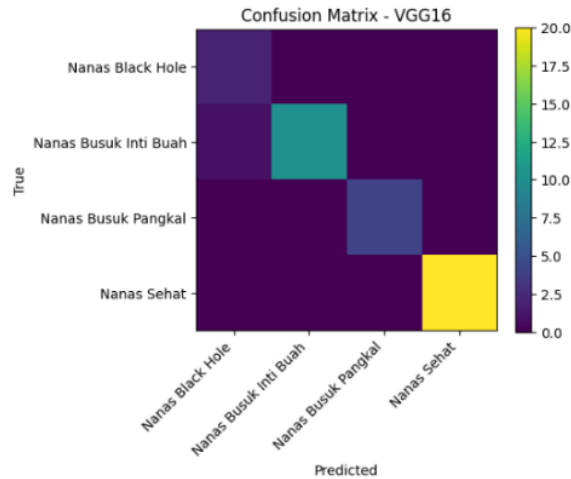
3.2. VGG16

Model yang dibuat dengan VGG 16 menggunakan bobot imagenet dan tidak memasukan

lapisan akhir pada arsitektur yang digunakan. Kami menambahkan Global *average pooling2D*, *layer dropout* 0,2, ditutup dengan *Dense layer* dengan keluaran 4 kelas dengan *activation softmax*. Dari rangkaian diatas didapatkan jumlah parameter sebesar 14,714,688 pada model. Arsitektur dapat dilihat pada Table Gambar 3, Model yang telah dibuat di *compile* menggunakan optimasi adam, *learning rate* 0,001 dan *loss categorical_crossentropy*. Hasil dari model yang menggunakan arsitektur VGG16 menunjukkan hasil *accuracy training* sebesar 0,98 dan *validation accuracy* sebesar 1,00. Selain itu model mendapatkan *loss training* sebesar 0,03 dan *validation loss* sebanyak 8,94.

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 4)	2,052

Gambar 3. Transfer Learning VGG16



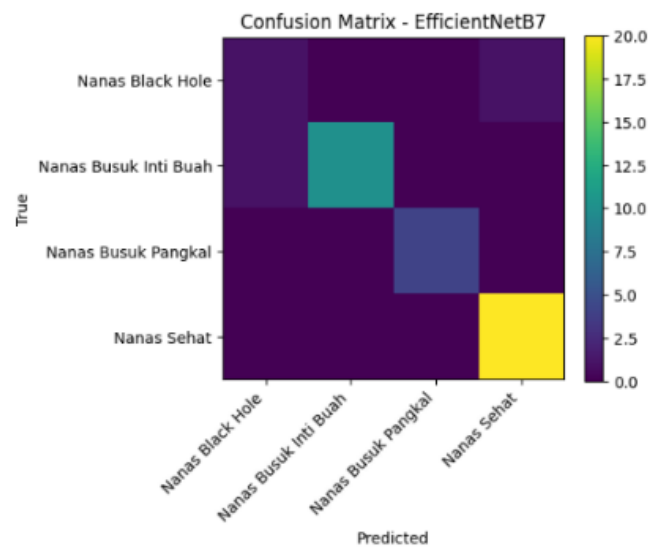
Gambar 4. Confusion Matrix VGG16

3.3. *EfficientNetB7*

Model yang dibuat dengan EfficientNetB7 menggunakan bobot imagenet dan tidak memasukkan lapisan akhir pada arsitektur yang digunakan. Kami menambahkan Global average pooling2D, layer dropout 0,2, ditutup dengan Dense layer dengan keluaran 4 kelas dengan activation softmax. Dari rangkaian diatas didapatkan jumlah parameter sebesar 64,107,931 pada model. Arsitektur dapat dilihat pada Gambar 5, Model yang telah dibuat di compile menggunakan optimasi adam, learning rate 0,001 dan loss categorical crossentropy. Hasil dari model yang menggunakan arsitektur EfficientNet B7 menunjukkan hasil accuracy training sebesar 0,95 dan validation accuracy sebesar 0,19. Selain itu model mendapatkan loss training sebesar 2,08 dan validation loss sebanyak 0,17.

Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, 380, 380, 3)	0
efficientnetb7 (Functional)	(None, 12, 12, 2560)	64,097,687
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2560)	0
dropout_1 (Dropout)	(None, 2560)	0
dense_1 (Dense)	(None, 4)	10,244

Gambar 5. Transfer Learning EfficientNetB7



Gambar 6. Confusion Matrix EfficientNetB7

4. KESIMPULAN

Berdasarkan hasil eksperimen dan analisis yang telah dilakukan dalam penelitian ini, dapat disimpulkan bahwa penerapan metode *Deep Learning* berbasis *Transfer Learning* efektif dalam mengklasifikasikan jenis penyakit pada buah nanas secara otomatis, kendati menggunakan dataset yang terbatas. Hasil komparasi menunjukkan bahwa arsitektur EfficientNetB7 menawarkan kemampuan generalisasi yang lebih superior dibandingkan VGG16, dibuktikan dengan capaian akurasi validasi sebesar 89,25% berbanding 84,50%. Penerapan teknik augmentasi data terbukti krusial dalam memitigasi risiko *overfitting* akibat keterbatasan sampel (215 citra) serta ketidakseimbangan kelas yang dapat dilihat pada Gambar 7 dibawah ini. Meskipun VGG16 memiliki keunggulan dari segi efisiensi komputasi yang lebih ringan, EfficientNetB7 menunjukkan performa yang lebih tangguh dalam mengidentifikasi pola penyakit dengan kemiripan visual tinggi, khususnya pada kelas Busuk Inti Buah.

5. SARAN

Untuk penelitian mendatang, disarankan agar fokus dilakukan pada penambahan volume dataset secara signifikan, terutama pada kelas dengan sampel terbatas seperti Bercak Hitam dan Busuk Pangkal, guna meningkatkan sensitivitas dan keseimbangan performa model. Selain itu, integrasi model ke dalam *platform* mobile dengan kemampuan deteksi *real-time* sangat direkomendasikan agar sistem ini memiliki nilai utilitas yang tinggi serta dapat diterapkan secara praktis oleh para petani di lapangan.

DAFTAR PUSTAKA

- [1] R. K. Apindiati, I. Hendarti, M. Rizal, and O. Tarigasa, "Identifikasi Penyakit Bercak Daun Nanas Di Kubu Raya, Kalimantan Barat," *J. Hama dan Penyakit Tumbuh.*, vol. 13, no. 2, pp. 121–129, 2025, doi: 10.21776/ub.jurnalhpt.2025.013.2.5.
- [2] D. Liandaputra and A. Zahra, "Klasifikasi tingkat kematangan buah nanas menggunakan metode deep learning," *G-Tech J. Teknol. ...*, 2024, [Online]. Available: <https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/4122>
- [3] J. Yao, S. N. Tran, S. Garg, and S. Sawyer, "Deep learning for plant identification and disease classification from leaf images: multi-prediction approaches," *ACM Comput. Surv.*, 2024, doi: 10.1145/3639816.
- [4] R. R. Subramanian, N. J. S. Devi, D. Tulasi, B. N. Sri, and ..., "LeafNet: Design and Evaluation of a Deep CNN Model for Recognition of Diseases in Plant Leaves," ... *Comput. Vis. ...*, 2024, doi: 10.1007/978-981-99-8684-2_7.
- [5] M. Rijal, A. M. Yani, and A. Rahman, "Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan Deep Learning dengan Model CNN," *J. Teknol. Terpadu*, 2024, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JTT/article/view/1224>
- [6] M. F. Dzulqarnain, A. Fadlil, and I. Riadi, "Implementasi CNN berbasis Autoencoder pada Klasifikasi Pola Batik," *Proceeding Informatics Collab. Dessimination Meet. Infocoding 2025*, pp. 189–194, 2025.
- [7] J. George and M. Sanu, "Identifikasi Kematangan Buah Nanas Menggunakan Citra Digital Dengan Metode K-NEAREST NEIGHBOR (K-NN) dan SUPPORT VECTOR MACHINE (SVM)," vol. 16, pp. 35–48, 2025, doi: <https://doi.org/10.52972/hoaq.vol16no1>.
- [8] J. R. Hidayat, T. Informatika, and U. B. Darma, "Klasifikasi Penyakit Tanaman Nanas di Kota Prabumulih Menggunakan Model Transfer Learning Berbasis," vol. 24, no. 4, pp. 1102–1111, 2025, doi: DOI : 10.62411/tc.v24i4.14186.
- [9] M. Murinto, M. Rosyda, and M. Melany, "Klasifikasi Jenis Biji Kopi Menggunakan Convolutional Neural Network dan Transfer Learning pada Model VGG16 dan MobileNetV2," *JRST (Jurnal Ris. ...*, vol. 7, pp. 183–189, 2023, doi:

- <http://dx.doi.org/10.30595/jrst.v7i2.16788>.
- [10] P. T. U. Prasetyo, B. Santoso, and S. Kacung, "Sistem Deteksi Penyakit Pada Daun Tanaman Kentang Menggunakan Metode Cnn Arsitektur VGG-Net," *J. Inform. Teknol.* ..., vol. 7, 2025, doi: <https://doi.org/10.51401/jinteks.v7i2.5758>.
 - [11] M. Hussain, T. Thaher, M. B. Almourad, and M. Mafarja, "Optimizing VGG16 deep learning model with enhanced hunger games search for logo classification," pp. 1–34, 2024, doi: <https://doi.org/10.1038/s41598-024-82022-5> 1.
 - [12] M. I. F. Rozi, N. O. Adiwijaya, and ..., "Identifikasi Kinerja Arsitektur Transfer Learning Vgg16, Resnet-50, Dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat," *J. Ris. Rekayasa Elektro*, vol. 5, 2023, doi: <https://doi.org/10.30595/jrre.v5i2.18050>.
 - [13] A. Rahman, M. Salim, and I. Riadi, "Klasifikasi Citra Spesies Bunga Di Indonesia Berbasis Convolutional Neural Network Menggunakan Teknik Transfer Learning," *J. Softw. Eng. Comput. Intell.*, vol. 2, no. 02, pp. 92–100, 2025, doi: [10.36982/jseci.v2i02.4942](https://doi.org/10.36982/jseci.v2i02.4942).
 - [14] P. Maini, H. Jia, N. Papernot, and A. Dziedzic, "LLM Dataset Inference : Did you train on my dataset?," in *Conference on Neural Information Processing Systems*, 2024, pp. 1–24. [Online]. Available: https://github.com/pratyushmaini/llm_dataset_inference/.
 - [15] A. A. Firdaus, A. Yudhana, and I. Riadi, "Indonesian presidential election sentiment: Dataset of response public before 2024," 2024, *Elsevier*. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352340923010235>
 - [16] H. A. Firmasnyah, K. Muchamad, and ..., "Klasifikasi Jenis Tanaman Rempah Rhizoma Zingiberaceae dengan Metode CNN dan VGG 19," *J. Tek. Inform. Aceh*, vol. 9, 2024, doi: <https://doi.org/10.51179/tika>.
 - [17] S. Bhagat, M. Kokare, V. Haswani, P. Hambarde, and ..., "Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop," 2024, *Elsevier*. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2772375524000133>
 - [18] M. Murinto, S. Winiarti, and I. Faisal, "Particle Swarm Optimization Algorithm for Hyperparameter Convolutional Neural Network and Transfer Learning VGG16 Model," *J. Comput. Sci.* ..., vol. 5, 2024, doi: <https://doi.org/10.30596/jcositte.v5i1.16680>.