

Klasifikasi Penyakit Daun Padi Berbasis Transfer Learning Convolutional Neural Network menggunakan MobileNetV2 dan EfficientNetB0

Rice Leaf Disease Classification Based on Convolutional Neural Network Transfer Learning Using MobileNetV2 and EfficientNetB0

Nuranissa D. Paemo^{*1}, Abdul Rahman Ismail², Rusni Harun³

^{1,3} Fakultas Ilmu Komputer dan Sains, Universitas Ichsan Gorontalo Utara,

² Prodi Teknologi Rekayasa dan Perangkat Lunak, Universitas Negeri Gorontalo

¹nuranissadpaemo@gmail.com, ²abdulrahmanismail@ung.ac.id, ³rusniharun28@gmail.com

Received: October 25, 2025 | Revised: October 31, 2025 | Accepted: December 02, 2025

Abstrak

Tanaman padi merupakan komoditas strategis yang sangat vital bagi ketahanan pangan nasional Indonesia. Identifikasi penyakit padi secara konvensional yang mengandalkan pengamatan visual langsung memiliki keterbatasan sehingga dibutuhkan pengembangan sistem identifikasi otomatis berbasis *Transfer Learning CNN (Convolutional Neural Network)* pada citra daun padi menjadi sangat penting. Penelitian ini menerapkan model arsitektur MobileNetV2 Dan EfficientNetB0 untuk membandingkan kinerja kedua model dalam mengklasifikasikan penyakit pada daun padi. Dataset yang digunakan 2756 dari Kaggle, data validation terdiri dari 690 gambar dan data test 383 gambar yang diklasifikasikan ke dalam 6 kategori penyakit daun padi. Dataset mencakup enam kelas yang terdiri atas lima kategori penyakit daun padi (*Bacterial Leaf Blight*, *Brown Spot*, *Leaf Blast*, *Leaf scald*, *Sheath Blight*) dan satu kategori daun sehat (*Healthy Rice Leaf*). Tahapan preprocessing seperti normalisasi piksel, augmentasi, dan pengubahan ukuran menjadi 224x224 piksel. Tahap pelatihan dilakukan 20 epoch dengan learning rate sebesar 0,001 menggunakan optimizer Adam. Berdasarkan hasil pengujian dari arsitektur MobileNetV2 diperoleh hasil akurasi 83.29%, presisi 87.20%, recall 76.50%, dan f1 score 82.87%. Sedangkan arsitektur EfficientNetB0 akurasi 86.95%, presisi 90.00%, recall 80.15%, dan f1 score 86.78%. Perbedaan ini menegaskan bahwa meskipun kedua model efektif untuk klasifikasi penyakit padi, EfficientNetB0 menawarkan margin unggul sekitar 3–4% pada setiap metrik, menjadikannya pilihan lebih stabil untuk kebutuhan deteksi dini penyakit daun padi yang akurat. Penelitian ini memberikan kontribusi untuk mempercepat proses identifikasi serta pengambilan keputusan di bidang pertanian.

Kata kunci: Penyakit Daun Padi, Klasifikasi, CNN, MobileNetV2, EfficientNetB0

Abstract

Rice is a strategic commodity that plays a vital role in Indonesia's national food security. Conventional identification of rice diseases, which relies on direct visual inspection, has several limitations; therefore, developing an automated identification system using Transfer Learning-based Convolutional Neural Networks (CNNs) has become increasingly important. This study employs MobileNetV2 and EfficientNetB0 architectures to compare the performance of both models in classifying rice leaf diseases. The dataset consists of 2,756 images obtained from Kaggle, with 690 images used for validation and 383 images for testing, classified into six categories of rice leaf conditions. These categories include five disease classes—Bacterial Leaf Blight, Brown Spot, Leaf Blast, Leaf Scald, and Sheath Blight—and one healthy leaf class. Preprocessing stages include pixel normalization, data augmentation, and resizing to 224×224

pixels. Model training was conducted for 20 epochs using a learning rate of 0.001 with the Adam optimizer. Evaluation results show that MobileNetV2 achieved an accuracy of 83.29%, precision of 87.20%, recall of 76.50%, and an F1-score of 82.87%. In comparison, EfficientNetB0 achieved an accuracy of 86.95%, precision of 90.00%, recall of 80.15%, and an F1-score of 86.78%. These findings highlight that although both models perform effectively for rice disease classification, EfficientNetB0 consistently provides a superior margin of approximately 3–4% across all metrics, making it a more stable choice for accurate early detection of rice leaf diseases. This study contributes to accelerating the identification process and supporting decision-making in the agricultural sector.

Keywords: Rice Leaf Disease, Classification, CNN, MobileNetV2, EfficientNetB0

1. PENDAHULUAN

Tanaman padi merupakan komoditas strategis yang sangat vital bagi ketahanan pangan nasional Indonesia[1][2]. Pada tahun 2024, luas panen padi mencapai sekitar 10,05 juta hektare dengan produksi sebanyak 53,14 juta ton gabah kering giling. Namun, produksi padi menghadapi berbagai tantangan serius, salah satunya adalah serangan organisme pengganggu tanaman (OPT) termasuk berbagai jenis penyakit yang menyerang tanaman padi [3]. Berdasarkan data prakiraan OPT pada musim tanam 2024, kejadian serangan penyakit utama padi mencapai 74.921,94 hektare, yang meliputi blas (10.993,09 ha), hawar daun bakteri/HDB (10.928,90 ha), dan tungro (265,54 ha)[3]. Penyakit-penyakit ini dapat menurunkan hasil panen secara signifikan jika tidak terdeteksi dan ditangani sejak dini[4]. Identifikasi penyakit padi secara konvensional yang mengandalkan pengamatan visual langsung oleh petani atau petugas lapangan memiliki keterbatasan, seperti memerlukan keahlian khusus, subjektivitas penilaian, dan memakan waktu yang cukup lama [5][6]. Perkembangan teknologi deep learning, khususnya *Convolutional Neural Network* (CNN), telah membuka peluang baru dalam otomatisasi deteksi penyakit tanaman[7]. CNN merupakan arsitektur jaringan saraf tiruan yang dirancang khusus untuk memproses data citra dengan meniru cara kerja korteks visual manusia. Metode ini mampu mempelajari fitur-fitur penting dari citra secara otomatis melalui lapisan konvolusi, pooling, dan fully-connected layer[8][9].

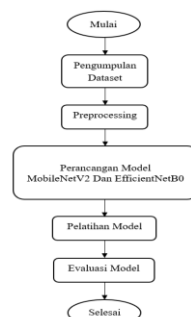
Berbagai penelitian telah membuktikan efektivitas CNN dalam klasifikasi penyakit tanaman padi. Penelitian menggunakan arsitektur ResNet50V2 berhasil mengidentifikasi delapan jenis penyakit daun padi dengan akurasi mencapai 94,14%. Studi lain yang menggunakan model CNN standar untuk tiga kelas klasifikasi penyakit padi menunjukkan *training accuracy* sebesar 0,9905 dan *validation accuracy* 0,8000 pada 50 epoch. Keunggulan metode CNN terletak pada kemampuannya mengekstrak fitur hierarkis secara otomatis tanpa memerlukan ekstraksi fitur manual yang kompleks [10]. Penelitian selanjutnya menggunakan model MobileNetV2 ternyata sangat berhasil dalam mengklasifikasikan objek dengan akurasi 78,33% pada 50 epoch[11]. Pendekatan transfer learning dengan arsitektur pre-trained seperti ResNet50V2, VGG16, dan DenseNet telah terbukti meningkatkan performa klasifikasi penyakit padi. Transfer learning memungkinkan model memanfaatkan pengetahuan yang telah dipelajari dari dataset besar untuk diterapkan pada domain spesifik dengan jumlah data yang lebih terbatas[12]. Dalam konteks efisiensi komputasi, arsitektur *lightweight* seperti MobileNetV2 dan EfficientNetB0 menjadi pilihan menarik karena menawarkan keseimbangan antara akurasi tinggi dan beban komputasi yang rendah, sehingga cocok untuk implementasi pada perangkat terbatas sumber daya seperti perangkat seluler. MobileNetV2 menunjukkan performa baik dengan parameter yang kompak dan akurasi hingga 95% pada klasifikasi penyakit daun padi [13], sementara EfficientNetB0 dikenal efektif dalam menangkap fitur kompleks dengan skalabilitas yang optimal[9][14]. Selain itu, teknik data augmentation juga berperan penting dalam memperbanyak variasi data training untuk

meningkatkan generalisasi model [15]. Meskipun berbagai penelitian telah membuktikan efektivitas CNN dalam klasifikasi penyakit padi dengan akurasi tinggi, sebagian besar studi menggunakan arsitektur yang kompleks dan berat seperti ResNet50V2, VGG16, atau DenseNet yang memerlukan sumber daya komputasi besar dan sulit diimplementasikan pada perangkat terbatas. Penelitian yang membandingkan arsitektur lightweight secara komprehensif untuk kasus penyakit padi masih terbatas, padahal hal ini penting untuk pengembangan sistem yang dapat diakses oleh petani di lapangan.

Berdasarkan latar belakang dan gap penelitian yang telah diuraikan, penelitian ini bertujuan untuk (1) Mengembangkan sistem klasifikasi otomatis penyakit daun padi menggunakan pendekatan transfer learning dengan dua arsitektur lightweight CNN yaitu MobileNetV2 dan EfficientNetB0; (2) Membandingkan kinerja kedua arsitektur berdasarkan metrik akurasi, presisi, recall, dan F1-score untuk enam kelas penyakit daun padi; (3) Menganalisis kelebihan dan kekurangan masing-masing model dalam konteks implementasi pada perangkat terbatas sumber daya; (4) Memberikan rekomendasi arsitektur terbaik untuk aplikasi deteksi dini penyakit padi yang cepat, akurat, dan dapat diakses oleh petani. Hasil Yang diharapkan dapat menjadi alat bantu bagi petani dalam melakukan deteksi dini penyakit padi secara cepat, akurat, dan objektif, sehingga dapat mempercepat pengambilan keputusan dalam pengendalian penyakit dan meminimalkan penurunan hasil panen[16].

2. METODE PENELITIAN

Pada penelitian ini digunakan metode Transfer Learning berbasis CNN (Convolutional Neural Network) untuk mengenali pola pada citra daun padi. Dua model arsitektur yang diterapkan, yaitu MobileNetV2 dan EfficientNetB0 digunakan untuk membandingkan kinerja keduanya dalam mengklasifikasikan penyakit pada daun padi. Tahapan penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Pada gambar 1 menggambarkan alur penelitian yang dimulai dengan pengumpulan dataset citra daun padi dari sumber tepercaya. Tahap berikutnya adalah preprocessing data untuk menstandarkan ukuran, kualitas, dan kebersihan citra agar siap diproses lebih lanjut. Setelah itu, model dirancang menggunakan metode CNN dengan membandingkan kinerja dua arsitektur lightweight, yakni MobileNetV2 Dan EfficientNetB0, kemudian dilakukan pelatihan model menggunakan skema transfer learning yang sesuai. Pada tahap akhir, hasil prediksi dianalisis melalui evaluasi metrik seperti akurasi, presisi, recall, dan F1-score per kelas untuk menilai performa dan menarik kesimpulan penelitian.

2.1 Pengumpulan Data

Penelitian ini Menggunakan dataset citra penyakit padi yang diperoleh dari platform Kaggle dengan judul “Rice_Leaf_AUG”. Dataset mencakup enam kelas yang terdiri atas lima

kategori penyakit daun padi dan satu kategori daun sehat. Seluruh citra dilakukan pemeriksaan awal untuk memastikan kesesuaian label dan kualitas gambar (ketajaman, fokus, dan ketiadaan duplikasi). Berikut adalah jenis daun padi yang digunakan dalam eksperimen dapat dilihat pada Tabel 1.

Tabel 1. Jenis Daun Padi		
No	Nama	Gambar
1	<i>Bacterial Leaf Blight</i>	
2	<i>Brown Spot</i>	
3	<i>Healthy Rice Leaf</i>	
4	<i>Leaf Blast</i>	
5	<i>Leaf scald</i>	
6	<i>Sheath Blight</i>	

2.2 Preprocessing

Preprocessing merupakan tahap awal dalam pengolahan data, yang berfungsi untuk mempersiapkan data agar siap digunakan pada proses klasifikasi dan analisis selanjutnya. Tahap ini bertujuan meningkatkan kualitas data sehingga model dapat menghasilkan hasil prediksi model yang lebih akurat [17]. Tahapan ini dimulai dari mengubah ukuran semua gambar ke 224x224 piksel, *batch size* diubah menjadi 32, kemudian normalisasi nilai piksel ke skala [0,1], kemudian split dataset atau membagi dataset yang mau dilatih menjadi data training, data validation, dan data test. Adapun data training terdiri dari 2756 gambar, data validation terdiri dari 690 gambar dan data test 383 gambar yang diklasifikasikan ke dalam 6 kategori penyakit daun padi.

2.3 Perancangan Model

a. Arsitektur MobileNetV2

Arsitektur model yang digunakan mengadopsi Transfer Learning berbasis MobileNetV2 dengan resolusi input 224×224×3 untuk menyeimbangkan akurasi dan efisiensi komputasi pada tugas klasifikasi berjumlah enam kelas. Model dibangun dengan membekukan seluruh bobot backbone MobileNetV2, kemudian menambahkan GlobalAveragePooling2D untuk mereduksi dimensi spasial menjadi vektor fitur 1.280 kanal, diikuti lapisan dropout sebagai regularisasi, dan

sebuah classifier Dense berukuran 6 neuron sebagai kepala klasifikasi. Konfigurasi ini menghasilkan total 2.265.670 parameter, dengan 7.686 parameter dapat dilatih pada kepala klasifikasi dan 2.257.984 parameter non-trainable pada backbone, sehingga proses pelatihan fokus pada penyesuaian representasi tingkat akhir tanpa mengubah representasi dasar yang sudah dipelajari dari data skala besar. Desain ini dipilih untuk mempercepat konvergensi, mengurangi risiko overfitting pada dataset domain spesifik, serta mempertahankan footprint memori yang ringkas untuk kemudahan implementasi pada lingkungan terbatas sumber daya. Berikut adalah arsitektur model dapat dilihat pada tabel 2.

Tabel 2. Model MobileNetV2

Layer (type)	Output Shape	Param
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 6)	7,686

Total params: 2,265,670 (8.64 MB)

Trainable params: 7,686 (30.02 KB)

Non-trainable params: 2,257,984 (8.61 MB)

b. Arsitektur EfficientNetB0

Arsitektur model EfficientNetB0 dengan ukuran input $224 \times 224 \times 3$, yang dikenal efektif dalam menyeimbangkan kedalaman, lebar, dan resolusi jaringan untuk memperoleh representasi fitur yang kaya pada biaya komputasi moderat. Seluruh bobot backbone dibekukan, kemudian keluaran fitur berukuran $7 \times 7 \times 1280$ diringkas menggunakan GlobalAveragePooling2D menjadi vektor 1.280 dimensi, dilanjutkan lapisan dropout sebagai regularisasi, dan kepala klasifikasi Dense dengan 6 neuron untuk prediksi multi-kelas. Arsitektur ini memiliki total 4.057.257 parameter, dengan 7.686 parameter trainable terbatas pada kepala klasifikasi dan 4.049.571 parameter non-trainable pada backbone, sehingga pelatihan berfokus pada adaptasi tingkat akhir sambil mempertahankan pengetahuan dasar yang telah dipelajari. Konfigurasi ini dipilih untuk meningkatkan akurasi dibanding backbone yang lebih ringan, namun tetap menjaga footprint memori sekitar 15,48 MB agar kompatibel dengan skenario komputasi terbatas. Berikut adalah arsitektur model dapat dilihat pada tabel 3.

Tabel 3. Model EfficientNetB0

Layer (type)	Output Shape	Param
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571

global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 6)	7,686

Total params: 4,057,257 (15.48 MB)

Trainable params: 7,686 (30.02 KB)

Non-trainable params: 4,049,571(15.45 MB)

2.4 Pelatihan Model

Tahap pelatihan dilakukan selama maksimal 20 epoch dengan learning rate awal sebesar 0,001 menggunakan optimizer Adam dan fungsi *loss categorical_crossentropy* untuk tugas klasifikasi enam kelas. Kedua model MobileNetV2 dan EfficientNetB0 dikompilasi dengan set metrik yang sama. Untuk mencegah *overfitting* serta memastikan bobot terbaik yang benar-benar generalis, diterapkan *callback Early Stopping* yang memantau *val_accuracy* dengan menghentikan pelatihan jika tidak ada peningkatan dalam 5 epoch, dan opsi *restore_best_weights=True* sehingga pada akhir proses pelatihan model otomatis kembali ke bobot validasi terbaik. Proses *fit* dijalankan terpisah untuk masing-masing backbone dengan pasangan dataset latih dan validasi yang sesuai, sehingga kurva metrik dan histori pelatihan dapat dibandingkan secara sejajar antara MobileNetV2 dan EfficientNetB0.

2.5 Evaluasi

Confusion matrix digunakan dalam proses evaluasi klasifikasi untuk menilai kinerja dan tingkat keakuratan suatu model [18] [19]. Melalui confusion matrix, kinerja model dapat diukur berdasarkan empat komponen utama, yaitu:

- True Positive(TP) merupakan jumlah data yang benar-benar termasuk dalam kelas positif dan berhasil diprediksi dengan tepat sebagai positif oleh model.
- True Negative(TN) merupakan jumlah data yang sebenarnya termasuk dalam kelas negatif dan berhasil diprediksi dengan benar sebagai negatif oleh model.
- False Positive(FP) Merupakan Jumlah data yang sebenarnya negatif namun keliru diprediksi sebagai positif oleh model, dikenal sebagai kesalahan tipe I.
- False Negative(FN) merupakan jumlah data yang sebenarnya positif namun keliru diprediksi sebagai negatif oleh model, dikenal sebagai kesalahan tipe II.

Akurasi (*accuracy*) menunjukkan perbandingan antara jumlah prediksi yang benar (baik positif maupun negatif) terhadap keseluruhan jumlah data, atau dapat diartikan sebagai tingkat kesesuaian antara hasil prediksi dengan nilai aktual. Nilai akurasi dihitung menggunakan persamaan [20].

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

Presisi (*precision*) adalah rasio antara jumlah prediksi positif yang benar dengan total data yang diprediksi positif. Metrik ini menunjukkan sejauh mana model mampu memprediksi kelas positif dengan tepat. Nilai presisi ditentukan melalui persamaan [20]

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

Recall adalah rasio antara jumlah prediksi positif yang benar dengan seluruh data yang sebenarnya termasuk dalam kelas positif. Nilai recall menggambarkan kemampuan model dalam mendeteksi semua contoh positif yang ada di dataset, dan diperoleh menggunakan persamaan[20]

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

F1-Score adalah nilai rata-rata harmonik antara presisi dan recall. Metrik ini digunakan terutama ketika data bersifat tidak seimbang (imbalanced class), karena mampu menggabungkan kedua metrik tersebut menjadi satu nilai tunggal yang lebih representatif terhadap performa model klasifikasi. Nilai F1-Score dihitung berdasarkan persamaan [20].

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

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pelatihan dan Pengujian Model MobileNetV2 dan EfficientNetB0

Pada penelitian ini, proses pelatihan model dilakukan 20 epoch untuk memastikan bahwa model dapat belajar secara bertahap dan mencapai stabilitas performa. Selama pelatihan, setiap epoch mencatat metrik penting seperti akurasi, loss, precision, dan recall, baik pada data pelatihan maupun data validasi. Hasil lengkap dari proses pelatihan tersebut ditampilkan pada gambar 2 yang memperlihatkan bagaimana performa model berkembang dari epoch awal hingga epoch akhir.

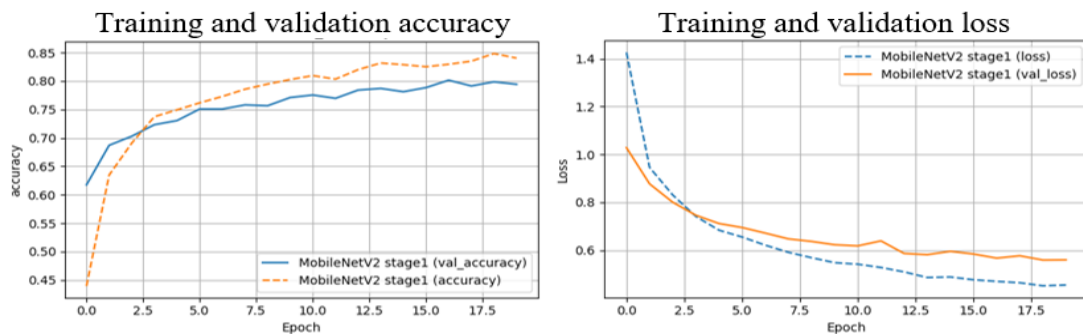
Epoch 1/20	
87/87	104s 1s/step - accuracy: 0.3260 - loss: 1.7253 - precision: 0.4391 - recall: 0.1538 - val_accuracy: 0.6174 - val_loss: 1.0285 - val_precision: 0.7540 - val_recall: 0.4130
Epoch 2/20	
87/87	140s 1s/step - accuracy: 0.6355 - loss: 0.9798 - precision: 0.7589 - recall: 0.4663 - val_accuracy: 0.6870 - val_loss: 0.8769 - val_precision: 0.8102 - val_recall: 0.5319
Epoch 3/20	
87/87	95s 1s/step - accuracy: 0.6749 - loss: 0.8679 - precision: 0.7853 - recall: 0.5481 - val_accuracy: 0.7029 - val_loss: 0.8000 - val_precision: 0.8126 - val_recall: 0.5971
Epoch 4/20	
87/87	138s 1s/step - accuracy: 0.7428 - loss: 0.7419 - precision: 0.8205 - recall: 0.6386 - val_accuracy: 0.7232 - val_loss: 0.7456 - val_precision: 0.8406 - val_recall: 0.6116
Epoch 5/20	
87/87	143s 1s/step - accuracy: 0.7415 - loss: 0.6935 - precision: 0.8223 - recall: 0.6456 - val_accuracy: 0.7304 - val_loss: 0.7110 - val_precision: 0.8416 - val_recall: 0.6391
Epoch 6/20	
87/87	145s 1s/step - accuracy: 0.7560 - loss: 0.6454 - precision: 0.8411 - recall: 0.6726 - val_accuracy: 0.7507 - val_loss: 0.6942 - val_precision: 0.8294 - val_recall: 0.6623
Epoch 7/20	
87/87	138s 1s/step - accuracy: 0.7879 - loss: 0.5804 - precision: 0.8497 - recall: 0.7075 - val_accuracy: 0.7580 - val_loss: 0.6468 - val_precision: 0.8304 - val_recall: 0.6884
Epoch 8/20	
87/87	97s 1s/step - accuracy: 0.7889 - loss: 0.5693 - precision: 0.8593 - recall: 0.7138 - val_accuracy: 0.7565 - val_loss: 0.6361 - val_precision: 0.8421 - val_recall: 0.6957
Epoch 9/20	
87/87	95s 1s/step - accuracy: 0.8040 - loss: 0.5469 - precision: 0.8629 - recall: 0.7346 - val_accuracy: 0.7710 - val_loss: 0.6222 - val_precision: 0.8488 - val_recall: 0.7159
Epoch 10/20	
87/87	143s 1s/step - accuracy: 0.8105 - loss: 0.5200 - precision: 0.8681 - recall: 0.7505 - val_accuracy: 0.7754 - val_loss: 0.6175 - val_precision: 0.8442 - val_recall: 0.7145
Epoch 11/20	
87/87	103s 1s/step - accuracy: 0.7996 - loss: 0.5506 - precision: 0.8560 - recall: 0.7456 - val_accuracy: 0.7696 - val_loss: 0.6384 - val_precision: 0.8342 - val_recall: 0.7072
Epoch 12/20	
87/87	96s 1s/step - accuracy: 0.8100 - loss: 0.5216 - precision: 0.8615 - recall: 0.7520 - val_accuracy: 0.7841 - val_loss: 0.5858 - val_precision: 0.8499 - val_recall: 0.7304
Epoch 13/20	
87/87	142s 1s/step - accuracy: 0.8249 - loss: 0.5156 - precision: 0.8761 - recall: 0.7615 - val_accuracy: 0.7870 - val_loss: 0.5807 - val_precision: 0.8523 - val_recall: 0.7362
Epoch 14/20	
87/87	95s 1s/step - accuracy: 0.8412 - loss: 0.4618 - precision: 0.8841 - recall: 0.7855 - val_accuracy: 0.7812 - val_loss: 0.5946 - val_precision: 0.8515 - val_recall: 0.7232
Epoch 15/20	
87/87	144s 1s/step - accuracy: 0.8137 - loss: 0.4882 - precision: 0.8703 - recall: 0.7622 - val_accuracy: 0.7884 - val_loss: 0.5835 - val_precision: 0.8424 - val_recall: 0.7435
Epoch 16/20	
87/87	105s 1s/step - accuracy: 0.8431 - loss: 0.4393 - precision: 0.8925 - recall: 0.7970 - val_accuracy: 0.8014 - val_loss: 0.5663 - val_precision: 0.8550 - val_recall: 0.7435
Epoch 17/20	
87/87	95s 1s/step - accuracy: 0.8370 - loss: 0.4641 - precision: 0.8776 - recall: 0.7847 - val_accuracy: 0.7913 - val_loss: 0.5761 - val_precision: 0.8448 - val_recall: 0.7493
Epoch 18/20	
87/87	99s 1s/step - accuracy: 0.8523 - loss: 0.4331 - precision: 0.8918 - recall: 0.8003 - val_accuracy: 0.7986 - val_loss: 0.5583 - val_precision: 0.8425 - val_recall: 0.7522
Epoch 19/20	
87/87	96s 1s/step - accuracy: 0.8457 - loss: 0.4570 - precision: 0.8860 - recall: 0.7984 - val_accuracy: 0.7942 - val_loss: 0.5590 - val_precision: 0.8524 - val_recall: 0.7449
Epoch 20/20	
87/87	

Gambar 2. Proses Pelatihan Model MobileNetV2

Proses pelatihan model MobileNetV2 pada gambar 2 menunjukkan peningkatan performa yang konsisten dari epoch ke epoch. Pada tahap awal, akurasi masih berada di kisaran 63–67%, dengan nilai loss yang relatif tinggi. Seiring berjalannya pelatihan, baik akurasi maupun metrik lain seperti precision dan recall mengalami peningkatan bertahap. Memasuki pertengahan hingga akhir epoch, model mencapai akurasi pelatihan di atas 80% dan stabil pada kisaran 83–85%, sedangkan akurasi validasi meningkat hingga sekitar 78–80%. Nilai loss validasi juga menurun signifikan, menandakan kemampuan generalisasi model yang makin baik dan gap antara training–validation yang semakin kecil. Performa metrik evaluasi seperti precision dan recall pada data validasi turut menunjukkan konsistensi, mengindikasikan bahwa MobileNetV2 berhasil belajar pola fitur dengan baik tanpa mengalami overfitting yang berat.

Berdasarkan hasil proses pelatihan dan pengujian menggunakan arsitektur MobileNetV2, grafik akurasi dan loss yang dihasilkan ditampilkan pada Gambar 3. Grafik ini memberikan

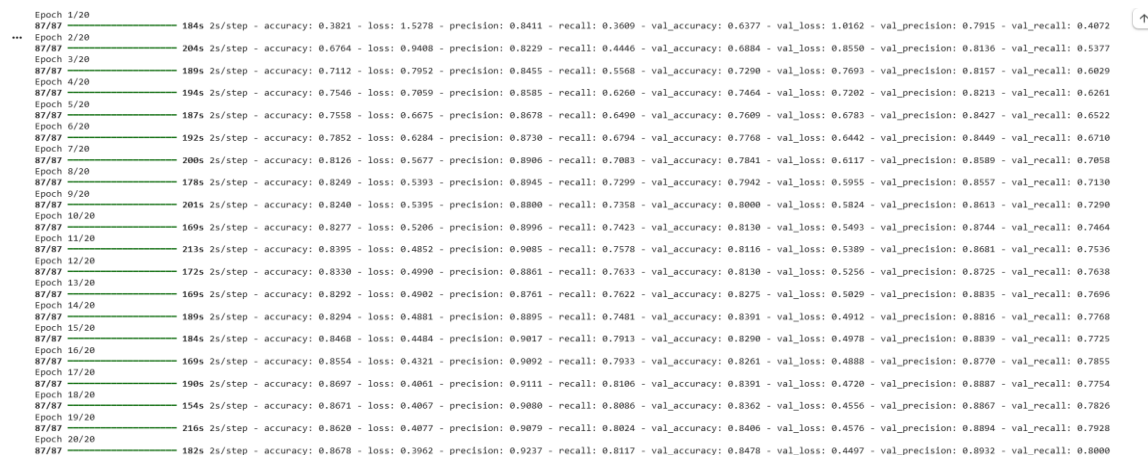
gambaran mengenai perkembangan performa model dari epoch awal hingga akhir, baik pada data pelatihan maupun data validasi.



Gambar 3. Grafik Akurasi dan Loss MobileNetV2

Grafik akurasi menunjukkan bahwa performa model MobileNetV2 meningkat secara konsisten sepanjang proses pelatihan, di mana akurasi training dan validation sama-sama naik dari sekitar 0,50 pada epoch awal hingga mendekati 0,85 pada epoch akhir. Kurva keduanya tampak relatif berdekatan, menandakan bahwa model tidak mengalami overfitting yang berarti dan mampu melakukan generalisasi dengan baik terhadap data uji. Sementara itu, grafik loss memperlihatkan tren penurunan yang stabil pada training loss maupun validation loss, dari nilai awal di atas 1,3 hingga berada di kisaran 0,4 pada akhir epoch. Kedekatan antara kurva training dan validation loss menunjukkan proses pelatihan yang efektif serta stabil, di mana model mampu mempelajari pola data tanpa kehilangan kemampuan generalisasi. Secara keseluruhan, kedua grafik ini mengonfirmasi bahwa konfigurasi MobileNetV2 berhasil memberikan performa yang solid dan seimbang antara akurasi tinggi dan nilai loss yang rendah.

Sedangkan Proses pelatihan dan pengujian pada arsitektur EfficientNetB0 dapat dilihat pada gambar 4.

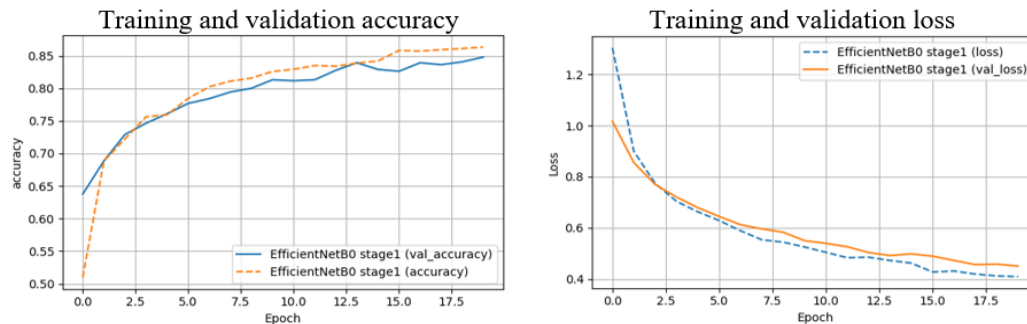


Gambar 4. Proses Pelatihan Model EfficientNetB0

Proses pelatihan EfficientNetB0 pada gambar 4 menunjukkan performa yang stabil dan meningkat secara konsisten sepanjang 20 epoch. Pada awal pelatihan, akurasi model berada di kisaran 80%, kemudian secara bertahap naik hingga mencapai sekitar 84-86% di akhir epoch, dengan penurunan loss yang cukup signifikan. Akurasi validasi juga mengikuti pola kenaikan yang stabil dan bertahan pada kisaran 83-84%, menandakan bahwa model mampu melakukan generalisasi dengan baik tanpa mengalami overfitting yang berarti. Metrik lain seperti precision

dan recall pada data pelatihan maupun validasi turut menunjukkan konsistensi dan nilai yang tinggi, mencerminkan kemampuan EfficientNetB0 dalam mengekstraksi fitur penting secara efektif. Secara keseluruhan, hasil training ini mengindikasikan bahwa konfigurasi arsitektur dan hyperparameter yang digunakan berhasil mengoptimalkan kinerja model secara seimbang antara akurasi, precision, recall, dan stabilitas loss.

Berdasarkan hasil proses pelatihan dan pengujian menggunakan arsitektur MobileNetV2, grafik akurasi dan loss yang dihasilkan ditampilkan pada gambar 4. Grafik ini memberikan gambaran mengenai perkembangan performa model dari epoch awal hingga akhir, baik pada data pelatihan maupun data validasi.

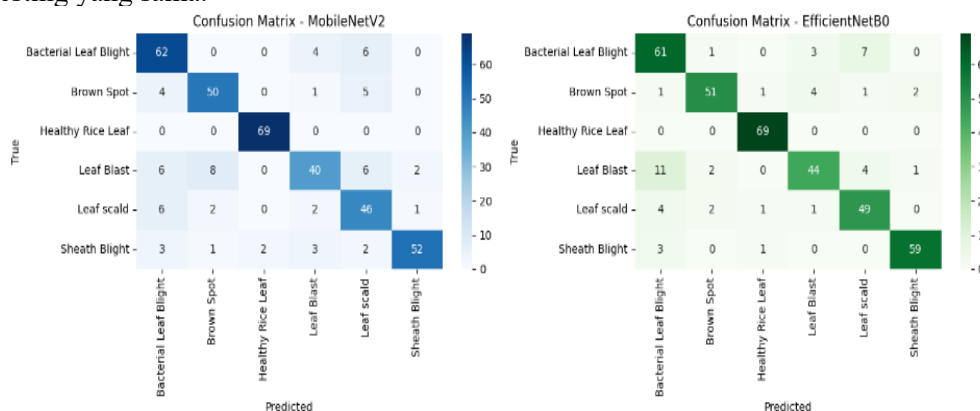


Gambar 5. Grafik Akurasi dan Loss EfficientNetB0

Grafik akurasi menunjukkan bahwa model EfficientNetB0 mengalami peningkatan performa yang stabil sepanjang proses pelatihan, di mana akurasi training dan validation sama-sama naik dari sekitar 0,55 pada epoch awal hingga mencapai kisaran 0,84–0,86 pada epoch akhir. Kedua kurva akurasi terlihat sangat berdekatan, menandakan bahwa model memiliki kemampuan generalisasi yang baik tanpa indikasi overfitting. Pada grafik loss, baik training loss maupun validation loss menurun secara konsisten dari nilai awal sekitar 1,2 hingga mendekati 0,45–0,48 di akhir pelatihan. Kedekatan kedua kurva loss ini mengonfirmasi bahwa proses optimasi berjalan efektif dan stabil, di mana model tidak hanya belajar secara mendalam tetapi juga mempertahankan keseimbangan antara data pelatihan dan pengujian. Secara keseluruhan, grafik ini menunjukkan bahwa EfficientNetB0 mampu mencapai performa yang sangat baik dengan pola pembelajaran yang halus, stabil, dan efisien.

3.2 Confusion Matrix

Untuk analisis lebih mendalam tentang performa per-kelas kedua model, confusion matrix digunakan untuk mengidentifikasi pola kesalahan klasifikasi dan kelas-kelas yang sering tertukar. Gambar 6 menyajikan perbandingan confusion matrix MobileNetV2 dan EfficientNetB0 pada data testing yang sama.



Gambar 6. Confusion Matrix MobileNetV2 dan EfficientNetB0

Secara umum, kedua model menunjukkan diagonal utama yang dominan menandakan klasifikasi yang baik pada sebagian besar kelas, dengan performa kuat khususnya pada kelas Healthy Rice Leaf dan Sheath Blight yang memiliki jumlah prediksi benar tertinggi di kedua model. MobileNetV2 mencatat prediksi benar 69 untuk Healthy Rice Leaf dan 52 untuk Sheath Blight, sementara EfficientNetB0 menunjukkan 69 untuk Healthy Rice Leaf dan meningkat menjadi 59 pada Sheath Blight, mengindikasikan keunggulan EfficientNetB0 pada kelas terakhir tersebut. Pada Bacterial Leaf Blight dan Brown Spot, kedua model relatif seimbang dengan sedikit variasi salah klasifikasi menuju kelas Leaf Blast maupun Healthy, namun nilai benar tetap mendominasi.

Perbedaan paling mencolok terlihat pada kelas Leaf Blast dan Leaf Scald, di mana EfficientNetB0 sedikit lebih konsisten, Leaf Blast mencapai 44 prediksi benar dibanding 40 pada MobileNetV2, dan Leaf Scald 49 dibanding 46, menunjukkan peningkatan reduksi kesalahan silang ke kelas-kelas berdekatan. Meskipun masih terdapat kebingungan antar kelas bercorak gejala mirip misalnya sebagian sampel Bacterial Leaf Blight dan Brown Spot yang bergeser ke Leaf Blast besarnya kesalahan relatif kecil terhadap total sampel per kelas. Temuan ini menegaskan bahwa meski keduanya kompetitif, EfficientNetB0 memberikan margin akurasi per kelas yang lebih baik pada kategori dengan kemiripan visual tinggi, khususnya Sheath Blight, Leaf Blast, dan Leaf Scald. Untuk detail lebih lanjut mengenai performa confusion matrix dapat dilihat pada tabel dibawah tabel 4.

Tabel 4. Hasil Confusion Matrix MobileNetV2

Kelas	Precision	Recall	F1-score	Support
Bacterial Leaf Blight	0.7654	0.8611	0.8105	72
Brown Spot	0.8197	0.8333	0.8264	60
Healthy Rice Leaf	0.9718	1.0000	0.9857	69
Leaf Blast	0.8000	0.6452	0.7143	62
Leaf scald	0.7077	0.8070	0.7541	57
Sheath Blight	0.9455	0.8254	0.8814	63
Accuracy			0.8329	383

Pada tabel 4 MobileNetV2, performa keseluruhan mencapai akurasi 83,29% dengan F1-score kuat pada kelas Healthy Rice Leaf (0,9857) serta Sheath Blight (0,8814). Brown Spot dan Bacterial Leaf Blight juga terklasifikasi baik dengan F1-score masing-masing 0,8264 dan 0,8105, sementara tantangan utama muncul pada Leaf Blast (F1-score 0,7143) dan Leaf Scald (F1-score 0,7541) yang menunjukkan recall cukup tinggi namun precision lebih rendah, hal ini mengindikasikan sebagian prediksi masih tertukar ke kelas serupa secara visual. Dengan total 383 sampel, model sudah stabil tetapi masih menyisakan ruang perbaikan pada dua kelas bercorak gejala mirip.

Tabel 5. Hasil Confusion Matrix EfficientNetB0

Kelas	Precision	Recall	F1-score	Support
Bacterial Leaf Blight	0.7625	0.8472	0.8026	72
Brown Spot	0.9107	0.8500	0.8793	60
Healthy Rice Leaf	0.9583	1.0000	0.9787	69
Leaf Blast	0.8462	0.7097	0.7719	62
Leaf scald	0.8033	0.8596	0.8305	57
Sheath Blight	0.9516	0.9365	0.9440	63
Accuracy			0.8695	383

Pada tabel 5 EfficientNetB0 memberikan peningkatan akurasi menjadi 86,95% dan perbaikan merata pada hampir semua kelas, terutama Brown Spot (precision 0,9107, F1-score 0,8793) serta Leaf Scald (recall 0,8596, F1-score 0,8305). Kinerja terbaik tetap pada Healthy Rice Leaf dengan recall sempurna (1,0000 dan F1-score 0,9787), diikuti Sheath Blight (F1-score 0,9440) yang menunjukkan ketangguhan model membedakan gejala pada pelepah. Meski Leaf Blast masih menjadi kelas tersulit, F1-score meningkat ke 0,7719 berkat precision yang lebih tinggi dibanding MobileNetV2. Secara praktis, hasil ini menegaskan bahwa EfficientNetB0 lebih andal untuk operasional lapangan karena kesalahan antar kelas berkurang dan konsistensi metrik lebih baik di seluruh kategori.

Secara keseluruhan performa kedua model dirangkum dalam Tabel 6 yang menyajikan metrik precision, recall, dan F1-score rata-rata dari keseluruhan kelas.

Tabel 6. Perbandingan Performa MobileNetV2 dan EfficientNetB0

Model	Precision	Recall	F1-Score
MobileNetV2	0.832898	0.872024	0.828726
EfficientNetB0	0.869452	0.900293	0.867851

Tabel 6 merangkum perbandingan performa dua arsitektur transfer learning, MobileNetV2 dan EfficientNetB0, pada tugas klasifikasi enam kelas penyakit daun padi. Keunggulan EfficientNetB0 dibanding MobileNetV2 dalam klasifikasi penyakit daun padi dapat dikaitkan dengan kapasitas ekstraksi fitur yang lebih seimbang dan kaya pada backbone-nya. EfficientNetB0 dirancang dengan prinsip compound scaling yang secara simultan meningkatkan kedalaman, lebar kanal, dan resolusi representasi sehingga jaringan memperoleh ruang yang cukup untuk memodelkan detail tekstur bercak, bentuk lesi, dan variasi warna halus yang membedakan kelas-kelas penyakit bergejala mirip seperti Leaf Blast dan Leaf Scald, tanpa menyebabkan model menjadi terlalu besar, selisih akurasi 3–4% yang diperoleh menunjukkan bahwa desain arsitektur EfficientNetB0 menghasilkan representasi fitur yang lebih kaya dan lebih mudah digeneralisasi dibandingkan MobileNetV2, yang lebih agresif mengompresi kanal demi efisiensi komputasi.

Sistem klasifikasi penyakit daun padi berbasis MobileNetV2 dan EfficientNetB0 ini berpotensi besar diimplementasikan dalam ekosistem smart farming, khususnya sebagai aplikasi mobile atau platform IoT yang dapat diakses langsung oleh petani, di mana petani cukup memotret daun padi di lahan, kemudian sistem secara otomatis mengidentifikasi jenis penyakit dan menampilkan rekomendasi tindakan pengendalian.

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa transfer learning dengan lightweight CNN efektif untuk mengklasifikasikan enam kelas penyakit daun padi dan dapat diimplementasikan pada perangkat terbatas sumber daya. EfficientNetB0 muncul sebagai model paling tepat untuk deteksi dini karena memberikan kinerja paling stabil pada kelas-kelas bergejala mirip, sementara MobileNetV2 layak dipilih pada skenario dengan keterbatasan memori yang ketat. Implikasi praktisnya, kedua model dapat mendasari pengembangan aplikasi diagnosis penyakit padi berbasis mobile yang mudah diakses petani. Untuk meningkatkan generalisasi dan mengurangi kebingungan antar kelas. Saran untuk penelitian berikutnya, melakukan fine-tuning selektif pada backbone serta memperluas dan menyeimbangkan dataset pada berbagai varietas dan kondisi lingkungan yang berbeda untuk memastikan generalisasi model yang lebih stabil pada skenario nyata di lapangan.

DAFTAR PUSTAKA

- [1] R. K. Rita Rahim, Ani Dela, Rosiana Nurfalah, Yulia Anggraeni, Sanjiwan Pasaribu, Nabila Dwi Utami, "Dinamika Ketahanan Pangan: Analisis Pengaruh Luas Panen Padi, Konsumsi Beras, Harga Beras, dan Jumlah Penduduk Terhadap Produksi Padi di Wilayah Sentra Padi di Indonesia Tahun 2017-2021," *Innov. J. Soc. Sci. Res.*, vol. 4, no. 3, pp. 17083–17093, 2024, [Online]. Available: <https://j-innovative.org/index.php/Innovative/article/view/12524>
- [2] J. Algoritme *et al.*, "Penerapan Algoritma K-Nearest Neighbor Dalam Klasifikasi Penyakit Daun Padi Menggunakan Ekstraksi HOG 1,2,3," vol. 5, no. 3, 2025, doi: 10.35957/algoritme.v6i1.12154.
- [3] A. Y. Ani Widart, Anik Kurniat, "Prakiraan Serangan Opt Utama Padi, Jagung, Kedelai, Aneka Kacang Dan Umbi Di Indonesia MT 2024/2025," 2024, [Online]. Available: https://bdsp2.pertanian.go.id/siperditan/rekomenpdf/opttp/BUKU_PRAKIRAAN_OPT_PANGAN_MT_2024-2025_FINAL.pdf
- [4] T. Febriyanto and S. Syofian, "Implementasi Deep Learning Menggunakan Vision Transformer Untuk Klasifikasi Penyakit Daun Padi," *J. TIFDA (Technology Inf. Data Anal.*, vol. 1, no. 2, pp. 34–39, Dec. 2024, doi: 10.70491/tifda.v1i2.47.
- [5] M. F. A. Maulana, N. M. Anggadimas, and D. A. Sani, "Klasifikasi Citra Penyakit Daun Padi Dengan Metode CNN Menggunakan Arsitektur ResNet50V2," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 10, no. 2, pp. 517–529, Jul. 2025, doi: 10.24114/cess.v10i2.66960.
- [6] N. Farihah, "Deteksi Dini Penyakit Daun Padi Menggunakan Integrasi Glcm Dan KNN," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, Jul. 2025, doi: 10.23960/jitet.v13i3.6543.
- [7] M. Khoiruddin, A. Junaidi, and W. A. Saputra, "Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network," *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 2, no. 1, pp. 37–45, Feb. 2022, doi: 10.20895/dinda.v2i1.341.
- [8] H. M. Yusuf, S. A. Yusuf, A. H. Abubakar, M. Abdullahi, and I. H. Hassan, "A systematic review of deep learning techniques for rice disease recognition: Current trends and future directions," *Franklin Open*, vol. 8, p. 100154, Sep. 2024, doi: 10.1016/j.fraope.2024.100154.
- [9] E. Ramadhan, S. Akbar, M. F. Al-farizi, T. Agustin, and P. Informatika, "Klasifikasi Tingkat Keparahan Penyakit Leaf Blast Pada Tanaman Padi Menggunakan Efficientnetb0 Menggunakan," no. November, pp. 429–440, 2024.
- [10] 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*, vol. 10, no. 1, pp. 56–62, Jul. 2024, doi: 10.54914/jtt.v10i1.1224.
- [11] I. F. Annur, J. Umami, M. N. Annafii, N. Trisnaningrum, and O. V. Putra, "Klasifikasi Tingkat Keparahan Penyakit Leafblast Tanaman Padi Menggunakan MobileNetv2," *Fountain Informatics J.*, vol. 8, no. 1, pp. 7–14, May 2023, doi: 10.21111/fij.v8i1.9419.
- [12] E. Anggiratih, S. Siswanti, S. K. Octaviani, and A. Sari, "Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 dengan Transfer Learning," *J. Ilm. SINUS*, vol. 19, no. 1, p. 75, Jan. 2021, doi: 10.30646/sinus.v19i1.526.
- [13] T. Turahman, E. Hasmin, and K. Aryasa, "Analisis Perbandingan Metode Convolutional Neural Network (CNN) dan MobileNet dalam Klasifikasi Penyakit Daun Padi," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 9, no. 1, pp. 368–377, Nov. 2024, doi: 10.35870/jtik.v9i1.3218.
- [14] A. C. Milano, "Klasifikasi Penyakit Daun Padi Menggunakan Model Deep Learning Efficientnet-B6," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3855.

-
- [15] T. Daniya and S. Vigneshwari, "Rice Plant Leaf Disease Detection and Classification Using Optimization Enabled Deep Learning," *J. Environ. Informatics*, 2023, doi: 10.3808/jei.202300492.
 - [16] C. G. Simhadri, H. K. Kondaveeti, V. K. Vatsavayi, A. Mitra, and P. Ananthachari, "Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques," *Inf. Process. Agric.*, vol. 12, no. 2, pp. 151–168, Jun. 2025, doi: 10.1016/j.inpa.2024.04.006.
 - [17] Q. N. Azizah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet," *sudo J. Tek. Inform.*, vol. 2, no. 1, pp. 28–33, Feb. 2023, doi: 10.56211/sudo.v2i1.227.
 - [18] S. Sathyanarayanan, "Confusion Matrix-Based Performance Evaluation Metrics," *African J. Biomed. Res.*, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.
 - [19] M. T. Roseno, S. Oktarina, Y. Nearti, H. Syaputra, and N. Jayanti, "Comparing CNN Models for Rice Disease Detection: ResNet50, VGG16, and MobileNetV3-Small," *J. Inf. Syst. Informatics*, vol. 6, no. 3, pp. 2099–2109, Sep. 2024, doi: 10.51519/journalisi.v6i3.865.
 - [20] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 1, pp. 58–65, Feb. 2022, doi: 10.29100/jipi.v7i1.2410.