

# Evaluasi Komparatif Lightweight Convolutional Neural Network Untuk Klasifikasi Penyakit Daun dan Hama Tanaman Padi

Afis Julianto\*, Miftahul Jannah

Program Studi Teknik Informatika, Politeknik Negeri Bengkalis, Bengkalis, Indonesia

Email: <sup>1,\*</sup>afisjulianto@polbeng.ac.id, <sup>2</sup>miftahuljannah@polbeng.ac.id

Email Penulis Korespondensi: afisjulianto@polbeng.ac.id

Submitted: 13/11/2025; Accepted: 16/12/2025; Published: 26/12/2025

**Abstrak**—Tanaman padi merupakan komoditas penting bagi ketahanan pangan nasional namun produktivitasnya sering menurun akibat serangan penyakit daun dan hama. Metode identifikasi konvensional berbasis pengamatan visual dinilai kurang efisien dan rentan terhadap subjektivitas terutama karena variasi gejala yang kompleks. Penelitian ini bertujuan mengevaluasi dan membandingkan performa beberapa arsitektur CNN *lightweight* dalam mendeteksi penyakit dan hama padi secara akurat dan efisien pada perangkat dengan sumber daya terbatas. Empat arsitektur CNN *lightweight* dibandingkan yaitu MobileNetV2, EfficientNetV2B3, NasNetMobile dan Custom CNN *Lightweight* menggunakan dataset 13 kelas yang telah melalui *preprocessing*, augmentasi dan penyeimbangan data. Model dilatih selama 100 epoch dengan optimizer Adam. Hasil pengujian menunjukkan bahwa EfficientNetV2B3 memberikan performa terbaik dengan akurasi, *presisi*, *recall*, dan *F1-score* sebesar 97%, diikuti MobileNetV2 dan NasNetMobile dengan akurasi 94%. Custom CNN *Lightweight* menghasilkan akurasi 91% dengan ukuran model hanya 0,53 MB. Penelitian ini memberikan rekomendasi model CNN *Lightweight* yang akurat dan efisien untuk mendukung deteksi penyakit dan hama padi pada perangkat mobile, sistem IoT dan perangkat *edge*.

**Kata Kunci:** Klasifikasi; CNN Lightweight; Penyakit Daun dan Hama Padi; EfficientNetV2B3

**Abstract**—Rice is a critical commodity for national food security; however, its productivity is frequently reduced due to leaf diseases and pests. Conventional identification methods that rely on visual observation are often inefficient and prone to subjectivity, particularly given the complex and variable nature of symptoms. This study to evaluate and compare the performance of several lightweight CNN architectures in accurately and efficiently detecting rice leaf diseases and pests on resource constrained devices. This study compares four CNN lightweight architectures: MobileNetV2, EfficientNetV2-B3, NasNetMobile, and a custom CNN Lightweight Architecture, all using a 13-class dataset that underwent preprocessing, augmentation, and data balancing. The models were trained for 100 epochs using the Adam optimizer. Experimental results show that EfficientNetV2B3 achieved the best performance, with 97% accuracy, precision, recall, and F1-score, followed by MobileNetV2 and NasNetMobile, which achieved 94% accuracy. The Custom CNN lightweight model produced 91% accuracy with a model size of only 0.53 MB. Overall, this study provides recommendations for developing accurate and efficient lightweight CNN models to support rice disease and pest detection on mobile devices, IoT systems, and edge computing platforms.

**Keywords:** Classification; CNN Lightweight; Rice Leaf Diseases and Pests; EfficientNetV2B3

## 1. PENDAHULUAN

Tanaman padi merupakan salah satu komoditas pangan utama yang memiliki peran strategis dalam menjaga ketahanan pangan nasional. Namun produksi padi di Indonesia masih menghadapi tantangan serius berupa serangan berbagai jenis penyakit daun dan hama yang dapat menurunkan hasil panen secara signifikan [1][2]. Penyakit pada daun seperti *blast*, *brown spot*, *tungro*, busuk pelepah serta hama seperti wereng coklat dan penggerek batang diketahui mampu merusak tanaman padi pada fase pertumbuhan yang berdampak langsung pada produktivitas dan kesejahteraan petani [3][4]. Deteksi dini dan klasifikasi yang akurat terhadap penyakit dan hama padi menjadi sangat penting agar tindakan pengendalian dapat dilakukan secara cepat dan tepat sasaran.

Saat ini proses identifikasi penyakit dan hama padi secara konvensional masih bergantung pada pengamatan visual oleh manusia yang bersifat subjektif memerlukan keahlian khusus dan tidak efisien jika diterapkan dalam skala luas dan keterbatasan tenaga ahli serta kondisi lapangan yang bervariasi [5][6]. Di sisi lain tingginya variasi visual dari gejala penyakit dan hama pada daun dari segi warna, bentuk, maupun intensitas serangan menimbulkan tantangan tersendiri dalam proses diagnosis [7]. Oleh karena itu dibutuhkan sistem berbasis teknologi pengenalan citra yang mampu melakukan identifikasi secara cepat dan akurat.

Dalam beberapa tahun terakhir kemajuan teknologi kecerdasan buatan telah membawa perubahan signifikan pada bidang pertanian khususnya dalam penerapan metode pengolahan citra untuk pemantauan dan diagnostik tanaman. Salah satu metode yang banyak digunakan adalah *convolutional neural network* (CNN) dirancang khusus untuk mengelola data citra dua dimensi dan memiliki kemampuan tinggi dalam mengenali pola visual yang rumit dan beragam [8]. CNN telah banyak diimplementasikan dalam berbagai bidang pertanian seperti klasifikasi tanaman [9], [10], deteksi penyakit [11], [12] dan identifikasi hama [13]. Keunggulan CNN terletak pada kemampuannya dalam melakukan ekstraksi fitur secara otomatis dari gambar tanpa memerlukan intervensi manusia secara langsung sehingga meningkatkan akurasi dan efisiensi proses klasifikasi [14].

Penelitian yang dilakukan oleh Maulana, et al (2025) [15] menerapkan arsitektur ResNet50V2 untuk mengklasifikasikan delapan jenis penyakit daun padi menggunakan dataset 15.241 citra yang telah diolah dan mencapai akurasi tertinggi 94,14% membuktikan bahwa CNN dengan arsitektur besar mampu mendeteksi penyakit dengan akurat. Ritharson et al (2024) [16] mengembangkan metode deteksi penyakit daun padi berbasis *deep learning*

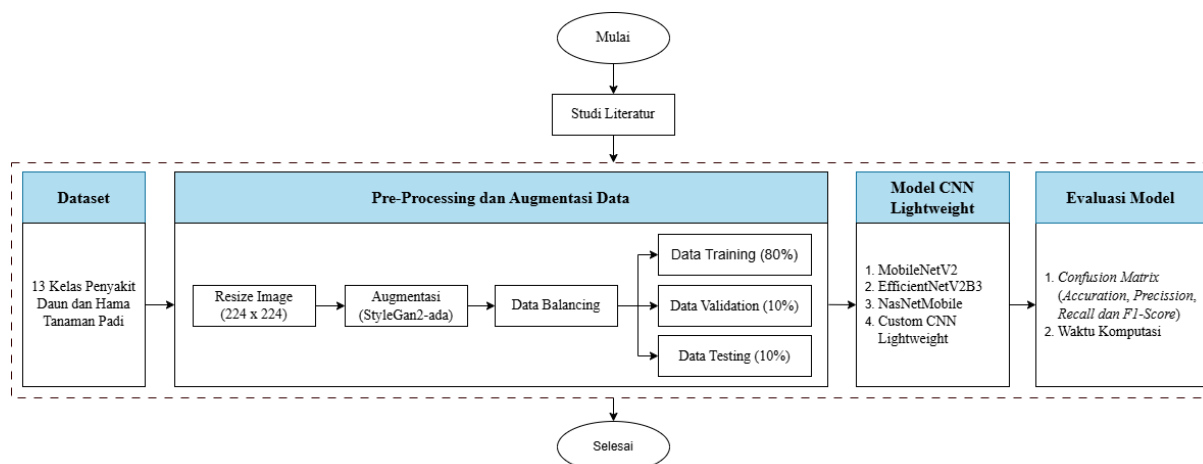
menggunakan 5.932 citra dalam sembilan kelas penyakit, menerapkan augmentasi data dan membandingkan model CNN khusus dengan arsitektur populer seperti VGG16, ResNet50 dan InceptionV3, di mana model custom VGG16 menghasilkan akurasi tertinggi 99,94% dengan ukuran model 59,5 MB. Namun tingginya kebutuhan komputasi dari arsitektur CNN seperti ResNet, VGG dan Inception menjadi hambatan utama dalam implementasi pada perangkat mobile atau sistem edge khususnya yang digunakan oleh petani di wilayah pedesaan dengan keterbatasan sumber daya [17]. Sebagai upaya mengatasi keterbatasan tersebut berbagai arsitektur CNN *lightweight* seperti MobileNet, EfficientNet dan NasNet telah dikembangkan dengan tujuan mengurangi jumlah parameter serta konsumsi daya komputasi tanpa mengalami penurunan akurasi yang signifikan.

Penelitian Sitorus et al (2025) [18] melakukan evaluasi komparatif antara MobileNetV2 dan EfficientNetB0 untuk deteksi penyakit daun jagung menemukan bahwa keduanya mampu memberikan efisiensi komputasi yang jauh lebih baik dibanding model konvensional dengan performa klasifikasi tetap kompetitif menghasilkan akurasi tertinggi 91,73%. Penelitian lain yang dilakukan Yuliany et al (2022) [19] berhasil menerapkan model CNN ringan dengan strategi augmentasi data dalam klasifikasi hama padi dan memperoleh hasil klasifikasi tertinggi 77,33%. Penelitian Pal et al (2025) [20] mengembangkan arsitektur Mob-Res gabungan *residual learning* dan *MobileNetV2* yang ringan dengan 3,51 juta parameter mampu mencapai akurasi hingga 99,47% dan adaptif lintas domain menjadikannya solusi efisien untuk deteksi penyakit tanaman berbasis perangkat mobile. Sebagian penelitian terdahulu masih berfokus pada penggunaan satu jenis arsitektur *CNN lightweight* dan terbatas pada klasifikasi penyakit tanaman tanpa mengintegrasikan deteksi hama secara bersamaan.

Penelitian ini bertujuan untuk menguji dan membandingkan performa empat arsitektur *CNN lightweight* dalam klasifikasi penyakit daun dan hama pada tanaman padi menggunakan *confusion matrix* seperti akurasi, waktu inferensi dan ukuran model. Hasil yang diperoleh diharapkan dapat menghasilkan rekomendasi model optimal yang efisien serta dapat diimplementasikan secara langsung pada perangkat mobile atau sistem pertanian digital dengan keterbatasan sumber daya. Penelitian ini tidak hanya berkontribusi terhadap pengembangan teknologi klasifikasi citra di bidang pertanian tetapi juga mendukung transformasi pertanian konvensional menuju sistem pertanian digital yang lebih cerdas, efisien dan berkelanjutan.

## 2. METODOLOGI PENELITIAN

Tahapan penelitian merupakan rangkaian langkah yang dilakukan secara sistematis dan terstruktur untuk mencapai tujuan penelitian. Gambar 1 menunjukkan tahapan penelitian yang meliputi studi literatur, pengumpulan dataset, preprocessing dan augmentasi data, pelatihan model, serta evaluasi model.



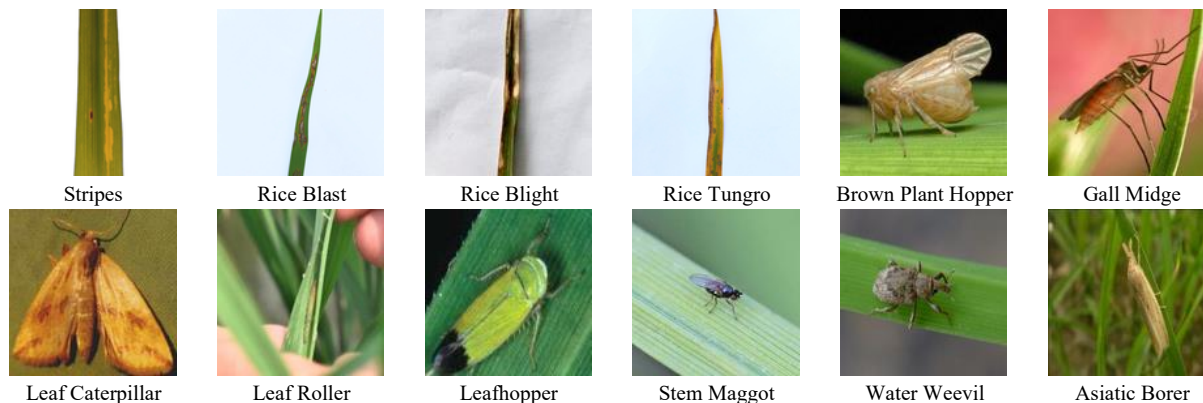
**Gambar 1.** Tahapan Penelitian

Gambar 1 menggambarkan tahapan penelitian yang dilakukan untuk membandingkan berbagai model CNN *Lightweight* dalam klasifikasi penyakit daun dan hama tanaman padi. Penelitian ini terdiri dari beberapa langkah yang dilaksanakan secara sistematis dan terstruktur. Tahapan dimulai dengan studi literatur untuk menentukan pendekatan dan metode yang relevan, kemudian dilanjutkan dengan pengumpulan dataset yang mencakup 13 kelas citra penyakit daun dan hama tanaman padi. Selanjutnya dilakukan preprocessing dan augmentasi data guna meningkatkan kualitas serta variasi citra. Proses berikutnya adalah pelatihan model menggunakan beberapa arsitektur CNN *Lightweight* seperti MobileNetV2, EfficientNetV2B3, NasNetMobile dan Custom CNN *Lightweight*. Tahapan selanjutnya yaitu evaluasi performa model melalui analisis confusion matrix yang meliputi pengukuran akurasi, presisi, recall dan F1-score, serta perhitungan waktu komputasi untuk menentukan model dengan kinerja paling optimal.

### 2.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan hasil penggabungan dari dua sumber dataset yang berbeda namun saling melengkapi dalam konteks identifikasi gangguan pada tanaman padi yang mencakup penyakit daun dan

serangan hama. Dataset pertama diperoleh dari Kaggle [21] yang berisi citra daun padi dengan berbagai jenis penyakit, sedangkan dataset kedua bersumber dari Zenodo [22] yang memuat citra hama tanaman padi seperti yang di tampilkan pada Gambar 2. Kedua dataset tersebut digabungkan untuk memperluas cakupan model klasifikasi agar mampu mengenali baik penyakit maupun serangan hama secara bersamaan. Terdiri dari 13 kelas dengan total gambar 3785.



Gambar 2. Citra penyakit daun dan hama tanaman padi

## 2.2 Preprocessing dan Augmentasi Data

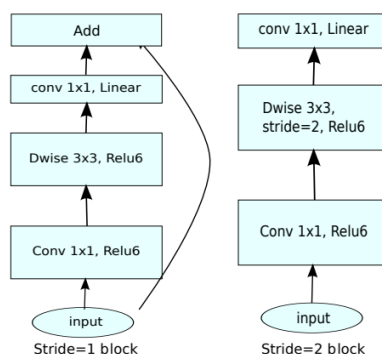
Pada tahap ini dilakukan serangkaian proses untuk menyiapkan dataset sebelum digunakan dalam pelatihan model. Proses dimulai dengan resize citra ke ukuran  $224 \times 224$  piksel agar sesuai dengan standar input pada model CNN *lightweight* yang digunakan. Penyeragaman ukuran ini penting untuk memastikan konsistensi dimensi antar sampel dan mengoptimalkan efisiensi proses komputasi. Selanjutnya dilakukan proses augmentasi data menggunakan teknik StyleGAN2-ADA (*Adaptive Discriminator Augmentation*). Teknik ini digunakan untuk menghasilkan citra sintesis yang realistis dengan mempertahankan karakteristik visual dari citra asli. Augmentasi berbasis GAN ini bertujuan untuk memperluas variasi data tanpa mengubah label kelas, sehingga meningkatkan kemampuan model dalam melakukan generalisasi serta mengurangi risiko *overfitting* pada data pelatihan.

## 2.3 Model CNN *Lightweight*

Pada tahapan ini dilakukan proses pengujian model dengan mengimplementasikan berbagai arsitektur CNN *lightweight* dalam klasifikasi citra penyakit daun dan hama tanaman padi. Beberapa arsitektur yang diterapkan meliputi MobileNetV2, EfficientNetV2B3, NasNetMobile dan Custom CNN *lightweight* selanjutnya dibandingkan untuk menilai kinerja dan performa masing-masing model. Secara keseluruhan arsitektur CNN *lightweight* ini dirancang dengan bobot model yang ringan namun mampu menghasilkan akurasi tinggi sehingga dapat mengekstraksi fitur citra secara efisien dan melakukan klasifikasi secara optimal. Pendekatan ini menjadi solusi dalam penelitian klasifikasi citra pertanian yang membutuhkan model yang memiliki performa tinggi dengan efisiensi komputasi yang baik.

### a. MobileNetV2

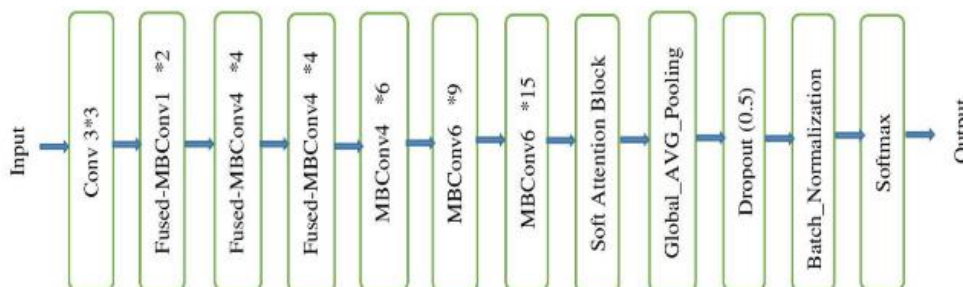
MobileNetV2 adalah arsitektur yang dirancang untuk mengurangi *complexity cost* dan ukuran model dalam *neural network* [23]. MobileNetV2 sangat cocok digunakan pada perangkat dengan daya komputasi rendah seperti *smartphone* atau *raspberrypi*. Arsitektur ini terdiri dari dua jenis blok utama yaitu *residual block* yang berfungsi mempertahankan informasi fitur dan *downsampling block* yang digunakan untuk mengurangi dimensi spasial citra guna efisiensi komputasi. Secara keseluruhan MobileNetV2 terdiri dari 32 filter lapisan konvolusi awal yang terhubung penuh dan diikuti oleh 19 lapisan *bottleneck residual* yang membentuk inti dari arsitekturnya [24]. Arsitektur MobileNetV2 dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur MobileNetV2

b. EfficientNetV2B3

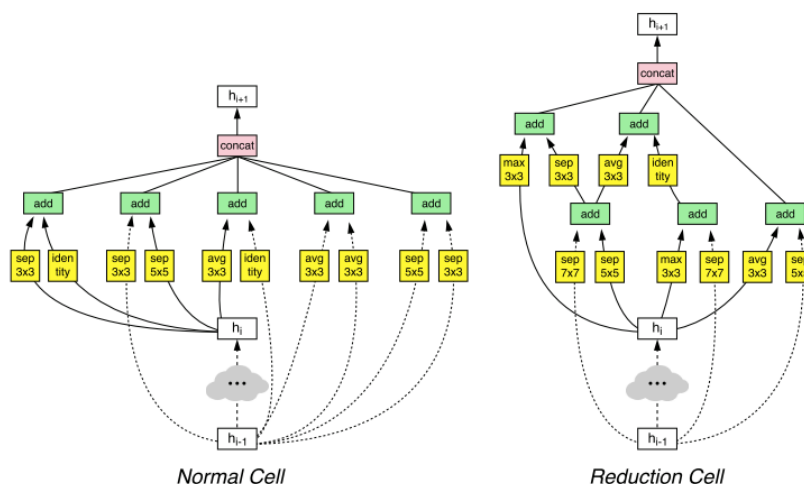
EfficientNetV2 adalah arsitektur CNN yang terkenal dengan efisiensi parameter dan kecepatan pelatihan yang tinggi [25]. EfficientNetV2B3 merupakan versi pengembangan dari EfficientNet yang dirancang untuk memberikan keseimbangan antara akurasi tinggi dan efisiensi komputasi. Model ini menggunakan *compound scaling* yang mengoptimalkan kedalaman, lebar dan resolusi input secara bersamaan yang memungkinkan penggunaan sumber daya yang lebih efisien. EfficientNetV2B3 mengadopsi teknik *Fused-MBConv blocks* yang mempercepat proses pelatihan dan inferensi [26]. Arsitektur EfficientNetV2B3 dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur EfficientNetV2B3

c. NasNetMobile

NASNetMobile merupakan varian ringan dari arsitektur *Neural Architecture Search Network* (NASNet) yang dikembangkan melalui pendekatan *Neural Architecture Search* (NAS), yaitu metode otomatis untuk menemukan arsitektur jaringan saraf terbaik berdasarkan proses pencarian pada dataset berukuran kecil sebelum diterapkan pada dataset besar seperti ImageNet [27]. Arsitektur ini terdiri dua komponen utama *normal cell* dan *reduction cell* yang disusun secara berulang dalam urutan optimal untuk menghasilkan efisiensi parameter sekaligus menjaga tingkat akurasi yang tinggi. NASNetMobile memiliki 765 lapisan yang tersusun dari 12 *Normal Cell* dan 3 *Reduction Cell* mampu mencapai keseimbangan antara efisiensi komputasi dan performa klasifikasi sehingga menjadi pilihan ideal untuk implementasi pada perangkat dengan sumber daya terbatas namun tetap menuntut akurasi tinggi seperti aplikasi mobile vision dan *edge computing* [28]. Arsitektur NasNetMobile dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur NasNetMobile

d. Custom CNN Lightweight

Arsitektur custom CNN *lightweight* dari rangkaian lapisan ekstraksi fitur dan klasifikasi yang tersusun secara bertahap untuk menghasilkan performa identifikasi yang optimal. Proses dimulai dengan lapisan Conv2D berisi 16 filter dengan keluaran  $224 \times 224 \times 16$  yang dilanjutkan dengan *BatchNormalization* dan *MaxPooling2D* sehingga ukuran fitur berkurang menjadi  $112 \times 112 \times 16$ . Tahapan berikutnya menerapkan Conv2D dengan 32 filter, *BatchNormalization* dan *MaxPooling2D* hingga menghasilkan keluaran  $56 \times 56 \times 32$ . Pada blok ketiga jumlah filter ditingkatkan menjadi 64 dan kembali melalui proses normalisasi serta *MaxPooling2D* yang mereduksi fitur menjadi  $28 \times 28 \times 64$ . Blok selanjutnya menggunakan Conv2D dengan 128 filter serta *BatchNormalization* dan *MaxPooling2D* hingga menghasilkan fitur  $14 \times 14 \times 128$ . Untuk meningkatkan efisiensi komputasi digunakan *SeparableConv2D* dengan 128 filter yang diikuti *BatchNormalization* dan *MaxPooling2D* terakhir sehingga fitur menjadi  $7 \times 7 \times 128$ . Tahap akhir ekstraksi fitur dilakukan melalui *GlobalAveragePooling2D* yang menghasilkan representasi vektor berukuran 128. Pada tahap klasifikasi model memanfaatkan *Dense layer* dengan 128 unit yang dilengkapi *dropout* untuk mencegah *overfitting* dan ditutup dengan Dense layer berisi 13 unit *softmax* sebagai

lapisan keluaran untuk menentukan probabilitas setiap kelas. Arsitektur ini secara keseluruhan dirancang untuk menyeimbangkan kemampuan ekstraksi fitur yang kuat dengan efisiensi komputasi melalui kombinasi konvolusi standar dan separable *convolution*. Arsitektur custom CNN *lightweight* dapat dilihat pada Tabel 1.

**Tabel 1.** Arsitektur Custom CNN Lightweight

Layer	Output Shape	Param
conv2d (Conv2D)	(None, 224, 224, 16)	448
batch_normalization (BatchNormalization)	(None, 224, 224, 16)	64
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_1 (Conv2D)	(None, 112, 112, 32)	4,640
batch_normalization_1 (BatchNormalization)	(None, 112, 112, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	18,496
batch_normalization_2 (BatchNormalization)	(None, 56, 56, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	73,856
batch_normalization_3 (BatchNormalization)	(None, 28, 28, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
separable_conv2d (SeparableConv2D)	(None, 14, 14, 128)	17,664
batch_normalization_4 (BatchNormalization)	(None, 14, 14, 128)	512
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 128)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 128)	0
dense (Dense)	(None, 128)	16,512
dropout (Dropout)	(None, 128)	0

## 2.4 Evaluasi Model

Evaluasi pada penelitian ini menggunakan indikator akurasi, presisi, recall, dan F1-score untuk menilai kinerja model klasifikasi multikelas. Confusion matrix adalah alat evaluasi yang digunakan untuk mengukur performa model klasifikasi, dengan menampilkan jumlah prediksi benar dan salah yang dibagi berdasarkan kelas aktual dan kelas prediksi. Untuk kasus dengan *multiclass confusion matrix* berbentuk matriks  $n \times n$  di mana  $n$  adalah jumlah kelas. Elemen matriks  $x_{ij}$  merepresentasikan jumlah sampel dari kelas aktual  $i$  yang diprediksi sebagai kelas  $j$ . Untuk menghitung akurasi (A), *presisi* (P), *recall* (R), dan nilai *f-measure* (F1) untuk setiap kelas  $i$  yang dihitung menggunakan persamaan (1)-(4) [29]:

$$A_i = \frac{TTP_{all}}{(Total\ Number\ of\ Entries)} \quad (1)$$

$$P_i = \frac{(TTP_{all})}{(TTP_{all} + TF_{Pi})} \quad (2)$$

$$R_i = \frac{(TTP_{all})}{(TTP_{all} + TF_{Ni})} \quad (3)$$

$$F_i = \frac{2(P_i)(R_i)}{P_i + R_i} \quad (4)$$

Evaluasi performa model CNN *lightweight* pada penelitian ini tidak hanya didasarkan pada hasil *confusion matrix*, tetapi juga mempertimbangkan efisiensi komputasi yang mencakup waktu pelatihan, ukuran model dan jumlah parameter. Model dengan jumlah parameter yang lebih sedikit umumnya memiliki waktu komputasi yang lebih cepat dan ukuran file yang lebih kecil sehingga lebih efisien untuk diterapkan pada perangkat dengan sumber daya terbatas.

## 3. HASIL DAN PEMBAHASAN

Dalam penelitian ini eksperimen dilakukan dengan menerapkan *batch size* 64 pada proses pelatihan dan pengujian model. Adam [30] dengan *learning rate* sebesar 0,0001 digunakan untuk meminimalkan fungsi loss pada model CNN *Lightweight* selama proses pelatihan berlangsung. Seluruh eksperimen dalam penelitian ini dijalankan menggunakan TensorFlow pada platform Google Colaboratory Pro+ sebagai penyedia *cloud computing* yang memiliki spesifikasi RAM 16 GB dan GPU Nvidia T4.

### 3.1 Dataset Penelitian

Tahap ini dataset dilakukan balancing yaitu proses penyeimbangan jumlah sampel pada setiap kelas agar distribusi data menjadi proporsional. Hal ini dilakukan untuk menghindari bias model terhadap kelas mayoritas dan meningkatkan performa klasifikasi pada kelas minoritas. Data hasil balancing kemudian dibagi menjadi data *training*

80%, *validation* 10% dan *testing* 10%. Secara rinci dataset yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2. Dataset penelitian ini berjumlah 19.500 citra yang terbagi ke dalam 13 kelas penyakit dan hama padi. Data berasal dari 3.785 gambar asli yang kemudian diseimbangkan menjadi 1.500 citra per kelas melalui augmentasi StyleGAN2-ADA, sehingga menghasilkan variasi visual yang lebih luas untuk mendukung performa model.

**Tabel 2.** Dataset penyakit dan hama daun padi dengan augmentasi stylegan2-ada

Nama Label	Kategori	Gambar Asli	Augmentsi StyleGan2-ada	Total Gambar	Pembagian Dataset		
					Train (80%)	Val (10%)	Test (10%)
Healthy	Daun Sehat	600	900	1500	1200	150	150
Blast	Penyakit	80	1420	1500	1200	150	150
Blight	Penyakit	80	1420	1500	1200	150	150
Tungro	Penyakit	80	1420	1500	1200	150	150
Stripes	Penyakit	265	1235	1500	1200	150	150
Asiatic Borer	Hama	500	1000	1500	1200	150	150
Brown Plant Hopper	Hama	345	1155	1500	1200	150	150
Gall Midge	Hama	217	1283	1500	1200	150	150
Leaf Caterpillar	Hama	153	1347	1500	1200	150	150
Leaf Roller	Hama	716	784	1500	1200	150	150
Leafhopper	Hama	244	1256	1500	1200	150	150
Stem Maggot	Hama	90	1410	1500	1200	150	150
Water Weevil	Hama	415	1085	1500	1200	150	150

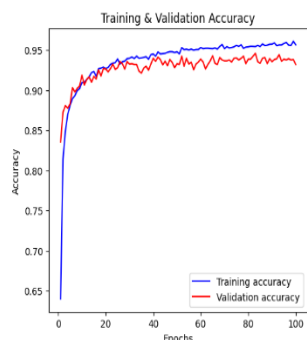
### 3.2 Hasil Pelatihan Model CNN Lightweight

Pengujian dilakukan pada data *training* dan data *validation* menggunakan empat arsitektur CNN yaitu MobileNetV2, NASNetMobile, EfficientNetV2B3 dan Custom CNN Lightweight. Tujuan dari eksperimen ini adalah untuk memperoleh informasi mengenai tingkat akurasi, nilai loss, serta waktu komputasi yang diperlukan oleh masing-masing arsitektur selama proses pelatihan data. Seluruh eksperimen dijalankan dengan jumlah 100 epoch. Hasil perbandingan akurasi, loss, dan waktu komputasi dari setiap arsitektur CNN tersebut dapat dilihat pada Tabel 3.

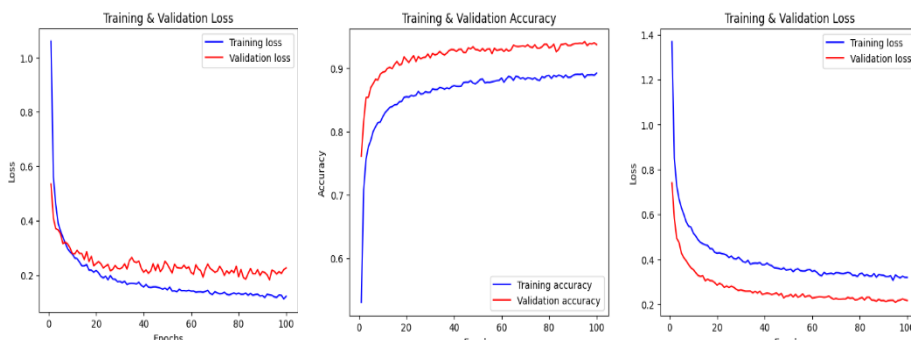
**Tabel 3.** Akurasi, loss dan waktu komputasi pelatihan pada 100 Epoch

Arsitektur CNN Lightweight	Training Akurasi (%)	Validation Akurasi (%)	Training Loss	Validation Loss	Total Waktu Pelatihan (Minute)	Parameter	Ukuran (MB)
MobileNetV2	95.86	93.18	0.1143	0.2258	548	2,423,629	9.25
NasNetMobile	88.98	93.69	0.3205	0.2173	558	4,406,689	16.81
EfficienNetV2B3	97.87	95.85	0.0832	0.1485	575	13,129,035	50.08
Custom CNN Lightweight	97.12	91.44	0.0861	0.3172	565	134.765	0.53

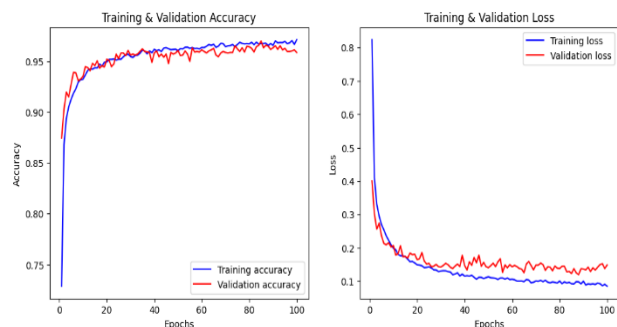
Hasil percobaan pada Tabel 3 menunjukkan bahwa setiap arsitektur CNN *Lightweight* memiliki karakteristik performa yang berbeda. EfficientNetV2B3 memberikan hasil terbaik dengan akurasi pelatihan 97,87% dan akurasi validasi 95,85% serta nilai loss terendah meskipun memiliki jumlah parameter terbesar dan waktu pelatihan paling lama. MobileNetV2 menampilkan performa yang stabil dengan akurasi validasi 93,18%, namun selisih loss antara pelatihan dan validasi. NasNetMobile memiliki akurasi pelatihan yang lebih rendah tetapi akurasi validasi cukup tinggi 93,69% menandakan kemampuan generalisasi yang baik meskipun *loss* relatif besar. Sementara itu Custom CNN Lightweight menjadi model paling efisien dengan ukuran model sangat kecil 0,53 MB dan jumlah parameter jauh lebih sedikit namun akurasi validasinya 91,44% lebih rendah dibanding model lain.



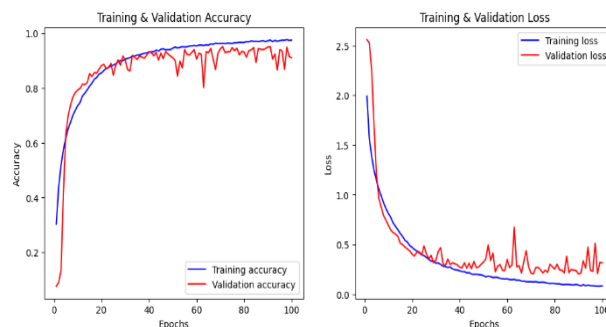
**Gambar 6.** MobileNetV2



**Gambar 7.** NasNetMobile



Gambar 8. EfficientNetV2B3



Gambar 9. Custom CNN Lightweight

Berdasarkan grafik akurasi dan *loss* pada Gambar 6 hingga Gambar 9, terlihat bahwa setiap arsitektur CNN Lightweight menunjukkan pola konvergensi yang berbeda selama proses pelatihan 100 epoch. MobileNetV2 dan EfficientNetV2B3 memperlihatkan kurva akurasi yang meningkat secara stabil dan selisih kecil antara akurasi pelatihan dan validasi menandakan kemampuan generalisasi yang baik. EfficientNetV2B3 menunjukkan konvergensi tercepat dan nilai *loss* paling rendah sehingga menjadi model dengan performa paling optimal. NasNetMobile memiliki akurasi pelatihan lebih rendah pada awal pelatihan namun akurasi validasinya meningkat dan relatif stabil meskipun fluktuasi *loss* validasi masih terlihat menandakan proses pembelajaran yang kurang halus. Sementara itu Custom CNN Lightweight menunjukkan akurasi pelatihan tinggi namun kurva validasi lebih berfluktuasi menunjukkan kecenderungan *overfitting* meskipun *loss* validasinya relatif kecil. Secara keseluruhan grafik menunjukkan bahwa EfficientNetV2B3 paling konsisten dan stabil sedangkan Custom CNN Lightweight unggul dari segi efisiensi tetapi memerlukan penyesuaian lebih lanjut untuk meningkatkan stabilitas validasi.

### 3.3 Hasil Klasifikasi Citra

Pelatihan model EfficientNetV2B3 menunjukkan kinerja yang sangat baik dalam proses klasifikasi. Pengujian dilakukan dengan mengambil beberapa sampel gambar secara acak dari seluruh kelas untuk mengevaluasi kemampuan generalisasi model. Setiap gambar divisualisasikan bersama *true* label dan hasil prediksi. Label berwarna hitam menandakan prediksi benar sedangkan label merah menunjukkan kesalahan klasifikasi. Hasil klasifikasi model EfficientNetV2B3 dapat dilihat pada Gambar 10.



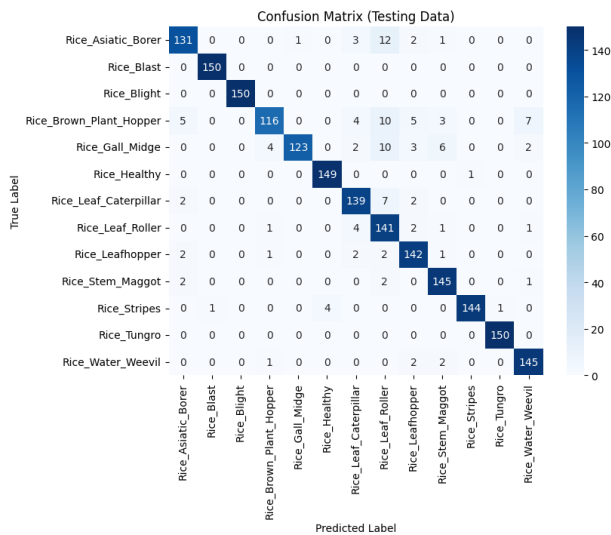
Gambar 10. Prediksi model EfficientNetV2B3

Pada Gambar 10 terlihat bahwa sebagian besar citra berhasil diprediksi dengan benar oleh model. Namun terdapat beberapa kesalahan klasifikasi, misalnya gambar kelas *rice blight* yang seharusnya dikenali sebagai *rice blight* tetapi justru diprediksi sebagai kelas *rice blast*. Meskipun model telah mampu mengenali sebagian besar kelas

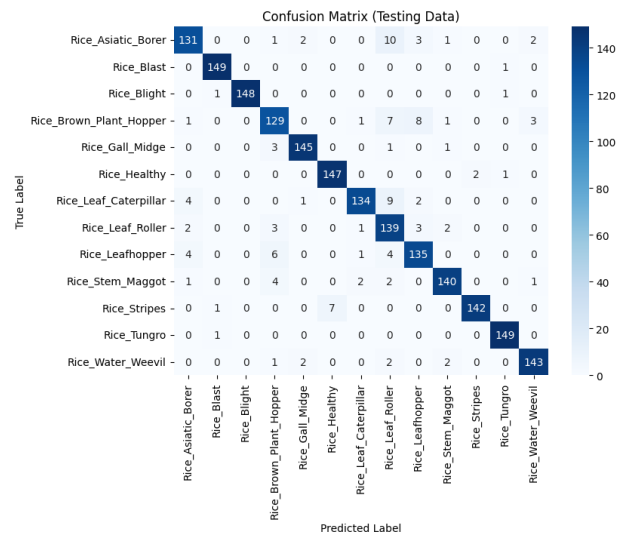
dengan baik namun masih terdapat beberapa kelas yang memiliki kemiripan pola visual sehingga menyebabkan terjadinya kesalahan prediksi pada kondisi tertentu.

### 3.4 Evaluasi Model

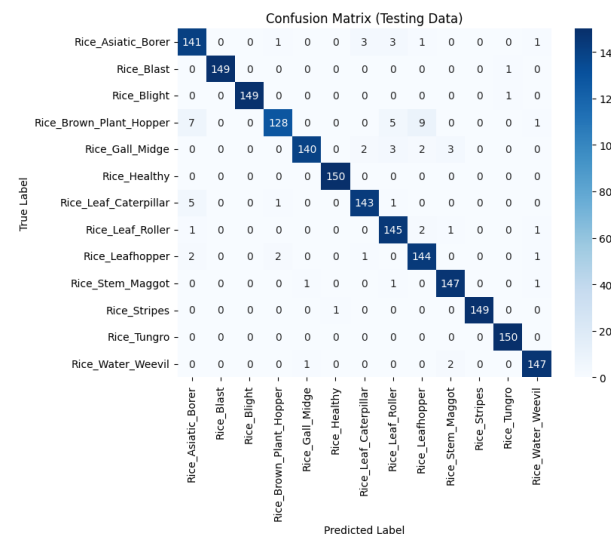
Pengujian pada tahap ini dilakukan menggunakan *confusion matrix* yaitu metode evaluasi yang mampu memberikan informasi mengenai nilai *precision*, *recall*, *f1-score* dan akurasi, serta jumlah prediksi benar maupun salah pada setiap kelas sehingga sangat efektif untuk menganalisis performa model klasifikasi. Melalui evaluasi ini dapat mengidentifikasi pola kesalahan, tingkat kemiripan antar kelas serta kemampuan model dalam membedakan fitur visual pada masing-masing kategori penyakit dan hama tanaman padi. Proses evaluasi diterapkan pada seluruh arsitektur yang diuji seperti MobileNetV2, NASNetMobile, EfficientNetV2B3 dan model Custom CNN Lightweight untuk memperoleh model terbaik yang dihasilkan selama proses pelatihan. Penggunaan berbagai arsitektur memberikan gambaran yang lebih luas mengenai keunggulan relatif masing-masing model dalam hal kompleksitas jaringan, kebutuhan komputasi dan kemampuan generalisasi terhadap data, sehingga hasil akhir yang dicapai mencerminkan model yang paling efektif untuk klasifikasi penyakit daun dan hama tanaman padi.



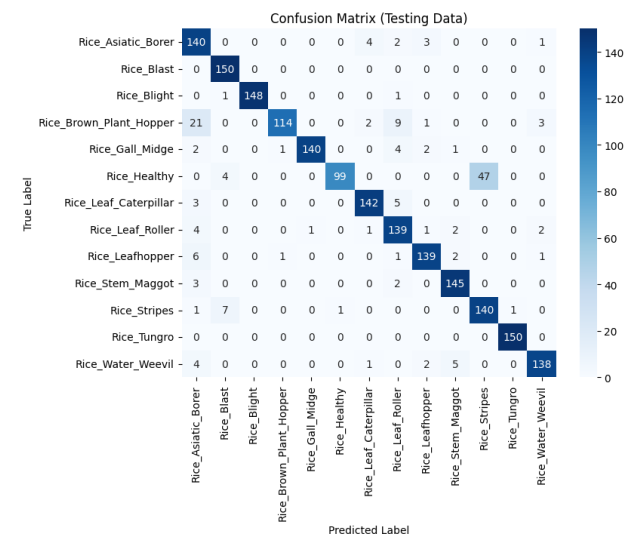
Gambar 11. MobileNetV2



Gambar 12. NasNetMobile



Gambar 13. EfficientNetV2B3



Gambar 14. Custom CNN Lightweight

Berdasarkan *confusion matrix* Gambar 11 sampai Gambar 14, seluruh arsitektur menunjukkan performa klasifikasi yang sangat baik dengan pola diagonal yang dominan menandakan sebagian besar citra pada 13 kelas penyakit dan hama padi berhasil diprediksi secara benar. MobileNetV2 dan EfficientNetV2B3 tampak menghasilkan prediksi yang paling konsisten dengan kesalahan klasifikasi yang sangat minimal sedangkan NASNetMobile masih menunjukkan beberapa *misclassification* pada kelas yang memiliki kemiripan visual. Model Custom CNN *Lightweight* juga menunjukkan performa yang kompetitif meskipun lebih ringan namun masih terdapat sedikit kesalahan pada kelas tertentu yang tingkat kemiripannya tinggi. Secara keseluruhan hasil ini menunjukkan bahwa

seluruh arsitektur mampu melakukan generalisasi dengan baik pada data *testing* dengan perbedaan performa yang ditentukan oleh kompleksitas arsitektur masing-masing model. Berdasarkan confusion matrix, kelas yang sulit diklasifikasikan dengan benar adalah *brown plant hopper* yang mengalami *misclassification* ke *gall midge* dan *leafhopper* serta kelas *leaf caterpillar* yang sesekali diprediksi sebagai *leafhopper*. Kesalahan ini terjadi karena beberapa kelas hama dengan karakteristik morfologi yang mirip masih menimbulkan tantangan dalam proses klasifikasi. Evaluasi model training dengan data testing dapat dilihat dengan rinci pada Tabel 4.

**Tabel 4.** Evaluasi model training dengan data testing

Arsitektur CNN Lightweight	Testing Akurasi (%)	Precision (%)	Recall (%)	F1 Score (%)
MobileNetV2	94.00	94.00	94.00	94.00
NasNetMobile	94.00	94.00	94.00	94.00
EfficientNetV2B3	97.00	97.00	97.00	97.00
Custom CNN Lightweight	91.00	92.00	91.00	91.00

Hasil evaluasi Tabel 4 menunjukkan bahwa EfficientNetV2B3 memberikan kinerja terbaik dibandingkan arsitektur lainnya dengan nilai akurasi, *precision*, *recall* dan *F1-score* sebesar 97%, menunjukkan kemampuan model yang sangat stabil dalam mengenali seluruh kelas hama dan penyakit padi. MobileNetV2 dan NasNetMobile menunjukkan performa yang sama dengan akurasi 94%, mengindikasikan bahwa kedua arsitektur tersebut mampu melakukan generalisasi dengan baik meskipun memiliki kompleksitas yang berbeda. Sedangkan untuk model Custom CNN Lightweight menghasilkan akurasi 91% sedikit lebih rendah dibandingkan model lainnya yang diuji, menunjukkan bahwa meskipun model ini lebih ringan dan efisien untuk implementasi perangkat edge, kemampuan identifikasinya masih memerlukan peningkatan terutama pada kelas-kelas yang memiliki kemiripan visual tinggi. Secara keseluruhan perbandingan ini menunjukkan bahwa penggunaan arsitektur modern seperti EfficientNetV2B3 memberikan keunggulan signifikan dalam hal akurasi dan konsistensi prediksi tetapi model ringan seperti Custom CNN tetap relevan untuk aplikasi di perangkat dengan sumber daya terbatas.

## 4. KESIMPULAN

Penelitian ini berfokus pada evaluasi performa model klasifikasi penyakit dan hama tanaman padi menggunakan empat arsitektur CNN Lightweight yaitu MobileNetV2, NASNetMobile, EfficientNetV2B3 dan Custom CNN Lightweight. Tujuan utama penelitian ini adalah mengidentifikasi arsitektur yang mampu menghasilkan akurasi tinggi sekaligus memiliki efisiensi komputasi yang sesuai untuk implementasi pada perangkat mobile, IoT dan sistem edge. Hasil pelatihan selama 100 *epoch* menunjukkan bahwa pemilihan arsitektur memberikan pengaruh signifikan terhadap kualitas klasifikasi. EfficientNetV2B3 terbukti menjadi model paling unggul dengan akurasi pengujian, *precision*, *recall* dan *f1-score* masing-masing sebesar 97%, sedangkan MobileNetV2 dan NASNetMobile menunjukkan performa kompetitif dengan akurasi 94%. Custom CNN Lightweight tampil sebagai arsitektur paling efisien dengan ukuran model hanya 0,53 MB, meskipun akurasi pengujiannya sebesar 91% mengindikasikan perlunya peningkatan pada kelas-kelas yang memiliki kemiripan visual tinggi. Analisis *confusion matrix* juga memperlihatkan bahwa seluruh arsitektur mampu menghasilkan pola prediksi diagonal yang kuat menandakan kemampuan generalisasi yang baik pada data uji. Meskipun demikian, penelitian ini masih dapat dilakukan pengembangan khususnya dalam meningkatkan stabilitas model ringan melalui optimasi struktur jaringan, penyesuaian jumlah lapisan serta penerapan teknik augmentasi data yang lebih beragam agar model semakin adaptif terhadap variasi citra di kondisi lapangan yang kompleks.

## ACKNOWLEDGEMENTS

Penelitian ini didukung oleh Pusat Penelitian dan Pengabdian kepada Masyarakat (P3M) Politeknik Negeri Bengkalis melalui dana Penelitian PNPB Tahun Anggaran 2025 dengan Skema Penelitian Dosen Pemula.

## REFERENCES

- [1] A. U. Astagina, E. Juniar, S. Mutmainah, and T. A. Lorosae, "Klasifikasi Hama dan Penyakit Tanaman Padi Menggunakan Algoritma Decision Tree," *Journal of Computer Science and Informatics*, vol. 2, no. 2, 2025, doi: 10.34304/scientific.v2i2.376.
- [2] Y. Istikorini, S. N. Rohmah, A. Y. Maulina, M. Fiqri, M. R. Fortunata, V. K. A. S. Sutanto, D. P. Ardian, F. Z. Navizan, and I. F. Alam, "Penyuluhan Hama dan Penyakit pada Tanaman Padi dan Hortikultura di Desa Cihamerang, Sukabumi," *Jurnal Pusat Inovasi Masyarakat*, vol. 7, no. 1, 2025, doi: 10.29244/jpim.7.1.102-115.
- [3] N. Habibah, R. A. M. Ramadhan, N. H. Emila, J. Sani, and N. Wulandari, "Inventarisasi Hama Penyakit Tanaman Padi di Desa Sukaharja Kecamatan Cisayong Kabupaten Tasikmalaya," *Jurnal Pertanian Cemara*, vol. 21, no. 1, 2024, doi: 10.24929/fp.v21i1.3420.
- [4] Wagiyanti, H. Hamidson, and Suwandi, "Intensity and Incidence of Pest Disease Attacks on Rice Plants in Enggal Rejo Village, Air Salek Subdistrict," *Journal of Global Sustainable Agriculture*, vol. 4, no. 2, 2024, doi: 10.32502/jgsa.v4i2.8408.



- [5] M. S. Pramono and A. P. Wibowo, "Penerapan Convolutional Neural Network untuk Identifikasi Penyakit pada Tanaman Padi dari Citra Daun Menggunakan ResNet-101," *Djtechno: Jurnal Teknologi Informasi*, vol. 5, no. 3, 2024, doi: 10.46576/djtechno.v5i3.5098.
- [6] R. R. Burhanuddin, "Klasifikasi Penyakit Padi Melalui Citra Daun Menggunakan Metode Naive Bayes," *JITET: Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4012.
- [7] H. Terzioğlu, A. Gölcük, A. M. A. Shakarji, and M. Y. Al-Bayati, "Comparative Analysis of Deep Learning-Based Feature Extraction and Traditional Classification Approaches for Tomato Disease Detection," *Agronomy*, vol. 15, no. 7, 2025, doi: 10.3390/agronomy15071509.
- [8] W. Jia, J. Gao, W. Xia, Y. Zhao, H. Min, and J.-T. Lu, "A Performance Evaluation of Classic Convolutional Neural Networks for 2D and 3D Palmprint and Palm Vein Recognition," *International Journal of Automation and Computing*, vol. 18, no. 1, 2021, doi: 10.1007/s11633-020-1257-9.
- [9] R. Kusuma and R. Rajkumar, "Plant Leaf Disease Detection and Classification Using Artificial Intelligence Techniques: A Review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 38, no. 2, 2025, doi: 10.11591/ijeecs.v38.i2.pp1308-1323.
- [10] S. Z. M. Zaki, M. A. Zulkifley, M. M. Stofa, N. A. M. Kamari, and N. A. Mohamed, "Classification of Tomato Leaf Diseases Using MobileNet V2," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, 2020, doi: 10.11591/ijai.v9.i2.pp290-296.
- [11] A. Julianto and A. Sunyoto, "A Performance Evaluation of Convolutional Neural Network Architecture for Classification of Rice Leaf Disease," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, 2021, doi: 10.11591/ijai.v10.i4.pp1069-1078.
- [12] U. Suttapakti and A. Bunpeng, "Potato Leaf Disease Classification Based on Distinct Color and Texture Feature Extraction," in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT)*, 2019, doi: 10.1109/ISCIT.2019.8905128
- [13] P. Deepika and B. Arthi, "Prediction of Plant Pest Detection Using Improved Mask-FRCNN in Cloud Environment," *Measurement: Sensors*, vol. 24, 2022, doi: 10.1016/j.measen.2022.100549.
- [14] S. Liu, J. Zhao, H. Wang, R. Li, and Z. Lu, "Using Convolutional Neural Networks to Realize Deep Recognition Analysis of Power Technology Standard Images," *Procedia Computer Science*, vol. 243, 2024, doi: 10.1016/j.procs.2024.09.040.
- [15] M. F. A. Maulana, N. M. Anggadimas, and D. A. Sani, "Klasifikasi Citra Penyakit Daun Padi dengan Metode CNN Menggunakan Arsitektur ResNet50V2," *CESS: Journal of Computer Engineering, System and Science*, vol. 10, no. 2, 2025, doi: 10.24114/cess.v10i2.66960.
- [16] P. I. Ritharson, K. Raimond, X. A. Mary, J. E. Robert, and A. J., "DeepRice: A Deep Learning and Deep Feature-Based Classification of Rice Leaf Disease Subtypes," *Artificial Intelligence in Agriculture*, vol. 11, 2024, doi: 10.1016/j.aiaa.2023.11.001.
- [17] D. Ngo, H. C. Park, and B. Kang, "Edge Intelligence: A Review of Deep Neural Network Inference in Resource-Limited Environments," *Electronics*, vol. 14, no. 12, 2025, doi: 10.3390/electronics14122495.
- [18] Y. A. Sitorus et al., "Evaluasi Komparatif Arsitektur Lightweight CNN, MobileNetV2, dan EfficientNetB0 dalam Deteksi Penyakit Daun Jagung," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 8, 2025.
- [19] S. Yuliany, Aradea, and A. N. Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network," *Jurnal Buana Informatika*, vol. 13, no. 1, 2022, doi: 10.24002/jbi.v13i1.5022.
- [20] C. Pal, S. Karmakar, I. Mukherjee, and P. P. Chakrabarti, "A Lightweight and Explainable CNN Model for Empowering Plant Disease Diagnosis," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-94083-1.
- [21] A. Julianto, A. Sunyoto, and F. W. Wibowo, "Optimasi Hyperparameter Convolutional Neural Network untuk Klasifikasi Penyakit Tanaman Padi," *TEKNIMEDIA: Teknologi Informasi dan Multimedia*, vol. 3, no. 2, 2022, doi: 10.46764/teknimedia.v3i2.77.
- [22] L. D. Quach, Q. K. Nguyen, Q. A. Nguyen, and L. T. T. Lan, "Rice Pest Dataset Supports the Construction of Smart Farming Systems," *Data in Brief*, vol. 52, 2024, doi: 10.1016/j.dib.2024.110046.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, doi: 10.1109/CVPR.2018.00474.
- [24] H. A. Mubarak and R. Novita, "Klasifikasi Citra X-Ray Tuberkulosis Menggunakan Convolutional Neural Networks," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 4, 2025, doi: 10.47065/bits.v6i4.6515.
- [25] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," *Proceedings of Machine Learning Research*, vol. 139, 2021, doi: 10.48550/arXiv.2104.00298.
- [26] S. Ibrahim, H. A. Abdallah, K. M. Amin, R. I. Alkanhel, and M. Ibrahim, "Soft Attention-Based EfficientNetV2B3 Model for Skin Cancer Disease Classification Using Dermoscopy Images," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3486153.
- [27] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, doi: 10.1109/CVPR.2018.00907.
- [28] T. S. Winanto, C. Rozikin, and A. Jamaludin, "Analisa Performa Arsitektur Transfer Learning untuk Mengidentifikasi Penyakit Daun pada Tanaman Pangan," *Journal of Applied Informatics and Computing*, vol. 7, no. 1, 2023, doi: 10.30871/jaic.v7i1.5991.
- [29] C. Manliguez, "Generalized Confusion Matrix for Multiple Classes," pp. 5–7, 2016, doi: 10.13140/RG.2.2.31150.51523.
- [30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference for Learning Representations*, 2014, doi: 10.48550/arXiv.1412.6980.