

Analisis Perbandingan Arsitektur *Convolutional Neural Network* pada Klasifikasi Jenis Penyakit Daun Padi

Erdina Turnip¹, Anief Fauzan Rozi^{2*}

^{1,2} Sistem Informasi, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta
Gg. Jemb. Merah No.84C, Soropadan, Condongcatur, Kec. Depok, Kabupaten Sleman, DIY

¹erdinaaturnip11@gmail.com

²anief@mercubuana-yogya.ac.id

Intisari— Penyakit pada tanaman padi merupakan ancaman serius bagi produksi beras di Indonesia, dengan kerugian tahunan mencapai 200.000-300.000 ton. Deteksi dini dan diagnosis akurat penyakit daun padi sangat penting untuk pengendalian yang efektif, namun metode manual membutuhkan waktu dan tenaga yang besar. Penelitian ini bertujuan membandingkan kinerja arsitektur *Convolutional Neural Network* (*CNN*), yaitu *Xception* dan *NASNetMobile*, dalam mengklasifikasikan jenis penyakit daun padi. Metode yang digunakan meliputi pengumpulan dataset 670 gambar daun padi, *preprocessing data*, perancangan dan pelatihan model *CNN*, serta evaluasi performa menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa arsitektur *Xception* mengungguli *NASNetMobile* dengan akurasi 93% berbanding 83%. *Xception* juga menunjukkan performa yang lebih stabil dan konsisten dalam mengklasifikasikan berbagai jenis penyakit daun padi, terutama untuk *Bacterial leaf blight* dan *Brown spot*. Penelitian ini memberikan wawasan baru tentang efektivitas arsitektur *CNN* dalam klasifikasi penyakit tanaman, yang dapat bermanfaat untuk pengembangan sistem deteksi penyakit yang lebih akurat dan efisien di masa depan.
Keywords— Klasifikasi, *Xception*, *NASNetMobile*, *CNN*, padi

Abstract— Rice diseases are a serious threat to rice production in Indonesia, with annual losses reaching 200,000-300,000 tons. Early detection and accurate diagnosis for rice leaf diseases are essential for effective control, but manual methods require a lot of time and effort. This study aims to compare the performance of *Convolutional Neural Network* (*CNN*) architectures, namely *Xception* and *NASNetMobile*, in classifying rice leaf disease types. The methods used include collecting a dataset of 670 rice leaf images, data preprocessing, *CNN* model design and training, and performance evaluation using accuracy, precision, recall, and F1-score metrics. The results show that the *Xception* architecture outperforms *NASNetMobile* with 93% accuracy versus 83%. *Xception* also showed more stable and consistent performance in classifying different types of rice leaf diseases, especially for *Bacterial leaf blight* and *Brown spots*. This study provides new insights into the effectiveness of *CNN* architecture in plant disease classification, which can be beneficial for the development of more accurate and efficient disease detection systems in the future.

Keywords— Classification, *Xception*, *NASNetMobile*, *CNN*, padi

I. PENDAHULUAN

Di Indonesia, beras merupakan makanan pokok bagi sebagian besar penduduk dan berperan strategis dalam mempertahankan produksi padi yang optimal untuk memastikan ketahanan pangan dan meningkatkan kesejahteraan seluruh Masyarakat [1]. Namun produksi padi seringkali mendapat berbagai kendala, Diperkirakan sekitar 200.000 hingga 300.000 ton beras rusak akibat serangan penyakit pada daun padi di Indonesia setiap tahunnya [2]. Penyakit yang menyerang padi antara lain penyakit busuk daun bakteri, penyakit busuk tungro, penyakit busuk coklat, penyakit hawar daun, penyakit busuk batang, dan penyakit api palsu [3]. Dalam mencegah penyebaran penyakit pada masa tanam merupakan hal yang penting dan harus dilakukan dengan benar agar kuantitas dan kualitas tidak terganggu pada saat panen [4]. Kesalahan

penanganan atau kurangnya pengetahuan dapat menyebabkan gagal panen dan kerugian bagi petani [5]. Namun proses identifikasi manual yang dilakukan oleh petani memakan waktu lama dan memerlukan banyak tenaga serta keterampilan khusus [6].

Perkembangan teknologi kecerdasan buatan dan teknologi *image processing deep learning* telah membuka peluang baru untuk deteksi otomatis [7]. Metode yang banyak digunakan adalah *Convolutional Neural Network* (*CNN*), yang telah terbukti efektif dalam berbagai tugas pemrosesan gambar seperti klasifikasi, deteksi objek, dan segmentasi [8]. *CNN* adalah jenis arsitektur pembelajaran mendalam yang berspesialisasi dalam pemrosesan data gambar. *CNN* terdiri dari beberapa lapisan konvolusional yang bertindak sebagai ekstraktor fitur dan lapisan terhubung penuh dan bertanggung

jawab untuk melakukan klasifikasi berdasarkan fitur yang diekstraksi [9].

Teknik yang digunakan adalah *deep learning* dengan algoritma *Convolutional Neural Network (CNN)* yang membandingkan arsitektur *Xception* dan *NASNetMobile* untuk mengklasifikasikan berbagai jenis penyakit padi berdasarkan gejala visual pada daun. *Xception* dan *NASNetMobile* dipilih karena kedua model ini memberikan arsitektur yang efisien dan canggih untuk menangani tugas klasifikasi gambar. *Xception* menggunakan *convolution depthwise separable*, yang memungkinkan proses yang lebih efisien dan telah terbukti berhasil dalam klasifikasi citra, sementara *NASNetMobile*, yang berasal dari arsitektur otomatis, menawarkan fleksibilitas dan kinerja yang baik.

Dalam beberapa tahun terakhir, klasifikasi penyakit pada tanaman padi telah menggunakan teknologi *deep learning* dengan arsitektur seperti *MobileNet* dan *NASNetMobile*. Arsitektur *Xception* dirancang untuk meningkatkan akurasi klasifikasi gambar, sementara *NASNetMobile* adalah skema yang dirancang untuk meningkatkan kinerja perangkat mobile. Penelitian sebelumnya oleh [10] membandingkan kinerja arsitektur model yang berbeda dalam klasifikasi penyakit tanaman menggunakan gambar daun. Dari hasil pengujian 9 kali percobaan, model *Convolutional Neural Network (CNN)* menggunakan arsitektur *Xception* mencapai akurasi pelatihan tertinggi sebesar 93,81% dan akurasi validasi sebesar 81,75%, serta memiliki kemampuan yang sangat baik dalam mengidentifikasi pola dan fitur pada data pelatihan.

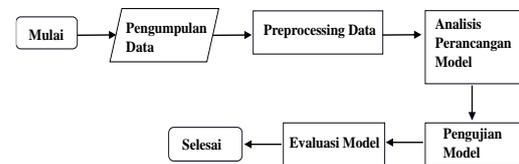
Penelitian oleh [11] dengan membandingkan 2 model *deep learning MobileNet* dan *NASNetMobile* untuk mendeteksi penyakit pada daun kentang. Dilakukan pemisahan data latih dan data uji dengan perbandingan 90:10. Hasil penelitian mencapai akurasi 90,96%, presisi 90,86%, recall 91,11%, dan skor F1 92,93%. Temuan ini menunjukkan bahwa *NASNetMobile* lebih baik daripada *MobileNet* dalam deteksi penyakit tanaman berbasis analisis gambar, memberikan dasar yang kuat untuk pengembangan sistem deteksi penyakit tanaman yang lebih baik.

Berbagai penelitian telah menunjukkan potensi model *deep learning* dalam klasifikasi penyakit tanaman, tetapi perbandingan langsung antara arsitektur *Xception* dan *NASNetMobile* khusus untuk daun padi masih terbatas. Penelitian ini bertujuan untuk membandingkan kinerja arsitektur *Xception* dan *NASNetMobile* dalam mengklasifikasikan berbagai jenis penyakit pada daun padi, seperti *bacterial leaf blight*, *brown spot*, dan *healthy rice leaf*. Hasil penelitian diharapkan dapat memberikan wawasan lebih mendalam tentang efektivitas masing-masing arsitektur dalam mengklasifikasikan penyakit pada tanaman padi.

II. METODOLOGI PENELITIAN

Sebagai landasan untuk pengembangan model *deep learning*, penelitian ini dimulai dengan pengumpulan data. Selanjutnya, data diproses melalui berbagai proses, termasuk *preprocessing*,

penyesuaian dimensi gambar, dan pembagian dataset untuk pelatihan dan validasi. Fokus langkah selanjutnya adalah perancangan dan pelatihan model. Pada tahap akhir, model



yang telah dilatih diuji untuk mengetahui seberapa efektif kedua model. Alur penelitian yang akan dilakukan dapat dilihat pada gambar 1.

Gambar 1. Alur Penelitian

Pengumpulan Data

Pada tahap pengumpulan data ini berasal dari dataset gambar penyakit daun padi dari *repository kaggle*. Dataset ini terdiri dari 670 gambar dengan 3 kelas penyakit daun padi yaitu *bacterial leaf blight*, *Brown spot*, dan *healthy leaf*. Setiap gambar telah diberi label kelas, yang memungkinkan identifikasi dan kategorisasi yang tepat. Proses ini melibatkan analisis menyeluruh terhadap setiap gambar untuk memastikan bahwa label kelas ditetapkan dengan akurat.

Preprocessing data

Setelah mengumpulkan data gambar, langkah selanjutnya adalah *preprocessing data*. *Preprocessing* adalah tahap pengolahan gambar yang bertujuan untuk menghasilkan data yang lebih baik dan siap untuk proses selanjutnya. Proses *preprocessing* dimulai dengan *resize* gambar, *augmentasi data*, dan *split data* yang dibagi menjadi dua, yaitu data *training*, data *validation*.

TABEL I
PEMBAGIAN DATASET

Labels	Data Training 80%	Data Validation 20%
<i>Bacterial leaf blight</i>	179	45
<i>Brown spot</i>	178	44
<i>Healthy leaf</i>	179	45
Total	536	134

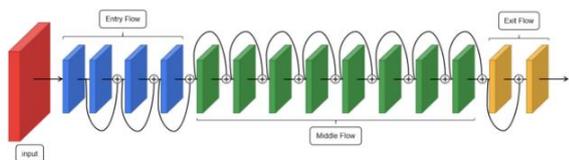
Tabel I menunjukkan distribusi data untuk pelatihan dan validasi dengan pembagian dataset menjadi dua bagian: *training*, dan *validation*. Gambar untuk *training* dipisahkan sebesar 80%, dan untuk *validation* sebesar 20%. Sehingga, total gambar untuk *training* adalah 536 gambar, dan untuk *validation* adalah 134 foto.

Analisis dan Perancangan Model CNN

Pada tahap ini akan dibangun beberapa model menggunakan metode CNN dengan keras, model yang dibangun dan akan dianalisis yaitu *Xception* dan *NASNetMobile*.

Xception Arsitektur

Gambar 2 menunjukkan arsitektur model *Xception* merupakan varian dari jaringan saraf konvolusional (CNN). Arsitektur model memiliki lapisan konvolusional untuk mengekstraksi fitur, lapisan pengumpulan maksimal untuk mengurangi dimensi, dan lapisan *dropout* untuk mencegah *overfitting* untuk klasifikasi [12] yang terdiri dari tiga bagian utama: *Entry Flow*, *Middle Flow*, dan *Exit Flow*. *Entry Flow* (biru), terdiri dari beberapa lapisan konvolusi yang digunakan untuk mengekstraksi fitur awal dari input. *Middle Flow* (hijau), terdiri dari beberapa blok *residual* yang lebih dalam, yang dirancang untuk menangkap fitur yang lebih kompleks melalui lapisan konvolusional yang saling berhubungan. *Exit Flow* (kuning) digunakan untuk menggabungkan fitur yang diekstraksi dan menghasilkan *output* akhir. Dibandingkan dengan model CNN konvensional, model *Xception* menggunakan *convolution depthwise separable* untuk meningkatkan efisiensi dan akurasi pengenalan pola serta mengurangi jumlah parameter yang diperlukan.

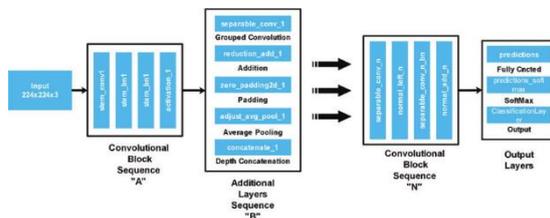


Gambar 2. Arsitektur *Xception*

NASNetMobile Arsitektur

Gambar 3 menunjukkan arsitektur model *NASNetMobile* yang digunakan untuk mengklasifikasikan berbagai jenis penyakit daun padi ditunjukkan dalam gambar berikut. Model ini terdiri dari tiga bagian utama: *Sequence Convolutional Block "A"* yaitu mengolah input gambar daun padi, *Sequence Convolutional Block "B"* melibatkan lapisan konvolusi terpisah, penambahan, *padding*, *average pooling*, dan *depth concatenation* untuk ekstraksi fitur lebih dalam, dan *Sequence*

Convolutional Block "N" mengolah fitur lebih lanjut. Arsitektur *NASNetMobile* dibuat untuk meningkatkan produktivitas dan kinerja klasifikasi.



Gambar 3. Arsitektur *NASNetMobile*

Pelatihan dan Pengujian Model

Untuk pelatihan dan pengujian model dilakukan peningkatan variasi data pelatihan dan menghindari *overfitting*, teknik augmentasi seperti rotasi, *flipping*, dan *zooming* juga digunakan. Kedua model, *NASNetMobile* dan *Xception*, menggunakan bobot yang telah dilatih sebelumnya pada dataset *ImageNet*. Setelah itu, model akan dikompilasi dengan *optimizer Adam* dengan nilai pembelajaran 0.0001, yang meningkatkan ketelitian model. Proses selanjutnya adalah melatih model selama 20 *epoch* untuk mencapai tingkat kesalahan dan akurasi yang paling tinggi.

Evaluasi Model

Tahap penting dalam proses pengembangan model adalah evaluasi model. Tujuan evaluasi ini adalah menentukan model mana yang paling cocok untuk representasi data yang digunakan dan seberapa efektif model tersebut. Dengan menggunakan data pengujian yang tidak terlihat selama pelatihan, evaluasi model dilakukan untuk mengevaluasi kemampuan *NASNetMobile* dan *Xception* dalam klasifikasi jenis penyakit daun padi. Akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix* digunakan untuk memeriksa kedua model.

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \times recall \times precision}{recall + precision}$$

Formula di atas menunjukkan bahwa :

- TP = Menunjukkan jumlah prediksi benar positif.
- TN = Menunjukkan jumlah prediksi benar negative
- FP = Menunjukkan jumlah prediksi salah positif
- FN = Menunjukkan jumlah prediksi salah negatif

III. HASIL DAN PEMBAHASAN

Dalam penelitian ini, dilakukan perbandingan kinerja dua arsitektur *deep learning*, yaitu *Xception* dan *NASNetMobile*, dalam klasifikasi penyakit daun padi. Kedua model dilatih dan diuji menggunakan dataset citra daun padi yang mencakup beberapa jenis penyakit umum.

Hasil Pelatihan dan Pengujian Model

Pelatihan dan pengujian pada dua arsitektur *CNN*, yaitu *Xception* dan *NASNetMobile* untuk mengklasifikasikan jenis penyakit daun padi dalam penelitian ini. Hasil pelatihan model *Convolutional Neural Network (CNN)* merujuk pada performa dan kualitas model setelah dilatih menggunakan data pelatihan sebelumnya. Proses pelatihan melibatkan penyesuaian parameter model berdasarkan data pelatihan, sehingga model dapat belajar memahami pola dan fitur yang ada dalam data tersebut. Hasil pelatihan *CNN* dapat dievaluasi dengan menggunakan berbagai *matrix* dan indikator.

Epoch 3/20									
20/20	262s	12s/step	accuracy: 0.6217	loss: 0.5099	val_accuracy: 0.6344	val_loss: 0.4346			
Epoch 4/20									
20/20	11s	309ms/step	accuracy: 0.6790	loss: 0.2973	val_accuracy: 0.8667	val_loss: 0.3622			
Epoch 5/20									
20/20	241s	12s/step	accuracy: 0.8506	loss: 0.3978	val_accuracy: 0.8281	val_loss: 0.3547			
Epoch 6/20									
20/20	10s	280ms/step	accuracy: 0.8750	loss: 0.5111	val_accuracy: 0.9333	val_loss: 0.2105			
Epoch 7/20									
20/20	274s	13s/step	accuracy: 0.8739	loss: 0.3212	val_accuracy: 0.8258	val_loss: 0.3736			
Epoch 8/20									
20/20	11s	298ms/step	accuracy: 0.8750	loss: 0.2951	val_accuracy: 0.9000	val_loss: 0.2701			
Epoch 9/20									
20/20	351s	11s/step	accuracy: 0.8818	loss: 0.3072	val_accuracy: 0.8641	val_loss: 0.3190			
Epoch 10/20									
20/20	11s	292ms/step	accuracy: 0.9062	loss: 0.2122	val_accuracy: 0.8667	val_loss: 0.3071			
Epoch 11/20									
20/20	245s	12s/step	accuracy: 0.9008	loss: 0.2935	val_accuracy: 0.8687	val_loss: 0.3099			
Epoch 12/20									
20/20	10s	274ms/step	accuracy: 0.8750	loss: 0.3281	val_accuracy: 0.9667	val_loss: 0.1429			
Epoch 13/20									
20/20	240s	11s/step	accuracy: 0.9139	loss: 0.2696	val_accuracy: 0.8953	val_loss: 0.2710			
Epoch 14/20									
20/20	10s	257ms/step	accuracy: 0.9062	loss: 0.1859	val_accuracy: 0.9667	val_loss: 0.2283			
Epoch 15/20									
20/20	206s	10s/step	accuracy: 0.9048	loss: 0.2454	val_accuracy: 0.9078	val_loss: 0.2938			
Epoch 16/20									
20/20	9s	225ms/step	accuracy: 0.9688	loss: 0.1592	val_accuracy: 1.0000	val_loss: 0.1132			

Gambar 4. Hasil pelatihan *Xception*

Epoch 3/20									
16/16	51s	2s/step	accuracy: 0.7457	loss: 0.5296	val_accuracy: 0.8047	val_loss: 0.5534			
Epoch 4/20									
16/16	1s	14ms/step	accuracy: 0.7500	loss: 0.6473	val_accuracy: 0.7500	val_loss: 0.4393			
Epoch 5/20									
16/16	34s	2s/step	accuracy: 0.8198	loss: 0.6485	val_accuracy: 0.7422	val_loss: 0.6239			
Epoch 6/20									
16/16	1s	14ms/step	accuracy: 0.9062	loss: 0.2857	val_accuracy: 0.5000	val_loss: 1.6386			
Epoch 7/20									
16/16	33s	2s/step	accuracy: 0.8794	loss: 0.3080	val_accuracy: 0.7344	val_loss: 0.6563			
Epoch 8/20									
16/16	1s	11ms/step	accuracy: 0.8750	loss: 0.4170	val_accuracy: 0.7500	val_loss: 0.5425			
Epoch 9/20									
16/16	33s	2s/step	accuracy: 0.8552	loss: 0.3158	val_accuracy: 0.6406	val_loss: 0.9255			
Epoch 10/20									
16/16	1s	12ms/step	accuracy: 0.8125	loss: 0.2394	val_accuracy: 0.7200	val_loss: 0.8714			
Epoch 11/20									
16/16	33s	2s/step	accuracy: 0.8739	loss: 0.2939	val_accuracy: 0.7500	val_loss: 0.7426			
Epoch 12/20									
16/16	1s	13ms/step	accuracy: 0.8750	loss: 0.2291	val_accuracy: 1.0000	val_loss: 0.0452			
Epoch 13/20									
16/16	33s	2s/step	accuracy: 0.8762	loss: 0.2436	val_accuracy: 0.8047	val_loss: 0.3629			
Epoch 14/20									
16/16	1s	13ms/step	accuracy: 0.8438	loss: 0.3371	val_accuracy: 0.7500	val_loss: 0.4930			
Epoch 15/20									
16/16	32s	2s/step	accuracy: 0.9037	loss: 0.2586	val_accuracy: 0.6010	val_loss: 1.1282			
Epoch 16/20									
16/16	1s	15ms/step	accuracy: 0.8438	loss: 0.1852	val_accuracy: 1.0000	val_loss: 0.1188			

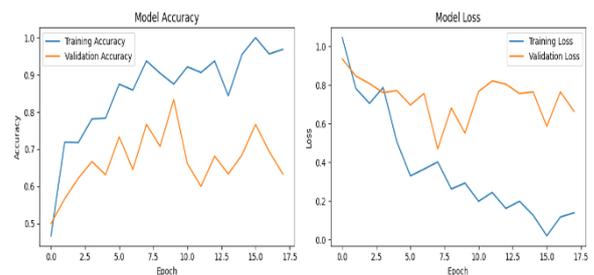
Gambar 5. Hasil pelatihan *NASNetMobile*

Dari Gambar 4 dan 5 dapat dilihat hasil proses pelatihan dua model *Xception* dan *NASNetMobile*, dilakukan dalam 20 epoch dengan hasil yang berbeda. Model *Xception* menunjukkan

peningkatan kinerja yang konsisten, dengan akurasi pelatihan meningkat dari 82,17% menjadi 96,88% dan akurasi validasi mencapai 100% di epoch terakhir. Waktu pelatihan *Xception* bervariasi antara 9-274 detik per epoch. Sementara itu, model *NASNetMobile* menunjukkan peningkatan yang sedikit, dengan akurasi pelatihan meningkat dari 74,57% menjadi 84,38% dan waktu pelatihan 1-51 detik per epoch. Kedua model memiliki loss yang rendah. Secara keseluruhan, *Xception* menunjukkan kinerja yang lebih baik dan lebih stabil.

Hasil Evaluasi Model

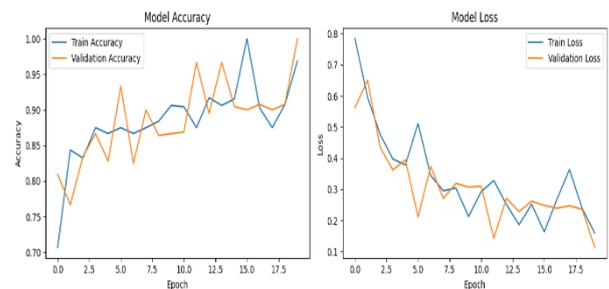
Dalam evaluasi model beberapa aspek penting yang diukur meliputi akurasi, presisi, recall, dan F1-score dari model *Xception* dan *NASNetMobile*. Setiap percobaan pengujian dievaluasi dan disesuaikan untuk menentukan konfigurasi parameter yang menghasilkan akurasi yang baik. Analisis ini membantu memahami bagaimana perubahan parameter mempengaruhi kinerja model, mengidentifikasi overfitting atau underfitting, dan mempertahankan data yang stabil. Dengan demikian dari hasil pelatihan pada kedua model, dapat ditentukan bahwa hasil training loss dapat dilihat dari grafik



dibawah ini.

Gambar 6. Grafik training-loss *Xception*

Gambar 6 menunjukkan performa model *Xception* yang menunjukkan accuracy dan loss pada data pelatihan dan validasi. Akurasi model umumnya meningkat seiring waktu, namun akurasi pelatihan mencapai sekitar 100%, dan akurasi validasi mencapai 95% di akhir pelatihan. Di sisi lain, loss model terus menurun, dengan loss pelatihan menurun lebih cepat dibandingkan loss validasi, dan keduanya mendekati 0,2 pada akhir pelatihan. Grafik ini menunjukkan bahwa model berhasil belajar dari data pelatihan dan menunjukkan proses



pembelajaran yang baik.

Gambar 7. Grafik training-loss *NASNetMobile*

Gambar 7 menunjukkan kinerja model *NASNetMobile* dengan Akurasi pelatihan meningkat secara signifikan dan Loss pelatihan menurun signifikan, namun loss validasi tetap tinggi dan tidak stabil. Pola ini mengindikasikan overfitting, di mana model beradaptasi terlalu baik. Diperlukan penyesuaian model untuk meningkatkan performa pada data baru.

Dari hasil kedua grafik menunjukkan bahwa *Xception* memiliki keunggulan dalam hal akurasi klasifikasi, sementara *NASNetMobile* unggul dalam efisiensi komputasi dan ukuran model. Pemilihan model tergantung pada kebutuhan spesifik implementasi, dengan mempertimbangkan manfaat antara akurasi dan efisiensi.

Berdasarkan analisis confusion *matrix* yang sudah dilakukan, model pertama lebih direkomendasikan untuk digunakan karena menunjukkan performa yang lebih unggul dan konsisten di semua kelas. Nilai parameter akurasi, precision, recall, dan F1-score akan digunakan untuk membandingkan kinerja model pembelajaran mesin yang menggunakan arsitektur *Xception* dan yang menggunakan arsitektur *NASNetMobile* secara keseluruhan.

TABEL II
PERBANDINGAN PERFORMA *XCEPTION* DAN *NASNETMOBILE*

<i>Matrix</i>	<i>Xception</i>	<i>NASNetMobile</i>
Akurasi	93%	83%
Macro Avg F1	93%	83%
Weighted Avg f1	93%	83%
<i>Bacterial leaf blight</i> F1-score	97%	76%
<i>Brown spot</i> F1-score	92%	82%
<i>Healthy leaf</i> F1-score	91%	91%

Hasil evaluasi dari tabel II menunjukkan bahwa model *Xception* secara konsisten mengungguli *NASNetMobile* dalam klasifikasi penyakit daun padi. *Xception* mencapai akurasi keseluruhan 0.93, lebih tinggi dibandingkan *NASNetMobile* yang mencapai 0.83. Konsistensi performa *Xception* terlihat dari nilai macro average dan weighted average F1-score yang sama-sama 0.93, sementara *NASNetMobile* konsisten di 0.83. Dalam klasifikasi per kelas, *Xception* menunjukkan keunggulan signifikan, terutama untuk *Bacterial leaf blight* (F1-score 0.97 dengan 0.76) dan *Brown spot* (0.92 dengan 0.82). Kedua model menunjukkan performa yang identik untuk kelas *Healthy leaf* dengan F1-score 0.91. Secara keseluruhan, berdasarkan metrik evaluasi yang diberikan, *Xception*

menunjukkan kestabilan yang lebih tinggi dalam mengklasifikasikan berbagai jenis penyakit daun padi.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa model *Xception* menunjukkan kinerja yang lebih unggul dibandingkan *NASNetMobile* dalam klasifikasi penyakit daun padi. *Xception* mencapai akurasi keseluruhan 93% dan nilai F1-score yang lebih tinggi untuk berbagai jenis penyakit, sementara *NASNetMobile* mencapai akurasi 83%. Analisis grafik training-loss mengindikasikan bahwa *Xception* berhasil belajar dengan baik dari data pelatihan. Meskipun *NASNetMobile* memiliki keunggulan dalam efisiensi komputasi dan ukuran model yang lebih kecil, *Xception* lebih direkomendasikan untuk klasifikasi penyakit daun padi karena performa yang lebih tinggi dan konsisten di semua kelas. Namun, pemilihan model akhir harus mempertimbangkan konteks aplikasi, potensi dampak kesalahan klasifikasi, dan keseimbangan antara akurasi dan efisiensi komputasi. Jika kecepatan dan efisiensi menjadi prioritas, *NASNetMobile* bisa menjadi pilihan yang baik, terutama jika disertai dengan langkah-langkah mitigasi overfitting, seperti *augmentasi data* atau *regularisasi* yang lebih baik. Penelitian lebih lanjut disarankan untuk mengeksplorasi kombinasi atau hibrida dari kedua model ini, atau bahkan menguji arsitektur lain yang mungkin memberikan keseimbangan optimal antara akurasi dan efisiensi komputasi. Selain itu, uji coba lapangan pada berbagai kondisi lingkungan dan varietas padi yang berbeda diperlukan untuk memastikan model yang dipilih dapat beradaptasi dan tetap memberikan performa yang baik.

REFERENSI

- [1] Suwarno, "Meningkatkan Produksi Padi Menuju Ketahanan Pangan yang Lestari," *J. Pangan*, vol. 19, no. 3, pp. 233–243, 2010.
- [2] W. G. W. Oktaviana Ulfah Nur, Hendrawan Ricky, Khoirul Annas Alfian Dwi, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model," vol. 5, no. 158, pp. 9–11, 2021.
- [3] S. D. N. dan R. M. Veronika Susanti, "Perkembangan Penyakit dan Pertumbuhan Lima Varietas Padi (*Oryza sativa* L.) dengan Sistem Tanam Blok The Development of a Disease and Growth Five Varieties of Rice (*Oryza sativa* L.) with a System of Planting Block," *J. Agrotek. Trop*, vol. 7, no. 1, pp. 8–19, 2018.
- [4] V. Darwis, "Potensi Kehilangan Hasil Panen Dan Pasca Panen Jagung Di Kabupaten Lampung Selatan," *J. Food Syst. Agribus.*, vol. 2, no. 1, pp. 55–66, 2019, doi: 10.25181/jofsa.v2i1.1110.
- [5] S. Sheila, I. Permata Sari, A. Bagas Saputra, M. Kharil Anwar, and F. Restu Pujiyanto, "Deteksi Penyakit Pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)," *Multinetics*, vol. 9, no. 1, pp. 27–34, 2023, doi: 10.32722/multinetics.v9i1.5255.
- [6] S. Yuliany, Aradea, and Andi Nur Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022, doi: 10.24002/jbi.v13i1.5022.
- [7] L. Pinto-Coelho, "How Artificial Intelligence Is Shaping Medical Imaging Technology: A Survey of Innovations and Applications," *Bioengineering*, vol. 10, no. 12, 2023, doi: 10.3390/bioengineering10121435.

- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.
- [10] A. Akram, K. Fayakun, and H. Ramza, "Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network," *Build. Informatics, Technol. Sci.*, vol. 5, no. 2, pp. 397–406, 2023, doi: 10.47065/bits.v5i2.4063.
- [11] A. Fuadi and A. Suharso, "Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang," *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 3, pp. 701–710, 2022, doi: 10.29100/jipi.v7i3.3026.
- [12] M. F. R. Nurhikma Arifin, Maratuttahirah, Juprianus Rusman, "Leaf Disease Detection in Tomato Plants Using Xception Model in Convolutional Neural Network Method," *J. Tek. Inform.*, vol. 5, no. 2, pp. 571–577, 2024.