

KLASIFIKASI TINGKAT KEPARAHAN PENYAKIT LEAF BLAST PADA TANAMAN PADI MENGGUNAKAN EFFICIENTNETB0 MENGGUNAKAN OPTIMASI CLAHE

Ezar Ramadhan*¹, Syakara Akbar², Muhammad Firdaus Al-Farizi³, Tinuk Agustin⁴

¹²³⁴Prodi Informatika, STMIK Amikom Surakarta

¹²³⁴Sukoharjo, Indonesia

Email: ¹ezar.10387@mhs.amikomsolo.ac.id,

²syakara.10384@mhs.amikomsolo.ac.id, ³muhammad.10411@mhs.amikomsolo.ac.id,

⁴agustin.amikom@gmail.com

Abstract

The purpose of this research is to create a classification system for blast disease severity in rice plants using the EfficientNetB0 architecture that has been enhanced with Contrast Limited Adaptive Histogram Equalization (CLAHE). This research technique consists of various steps. Initially, a dataset was collected from the Kaggle platform, which included 300 photos of rice leaves. The photos were then pre-processed with the CLAHE method to enhance contrast. Then data augmentation was performed using ImageDataGenerator to enrich the image variants. Finally, model training was performed using a modified EfficientNetB0 architecture that included more layers to improve accuracy. The findings show that the final model has a validation accuracy of 90%, with the best performance found in the "healthy-light" class. However, the low accuracy and recall figures show that recognizing the "moderate" and "severe" groups is still a challenge. This research makes a significant contribution to the field by building a more precise and efficient classification system for determining blast disease severity. This strategy can help farmers make better decisions in rice crop management. In addition, this research paves the way for further development in dataset balancing and implementation of more advanced deep learning techniques to improve classification performance.

Keywords: CLAHE, Deep Learning, EfficientNetB0, Classification, Blast Disease, Rice Plant.

Abstraksi

Tujuan dari penelitian ini adalah untuk membuat sistem klasifikasi tingkat keparahan penyakit blas pada tanaman padi dengan menggunakan arsitektur EfficientNetB0 yang telah disempurnakan dengan Contrast Limited Adaptive Histogram Equalization (CLAHE). Teknik penelitian ini terdiri dari berbagai langkah. Pada awalnya, kumpulan data dikumpulkan dari platform Kaggle, yang mencakup 300 foto daun padi. Foto-foto tersebut kemudian dipra-proses dengan metode CLAHE untuk meningkatkan kontras. Kemudian augmentasi data dilakukan dengan menggunakan ImageDataGenerator untuk memperkaya varian gambar. Terakhir, pelatihan model dilakukan dengan menggunakan arsitektur EfficientNetB0 yang dimodifikasi yang menyertakan lebih banyak lapisan untuk meningkatkan akurasi. Temuan menunjukkan

bahwa model akhir memiliki akurasi validasi sebesar 90%, dengan performa terbaik ditemukan pada kelas “sehat-cahaya”. Akan tetapi, angka akurasi dan recall yang rendah menunjukkan bahwa mengenali kelompok “sedang” dan “berat” masih menjadi tantangan. Penelitian ini memberikan kontribusi yang signifikan di lapangan dengan membangun sistem klasifikasi yang lebih tepat dan efisien untuk menentukan tingkat keparahan penyakit blas. Strategi ini dapat membantu petani membuat keputusan yang lebih baik dalam pengelolaan tanaman padi. Selain itu, penelitian ini membuka jalan untuk pengembangan lebih lanjut dalam penyeimbangan dataset dan implementasi teknik deep learning yang lebih canggih untuk meningkatkan kinerja klasifikasi.

Kata Kunci: CLAHE, Deep Learning, EfficientNetB0, Klasifikasi, Penyakit Blast, Tanaman Padi.

1. PENDAHULUAN

Padi (*Oryza sativa L.*) adalah komoditas utama dalam pertanian, yang memainkan peran penting dalam memasok kebutuhan pangan global, terutama di negara-negara agraris seperti Indonesia [1]. Berbagai keadaan, termasuk wabah penyakit, dapat berdampak pada hasil panen padi [2]. Salah satu penyakit yang terkenal adalah penyakit *leaf blast*, yang disebabkan oleh jamur *Pyricularia grisea* [3]. Penyakit ini merusak daun tanaman padi sehingga tidak dapat tumbuh dengan baik [4]. Menentukan tingkat keparahan penyakit secara manual mungkin sulit dilakukan, terutama karena penilaian *visual* pada dasarnya bersifat subjektif dan memakan waktu [5].

Penelitian sebelumnya meneliti bagaimana teknologi berbasis pemrosesan gambar digital dapat membantu klasifikasi tingkat keparahan penyakit *blast* pada tanaman padi [6]. Dalam penelitian terbaru, [7] menggunakan arsitektur *MobileNetV2* untuk mengklasifikasikan tingkat keparahan penyakit *leaf blast* pada kumpulan data 300 foto daun padi. Penelitian ini memiliki tingkat akurasi 78,33%, yang merupakan hasil yang positif. Namun, masih ada peluang untuk modifikasi untuk membuat model yang lebih praktis [8]. Selain itu, *dataset* yang digunakan memiliki kendala intrinsik dalam variabilitas data, yang mengganggu kemampuan model untuk menggeneralisasi [9].

Tujuan dari penelitian ini adalah untuk meningkatkan ketepatan kategorisasi tingkat keparahan penyakit *leaf blast* dengan mengimplementasikan kerangka kerja arsitektur *EfficientNetB0* yang lebih efisien dalam penggunaan parameter model [10]. Selain itu, pendekatan Contrast Limited Adaptive Histogram Equalization (CLAHE) digunakan untuk meningkatkan kualitas gambar dengan meningkatkan kontras, sehingga model dapat memahami pola keparahan penyakit dengan lebih baik [11]. Integrasi desain *EfficientNetB0* dengan optimasi CLAHE merupakan hal yang unik dalam penelitian ini, dan diharapkan dapat mengurangi batasan kinerja yang terlihat dalam penyelidikan sebelumnya [12].

2. TINJAUAN PUSTAKA

Penelitian [13] meneliti lima arsitektur CNN (GoogleNet, AlexNet, VGG-16, LeNet-5, dan ResNet-50) untuk pengenalan tulisan tangan Arab. Hasilnya, VGG-16 mendapatkan akurasi tertinggi 83,99%. Namun, penelitian ini tidak mengoptimalkan lebih lanjut dataset atau arsitektur model, dan juga tidak menggunakan teknik pra-pemrosesan gambar untuk meningkatkan kualitas data. Penelitian ini mengusulkan peningkatan akurasi dengan menggabungkan VGG-16 atau desain lain dengan pendekatan prapemrosesan seperti CLAHE.

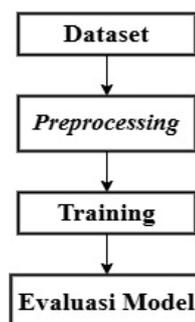
Penelitian [14] menilai seberapa baik kinerja dua arsitektur CNN AlexNet dan GoogLeNet dalam kategorisasi gambar untuk diagnosis pneumonia. Temuan menunjukkan bahwa tingkat akurasi GoogLeNet adalah 78%, sedangkan AlexNet adalah 79%. Pengoptimalan *hyperparameter* dan metode pra-pemrosesan data seperti peningkatan kualitas gambar tidak digunakan dalam penelitian ini. Untuk meningkatkan ketepatan dan ketergantungan diagnosis pneumonia, penelitian ini menyarankan untuk menggunakan arsitektur yang lebih efektif, seperti EfficientNet, dan metode pra-pemrosesan data, seperti CLAHE.

3. METODE PENELITIAN

Penelitian ini bertujuan untuk mencapai akurasi yang tinggi dalam mengklasifikasikan tingkat keparahan gambar pada penyakit *leaf blast* pada tanaman padi menggunakan arsitektur *EfficientNetB0* [15]. Arsitektur *EfficientNet B0* dipilih karena kemampuannya untuk meningkatkan akurasi sekaligus menggunakan lebih sedikit parameter, sehingga ideal untuk dataset berukuran kecil hingga menengah [16].

Prosedur prapemrosesan, yang mencakup CLAHE dan augmentasi data, meningkatkan kualitas data yang masuk, sehingga dapat mengatasi perbedaan kualitas gambar [17].

Transfer learning memungkinkan penggunaan bobot dari *dataset* yang besar, sehingga mengurangi kebutuhan pelatihan awal dan meningkatkan efisiensi pelatihan [18]. Strategi yang digunakan dalam penelitian ini dikembangkan untuk menghasilkan hasil terbaik dan dapat digunakan untuk tantangan yang serupa [19]. Peneliti membuat langkah-langkah penelitian untuk membangun model kategorisasi [20], antara lain: *Dataset*, *Preprocessing*, *Training*, dan *Evaluasi Model*. Gambar 1 mengilustrasikan tahapan penelitian ini.



Gambar 1. Tahap tahap penelitian

3.1. Dataset

Dataset didapat dari platform Kaggle dengan judul Klasifikasi Penyakit Daun Padi CNN. *Dataset* ini mencakup tiga folder yang mengklasifikasikan citra tingkat keparahan penyakit *leaf blast* pada tanaman padi dengan menggunakan tipe file “jpg”. *Dataset* ini berisi total 300 data citra dengan tiga kelas. sehat-ringan sebanyak 100 citra , sedang sebanyak 100 citra, berat sebanyak 100 citra. Tabel distribusi data citra dalam *dataset* ditunjukkan pada Tabel 1.

Tabel 1. Distribusi data citra dalam dataset

Kelas	Jumlah Data Citra
sehat-ringan	100
sedang	100
berat	100
TOTAL	300

3.2. Preprocessing Data

Untuk menyiapkan *dataset* yang siap untuk pelatihan dan penilaian model, *preprocessing* data dilakukan pada tahap pertama penelitian. Gambar tanaman padi yang terkena penyakit *leaf blast* dikumpulkan ke dalam tiga kategori tingkat keparahan: sedang, parah, dan sehat-ringan. Untuk merepresentasikan satu kelas, setiap kelas disimpan dalam direktori tersendiri. Hal ini mempermudah penamaan setiap kelas secara numerik sebagai target keluaran pelatihan. Teknik *Contrast Limited Adaptive Histogram Equalization* (CLAHE) digunakan pada *channel L* (*luminance*) dalam ruang warna LAB untuk meningkatkan kualitas gambar. Dengan menggunakan pendekatan CLAHE, kontras gambar dinaikkan untuk membuat karakteristik penyakit menjadi lebih jelas.

Dalam proses CLAHE, gambar dikonversi ke ruang warna LAB, *channel L* dipisahkan, CLAHE diterapkan ke *channel L*, *channel a* dan *b* digabungkan kembali, dan kemudian gambar dikonversi ke ruang warna BGR. Untuk mengakomodasi input model *EfficientNetB0*, gambar yang ditingkatkan kontrasnya, kemudian diskalakan ke 224x224 piksel. Untuk mempercepat konvergensi model selama pelatihan, *dataset* gambar ini diubah menjadi *array numpy* dan dinormalisasi menggunakan fungsi *preprocess_input* dari Keras. Untuk menjamin bahwa evaluasi model dilakukan pada data baru yang tidak terlihat selama pelatihan, *dataset* kemudian dibagi menjadi dua bagian menggunakan *train_test_split* yaitu 80% untuk *data training* dan 20% untuk *data validasi*. Tabel 2 menunjukkan distribusi *dataset* yang telah di bagi.

Tabel 2. Distribusi citra data hasil split data

Nama Data	Kelas	Jumlah	Jumlah
Data Training	sehat-ringan	80	240
	sedang	80	
	berat	80	

Nama Data	Kelas	Jumlah	Jumlah
Data Validasi	sehat-ringan	20	60
	sedang	20	
	berat	20	
TOTAL		300	300

3.3. Training

Arsitektur *CNN EfficientNetB0*, yang terkenal dengan efektivitas dan kinerjanya yang luar biasa dalam aplikasi klasifikasi gambar, digunakan untuk tahap pelatihan model selanjutnya. Untuk meningkatkan akurasi dan mengurangi waktu pelatihan, model ini diterapkan dengan menggunakan metodologi *transfer learning*. Untuk memungkinkan penyesuaian untuk klasifikasi tiga kelas, model *EfficientNetB0* dimuat dengan bobot yang telah dilatih sebelumnya dari ImageNet tetapi tidak memiliki lapisan atas. Lebih sedikit data dan waktu pelatihan ulang yang dibutuhkan karena bobot yang telah dilatih sebelumnya memungkinkan model untuk menggunakan karakteristik yang sebelumnya telah dipelajari dari data set yang cukup besar.

Di atas *EfficientNetB0*, sejumlah lapisan khusus ditambahkan untuk memodifikasi model untuk klasifikasi tiga kelas, lapisan *Dense* dengan ukuran 1024 dengan aktivasi *ReLU* untuk mengekstrak fitur tambahan dari gambar seperti *Batch Normalization* untuk menstabilkan dan mempercepat pelatihan, *Dropout* sebesar 0,5 untuk mengurangi *overfitting*, lapisan *output Dense* dengan tiga unit dan fungsi aktivasi *softmax* untuk klasifikasi multi-kelas, dan *GlobalAveragePooling2D* untuk meratakan fitur dari lapisan konvolusi. Lapisan awal *EfficientNetB0* mempertahankan bobot yang telah dilatih sebelumnya, tetapi 50 lapisan terakhir dijadwalkan untuk dilatih ulang. Tahap ini mempertahankan fitur-fitur dasar yang telah dipelajari ImageNet sambil memungkinkan model untuk mempelajari fitur-fitur yang lebih khusus dari kumpulan data.

Dengan *learning rate* awal $1e-4$, model disusun menggunakan pengoptimal Adam. Metrik akurasi digunakan untuk menilai kinerja model, dan fungsi *loss*, *sparse categorical crossentropy*, sesuai untuk klasifikasi multi-kelas dengan input label numerik. Augmentasi data menggunakan *ImageDataGenerator* diterapkan pada data pelatihan untuk meningkatkan variasi data pelatihan dan meningkatkan kapasitas model untuk generalisasi. Teknik augmentasi ini mencakup perubahan skala dan kecerahan, yang menambahkan variasi acak pada gambar, rotasi hingga 20 derajat, dan pergeseran hingga 30% pada arah horizontal dan vertikal.

Untuk mengelola *learning rate*, menghindari *overfitting*, dan mempertahankan bobot model yang optimal, beberapa *callback* digunakan selama proses pelatihan. *ReduceLROnPlateau* mengurangi *learning rate* jika kehilangan validasi tidak menjadi lebih baik. *EarlyStopping* memulihkan bobot optimal dan menghentikan pelatihan jika akurasi validasi tidak meningkat setelah jumlah epoch yang telah ditentukan. Bobot model yang optimal sesuai dengan akurasi validasi disimpan dalam *ModelCheckpoint*.

3.4. Evaluasi Model

Pada tahap evaluasi model, model terbaik yang tersimpan dimuat ulang dan diuji dengan data validasi untuk menentukan akurasi. Pengukuran akurasi ini memberikan indikasi awal kemampuan model dalam mengklasifikasikan tingkat keparahan penyakit. Untuk pemeriksaan yang lebih mendalam, prediksi pada data validasi dievaluasi menggunakan *confusion matrix*, yang memberikan informasi mengenai prediksi yang benar dan salah untuk setiap kelas, termasuk *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN). Metrik evaluasi lainnya, seperti presisi, *recall*, dan *f1-score*, dihitung untuk setiap kelas dan disajikan dalam bentuk laporan klasifikasi.

Precision menilai akurasi model dalam mengenali kelas tertentu, *recall* mengindikasikan kapasitas model untuk mengenali semua gambar dalam kelas tersebut, dan *f1-score* memberikan keseimbangan antara *precision* dan *recall*. Hasil pengujian menunjukkan bahwa model memiliki akurasi validasi sekitar 90%, yang menunjukkan kinerja yang tinggi dalam kategorisasi penyakit leaf blast pada tanaman padi. *Confusion matrix* dan *classification report* menunjukkan bahwa model mengenali kelas sehat-ringan dengan akurasi yang tinggi tetapi kesulitan untuk mengenali kelas parah dan sedang, yang menunjukkan bahwa model perlu ditingkatkan lebih lanjut untuk meningkatkan akurasi pada kelas-kelas yang lebih sulit.

4. HASIL DAN PEMBAHASAN

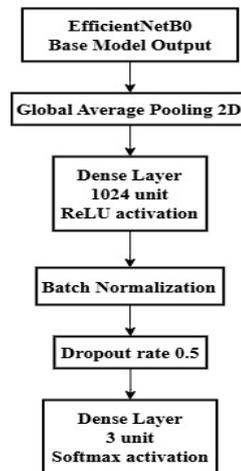
4.1. Hasil Training

Jika Tahap augmentasi data mengikuti tahap persiapan data, yang meliputi penskalaan gambar dan mempartisi data ke dalam set pelatihan dan validasi. Augmentasi data digunakan untuk meningkatkan variasi dalam *set training*, sehingga model dapat belajar dengan lebih baik.

Dalam penelitian ini, augmentasi data dilakukan dengan menggunakan fitur *ImageDataGenerator* dari Keras. Beberapa parameter yang digunakan adalah *shear_range* = 0.3 untuk menambahkan efek geseran pada gambar dan meningkatkan variasi orientasi, *brightness_range* = [0.8, 1.2] untuk mengubah kecerahan gambar dan membuat model lebih kuat terhadap perubahan pencahayaan, *rotation_range* = 20 untuk merotasi gambar secara acak hingga 20 derajat, *zoom_range* = 0.3 untuk memperbesar atau memperkecil gambar secara acak, serta *horizontal_flip* = *True* dan *vertical_flip* = *True* untuk membalikkan gambar secara horizontal dan vertikal. Fungsi *ImageDataGenerator* akan terus menghasilkan gambar baru yang diperbesar selama fase pelatihan, sehingga model dapat belajar dari data yang lebih luas.

Selanjutnya, proses pelatihan model dilakukan. Arsitektur *EfficientNetB0*, dengan bobot yang dilatih pada *dataset ImageNet* (*transfer learning*), menjadi dasar dari penelitian ini. Agar model dapat belajar secara lebih eksplisit dari *dataset* penyakit tanaman padi, lapisan teratas dari model *EfficientNetB0* (lapisan teratas) dibuka untuk pelatihan (*fine-tuning*). Selain itu, berbagai lapisan ditambahkan di atas lapisan teratas *EfficientNetB0*, seperti *GlobalAveragePooling2D*, *Dense 1024* dengan aktivasi *ReLU*,

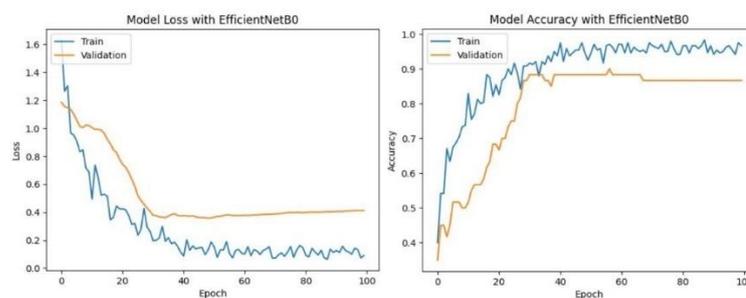
BatchNormalization, Dropout 0.5, dan lapisan keluaran Dense 3 dengan aktivasi Softmax berdasarkan tiga jenis penyakit. Gambar 2 menunjukkan arsitektur model.



Gambar 2. Arsitektur model dengan custom layer

Proses *training* berlangsung selama 100 epoch dengan ukuran *batch* 32, menggunakan pengoptimal Adam dengan *learning rate awal* $1e-4$. Beberapa *callback* juga ditambahkan untuk meningkatkan stabilitas pelatihan, seperti *ReduceLROnPlateau* untuk menurunkan *learning rate*, *EarlyStopping* untuk menghentikan pelatihan jika metrik validasi tidak membaik, dan *ModelCheckpoint* untuk menyimpan model dengan *val_accuracy* tertinggi.

Setelah proses *training* selesai, model terbaik yang disimpan dimuat ulang dan dievaluasi. Pada pemeriksaan akhir, model tersebut memiliki akurasi validasi 90% dan akurasi *training* 88,99%. Visualisasi akurasi dan kerugian juga disediakan untuk menunjukkan pola pelatihan dan validasi. Gambar 3 menunjukkan grafik hasil *training*.



Gambar 3. Grafik *accu*cy dan *loss* hasil *training*

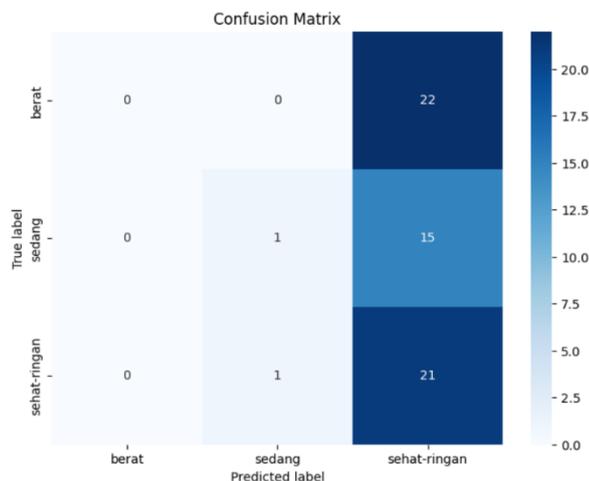
4.2. Hasil Evaluasi Model

Visualisasi dilakukan dengan menggunakan *confusion matrix* untuk mempermudah evaluasi model dan hasil pelatihan. Dari 22 gambar penyakit dengan tingkat keparahan "berat" yang digunakan sebagai data uji, tidak ada satu pun yang berhasil diidentifikasi dengan benar, menurut tabel *confusion matrix*. Setiap gambar yang seharusnya diklasifikasikan sebagai "berat" justru diprediksi sebagai "sehat-ringan", menunjukkan bahwa model kesulitan menemukan gambar dengan tingkat keparahan

penyakit yang berat sebaliknya, mereka lebih sering mengklasifikasikan kondisi tersebut sebagai kategori yang lebih ringan.

Selain itu, dari 16 gambar yang seharusnya termasuk dalam kelas "sedang", hanya satu gambar yang diprediksi dengan benar oleh model, sedangkan lima gambar lainnya diklasifikasikan sebagai "sehat-ringan", menunjukkan bahwa meskipun model mampu mengidentifikasi sebagian kecil dari kelas "sedang", sebagian besar data dengan keparahan sedang juga diklasifikasikan sebagai kondisi yang lebih ringan.

Dari 22 gambar yang seharusnya masuk dalam kelas "sehat-ringan", model mampu mengidentifikasi 21 gambar dengan benar sebagai "sehat-ringan", tetapi satu gambar dianggap salah sebagai "sedang". Hasil ini menunjukkan bahwa, meskipun masih ada kesalahan prediksi, model masih dapat mengidentifikasi kelas "sehat-ringan" dengan cukup baik dibandingkan dengan kelas lainnya. Gambar 4 menunjukkan table *confusion matrix*



Gambar 4. Tabel *confusion matrix*

Secara keseluruhan, model mendeteksi kelas "sehat-ringan" dengan lebih akurat daripada kelas "berat" dan "sedang". Namun, ada bias dalam model ini yang menyebabkan banyak data dengan tingkat keparahan yang lebih tinggi diklasifikasikan sebagai "sehat-ringan", yang menunjukkan bahwa model mungkin sulit membedakan kondisi penyakit yang lebih parah.

Pengamatan dilakukan pada *classification report* untuk mengetahui *precision*, *recall*, dan *f1-score* pada masing-masing kelas untuk melihat lebih detail kinerja model dalam memprediksi tingkat keparahan penyakit pada tanaman padi.

Precision adalah parameter yang digunakan untuk mengukur ketepatan model dalam memprediksi kelas tertentu, yaitu seberapa banyak dari prediksi yang dibuat oleh model yang benar. *Recall*, atau sensitivitas, adalah parameter yang digunakan untuk mengukur seberapa baik model mendeteksi data dari kelas yang seharusnya. *F1-score*, yang berfungsi sebagai rata-rata harmonis dari *precision* dan *recall*, berguna untuk mengevaluasi seberapa seimbang ketepatan dan sensitivitas model.

Berdasarkan tabel *classification report*, untuk kelas “berat” model memiliki nilai *precision* sebesar 0.00, *recall* sebesar 0.00, dan *f1-score* sebesar 0.00 dengan jumlah sampel sebanyak 22 sampel. Hal ini mengindikasikan bahwa model tidak dapat mengidentifikasi kelas “berat” sama sekali, sehingga tidak ada sampel yang diklasifikasikan dengan benar dalam kategori ini. Kurangnya prediksi pada kelas “berat” ini mencerminkan kurangnya efektivitas model dalam mengenali kelas dengan tingkat keparahan tinggi.

Pada kelas “sedang”, model memiliki *presisi* 0,50, *recall* 0,06, dan *f1-score* 0,11 dari total 16 sampel. Nilai presisi yang relatif tinggi untuk kelas ini mengindikasikan bahwa jika model memprediksi “sedang”, sekitar 50% dari prediksi tersebut benar. Namun, nilai *recall* yang rendah (0.06) menunjukkan bahwa model hampir tidak dapat mendeteksi sebagian besar data dari kelas “sedang”. Hanya beberapa sampel “sedang” yang benar-benar terdeteksi dengan benar oleh model.

Sementara itu, pada kelas “sehat-ringan”, model memperoleh nilai presisi sebesar 0,36, nilai *recall* sebesar 0,95, dan nilai *f1-score* sebesar 0,53 dari total 22 sampel. Nilai *recall* yang tinggi (0.95) menunjukkan bahwa model dapat mendeteksi sebagian besar sampel “sehat-ringan” dengan baik. Namun, nilai *precision* yang cukup rendah (0.36) mengindikasikan bahwa tidak semua prediksi “sehat-ringan” benar, karena ada beberapa data dari kelas lain yang salah diprediksi sebagai “sehat-ringan”.

Secara keseluruhan, model ini mencapai tingkat akurasi 37% di seluruh 60 set data uji. Akurasi rata-rata makro adalah 0.29, *recall* 0.34, dan *f1-score* 0.21, yang mengindikasikan bahwa model memiliki kinerja yang sama di setiap kelas. Nilai akurasi rata-rata tertimbang adalah 0.27, *recall* 0.37, dan *f1-score* 0.22, yang memperhitungkan proporsi sampel di setiap kelas. Hasil ini menunjukkan bahwa model ini kurang baik dalam mendeteksi kelas “parah” dan “sedang”, tetapi berkinerja baik dalam membedakan kelas “sehat-ringan”. Gambar 5 menunjukkan *classification report*.

Classification Report:				
	precision	recall	f1-score	support
berat	0.00	0.00	0.00	22
sedang	0.50	0.06	0.11	16
sehat-ringan	0.36	0.95	0.53	22
accuracy			0.37	60
macro avg	0.29	0.34	0.21	60
weighted avg	0.27	0.37	0.22	60

Gambar 5. *Classification Report*

Model *EfficientNetB0* dengan prapemrosesan CLAHE menunjukkan peningkatan yang signifikan dalam akurasi validasi, mencapai 90%. Hasil ini menyoroti potensi pendekatan yang diusulkan dalam mengatasi keterbatasan metode manual, yang dikenal subjektif dan memakan waktu. Namun, hasil *confusion matrix* menunjukkan adanya potensi masalah dengan bias model, di mana kelas “sehat-ringan” menunjukkan

akurasi yang lebih besar daripada kelas “sedang” dan “berat”. Hal ini menyoroti perlunya peningkatan kemampuan untuk mengenali pola-pola yang rumit.

Dibandingkan dengan temuan Imam dkk (2023), yang menggunakan *MobileNetV2* dan mencapai tingkat akurasi 78,33%, pendekatan ini menunjukkan peningkatan yang signifikan. Hal ini dikuatkan oleh kemampuan model dan peningkatan kualitas gambar yang dihasilkan oleh CLAHE. Namun demikian, dataset yang terbatas (terdiri dari 300 gambar) dan bias antar-kelas merupakan tantangan yang perlu diperhatikan, seperti melalui penerapan penyeimbangan dataset atau *weighted loss*.

5. KESIMPULAN

Penelitian ini menunjukkan bahwa arsitektur EfficientNetB0 yang dioptimalkan dengan CLAHE dan augmentasi data menggunakan ImageDataGenerator berhasil meningkatkan akurasi validasi hingga 90% dalam klasifikasi tingkat keparahan *leaf blast*. Model ini unggul pada kelas "sehat-ringan" tetapi kesulitan mengidentifikasi kelas "sedang" dan "parah," menunjukkan bias yang perlu diperbaiki. Meskipun pendekatan ini membantu mengatasi keterbatasan penilaian manual, ketidakmampuan model mengenali pola penyakit yang kompleks menegaskan perlunya *dataset* yang lebih seimbang dan metode yang lebih adaptif. Pengembangan ini diharapkan dapat meningkatkan kinerja model untuk aplikasi pertanian yang lebih efektif.

Penelitian ini menunjukkan kemampuan penggunaan EfficientNetB0 bersama dengan CLAHE dan penambahan data. Namun, masih ada tantangan dalam mengidentifikasi secara akurat kelas “sedang” dan “parah”. Untuk memajukan bidang penelitian ini, beberapa jalan untuk penyelidikan di masa depan direkomendasikan, termasuk: Peningkatan Dataset : Menggunakan dataset yang lebih besar dan beragam untuk meningkatkan kemampuan generalisasi model, terutama pada kelas dengan keparahan tinggi; Teknik Balancing Data : Menerapkan metode seperti *weighted loss* atau *oversampling* untuk mengatasi ketidakseimbangan data antar kelas; Eksplorasi Model Lain : Membandingkan performa EfficientNetB0 dengan arsitektur lain seperti ResNet atau DenseNet untuk memahami keunggulan masing-masing. Langkah-langkah ini diharapkan dapat meningkatkan akurasi dan mengurangi bias model, sehingga lebih optimal untuk implementasi di lapangan.

DAFTAR PUSTAKA

- [1] L. Trihardianingsih, A. Sunyoto, and T. Hidayat, “Classification of Tea Leaf Diseases Based on ResNet-50 and Inception V3,” *Sinkron*, vol. 8, no. 3, pp. 1564–1573, 2023, doi: 10.33395/sinkron.v8i3.12604.
- [2] Galih Wasis Wicaksono and Andreawan, “ResNet101 Model Performance Enhancement in Classifying Rice Diseases with Leaf Images,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 345–352, 2023, doi: 10.29207/resti.v7i2.4575.
- [3] T. Agustin, M. H. Purwiantoro, E. Utami, and H. A. Fatta, “CNN and SVM Combination for Multi-Class Classification of Diabetic Retinopathy Based on Fundus Imaging,” *Telematika*, vol. 15, no. 2, pp. 108–119, 2022.

- [4] S. Basir, N. A. Abdul Aziz, and N. Z. Abiddin, "Semantic Segmentation of Paddy Parcels Using Deep Neural Networks Based on DeepLabV3," *Int. Conf. Geogr. Inf. Syst. Theory, Appl. Manag. GISTAM - Proc.*, no. Gistam, pp. 173–180, 2024, doi: 10.5220/0012698200003696.
- [5] P. Sitompul, H. Okprana, A. Prasetyo, and G. Artikel, "Identifikasi Penyakit Tanaman Padi Melalui Citra Daun Menggunakan DenseNet 201," *JOMLAI J. Mach. Learn. Artif. Intell.*, vol. 1, no. 2, pp. 143–150, 2022, doi: 10.55123/jomlai.v1i2.889.
- [6] D. Nurfadhillah, G. Santoso, Fatimah, G. Murti Wibowo, Darmini, and Nuryatno, "Effectiveness of Automatic Detection of Osteoarthritis using Convolutional Neural Network (CNN) Method with DenseNet201 on Digital Images of Knee Joint Radiography," *E3S Web Conf.*, vol. 448, pp. 1–10, 2023, doi: 10.1051/e3sconf/202344802052.
- [7] 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, 2023, doi: 10.21111/fij.v8i1.9419.
- [8] K. Albarrak, Y. Gulzar, Y. Hamid, A. Mehmood, and A. B. Soomro, "A Deep Learning-Based Model for Date Fruit Classification," *Sustain.*, vol. 14, no. 10, 2022, doi: 10.3390/su14106339.
- [9] P. N. Srinivasu, J. G. Sivasai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, vol. 21, no. 8, pp. 1–27, 2021, doi: 10.3390/s21082852.
- [10] Y. Deng *et al.*, "Precision Detection of Salt Stress in Soybean Seedlings Based on Deep Learning and Chlorophyll Fluorescence Imaging," *Plants*, vol. 13, no. 15, 2024, doi: 10.3390/plants13152089.
- [11] A. P. Syahputra, A. C. Siregar, and R. W. S. Insani, "Comparison of CNN Models With Transfer Learning in the Classification of Insect Pests," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 17, no. 1, p. 103, 2023, doi: 10.22146/ijccs.80956.
- [12] T. Nurmayanti, D. Hartini, T. Rohana, S. Arum, P. Lestari, and D. Wahiddin, "Comparison of K-Nearest Neighbors and Convolutional Neural Network Algorithms in Potato Leaf Disease Classification," *J. Sist. Inf. dan Ilmu Komput. Prima*, vol. 8, no. 1, pp. 360–372, 2024.
- [13] G. S. Nugraha, M. I. Darmawan, and R. Dwiyanaputra, "Comparison of CNN's Architecture GoogleNet, AlexNet, VGG-16, Lenet -5, Resnet-50 in Arabic Handwriting Pattern Recognition," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 2, 2023, doi: 10.22219/kinetik.v8i2.1667.
- [14] A. Rachmatullah Pratama and A. F. Cobantoro, "Klasifikasi Citra Pneumonia Menggunakan Arsitektur Convolutional Neural Network (Cnn) Pneumonia Image Classificati on Using Convolutinal Neural Network (Cnn) Architecture," *J. Ilm. NERO*, vol. 8, no. 2, p. 2023, 2023.
- [15] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, 2023, doi: 10.1016/j.patcog.2023.109347.
- [16] P. Yang and X. Zhang, "A Dual-Branch Fusion of a Graph Convolutional Network and a Convolutional Neural Network for Hyperspectral Image Classification," *Sensors*, vol. 24, no. 14, 2024, doi: 10.3390/s24144760.

- [17] K. Jabeen *et al.*, "Breast Cancer Classification from Ultrasound Images Using Probability-Based Optimal Deep Learning Feature Fusion," *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22030807.
- [18] Z. Liu *et al.*, "Deep learning based brain tumor segmentation: a survey," *Complex Intell. Syst.*, vol. 9, no. 1, pp. 1001–1026, 2023, doi: 10.1007/s40747-022-00815-5.
- [19] T. T. Nguyen *et al.*, "Deep learning for deepfakes creation and detection: A survey," *Comput. Vis. Image Underst.*, vol. 223, 2022, doi: 10.1016/j.cviu.2022.103525.
- [20] Y. S. Taspinar, I. Cinar, and M. Koklu, "Classification by a stacking model using CNN features for COVID-19 infection diagnosis," *J. Xray. Sci. Technol.*, vol. 30, no. 1, pp. 73–88, 2022, doi: 10.3233/XST-211031.