

Implementasi EfficientNet-B4 untuk Klasifikasi Penyakit Daun Tanaman Kentang

^{1*}Muchamad Fajar Shodiq, ²Danar Putra Pamungkas, ³Patmi Kasih

^{1,2,3} Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹fajarshodiq700@gmail.com, ²danar@unpkediri.ac.id, ³fatkasi@gmail.com

Penulis Korespondens : Muchamad Fajar Shodiq

Abstrak— Kentang (*Solanum tuberosum* L.) merupakan tanaman hortikultura penting di Indonesia, namun rentan terhadap serangan penyakit. Identifikasi dini diperlukan agar petani dapat melakukan pengendalian secara tepat. Metode tradisional identifikasi penyakit umumnya memerlukan keahlian khusus dan waktu yang lama. Pemanfaatan teknologi, khususnya pengolahan citra digital untuk mendeteksi penyakit tanaman. Penelitian ini bertujuan untuk mengklasifikasikan penyakit daun kentang menggunakan model EfficientNet-B4. Data berupa citra daun kentang dengan tujuh kategori kondisi yang dikumpulkan di lingkungan tidak terkontrol. Model diuji dalam tiga skenario: 20 epoch, 50 epoch, dan 20 epoch dengan augmentasi. Evaluasi dilakukan menggunakan akurasi, presisi, recall, dan F1-score. Hasil terbaik diperoleh pada skenario 20 epoch tanpa augmentasi dengan akurasi 78,90%, presisi 79,82%, recall 78,90%, dan F1-score 79,05%. Skenario 50 epoch mengalami overfitting, sedangkan augmentasi tidak meningkatkan performa secara signifikan. Hasil ini menunjukkan potensi EfficientNet-B4 dalam klasifikasi penyakit daun kentang.

Kata Kunci— daun kentang, deep learning, EfficientNet-B4, klasifikasi citra

Abstract— Potato (*Solanum tuberosum* L.) is an important horticultural crop in Indonesia but is vulnerable to disease attacks. Early identification is necessary for farmers to take appropriate control measures. Traditional methods for disease identification generally require special expertise and are time-consuming. The utilization of technology, especially digital image processing, offers a way to detect plant diseases. This study aims to classify potato leaf diseases using the EfficientNet-B4 model. The data consist of potato leaf images with seven condition categories collected in uncontrolled environments. The model was tested under three scenarios: 20 epochs, 50 epochs, and 20 epochs with augmentation. Evaluation was performed using accuracy, precision, recall, and F1-score. The best results were obtained in the 20-epoch scenario without augmentation, with an accuracy of 78.90%, precision of 79.82%, recall of 78.90%, and F1-score of 79.05%. The 50-epoch scenario experienced overfitting, while augmentation did not significantly improve performance. These results demonstrate the potential of EfficientNet-B4 in classifying potato leaf diseases.

Keywords— deep learning, EfficientNet-B4, image classification, plant disease, potato leaf

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Kentang (*Solanum tuberosum* L.) merupakan tanaman hortikultura penting di Indonesia. Membudidayakan tanaman kentang terdapat beberapa hal yang perlu diperhatikan dan tidak terlepas dari permasalahan terserang penyakit [1]. Identifikasi dini dan akurat terhadap penyakit ini dapat membantu petani mengambil tindakan yang tepat untuk mengendalikan penyebaran penyakit dan mengurangi kerugian. Beragamnya penyakit yang dapat terjadi pada kentang tentunya membuat para petani kesulitan untuk mengidentifikasi penyakit yang me-

nyerang tanaman kentang. Hal ini karena penyakit pada kentang memiliki gejala dan pengobatan berbeda dari tanaman lainnya [2].

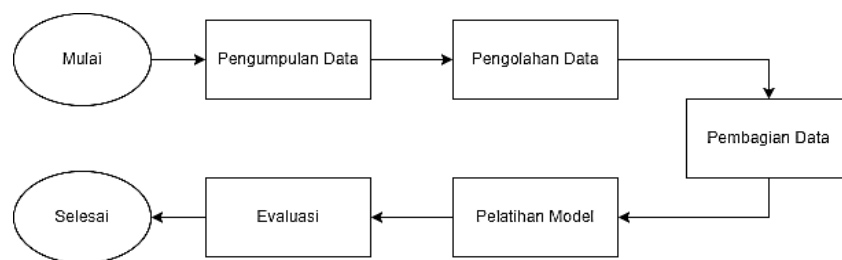
Dalam menangani masalah pada penyakit pada daun kentang ini telah banyak dilakukan, tidak hanya didalam bidang pertanian saja tetapi pada bidang teknologi pun turut ikut andil, salah satunya adalah pemanfaatan bidang informatika dalam mengidentifikasi penyakit yang ada pada tanaman kentang dengan menggunakan image processing atau biasa disebut pengolahan citra digital[3]. Dengan berkembangnya teknologi saat ini, telah banyak penelitian yang mengembangkan pengolahan citra digital dalam bidang pertanian untuk mengidentifikasi penyakit tanaman.

Penelitian sebelumnya telah menerapkan berbagai model Convolutional Neural Network (CNN) untuk mengklasifikasikan penyakit tanaman dengan hasil yang beragam. Misalnya, Penelitian Sabrina dan rekannya mengembangkan dataset Potato Leaf Disease Dataset in Uncontrolled Environment yang mencerminkan kondisi nyata di lapangan dengan hasil model CNN EfficientNetV2-B3 berhasil mencapai akurasi tertinggi sebesar 73,63%, diikuti oleh MobileNetV3-Large 72,03%, ResNet50 68,17%, VGG-16 59,81%, dan DenseNet121 59,16%[4]. model EfficientNet-B4 dengan fine-tuning berhasil mengklasifikasikan penyakit pada citra kakao dengan akurasi mencapai 97,3%[5], sementara model yang sama dengan optimizer Ranger menghasilkan akurasi sebesar 96% pada klasifikasi penyakit daun mentimun[6]. Model EfficientNet-B6 diterapkan untuk klasifikasi penyakit daun padi mencapai akurasi tertinggi sebesar 77,05[7], sedangkan model VGG-19 berhasil mencapai akurasi 93,18% untuk klasifikasi penyakit pada daun padi[8]. Selain itu, model AlexNet menunjukkan akurasi sebesar 90% pada klasifikasi penyakit daun jagung[9].

Berdasarkan uraian tersebut EfficientNet-B4 menunjukkan akurasi yang tinggi, penelitian ini bertujuan untuk mengidentifikasi penyakit pada daun kentang serta mengevaluasi akurasi metode EfficientNet-B4 dalam melakukan klasifikasi, dengan menggunakan dataset *Potato Leaf Disease in Uncontrolled Environment* sebagai data pelatihan.

II. METODE

Pada penelitian ini, proses klasifikasi penyakit daun tanaman kentang dilakukan melalui beberapa tahapan yang digambarkan dalam Gambar 1 berikut :






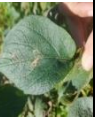



Gambar 1. Kerangka Kerja Penelitian

Gambar 1 tersebut menunjukkan kerangka kerja penelitian, yang terdiri dari 5 tahap utama, yaitu : pengumpulan data, pengolahan data untuk menyesuaikan citra dengan kebutuhan arsitektur model, pembagian data untuk membagi data menjadi data pelatihan, data validasi dan data testing, serta evaluasi untuk mengukur performa model dalam mengklasifikasi penyakit daun tanaman kentang. Masing-masing tahapan kerangka kerja penelitian akan dijelaskan lebih lanjut sebagai berikut :

A. Pengumpulan Data

Dataset pada penelitian ini diambil dari penelitian sabrina yang bersifat publik <https://data.mendeley.com/datasets/ptz377bwb8/1>, dataset tersebut bernama Potato Leaf Disease in Uncontrolled Environment terdiri dari total 3.076 citra daun kentang dengan ukuran 1500px x 1500px yang terbagi dalam 7 kelas, yaitu: Bacteria sebanyak 569 gambar, Fungi 748 gambar, Healthy (sehat) 201 gambar, Nematode 68 gambar, Pest (hama) 611 gambar, Phytophthora 347 gambar, dan Virus 532 gambar, contoh dataset pada tabel 1 berikut :

Tabel 1. Dataset

Jenis Penyakit	Virus	Bakteri	Jamur	Hama	Nematoda	Phytophthora	Sehat
Gambar Daun Kentang							

B. Pengolahan Data

Pada penelitian ini data diolah dengan tahap preprocessing data dan pembagian data

1. Preprocessing Data

Pada tahap pengolahan data, citra terlebih dahulu diubah ukurannya menjadi 380×380 piksel agar sesuai dengan arsitektur model yang digunakan. Selanjutnya, dilakukan normalisasi terhadap nilai piksel untuk memastikan data berada dalam rentang yang sesuai dan mempercepat proses pelatihan model.

2. Pembagian Data

Tabel 2. Pembagian Data

Dataset	Data Training	Data Validation	Data Testing
Bakteri(bacteria)	448	45	76
Jamur (Fungi)	602	69	77
Sehat (Healthy)	158	25	18
Nematode	55	7	6
Hama(Pest)	493	57	61
Busuk Daun (Phytophthora)	281	43	23
Virus	423	62	47

Dataset dibagi menjadi tiga subset, yaitu data latih (training) sebesar 80%, data validasi (validation) sebesar 10%, dan data uji (testing) sebesar 10%. Pembagian ini bertujuan untuk melatih model, mengevaluasi kinerja selama proses pelatihan, serta menguji performa model terhadap data yang belum pernah dilihat sebelumnya.

C. Pelatihan Model

Pelatihan model dilakukan menggunakan EfficientNet, Efficient-Net adalah suatu arsitektur CNN yang ditemukan dengan melakukan scaling secara teratur pada tiga komponen, yaitu kedalaman, lebar, dan resolusi[10]. Pada penelitian ini memanfaatkan *EfficientNet-B4* sebagai ekstraktor fitur. Lapisan EfficientNet-B4 tidak dilatih ulang (non-trainable) dengan jumlah parameter sebesar 17,6 juta. Di atasnya ditambahkan *Dense layer* sebanyak dua lapis—lapisan pertama memiliki 1.024 neuron, dilengkapi *Dropout*, dan dilanjutkan dengan lapisan output berjumlah 7 neuron sesuai dengan jumlah

kelas. Total parameter model mencapai 19,5 juta, dengan sekitar 1,8 juta parameter yang dilatih selama proses training.

D. Evaluasi

Tahap evaluasi akurasi dilakukan dengan menggunakan matriks klasifikasi yaitu akurasi, presisi, recall dan F1-Score dengan rumus sebagai berikut.

1. Accuracy

Accuracy merupakan rasio prediksi benar (positif, negatif) dengan keseluruhan data.

Keterangan :

TP = jumlah data positif yang terklasifikasi benar.

TN = jumlah data negatif yang terklasifikasi benar.

FN = jumlah data negatif namun terklasifikasi salah.

FP = jumlah data positif namun terklasifikasi salah.

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

2. Precision

Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.

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

Keterangan :

TP = jumlah data positif yang terklasifikasi benar.

FP = jumlah data positif namun terklasifikasi salah.

3. Recall

Recall Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

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

Keterangan :

TP = jumlah data positif yang terklasifikasi benar.

FN = jumlah data negatif namun terklasifikasi salah.

4. F1-Score

F1-Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan.

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

II. HASIL DAN PEMBAHASAN

A. Implementasi Desain Sistem

1. Input Dataset

Contoh Gambar per Kelas - Total: 3076



Gambar 2. Input Dataset

Tahapan pertama adalah memanggil dataset gambar daun kentang yang terdiri dari 3.076 gambar dengan 7 kategori berbeda. Dataset ini menjadi bahan utama yang akan diproses oleh sistem. Visualisasi pada gambar 2 memperlihatkan beberapa contoh citra dari dataset asli beserta label dan jumlah data pada setiap kelas.

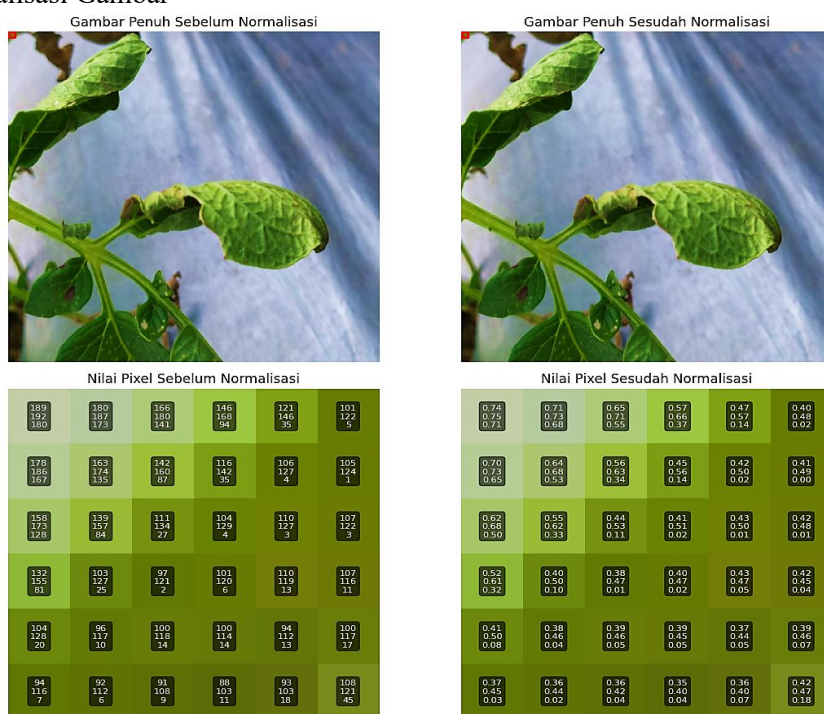
2. Resize Gambar



Gambar 3. Resize Data

Gambar 3 menunjukkan perbandingan antara gambar asli dan gambar hasil resize. Gambar dalam dataset awalnya memiliki resolusi yang tinggi, yaitu 1500x1500 piksel. Oleh karena itu, gambar diubah ukurannya menjadi 380x380 piksel, yang merupakan ukuran input standar dari arsitektur EfficientNet-B4. Proses resize ini bertujuan untuk menyesuaikan gambar dengan arsitektur model sekaligus menghemat sumber daya komputasi tanpa kehilangan informasi penting dari gambar.

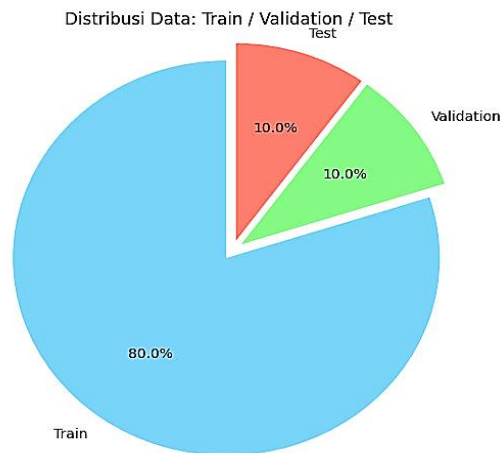
3. Normalisasi Gambar



Gambar 4. Normalisasi Data

Gambar 4 menunjukkan perbandingan antara gambar sebelum dan sesudah dinormalisasi. Gambar dinormalisasi dengan cara mengubah skala piksel dari 0–255 menjadi 0–1. Tujuannya agar proses pelatihan lebih cepat dan stabil, karena model akan lebih mudah mengenali pola dari data yang memiliki skala seragam.

4. Pembagian Data



Gambar 5. Pembagian Data

Gambar 5 memperlihatkan proporsi data yang telah dibagi, ditampilkan dalam bentuk diagram pie. data dibagi menjadi tiga bagian: data pelatihan (80%), data validasi (10%), dan data pengujian (10%). Pembagian ini dilakukan secara acak namun konsisten menggunakan parameter `random_state`.

5. Training Model

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
efficientnet-b4 (Functional)	(None, 1792)	17673816
dense (Dense)	(None, 1024)	1836032
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 7)	7175

```
=====  
Total params: 19517023 (74.45 MB)  
Trainable params: 1843207 (7.03 MB)  
Non-trainable params: 17673816 (67.42 MB)  
=====
```

Gambar 6. Training Model

Model kemudian dilatih menggunakan data training yang sudah disiapkan. Dalam penelitian ini digunakan arsitektur EfficientNet-B4 seperti digambar 6, yang dikenal cukup efektif untuk klasifikasi gambar. Selama pelatihan, kinerja model juga dipantau menggunakan data validasi.

6. Evaluasi

Setelah model selesai dilatih, langkah berikutnya adalah menguji performanya menggunakan data uji. Hasil klasifikasi akan dianalisis menggunakan beberapa metrik evaluasi, seperti akurasi, presisi, recall, F1-score, serta confusion matrix.

B. Skenario Uji

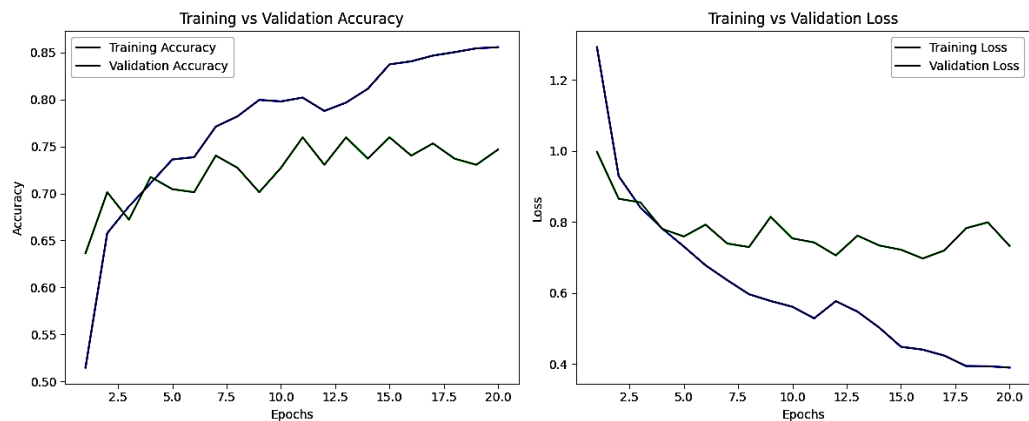
Pada bagian skenario uji disajikan hasil penelitian klasifikasi penyakit daun kentang menggunakan model EfficientNet-B4 dengan tiga skenario pelatihan yang berbeda. Setiap

skenario dirancang untuk mengevaluasi pengaruh jumlah epoch serta penggunaan teknik augmentasi terhadap performa model. Rincian skenario disajikan pada tabel 3.

Tabel 3. Skenario Pengujian

Skenario	Epoch	Augmentasi Data
1	20	× (tanpa Augmentasi)
2	50	× (tanpa Augmentasi)
3	20	✓ (Augmentasi)

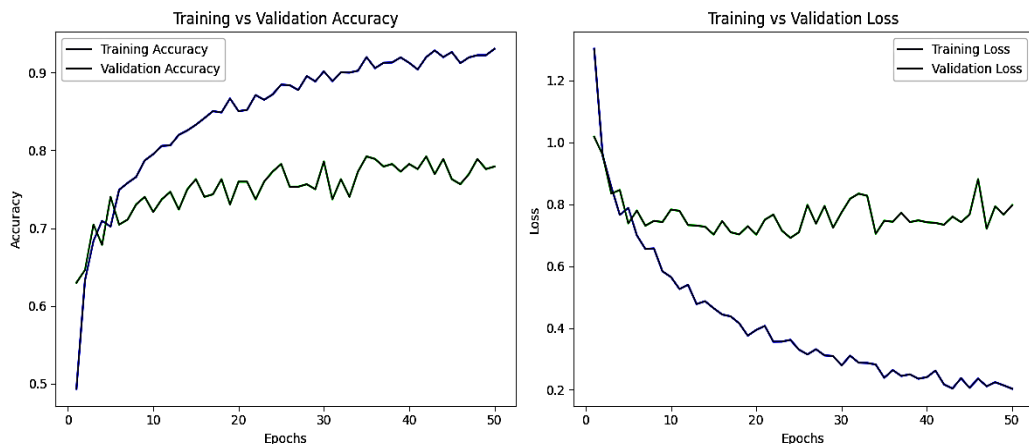
1. Skenario 1 Epoch 20



Gambar 7. Grafik Epoch 20

Skenario pertama menggunakan 20 epoch tanpa penerapan augmentasi data. Berdasarkan Gambar 2, akurasi pelatihan meningkat secara konsisten hingga mencapai 85%, sedangkan akurasi validasi cenderung stagnan di kisaran 74–75%. Meskipun begitu, akurasi testing pada konfigurasi ini mencapai 78,90%, yang merupakan hasil terbaik di antara semua konfigurasi yang diuji. Grafik *loss* menunjukkan bahwa model mulai mengalami overfitting setelah epoch ke-10, ditandai dengan perbedaan antara *training loss* dan *validation loss* yang semakin besar. Pada skenario ini, model berhasil mencapai nilai presisi sebesar 79,82%, recall 78,90%, dan F1-score 79,05%.

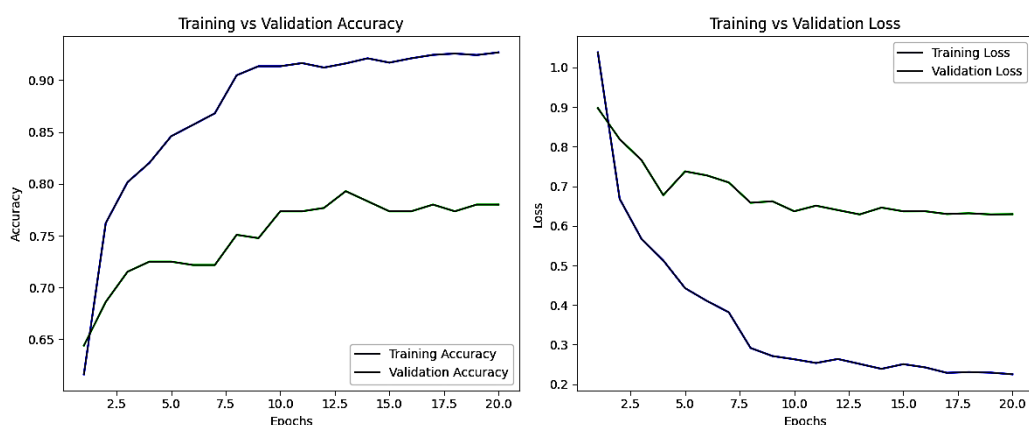
2. Skenario 2 Epoch 50



Gambar 8. Grafik Epoch 50

Pada skenario kedua, model dilatih selama 50 epoch. Akurasi pelatihan mengalami peningkatan hingga lebih dari 92%, namun akurasi validasi tidak menunjukkan peningkatan yang signifikan dibandingkan konfigurasi sebelumnya serta akurasi testing sedikit menurun menjadi 78,57%. Grafik *loss* menunjukkan bahwa model mengalami overfitting yang lebih parah, ditandai dengan *training loss* yang terus menurun sedangkan *validation loss* cenderung stagnan dan fluktuatif. Hal ini mengindikasikan bahwa pelatihan terlalu lama justru menurunkan kemampuan generalisasi model. Pada skenario ini, model memperoleh presisi sebesar 79,87%, recall 78,57%, dan F1-score 78,62%.

3. Skenario 3 Epoch 20 + Augmentasi Data



Gambar 9. Grafik Epoch 20 + Augmentasi Data

Skenario ketiga menerapkan teknik augmentasi data seperti rotasi, flipping, zooming, dan shifting, dengan tujuan meningkatkan kemampuan generalisasi model. Akan tetapi, pada pelatihan selama 20 epoch, akurasi validasi justru menurun menjadi 76,53% dengan presisi 77,22%, recall 76,53%, dan F1-score 76,44%. Berdasarkan Gambar 4, terlihat bahwa meskipun akurasi pelatihan mencapai lebih dari 93%, akurasi validasi tidak mengalami peningkatan yang sebanding. Hal ini kemungkinan disebabkan oleh kompleksitas tambahan dari data augmentasi, yang memerlukan waktu pelatihan lebih lama agar model mampu beradaptasi dengan variasi data baru.

IV. KESIMPULAN

Berdasarkan hasil yang telah didapat dan analisis performa model EfficientNet-B4 pada klasifikasi penyakit daun kentang, diperoleh beberapa kesimpulan. Model menunjukkan performa terbaik pada skenario pelatihan selama 20 epoch tanpa augmentasi, dengan akurasi 78,90%, presisi 79,82%, recall 78,90%, dan F1-score 79,05%. Penambahan jumlah epoch menjadi 50 tanpa augmentasi justru menyebabkan penurunan performa dan indikasi overfitting, meskipun nilai presisinya sedikit lebih tinggi. Penggunaan augmentasi data juga tidak meningkatkan performa secara signifikan karena model membutuhkan waktu pelatihan lebih panjang untuk dapat memahami variasi data hasil augmentasi. Dengan demikian, dapat disimpulkan bahwa model EfficientNet-B4 cukup efektif dalam mengklasifikasikan penyakit daun kentang, tetapi perlu pengaturan parameter pelatihan yang tepat agar performa dapat ditingkatkan lebih lanjut.

DAFTAR PUSTAKA

- [1] W. F. Hidayat, T. Asra, and A. Setiadi, "Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression," *Indonesian Journal on Software Engineering (IJSE)*, vol. 8, no. 2, pp. 173–179, 2022. Doi: <http://ejournal.bsi.ac.id/ejurnal/index.php/ijse>
- [2] P. T. Ompusunggu, "Klasifikasi Penyakit Tanaman pada Daun Kentang dengan Metode Convolutional Neural Network Arsitektur MobileNet," *Jurnal Fusion*, vol. 2, no. 1, pp. 1–12, 2022.
- [3] A. J. Rozaqi, A. Sunyoto, and R. Arief, "Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network," *Citec Journal*, 2021.
- [4] N. H. Shabrina, S. Indarti, R. Maharani, D. A. Kristiyanti, Irmawati, N. Prastomo, and T. M. Adilah, "A Novel Dataset of Potato Leaf Disease in Uncontrolled Environment," *Data in Brief*, vol. 52, 2023, doi: 10.1016/j.dib.2023.109955.
- [5] A. G. Pradana, D. R. I. M. Setiadi, and A. R. Muslikh, "Fine Tuning Model Convolutional Neural Network EfficientNet-B4 dengan Augmentasi Data untuk Klasifikasi Penyakit Kakao," *Journal of Information System and Application Development*, vol. 2, no. 1, pp. 1–11, 2024, doi: 10.26905/jisad.v2i1.11899.
- [6] P. Zhang, L. Yang, and D. Li, "EfficientNet-B4-Ranger: A Novel Method for Greenhouse Cucumber Disease Recognition Under Natural Complex Environment," *Comput. Electron. Agric.*, vol. 176, p. 105652, 2020, doi: 10.1016/j.compag.2020.105652.
- [7] A. C. Milano, "Klasifikasi Penyakit Daun Padi Menggunakan Model Deep Learning EfficientNet-B6," *J. Inform. dan Tek. Elektro Terapan*, vol. 12, no. 1, 2024, doi: 10.23960/jitet.v12i1.3855.
- [8] R. Shinta, Jasril, M. Irsyad, and F. Yanto, "Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19," *J. Sains dan Informatika*, vol. 9, no. 1, pp. 37–45, 2023, doi: 10.22216/jsi.v9i1.2175.
- [9] Q. N. Azizah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet," *Sudo J. Tek. Inform.*, vol. 2, no. 1, pp. 28–33, 2023, doi: 10.56211/sudo.v2i1.227.
- [10] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 10691–10700.