

ANALISIS KOMPARATIF EFEKTIVITAS OPTIMIZER PADA MODEL INCEPTIONV3 DAN EFFICIENTNETB3 MENGGUNAKAN TRANSFER LEARNING UNTUK KLASIFIKASI PENYAKIT DAUN TOMAT

Muhammad Thorikin Zuniarto, M. Faris Al Hakim

Teknik Informatika, Universitas Negeri Semarang
Sekaran, Gunungpati, Semarang, Jawa Tengah 50229, Indonesia
muhammadzuniarto22@students.unnes.ac.id

ABSTRAK

Penyakit daun tomat dapat menurunkan kualitas dan produktivitas tanaman sehingga diperlukan sistem identifikasi yang cepat dan akurat. Identifikasi penyakit secara manual masih memiliki keterbatasan karena membutuhkan waktu dan keahlian khusus dalam mengenali gejala penyakit. Penelitian ini bertujuan menganalisis dan membandingkan efektivitas optimizer pada model InceptionV3 dan EfficientNetB3 menggunakan pendekatan transfer learning untuk klasifikasi penyakit daun tomat. Dataset yang digunakan berasal dari PlantVillage sebanyak 10.000 citra dengan 10 kelas penyakit. Tahapan penelitian meliputi preprocessing, augmentasi citra, pembagian dataset dengan rasio 80:10:10, serta pelatihan model menggunakan optimizer Adam, RMSprop, SGD Momentum, AdamW, dan Nadam. Seluruh eksperimen dilakukan menggunakan konfigurasi pelatihan yang sama untuk menjaga validitas perbandingan antar optimizer. Evaluasi performa dilakukan menggunakan accuracy, precision, recall, F1-score, MCC, kappa, dan Top-3 accuracy. Hasil penelitian menunjukkan bahwa RMSprop memberikan performa terbaik dengan akurasi sebesar 99,1% pada InceptionV3 dan 99,2% pada EfficientNetB3. Selain itu, EfficientNetB3 menunjukkan performa klasifikasi yang lebih baik dibandingkan InceptionV3. Penelitian selanjutnya berpotensi mengembangkan optimasi hyperparameter dan penerapan dataset lapangan untuk meningkatkan kemampuan generalisasi model.

Kata kunci : *Deep learning, Efektivitas Optimizer, EfficientNetB3, InceptionV3, Penyakit Daun Tomat, Transfer Learning*

1. PENDAHULUAN

Perkembangan teknologi digital pada sektor pertanian semakin meningkat untuk membantu meningkatkan kualitas dan produktivitas hasil panen. Salah satu permasalahan yang sering terjadi pada tanaman tomat adalah serangan penyakit daun seperti *early blight*, *late blight*, dan *leaf mold* yang dapat menyebabkan penurunan kualitas serta hasil produksi tanaman. Identifikasi penyakit daun tomat secara manual masih memiliki keterbatasan karena membutuhkan waktu, ketelitian, dan keahlian khusus dalam mengenali gejala penyakit. Oleh karena itu, diperlukan sistem otomatis yang mampu melakukan klasifikasi penyakit daun tomat secara cepat dan akurat.

Perkembangan *deep learning*, khususnya *Convolutional Neural Network* (CNN), menunjukkan performa yang baik dalam bidang klasifikasi citra. Model arsitektur seperti InceptionV3 dan EfficientNetB3 banyak digunakan karena mampu mengekstraksi fitur citra secara efektif dengan tingkat akurasi yang tinggi. InceptionV3 memiliki kemampuan mengurangi kompleksitas komputasi melalui *factorized convolution*, sedangkan EfficientNetB3 meningkatkan performa model menggunakan metode *compound scaling* [1].

Selain itu, penggunaan *transfer learning* dapat membantu meningkatkan performa model dengan memanfaatkan bobot pelatihan sebelumnya sehingga proses pelatihan menjadi lebih efisien [2].

Optimizer menjadi salah satu faktor penting yang memengaruhi performa model *deep learning*. Optimizer berfungsi memperbarui bobot model selama proses pelatihan untuk memperoleh nilai *loss* yang optimal. Beberapa optimizer seperti Adam, RMSprop, SGD Momentum, AdamW, dan Nadam memiliki karakteristik berbeda dalam kecepatan konvergensi, stabilitas pelatihan, dan kemampuan generalisasi model [3].

Pemilihan optimizer yang tepat sangat berpengaruh terhadap hasil klasifikasi citra, terutama pada model *transfer learning* dengan arsitektur yang kompleks.

Penelitian sebelumnya menunjukkan bahwa penggunaan CNN dan *transfer learning* mampu menghasilkan akurasi tinggi pada klasifikasi penyakit daun tanaman. Penelitian lain membandingkan beberapa arsitektur CNN dan menunjukkan bahwa *EfficientNet* memiliki performa lebih baik dibandingkan model CNN konvensional. Selain itu, optimizer adaptif seperti Adam dan RMSprop diketahui mampu memberikan proses pelatihan yang lebih stabil dibandingkan optimizer konvensional. Namun, sebagian besar penelitian masih berfokus pada perbandingan arsitektur model atau peningkatan akurasi tanpa melakukan analisis mendalam terhadap pengaruh optimizer pada model InceptionV3 dan EfficientNetB3. Berdasarkan kondisi tersebut, penelitian ini memiliki kebaruan berupa analisis komparatif efektivitas beberapa optimizer modern

pada dua arsitektur CNN menggunakan pendekatan *transfer learning*.

Penelitian ini bertujuan menganalisis dan membandingkan efektivitas optimizer pada model *InceptionV3* dan *EfficientNetB3* untuk klasifikasi penyakit daun tomat. Rencana penyelesaian masalah dilakukan menggunakan dataset *PlantVillage* yang terdiri dari 10 kelas penyakit daun tomat. Tahapan penelitian meliputi *preprocessing*, augmentasi citra, pembagian data latih, validasi, dan pengujian, pelatihan model menggunakan beberapa optimizer, serta evaluasi performa berdasarkan *accuracy*, *precision*, *recall*, *F1-score*, *MCC*, *kappa*, dan *Top-3 accuracy*. Seluruh eksperimen dilakukan menggunakan konfigurasi pelatihan yang sama agar perbandingan performa antar optimizer dapat dilakukan secara objektif. Hasil penelitian diharapkan dapat memberikan rekomendasi kombinasi model dan optimizer yang optimal untuk klasifikasi penyakit daun tomat serta menjadi referensi pengembangan penelitian *deep learning* di bidang pertanian.

2. TINJAUAN PUSTAKA

Tinjauan pustaka pada penelitian ini bertujuan untuk memberikan landasan teoritis sekaligus memperkuat arah penelitian yang berfokus pada analisis komparatif efektivitas optimizer dalam model *deep learning* [4].

Pembahasan mencakup konsep *deep learning* dalam klasifikasi citra, karakteristik arsitektur *InceptionV3* dan *EfficientNetB3*, peran *transfer learning*, optimizer dalam proses pelatihan model, serta metode evaluasi performa model. Selain itu, tinjauan pustaka juga memuat penelitian terdahulu yang relevan sebagai dasar pengembangan penelitian.

2.1. Deep Learning untuk Klasifikasi Citra

Deep learning merupakan bagian dari *machine learning* yang menggunakan jaringan saraf tiruan berlapis untuk mempelajari representasi data secara otomatis. Dalam bidang klasifikasi citra, pendekatan ini mampu mengekstraksi fitur kompleks langsung dari data tanpa memerlukan ekstraksi fitur manual. Model berbasis *Convolutional Neural Network* (CNN) menjadi salah satu metode yang banyak digunakan karena mampu mengenali pola visual seperti bentuk, tekstur, dan warna secara efektif [5].

Penerapan *deep learning* pada klasifikasi penyakit tanaman menunjukkan performa yang baik dengan tingkat akurasi yang tinggi sehingga banyak digunakan dalam pengembangan sistem pertanian cerdas.

2.2. Arsitektur InceptionV3

InceptionV3 merupakan arsitektur CNN yang dikembangkan untuk meningkatkan efisiensi komputasi tanpa mengurangi performa klasifikasi. Model ini menggunakan *Inception module* yang memungkinkan ekstraksi fitur dilakukan secara paralel menggunakan berbagai ukuran kernel. Selain itu,

teknik *factorized convolution* digunakan untuk mengurangi jumlah parameter dan mempercepat proses pelatihan [6].

InceptionV3 banyak digunakan pada klasifikasi citra karena memiliki performa yang stabil, terutama ketika diterapkan menggunakan *transfer learning*.

2.3. Arsitektur EfficientNetB3

EfficientNetB3 merupakan bagian dari keluarga *EfficientNet* yang menggunakan metode *compound scaling* untuk mengoptimalkan kedalaman, lebar, dan resolusi jaringan secara bersamaan. Pendekatan ini memungkinkan model menghasilkan akurasi tinggi dengan jumlah parameter yang lebih efisien dibandingkan arsitektur CNN konvensional [7].

EfficientNetB3 dikenal memiliki keseimbangan yang baik antara performa dan efisiensi komputasi sehingga banyak digunakan pada klasifikasi citra tanaman.

2.4. Transfer Learning

Transfer learning merupakan teknik yang memanfaatkan model *pretrained* untuk menyelesaikan permasalahan baru dengan dataset yang lebih kecil. Pendekatan ini membantu mempercepat proses pelatihan dan meningkatkan performa model karena model telah memiliki kemampuan dasar dalam mengenali pola visual [8].

Pada penelitian ini, *transfer learning* diterapkan menggunakan model *pretrained InceptionV3* dan *EfficientNetB3* yang kemudian dilakukan *fine-tuning* pada beberapa layer agar sesuai dengan dataset penyakit daun tomat.

2.5. Optimizer dalam Deep Learning

Optimizer merupakan algoritma yang digunakan untuk memperbarui bobot model selama proses pelatihan guna meminimalkan nilai *loss*. Optimizer seperti Adam, RMSprop, SGD Momentum, AdamW, dan Nadam memiliki karakteristik yang berbeda dalam kecepatan konvergensi, stabilitas pelatihan, dan kemampuan generalisasi model [9].

Optimizer adaptif seperti Adam dan RMSprop umumnya mampu memberikan proses pelatihan yang lebih stabil dibandingkan optimizer konvensional sehingga sering digunakan pada model *deep learning* modern.

2.6. Evaluasi Performa Model

Evaluasi performa model klasifikasi citra dilakukan menggunakan beberapa metrik seperti *accuracy*, *precision*, *recall*, *F1-score*, *MCC*, *kappa*, dan *Top-3 accuracy*. Selain itu, *confusion matrix*, *ROC curve*, dan *precision-recall curve* digunakan untuk menganalisis performa model secara lebih mendalam [10].

Penggunaan berbagai metrik evaluasi ini penting untuk memberikan gambaran performa model secara komprehensif.

2.7. Penelitian Terdahulu

Penelitian terkait klasifikasi penyakit tanaman menggunakan *deep learning* telah banyak dilakukan menggunakan berbagai arsitektur CNN dan metode *transfer learning*. Penelitian sebelumnya menunjukkan bahwa *InceptionV3* mampu memberikan performa klasifikasi yang baik dengan efisiensi komputasi yang tinggi [11].

Penelitian lain menunjukkan bahwa *EfficientNetB3* mampu menghasilkan akurasi tinggi dengan jumlah parameter yang lebih efisien dibandingkan model CNN konvensional [12].

Beberapa penelitian juga membahas pengaruh optimizer terhadap performa *deep learning*. Optimizer adaptif seperti Adam dan RMSprop diketahui mampu memberikan stabilitas pelatihan dan konvergensi yang lebih baik dibandingkan SGD pada klasifikasi citra tanaman [13], [14].

Namun, sebagian besar penelitian masih berfokus pada penggunaan satu model atau satu optimizer tertentu. Oleh karena itu, penelitian ini dilakukan untuk menganalisis efektivitas beberapa optimizer pada model *InceptionV3* dan *EfficientNetB3* menggunakan pendekatan *transfer learning* untuk klasifikasi penyakit daun tomat.

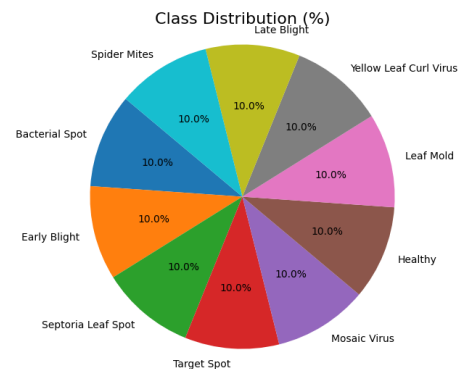
Penelitian ini menggunakan pendekatan eksperimen berbasis *deep learning* untuk menganalisis dan membandingkan efektivitas optimizer pada model *InceptionV3* dan *EfficientNetB3* dalam klasifikasi penyakit daun tomat. Tahapan penelitian meliputi pengumpulan dataset, *preprocessing* citra, augmentasi data, pembagian dataset, perancangan model, pelatihan model, serta evaluasi performa model [15].

Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1 yang menggambarkan proses penelitian mulai dari pengumpulan dataset hingga analisis hasil pengujian.

Dataset yang digunakan berasal dari *PlantVillage* yang terdiri dari 10 kelas penyakit daun tomat dengan total 10.000 citra. Dataset kemudian dibagi menjadi data latih, validasi, dan pengujian dengan rasio 80:10:10 menggunakan metode *stratified split* untuk menjaga distribusi kelas tetap seimbang. Contoh visualisasi dataset dan distribusi kelas ditunjukkan pada Gambar 2 dan Gambar 3.



Gambar 2. Visualisasi Dataset



Gambar 3. Distribusi Kelas



Gambar 1. Alur Penelitian

Dataset kemudian dibagi menjadi data latih, validasi, dan pengujian dengan rasio 80:10:10 untuk menjaga validitas proses pelatihan dan evaluasi model.

Tahap *preprocessing* dilakukan dengan melakukan *resizing* citra sesuai kebutuhan arsitektur model, yaitu 299×299 piksel untuk *InceptionV3* dan 300×300 piksel untuk *EfficientNetB3*. Selanjutnya dilakukan normalisasi menggunakan *preprocess_input* serta augmentasi citra berupa *rotation*, *zoom*, *horizontal flip*, *width shift*, *height shift*, dan *shear* untuk meningkatkan variasi data pelatihan. Setelah *preprocessing*, dataset digunakan untuk proses pelatihan menggunakan metode *transfer learning* pada model *InceptionV3* dan *EfficientNetB3*.

Pada tahap *transfer learning*, model *pretrained* menggunakan bobot *ImageNet* dengan proses *feature extraction* dan *fine-tuning*. Pelatihan dilakukan menggunakan beberapa optimizer yaitu Adam, RMSprop, SGD Momentum, AdamW, dan Nadam untuk membandingkan pengaruh optimizer terhadap performa model.

Seluruh eksperimen dilakukan menggunakan konfigurasi pelatihan yang sama untuk menjaga validitas perbandingan antar optimizer. Pembagian dataset, *preprocessing*, augmentasi citra, jumlah *epoch*, *batch size*, *loss function*, dan skema *transfer learning* dibuat konsisten pada seluruh eksperimen sehingga perbedaan performa terutama dipengaruhi oleh optimizer yang digunakan.

Seluruh eksperimen menggunakan konfigurasi pelatihan yang sama untuk menjaga validitas perbandingan antar optimizer. Model InceptionV3 menggunakan *input size* 299×299 piksel, sedangkan EfficientNetB3 menggunakan *input size* 300×300 piksel. Proses pelatihan dilakukan dengan *batch size*

32, *epoch head training* sebanyak 10, *epoch fine-tuning* sebanyak 20, *random seed* 42, *loss function* *categorical crossentropy*, serta pendekatan *transfer learning* berbasis *fine-tuning*. Dataset dibagi menjadi data latih, validasi, dan pengujian dengan rasio 80:10:10 pada seluruh eksperimen.

Evaluasi performa model dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-score*, *MCC*, *kappa*, dan *Top-3 accuracy*. Selain itu, *confusion matrix*, *ROC curve*, *precision-recall curve*, *accuracy curve*, dan *loss curve* digunakan untuk menganalisis stabilitas pelatihan dan kemampuan klasifikasi model secara lebih mendalam.

4. HASIL DAN PEMBAHASAN

4.1. Rekapitulasi Hasil Pengujian

Hasil pengujian seluruh kombinasi model dan optimizer dirangkum pada Tabel 4.1 untuk mempermudah analisis perbandingan performa berdasarkan *accuracy*, *precision*, *recall*, *F1-score*, *MCC*, *kappa*, dan *Top-3 accuracy*.

Tabel 1. Rekapitulasi Hasil Pengujian

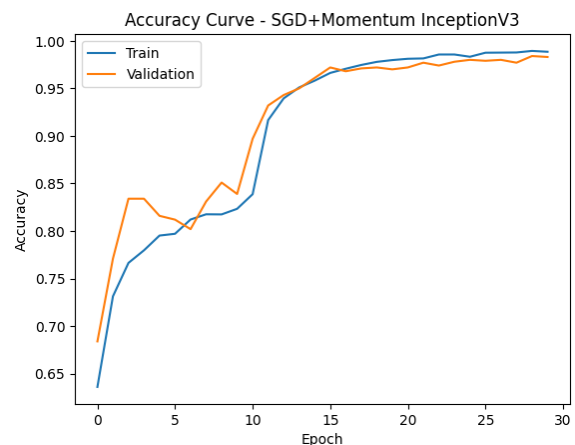
Model Inceptionv3							
Optimizer	Accuracy	Precision	Recall	F1-Score	MCC	Kappa	Top-3 Accuracy
Adam	0.986	0.986	0.986	0.986	0.984	0.984	0.998
RMSprop	0.991	0.991	0.991	0.991	0.989	0.989	1.0
SGD+Momentum	0.989	0.989	0.989	0.989	0.987	0.987	1.0
Adamw	0.988	0.988	0.988	0.988	0.986	0.986	1.0
Nadam	0.986	0.986	0.986	0.986	0.984	0.984	0.999
Model EfficientNetB3							
Optimizer	Accuracy	Precision	Recall	F1-Score	MCC	Kappa	Top-3 Accuracy
Adam	0.990	0.990	0.990	0.989	0.988	0.988	0.999
RMSprop	0.992	0.992	0.992	0.992	0.991	0.991	1.0
SGD+Momentum	0.976	0.976	0.976	0.976	0.973	0.973	0.995
Adamw	0.991	0.991	0.991	0.991	0.990	0.990	1.0
Nadam	0.990	0.990	0.990	0.990	0.988	0.988	0.999

Berdasarkan hasil pengujian, seluruh model menunjukkan performa klasifikasi yang sangat baik dengan nilai *accuracy* di atas 97%. Optimizer RMSprop memberikan hasil terbaik pada kedua model dengan *accuracy* sebesar 99,1% pada *InceptionV3* dan 99,2% pada *EfficientNetB3*. Selain itu, *EfficientNetB3* menunjukkan performa yang sedikit lebih tinggi dibandingkan *InceptionV3*, namun lebih sensitif terhadap pemilihan optimizer, terutama pada penggunaan SGD Momentum yang menghasilkan *accuracy* terendah sebesar 97,6%.

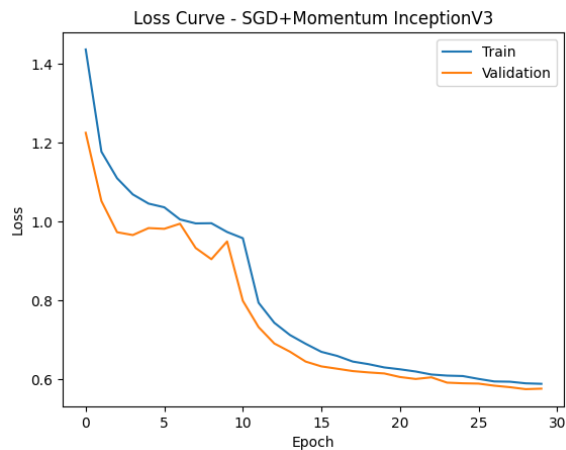
Seluruh eksperimen dilakukan menggunakan konfigurasi pelatihan yang sama sehingga perbedaan performa lebih dipengaruhi oleh karakteristik optimizer yang digunakan. Hasil ini menunjukkan bahwa optimizer adaptif seperti RMSprop, Adam, AdamW, dan Nadam mampu memberikan konvergensi yang lebih stabil dibandingkan SGD Momentum pada proses *transfer learning*.

4.2. Optimizer SGD (Momentum)

Visualisasi *accuracy curve* dan *loss curve* pada optimizer SGD Momentum digunakan untuk menganalisis stabilitas pelatihan dan proses konvergensi model *InceptionV3* pada gambar 4. dan 5.



Gambar 4. Accuracy Curve SGD - InceptionV3

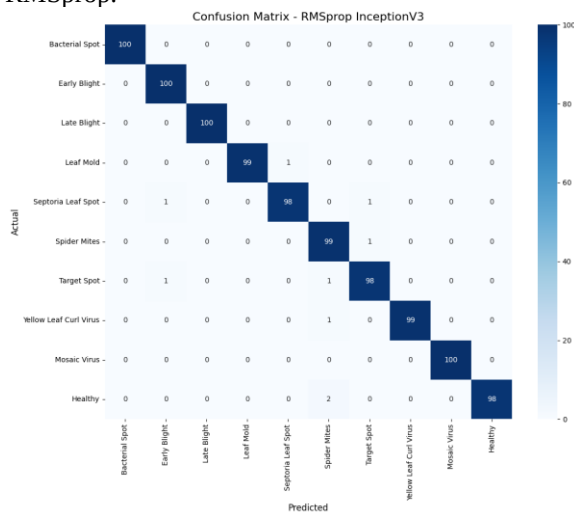


Gambar 5. Loss Curve SGD - InceptionV3

Optimizer SGD Momentum menunjukkan proses pelatihan yang stabil, namun peningkatan *accuracy* berlangsung lebih lambat dibandingkan optimizer adaptif. Kurva *loss* juga menunjukkan proses konvergensi yang membutuhkan lebih banyak *epoch* untuk mencapai performa optimal.

4.3. Optimizer RMSprop

Gambar 6 memperlihatkan *confusion matrix* dari model *InceptionV3* menggunakan optimizer RMSprop.

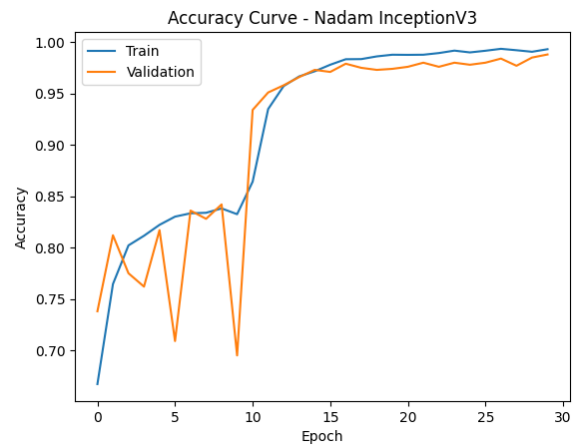


Gambar 6. Confusion Matrix RMSprop - InceptionV3

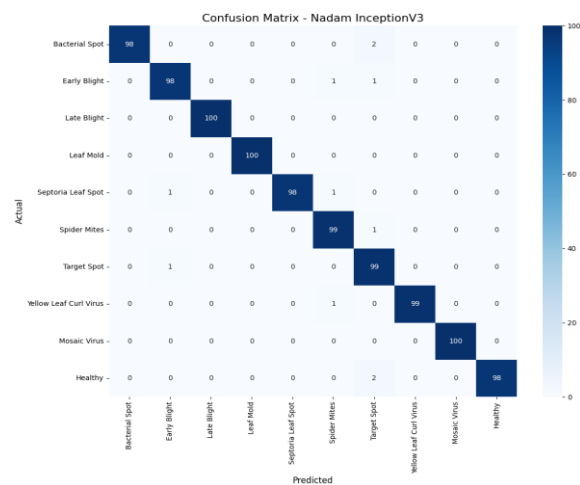
RMSprop menghasilkan performa terbaik pada model *InceptionV3* dengan tingkat kesalahan klasifikasi yang sangat rendah. Hal ini menunjukkan bahwa RMSprop mampu menyesuaikan *learning rate* secara adaptif sehingga proses pelatihan menjadi lebih stabil.

4.4. Optimizer Nadam

Performa model menggunakan optimizer Nadam dapat diamati melalui *accuracy curve* dan *confusion matrix* pada Gambar 7 dan Gambar 8.



Gambar 7. Accuracy Curve Nadam - InceptionV3

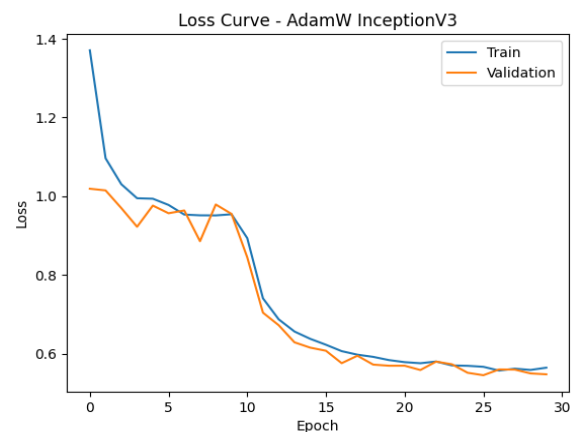


Gambar 8. Confusion Matrix Nadam - InceptionV3

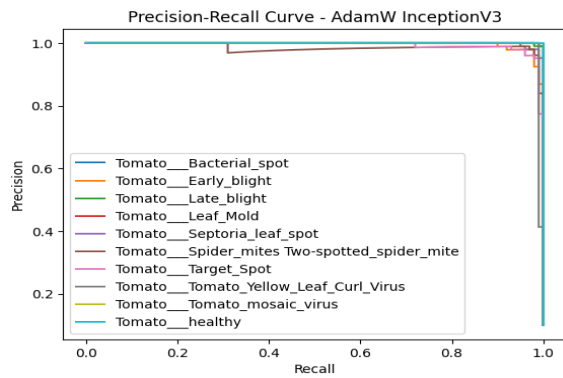
Nadam menunjukkan peningkatan *accuracy* yang cepat pada awal pelatihan dan stabil setelah beberapa *epoch*. Hasil *confusion matrix* juga menunjukkan distribusi klasifikasi yang cukup merata pada setiap kelas.

4.5. Optimizer AdamW

Visualisasi *loss curve* dan *precision-recall curve* pada optimizer AdamW ditampilkan pada Gambar 9 dan Gambar 10.



Gambar 9. Loss Curve AdamW - InceptionV3

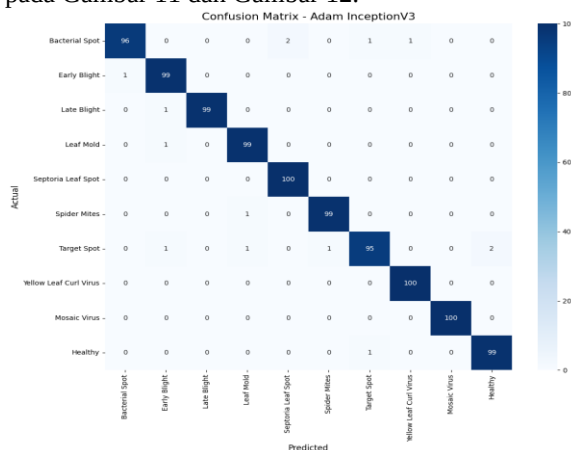


Gambar 10. Precision-Recall Curve AdamW - InceptionV3

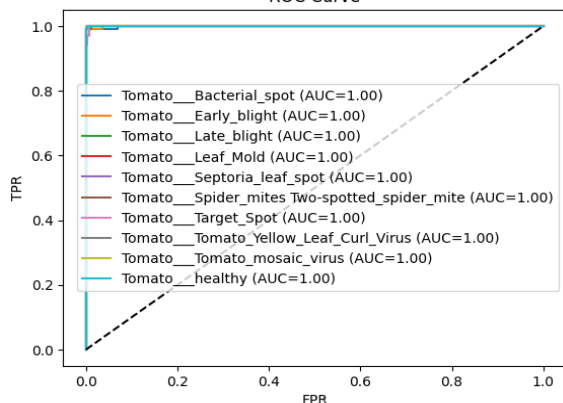
AdamW menunjukkan penurunan loss yang stabil dengan *precision* dan *recall* yang seimbang. Penggunaan *weight decay* pada AdamW membantu mengurangi *overfitting* selama proses pelatihan.

4.6. Optimizer Adam

Evaluasi model menggunakan optimizer Adam ditunjukkan melalui *confusion matrix* dan *ROC curve* pada Gambar 11 dan Gambar 12.



Gambar 11. Confusion Matrix Adam - InceptionV3

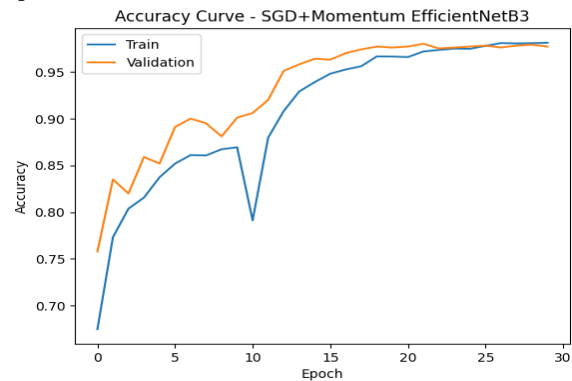


Gambar 12. ROC Curve Adam - InceptionV3

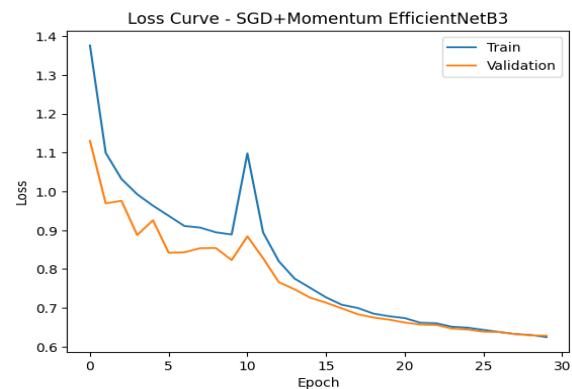
Adam menghasilkan performa klasifikasi yang baik dengan nilai AUC mendekati 1 pada *ROC curve*. Hal ini menunjukkan kemampuan model dalam membedakan setiap kelas secara optimal

4.7. Optimizer SGD (Momentum)

Gambar 13 dan Gambar 14 menunjukkan proses pelatihan model *EfficientNetB3* menggunakan optimizer SGD Momentum.



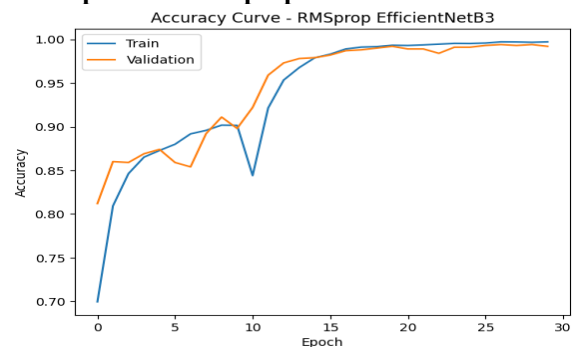
Gambar 13. Accuracy Curve SGD - EfficientNetB3



Gambar 14. Loss Curve SGD - EfficientNetB3

Pada *EfficientNetB3*, SGD Momentum menunjukkan fluktuasi pada kurva *loss* sehingga proses konvergensi menjadi kurang stabil. Hal ini menyebabkan performa model lebih rendah dibandingkan optimizer lainnya.

4.8. Optimizer RMSprop



Gambar 15. Accuracy Curve RMSprop - EfficientNetB3

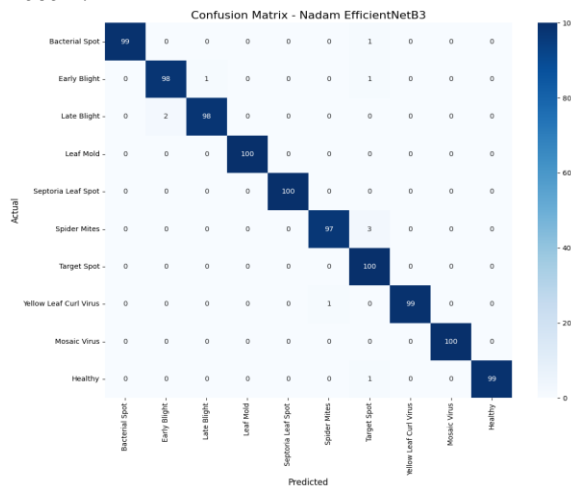
Stabilitas pelatihan model dengan optimizer RMSprop dapat dilihat pada *loss curve* dan *accuracy curve* di Gambar 14 dan Gambar 15.

RMSprop kembali menunjukkan performa terbaik pada *EfficientNetB3* dengan *accuracy* tertinggi

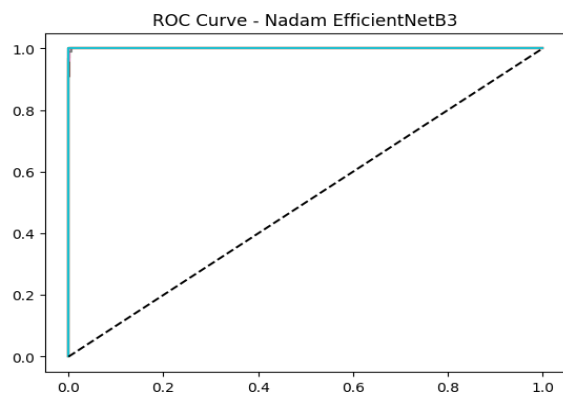
dan proses konvergensi yang stabil. Optimizer ini mampu menyesuaikan *learning rate* secara adaptif sehingga sesuai digunakan pada arsitektur yang lebih kompleks.

4.9. Optimizer Nadam

Gambar 16 dan Gambar 17 menampilkan hasil *confusion matrix* dan *ROC curve* pada optimizer Nadam.



Gambar 16. Confusion Matrix Nadam - EfficientNetB3



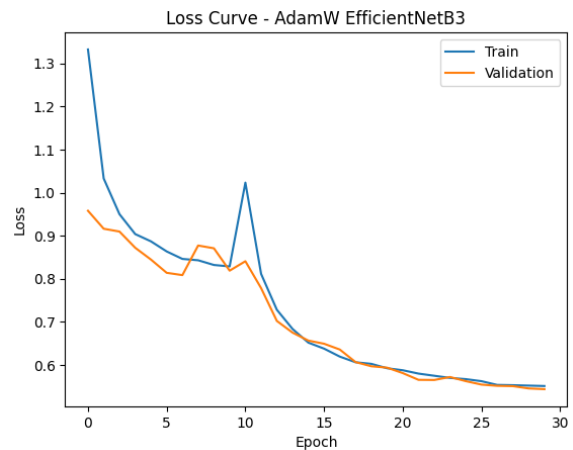
Gambar 17. ROC Curve Nadam - EfficientNetB3

Nadam menghasilkan performa klasifikasi yang stabil dengan tingkat kesalahan yang rendah pada hampir seluruh kelas. *ROC curve* juga menunjukkan nilai AUC yang tinggi pada setiap kelas.

4.10. Optimizer AdamW

Proses konvergensi model menggunakan optimizer AdamW ditunjukkan pada Gambar 18.

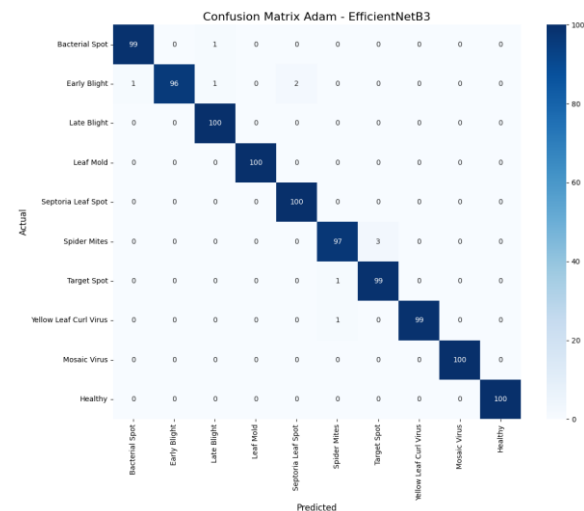
Kurva loss menunjukkan bahwa optimizer AdamW pada *EfficientNetB3* mampu menghasilkan proses konvergensi yang stabil dengan penurunan *loss train* dan *validation* yang konsisten serta gap yang kecil sehingga mengindikasikan kemampuan generalisasi model yang baik.



Gambar 18. Loss Curve Adamw - EfficientNetB3

4.11. Optimizer Adam

Tingkat ketepatan klasifikasi model menggunakan optimizer Adam dapat dilihat pada *confusion matrix* di Gambar 19.



Gambar 19. confusion matrix Adam - EfficientNetB3

Confusion matrix menunjukkan bahwa optimizer Adam pada *EfficientNetB3* mampu mengklasifikasikan hampir seluruh kelas dengan sangat baik dengan tingkat kesalahan yang rendah sehingga menghasilkan performa klasifikasi yang stabil dan akurat.

4.12. Analisis Komparatif

Berdasarkan seluruh hasil pengujian, optimizer memiliki pengaruh yang signifikan terhadap performa model klasifikasi penyakit daun tomat. Optimizer adaptif seperti RMSprop, Adam, AdamW, dan Nadam secara umum memberikan hasil yang lebih baik dibandingkan SGD Momentum. Hal ini disebabkan optimizer adaptif mampu menyesuaikan *learning rate* secara otomatis pada setiap parameter sehingga proses konvergensi menjadi lebih stabil.

InceptionV3 menunjukkan performa yang relatif konsisten pada seluruh optimizer. Sementara itu, *EfficientNetB3* menghasilkan *accuracy* yang lebih

tinggi, namun lebih sensitif terhadap pemilihan optimizer. Sensitivitas tersebut terlihat dari penurunan performa yang cukup signifikan pada penggunaan SGD Momentum.

RMSprop menjadi optimizer terbaik pada kedua model karena mampu menghasilkan *accuracy* tertinggi serta stabilitas pelatihan yang baik. Hasil ini menunjukkan bahwa optimizer adaptif lebih sesuai digunakan pada proses *transfer learning* dengan arsitektur CNN yang kompleks seperti *InceptionV3* dan *EfficientNetB3*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pemilihan optimizer memiliki pengaruh yang signifikan terhadap performa model *deep learning* dalam klasifikasi penyakit daun tomat. Seluruh eksperimen dilakukan menggunakan konfigurasi pelatihan yang sama sehingga perbedaan performa yang dihasilkan terutama dipengaruhi oleh karakteristik optimizer yang digunakan.

Optimizer adaptif seperti RMSprop, Adam, AdamW, dan Nadam secara umum memberikan performa yang lebih baik dibandingkan SGD Momentum, baik dari sisi *accuracy* maupun stabilitas proses pelatihan. Pada model *InceptionV3*, RMSprop menghasilkan performa terbaik dengan *accuracy* sebesar 99,1%, sedangkan pada *EfficientNetB3* RMSprop kembali menunjukkan hasil tertinggi dengan *accuracy* sebesar 99,2%. Selain itu, *EfficientNetB3* menunjukkan performa klasifikasi yang sedikit lebih baik dibandingkan *InceptionV3*, namun lebih sensitif terhadap pemilihan optimizer, terutama pada penggunaan SGD Momentum yang menghasilkan penurunan performa cukup signifikan.

Hasil penelitian juga menunjukkan bahwa optimizer adaptif mampu menghasilkan proses konvergensi yang lebih stabil pada arsitektur CNN yang kompleks menggunakan pendekatan *transfer learning*. Dengan demikian, RMSprop dapat direkomendasikan sebagai optimizer yang efektif untuk klasifikasi penyakit daun tomat menggunakan model *InceptionV3* dan *EfficientNetB3*.

Penelitian selanjutnya dapat dilakukan dengan mengembangkan optimasi *hyperparameter* seperti *learning rate*, *batch size*, dan jumlah *fine-tuning layer* untuk meningkatkan performa model. Selain itu, penggunaan dataset lapangan dengan kondisi pencahayaan dan latar belakang yang lebih bervariasi juga berpotensi meningkatkan kemampuan generalisasi model pada implementasi di dunia nyata.

DAFTAR PUSTAKA

- [1] M. Tan dan Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks BT - Proceedings of the International Conference on Machine Learning (ICML)," 2021.
- [2] I. Khan, S. S. Sohail, D. Ø. Madsen, dan B. K. Khare, "Deep transfer learning for fine-grained maize leaf disease classification," *J. Agric. Food Res.*, vol. 18, hal. 101148, 2024, doi: 10.1016/j.jafr.2024.101148.
- [3] F. O. Isinkaye, M. O. Olusanya, dan P. K. Singh, "Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations: A comprehensive review," *Heliyon*, vol. 10, no. 8, hal. e29583, 2024, doi: 10.1016/j.heliyon.2024.e29583.
- [4] T. D. Salka, "Plant leaf disease detection and classification using deep learning techniques: A review," *Artif. Intell. Rev.*, vol. 58, no. 4, hal. 1–35, 2025.
- [5] W. Shafik, "Deep learning technique for plant disease classification and detection," *BMC Plant Biol.*, vol. 25, no. 1, hal. 1–18, 2025.
- [6] N. Manogaran, "An automated plant leaf disease classification framework using deep learning," *Intell. Syst. Agric.*, vol. 4, no. 2, hal. 115–129, 2026.
- [7] M. Subhan, "Crop disease detection using EfficientNetB0 deep learning framework," *Inf. Syst. Front.*, vol. 28, no. 1, hal. 1–16, 2026.
- [8] D. K. Saha, M. R. Ahmed, T. D. Nath, dan M. F. Mridha, "Fusing explainable deep learning ensembles and LLM recommendations for real-time plant leaf disease diagnosis," *Intell. Syst. with Appl.*, vol. 25, hal. 200596, 2025, doi: 10.1016/j.iswa.2025.200596.
- [9] S. Srinivasan, "Sugarcane leaf disease classification using deep neural networks," *BMC Plant Biol.*, vol. 25, no. 1, hal. 1–20, 2025.
- [10] M. K. Elfouly, "A deep learning-based framework for large-scale plant disease detection," *J. Big Data*, vol. 12, no. 1, hal. 1–25, 2025.
- [11] M. D. T. Mallick, S. Banerjee, N. Thakur, dan A. Chakrabarti, "Evaluation of State-of-the-Art Deep Learning Techniques for Plant Disease and Pest Detection," *Comput. Mater. Contin.*, vol. 78, no. 3, hal. 1–25, 2025.
- [12] P. Kumar, "Zero shot plant disease classification with semantic attributes," *Artif. Intell. Rev.*, vol. 57, no. 6, hal. 1–28, 2024.
- [13] M. P. Aishwarya, "Ensemble of CNN models for classification of groundnut diseases using deep learning," *Smart Agric. Technol.*, vol. 5, hal. 100213, 2023, doi: 10.1016/j.atech.2023.100213.
- [14] M. Thanjaivadivel, "EnConv: Enhanced CNN for leaf disease classification," *Int. J. Inf. Technol.*, vol. 17, no. 2, hal. 1–15, 2025.
- [15] B. Sowmya, "Deep learning based plant health disease detection in tomato plants using Inception and YOLO," *Discov. Appl. Sci.*, vol. 7, no. 1, hal. 1–17, 2025.