

Klasifikasi Penyakit Pada Daun Cabai Menggunakan Arsitektur VGG16

Ahmad Sanusi Mashuri¹, Andi Sunyoto², Kusnawi³

^{1,2,3} Universitas Amikom Yogyakarta, Indonesia

Article Info

Article history:

Diterima 22 Juli 2024

Revisi 25 Juli 2024

Diterbitkan 3 Oktober 2024

Keywords:

Convolutional Neural Network
VGG16

Klasifikasi Penyakit pada Daun
Cabai

ABSTRAK

Penyakit pada tanaman cabai dapat mengancam produktivitas dan kualitas hasil panen jika tidak terdeteksi dan diatasi secara tepat waktu. Untuk meningkatkan deteksi dini penyakit pada tanaman cabai, kami mengembangkan sistem klasifikasi menggunakan Metode Convolutional Neural Network (CNN) dengan arsitektur VGG16, sebuah jaringan saraf konvolusional yang telah terbukti efektif dalam pengolahan gambar kompleks. Penelitian ini memanfaatkan dataset citra daun cabai yang terdiri dari beberapa kelas penyakit yang umum dijumpai, termasuk *Healthy*, *Yellowish*, *whitefly*, *leafcurl* dan *leafspot*. Citra-citra ini diolah dan dinormalisasi untuk pelatihan dan pengujian model. Arsitektur VGG16 digunakan sebagai model dasar, yang telah dipre-trained pada dataset ImageNet untuk meningkatkan kinerja klasifikasi. Proses pelatihan model dilakukan dengan memanfaatkan teknik transfer learning, di mana lapisan-lapisan akhir dari VGG16 disesuaikan dengan dataset penyakit daun cabai. Selama pengujian, sistem berhasil mengenali dan mengklasifikasikan penyakit pada daun cabai dengan tingkat akurasi yang tinggi. Hasil evaluasi menunjukkan bahwa arsitektur VGG16 menghasilkan tingkat akurasi terbaik 0.9962% dan validasi 0.9936% serta error dari masing masing proses Training dan validasi yaitu 0.0141 dan 0.0322 sedangkan waktu komputasi yang dibutuhkan untuk training dan validasi adalah 7 detik.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ahmad Sanusi Mashuri,

Universitas Amikom Yogyakarta

Jl. Ring Road Utara, Ngringin, Condongcatu, Daerah Istimewa Yogyakarta 55281, Indonesia

Email: sanusi@students.amikom.ac.id

1. PENDAHULUAN

Cabai (*Capsicum annum* L.) merupakan salah satu komoditas penting dalam pertanian, baik untuk konsumsi langsung maupun sebagai bahan baku industri makanan [1]. Namun, produktivitas tanaman cabai sering kali terancam oleh serangan penyakit pada daunnya. Penyakit-penyakit ini dapat menyebabkan penurunan produksi yang signifikan jika tidak ditangani dengan tepat dan cepat [2]

Pendeteksian penyakit pada tanaman secara manual oleh petani sering kali memerlukan waktu dan keahlian yang cukup, serta dapat memberikan hasil yang tidak konsisten. Oleh karena itu, penggunaan teknologi pengolahan citra dan kecerdasan buatan (artificial intelligence/AI) dapat menjadi solusi yang efektif dalam mendeteksi penyakit pada tanaman secara otomatis. Salah satu pendekatan yang telah dikembangkan adalah menggunakan arsitektur jaringan saraf konvolusional (*convolutional neural network/CNN*) yang terkenal,

seperti *VGG16*. Arsitektur *VGG16* telah terbukti efektif dalam berbagai tugas klasifikasi gambar, termasuk dalam konteks identifikasi dan klasifikasi penyakit pada tanaman [3]

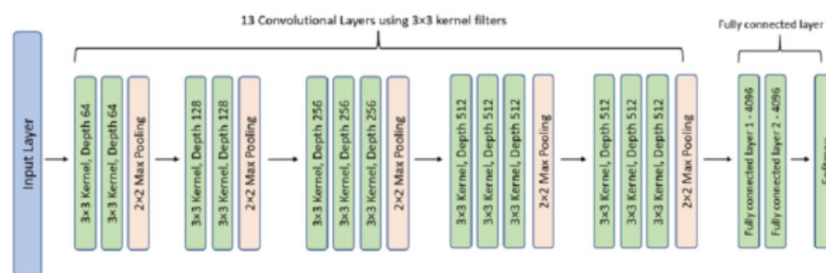
Dalam penelitian ini, kami memfokuskan pada pengembangan sistem untuk klasifikasi penyakit pada daun cabai menggunakan arsitektur *VGG16*. Metode ini memungkinkan kami untuk mengidentifikasi penyakit-penyakit yang umum terjadi pada daun cabai secara akurat dan efisien [4], berdasarkan citra digital yang diambil dari lapangan. Penggunaan teknologi ini diharapkan dapat membantu petani untuk mendeteksi dini penyakit pada tanaman cabai, sehingga tindakan kontrol dan pengendalian dapat dilakukan lebih cepat [5]. Dengan demikian, produktivitas dan kualitas hasil panen dapat dipertahankan atau ditingkatkan [6]

2. METODE

Pada tahapan ini, semua gambar yang digunakan untuk pelatihan dan pengujian akan diperkecil ukurannya ke dalam beberapa ukuran yaitu 128x128, 150x150, 224x224 dan 299x299 untuk melihat pengaruhnya terhadap akurasi dari hasil prediksi pada masing-masing model *CNN*. Ukuran gambar diperkecil dengan tujuan untuk mengurangi beban komputasi yang berat dan pemrosesan yang lebih cepat [7] Dataset kemudian dibagi menjadi data latih, data validasi dan data uji. Data latih selanjutnya akan diaugmentasi sedangkan data uji akan digunakan pada saat tahapan pengujian untuk mengetahui performa dari setiap model *CNN*.

Convolutional Neural Network adalah salah satu metode klasifikasi maupun prediksi yang bisa diimplementasikan pada data gambar, suara maupun teks. Pada umumnya ada tiga tipe layer Arsitektur *CNN* yaitu convolution layer, pooling layer, dan fully connected layer. Ekstraksi informasi pada gambar sering kali digunakan Convolution layer dan pooling layer [8][9]. Sedangkan fully connected layer digunakan sebagai prediksi serta memberi label pada data yang paling tepat sebagai hasil prediksi maupun klasifikasi. [10][11] Transfer Learning merupakan teknik yang memanfaatkan model yang sudah ditraining sebelumnya (pretrained model) untuk digunakan mengklasifikasikan dataset yang baru sehingga tidak perlu untuk melakukan training data dari awal. dan dilakukan penyesuaian pada bagian akhir dari model [12]. Secara umum arsitektur *CNN* terdiri dari beberapa layer, layer-layer ini yang melakukan ekstraksi fitur-fitur yang ada pada dataset.

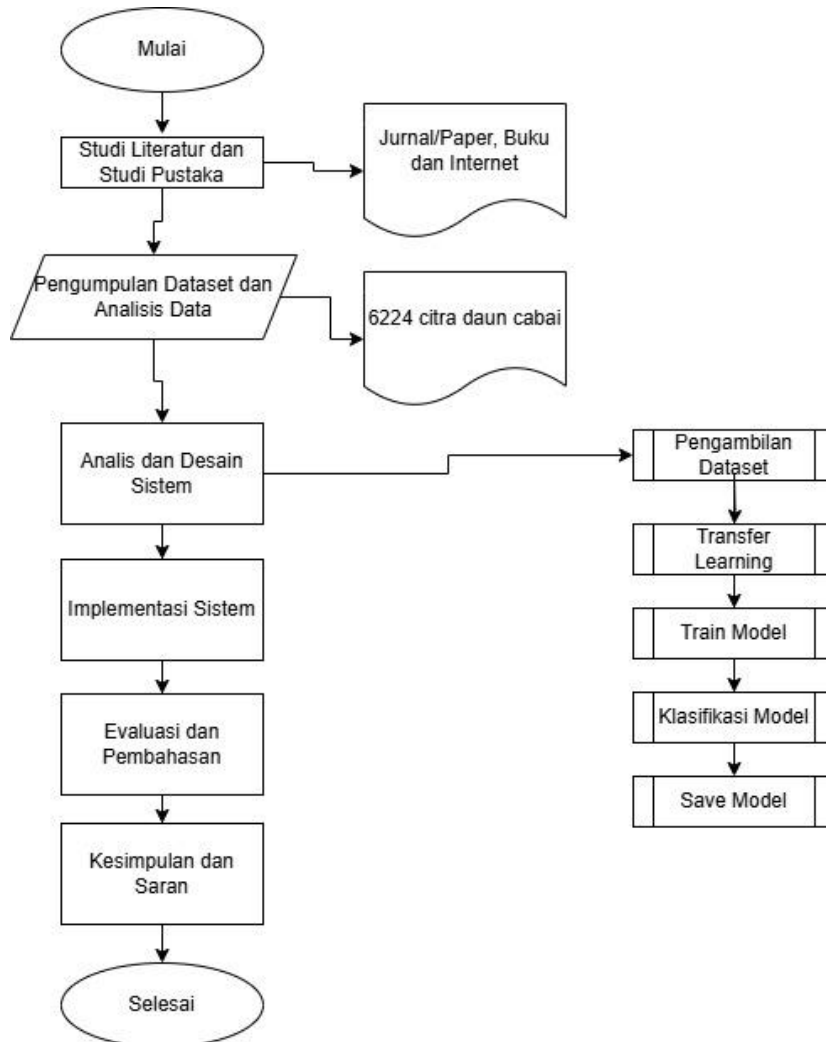
Arsitektur *VGGNet* [13] adalah arsitektur pengembangan dari alexnet yang mana arsitektur tersebut fokus pada proses ekstraksi fitur di layer konvolusi sehingga mampu mendapatkan representasi citra yang banyak untuk dapat di klasifikasi. Arsitektur *VGG16* merupakan bagian dari model deep learning meliputi dari 16 layer. Model arsitektur *VGG16* dapat dilihat gambar 1. Sedangkan arsitektur *VGG16* sendiri memiliki 13 layer lapisan konvolusi, dan 2 lapisan digunakan sebagai fully_connected, serta 1 lapisan klasifikasi. [14][15]



Gambar 1 Model Arsitektur *VGG16*

Gambar 1. tiap susunan konvolusi diatas memiliki dimensi kernel 3x3. Perbedaannya pada tiap susunan konvolusi dari masing- masing susunan jumlah filter. Jumlah filter 64 ada pada2 susunan konvolusi awal, sedangkan jumlah kernel 128 susunan 3 serta 4. Begitu pada susunan konvolusi lain yang jumlah filternya berbeda ialah 256 (susunan 4 5 6) serta 512 (susunan 7 8 9 10 11 12). 2x2 max pooling digunakan sesudah susunan konvolusi 2 4 7 10 serta 13. output pooling terakhir akan dihubungkan ke layer fully_connected, dan pada akhirnya terhubung ke klasifikasi guna untuk penentuan class ataupun label dataset.

Penelitian ini juga mencakup serangkaian langkah metodis sebagaimana digambarkan dalam model penelitian kami, dimulai dari pengumpulan data hingga penarikan kesimpulan, yang akan diuraikan lebih lanjut dan divisualisasikan dalam Gambar 2.



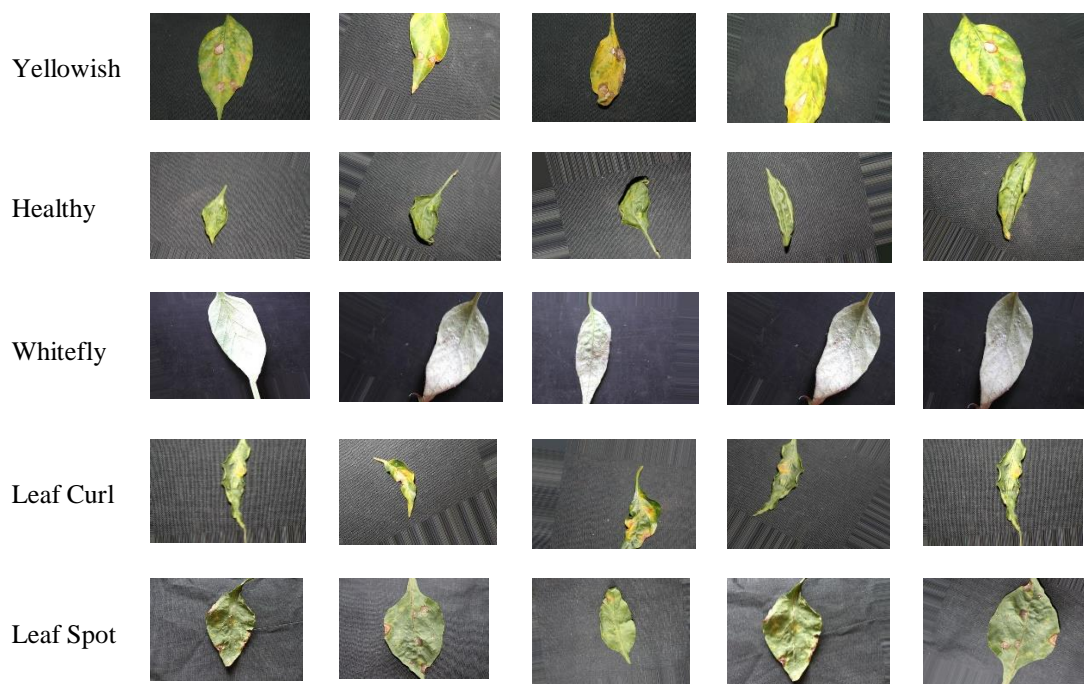
Gambar 2. Bagan Alur Penelitian

2.1 Studi Literatur atau Pustaka

Studi literatur penelitian dilakukan untuk mencari serta digunakan untuk mengumpulkan dataset/sumber informasi terkait dengan penelitian yang akan dilakukan seperti dasar metodologi, metodologi penelitian, teori dan rujukan peneliti. Metode pengumpulan data yang dilakukan pada penelitian ini adalah studi literatur dan studi pustaka.

2.2 Pengumpulan data

Dataset yang digunakan dalam penelitian ini yakni dataset Chilli_Leaf_Dataset (Mahaning Hubballi-2023). Jumlah dataset yang digunakan dalam penelitian ini menggunakan data sebanyak 6224 Daun cabai, untuk masing-masing kelas (*Healthy*, *Yellowish*, *Leaf Curl*, *Leaf Spot*, dan *Whitefly*). Gambar 3 merupakan contoh dari dataset yang kami gunakan.



Gambar 3. Sample Dataset Citra Daun Cabai

2.3 Pengolahan (Analisis Data)

Analisis data dilakukan untuk proses memisahkan citra daun cabai dengan 5 kategori penyakit yaitu *Healthy*, *Yellowish*, *Leaf Curl*, *Leaf Spot*, dan *Whitefly*. [15]

<i>Healthy</i>	Merupakan daun cabai sehat
<i>Yellowish</i>	Daun cabai terkena penyakit kekuningan
<i>Leaf Curl</i>	Daun cabai terkena penyakit keriting daun
<i>Leaf Spot</i>	Daun cabai terkena penyakit bercak daun
<i>Whitefly</i>	Daun cabai terkena penyakit kutu kebul

2.4 Analisis dan Desain Sistem

Tahapan berupa perencanaan, gambaran dan proses pembuatan sistem yang akan di bangun metode klasifikasi transfer learning pre-trained model.

2.4.1 Transfer Learning

Pada penelitian ini, kami menggunakan arsitektur *VGG16* yang telah dilatih sebelumnya dengan bobot dari dataset *ImageNet*. Model *VGG16* digunakan sebagai dasar (base model) dengan mengabaikan lapisan teratas (`include_top=False`) dan mengatur ukuran input sebesar 224x224 piksel dengan tiga saluran warna (RGB). Untuk menjaga bobot model *VGG16* agar tidak berubah selama pelatihan, kami mengatur parameter `trainable` menjadi `False`. Selanjutnya, kami membangun model sekuensial dengan menambahkan beberapa lapisan tambahan. Lapisan-lapisan ini meliputi lapisan konvolusi dengan 32 filter, kernel berukuran 3x3, aktivasi ReLU, dan padding 'same', diikuti oleh lapisan max pooling dengan ukuran pool 2x2 dan stride 2. Kami juga menambahkan lapisan dropout dengan tingkat drop 50% untuk mencegah overfitting. Kemudian, kami meratakan (flatten) hasil dari lapisan sebelumnya dan menambah lapisan dense dengan 5 neuron keluaran dan aktivasi softmax untuk klasifikasi. Model ini diberi nama *VGG16_model_05*.

2.4.2 Train Model

Pada tahap pelatihan, model *VGG16_model_05* dilatih menggunakan generator data untuk pelatihan dan validasi. Proses pelatihan dilakukan selama 100 epoch. Generator data pelatihan (`train_gen`) digunakan untuk memasukkan data gambar ke dalam model selama pelatihan, sedangkan generator data validasi (`valid_gen`) digunakan untuk mengevaluasi

kinerja model pada setiap epoch. Model dilatih untuk meminimalkan loss function `categorical_crossentropy` menggunakan optimizer Adam dengan learning rate 0.0001. Selama proses pelatihan, metrik akurasi digunakan untuk memantau kinerja model. History dari proses pelatihan ini disimpan dalam variabel history, yang mencakup informasi tentang loss dan akurasi pada set pelatihan dan validasi untuk setiap epoch.

2.4.3 Klasifikasi Model

selanjutnya, kami membangun model menggunakan arsitektur *VGG16* yang telah dilatih sebelumnya dengan bobot dari dataset *ImageNet*. Model *VGG16* digunakan sebagai dasar (base model) dengan mengabaikan lapisan teratas (`include_top=False`) dan mengatur ukuran input sebesar 224x224 piksel dengan tiga saluran warna (RGB). Untuk menjaga bobot model *VGG16* agar tidak berubah selama pelatihan, kami mengatur parameter trainable menjadi False. Kami kemudian membangun model sekuensial dengan menambahkan beberapa lapisan tambahan: lapisan konvolusi dengan 32 filter, kernel berukuran 3x3, aktivasi ReLU, dan padding 'same', diikuti oleh lapisan max pooling dengan ukuran pool 2x2 dan stride 2. Kami juga menambahkan lapisan dropout dengan tingkat drop 50% untuk mencegah overfitting, kemudian melanjutkan dengan lapisan flatten, dan akhirnya menambah lapisan dense dengan 5 neuron keluaran dan aktivasi softmax untuk klasifikasi. Model ini diberi nama *VGG16_model_05*. Kami mengkompilasi model menggunakan optimizer Adam dengan learning rate 0.001, loss function `categorical_crossentropy`, dan metrik akurasi.

2.4.4 Save Model

Selama proses penyimpanan model menggunakan metode `model.save()`, kami menerima peringatan dari Keras yang menyatakan bahwa format file HDF5 yang digunakan dianggap sebagai format lama (legacy). Keras merekomendasikan untuk menggunakan format asli Keras, seperti `model.save('my_model.keras')`, untuk penyimpanan model. Peringatan ini menunjukkan bahwa meskipun HDF5 masih didukung, penggunaan format asli Keras lebih disarankan karena menawarkan dukungan yang lebih baik dan fitur-fitur terbaru. Oleh karena itu, untuk memastikan kompatibilitas dan pemanfaatan fitur terbaru dari Keras, kami menyarankan untuk menyimpan model menggunakan format `.keras`.

2.5 Evaluasi dan Hasil Pembahasan

Secara keseluruhan, pendekatan ini menunjukkan bagaimana penggunaan transfer learning dengan model pra-terlatih seperti *VGG16* dapat mempercepat pengembangan model yang efektif dan akurat. Penambahan lapisan tambahan dan penggunaan metode pelatihan yang tepat memungkinkan penyesuaian model dengan data spesifik. Selain itu, mengikuti rekomendasi terbaru dalam penyimpanan model memastikan efisiensi dan kompatibilitas di masa mendatang.

3. HASIL DAN PEMBAHASAN

Pada pengujian penelitian ini digunakan laptop dell e7270 dengan spesifikasi core i7 gen 6, ram 16gb, ssd 512gb, dan sistem operasi windows 10 pro serta menggunakan tools bantu Visual Code untuk implementasi dari program. Berikut table implementasi dan hasil penelitian setelah dilakukan training dan validasi dengan menggunakan deep pretrained model *VGG16*.

Table I		TABLE I. TABLE TRANSFER LEARNING
	Nama	Parameter
	Optimization	Adam
	Batch size	79
	Shape input	7, 7, 512
	Learning Rate	0.001
	Activation	Softmax

Model yang dikembangkan, yang diberi nama "sequential_3", terdiri dari beberapa lapisan yang diatur secara berurutan untuk mengklasifikasikan penyakit pada daun cabai. Arsitektur model dimulai dengan penggunaan arsitektur *VGG16* sebagai base model, yang menghasilkan output dengan dimensi (7, 7, 512) dan jumlah

parameter non-trainable sebesar 14,714,688. Model ini tidak menyertakan lapisan teratas *VGG16*, sehingga memungkinkan penambahan lapisan khusus untuk tugas klasifikasi ini.

Setelah lapisan *VGG16*, model dilanjutkan dengan lapisan konvolusi (Conv2D) yang memiliki 32 filter dengan kernel 3x3, yang mengubah dimensi output menjadi (7, 7, 32) dengan total parameter sebanyak 147,488. Lapisan konvolusi ini diikuti oleh lapisan pooling (MaxPooling2D) dengan ukuran pool 2x2, yang mengurangi dimensi fitur menjadi (3, 3, 32) tanpa menambah parameter baru. Untuk mencegah overfitting, lapisan dropout (Dropout) diterapkan dengan proporsi 50%, yang mempertahankan dimensi output yang sama.

Selanjutnya, hasil dari lapisan sebelumnya diratakan (Flatten) menjadi vektor dengan panjang 288, dan diteruskan ke lapisan dense (Dense) dengan 5 neuron keluaran dan fungsi aktivasi softmax, yang digunakan untuk klasifikasi akhir. Lapisan dense ini menambahkan 1,445 parameter trainable. Secara keseluruhan, model ini memiliki total 14,863,621 parameter, di mana 148,933 parameter dapat dilatih dan 14,714,688 parameter adalah non-trainable.

Table II

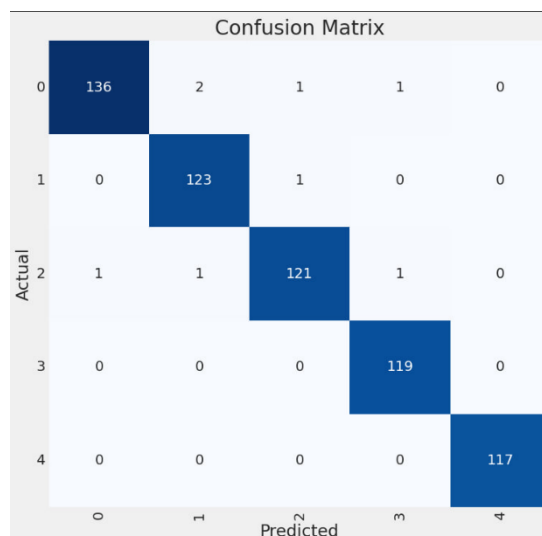
TABLE II. TABLE HASIL AKURASI DAN EROR

Data	Rate	0.0001	
	Training	Val	Waktu
Akurasi	0.9962	0.9936	7 Detik
Eror	0.0141	0.0322	7 Detik

Table III

TABLE III. TABLE CONFUSION MATRIX

	Precision	Recall	F1-Score	Support
0	0.99	0.97	0.98	140
1	0.98	0.99	0.98	124
2	0.98	0.98	0.98	124
3	0.98	1.00	0.99	119
4	1.00	1.00	1.00	117

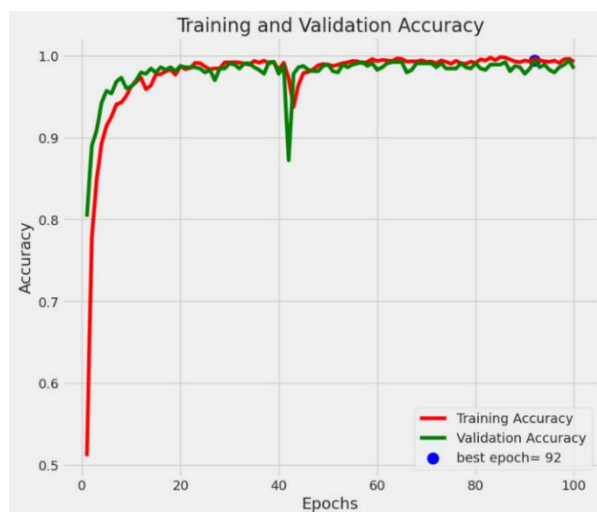


Gambar 4 *Confusion Matrix*

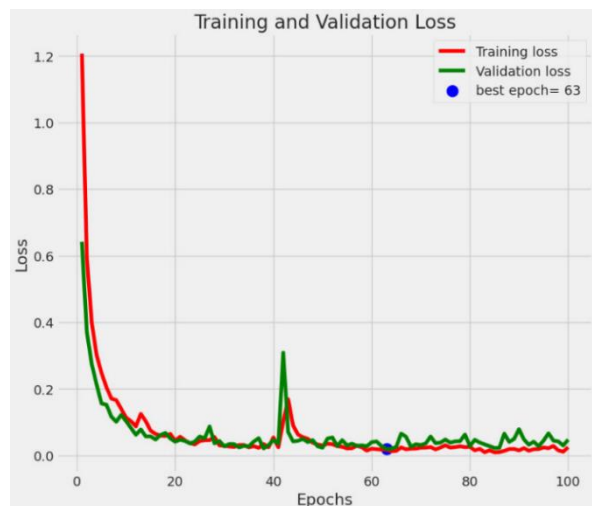
Pada table 3 dan gambar 4, Hasil evaluasi model klasifikasi pada dataset uji ditunjukkan dalam laporan klasifikasi berikut. Model menunjukkan performa yang sangat baik dengan tingkat akurasi keseluruhan mencapai 99%. Berdasarkan metrik precision, recall, dan f1-score, model mencapai nilai tinggi untuk setiap kelas yang diuji. Secara spesifik:

Class	Penjelasan
0	kelas 0, model mencapai precision sebesar 0.99, recall 0.97, dan f1-score 0.98, dengan dukungan sebanyak 140 sampel.
1	kelas 1, nilai precision adalah 0.98, recall 0.99, dan f1-score 0.98, dengan 124 sampel.
2	Kelas 2 menunjukkan precision 0.98, recall 0.98, dan f1-score 0.98, juga dengan 124 sampel.
3	Kelas 3 memiliki precision 0.98, recall 1.00, dan f1-score 0.99, dengan 119 sampel.
4	Kelas 4 memperoleh nilai sempurna dengan precision, recall, dan f1-score masing-masing 1.00, serta dukungan 117 sampel.

Akurasi makro dan rata-rata tertimbang juga menunjukkan performa model yang konsisten dengan nilai 0.99. Metrik ini menunjukkan bahwa model tidak hanya bekerja dengan baik pada setiap kelas tetapi juga secara keseluruhan pada dataset uji, menjadikannya alat yang sangat efektif untuk klasifikasi penyakit pada daun cabai.



Gambar 5 Grafik Akurasi



Gambar 6 Grafik Los Validasi

Berdasarkan hasil gambar 5 dan 6 dapat disimpulkan bahwa menggunakan deep pretrained model *VGG16* dengan epoch 100 batch size 79, learning rate 0.0001 sedangkan untuk parameter yang digunakan optimasi adalah adam serta aktivasi parameter adalah softmax. Dataset training berjumlah 6224 data citra daun cabai

dengan kategori *Healthy*, *Yellowish*, *whitefly*, *leafcurl* dan *leafspot*, target size yang digunakan adalah 80x80x30 menghasilkan tingkat akurasi training 0.9962% sedangkan validasi menggunakan data dan akurasi validasi terbaik adalah 0.9936% dan best epoch pada validasi akurasi diperoleh nilai sebesar 92 sedangkan best epoch pada los validasi sebesar 63 dengan waktu komputasi yang di butuhkan adalah 7 detik. Dari perbandingan tersebut dapat disimpulkan bahwa jumlah epoch berpengaruh terhadap tingkat akurasi, selain itu juga semakin banyak epoch yang digunakan maka semakin tinggi tingkat akurasi yang didapatkan.

4. KESIMPULAN

Hasil penelitian ini mengusulkan metode deep pre-trained model *VGG16* dengan menggunakan tools bantu visual studio code menghasilkan tingkat akurasi terbaik 0.9962% dan validasi 0.9936% serta error dari masing masing proses Training dan validasi yaitu 0.0141 dan 0.0322 sedangkan waktu komputasi yang dibutuhkan untuk training dan validasi adalah 7 detik.

UCAPAN TERIMA KASIH

Kami, sebagai penulis, ingin mengucapkan rasa terima kasih kepada pembimbing kami, yaitu Dr. Andi Sunyoto, M. Kom., dan Kusnawi, S.Kom., M.Eng., atas bimbingan, dukungan, dan motivasi yang telah diberikan selama proses penelitian ini.

REFERENSI

- [1] K. Sebastian, M. S. Arya, U. R. Reshma, S. J. Anaswara, and S. S. Thampi, "Impact of Plant Growth Regulators on Fruit Production," *Int. J. Curr. Microbiol. Appl. Sci.*, vol. 8, no. 02, pp. 800–814, 2019, doi: 10.20546/ijcmas.2019.802.092.
- [2] V. P. Semenov, L. Y. Baranova, and T. S. Yagya, "Application of Artificial Intelligence in Medicine," *Proc. 2022 25th Int. Conf. Soft Comput. Meas. SCM 2022*, vol. 41, no. 6, pp. 262–265, 2022, doi: 10.1109/SCM55405.2022.9794891.
- [3] H. Salman, J. Grover, and T. Shankar, "Hierarchical Reinforcement Learning for Sequencing Behaviors," vol. 2733, no. March, pp. 2709–2733, 2018, doi: 10.1162/NECO.
- [4] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cogn. Syst. Res.*, vol. 53, pp. 31–41, 2019, doi: 10.1016/j.cogsys.2018.04.006.
- [5] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning - A Review," *IEEE Access*, vol. 9, no. Ccv, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [6] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. July 2017, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.
- [7] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, "Pneumonia Detection Using CNN based Feature Extraction," *Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019*, pp. 1–7, 2019, doi: 10.1109/ICECCT.2019.8869364.
- [8] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 9992–10002, 2021, doi: 10.1109/ICCV48922.2021.00986.
- [9] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.
- [10] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [11] T. Zhou, S. Ruan, and S. Canu, "A review: Deep learning for medical image segmentation using multi-modality fusion," *Array*, vol. 3–4, 2019, doi: 10.1016/j.array.2019.100004.
- [12] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *Proc. Mach. Learn. Res.*, vol. 139, pp. 10347–10357, 2021.
- [13] V. Atliha and D. Sesok, "Comparison of VGG and ResNet used as Encoders for Image Captioning," *2020 IEEE Open Conf. Electr. Electron. Inf. Sci. eStream 2020 - Proc.*, pp. 1–4, 2020, doi: 10.1109/eStream50540.2020.9108880.
- [14] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, 2017, doi:

- 10.3390/s17092022.
- [15] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, no. October 2017, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032.