



Klasifikasi Citra Penyakit Daun Tomat Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur VGG-19

Fitri Handayani^{1*}, Dr. Baidarus², Sunanto³, Reny Medikawati Taufiq⁴, Bayu Anugerah Putra⁵, Chelina Anggraini Putri⁶

Email: ¹fitrihandayani@umri.ac.id, ²bayu@umri.ac.id, ³sunanto@umri.ac.id, ⁴renymedikati@umro.ac.id,

⁵bayuanugerahputra@umri.ac.id, ⁶210401243@student.umri.ac.id

^{1,2,3,4,5,6}Teknik Informatika, Ilmu Komputer, Universitas Muhammadiyah Riau

Diterima: 09 Desember 2025 | Direvisi: - | Disetujui: 11 Desember 2025

©2025 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Tomat dalam bahasa latin yaitu *Solanum lycopersicum* merupakan komoditas jenis hortikultura bernilai ekonomi tinggi di Indonesia. Produksi tomat dapat menurun akibat serangan penyakit daun yang sulit diidentifikasi secara manual karena kemiripan gejala antar penyakit. Adapun tujuan penelitian yaitu untuk menerapkan sistem klasifikasi penyakit daun tomat berbasis deep learning menggunakan arsitektur Convolutional Neural Network (CNN) VGG-19. Dataset diperoleh dari Kaggle, berisi 6.600 citra daun tomat terbagi dalam enam kelas penyakit dan satu kelas daun sehat. Tahapan penelitian meliputi preprocessing (resize, normalisasi), augmentasi data, pembagian dataset (80% latih, 20% uji), pelatihan model dengan transfer learning serta fine-tuning untuk optimasi. Evaluasi menggunakan confusion matrix dan classification report meliputi akurasi, presisi, recall, dan F1-score. Hasil pengujian menunjukkan model VGG-19 mencapai akurasi 97% pada data uji, dengan rata-rata presisi, recall, dan F1-score berjumlah 0,97. Hasil penelitian didapat VGG-19 efektif dalam mengidentifikasi penyakit daun tomat dan berpotensi diimplementasikan sebagai sistem deteksi berbasis web atau mobile untuk membantu petani melakukan diagnosis dini dan penanganan tepat.

Kata kunci: Convolutional Neural Network (CNN), Deep Learning, Klasifikasi, Pemrosesan Citra, Penyakit Daun, Tomat, VGG-19

Tomato Leaf Disease Image Classification Using Convolutional Neural Network (CNN) Method With VGG-19 Architecture

Abstract

Tomatoes, known as *Solanum lycopersicum* in Latin, are a type of horticultural commodity with high economic value in Indonesia. Tomato production can decrease due to leaf diseases that are hard to identify manually because the symptoms of different diseases often appear similar. The purpose of this study is to apply a deep learning-based tomato leaf disease classification system using the Convolutional Neural Network (CNN) VGG-19 architecture. The dataset was obtained from Kaggle and contains 6,600 images of tomato leaves divided into six disease classes and one healthy leaf class. The research stages include preprocessing (resizing, normalization), data augmentation, dataset division (80% training, 20% testing), model training with transfer learning, and fine-tuning for optimization. The evaluation using the confusion matrix and classification report includes accuracy, precision, recall, and F1-score. Test results show that the VGG-19 model achieved 97% accuracy on the test data, with an average precision, recall, and F1-score of 0.97. These findings show that VGG-19 effectively identifies tomato leaf diseases and could be applied in web- or mobile-based detection systems to help farmers with early diagnosis and proper treatment.

Keywords: Image processing, Convolutional Neural Networks (CNN), Deep learning, Classification, Leaf disease, Tomatoes, VGG-19

1. PENDAHULUAN

Tomat (*Solanum lycopersicum*) adalah salah satu komoditas hortikultura penting dengan nilai ekonomi tinggi yang banyak dibudidayakan di Indonesia. Berdasarkan data dari Badan Pusat Statistik, produksi tomat pada 2022 mencapai 1,16 juta ton, namun menurun pada 2023. Salah satu penyebab penurunan tersebut adalah serangan hama dan penyakit daun yang dapat menurunkan kualitas, serta kandungan gizi tomat [1]. Selain itu, produktivitas tomat juga menjadi terhambat dikarenakan penyakit tanaman. Oleh karena itu, diagnosa penyakit menjadi krusial dalam meningkatkan stabilitas produksi [9]. Pada masyarakat, Penyakit pada daun tanaman tomat sulit dikenali hanya dengan pengamatan visual, karena gejalanya sering tampak mirip satu sama lain. Kondisi ini membuat identifikasi manual memiliki banyak kelemahan. Penurunan produksi tomat dapat dicegah apabila upaya pengendalian penyakit dilakukan secara tepat dan sesuai. [10].

Metode deep learning, Terkhususnya Convolutional Neural Network (CNN), sudah banyak digunakan dalam klasifikasi citra karena kemampuannya mengekstraksi fitur visual secara otomatis. Salah satu arsitektur yang terbukti efektif adalah VGG-19, yang terdiri dari 19 lapisan dengan struktur sederhana namun mendalam. Arsitektur ini telah dilatih pada dataset ImageNet sehingga mampu mengenali pola visual umum dan dapat diadaptasikan untuk klasifikasi penyakit daun tomat.

Penelitian terdahulu menunjukkan berbagai pendekatan dalam klasifikasi penyakit daun. Misalnya, KNN dengan GLCM mencapai akurasi 92,89% [2], Decision Tree dan Random Forest masing-masing 90% dan 94% [3], serta CNN dengan arsitektur MobileNet mencapai 88% [1]. Selain itu, penerapan metode ResNet-50 juga dapat menjadi metode yang efektif dalam melakukan klasifikasi penyakit tomat dengan akurasi 93% [8]. Penerapan VGG-19 sendiri pada citra penyakit tanaman telah menunjukkan akurasi tinggi, seperti pada penelitian Shinta [4] yang mengklasifikasikan penyakit daun padi dengan akurasi 93%.

Selain itu, model VGG-19 juga telah diaplikasikan dalam klasifikasi tingkat kematangan buah. Wicaksono & Amrulloh [5] menggunakannya untuk buah pisang Cavendish dan memperoleh akurasi hingga 98%, sementara Gerhana [6] menerapkannya pada buah kopi dengan akurasi 86%. Hasil-hasil tersebut menunjukkan potensi VGG-19 yang konsisten dalam mendeteksi pola visual pada berbagai objek pertanian.

Berdasarkan penjelasan diatas, penelitian ini dilakukan untuk mengimplementasikan CNN berbasis VGG-19 dalam klasifikasi penyakit daun tomat. Model diharapkan mampu mengenali enam kelas, yakni Tomato Target Spot, Tomato Late Blight, Tomato Leaf Mold, Tomato Spider Mites (Two-spotted), Tomato Septoria Leaf Spot, serta Tomato Healthy. Penelitian ini juga diarahkan untuk memberikan kontribusi dalam pengembangan sistem pendeteksian penyakit tanaman yang lebih efisien, tepat, dan mudah diterapkan oleh para petani di lapangan.

2. METODE PENELITIAN

Dalam penelitian ini, proses klasifikasi penyakit daun tomat terdiri dari beberapa tahapan seperti yang ditunjukkan pada Gambar 1.



Gambar 1 Kerangka penelitian

Pada Gambar 1 ditunjukkan kerangka kerja Klasifikasi penyakit pada daun tomat dilakukan menggunakan model CNN dengan memanfaatkan arsitektur VGG-19. Proses dimulai dari pengumpulan dataset citra daun tomat yang diperoleh dari Kaggle. Data kemudian melalui tahap preprocessing berupa resizing ke ukuran 224×224 piksel sesuai kebutuhan VGG-19, serta augmentasi untuk menambah variasi citra dan mengurangi risiko overfitting. Selanjutnya dataset dibagi menjadi tiga bagian, yaitu data latih untuk proses pembelajaran, data validasi untuk memantau kinerja model, serta data uji untuk mengevaluasi performa akhir.

Model CNN dibangun dengan arsitektur VGG-19 yang dikenal unggul dalam ekstraksi fitur citra. Proses evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, dan f1-score untuk menilai kemampuan model dalam melakukan klasifikasi. jenis penyakit daun tomat. Dengan alur ini, seluruh tahapan mulai dari pengumpulan data hingga evaluasi dapat menghasilkan model yang optimal dan menjadi dasar dalam penarikan kesimpulan penelitian.

2.1. Pengumpulan Dataset

Dataset citra daun tanaman tomat ini diperoleh dari situs Kaggle yang Bernama “Tomato leaf disease detection”. Dataset yang digunakan 6.600 dimana terdapat 6 kelas dengan jumlah per kelas 1.100 citra daun. Berikut merupakan sampel citra daun tomat pada setiap kelas. Sampel dataset penyakit daun tomat dapat dilihat pada Gambar 2.



Gambar 2 sampel citra penyakit daun tomat

Gambar 2 merupakan sampel citra penyakit daun tomat yang terdiri dari kelas *Healthy*, kelas *Late Blight*, kelas *Leaf Mold*, kelas *Septoria Leaf Spot*, kelas *Spider Mites Two-Spotted*, dan kelas *Target Spot*.

2.2. Preprocessing data

Sebelum data dibagi kedalam data training, validasi, dan testing, citra daun tomat terlebih dahulu melewati tahap preprocessing dataset. Tahapan awal ini sangat penting untuk memastikan kualitas citra yang lebih optimal. Berikut Langkah-langkah preprocessing dataset diuraikan dalam penelitian ini:

1. **Resize** : dilakukan untuk menyamakan ukuran citra agar proses perhitungan dan pengenalan citra dapat berjalan lebih mudah dan efisien. Langkah ini sangat diperlukan agar dalam pra-pemrosesan data gambar bisa memastikan bahwa semua gambar yang dimasukkan ke dalam model memiliki ukuran yang konsisten. Ukuran citra disesuaikan menjadi 224x224 piksel mengikuti standar input yang dibutuhkan oleh arsitektur VGG-19.
2. **Normalisasi Data**: Normalisasi dilakukan dengan mengubah nilai piksel citra dari rentang 0–255 menjadi 0–1. Proses ini bertujuan untuk mempercepat pelatihan model, menjaga kestabilan pembelajaran, dan menyesuaikan data dengan kebutuhan arsitektur VGG-19.
3. **Augmentasi data** : Tahap augmentasi tidak digunakan pada proses klasifikasi yang memanfaatkan citra asli. Augmentasi sendiri merupakan teknik yang berfungsi untuk menambah jumlah citra ketika dataset yang tersedia terbatas. Tujuan dari penerapan augmentasi adalah untuk meningkatkan akurasi model dalam penelitian. Proses ini dilakukan dengan memanfaatkan fungsi Image Data Generator yang tersedia dalam library TensorFlow.

2.3. Pembagian data

Dataset pada penelitian ini dibagi menjadi tiga bagian, yaitu data training yang digunakan untuk membangun model, data testing yang dipakai untuk mengevaluasi kinerja model selama proses pelatihan, serta data validasi yang berfungsi menilai performa model secara objektif di luar data pelatihan. Dataset yang digunakan terdiri dari 6.600 citra daun tomat yang terbagi dalam 6 kelas, masing-masing berjumlah 1.100 citra. Pembagian data dilakukan dengan proporsi 80% untuk training 20% untuk testing, dan untuk validasi diambil secara acak sebanyak 20% dari data training menggunakan fungsi “ImageDataGenerator” dari library TensorFlow. Data validasi berfungsi untuk meningkatkan akurasi serta kualitas prediksi pada model klasifikasi. Dengan adanya data validasi, performa model menjadi lebih konsisten dan tidak mudah terpengaruh oleh variasi data, sehingga hasil prediksi yang diperoleh lebih optimal. Selain itu, proses validasi juga berperan penting dalam menangani permasalahan seperti adanya data yang hilang (missing value), yang apabila diabaikan dapat menurunkan kualitas data secara keseluruhan maupun kinerja model.

2.4. Pemodelan dengan Arsitektur VGG-19

Pada tahap ini dilakukan pemilihan model CNN yang paling sesuai untuk mengklasifikasikan penyakit pada daun tomat berdasarkan data yang tersedia. Pembangunan model CNN dilakukan dengan memanfaatkan library Python seperti Keras dan TensorFlow. CNN termasuk dalam kategori Deep Neural Network karena sering diterapkan pada data citra dan memiliki arsitektur jaringan yang dalam. Meskipun cara kerjanya mirip dengan jaringan saraf pada umumnya, perbedaannya terletak pada setiap lapisan dalam CNN yang berbentuk dua dimensi dan menjalankan proses konvolusi terhadap input dari lapisan sebelumnya[7] . Struktur utama pada CNN terdiri atas convolutional layer, pooling layer, dan fully connected layer.

Proses dimulai dengan pembuatan model, dilanjutkan dengan ekstraksi fitur melalui operasi konvolusi dan fungsi aktivasi. Kemudian, dilakukan pooling untuk mengurangi dimensi data, dan diakhiri dengan proses klasifikasi menggunakan softmax untuk menentukan kelas citra secara akhir. Selama proses klasifikasi, citra berukuran 224x224 piksel akan mengalami transformasi bertahap melalui setiap lapisan. Lapisan konvolusi pertama menghasilkan peta fitur (feature map) dengan 64 channel, kemudian dilakukan pooling untuk mengurangi ukuran spasial menjadi 112x112. Proses ini dilanjutkan secara berjenjang melalui lapisan konvolusi berikutnya, hingga diperoleh representasi fitur yang semakin dalam dan kompleks namun lebih ringkas secara dimensi. Setelah melalui seluruh lapisan konvolusi dan pooling, fitur citra diratakan (flatten) menjadi vektor satu dimensi dan diproses melalui tiga fully connected layer. Akhirnya, fungsi aktivasi softmax menghasilkan probabilitas dari setiap kelas dan menentukan kelas citra berdasarkan probabilitas tertinggi. Tahapan ini memungkinkan model VGG-19 mengubah citra mentah menjadi representasi numerik yang dapat dianalisis secara akurat oleh sistem klasifikasi.

2.5. Pelatihan Model

Proses pelatihan model memanfaatkan 5.280 citra pada data latih yang terbagi ke dalam enam kelas penyakit daun tomat, yaitu Tomato Septoria leaf spot, Tomato Spider mites Two-spotted spider mite, Tomato Leaf Mold, Tomato Target Spot, Tomato healthy, dan Tomato Late blight. Selain itu, sebanyak 660 citra digunakan sebagai data validasi untuk memonitor performa model selama proses pelatihan berlangsung. Pelatihan model ini menggunakan algoritma optimisasi Adam untuk mempercepat konvergensi dan meningkatkan akurasi.

2.6 Evaluasi Model

Dalam evaluasi model menggunakan confusion matrix, dilakukan perhitungan terhadap empat elemen penting, yaitu True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN). Nilai-nilai ini menjadi dasar dalam menghitung empat metrik utama yang sering digunakan untuk menilai kinerja model klasifikasi, yaitu precision, recall, F1 score, dan accuracy. Pemilihan metrik tersebut didasarkan pada keefektifannya dalam menangani tugas klasifikasi serta karena sering digunakan dalam berbagai penelitian. Accuracy menunjukkan seberapa tepat model dalam mengklasifikasikan data pada dataset uji. Precision menggambarkan ketepatan model dalam mengidentifikasi hasil positif yang benar pada kedua kelas. Sementara itu, recall mengukur seberapa baik model mampu mengenali kasus positif yang sebenarnya. F1 score sendiri merupakan gabungan dari precision dan recall, dengan nilai antara 0 hingga 1, yang mencerminkan kemampuan model dalam membedakan antara kategori secara seimbang.

3. HASIL DAN PEMBAHASAN

3.1. Pengumpulan dataset

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari situs web www.kaggle.com. Kaggle merupakan platform yang menyediakan berbagai dataset untuk keperluan machine learning dan deep learning. Dataset ini diperoleh dengan menggunakan kata kunci "Tomato Leaf Diseases". Dataset tersebut berisi citra berwarna dari daun tanaman tomat yang terinfeksi penyakit. Secara keseluruhan, terdapat 6.600 citra yang terbagi ke dalam enam kategori penyakit, seperti yang diuraikan dalam Tabel 1

Tabel 1 Dataset

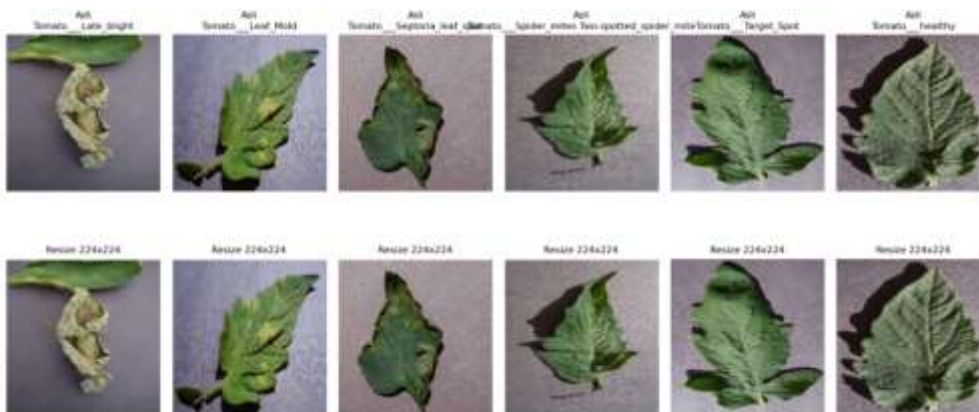
No	Jenis Penyakit	Jumlah
1	Daun sehat	1100
2	<i>Target spot</i> (Bintik Bercak)	1100
3	<i>Late blight</i> (Daun busuk)	1100
4	<i>Leaf mold</i> (jamur daun)	1100
5	<i>Spider Mites Two-Spotted Spider Mite</i> (Tungau laba-laba berbintik dua)	1100
6	<i>Septoria Leaf Spot</i> (Bercak daun septoria)	1100

3.2. Preprocessing Data

Pada tahap preprocessing ini dilakukan dua tahap yaitu resize, normalisasi dan augmentasi. Langkah-langkah ini dilakukan untuk memastikan dataset siap digunakan dalam pelatihan model. Adapun tahapannya yg diuraikan dibawah ini:

1. Resize Data

Sebelum dataset dimasukkan ke dalam model, terlebih dahulu dilakukan proses mengubah resolusi citra menjadi 224×224×3 piksel. Langkah ini dilakukan untuk menyesuaikan ukuran citra dengan kebutuhan arsitektur model serta membantu mengurangi dimensi data, sehingga proses pelatihan dapat berlangsung lebih efisien dan fokus pada objek yang dianalisis. Pada Gambar 3 merupakan hasil dari preprocessing tahap resizing data



Gambar 3 Resize data

Pada Gambar 3 langkah ini digunakan untuk menyesuaikan ukuran (resize) seluruh gambar pada dataset yang telah seimbang sehingga memiliki dimensi yang seragam 224x224 piksel.

2. Normalisasi data

Tahap selanjutnya dalam preprocessing adalah normalisasi, yaitu mengubah skala nilai piksel menjadi rentang 0 hingga 1. Proses ini dilakukan untuk membantu model belajar secara lebih optimal dan meningkatkan akurasi selama pelatihan.

3. Augmentasi Data

```
# 3. IMAGE GENERATORS
# =====
datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest')
```

Gambar 4 Augmentasi Data

Pada Gambar 4 ImageDataGenerator digunakan untuk melakukan augmentasi pada data gambar secara acak selama proses pelatihan. Augmentasi yang diterapkan meliputi rotasi hingga 20 derajat (rotation_range=20), pergeseran horizontal dan vertikal hingga 10% dari ukuran asli (width_shift_range=0.1 dan height_shift_range=0.1), kemiringan hingga 10% (shear_range=0.1), dan pembesaran hingga 20% (zoom_range=0.2). Selain itu, gambar juga dibalik secara horizontal dan vertikal (horizontal_flip=True dan vertical_flip=True). Area kosong yang dihasilkan dari transformasi diisi menggunakan piksel terdekat (fill_mode='nearest') agar hasil augmentasi tetap terlihat alami.

3.3. Pembagian Data

Setelah proses preprocessing selesai, data kemudian dipisahkan menjadi tiga bagian, yaitu data training, data testing, dan data validasi. Pembagiannya menggunakan proporsi 80% untuk data training dan 20% untuk data testing, sementara data validasi diambil dari 20% bagian data training.

Pemisahan data dilakukan menggunakan library TensorFlow dengan jenis pembagian kelas categorical classification. Dataset yang digunakan merupakan dataset daun tomat yang terbagi menjadi 6 kelas, yaitu Tomato Septoria leaf spot, Tomato spider mite Two-spotted spider mite, Tomato Leaf Mold, Tomato Target Spot, Tomato healthy, dan Tomato late blight. Jumlah data yang digunakan adalah 5.280 gambar untuk data latih (training), 660 gambar untuk data validasi (validation), dan 660 gambar untuk data uji (testing), dengan total keseluruhan 6.600 gambar. Gambar 5 dibawah merupakan hasil dari pembagian data

```
Found 5280 images belonging to 6 classes.
Found 660 images belonging to 6 classes.
Found 660 images belonging to 6 classes.
```

Gambar 5 Pembagian data

3.4 Pemodelan dengan arsitektur VGG-19

Pada Gambar 6 dibawah merupakan bagian dari proses pemodelan menggunakan Convolutional Neural Network (CNN), yang dalam hal ini memanfaatkan arsitektur VGG-19 sebagai base model atau model dasar. VGG-19 dipanggil dengan bobot awal (pre-trained weights) dari ImageNet dan parameter include_top=False, yang berarti lapisan klasifikasi bawaan VGG-19 tidak digunakan sehingga model hanya berfungsi sebagai feature extractor untuk mengekstrak ciri-ciri penting dari gambar input berukuran 224x224 piksel dengan 3 kanal warna (RGB). Pada tahap awal, seluruh lapisan VGG-19 dibekukan (trainable=False) agar bobot yang sudah dilatih dari ImageNet tidak berubah, sehingga pelatihan awal hanya berfokus pada lapisan tambahan di bagian akhir (custom classifier).

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 6)	3,078

Total params: 20,027,462 (76.40 MB)
 Trainable params: 3,078 (12.02 KB)
 Non-trainable params: 20,024,384 (76.39 MB)

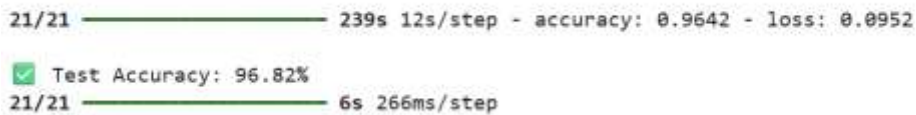
Gambar 6 Pemodelan dengan arsitektur VGG-19

Keluaran dari model dasar ini kemudian diproses menggunakan lapisan GlobalAveragePooling2D, yang bertugas meratakan (spatial pooling) hasil ekstraksi fitur menjadi satu vektor tanpa kehilangan informasi penting. Selanjutnya, ditambahkan lapisan Dropout dengan rasio 30% (Dropout(0.3)), yang berfungsi mengacak sebagian neuron saat pelatihan guna mengurangi risiko overfitting. Selanjutnya, output tersebut diteruskan ke lapisan Dense sebagai lapisan keluaran, dengan jumlah neuron yang disesuaikan dengan jumlah kelas pada dataset pelatihan (train_generator.num_classes) dan menggunakan fungsi aktivasi softmax untuk menghasilkan distribusi probabilitas pada setiap kelas. Model ini kemudian dikompilasi menggunakan optimizer Adam dengan learning rate 0,001 (1e-3), fungsi loss categorical_crossentropy yang sesuai untuk klasifikasi multikelas, serta metrik evaluasi accuracy untuk memantau kinerja model selama proses pelatihan. Pendekatan ini menggabungkan keunggulan transfer learning dari arsitektur CNN VGG-19 yang telah dilatih sebelumnya dengan lapisan klasifikasi khusus yang disesuaikan untuk dataset baru.

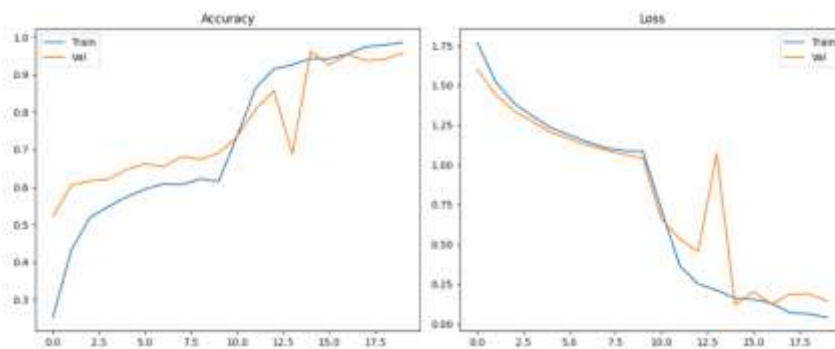
3.5 Pelatihan Model

Setelah membangun model menggunakan arsitektur VGG-19 yang termasuk dalam kategori Convolutional Neural Network (CNN), tahap berikutnya adalah melakukan proses pelatihan. Pelatihan dilakukan dengan memanfaatkan data training dan data validasi yang telah disiapkan. Proses optimasi menggunakan algoritma Adam dengan learning rate 0.001, yang berfungsi memperbarui bobot model berdasarkan gradien dari fungsi loss. Adapun fungsi loss yang diterapkan adalah categorical_crossentropy, karena permasalahan yang diselesaikan merupakan klasifikasi dengan banyak kelas.

Selama proses pelatihan, model memantau metrik accuracy untuk menilai kinerjanya pada setiap epoch. Pelatihan dilakukan melalui fungsi model.fit dengan total 40 epoch. Hasil pelatihan menunjukkan adanya peningkatan accuracy pada data training secara bertahap hingga mencapai 98,28% pada epoch ke-10, disertai penurunan nilai loss hingga 0,0501. Sementara itu, validation accuracy relatif stabil di kisaran 92–95%, yang mengindikasikan kemampuan model dalam melakukan generalisasi terhadap data baru. Adapun output yang dihasilkan dalam pelatihan sebagai berikut:



Gambar 7 output pelatihan model

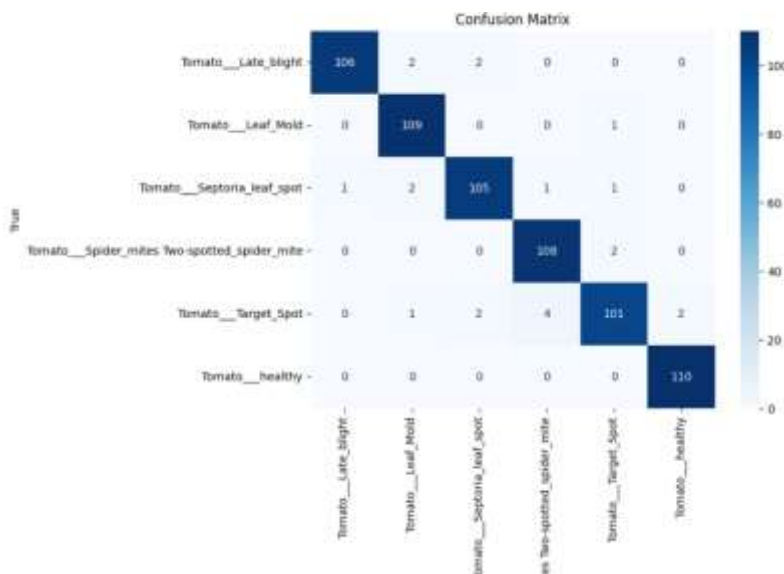


Gambar 8 Grafik dan Loss

Dari Gambar 8 diatas dapat disimpulkan bahwa pada awal pelatihan, model memiliki akurasi yang masih rendah dengan nilai loss yang tinggi. Seiring bertambahnya epoch, akurasi pada data pelatihan dan validasi meningkat secara signifikan, sementara nilai loss keduanya menurun tajam. Peningkatan akurasi berlangsung cukup stabil hingga mendekati akhir pelatihan, di mana validation accuracy mencapai kisaran 90–95% dan training accuracy mendekati 100%, menunjukkan kemampuan generalisasi yang baik. Tidak ditemukan tanda overfitting yang berarti, karena pola akurasi dan loss pada data pelatihan maupun validasi menunjukkan kecenderungan yang konsisten hingga epoch terakhir.

3.6 Evaluasi Model

Setelah proses pengujian model selesai, dilakukan analisis untuk menjelaskan performa model yang diperoleh. Kinerja model disajikan menggunakan confusion matrix, yang kemudian digunakan untuk menghitung metrik evaluasi seperti Accuracy, Precision, Recall, dan F1-Score dari model yang telah dibangun. Pada Gambar 9 dibawah merupakan hasil confusion matrix yang berupa grafik



Gambar 9 Confusion Matrix

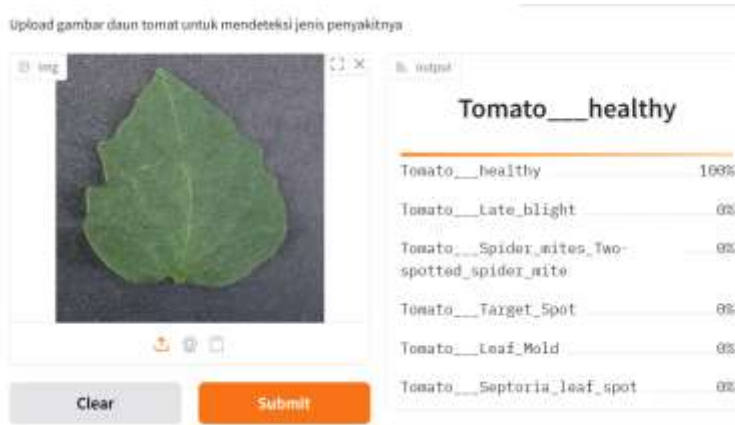
Adapun classification report yang diuraikan pada gambar dibawah ini:

Classification Report:

	precision	recall	f1-score	support
Tomato__Late_blight	0.99	0.96	0.98	110
Tomato__Leaf_Mold	0.96	0.99	0.97	110
Tomato__Septoria_leaf_spot	0.96	0.95	0.96	110
Tomato__Spider_mites Two-spotted_spider_mite	0.96	0.98	0.97	110
Tomato__Target_Spot	0.96	0.92	0.94	110
Tomato__healthy	0.98	1.00	0.99	110
accuracy			0.97	660
macro avg	0.97	0.97	0.97	660
weighted avg	0.97	0.97	0.97	660

Gambar 10 Classification Report

3.7 Sistem Identifikasi Penyakit Daun tomat



Gambar 11 Sistem Identifikasi Penyakit Daun Tomat

Gambar 11 menunjukkan sistem deteksi penyakit daun tomat yang dirancang untuk membantu pengguna mengidentifikasi jenis penyakit secara otomatis. Pengguna cukup mengunggah foto daun tomat melalui fitur upload, baik dengan menyeret gambar ke area yang disediakan maupun dengan memilih file langsung dari perangkat. Setelah gambar berhasil diunggah, sistem akan memproses citra tersebut menggunakan model deep learning yang telah dilatih—dalam hal ini arsitektur VGG-19—untuk mengenali pola dan karakteristik visual yang khas dari setiap jenis penyakit. Hasil deteksi kemudian ditampilkan pada bagian output, yang memuat informasi mengenai jenis penyakit yang teridentifikasi. Dengan antarmuka yang sederhana, sistem ini dapat membantu petani maupun peneliti untuk memperoleh diagnosa cepat tanpa harus memiliki keahlian khusus di bidang patologi tanaman, sehingga dapat mempercepat proses penanganan dan pencegahan penyebaran penyakit.

4. KESIMPULAN

Penelitian ini mampu menghasilkan sebuah sistem yang dapat mendeteksi penyakit pada daun tomat secara dini, dengan memanfaatkan metode CNN dan memodelkannya menggunakan arsitektur VGG-19. Sistem mampu mengklasifikasikan enam kelas penyakit maupun kondisi sehat melalui tahapan preprocessing, augmentasi data, serta implementasi antarmuka berbasis Gradio. Optimasi model dengan transfer learning (ImageNet), augmentasi variatif, dan fine-tuning lapisan akhir meningkatkan performa hingga rata-rata precision, recall, dan f1-score sebesar 97%, menunjukkan konsistensi model dalam mengenali pola visual daun tomat. Penelitian selanjutnya disarankan untuk menambah jumlah citra pada kelas dengan performa rendah, menerapkan variasi augmentasi dan fine-tuning parameter yang lebih optimal, menguji arsitektur CNN lain atau model ensemble, serta mengintegrasikan sistem ke aplikasi web atau mobile agar lebih mudah digunakan di lapangan.

DAFTAR PUSTAKA

- [1] P. Palupiningsih, A. R. Sujiwanto, and R. R. B. P. Prawirodirjo, "Analisis Perbandingan Performa Model Klasifikasi Kesehatan Daun Tomat menggunakan arsitektur VGG, MobileNet, dan Inception V3," *J. Ilmu Komput. dan Agri-Informatika*, vol. 10, no. 1, pp. 98–110, 2023, doi: 10.29244/jika.10.1.98-110.
- [2] Mungki Astiningrum, P. P. Arhandi, and N. A. Ariditya, "Identifikasi Penyakit Pada Daun Tomat Berdasarkan Fitur Warna Dan Tekstur," *J. Inform. Polinema*, vol. 6, no. 2, pp. 47–50, 2020, doi: 10.33795/jip.v6i2.320.
- [3] J. Basavaiah and A. Arlene Anthony, "Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques," *Wirel. Pers. Commun.*, vol. 115, no. 1, pp. 633–651, 2020, doi: 10.1007/s11277-020-07590-x.
- [4] R. Shinta, "klasifikasi citra penyakit daun tanaman padi menggunakan CNN dengan arsitektur vgg-19," *J. Ekon. Vol. 18, Nomor 1 Maret201*, vol. 2, no. 1, pp. 41–49, 2023.
- [5] A. D. P. Wicaksono and A. Amrulloh, "Klasifikasi Tingkat Kematangan Buah Pisang Cavendish Menggunakan Convolutional Neural Network (Cnn) Dengan Model Vgg-19," *positif J. Sist. dan Teknol. Inf.*, vol. 9, no. 2, pp. 107–114, 2023, doi: 10.31961/positif.v9i2.1778.
- [6] Y. A. Gerhana, R. R. Heryanto, and U. Syaripudin, "Implementation of Convolutional Neural Network CNN Algorithm to Detect Coffe Fruit Maturity," vol. 13, no. 2, pp. 47–50, 2024.

- [7] G. Y. Alhafis, J. Jasril, S. Sanjaya, F. Syafria, and E. Budianita, "Klasifikasi Citra Daging Sapi dan Daging Babi Menggunakan Ekstraksi Ciri dan Convolutional Neural Network," *Jurikom (Jurnal Ris. Komputer)*, vol. 9, no. 3, p. 653, 2022, doi: 10.30865/jurikom.v9i3.4175.
- [8] Raynold, Muhammad. Alva Hendi, "Deteksi Dan Klasifikasi Penyakit Daun Tomat Menggunakan ResNet-50," *J. Coscitech*, Vol. 6, No. 1, April 2025, hal. 9-20.
- [9] Okamisar, Hayami. Regiolina, Fuad. Evans, "Identifikasi penyakit tanaman tomat melalui citra daun menggunakan DenseNet201," *J. Coscitech*, Vol. 6, No. 2, Agustus 2025, hal. 166-174
- [10] Walascha, A., Febriana, A., Saputri, D., Sri Nur Haryanti, D., Tsania, R., Sanjaya, Y., & Priyanti. (2021). Review Artikel: Inventarisasi Jenis Penyakit yang Menyerang Daun Tanaman Padi (*Oryza sativa* L.). *Prosiding Seminar Nasional Biologi*, 1(2), 471-478. <https://semnas.biologi.fmipa.unp.ac.id/index.php/prosiding/article/download/150/300>