

Penerapan Deep Learning untuk Klasifikasi Penyakit Tomat melalui Citra Daun

Alibi Galih Exsandi¹, Aly Rahmat Nanda Saputra²

^{1,2}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: *¹alibigalih21@gmail.com, ²alynandasaputra@gmail.com

Abstrak – Tanaman tomat (*Solanum lycopersicum*) merupakan komoditas hortikultura bernilai ekonomi tinggi, namun produktivitasnya sering menurun akibat serangan berbagai penyakit daun. Proses identifikasi penyakit daun tomat secara manual memerlukan keahlian khusus dan waktu yang relatif lama, sehingga penanganan sering terlambat dan kurang efektif. Oleh karena itu, diperlukan solusi berbasis teknologi yang mampu mendeteksi penyakit daun tomat secara cepat dan akurat. Penelitian ini bertujuan mengembangkan sistem klasifikasi penyakit daun tomat menggunakan metode Deep Learning dengan arsitektur Convolutional Neural Network (CNN) yang diintegrasikan ke dalam aplikasi web. Citra daun tomat diproses melalui tahapan pra-pemrosesan berupa resize, normalisasi, serta data augmentation untuk meningkatkan kemampuan generalisasi model. CNN digunakan untuk mengekstraksi fitur visual dan mengklasifikasikan berbagai jenis penyakit daun tomat secara otomatis. Evaluasi performa model dilakukan menggunakan confusion matrix dan metrik akurasi, precision, recall, serta F1-score. Hasil penelitian menunjukkan bahwa model CNN mampu mengklasifikasikan penyakit daun tomat dengan performa yang baik dan konsisten. Sistem yang dikembangkan dapat digunakan sebagai alat bantu deteksi dini penyakit daun tomat, sehingga diharapkan mampu membantu petani dalam menentukan tindakan penanganan yang tepat serta meningkatkan produktivitas dan kualitas hasil panen tomat.

Kata Kunci — citra daun, CNN, deep learning, klasifikasi, penyakit tomat

1. PENDAHULUAN

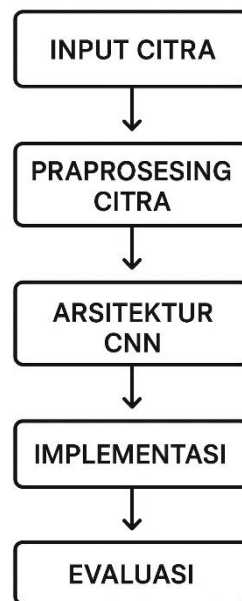
Tomat (*Solanum lycopersicum*) merupakan salah satu komoditas hortikultura yang memiliki nilai ekonomi tinggi di Indonesia,[1]. Namun, produktivitas tanaman ini sering menurun akibat serangan berbagai penyakit daun seperti *Late Blight*, *Leaf Mold*, *Septoria Leaf Spot*, *Powdery Mildew*, *Target Spot*, *Tomato Mosaic Virus*, *Tomato Yellow Leaf Curl Virus*, *healthy*, *Early_blight* serta hama Spider Mites,[2]. Serangan penyakit tersebut menyebabkan penurunan kualitas hasil panen dan kerugian bagi petani,[3].

Petani sering kali kesulitan mengidentifikasi jenis penyakit secara cepat karena proses diagnosis manual memerlukan keahlian khusus dan waktu yang lama. Hal ini sejalan dengan penelitian yang menyebutkan bahwa berbagai penyakit seperti septoria leaf spot, leaf mold, late blight, dan target spot merupakan penyebab utama penurunan kualitas tanaman tomat di berbagai wilayah tropis [4].

Di wilayah tropis membutuhkan waktu yang lama dalam proses identifikasi membuat penanganan penyakit menjadi kurang efektif. Kondisi ini menunjukkan perlunya penerapan teknologi yang mampu mendeteksi penyakit daun tomat secara cepat dan akurat[5]Salah satu solusi yang dapat digunakan adalah pemanfaatan teknologi Deep Learning, khususnya *Convolutional Neural Network* (CNN), yang memiliki kemampuan mengenali pola penyakit pada citra daun secara otomatis[6]Teknologi ini terbukti mampu mempercepat proses identifikasi penyakit tanaman karena sistem dapat melakukan klasifikasi langsung berdasarkan gambar daun tanpa memerlukan keahlian manual yang mendalam [7]

Otomatisasi dalam penerapan teknologi ini kemudian dikembangkan menjadi aplikasi web berbasis Streamlit yang dapat melakukan klasifikasi penyakit daun tomat, menampilkan rekomendasi penanganan, serta menyimpan riwayat hasil prediksi. Melalui penelitian ini, tujuan yang ingin dicapai adalah menghasilkan sebuah sistem yang mampu membantu petani mengidentifikasi penyakit daun tomat secara lebih cepat dan akurat[8]Aplikasi ini diharapkan membantu petani mengenali penyakit lebih dini, melakukan tindakan penanganan yang tepat, serta meningkatkan produktivitas dan kualitas hasil panen tomat di Indonesia.

2. METODE PENELITIAN



Gambar 1 Arsitektur Diagram CNN

2.1 Input Citra

Input citra merupakan tahap awal yang sangat menentukan dalam proses pengolahan citra menggunakan *Convolutional Neural Network* (CNN), karena seluruh proses analisis visual bergantung pada kualitas gambar yang pertama kali dimasukkan ke sistem. Pada penelitian ini, citra daun diunggah oleh pengguna melalui antarmuka aplikasi, sehingga citra masuk dalam kondisi mentah dengan berbagai variasi seperti perbedaan ukuran, pencahayaan, orientasi daun, serta kemungkinan adanya noise. Variasi tersebut dapat memengaruhi kemampuan CNN dalam menangkap pola tekstur dan bentuk yang menjadi ciri penyakit daun, sehingga konsistensi citra input menjadi faktor kunci.

Penelitian yang dibahas [9] menjelaskan bahwa kualitas representasi awal sangat memengaruhi performa CNN, karena model hanya dapat mempelajari fitur secara optimal apabila citra memiliki struktur visual yang cukup stabil. Artikel tersebut juga menegaskan bahwa tahap input citra merupakan fondasi yang menentukan apakah jaringan mampu mengenali pola yang lebih kompleks pada tahap ekstraksi fitur berikutnya. Oleh sebab itu, dalam penelitian ini, tahap input citra tidak sekadar proses memasukkan gambar, tetapi merupakan langkah krusial untuk memastikan akurasi dan efektivitas model CNN dalam mengidentifikasi penyakit pada daun.

2.2 Praprosesing Citra

Praprosesing citra merupakan tahapan penting dalam mempersiapkan gambar sebelum diproses oleh CNN, karena tahapan ini memastikan citra memiliki kualitas dan struktur yang konsisten sehingga jaringan dapat mengekstraksi fitur secara optimal. Pada penelitian ini, citra daun yang diunggah pengguna terlebih dahulu di-*resize* ke ukuran 128×128 piksel dan dinormalisasi dengan membagi nilai piksel menjadi rentang 0–1 agar proses pelatihan lebih stabil. Selain itu, untuk meningkatkan keanekaragaman data pelatihan, dilakukan *data augmentation* berupa rotasi, *flipping*, *zooming*, dan pergeseran gambar. Teknik augmentasi ini sangat penting untuk membantu model mengenali variasi bentuk dan kondisi daun yang mungkin berbeda-beda di dunia nyata.

Hal ini sejalan dengan survei komprehensif yang dijelaskan [10], bahwa augmentasi seperti rotasi, flipping, scaling, serta perubahan pencahayaan terbukti mampu meningkatkan kemampuan generalisasi CNN secara signifikan, terutama ketika dataset asli terbatas atau tidak seimbang. Penelitian tersebut menegaskan bahwa augmentasi tidak hanya memperbanyak variasi data, tetapi juga membantu CNN lebih tahan terhadap overfitting dengan memberikan

paparan terhadap berbagai kondisi visual. Dengan demikian, tahapan praprosesing khususnya normalisasi dan augmentasi memiliki peran krusial dalam memastikan model CNN dapat mempelajari pola citra secara akurat, stabil, dan representatif.

2.3 Arsitektur CNN

Convolutional Neural Network (CNN) merupakan arsitektur jaringan saraf dalam yang dirancang untuk mengekstraksi fitur visual melalui proses konvolusi yang dilakukan secara berlapis. CNN terdiri dari komponen utama seperti *convolution layer*, *activation function*, *pooling layer*, dan *fully connected layer* yang bekerja secara hierarkis untuk menangkap pola visual dari tingkat sederhana hingga kompleks. Dalam artikel yang dipublikasikan [11], dijelaskan bahwa CNN mampu mempertahankan struktur spasial citra melalui penggunaan filter yang bergerak secara lokal, sehingga jaringan dapat mengenali fitur penting seperti tepi, tekstur, dan bentuk objek tanpa memerlukan teknik ekstraksi fitur manual. Kemampuan ini membuat CNN efektif dalam mempelajari representasi visual yang semakin mendalam ketika jumlah lapisan konvolusi meningkat.

Sementara itu, penelitian dalam Sensors [12] menekankan bahwa performa CNN pada tugas visual modern sangat dipengaruhi oleh inovasi arsitektural seperti *batch normalization*, *residual connection*, dan *dropout* yang berfungsi mengatasi permasalahan *vanishing gradient* serta meningkatkan kemampuan generalisasi model. Artikel tersebut juga menyoroti bahwa CNN dapat dibuat lebih efisien melalui pengurangan jumlah parameter, pengaturan ukuran kernel, serta penyusunan blok konvolusi yang lebih ringan. Berdasarkan kedua referensi tersebut, arsitektur CNN merupakan inti dari sistem pengolahan citra modern karena kemampuannya dalam belajar otomatis dari data, fleksibel terhadap variasi visual, serta dapat dioptimalkan sesuai kebutuhan aplikasi.

2.4 Implementasi

Implementasi CNN pada penelitian ini dilakukan melalui beberapa tahapan, mulai dari persiapan dataset, pembangunan model, hingga proses pelatihan dan pengujian. Dataset citra daun yang telah dikumpulkan terlebih dahulu dibagi menjadi data pelatihan, validasi, dan pengujian sebelum dimasukkan ke dalam alur praprosesing. Setelah praprosesing selesai, model CNN dibangun menggunakan serangkaian *convolution layer* yang dikombinasikan dengan fungsi aktivasi ReLU untuk mengekstraksi fitur visual secara bertahap. Struktur model yang diterapkan sejalan dengan penjelasan dalam artikel [11], yang menunjukkan bahwa susunan lapisan konvolusi dan pooling secara bertingkat mampu menghasilkan representasi fitur yang lebih kaya dan akurat. Selanjutnya, lapisan *flatten* dan *fully connected* digunakan untuk memetakan fitur yang diekstraksi menjadi kelas penyakit daun yang akan diprediksi.

Pada tahap pelatihan, digunakan algoritma optimasi seperti Adam untuk menurunkan nilai *loss* melalui mekanisme *backpropagation*. Implementasi ini diperkuat dengan teknik regulasi seperti *dropout* dan *batch normalization*, sebagaimana dijelaskan pada artikel [12], yang berfungsi mencegah *overfitting* serta meningkatkan stabilitas model selama proses pelatihan. Selain itu, proses *early stopping* diterapkan untuk menghentikan pelatihan ketika akurasi tidak lagi meningkat, sehingga model tidak mengalami penurunan performa akibat pelatihan berlebihan. Setelah model CNN selesai dilatih, tahap implementasi terakhir dilakukan melalui integrasi model ke dalam aplikasi berbasis Streamlit, sehingga pengguna dapat mengunggah citra daun secara langsung dan memperoleh hasil klasifikasi penyakit dalam bentuk prediksi probabilistik. Dengan mengikuti tahapan implementasi ini, model CNN dapat bekerja secara efektif, stabil, dan sesuai dengan kebutuhan aplikasi deteksi penyakit daun.

2.5 Evaluasi

Evaluasi model CNN dilakukan untuk mengetahui sejauh mana jaringan mampu mengklasifikasikan penyakit daun dengan akurat dan konsisten. Pada penelitian ini, proses evaluasi dilakukan menggunakan sejumlah metrik, yaitu akurasi, precision, recall, dan F1-score

yang dihitung berdasarkan hasil prediksi model terhadap data uji. Selain itu, digunakan juga *confusion matrix* untuk melihat pola kesalahan klasifikasi secara lebih rinci pada setiap kelas penyakit daun. Penggunaan *confusion matrix* sangat penting karena matriks ini dapat menunjukkan jumlah prediksi benar dan salah pada tiap kelas, serta membantu mengidentifikasi kelas yang sering tertukar. Hal ini sejalan dengan penjelasan dalam artikel Confusion Matrix-Based Performance Evaluation Metrics yang dipublikasikan dalam [13] merupakan dasar dalam evaluasi model klasifikasi karena mampu menghasilkan metrik turunan seperti precision, recall, dan F1-score yang mencerminkan performa model secara lebih komprehensif.

Akurasi mengukur seberapa besar proporsi prediksi yang benar secara keseluruhan, dengan rumus:

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

Precision digunakan untuk mengetahui ketepatan model ketika memprediksi suatu kelas, terutama untuk menghindari kesalahan positif palsu. Rumus precision adalah:

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

Sementara recall (atau sensitivitas) mengukur kemampuan model mendeteksi semua data positif sebenarnya, dengan rumus:

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

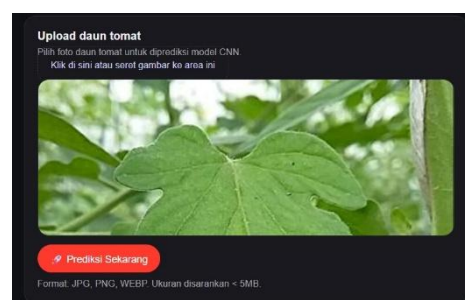
Untuk memberikan penilaian yang lebih seimbang antara precision dan recall, digunakan F1-score sebagai rata-rata harmonik keduanya:

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

Sementara itu, artikel Evaluation metrics and statistical tests for machine learning yang diterbitkan dalam [14] menegaskan bahwa metrik akurasi saja tidak cukup untuk menilai kualitas model, terutama pada dataset dengan distribusi kelas yang tidak merata. Oleh sebab itu, penggunaan kombinasi metrik seperti precision, recall, dan F1-score menjadi sangat penting untuk menggambarkan sensitivitas model terhadap tiap kelas serta mengukur keseimbangan antara prediksi benar dan salah. Berdasarkan kedua referensi tersebut, evaluasi pada penelitian ini dilakukan secara menyeluruh sehingga hasil yang diperoleh mampu mencerminkan kemampuan model CNN dalam mengenali penyakit daun secara akurat dan dapat dipercaya

3. HASIL DAN PEMBAHASAN

3.1 Input citra



Gambar 2 Input Citra

Tampilan antarmuka ini menunjukkan proses input citra sebagai tahap awal dalam sistem deteksi penyakit daun tomat berbasis CNN, di mana pengguna mengunggah gambar daun melalui fitur drag-and-drop sebelum sistem melakukan pra-pemrosesan dan prediksi. Pada tahap ini, citra asli dari pengguna menjadi data masukan yang nantinya diolah oleh model untuk mengidentifikasi kondisi daun secara otomatis. Diagram pada Gambar 2 menjelaskan alur umum mulai dari input citra, proses praprosesing, pemrosesan oleh model CNN, hingga

keluaran berupa hasil klasifikasi. Dengan demikian, gambar ini menggambarkan kerangka kerja utama sistem yang menjadi dasar proses pendeteksian penyakit daun.

3.2 Pra-pemrosesan

```
# Parameter utama
# -----
IMG_SIZE = 128 # ukuran gambar (128x128 piksel)
BATCH_SIZE = 32
EPOCHS = 20
DATA_DIR = "Dataset"

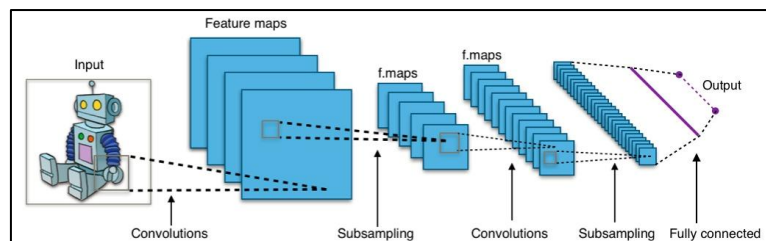
# Load & Augmentasi Data
# -----
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True
)

val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
```

Gambar 3 Pra-pemrosesan

Gambar 3 menunjukkan proses pra-pemrosesan citra sebelum dimasukkan ke dalam model CNN. Pada tahap ini, ukuran gambar diseragamkan menjadi 128×128 piksel dan nilai piksel dinormalisasi menggunakan $\text{rescale} = 1./255$ agar model dapat memproses citra dengan lebih stabil. Untuk data latih dilakukan augmentasi berupa rotasi, pergeseran, zoom, dan pembalikan horizontal guna menambah variasi citra sehingga model menjadi lebih robust dan tidak mudah mengalami overfitting. Sementara itu, data validasi dan data uji hanya melalui proses normalisasi tanpa augmentasi agar evaluasi model tetap akurat. Seluruh langkah ini memastikan bahwa citra sudah berada dalam kondisi optimal sebelum digunakan dalam pelatihan dan pengujian model CNN.

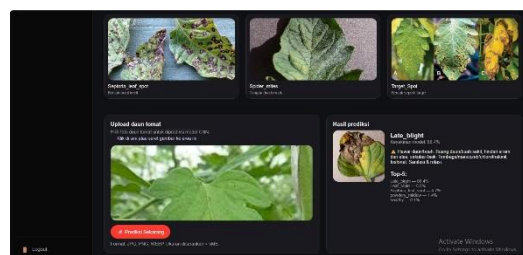
3.3 Arsitektur CNN



Gambar 4 Arsitektur CNN[15].

Gambar 4 menunjukkan arsitektur Convolutional Neural Network (CNN) secara keseluruhan. CNN memulai proses dengan input citra, yang kemudian diproses melalui lapisan konvolusi untuk mengekstraksi fitur lokal seperti tepi, sudut, atau pola tertentu. Hasil konvolusi menghasilkan feature maps, yang kemudian diperkecil dimensinya melalui subsampling atau pooling untuk mempertahankan fitur penting sekaligus meningkatkan efisiensi. Proses konvolusi dan pooling ini dapat diulang beberapa kali agar model dapat menangkap fitur yang semakin kompleks. Setelah itu, fully connected layer mengubah fitur menjadi vektor satu dimensi yang diteruskan ke lapisan output untuk melakukan klasifikasi, sehingga CNN mampu mengenali objek atau pola dalam citra secara akurat.

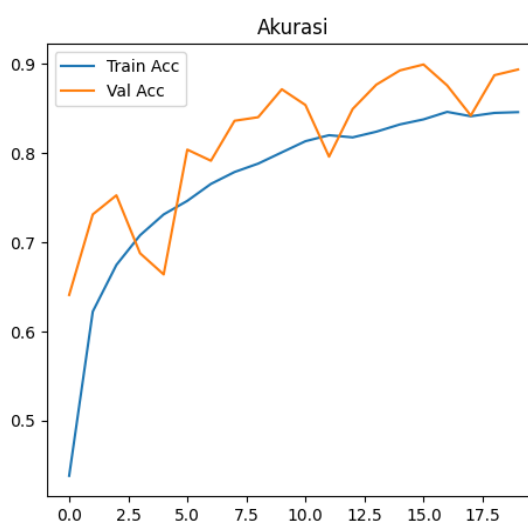
3.4 Implementasi Sistem



Gambar 5 Implementasi Sistem

Gambar 5 memperlihatkan tampilan aplikasi web yang digunakan untuk klasifikasi penyakit daun tomat berbasis Convolutional Neural Network (CNN). Pada antarmuka ini, pengguna mengunggah citra daun melalui fitur upload yang telah disediakan. Setelah citra diunggah, sistem secara otomatis melakukan proses pra-pemrosesan seperti mengubah ukuran gambar menjadi 128×128 piksel dan melakukan normalisasi nilai piksel agar citra sesuai dengan standar input model. Citra yang sudah diproses kemudian dikirim ke model CNN yang terdiri dari beberapa lapisan utama, yaitu Convolution layer, MaxPooling, Flatten, Dense, dan Softmax. Model ini akan mengekstraksi fitur penting dari citra dan menghasilkan prediksi berupa kelas penyakit daun tomat. Hasil identifikasi ditampilkan pada panel hasil dalam bentuk label penyakit beserta nilai probabilitas, sehingga pengguna dapat mengetahui kondisi daun dengan cepat dan mudah.

3.5 Hasil Evaluasi



Gambar 6. Akurasi

Grafik akurasi menunjukkan bahwa nilai akurasi training meningkat secara bertahap dari awal hingga akhir epoch. Akurasi validasi juga mengalami peningkatan dengan beberapa fluktuasi, namun tetap mengikuti tren yang meningkat. Pada akhir pelatihan, nilai akurasi validasi mendekati akurasi training, yang menandakan bahwa model mampu melakukan generalisasi dengan baik terhadap data validasi dan tidak mengalami overfitting yang signifikan.

	precision	recall	f1-score	support
Early_blight	0.88	0.93	0.91	2455
Late_blight	0.81	0.91	0.86	3113
Leaf_Mold	0.94	0.89	0.91	2754
Septoria_leaf_spot	0.88	0.77	0.82	2882
Spider_mites	0.94	0.88	0.91	1747
Two-spotted_spider_mite	0.93	0.77	0.84	1827
Target_Spot	0.98	0.98	0.98	2039
Tomato_Yellow_Leaf_Curl_Virus	1.00	0.94	0.97	2153
Tomato_mosaic_virus	0.83	0.99	0.90	3051
healthy	0.82	0.79	0.80	1004
powdery_mildew				
accuracy			0.89	23025
macro avg	0.90	0.89	0.89	23025
weighted avg	0.90	0.89	0.89	23025

Gambar 7. Precision

Secara keseluruhan, nilai precision pada setiap kelas tergolong baik, dengan sebagian besar kelas memiliki nilai di atas 0,80. Hal ini menunjukkan bahwa model mampu memberikan prediksi yang cukup akurat dan menghasilkan sedikit kesalahan pada sebagian besar kelas.

	precision	recall	f1-score	support
Early_blight	0.88	0.93	0.91	2455
Late_blight	0.81	0.91	0.86	3113
Leaf_Mold	0.94	0.89	0.91	2754
Septoria_leaf_spot	0.88	0.77	0.82	2882
Spider_mites Two-spotted_spider_mite	0.94	0.88	0.91	1747
Target_Spot	0.93	0.77	0.84	1827
Tomato_Yellow_Leaf_Curl_Virus	0.98	0.98	0.98	2039
Tomato_mosaic_virus	1.00	0.94	0.97	2153
healthy	0.83	0.99	0.90	3051
powdery_mildew	0.82	0.79	0.80	1004
accuracy			0.89	23025
macro avg	0.90	0.89	0.89	23025
weighted avg	0.90	0.89	0.89	23025

Gambar 8. Recall

Gambar 8 menunjukkan nilai recall untuk masing-masing kelas pada data uji. Secara umum, sebagian besar kelas memiliki nilai recall yang tinggi, yang menandakan bahwa model mampu mengenali sebagian besar data pada kelas sebenarnya dengan baik. Beberapa kelas seperti Tomato Yellow Leaf Curl Virus, Tomato mosaic virus, dan healthy memiliki nilai recall yang sangat tinggi, sedangkan kelas seperti Septoria leaf spot dan Target Spot memiliki nilai recall yang lebih rendah, yang menunjukkan masih terdapat beberapa data pada kelas tersebut yang belum berhasil terdeteksi oleh model.

	precision	recall	f1-score	support
Early_blight	0.88	0.93	0.91	2455
Late_blight	0.81	0.91	0.86	3113
Leaf_Mold	0.94	0.89	0.91	2754
Septoria_leaf_spot	0.88	0.77	0.82	2882
Spider_mites Two-spotted_spider_mite	0.94	0.88	0.91	1747
Target_Spot	0.93	0.77	0.84	1827
Tomato_Yellow_Leaf_Curl_Virus	0.98	0.98	0.98	2039
Tomato_mosaic_virus	1.00	0.94	0.97	2153
healthy	0.83	0.99	0.90	3051
powdery_mildew	0.82	0.79	0.80	1004
accuracy			0.89	23025
macro avg	0.90	0.89	0.89	23025
weighted avg	0.90	0.89	0.89	23025

Gambar 9. F1 score

F1-score merupakan nilai gabungan antara precision dan recall yang digunakan untuk menilai keseimbangan kinerja model. Berdasarkan hasil pada gambar, sebagian besar kelas memiliki nilai F1-score yang tinggi, yaitu berada di kisaran 0,80 hingga 0,98. Kelas Tomato Yellow Leaf Curl Virus dan Tomato mosaic virus memiliki nilai F1-score tertinggi, sedangkan kelas Septoria leaf spot dan powdery mildew memiliki nilai F1-score yang relatif lebih rendah. Secara keseluruhan, nilai F1-score menunjukkan bahwa model memiliki kinerja klasifikasi yang baik dan seimbang pada sebagian besar kelas.

4. KESIMPULAN

Penelitian ini berhasil dengan mendapatkan akurasi 89% yang guna mengembangkan sistem klasifikasi penyakit daun tomat berbasis CNN terintegrasi dalam aplikasi web. Sistem memproses citra daun melalui pra-pemrosesan dan augmentasi untuk meningkatkan kemampuan generalisasi model. Arsitektur CNN mampu mengekstraksi fitur visual secara efektif, sehingga model dapat mengenali pola penyakit dengan akurat. Evaluasi menggunakan confusion matrix serta metrik akurasi, precision, recall, dan F1-score menunjukkan performa model yang baik. Sistem ini dapat menjadi alat bantu efektif untuk deteksi dini penyakit daun tomat dan berpotensi dikembangkan lebih lanjut.

5. SARAN

Sistem klasifikasi penyakit daun tomat ini dapat dikembangkan lebih lanjut dengan menambahkan jenis penyakit yang lebih beragam serta meningkatkan arsitektur CNN untuk memperoleh akurasi yang lebih tinggi. Selain itu, integrasi fitur tambahan, seperti rekomendasi pengobatan atau pemantauan pertumbuhan tanaman, akan membuat aplikasi lebih bermanfaat dan praktis bagi pengguna.

DAFTAR PUSTAKA

- [1] A. Jupri, S. Suripto, S. Billah, T. Rozi, E. S. Prasedya, and N. Hidayah, “Pemanfaatan Komoditas Tomat (*Lycopersicum esculentum* Mill) Guna Menciptakan Nilai Tambah Dengan Melakukan Sosialisasi Pembuatan Manisan Tomat Di Desa Paok Pampang Kecamatan Sukamulia Lombok Timur,” *Jurnal Gema Ngabdi*, vol. 5, no. 2, pp. 236–240, Jul. 2023, doi: 10.29303/jgn.v5i2.381.
- [2] A. Maysela, N. Rohma, and N. Penulis:, “Diagnosa Penyakit Tanaman Tomat pada Citra Daun Menggunakan Metode Convolutional Neural Network (CNN),” JIMU, 2024.
- [3] D. R. Sihotang, D. Syafitri, D. C. Octaviana, P. A. Septianingrum, and A. Asy-Syfaityyah, “Identification of Pests and Diseases in Tomato Plants in Jeprono Village, Karangbangun District, Karanganyar Regency,” *Jurnal Biologi Tropis*, vol. 25, no. 1, pp. 381–393, Feb. 2025, doi: 10.29303/jbt.v25i1.8229.
- [4] A. R. Wang and N. H. Shabrina, “A deep learning-based mobile app system for visual identification of tomato plant disease,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 6, pp. 6992–7004, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6992-7004.
- [5] B. Kommey, E. Tamakloe, D. Opoku, T. Crispin, and J. Danquah, “Disease Detection in Tropical Tomato Leaves via Machine Learning Models,” *Jurnal ELTIKOM*, vol. 8, no. 2, pp. 179–191, Dec. 2024, doi: 10.31961/eltikom.v8i2.1340.
- [6] J. Lu, L. Tan, and H. Jiang, “Review on convolutional neural network (CNN) applied to plant leaf disease classification,” Aug. 01, 2021, *MDPI AG*. doi: 10.3390/agriculture11080707.
- [7] S. M. Sivalingam and L. D. Badabagni, “Encouraging hygiene permanence in tomato leaf and applying machine learning techniques,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 343–349, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp343-349.
- [8] Y. H. Natbais and A. B. S. Umbu, “Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih Tensorflow Lite,” *TEKNOTAN*, vol. 17, no. 2, p. 83, Aug. 2023, doi: 10.24198/jt.vol17n2.1.
- [9] F. A. Mohammed, K. K. Tune, B. G. Assefa, M. Jett, and S. Muhie, “Medical Image Classifications Using Convolutional Neural Networks: A Survey of Current Methods and Statistical Modeling of the Literature,” Mar. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make6010033.
- [10] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
- [11] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, “A Survey of CNN-Based Network Intrusion Detection,” Aug. 01, 2022, *MDPI*. doi: 10.3390/app12168162.
- [12] X. Xu *et al.*, “Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN,” *Sensors*, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22031215.
- [13] S. Sathyanarayanan, “Confusion Matrix-Based Performance Evaluation Metrics,” *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/ajbr.v27i4s.4345.
- [14] O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-56706-x.
- [15] “Convolutional neural network.”
- [16] S. Sathyanarayanan, “Confusion Matrix-Based Performance Evaluation Metrics,” *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/ajbr.v27i4s.4345.