



Perancangan Aplikasi *Machine Learning* Identifikasi Penyakit Daun Tanaman Tomat Sayur Menggunakan Metode *Convolutional Neural Network* Serta Rekomendasi Penanganan Dan Pencegahannya

Andi Muhammad Maulana^{1*}, Lela Nurlaela², Heru Winarno³

^{1*3}Program Studi Teknik Informatika, Institut Teknologi dan Bisnis Swadharma, Indonesia

Alamat : Jl. Malaka No.3, RT.7/RW.3, Roa Malaka, Kec. Tambora, Kota Jakarta Barat, DKI Jakarta

Email: andymaulana1221@gmail.com^{1*}, lela@swadharma.ac.id², heru.w@swadharma.ac.id³

Korespondensi penulis : andymaulana1221@gmail.com

Abstract. Leaf diseases can interfere with tomato productivity and crop quality, while field recognition that relies only on visual observation is vulnerable to differences in user interpretation. This research designed a web-based machine learning application that applies a Convolutional Neural Network (CNN) to distinguish ten tomato leaf conditions and present suitable treatment and prevention guidance. The development process followed CRISP-DM and used the Tomato Leaf Diseases Plant Village Augmented Data set. Each image was resized to 150 × 150 pixels, normalized, augmented for training, and allocated to training and validation subsets in an 80:20 proportion. The custom architecture contains three convolutional blocks, each combined with batch normalization and max pooling, followed by global average pooling, a dense layer, dropout, and a ten-unit softmax output. Model training was carried out for ten epochs with the Adam optimizer and a learning rate of 0.0001. The highest validation accuracy was 80.12%, whereas the classification report showed 80% overall accuracy and a weighted F1-score of 0.80. Tomato Yellow Leaf Curl Virus produced the strongest balanced performance, with an F1-score of 0.94. The selected model was deployed in Streamlit so that users can view the predicted disease, confidence value, explanation, and practical recommendations.

Keywords: Convolutional Neural Network; image classification; machine learning; Streamlit; tomato leaf disease.

Abstrak. Gangguan penyakit pada daun dapat berdampak pada mutu dan produktivitas tanaman tomat, sementara pengenalan gejala yang hanya mengandalkan pengamatan visual berisiko menghasilkan penilaian yang berbeda antar pengguna. Penelitian ini merancang aplikasi *machine learning* berbasis web dengan *Convolutional Neural Network* (CNN) untuk membedakan sepuluh kondisi daun tomat dan menyajikan rekomendasi penanganan serta pencegahannya. Proses pengembangan mengikuti CRISP-DM dan memanfaatkan *Tomato Leaf Diseases Plant Village Augmented Data*. Seluruh citra diseragamkan menjadi 150 × 150 piksel, dinormalisasi, diberi augmentasi pada data latih, lalu dialokasikan ke subset pelatihan dan validasi dengan proporsi 80:20. Arsitektur *custom* yang digunakan memuat tiga blok konvolusi yang dipadukan dengan *batch normalization* dan *max pooling*, kemudian dilanjutkan dengan *global average pooling*, *dense layer*, *dropout*, serta keluaran *softmax* berisi sepuluh neuron. Pelatihan berlangsung selama sepuluh *epoch* menggunakan *optimizer Adam* dan *learning rate* 0,0001. *Validation accuracy* tertinggi mencapai 80,12%, sedangkan *classification report* menunjukkan *accuracy* keseluruhan 80% dan *weighted F1-score* 0,80. *Tomato Yellow Leaf Curl Virus* memberikan hasil paling seimbang dengan *F1-score* 0,94. Model terpilih selanjutnya diterapkan pada *Streamlit* agar pengguna dapat memperoleh kelas prediksi, *confidence score*, penjelasan penyakit, serta rekomendasi praktis.

Kata kunci: Convolutional Neural Network; klasifikasi citra; *machine learning*; penyakit daun tomat; *Streamlit*.

1. LATAR BELAKANG

Tomat (*Solanum lycopersicum*) termasuk komoditas hortikultura yang luas dibudidayakan sekaligus dikonsumsi masyarakat. Produktivitas tanaman ini berkaitan erat dengan kondisi kesehatannya, sebab penyakit pada daun dapat mengurangi

kemampuan fotosintesis, mempercepat kerusakan jaringan, dan berdampak pada mutu hasil panen. Tanda awal serangan umumnya terlihat melalui perubahan warna, kemunculan bercak, nekrosis, deformasi, atau pola menyerupai mosaik di permukaan daun. Oleh karena itu, pengenalan gejala sejak dini diperlukan agar pengendalian dapat dilakukan sebelum gangguan meluas ke bagian tanaman lainnya.

Di lapangan, penyakit daun biasanya dikenali melalui pemeriksaan visual oleh petani maupun penyuluh. Meskipun praktis, hasil pemeriksaan tersebut dipengaruhi oleh pengalaman pengamat, keadaan pencahayaan, serta kemiripan gejala antara satu penyakit dan penyakit lainnya. Pengolahan citra digital dapat membantu menghasilkan penilaian yang lebih konsisten dengan memanfaatkan informasi warna, tekstur, bentuk bercak, dan pola kerusakan sebagai karakteristik visual (Jumadi et al., 2021; Nasution et al., 2025). Selanjutnya, pendekatan *machine learning* memungkinkan karakteristik tersebut dipelajari dari data pelatihan untuk memperkirakan kelas citra yang belum pernah diproses sebelumnya (Hariyanti et al., 2024; Septhya et al., 2023).

Salah satu pendekatan *deep learning* yang banyak digunakan untuk pengenalan citra ialah *Convolutional Neural Network* (CNN). Melalui susunan operasi konvolusi dan *pooling*, jaringan dapat membangun representasi fitur secara bertahap, mulai dari pola sederhana sampai karakteristik visual yang lebih kompleks. Penerapan CNN pada beragam tugas klasifikasi citra telah menunjukkan kinerja yang baik (Berliani & Basuki, 2025; Verdy & Hartati, 2024). Dalam konteks tanaman tomat, Anwari (2024), Natbais dan Umbu (2023), serta Putri et al. (2025) memperlihatkan bahwa penyakit dapat dikenali melalui citra daun. Namun, capaian model tetap bergantung pada keragaman data, distribusi kelas, kondisi pencahayaan, dan kedekatan ciri visual antarkelas.

Sebagian penelitian terdahulu lebih banyak menempatkan klasifikasi atau deteksi sebagai keluaran utama. Soenarto dan Basuki (2026), misalnya, mengevaluasi algoritma CNN dengan arsitektur YOLOv8 pada daun tanaman hortikultura tropis. Pada penelitian lain, model klasifikasi citra dikembangkan tanpa menyediakan informasi lanjutan bagi pengguna setelah label diprediksi. Kondisi tersebut membuka kebutuhan terhadap aplikasi yang tidak berhenti pada penentuan kelas, tetapi turut menampilkan *confidence score*, uraian penyakit, serta tindakan yang dapat dipertimbangkan. Penyajian melalui aplikasi *web* juga membuat hasil analisis lebih mudah diakses secara interaktif menggunakan peramban (Hermawan et al., 2024; Jessica et al., 2025).

Atas dasar kebutuhan tersebut, penelitian ini diarahkan untuk merancang sekaligus mengevaluasi aplikasi identifikasi penyakit daun tomat berbasis CNN pada sepuluh kelas, yang terdiri atas sembilan jenis penyakit dan satu kondisi sehat. Kontribusi terapan penelitian diwujudkan melalui penggabungan model klasifikasi citra dengan informasi penanganan dan pencegahan pada aplikasi *Streamlit*. Sistem diposisikan sebagai sarana identifikasi awal yang cepat dan informatif, bukan sebagai pengganti pemeriksaan ahli tanaman, khususnya ketika gejala sulit dibedakan atau kondisi lapangan tidak serupa dengan data pelatihan.

2. KAJIAN TEORETIS

2.1 Convolutional Neural Network

CNN mengolah citra dengan mempertahankan hubungan spasial antar piksel sehingga sesuai digunakan pada data berbentuk *grid*. *Kernel* pada lapisan konvolusi digeser di seluruh citra untuk menangkap karakteristik lokal, seperti tepi, tekstur, dan susunan warna. *Feature map* yang terbentuk kemudian diperkecil melalui *pooling* agar kebutuhan komputasi berkurang sekaligus mempertahankan informasi yang dominan. Setelah beberapa tahap ekstraksi, representasi fitur tingkat tinggi diteruskan ke bagian klasifikasi. *Batch normalization* membantu menjaga kestabilan aktivasi selama pelatihan, sedangkan *dropout* membatasi ketergantungan model pada neuron tertentu sehingga kecenderungan *overfitting* dapat dikurangi (Berliani & Basuki, 2025; Ramadhan et al., 2025).

Untuk klasifikasi dengan banyak kelas, *softmax* mengubah keluaran jaringan menjadi distribusi probabilitas bagi seluruh kategori. Proses pembelajaran dilakukan dengan menekan nilai *categorical crossentropy*. Kinerja model selanjutnya ditinjau melalui *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* menunjukkan proporsi prediksi yang benar secara keseluruhan, sedangkan *precision* dan *recall* memperlihatkan ketepatan prediksi positif serta kemampuan model menemukan sampel aktual pada masing-masing kelas. *F1-score* merangkum keseimbangan kedua metrik tersebut dan menjadi penting ketika jumlah citra antarkelas tidak sama.

2.2 Penyakit Daun Tomat dan Pengolahan Citra

Kelas yang dianalisis meliputi *Bacterial Spot*, *Early Blight*, *Late Blight*, *Leaf Mold*, *Septoria Leaf Spot*, *Spider Mites* *Two-Spotted Spider Mite*, *Target Spot*, *Tomato Mosaic Virus*, *Tomato Yellow Leaf Curl Virus*, dan *healthy*. Walaupun setiap kelas mempunyai karakteristik visual tertentu, beberapa di antaranya sama-sama menampilkan bercak atau

perubahan warna yang serupa. Oleh sebab itu, model perlu mempelajari gabungan informasi tekstur, bentuk bercak, warna, dan penyebaran kerusakan, bukan bergantung hanya pada satu karakteristik (Anwari, 2024; Natbais & Umbu, 2023).

Tahap praproses bertujuan menyamakan bentuk masukan sekaligus mendukung kestabilan proses belajar. *Resize* menetapkan ukuran citra yang seragam, sedangkan normalisasi memetakan intensitas piksel ke rentang 0–1. Augmentasi menambahkan variasi terkontrol pada orientasi, ukuran, posisi, dan pencahayaan citra. Dengan cara tersebut, keragaman sampel pelatihan dapat diperluas tanpa harus melakukan pelabelan data baru secara manual.

2.3 Aplikasi Web dan Pemodelan Sistem

Streamlit adalah *framework* berbasis *Python* yang dapat digunakan untuk menyajikan model *data science* dan *machine learning* dalam bentuk aplikasi web interaktif tanpa membangun komponen *frontend* secara terpisah dan kompleks (Hermawan et al., 2024; Jessica et al., 2025). Dalam penelitian ini, *framework* tersebut berfungsi sebagai antarmuka untuk mengunggah citra, menjalankan proses inferensi, dan menampilkan prediksi berikut informasi pendukungnya.

Sebelum implementasi, fungsi sistem dan hubungan antaraktivitas perlu dipetakan agar aplikasi yang dibangun sesuai dengan kebutuhan pengguna. *Use case diagram* digunakan untuk menunjukkan layanan yang dapat dijalankan oleh aktor, sedangkan *activity diagram* menggambarkan perpindahan aktivitas dari satu proses ke proses berikutnya (Surya & Haromain, 2023; Wayahdi & Ruziq, 2023). *Flowchart* melengkapi pemodelan tersebut dengan memperlihatkan alur pengolahan dari masukan sampai keluaran (Noija et al., 2023). Perancangan yang terstruktur membantu menjaga kesesuaian antara kebutuhan, implementasi fitur, dan tahapan pengujian aplikasi (Doni & Alfresi, 2025; Pangestu & Putra, 2025).

3. METODE PENELITIAN

Penelitian ini memakai pendekatan kuantitatif eksperimental dan menyusun prosesnya berdasarkan enam fase CRISP-DM, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Sumber data yang digunakan adalah *Tomato Leaf Diseases Plant Village Augmented Data* yang dipublikasikan oleh Basak (2025). *Dataset* telah dikelompokkan dalam sepuluh direktori kelas sehingga *ImageDataGenerator* dapat membaca label berdasarkan nama *folder*.

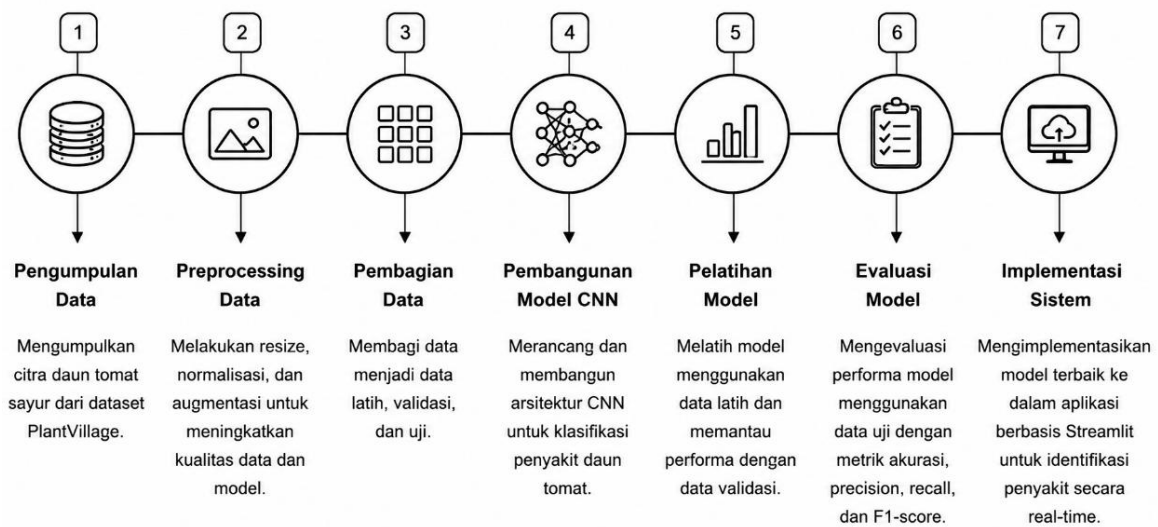
Seluruh data kemudian dibagi dengan proporsi 80% untuk pelatihan dan 20% untuk validasi. *Classification report* menunjukkan bahwa subset validasi yang dievaluasi berjumlah 49.394 citra.

Pada tahap persiapan data, ukuran citra diubah menjadi 150×150 piksel dan nilai intensitasnya diskalakan dengan membagi setiap piksel menggunakan 255. Variasi citra hanya ditambahkan pada subset pelatihan melalui *rotation_range=30*, *zoom_range=0,2*, *width_shift_range=0,2*, *height_shift_range=0,2*, dan *horizontal_flip=True*. Subset validasi dipertahankan tanpa augmentasi agar hasil evaluasi mewakili citra yang tidak mengalami modifikasi selama proses pengujian. Konfigurasi generator menggunakan *batch size 32* dan *class mode categorical*.

Model dikembangkan dengan *TensorFlow* dan *Keras* menggunakan *Sequential Model*. Bagian ekstraksi fitur terdiri atas tiga rangkaian *Conv2D*, *Batch Normalization*, dan *MaxPooling2D* yang masing-masing menggunakan 32, 64, dan 128 *filter*. Keluaran blok konvolusi diringkas melalui *Global Average Pooling*, kemudian diteruskan ke *Dense 256* neuron beraktivasi *ReLU*, *Dropout 0,5*, dan *Dense 10* neuron dengan aktivasi *softmax*. Susunan tersebut menghasilkan 129.738 parameter, yang mencakup 129.290 *trainable parameters* dan 448 *non-trainable parameters*.

Konfigurasi kompilasi menggunakan *optimizer Adam* pada *learning rate 0,0001*, fungsi kerugian *categorical_crossentropy*, dan *accuracy* sebagai metrik utama. Model dilatih selama sepuluh *epoch*. Selama proses tersebut, *ModelCheckpoint* memantau *val_accuracy* dan menyimpan bobot dengan hasil terbaik dalam format *HDF5*. *EarlyStopping* mengakhiri pelatihan apabila performa tidak lagi membaik, sedangkan *ReduceLROnPlateau* mengurangi *learning rate* ketika *validation loss* tidak menunjukkan kemajuan. Penilaian akhir dilakukan melalui kurva *accuracy* dan *loss*, *classification report*, serta *normalized confusion matrix*.

Bobot dengan performa validasi terbaik selanjutnya dipasang pada aplikasi *Streamlit*. Citra masukan dibatasi pada format JPG, JPEG, atau PNG. Setelah *file* diterima, sistem menerapkan *resize* dan normalisasi yang sama seperti pada tahap pelatihan, memuat model, kemudian memilih kelas dengan probabilitas terbesar. Antarmuka menampilkan nama kelas, *confidence score*, penjelasan kondisi daun, rekomendasi penanganan, dan langkah pencegahan.



Gambar 1. Tahapan pengembangan model dan implementasi sistem

Sumber: Hasil penelitian, 2026

Tabel 1. Arsitektur model CNN

No	Layer	Parameter	Fungsi
1	Input	$150 \times 150 \times 3$	Memasukkan citra dengan tiga kanal warna
2	Conv2D	32 filter, kernel 3×3 , ReLU	Menangkap karakteristik visual tingkat awal
3	Batch Normalization	-	Menjaga kestabilan nilai aktivasi
4	MaxPooling2D	Pool 2×2	Mengurangi ukuran peta fitur
5	Conv2D	64 filter, kernel 3×3 , ReLU	Mempelajari fitur visual lanjutan
6	Batch Normalization	-	Menstabilkan proses aktivasi
7	MaxPooling2D	Pool 2×2	Meringkas dimensi peta fitur
8	Conv2D	128 filter, kernel 3×3 , ReLU	Mengenali pola visual yang lebih kompleks
9	Batch Normalization	-	Mendukung kestabilan proses pelatihan
10	MaxPooling2D	Pool 2×2	Memperkecil <i>feature map</i>
11	Global Average Pooling	-	Merangkum keluaran menjadi vektor fitur
12	Dense	256 neuron, ReLU	Memetakan fitur menuju kelas
13	Dropout	0,5	Membatasi kecenderungan <i>overfitting</i>
14	Output	10 neuron, Softmax	Menghasilkan probabilitas untuk sepuluh kelas

Sumber: Hasil penelitian, 2026

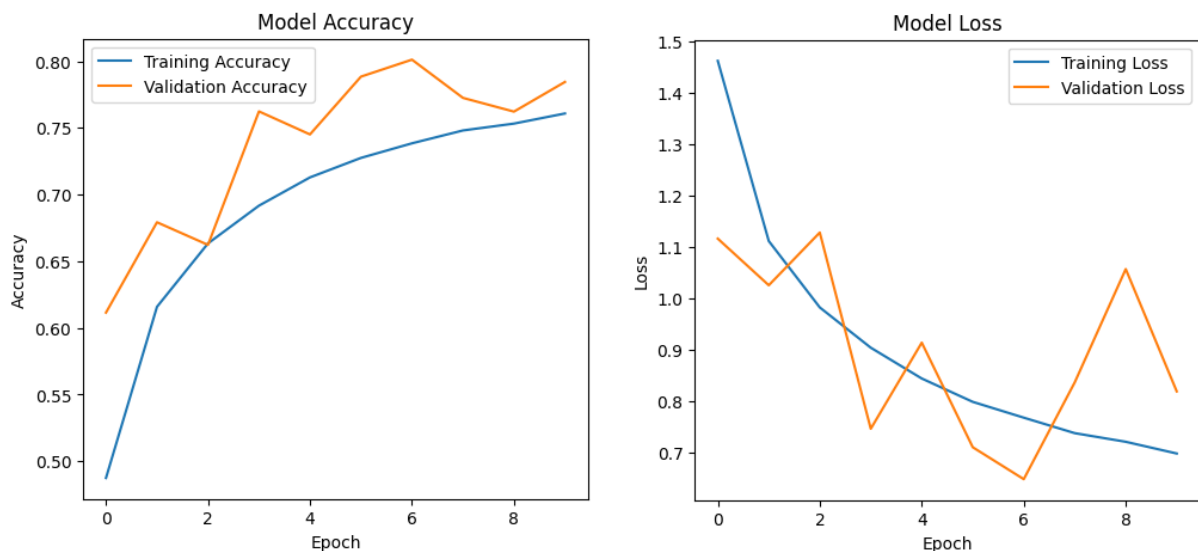
4. HASIL DAN PEMBAHASAN

4.1 Hasil Pelatihan Model

Selama sepuluh *epoch*, kemampuan model dalam mengenali karakteristik citra mengalami peningkatan. Pada *epoch* pertama, *training accuracy* tercatat 48,74%, sedangkan *validation accuracy* mencapai 61,15%. Nilai validasi tertinggi diperoleh pada *epoch* ketujuh, yaitu 80,12%, dengan *validation loss* 0,6477. Tiga *epoch* setelahnya menghasilkan *validation accuracy* 77,25%, 76,22%, dan 78,45%. Karena *EarlyStopping*

mengembalikan bobot terbaik, model yang digunakan pada tahap implementasi berasal dari *epoch* ketujuh, bukan dari *epoch* terakhir.

Kurva *accuracy* memperlihatkan kecenderungan meningkat pada data pelatihan maupun validasi. Pada beberapa *epoch*, *validation accuracy* berada di atas *training accuracy*. Pola tersebut dapat terjadi karena augmentasi dan *dropout* hanya diterapkan ketika model dilatih, sehingga sampel pelatihan menjadi lebih sulit dibandingkan sampel validasi. *Training loss* turun secara berurutan dari 1,4628 menjadi 0,6977, sementara *validation loss* mengalami fluktuasi dan naik tajam pada *epoch* kedelapan serta kesembilan sebelum kembali menurun pada *epoch* kesepuluh. Temuan ini menunjukkan bahwa model terus mempelajari pola data, tetapi kestabilan generalisasi mulai melemah setelah titik terbaik. Oleh karena itu, penggunaan *callback* mencegah pemilihan bobot terakhir yang performanya tidak optimal.



Gambar 2. Perubahan *accuracy* dan *loss* selama pelatihan
Sumber: Hasil penelitian, 2026

4.2 Classification Report

Evaluasi terhadap 49.394 citra pada *subset* validasi menghasilkan *accuracy* 0,80. Untuk *weighted average*, nilai *precision*, *recall*, dan *F1-score* berturut-turut adalah 0,82; 0,80; dan 0,80. Adapun *macro average* menghasilkan *precision* 0,76, *recall* 0,78, serta *F1-score* 0,75. Nilai *weighted average* yang lebih tinggi daripada *macro average* menunjukkan bahwa distribusi sampel tidak seimbang, sehingga kelas berukuran besar memberi pengaruh lebih dominan terhadap hasil agregat.

Tabel 2. Hasil *classification report* model CNN

Kelas	Precision	Recall	F1-score	Support
<i>Bacterial Spot</i>	0,91	0,72	0,81	5.786
<i>Early Blight</i>	0,57	0,73	0,64	2.720
<i>Late Blight</i>	0,69	0,81	0,74	5.191
<i>Leaf Mold</i>	0,88	0,69	0,77	2.587
<i>Septoria Leaf Spot</i>	0,69	0,87	0,77	4.817
<i>Spider Mites</i>	0,72	0,69	0,70	4.559
<i>Target Spot</i>	0,80	0,56	0,66	3.818
<i>Yellow Leaf Curl Virus</i>	0,96	0,92	0,94	14.572
<i>Tomato Mosaic Virus</i>	0,47	0,95	0,63	1.016
<i>Healthy</i>	0,89	0,83	0,86	4.328
Macro average	0,76	0,78	0,75	49.394
Weighted average	0,82	0,80	0,80	49.394

Sumber: Hasil penelitian, 2026

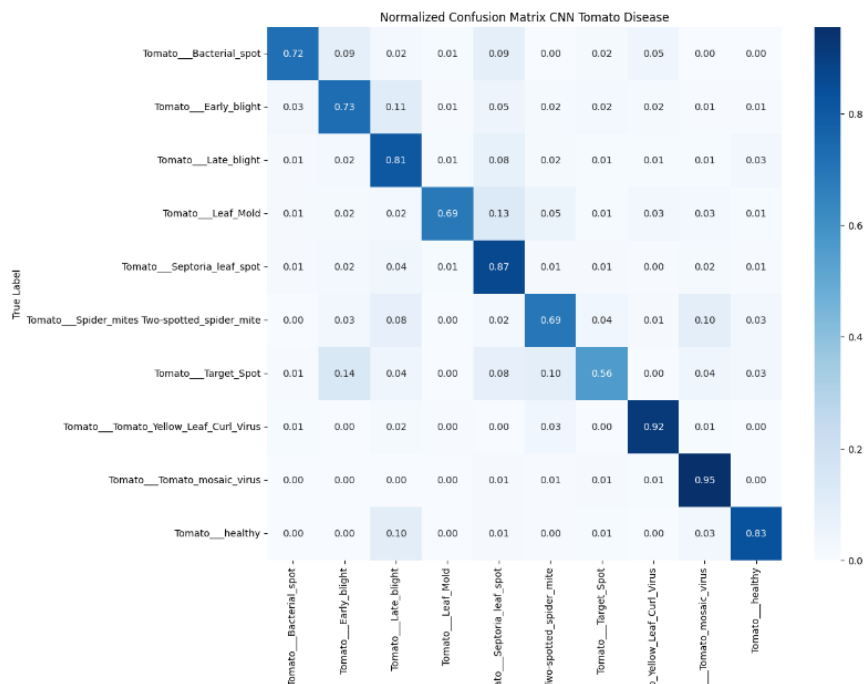
Tomato Yellow Leaf Curl Virus menjadi kelas dengan hasil paling konsisten karena mencapai *precision* 0,96, *recall* 0,92, dan *F1-score* 0,94. Performa kelas *healthy* juga relatif kuat dengan *F1-score* 0,86. Pada *Bacterial Spot*, *precision* mencapai 0,91 tetapi *recall* hanya 0,72. Artinya, sebagian besar prediksi yang diberikan sebagai *Bacterial Spot* memang tepat, meskipun masih terdapat citra aktual pada kelas tersebut yang tidak berhasil dikenali.

Karakteristik yang berbeda terlihat pada *Tomato Mosaic Virus*. *Recall* sebesar 0,95 menunjukkan bahwa hampir seluruh citra aktual berhasil ditemukan, tetapi *precision* 0,47 menandakan cukup banyak citra dari kelas lain ikut diberi label tersebut. *Target Spot* mencatat *recall* paling rendah, yaitu 0,56, dengan *F1-score* 0,66, sedangkan *Early Blight* memperoleh *F1-score* 0,64. Kesalahan klasifikasi ini berkaitan dengan kedekatan pola bercak, perubahan warna, dan kerusakan jaringan pada beberapa penyakit. Hasil tersebut sejalan dengan penelitian terdahulu yang menegaskan bahwa mutu data dan pemilihan arsitektur berpengaruh terhadap kemampuan CNN dalam memisahkan kelas dengan ciri visual serupa (Anwari, 2024; Soenarto & Basuki, 2026).

4.3 Normalized Confusion Matrix

Normalized confusion matrix menyajikan proporsi keluaran model untuk setiap kelas aktual. Angka pada diagonal utama menggambarkan *recall* masing-masing kelas. *Tomato Mosaic Virus* memperoleh nilai diagonal 0,95 dan *Yellow Leaf Curl Virus* 0,92. Nilai berikutnya ditunjukkan oleh *Septoria Leaf Spot* sebesar 0,87, *healthy* 0,83, serta *Late Blight* 0,81. *Target Spot* berada pada nilai terendah, yaitu 0,56; sebagian kesalahannya dialihkan ke *Early Blight* sebesar 0,14 dan *Spider Mites* sebesar 0,10. Pada

Leaf Mold, sebanyak 0,13 citra diprediksi sebagai *Septoria Leaf Spot* karena kemiripan karakteristik visual keduanya.



Gambar 3. Normalized confusion matrix sepuluh kelas daun tomat

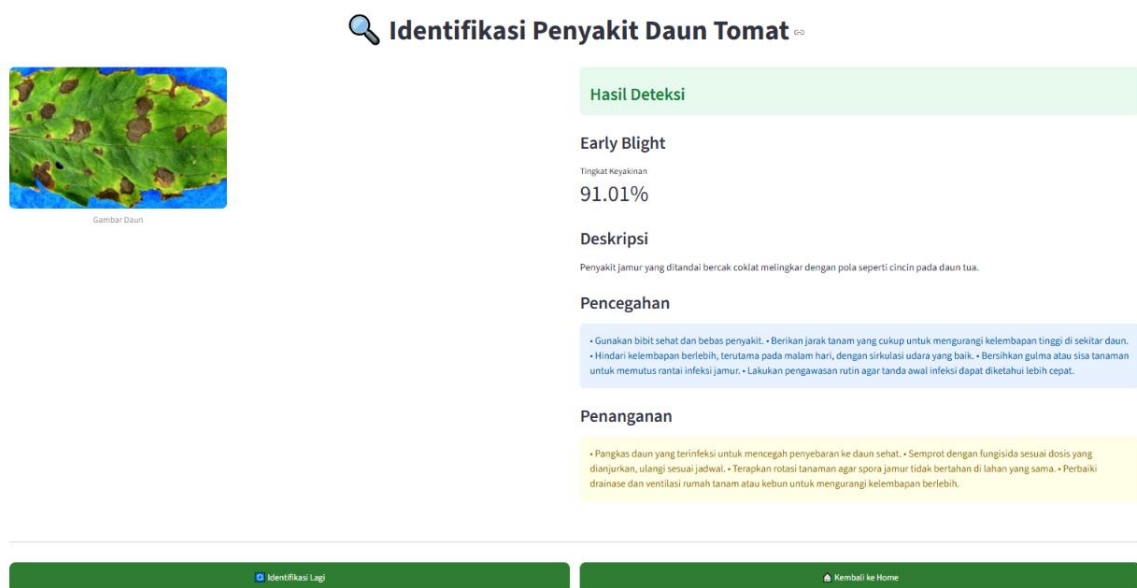
Sumber: Hasil penelitian, 2026

Matriks tersebut menunjukkan bahwa *recall* tinggi belum tentu diikuti *precision* yang sama tinggi. Pada *Tomato Mosaic Virus*, model mampu mengenali 95% sampel aktual, tetapi *precision* yang rendah mengindikasikan adanya prediksi berlebih dari kelas lain menuju kelas mosaik. Dengan demikian, performa klasifikasi multikelas tidak cukup dinilai hanya melalui satu ukuran. Pembacaan *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix* secara bersama-sama memberikan gambaran yang lebih utuh tentang pola kesalahan model.

4.4 Implementasi Aplikasi

Model terpilih diterapkan pada *Streamlit* dan disusun menjadi tiga bagian, yakni halaman *Home*, halaman identifikasi, dan halaman keluaran prediksi. Pada halaman identifikasi, pengguna dapat mengunggah citra sekaligus memperoleh validasi format *file*. Setelah proses prediksi dijalankan, aplikasi menyajikan citra masukan, kelas penyakit, *confidence score*, deskripsi, rekomendasi penanganan, serta tindakan pencegahan. Pengguna juga disediakan pilihan untuk melakukan identifikasi ulang atau kembali ke halaman utama.

Penerapan pada aplikasi mengubah hasil eksperimen model menjadi sarana pendukung informasi yang dapat digunakan secara langsung. *Streamlit* menghubungkan model *Python* dengan antarmuka *web* dalam satu alur pengembangan, sebagaimana dimanfaatkan pula pada berbagai aplikasi *machine learning* interaktif (Hermawan et al., 2024; Jessica et al., 2025). Rekomendasi yang ditampilkan dapat membantu pengguna mengambil respons awal, namun penerapannya tetap harus memperhatikan kondisi tanaman, tingkat keparahan serangan, dan arahan tenaga pertanian. Pengendalian yang sesuai, pemantauan berkala, serta praktik budidaya yang sehat tetap diperlukan untuk menekan penyebaran penyakit (Sulyanti et al., 2025).



Gambar 4. Tampilan hasil prediksi dan rekomendasi pada aplikasi

Sumber: Hasil penelitian, 2026

4.5 Implikasi dan Keterbatasan

Dari sisi penggunaan, aplikasi berpotensi membantu proses identifikasi awal secara lebih cepat dan terarah dibandingkan pengamatan tanpa dukungan sistem. Penyajian deskripsi dan rekomendasi juga membuat aplikasi dapat dimanfaatkan sebagai media pembelajaran bagi pengguna. Dari sisi teknis, arsitektur *custom* hanya memiliki 129.738 parameter sehingga relatif ringan apabila dibandingkan dengan banyak model *transfer learning* dan berpeluang dikembangkan untuk perangkat dengan kemampuan komputasi terbatas.

Batasan penelitian terutama berkaitan dengan karakteristik data dan pengujian aplikasi. *Dataset* yang digunakan bersifat terkurasi serta mencakup citra hasil augmentasi, sehingga variasi latar belakang, intensitas cahaya, sudut pemotretan, dan keadaan daun di

lapangan belum tentu sepenuhnya terwakili. Pengujian terkontrol menggunakan citra yang diambil langsung oleh petani belum dilakukan, demikian pula pengukuran *usability* pada kelompok pengguna. Ketidakseimbangan kelas juga terlihat dari perbedaan antara *macro average* dan *weighted average*. Penelitian berikutnya dapat memperluas citra lapangan, menerapkan strategi penyeimbangan data, menilai kalibrasi *confidence score*, membandingkan model *transfer learning*, dan melibatkan pengguna dalam evaluasi. Analisis SWOT juga dapat dimanfaatkan untuk memetakan peluang penerapan dan risiko adopsi sistem (Mashuri & Nurjannah, 2020; Suriono et al., 2021).

5. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan aplikasi berbasis CNN yang dapat mengidentifikasi sepuluh kondisi daun tomat. Arsitektur model dibentuk oleh tiga blok konvolusi, *global average pooling*, *dense layer*, dan *dropout* dengan keseluruhan 129.738 parameter. Performa terbaik pada proses pelatihan diperoleh di *epoch* ketujuh dengan *validation accuracy* 80,12%. Evaluasi terhadap 49.394 citra validasi menghasilkan *accuracy* 80% serta *weighted F1-score* 0,80. *Yellow Leaf Curl Virus* menunjukkan keseimbangan performa tertinggi melalui *F1-score* 0,94. Sebaliknya, *Target Spot*, *Early Blight*, dan *Tomato Mosaic Virus* masih memperlihatkan kesalahan yang berkaitan dengan kemiripan karakteristik dan ketidakseimbangan prediksi. Model telah diimplementasikan pada *Streamlit* untuk menyajikan kelas penyakit, *confidence score*, deskripsi, serta rekomendasi penanganan dan pencegahan. Aplikasi dapat dimanfaatkan sebagai pendukung identifikasi awal, sementara keputusan pengendalian tetap perlu mempertimbangkan situasi lapangan dan penilaian tenaga ahli. Pengembangan berikutnya disarankan memperbanyak citra lapangan, memperbaiki keseimbangan kelas, membandingkan arsitektur alternatif, menguji kalibrasi probabilitas, melakukan pengujian *usability*, dan memvalidasi rekomendasi secara agronomis.

UCAPAN TERIMA KASIH

Artikel ini disusun sebagai bagian dari penelitian skripsi pada Program Studi Teknik Informatika, Fakultas Teknologi, Institut Teknologi dan Bisnis Swadharma. Penulis menyampaikan apresiasi kepada dosen pembimbing serta seluruh pihak yang telah memberikan dukungan selama penyusunan penelitian, pengembangan model, dan penerapan aplikasi.

DAFTAR REFERENSI

- Anwari, A. C. (2024). Klasifikasi penyakit tanaman tomat melalui citra daun menggunakan metode Convolutional Neural Network. *JIMU: Jurnal Ilmiah Multidisipliner*, 2(3), 639–647. <https://doi.org/10.70294/jimu.v2i03.421>
- Basak, S. K. (2025). Tomato Leaf Diseases Plant Village Augmented Data [Data set]. Kaggle.
- Berliani, G. N., & Basuki, S. (2025). Klasifikasi rumah adat di Indonesia berbasis citra menggunakan Convolutional Neural Network. *Repositor*, 7(3), 407–420.
- Doni, D. S., & Alfresi, A. I. (2025). Perancangan sistem informasi e-learning menggunakan metode Rapid Application Development (RAD) di IAIQ Al-Ittifaqiah. *Jurnal Riset Sistem Informasi dan Teknologi Informasi*, 7(1), 628–635. <https://doi.org/10.52005/jursistekni.v7i1.462>
- Hariyanti, N. T., Rahmawati, T., & Wirapraja, A. (2024). Perancangan aplikasi mobile menggunakan machine learning untuk menentukan klasifikasi kategori berita. *Teknika*, 13(3), 487–492. <https://doi.org/10.34148/teknika.v13i3.1093>
- Hermawan, E., Panjaitan, S., & Ripanti, E. (2024). Sistem prediksi banjir rob Kota Pontianak berbasis machine learning menggunakan framework Streamlit. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 10(3), 351–361.
- Jessica, F., Thomas, W. C., Zepani, C. L., Fadillah, M. E., & Annisa, R. (2025). Aplikasi perkiraan efisiensi bahan bakar mobil dengan machine learning dan Streamlit. *JUKOMIKA (Jurnal Ilmu Komputer dan Informatika)*, 8(2), 86–92. <https://doi.org/10.54650/jukomika.v8i2.614>
- Jumadi, J., Yupianti, Y., & Sartika, D. (2021). Pengolahan citra digital untuk identifikasi objek menggunakan metode hierarchical agglomerative clustering. *JST (Jurnal Sains dan Teknologi)*, 10(2), 148–156. <https://doi.org/10.23887/jstundiksha.v10i2.33636>
- Mashuri, M., & Nurjannah, D. (2020). Analisis SWOT sebagai strategi meningkatkan daya saing. *JPS (Jurnal Perbankan Syariah)*, 1(1), 97–112. <https://doi.org/10.46367/jps.v1i1.205>
- Nasution, M. U., Harahap, L. S., & Syakbani, F. (2025). Tinjauan metode pengolahan citra digital. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(3), 1020–1028.
- Natbais, Y. H., & Umbu, A. B. S. (2023). Aplikasi deteksi penyakit pada daun tomat berbasis Android menggunakan model terlatih TensorFlow Lite. *Teknotan*, 17(2), 83–90. <https://doi.org/10.24198/jt.vol17n2.1>

- Noija, H. C., Wemaf, P. A., Nurdianty, O. A., Sohilait, W., Haumahu, S., Yusuf, H., Tomagola, K., La Ana, W. R., & Salaiswa, T. S. (2023). Perancangan sistem informasi akuntansi untuk analisis siklus pendapatan menggunakan DFD dan flowchart. *Jurnal Bisnis dan Manajemen*, 1(2), 577–592. <https://doi.org/10.61930/jurbisman.v1i2.188>
- Pangestu, I. B. P. H., & Putra, I. N. T. A. (2025). Perancangan aplikasi edukasi tarian adat tradisional Bali berbasis Android. *Jurnal Riset Sistem Informasi dan Teknologi Informasi*, 7(1), 617–627. <https://doi.org/10.52005/jursistekni.v7i1.464>
- Putri, R., Ismi, L., & Nudin, S. R. (2025). Deteksi penyakit pada tanaman tomat menggunakan model Inception V3 berbasis mobile. *Journal of Informatics and Computer Science*, 7(2), 520–524.
- Ramadhan, C., Atina, V., & Permatasari, H. (2025). Analisis perbandingan model CNN dan IndoBERT dalam sentimen berita politik Indonesia. *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis*, 110–118. <https://doi.org/10.47701/v1r9ka69>
- Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Rahmaddeni, Irawan, Y., & Hayami, R. (2023). Implementasi algoritma Decision Tree dan Support Vector Machine untuk klasifikasi penyakit kanker paru. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(1), 15–19. <https://doi.org/10.57152/malcom.v3i1.591>
- Soenarto, U. A. P., & Basuki, S. (2026). Analisa performa algoritma Convolutional Neural Network dengan arsitektur YOLOv8 untuk deteksi penyakit daun tanaman hortikultura tropis. *Repositor*, 8(1), 91–104.
- Sulyanti, E., Resti, Z., Rahma, H., Darnetty, Khairul, U., Nurhaliza, & Oktaviani, S. (2025). Endophytic bacteria consortia as biocontrol against brown spot disease and growth promotion in rice. *Jurnal Proteksi Tanaman*, 9(1), 1–13. <https://doi.org/10.25077/jpt.9.1.1-13.2025>
- Suriono, Z., Islam, U., & Sumatera, N. (2021). Analisis SWOT dalam perencanaan strategis. *Alacrity: Journal of Education*, 1(1), 94–103.
- Surya, A. A., & Haromain, I. (2023). Rancang bangun website lelang mobil menggunakan CodeIgniter pada PT ABC. *Jurnal Teknologi Terpadu*, 9(2), 133–142.
- Verdy, & Hartati, E. (2024). Klasifikasi penyakit mata menggunakan Convolutional Neural Network model ResNet-50. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, 1(3), 199–206. <https://doi.org/10.59407/jrsit.v1i3.529>

Wayahdi, M. R., & Ruziq, F. (2023). Pemodelan sistem penerimaan anggota baru dengan Unified Modeling Language. *Jurnal Minfo Polgan*, 12(1), 1514–1521. <https://doi.org/10.33395/jmp.v12i1.12870>