



Implementasi Model MaxViT untuk Deteksi Penyakit Daun Cabai Berbasis Mobile

Nur Puspita Amalia*, Salamun Rohman Nudin

Fakultas Vokasi, Program Studi Manajemen Informatika, Universitas Negeri Surabaya, Surabaya, Indonesia

Email: ^{1*}nurpuspita.22093@mhs.unesa.ac.id, ²salamunrohman@unesa.ac.id

Email Penulis Korespondensi: nurpuspita.22093@mhs.unesa.ac.id

Abstrak—Cabai merupakan komoditas hortikultura penting di Indonesia, namun produktivitasnya sering menurun akibat serangan penyakit daun seperti *Cercospora leaf spot*, *Powdery Mildew*, *Mites and Thrips*, dan *Nutritional Deficiency*. Proses identifikasi yang masih dilakukan secara manual membutuhkan waktu yang lebih lama serta berisiko menghasilkan diagnosis yang kurang akurat. Penelitian ini bertujuan mengembangkan sistem deteksi penyakit daun cabai berbasis *mobile* menggunakan model Multi-Axis Vision Transformer (MaxViT). *Dataset* yang digunakan berjumlah 10.987 citra daun cabai yang terdiri atas lima kelas dan dibagi menjadi data *training*, *validation*, dan *testing* dengan proporsi 70:15:15. Pelatihan model dilakukan menggunakan empat skenario *optimizer*, yaitu baseline model standar, SGD, Adam, AdamW. Hasil penelitian menunjukkan bahwa *optimizer* Adam memberikan performa terbaik dengan *accuracy* 99,45%, *precision* 99,44%, *recall* 99,45%, dan *F1-Score* 99,45%. Model terbaik dikonversi ke TensorFlow Lite dengan ukuran file sebesar 32,0 MB dan diimplementasikan ke aplikasi *mobile*. Aplikasi mampu mendeteksi penyakit melalui kamera maupun galeri serta menampilkan hasil deteksi dan saran penanganan. Hasil pengujian menunjukkan bahwa sistem dapat digunakan dengan baik pada berbagai kondisi penggunaan. Sistem ini diharapkan membantu pengguna mengenali penyakit daun cabai secara cepat, praktis, dan akurat.

Kata Kunci: *Deep Learning*; Aplikasi Mobile; Deteksi Penyakit Daun; Penyakit Daun Cabai; MaxViT

Abstract—Chili is an important horticultural commodity in Indonesia, but its productivity often decreases due to leaf diseases such as *Cercospora leaf spot*, *Powdery Mildew*, *Mites and Thrips*, and *Nutritional deficiency*. Manual disease identification requires more time and may produce less accurate diagnoses. This study aims to develop a mobile-based chili leaf disease detection system using the Multi-Axis Vision Transformer (MaxViT) model. The dataset consisted of 10,987 chili leaf images divided into five classes and split into training, validation, and testing data with a ratio of 70:15:15. Model training was carried out using four *optimizer* scenarios, namely a standard baseline model, SGD, Adam, and AdamW. The result showed that the Adam *optimizer* achieved the best performance with 99,45% accuracy, 99,44% precision, 99,45% recall, and 99,45% F1-Score. The best model was converted into TensorFlow Lite format with a file size of 32.0 MB and deployed in a mobile application. The application can detect diseases through camera capture or gallery images and provide prediction results along with disease descriptions and treatment recommendations. Testing results indicate that the system performs well under various usage conditions. This system is expected to help users identify chili leaf diseases quickly, practically, and accurately.

Keywords: Deep Learning; Mobile Application; Plant Disease Detection; Chili Leaf Disease; MaxViT

1. PENDAHULUAN

Cabai merupakan komoditas hortikultura di Indonesia yang memiliki peran strategis dalam sektor pertanian dan perekonomian Nasional. Masyarakat menggunakan cabai sebagai bahan pangan harian, sedangkan pasar menjadikan cabai sebagai komoditas yang berpengaruh terhadap laju inflasi karena perubahan harga yang cukup tinggi. Berdasarkan data Badan Pusat Statistik, produksi cabai besar di Indonesia pada tahun 2024 mencapai 1,47 juta ton atau turun sebesar 5,48% dibandingkan tahun sebelumnya, sedangkan produksi cabai rawit mencabai 1,57 juta ton atau naik sebesar 4,11% (Irjayanti dkk., 2025). Kondisi ini menunjukkan bahwa kestabilan produksi cabai masih menjadi permasalahan yang dipengaruhi oleh berbagai faktor, seperti perubahan iklim, serangan hama, dan penyakit tanaman. Dampak dari kondisi tersebut, tidak hanya dirasakan pada ketersediaan dan harga cabai di pasar, tetapi juga berpengaruh terhadap pendapatan petani dan kestabilan ekonomi nasional.

Tanaman cabai membutuhkan pengelolaan budidaya yang tepat agar mampu menghasilkan produksi maksimal. Gangguan selama masa pertumbuhan dapat menurunkan mutu buah dan jumlah panen. Serangan penyakit pada daun cabai berpotensi menurunkan hasil panen hingga 50% apabila penanganan tidak dilakukan dengan baik. Kondisi tersebut menunjukkan bahwa kesehatan daun menjadi faktor penting dalam keberhasilan budidaya cabai (Khasanah & Fachrie, 2024). Penyakit daun menjadi salah satu masalah utama karena daun berperan penting dalam proses fotosintesis. Kerusakan daun akan menurunkan kemampuan tanaman dalam menghasilkan energi sehingga pertumbuhan menjadi terhambat. Penelitian ini mengidentifikasi lima kondisi daun cabai, yaitu *healthy*, *cercospora leaf spot*, *mites and thrips*, *nutritional deficiency*, dan *powdery mildew*. Kelas *healthy* menunjukkan kondisi daun yang sehat, *cercospora leaf spot* ditandai dengan munculnya bercak pada daun, *mites and thrips* menyebabkan kerusakan akibat serangan hama, *nutritional deficiency* terjadi karena kekurangan unsur hara, sedangkan *powdery mildew* ditandai dengan lapisan putih pada permukaan daun yang dapat menghambat pertumbuhan tanaman (Rahman dkk., 2025).

Petani dan penyuluh pertanian masih banyak melakukan identifikasi penyakit daun cabai secara manual melalui pengamatan langsung. Cara tersebut memerlukan waktu yang cukup lama dan sangat bergantung pada pengalaman pengamat dalam mengenali gejala yang muncul. Perbedaan penilaian antar individu juga dapat menimbulkan hasil diagnosis yang tidak konsisten. Keterlambatan identifikasi menyebabkan penanganan penyakit menjadi kurang tepat



serta meningkatkan risiko kerusakan dan gagal panen. Oleh karena itu, sistem identifikasi otomatis diperlukan untuk membantu proses diagnosis penyakit daun cabai secara lebih cepat dan akurat di lapangan (Putri dkk., 2025).

Perkembangan *Artificial Intelligence* (AI) mendorong hadirnya berbagai inovasi pada sektor pertanian modern. Teknologi AI mampu mempelajari pola data dan menghasilkan keputusan berbasis data secara otomatis. Salah satu cabang AI yang banyak digunakan pada analisis citra adalah *computer vision*. Teknologi *computer vision* membantu sistem mengenali objek, pola, dan kondisi tertentu dari gambar digital secara otomatis. Teknologi ini telah diterapkan pada bidang keamanan, kesehatan, transportasi, dan pertanian cerdas (Upadhyay dkk., 2025). Model *deep learning* menjadi pendekatan yang banyak digunakan dalam pengembangan *computer vision* karena mampu mengekstraksi fitur citra otomatis. Metode yang paling sering digunakan dalam *computer vision* adalah *Convolutional Neural Network* (CNN) yang dikenal efektif dalam mengekstraksi fitur visual seperti warna, tekstur, dan bentuk secara otomatis (Putra & Prihartono, 2025). Namun, CNN memiliki keterbatasan dalam menangkap hubungan antar bagian citra yang berjauhan, sehingga kurang optimal dalam memahami konteks global suatu gambar.

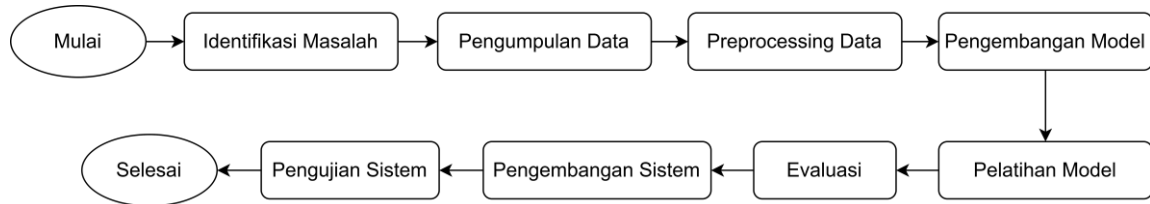
Arsitektur *Transformer* kemudian dikembangkan untuk mengatasi keterbatasan tersebut melalui mekanisme *self-attention*. *Vision Transformer* (ViT) memproses citra menjadi beberapa patch dan mempelajari hubungan antarbagian secara menyeluruh sehingga mampu memahami konteks global gambar dengan lebih baik (Neha & Bansal, 2024). Penelitian yang dilakukan oleh Ramadhana dkk. (2025) yang menerapkan *Vision Transformer* pada klasifikasi penyakit daun cabai mampu meningkatkan akurasi validasi sebesar 95,97% dan akurasi pengujian sebesar 93,55%. Meskipun memiliki performa yang baik, model ViT memerlukan sumber daya komputasi yang lebih besar dan waktu pelatihan yang lebih lama dibandingkan CNN, sehingga kurang efisien untuk diterapkan pada perangkat dengan keterbatasan sumber daya, seperti perangkat *mobile*. Model *Multi-Axis Vision Transformer* (MaxViT) dikembangkan sebagai arsitektur yang menggabungkan keunggulan CNN dan *Transformer*. MaxViT menggunakan komponen MBConv untuk menangkap fitur local serta *block attention* dan *grid attention* untuk memahami hubungan global citra. Pendekatan tersebut membuat MaxViT lebih efisien dan tetap memiliki performa tinggi (Tu dkk., 2022). Penelitian yang dilakukan oleh Rahman dkk. (2025) menunjukkan bahwa MaxViT memperoleh akurasi sebesar 90,86% pada daun cabai, 95,75% pada daun bawang. Penelitian lain yang dilakukan oleh Shahriar dkk., (2025) menunjukkan MaxViT memperoleh akurasi yang tinggi pada identifikasi penyakit kedelai. Hal ini menunjukkan bahwa MaxViT layak digunakan pada klasifikasi citra tanaman. Namun, penelitian tersebut belum berfokus secara khusus pada satu jenis tanaman cabai dan belum diimplementasikan dalam bentuk aplikasi *mobile* yang dapat digunakan langsung oleh petani. Berdasarkan tinjauan tersebut, terdapat kesenjangan penelitian yang perlu diperhatikan, antara lain keterbatasan variasi dataset yang berpotensi menyebabkan *overfitting*, kurangnya kajian terkait pengaruh algoritma optimisasi terhadap performa model berbasis *Transformer*, dan minimnya implementasi sistem yang digunakan secara langsung oleh petani melalui perangkat *mobile*. Hal ini menunjukkan bahwa meskipun akurasi model telah meningkat, aspek efisiensi, kemampuan generalisasi, serta penerapan praktis di lapangan masih memerlukan pengembangan lebih lanjut.

Penelitian ini mengintegrasikan model MaxViT berbasis transfer learning, analisis perbandingan algoritma optimisasi, serta pengembangan aplikasi *mobile* berbasis flutter yang dapat digunakan secara *offline*. *Transfer learning* memanfaatkan bobot pra-latih dari dataset besar seperti Imagenet untuk kemudian disesuaikan kembali melalui proses *fine-tuning* pada dataset daun cabai yang lebih spesifik sehingga pelatihan menjadi lebih efisien dan performa model dapat meningkat (Bichri dkk., 2023). Kinerja model *deep learning* tidak hanya dipengaruhi oleh arsitektur, tetapi juga dipengaruhi oleh *optimizer* pada proses pelatihan. *Optimizer* berfungsi memperbarui bobot model agar nilai loss menurun menuju kondisi optimal (Reyad dkk., 2023). Oleh karena itu, penelitian ini menerapkan *optimizer* SGD, Adam, dan AdamW untuk menganalisis pengaruh masing-masing metode terhadap kinerja model. Adam dikenal mampu menyesuaikan *learning rate* secara adaptif, sedangkan AdamW menambahkan *weight decay* yang membantu meningkatkan kemampuan generalisasi model (Mahajaya dkk., 2024).

Kontribusi utama dalam penelitian ini terletak pada penerapan arsitektur MaxViT menggunakan *transfer learning*, analisis perbandingan algoritma optimisasi, dan penerapan model ke dalam aplikasi *mobile* yang dapat digunakan secara langsung oleh pengguna untuk melakukan identifikasi penyakit secara cepat dan lebih akurat tanpa koneksi internet. Pendekatan ini tidak hanya berfokus pada peningkatan akurasi, tetapi juga memperhatikan efisiensi komputasi dan kemudahan penggunaan di lapangan, sehingga menghasilkan solusi yang lebih praktis dibandingkan penelitian sebelumnya. Dengan demikian, hasil penelitian ini diharapkan dapat membantu petani dalam melakukan deteksi penyakit secara cepat dan akurat, mengurangi ketergantungan terhadap tenaga ahli, serta meningkatkan efisiensi dan produktivitas sektor pertanian di Indonesia.

2. METODOLOGI PENELITIAN

Metodologi penelitian ini disusun secara terstruktur untuk menjelaskan tahapan pelaksanaan penelitian dalam mengembangkan sistem deteksi penyakit daun cabai menggunakan model MaxViT berbasis *mobile*. Setiap tahap saling berhubungan, dimulai dari identifikasi masalah hingga implementasi sistem, sehingga dapat menghasilkan model klasifikasi yang akurat dan mudah digunakan. Secara umum, tahapan penelitian meliputi identifikasi masalah, pengumpulan data, *preprocessing* data, pengembangan model, pelatihan model, evaluasi, pengembangan sistem, dan pengujian sistem. Alur penelitian secara keseluruhan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tahapan penelitian yang terdapat pada Gambar 1 dilakukan dalam pengembangan sistem deteksi penyakit daun cabai berbasis mobile menggunakan model MaxViT. Penelitian diawali dengan identifikasi masalah untuk menentukan kebutuhan dan tujuan penelitian. Tahap berikutnya adalah pengumpulan data, *preprocessing* data, pengembangan model, pelatihan model, dan evaluasi model untuk memperoleh performa terbaik. Model yang memiliki kinerja terbaik kemudian diimplementasikan ke dalam aplikasi *mobile* melalui tahap pengembangan sistem. Tahap terakhir adalah pengujian sistem untuk mengetahui kemampuan aplikasi dalam mendeteksi penyakit daun cabai pada berbagai kondisi penggunaan.


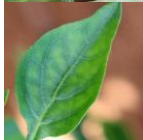

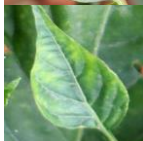
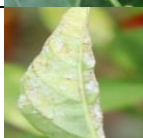
2.1 Identifikasi Masalah

Tahap identifikasi masalah merupakan langkah awal yang dilakukan untuk mengetahui dan memahami permasalahan utama dalam penelitian ini. Permasalahan yang diangkat dalam penelitian ini adalah ketidakstabilan produksi cabai yang salah satunya disebabkan oleh penyakit pada daun tanaman. Penyakit daun cabai sering kali sulit dikenali pada tahap awal karena gejalanya memiliki kemiripan, seperti perubahan warna, klorosis, dan munculnya bercak pada daun (Mursyidin & Mulyaningsih, 2024). Selama ini, proses identifikasi penyakit masih dilakukan secara manual oleh petani atau penyuluh pertanian melalui pengamatan langsung. Cara ini sangat bergantung pada pengalaman individu sehingga membutuhkan waktu cukup lama dan hasilnya belum tentu akurat. Akibatnya, penanganan penyakit sering terlambat dan dapat menurunkan hasil panen cabai (Putri dkk., 2025). Untuk mengatasi hal tersebut, diperlukan sistem yang dapat membantu mengidentifikasi penyakit daun cabai secara otomatis, cepat, dan lebih akurat. Oleh karena itu, penelitian ini menggunakan pendekatan pengolahan citra digital dengan model MaxViT untuk mendukung proses klasifikasi penyakit daun cabai secara lebih efektif dan objektif.

2.2 Pengumpulan Data

Pada tahap ini dilakukan proses pengumpulan dataset yang digunakan sebagai data pelatihan model dan data pengujian sistem. Dataset yang digunakan dalam penelitian ini diperoleh dari Mendeley Data dengan judul “Chili Dataset” yang disusun oleh (Aishwarya & Reddy, 2024). Dataset tersebut berisi citra daun cabai dengan ukuran 224 x 224 piksel. Secara keseluruhan, jumlah data yang digunakan dalam penelitian ini sebanyak 10.987 citra. Rincian distribusi jumlah data pada setiap kelas penyakit dapat dilihat pada Tabel 1.

Tabel 1. Rincian Dataset yang Digunakan

Kelas	Jumlah Data	Gambar
Cercospora	2.219	
Healthy	2.198	
Mites and Thrips	2.507	
Nutritional Deficiency	2.032	
Powdery Mildew	2.031	



Kelas	Jumlah Data	Gambar
Total	10.987	

Berdasarkan Tabel 1, data yang digunakan terdiri atas lima kelas, yaitu Cercospora, Healthy, Mites and Thrips, Nutritional deficiency, dan Powdery Mildew. Kelas Mites and Thrips memiliki jumlah data terbanyak, yaitu 2.507 citra, sedangkan kelas Powdery Mildew memiliki jumlah data paling sedikit, yaitu 2.031. Kelas Cercospora, Healthy, dan Nutritional deficiency memiliki jumlah data yang tidak jauh berbeda. Perbedaan jumlah data antar kelas relatif kecil. Jumlah data pada setiap kelas cukup merata untuk digunakan dalam proses pelatihan model.

2.3 Preprocessing Data

Tahapan *preprocessing* data dilakukan untuk mempersiapkan dataset agar sesuai dengan kebutuhan model sebelum proses pelatihan. Pada tahap ini, data diolah melalui tiga langkah, yaitu pembagian *dataset*, penyesuaian ukuran gambar, dan normalisasi nilai piksel. *Preprocessing* memiliki peran penting dalam meningkatkan kualitas data, menjaga keseragaman citra, serta membantu model mengenali pola visual dengan lebih baik sehingga performa klasifikasi dapat meningkat (Rafi dkk., 2025). *Dataset* dibagi menjadi tiga bagian sebagai data *training*, data *validation*, dan data *testing*. Data *training* digunakan untuk melatih model agar dapat mengenali pola dan karakteristik dari setiap kelas penyakit daun cabai. Data *validation* digunakan untuk memantau kinerja model selama proses pelatihan serta mencegah terjadinya *overfitting*. Sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model terhadap data baru yang belum pernah digunakan sebelumnya. Pembagian *dataset* dilakukan dengan proporsi 70% data *training*, 15% data *validation*, dan 15% data *testing*. Pembagian ini bertujuan agar distribusi data pada setiap kelas tetap seimbang serta menghasilkan model yang lebih stabil dan tidak bias (Winanto dkk., 2023). Rincian proporsi pembagian data dapat dilihat pada Tabel 2.

Tabel 2. Rincian Proporsi Pembagian *Dataset*

Data	Proporsi	Jumlah Citra
Training	70%	7.688
Validation	15%	1.649
Testing	15%	1.650
Total		10.987

Proses pembagian data pada seluruh *citra dataset* di *resize* menjadi 224×224 piksel untuk menyesuaikan ukuran input model, kemudian dilakukan normalisasi nilai piksel dari rentang 0-255 menjadi 0-1 untuk menyederhanakan data masukan sehingga model dapat belajar lebih cepat dan stabil. Normalisasi juga berperan dalam meningkatkan performa serta kemampuan generalisasi model pada proses klasifikasi citra daun (Murugavalli & Gopi, 2025). Dengan melalui seluruh tahapan *preprocessing* tersebut, *dataset* menjadi lebih terstruktur dan siap digunakan pada proses pelatihan maupun evaluasi model klasifikasi penyakit daun cabai.

2.4 Pengembangan Model

Pengembangan model pada penelitian ini dilakukan menggunakan platform Google Colab dengan memanfaatkan *framework TensorFlow* dan library *keras_cv_attention_models*. Model yang dibangun menggunakan pendekatan *Transfer learning* dengan arsitektur MaxViT_Tiny sebagai *backbone* utama. Pemilihan MaxViT_Tiny didasarkan pada kemampuannya dalam menggabungkan operasi konvolusi lokal dan mekanisme attention global, sehingga proses ekstraksi fitur citra dapat dilakukan secara lebih optimal. Arsitektur transformer seperti MaxViT dinilai mampu menangkap informasi lokal dan global secara seimbang, sehingga sesuai yang diterapkan pada klasifikasi citra duann yang memiliki pola visual kompleks (Rahman dkk., 2025).

Model menggunakan MaxViT_Tiny *pretrained ImageNet* sebagai *feature extractor* utama. *Backbone* ini berfungsi mengekstraksi fitur penting dari citra daun secara bertahap, mulai dari pola tepi, warna, tekstur, hingga bentuk kerusakan yang lebih kompleks. Pemanfaatan bobot awal dari ImageNet bertujuan agar model telah memiliki representasi fitur dasar yang baik sebelum dilatih pada dataset penelitian, sehingga proses pelatihan dapat berlangsung lebih cepat dan akurasi model meningkat. Selain itu ditambahkan custom head untuk menyesuaikan keluaran *backbone* dengan tugas klasifikasi penyakit daun cabai. Lapisan tambahan tersebut terdiri atas *Global Average Pooling*, *Batch Normalization*, *Dense layer 256* neuron dengan aktivasi *ReLU*, *Dropout 0,55*, dan *Dense output layer 5* neuron dengan aktivasi *softmax*. *Global Average Pooling* berfungsi menyederhanakan ukuran *feature map* menjadi vektor fitur, *Batch Normalization* membantu menjaga kestabilan distribusi data selama pelatihan, *Dense layer* digunakan untuk mempelajari hubungan antar fitur, sedangkan *Dropout* digunakan untuk mengurangi risiko *overfitting*. Ringkasan struktur model ditampilkan pada Tabel 3.

Tabel 3. Model Summary

Layer (Type)	Output Shape	Jumlah Parameter
Input Layer	(None, 224, 224, 3)	0
MaxViT_Tiny Backbone	(None, 7, 7, 512)	-
Global_Average_Pooling2D	(None, 512)	0



Layer (Tipe)	Output Shape	Jumlah Parameter
Batch_Normalization	(None, 512)	2,048
Dense (256, ReLU)	(None, 256)	131,328
Dropout (0,55)	(None, 256)	0
Dense Output (5, Softmax)	(None, 5)	1,285
Total Params:		30,322,125
Trainable Params:		30,273,485
Non-Trainable Params:		48,640

Proses *fine-tuning* yang diterapkan pada *backbone* MaxViT_Tiny menyebabkan sebagian besar parameter model yaitu 30,273,485 parameter bersifat *trainable* atau dapat dilatih kembali. Parameter sebanyak 48,640 bersifat *non-trainable* yang berasal dari komponen normalisasi. Parameter *trainable* tersebut terdiri dari *backbone* utama serta layer tambahan, yaitu *dense* 256 neuron sebanyak 131,328 parameter dan layer *output* sebanyak 1,285 parameter. Meskipun menggunakan varian MaxViT_Tiny, jumlah parameter model mencapai 30,273,485 parameter. Hal ini karena seluruh layer *backbone* diaktifkan untuk proses *fine-tuning* (*trainable=true*), sehingga hampir seluruh parameter *backbone* turut dilatih kembali bersama layer klasifikasi tambahan. Jumlah parameter yang dilatih ini diharapkan dapat meningkatkan kemampuan model dalam mengenali pola citra daun cabai dengan lebih baik.

2.5 Pelatihan Model

Proses pelatihan model dilakukan dengan menggunakan data *training* dan data *validation* untuk memperoleh model terbaik dari beberapa parameter yang telah ditentukan. Pada proses ini, parameter jaringan diperbarui secara bertahap untuk meminimalkan nilai *error* dan meningkatkan akurasi prediksi (Lie dkk., 2026). Pelatihan dilakukan dengan empat skenario berdasarkan jenis *optimizer* yang digunakan. Setiap skenario bertujuan untuk membandingkan performa model sehingga dapat diperoleh *optimizer* dengan hasil paling optimal. Skenario pelatihan model dapat dilihat pada Tabel 4.

Tabel 4. Skenario Pelatihan Model

Skenario	Epoch	Batch Size	Learning Rate	Optimasi
1	50	16	1e-3	Baseline
2	50	16	1e-5	SGD
3	50	16	1e-5	Adam
4	50	16	1e-5	AdamW

Berdasarkan Tabel 4, skenario pertama digunakan sebagai baseline dengan learning rate 1e-3, sedangkan skenario lainnya menggunakan *optimizer* SGD, Adam, AdamW dengan learning 1e-5. Seluruh skenario dilatih menggunakan 50 epoch, batch size 16, dan input citra berukuran 224×224 piksel. Perbedaan utama dari setiap skenario terletak pada *optimizer* yang digunakan. *Optimizer* SGD memakai momentum sebesar 0.9 untuk membantu mempercepat proses pembelajaran dan membuat perubahan bobot lebih stabil. *Optimizer* Adam dapat menyesuaikan learning rate secara otomatis pada setiap parameter sehingga proses pelatihan menjadi lebih cepat dan efisien. Sementara, *optimizer* AdamW menggunakan *weight decay* sebesar 1e-4 untuk mengurangi *overfitting*. Selama proses pelatihan, model mempelajari pola citra daun cabai dengan menurunkan nilai *error* menggunakan fungsi *categorical crossentropy*. Nilai *accuracy* digunakan untuk melihat tingkat ketepatan prediksi model pada setiap epoch. Untuk mencegah *overfitting* dan menghemat waktu pelatihan, digunakan *Earlystopping* yang akan menghentikan proses training jika nilai *validation loss* tidak membaik, dan *ModelCheckpoint* untuk menyimpan model dengan hasil terbaik selama pelatihan.

2.6 Evaluasi

Tahap evaluasi model dilakukan untuk menilai kinerja model yang telah dilatih dalam mengklasifikasikan citra penyakit daun cabai. Evaluasi ini bertujuan untuk mengetahui sejauh mana model mampu mengenali setiap kelas penyakit secara akurat dan cepat. Pada penelitian ini, proses evaluasi dilakukan menggunakan data uji melalui tiga pendekatan utama, yaitu *accuracy* dan *loss*, *confusion matrix*, dan *classification report*. Nilai *accuracy* digunakan untuk menunjukkan tingkat ketepatan model dalam melakukan prediksi, sedangkan *loss* menunjukkan besarnya kesalahan prediksi yang dihasilkan model. Semakin tinggi *accuracy* dan semakin rendah *loss*, maka performa model semakin baik. Penggunaan metrik akurasi juga banyak digunakan dalam penelitian klasifikasi citra untuk menilai tingkat ketepatan model dalam melakukan prediksi (Ash'shobir dkk., 2025).

Evaluasi dilakukan menggunakan *Confusion Matrix* yaitu metode yang digunakan untuk mengevaluasi seberapa baik model klasifikasi dalam memprediksi data. Matriks ini membandingkan hasil prediksi model dengan label sebenarnya pada data uji untuk mengetahui seberapa akurat model dalam mengenali setiap kelas. Dengan menggunakan *Confusion Matrix*, dapat diketahui bagian mana model sudah bekerja dengan baik dan bagian mana yang masih salah dalam melakukan klasifikasi (Sathyanarayanan, 2024). Matriks ini terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *Classification report* digunakan untuk menampilkan *accuracy*, nilai *precision*, *recall*, dan *F1-Score* pada setiap kelas. *Accuracy* menunjukkan tingkat ketepatan model dalam mengklasifikasikan data dengan benar. *Precision* yaitu mengukur seberapa tepat hasil prediksi positif yang

diberikan model dibandingkan data sebenarnya. *Recall* yaitu menggambarkan kemampuan model dalam mengenali data positif secara benar. Sedangkan *F1-Score*, yaitu menunjukkan keseimbangan antara nilai *precision* dan *recall* yang memberikan gambaran menyeluruh tentang kinerja model. Perhitungan metrik evaluasi dapat dilihat pada Persamaan (1), (2), (3), dan (4) (Murinto dkk., 2024).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FN} \tag{2}$$

$$Recall = \frac{TP}{TP+FP} \tag{3}$$

$$F1-Score = 2 \times \frac{recall \times precision}{recall + precision} \tag{4}$$

2.7 Pengembangan Sistem

Tahap pengembangan dilakukan setelah proses pelatihan dan evaluasi model. Pada tahap ini, model dengan performa terbaik diimplementasikan ke dalam sistem agar dapat dimanfaatkan secara langsung oleh pengguna. Pada penelitian ini, model yang dikembangkan menggunakan python diterapkan pada aplikasi *mobile* dengan bahasa pemrograman Dart melalui *framework* Flutter. Perbedaan lingkungan pengembangan tersebut diatas dengan memanfaatkan TensorFlow Lite, sehingga model dapat dijalankan secara efisien pada perangkat *mobile*. Model hasil pelatihan yang masih berformat .h5 selanjutnya dikonversi ke format .tflite agar ukuran file menjadi lebih ringan. Proses konversi model ke TensorFlow Lite juga banyak digunakan untuk meningkatkan efisiensi memori serta mempercepat waktu respon aplikasi *mobile* (Pamungkas dkk., 2026). Setelah itu, model diintegrasikan ke dalam aplikasi menggunakan Visual Studio Code. Aplikasi dirancang agar pengguna dapat mengambil gambar maupun mengunggah citra daun cabai, kemudian sistem akan menampilkan hasil identifikasi penyakit beserta saran penanganannya.

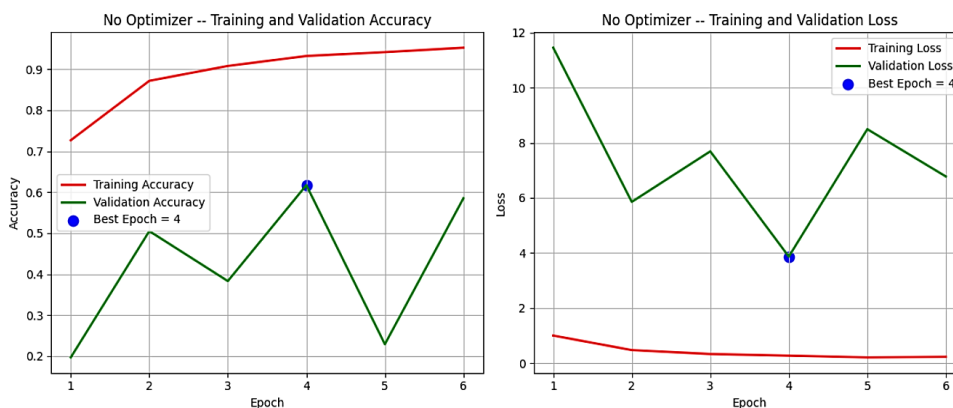
2.8 Pengujian Sistem

Tahap pengujian sistem dilaksanakan setelah proses pengembangan aplikasi selesai. Pada penelitian ini, pengujian dilakukan dengan melibatkan pengguna secara langsung dalam kondisi nyata. Pengguna diminta untuk mencoba aplikasi untuk melakukan identifikasi penyakit daun cabai melalui kamera atau galeri perangkat. Pengujian ini dilakukan dengan lima skenario penggunaan, seperti jarak pengambilan gambar, pencahayaan, sudut kamera, latar belakang, dan jumlah daun pada citra. Melalui tahap ini, diharapkan dapat mengetahui kemampuan sistem dalam mendeteksi penyakit daun cabai pada kondisi nyata.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini disajikan berdasarkan tahapan metode yang telah dijelaskan sebelumnya. Hasil yang ditampilkan meliputi performa model dalam mengklasifikasi penyakit daun cabai serta hasil penerapan model aplikasi *mobile*. Evaluasi dilakukan menggunakan metrik *accuracy*, *loss*, *confusion matrix*, *precision*, *recall*, dan *F1-Score*. Pembahasan ini berfokus pada analisis performa model, perbandingan *optimizer* yang digunakan, serta hasil implementasi sistem dalam mendeteksi penyakit daun cabai.

3.1 Hasil Pelatihan Model Skenario 1

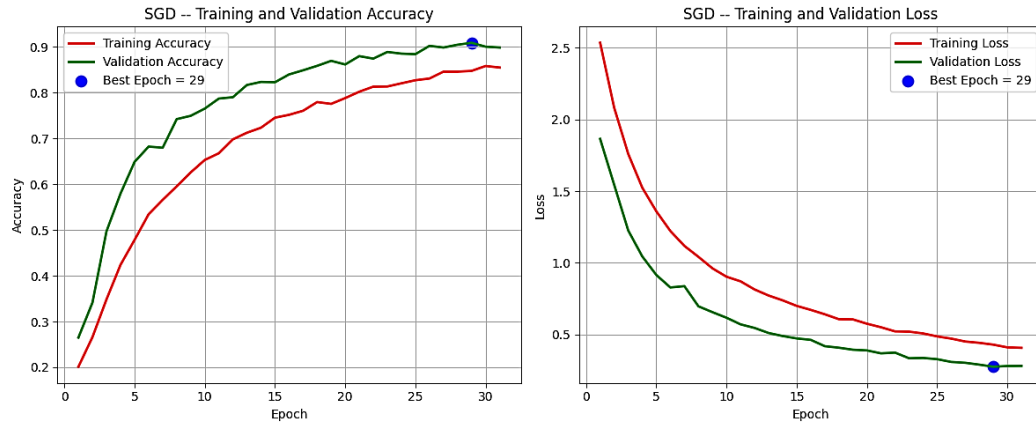


Gambar 2. Grafik *Loss* dan *Accuracy* Skenario 1

Pada Gambar 2 menunjukkan bahwa skenario baseline mampu meningkatkan akurasi data latih pada setiap *epoch*. Pada *epoch* pertama, akurasi mencapai sekitar 72% dengan nilai *loss* sebesar 0,99. Selanjutnya akurasi terus naik hingga melebihi 90% dan mencapai 95% pada *epoch* keenam, sedangkan nilai *loss* menurun menjadi sekitar 0,22.

Namun, pada data *validation* nilai akurasi dan *loss* masih mengalami perubahan yang cukup besar. Nilai akurasi validasi tertinggi pada skenario ini diperoleh pada *epoch* keempat sebesar 61,73% dengan nilai validasi *loss* sebesar 3,86. Setelah itu, performa validasi kembali menurun sehingga model belum stabil saat diuji data baru. Proses pelatihan berhenti pada *epoch* keenam karena *callback EarlyStopping*.

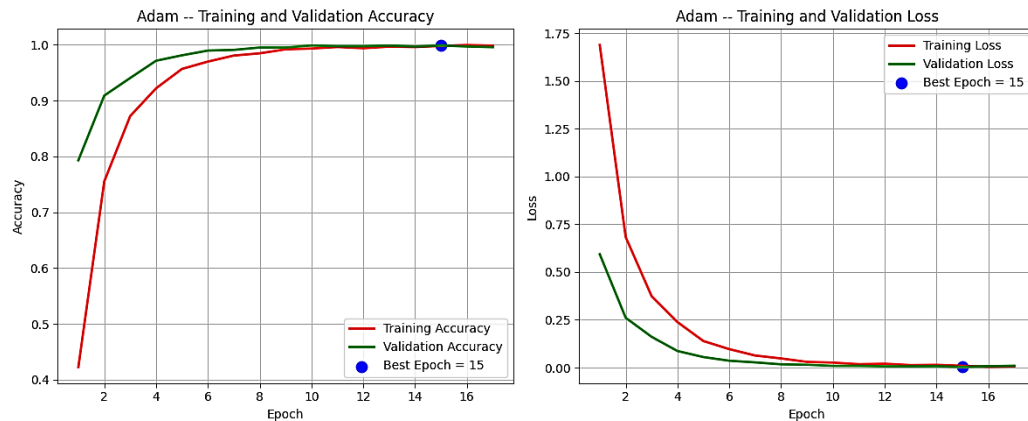
3.2 Hasil Pelatihan Model Skenario 2



Gambar 3. Grafik *Loss* dan *Accuracy* Skenario 2

Pada grafik skenario 2 menunjukkan bahwa proses pelatihan model menggunakan *optimizer* SGD. Akurasi training meningkat secara bertahap dari 20,12% pada *epoch* pertama menjadi 85,50% pada *epoch* ke-31, sedangkan nilai *loss training* menurun dari 2,53 menjadi 0,4. Pada data *validation*, akurasi meningkat dari 26,50% hingga mencapai nilai tertinggi sebesar 90,90% pada *epoch* ke-29. Nilai *validation loss* terendah diperoleh pada *epoch* ke-29 sebesar 0,2776. Setelah *epoch* ke-29, performa validasi tidak mengalami peningkatan yang sehingga proses pelatihan dihentikan pada *epoch* ke-31 oleh *callback Earlystopping*. Grafik *Accuracy* dan *Loss* bisa dilihat pada Gambar 3.

3.3 Hasil Pelatihan Model Skenario 3

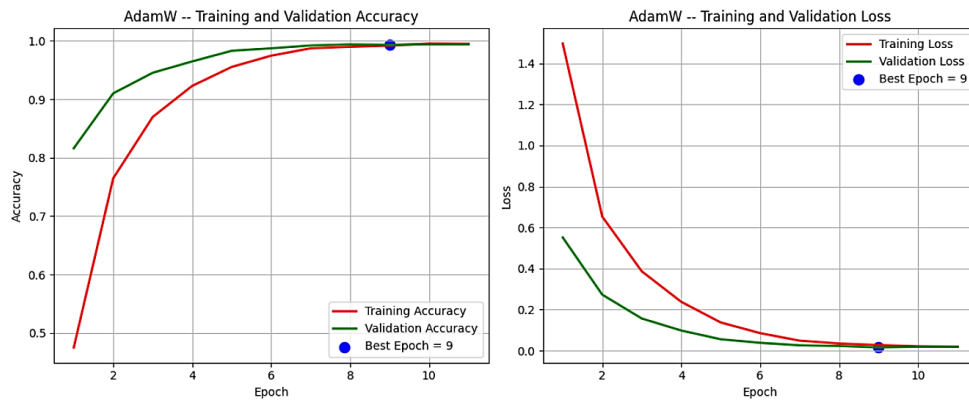


Gambar 4. Grafik *Loss* dan *Accuracy* Skenario 3

Pada grafik skenario 3 ini *optimizer* Adam mampu meningkatkan akurasi secara baik padan data *training* dan data *validation*. Akurasi *training* naik dari 42% pada *epoch* pertama menjadi 99% pada *epoch* ke-17, sedangkan nilai *loss* turun dari 1,68 menjadi sekitar 0,006. Pada data *validation*, akurasi meningkat menjadi 79,32% hingga mencapai 99,88%. Nilai *validation loss* terendah diperoleh pada *epoch* ke-15 sebesar 0,0032, sehingga *epoch* tersebut dipilih menjadi hasil pelatihan terbaik pada skenario ini yang bisa dilihat pada Gambar 4. Proses pelatihan berhenti pada *epoch* ke-17 oleh *callback Earlystopping* karena tidak terdapat peningkatan.

3.4 Hasil Pelatihan Model Skenario 4

Pada grafik skenario 4 ini, *optimizer* AdamW memberikan performa yang baik. Akurasi training meningkat dari sekitar 47% pada *epoch* pertama menjadi 99% pada *epoch* ke-11. Sedangkan nilai *loss* turun dari 1,49 menjadi 0,018. Pada data *validation*, akurasi meningkat dari 81,63% hingga mencapai 99,39%. Nilai *validation* terendah diperoleh pada *epoch* ke-9 sebesar 0,0148, sehingga dipilih sebagai hasil pelatihan terbaik pada skenario ini. Proses pelatihan berhenti pada *epoch* ke-11 yang bisa dilihat pada Gambar 5.



Gambar 5. Grafik *Loss* dan *Accuracy* Skenario 4

3.5 Hasil Akurasi Pada Semua Skenario

Penelitian ini melakukan evaluasi nilai akurasi pada data *training*, *validation*, dan *testing* untuk setiap skenario untuk melihat perbandingan kinerja model. Hasil perbandingan tersebut disajikan pada Tabel 5.

Tabel 5. Perbandingan Akurasi Setiap Skenario

Skenario	Optimizer	Hasil		
		Training (%)	Validation (%)	Testing (%)
1	Baseline	93,24	61,73	62,55
2	SGD	84,77	90,90	90,73
3	Adam	99,80	99,88	99,45
4	AdamW	99,19	99,33	98,97

Hasil akurasi setiap skenario menunjukkan bahwa penggunaan *optimizer* Adam pada skenario 3 menghasilkan performa terbaik dibandingkan *optimizer* lainnya yang ditunjukkan oleh nilai akurasi yang tinggi dan konsisten pada data *training*, *validation*, dan *testing* dengan akurasi pada data *testing* mencapai 99,45%. Sementara itu, skenario 4 yang menggunakan *optimizer* AdamW juga memberikan hasil yang baik dengan akurasi data *testing* sebesar 98,97%. Sedangkan skenario 2 dengan *optimizer* SGD menghasilkan performa cukup baik dan stabil dengan akurasi testing sebesar 90,73%. Sementara itu, skenario 1 menunjukkan performa paling rendah sehingga hasilnya belum optimal.

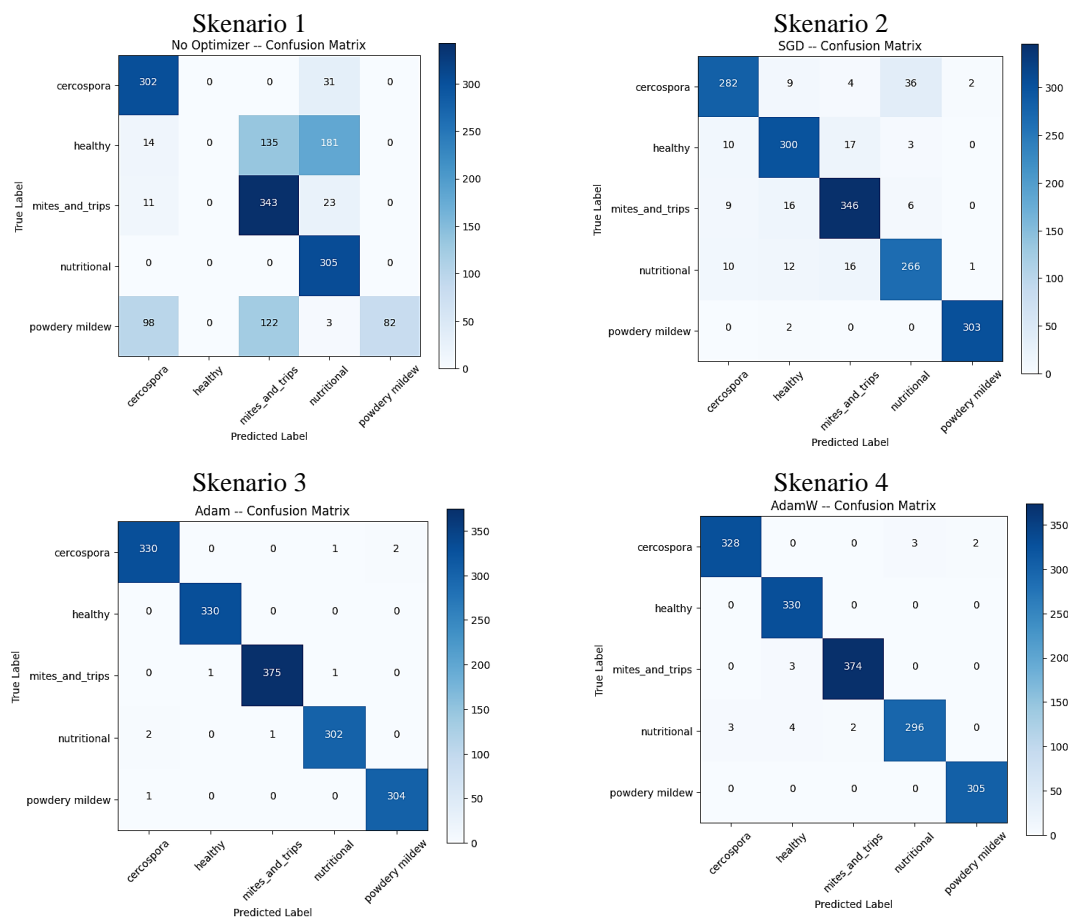
3.6 Hasil Evaluasi

Penelitian ini menggunakan empat metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-Score* untuk mengetahui performa model secara lebih detail. Hasil perbandingan metrik evaluasi untuk setiap skenario disajikan pada Tabel 6.

Tabel 6. Perbandingan Metrik Evaluasi Setiap Skenario

Skenario	Kelas	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Skenario 1	Cercospora	71,06	90,69	79,68	62,55
	Healthy	00,00	00,00	00,00	
	Mites and Thrips	57,17	90,98	70,21	
	Nutritional Deficiency	56,17	100	71,93	
	Powdery Mildew	100	26,89	42,38	
Skenario 2	Cercospora	90,68	84,68	87,58	90,73
	Healthy	88,50	90,91	89,69	
	Mites and Thrips	90,34	91,78	91,05	
	Nutritional Deficiency	85,53	87,21	86,36	
	Powdery Mildew	99,02	99,34	99,18	
Skenario 3	Cercospora	99,10	99,10	99,10	99,45
	Healthy	99,70	100	99,85	
	Mites and Thrips	99,73	99,47	99,60	
	Nutritional Deficiency	99,34	99,02	99,18	
	Powdery Mildew	99,35	99,67	99,51	
Skenario 4	Cercospora	99,09	98,50	98,80	98,97
	Healthy	97,92	100	98,95	
	Mites and Thrips	99,47	99,20	99,34	
	Nutritional Deficiency	99,00	97,05	98,01	
	Powdery Mildew	99,35	100	99,67	

Skenario pengujian menghasilkan nilai *accuracy*, *precision*, *recall*, dan *F1-Score* yang berbeda-beda. Pada skenario 1 baseline, model memperoleh akurasi sebesar 62,55%, sehingga kinerjanya masih tergolong rendah dan beberapa kelas belum dapat dikenali secara optimal, terutama kelas *healthy* yang memiliki nilai *precision*, *recall*, dan *F1-Score* sebesar 0%. Pada skenario 2 dengan *optimizer* SGD, performa model meningkat dengan akurasi mencapai 90,73%. Sebagian besar kelas memperoleh nilai evaluasi diatas 85%. Pada skenario 3 menggunakan *optimizer* Adam, model memperoleh hasil evaluasi tertinggi dengan akurasi sebesar 99,45%, semua kelas memiliki nilai *precision*, *recall*, dan *F1-Score* diatas 99%, sehingga model dapat mengenali setiap kelas secara sangat baik dan konsisten. Pada skenario 4 dengan *optimizer* AdamW, model juga menunjukkan performa yang baik dengan akurasi sebesar 98,97%. Seluruh kelas memperoleh nilai evaluasi diatas 97%, meskipun masih sedikit lebih rendah dibandingkan skenario 3. Berdasarkan hasil tersebut, penggunaan *optimizer* memberikan pengaruh penting terhadap performa model. *Optimizer* Adam pada skenario 3 menghasilkan kinerja paling optimal dibanding skenario lainnya. Penelitian ini juga menganalisis performa model menggunakan confusion matrix untuk melihat distribusi hasil prediksi terhadap kelas sebenarnya. Confusion matrix dari masing-masing skenario dapat dilihat pada Gambar 6.



Gambar 6. Confusion Matrix Pada Semua Skenario

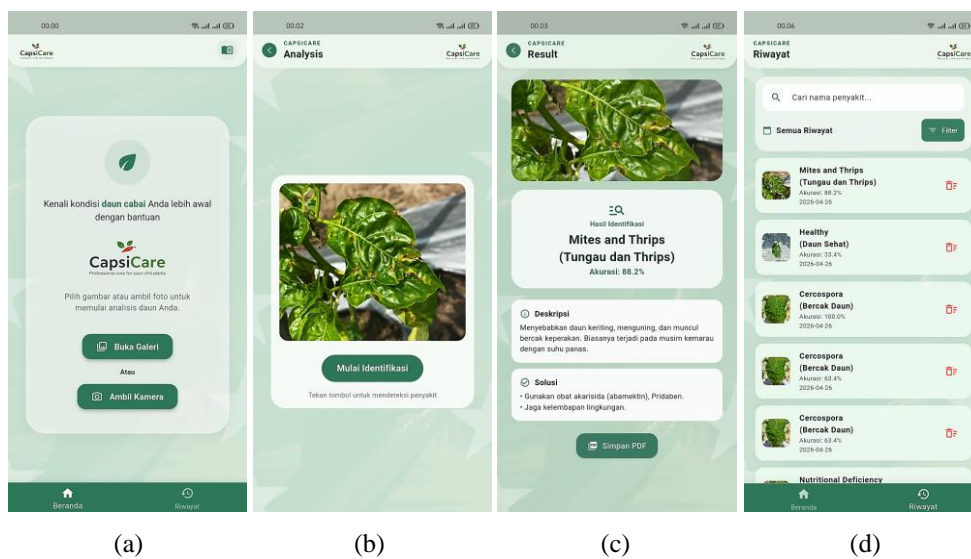
Hasil confusion matrix menunjukkan bahwa performa model sudah cukup baik, namun masih terdapat kesalahan prediksi pada sejumlah kelas. Penggunaan *optimizer* Adam dan AdamW menghasilkan kemampuan klasifikasi yang lebih baik pada setiap kelas penyakit daun cabai dengan tingkat kesalahan yang jauh lebih rendah dibandingkan baseline maupun SGD. Pada skenario baseline, kesalahan prediksi masih cukup tinggi, terutama pada kelas *healthy* yang sering diprediksi sebagai *nutritional* sebanyak 181 data dan *mites_and_thrips* sebanyak 135 data. Selain itu, pada kelas *powdery mildew* juga ditemukan banyak kesalahan prediksi yaitu sebagai *mites_and_thrips* sebanyak 122 data dan *cercospora* sebanyak 98 data. Skenario 2 menggunakan *optimizer* SGD menunjukkan performa yang lebih baik dibandingkan model tanpa *optimizer*, meskipun masih terdapat kesalahan klasifikasi, terutama pada kelas *mites_and_thrips* yang diprediksi sebagai *healthy* sebanyak 18 data. *Optimizer* Adam memberikan hasil paling optimal dengan jumlah kesalahan yang sedikit. Kesalahan yang masih terjadi antara lain 1 data *mites_and_thrips* diprediksi sebagai *healthy*, 1 data diprediksi sebagai *nutritional*, serta beberapa kesalahan kecil lainnya. *Optimizer* AdamW pada skenario 4 juga menunjukkan performa yang baik dan hampir mendekati Adam. Pemilihan hyperparameter terutama pada setiap jenis *optimizer* memberikan pengaruh yang cukup besar terhadap peningkatan akurasi model dibandingkan skenario tanpa *optimizer*, sehingga model lebih mampu membedakan setiap kelas penyakit daun cabai dengan baik.

Hasil pengujian menunjukkan bahwa model yang dikembangkan pada penelitian ini memiliki performa lebih baik dibandingkan penelitiannya sebelumnya. Penelitian ini memperoleh akurasi tertinggi sebesar 99,45% pada

skenario *optimizer* Adam. Peningkatan performa tersebut dipengaruhi oleh penerapan *transfer learning* pada model MaxViT, pemilihan *optimizer* yang sesuai, serta proses pelatihan dengan pembagian data yang seimbang. Selain itu, kombinasi kemampuan ekstraksi fitur lokal dan pemahaman konteks global pada arsitektur MaxViT membantu model mengenali pola penyakit daun cabai secara lebih akurat. Hasil ini menegaskan bahwa pendekatan yang digunakan mampu meningkatkan kinerja klasifikasi dibandingkan penelitian sebelumnya.

3.7 Hasil Pengembangan sistem

Hasil pengembangan sistem pada penelitian ini berupa aplikasi *mobile* yaitu CapsiCare yang dikembangkan menggunakan *framework* Flutter serta mengintegrasikan model klasifikasi penyakit daun cabai dalam format TensorFlow Lite (Tflite). Aplikasi ini dirancang pengguna melakukan deteksi penyakit daun cabai secara cepat dan praktis melalui perangkat *mobile*. Dengan integrasi model TFLite, proses deteksi dapat dilakukan secara real-time tanpa memerlukan koneksi internet. Aplikasi ini memiliki beberapa halaman utama yang saling terintegrasi dan dirancang dengan antarmuka yang mudah digunakan. Adapun bagian utama pada aplikasi ini yang dilihat pada Gambar 7.



Gambar 7. Tampilan Aplikasi CapsiCare (a) Halaman utama, (b) identifikasi penyakit, (c) hasil deteksi, (d) riwayat

Gambar 7 menunjukkan tampilan utama aplikasi CapsiCare yang terdiri atas halaman utama, halaman identifikasi penyakit, halaman hasil deteksi, dan halaman riwayat. Halaman utama berfungsi sebagai menu awal bagi pengguna untuk memulai proses deteksi penyakit daun cabai. Halaman ini menyediakan dua opsi utama, yaitu membuka galeri untuk memilih gambar dari penyimpanan perangkat dan menggunakan kamera untuk mengambil foto daun cabai secara langsung. Halaman identifikasi penyakit menampilkan citra daun cabai yang telah dipilih atau diambil sebelum proses klasifikasi dilakukan. Pengguna dapat menekan tombol mulai identifikasi untuk menjalankan proses deteksi. Halaman hasil deteksi menampilkan informasi berupa nama penyakit yang terdeteksi, nilai akurasi prediksi, deskripsi gejala, serta saran penanganan yang sesuai. Halaman ini juga menyediakan fitur simpan PDF untuk menyimpan hasil deteksi dalam bentuk laporan. Halaman riwayat menampilkan daftar hasil identifikasi yang telah dilakukan sebelumnya lengkap dengan gambar daun, nama penyakit, nilai akurasi, dan tanggal deteksi. Halaman ini juga dilengkapi fitur pencarian dan filter berdasarkan tanggal untuk memudahkan pengguna menemukan data riwayat tertentu serta memantau kondisi tanaman dari waktu ke waktu.

3.8 Analisis Performa Kinerja Sistem

Penelitian ini melakukan pengujian sistem melalui lima skenario untuk mengetahui performa aplikasi CapsiCare. Pada kondisi penggunaan nyata. Pengujian ini bertujuan mengukur kemampuan aplikasi dalam mendeteksi penyakit daun cabai berdasarkan variasi jarak pengambilan gambar, pencahayaan, sudut kamera, latar belakang, serta jumlah daun pada citra. Hasil analisis performa kinerja sistem disajikan pada Tabel 7.

Tabel 7. Analisis Kinerja Sistem Pada Setiap Skenario

No.	Skenario Penguji	Kondisi Uji	Hasil Terdeteksi	Akurasi
1.	Jarak	10 cm	Ya	99,7 %
		20 cm	Ya	60,8%
		30 cm	Ya	34,8%
2.	Pencahayaan	Cahaya terang	Ya	100%
		Cahaya redup	Ya	61,7%
		Cahaya buatan	Ya	66,7%



No.	Skenario Penguji	Kondisi Uji	Hasil Terdeteksi	Akurasi
3.	Sudut Kamera	Tegak lurus	Ya	95,6%
		Miring	Ya	89,8%
		Samping	Ya	94%
4.	Latar Belakang	Polos	Ya	72,9%
		Alami	Ya	89,3%
		Kompleks	Ya	88,2%
5.	Jumlah Daun	Single daun	Ya	99,9%
		Multi daun	Ya	78%

Berdasarkan hasil analisis kinerja sistem pada Tabel 7, faktor jarak pengambilan gambar, pencahayaan, sudut kamera, latar belakang, dan jumlah daun memengaruhi performa sistem dalam mendeteksi penyakit daun cabai. Hasil terbaik diperoleh ketika gambar diambil dari jarak dekat dengan pencahayaan terang alami, serta posisi kamera tegak lurus terhadap objek daun karena detail gejala dapat terlihat dengan lebih jelas. Sebaliknya, penggunaan jarak yang lebih jauh dan pencahayaan yang kurang memadai menyebabkan hasil deteksi menurun. Sistem tetap mampu bekerja dengan baik pada berbagai sudut pengambilan gambar maupun kondisi latar belakang yang berbeda. Selain itu, citra dengan single daun memberikan hasil yang lebih baik dibandingkan beberapa daun dalam satu gambar karena objek terlihat lebih fokus dan tidak saling menutupi.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, sistem deteksi penyakit daun cabai berbasis *mobile* dengan model MaxViT berhasil dikembangkan dengan baik. Sistem ini mampu mengklasifikasikan lima kondisi daun cabai, yaitu *Cercospora*, *Healthy*, *Mites and Thrips*, *Nutritional Deficiency*, dan *Powder Mildew*. Hasil pengujian menunjukkan bahwa pemilihan *optimizer* berpengaruh terhadap performa model. Dari empat skenario yang diuji, *optimizer* Adam memberikan hasil terbaik dengan akurasi sebesar 99,45%, *precision* 99,44%, *recall* 99,45%, dan *F1-Score* 99,45%. Nilai tersebut menunjukkan bahwa model mampu melakukan klasifikasi secara akurat dan konsisten pada seluruh kelas data uji. Analisis confusion matrix juga menunjukkan bahwa penggunaan *optimizer* Adam menghasilkan tingkat kesalahan prediksi paling rendah dibandingkan dengan skenario lainnya, sehingga model lebih baik dalam membedakan karakteristik visual setiap penyakit daun cabai. Model terbaik kemudian berhasil diimplementasikan ke dalam aplikasi *mobile* CapsiCare menggunakan *framework* Flutter dan Format TensorFlow Lite, sehingga dapat dijalankan secara *offline* pada perangkat *smartphone*. Hasil pengujian aplikasi menunjukkan bahwa sistem dapat berfungsi dengan baik pada berbagai kondisi penggunaan, terutama pada jarak dekat, pencahayaan terang alami, dan posisi kamera tegak lurus terhadap objek daun. Namun, penelitian ini masih memiliki keterbatasan karena pengujian hanya dilakukan pada *dataset* tertentu dan belum mencakup variasi kondisi lapangan yang lebih luas, seperti kerusakan daun yang parah, pencahayaan ekstrem, maupun perbedaan spesifikasi kamera perangkat. Penelitian selanjutnya disarankan untuk menambah jumlah data, memperluas variasi kondisi citra, dan menambahkan kelas penyakit daun cabai agar sistem mampu mengenali jenis penyakit yang lebih beragam sesuai kondisi nyata di lapangan. Pengembangan berikutnya dapat mencoba arsitektur *deep learning* lain atau kombinasi *optimizer* yang berbeda untuk membandingkan performa akurasi, efisiensi, dan kecepatan inferensi pada perangkat *mobile*. Pengujian lapangan dengan melibatkan lebih banyak petani maupun penyuluh pertanian agar aplikasi dapat dikembangkan menjadi lebih efektif, sesuai kebutuhan pengguna, serta meningkatkan manfaat aplikasi bagi petani maupun masyarakat.

REFERENCES

- Aishwarya, M. P., & Reddy, A. P. (2024). Dataset of chilli and onion plant leaf images for classification and detection. *Data in Brief*, 54, 110524. <https://doi.org/10.1016/j.dib.2024.110524>
- Ash'shobir, A. H. A., Harli, K. G. P., Rudi, A. P. P., Putro, I. G. S., & Cahyono, O. D. P. (2025). Sistem Deteksi Kualitas Cabai Rawit Menggunakan Metode YOLO (You Only Look Once). *Modem: Jurnal Informatika dan Sains Teknologi*, 3(1), 114–132. <https://doi.org/10.62951/modem.v3i1.363>
- Bichri, H., Chergui, A., & Hain, M. (2023). Image Classification with Transfer Learning Using a Custom Dataset: Comparative Study. *Procedia Computer Science*, 220, 48–54. <https://doi.org/10.1016/j.procs.2023.03.009>
- Irjayanti, A. D., Khairunnisa, D. A., Stiyarningsih, H., Putri, I. M., Sumartini, N. P., & Areka, S. K. (2025). *Statistik Hortikultura 2024 (Statistics of Horticulture 2024)*. Badan Pusat Statistik. <https://www.bps.go.id/id/publication/2025/06/10/aab67e4d36ea6d7bed30d79f/statistics-of-horticulture-2024.html>
- Khasanah, N. U., & Fachrie, M. (2024). Klasifikasi Jenis Penyakit Tanaman Cabai Menggunakan Arsitektur DenseNet201. *SINTECH (Science and Information Technology) Journal*, 7(3), 148–158. <https://doi.org/10.31598/sintechjournal.v7i3.1708>
- Lie, J., Rahman, A., & Udjulawa, D. (2026). Klasifikasi Penyakit Daun Kentang Berbasis CNN MobileNetV2 dengan Optimasi Randomize Search. *TIN: Terapan Informatika Nusantara*, 6(9), 1603–1612. <https://doi.org/10.47065/tin.v6i9.9108>
- Mahajaya, N. S., Ayu, P. D. W., & Huizen, R. R. (2024). Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB



- terhadap Model Vision Transformer pada Klasifikasi Penyakit Paru-paru. *Prosiding Seminar Hasil Penelitian Informatika dan Komputer (SPINTER 2024)*, 1(2), 818–823. <https://spinter.stikom-bali.ac.id/index.php/spinter/article/view/222>
- Murinto, Winiarti, S., & Pujiyanta, A. (2024). Identification of Chili Plant Diseases Based on Leaves Using Hyperparameter Optimization Architecture Convolutional Neural Network. *International Journal of Advanced Computer Science and Applications*, 15(11), 859–865. <https://doi.org/10.14569/IJACSA.2024.0151185>
- Mursyidin, A. H., & Mulyaningsih, T. (2024). Karakterisasi Morfologi dan Anatomi Daun Cabai Rawit (*Capsicum frutescens* L.) yang Terinfeksi *Cercospora capsici* di Lombok Timur. *Jurnal HPT*, 12(3), 173–179. <https://doi.org/10.21776/ub.jurnalhpt.2024.012.3.5>
- Murugavalli, S., & Gopi, R. (2025). Plant Leaf Disease Detection Using Vision Transformers for Precision Agriculture. *Scientific Reports*, 15, 22361. <https://doi.org/10.1038/s41598-025-05102-0>
- Neha, F. N. U., & Bansal, A. (2024). Understanding the architecture of vision transformer and its variants: A review. *1st International Conference on Innovative Engineering Sciences and Technological Research, ICIESTR 2024 - Proceedings*. <https://doi.org/10.1109/ICIESTR60916.2024.10798341>
- Pamungkas, A. S., Triono, J. M., Utomo, E. P. W., & Paramita, C. (2026). Implementasi MobileNetV2 pada Aplikasi Mobile untuk Penilaian Objektif Kondisi Fisik Ponsel Bekas. *TIN: Terapan Informatika Nusantara*, 6(9), 1731–1741. <https://doi.org/10.47065/tin.v6i9.8947>
- Pranta, A. S. U. K., Fardin, H., Debnath, J., Hossain, A., Sakib, A. H., Ahmed, M. R., Haque, R., Reza, A. W., & Dewan, M. A. A. (2025). A Novel MaxViT Model for Accelerated and Precise Soybean Leaf and Seed Disease Identification. *Computers*, 14, 197. <https://doi.org/10.3390/computers14050197>
- Putra, A., & Prihartono, W. (2025). Klasifikasi Penyakit Daun Cabai Dengan Metode CNN Untuk Deteksi Awal. *Jurnal Profesi Insinyur Universitas Lampung*, 6(1). <https://doi.org/10.23960/jpi.v6n1.147>
- Putri, L. P., Muttaqin, M. R., & Ramadhan, Y. R. (2025). Klasifikasi Penyakit Daun Cabai Menggunakan Model MobileNet. *TeknoIS: Jurnal Ilmiah Teknologi Informasi dan Sains*, 15(1), 74–86. <https://doi.org/10.36350/jbs.v15i1.297>
- Rafi, M., Firdaus, A., Mardhiyyah, R., & Sanjaya, F. I. (2025). *TIN: Terapan Informatika Nusantara Klasifikasi Citra Biji Kopi Sangrai Arabika dan Robusta Menggunakan Convolutional Neural Network TIN: Terapan Informatika Nusantara*. 6(7), 969–980. <https://doi.org/10.47065/tin.v6i7.8695>
- Rahman, H., Imran, H. M., Hossain, A., Siddiqui, M. I. H., & Sakib, A. H. (2025). Explainable vision transformers for real-time chili and onion leaf disease identification and diagnosis. *International Journal of Science and Research Archive*, 15(1), 1823–1833. <https://doi.org/10.30574/ijrsra.2025.15.1.1163>
- Ramadhana, F., Abdi Mahardika, G., Day, N., & Puspaningrum, E. Y. (2025). *Klasifikasi Penyakit Daun Cabai Menggunakan Vision Transformer*. 5, 2747–0563. <https://doi.org/10.33005/santika.v5i1.751>
- Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. *Neural Computing and Applications*, 35(23), 17095–17112. <https://doi.org/10.1007/s00521-023-08568-z>
- Sathyanarayanan, S. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, 27(4), 4023–4031. <https://doi.org/10.53555/ajbr.v27i4s.4345>
- Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). MaxViT: Multi-Axis Vision Transformer. *European Conference on Computer Vision (ECCV) 2022*, 1–20. <https://doi.org/10.48550/arXiv.2204.01697>
- Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., Subeesh, A., Upendar, K., Salem, A., & Elbeltagi, A. (2025). Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artificial Intelligence Review*, 58, 92. <https://doi.org/10.1007/s10462-024-11100-x>
- Winanto, T. S., Rozikin, C., & Jamaludin, A. (2023). Analisa Performa Arsitektur Transfer Learning Untuk Mengidentifikasi Penyakit Daun Pada Tanaman Pangan. In *Journal of Applied Informatics and Computing (JAIC)* (Vol. 7, Nomor 1). <https://doi.org/https://doi.org/10.30871/jaic.v7i1.5991>