
PENERAPAN *CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK KLASIFIKASI PENYAKIT TANAMAN CABAI BERBASIS APLIKASI *MOBILE* DENGAN PENDEKATAN *EDGE-AI*

Muhammad Kresna Mahardhika, Erna Dwi Astuti, Nulngafan
Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer,
Universitas Sains Al-Qur'an
kreskres1802@gmail.com

ABSTRAK

Tanaman cabai (*Capsicum sp.*) merupakan salah satu komoditas penting di Indonesia, namun rentan terhadap berbagai penyakit yang menyebabkan penurunan hasil panen. Deteksi dini terhadap penyakit cabai masih dilakukan secara manual, yang memerlukan keahlian khusus dan waktu yang tidak efisien. Penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit tanaman cabai menggunakan metode *Convolutional Neural Network* (CNN) berbasis MobileNetV3, yang dapat diimplementasikan ke dalam aplikasi *mobile* Android berbasis Edge-AI. Dataset terdiri dari 2,500 gambar daun dan buah cabai yang dibagi ke dalam beberapa kelas jenis penyakit seperti antraknosa, bercak daun, keriting daun, dan virus kuning serta 1 kelas untuk tanaman sehat. Citra diproses melalui tahapan *preprocessing* data, kemudian dilatih menggunakan arsitektur MobileNetV3-Small. Hasil pelatihan model terhadap dataset menunjukkan nilai *accuracy* sebesar 0.9843 atau 98% dan *validation accuracy* sebesar 0.9720 atau 97% serta nilai *loss* sebesar 0.0623 dan *loss accuracy* sebesar 0.1059. Kemudian, hasil pelatihan dievaluasi menggunakan data uji dan mendapatkan nilai *test accuracy* sebesar 0.9600 atau 96% dan *test loss* sebesar 0.1195. Setelah itu, model masuk ke dalam tahap *deployment* berupa aplikasi *mobile* berbasis Android dengan pendekatan Edge-AI.

Kata Kunci : Cabai, Convolutional Neural Network, MobileNetV3, Klasifikasi Penyakit Tanaman.

ABSTRACT

Chili plants (Capsicum sp.) are one of the important agricultural commodities in Indonesia, but they are vulnerable to various diseases that lead to decreased crop yields. Early detection of chili diseases is still conducted manually, which requires specialized expertise and is time-inefficient. This study aims to develop a chili plant disease classification model using the Convolutional Neural Network (CNN) method based on MobileNetV3, which can be implemented into an Android-based mobile application using an Edge-AI approach. The dataset consists of 2,500 images of chili leaves and fruits, categorized into several disease classes such as anthracnose, leaf spot, leaf curl, and yellow virus, along with one class for healthy plants. The images were processed through a data preprocessing stage and then trained using the MobileNetV3-Small architecture. The training results on the dataset showed an accuracy of 0.9843 or 98%, a validation accuracy of 0.9720 or 97%, a loss value of 0.0623, and a validation loss of 0.1059. The trained model was then evaluated using a test dataset, achieving a test accuracy of 0.9600 or 96% and a test loss of 0.1195. Finally, the model was deployed into an Android-based mobile application using an Edge-AI approach.

Keywords : Chili Pepper, Convolutional Neural Network, MobileNetV3, Plant Disease Classification.

1. PENDAHULUAN

Tanaman cabai (*Capsicum sp.*) merupakan salah satu komoditas hortikultura yang memiliki peran penting dalam sektor pertanian di Indonesia. Pada tahun 2017-2021, permintaan cabai rawit diproyeksikan mengalami peningkatan sebesar 2,65% tiap tahunnya meliputi kebutuhan bibit, konsumsi, serta bahan baku industri. (Balai Standardisasi Instrumen Pertanian [BSIP] Banten, 2017). Menurut Badan Pusat Statistik, untuk konsumsi cabai besar oleh rumah tangga di Indonesia sendiri mengalami peningkatan hingga 675,02 ribu ton pada tahun 2023. Permintaan yang tinggi sepanjang tahun menjadikan cabai sebagai tanaman bernilai ekonomi tinggi. Di sisi lain, kendala produksi cabai sering terjadi akibat serangan berbagai penyakit yang menyebabkan penurunan hasil panen secara signifikan. Serangan dari penyakit seperti layu batang, rontok buah, dan infeksi virus kuning (*Begomovirus*), berkontribusi terhadap penurunan produksi tanaman cabai secara signifikan. Dalam kondisi optimal, hasil panen cabai dapat mencapai 80 kilogram per periode panen. Namun, akibat infestasi patogen tersebut, produktivitas tanaman mengalami penurunan drastis hingga hanya mencapai 34 kilogram per periode panen (TVRI Yogyakarta News, 2025).

Dalam beberapa tahun terakhir, teknologi kecerdasan buatan (AI) telah berkembang pesat dan mulai diterapkan dalam berbagai bidang, termasuk bidang pertanian. Salah satu metode yang efektif dalam menganalisis dan mengklasifikasikan penyakit tanaman adalah *Convolutional Neural Network* (CNN), sebuah bagian dari *machine learning*, dan merupakan inti dari algoritma *deep learning* yang dapat mengenali pola gambar dengan akurasi yang tinggi (IBM, 2021).

Kurniawan, Sari, dan de Jong (2024) membandingkan beberapa arsitektur CNN dalam klasifikasi gambar, dengan hasil nilai *accuracy* sebesar 50% untuk arsitektur ResNet50, nilai *accuracy* sebesar 85% untuk arsitektur VGG16 dan nilai *accuracy* sebesar 91% untuk arsitektur MobileNet. Dari hasil perbandingan 3 arsitektur CNN tersebut, MobileNet terbukti efektif dalam melakukan klasifikasi gambar dengan nilai akurasi yang cukup tinggi. MobileNet memiliki ukuran

model yang lebih ringan dan waktu komputasi pelatihan yang lebih singkat, sehingga sangat cocok untuk diimplementasikan pada aplikasi *mobile* dan *embedded* dengan keterbatasan sumber daya (Sulistio & Lubis, 2023). Dengan memanfaatkan *Convolutional Neural Network* (CNN) dengan arsitektur MobileNet, klasifikasi penyakit pada tanaman cabai dapat dilakukan secara lebih cepat dan akurat dibandingkan metode konvensional dengan pengamatan manual oleh petani.

Selain itu, untuk memastikan klasifikasi penyakit dapat dilakukan secara fleksibel tanpa bergantung pada koneksi internet, pendekatan *Edge-AI* diterapkan dalam pengembangan aplikasi *mobile* dalam penelitian ini. *Edge-AI* adalah penerapan kecerdasan buatan (AI) di dalam sistem *Edge Computing*. *Edge-AI* merupakan integrasi antara teknologi *Edge Computing* dan kecerdasan buatan (AI), dimana proses komputasi algoritma AI dilakukan langsung pada perangkat pengguna, tanpa perlu mengirim data ke server pusat. Pendekatan ini memberikan berbagai keuntungan, seperti pengurangan latensi, efisiensi konsumsi energi, serta dukungan untuk aplikasi yang membutuhkan respons secara *real-time*. Selain itu, *Edge-AI* mampu mengatasi kendala yang sering muncul pada sistem berbasis cloud, seperti keterbatasan konektivitas jaringan dan risiko kebocoran data. Implementasi AI di tepi perangkat juga mendukung tugas-tugas yang sensitif terhadap waktu serta mengurangi beban *traffic* data, sehingga meningkatkan performa dan keamanan jaringan (Gill dkk., 2024).

Berdasarkan permasalahan yang telah diuraikan, penelitian ini bertujuan untuk mengembangkan model CNN menggunakan arsitektur MobileNet yang mampu melakukan klasifikasi penyakit tanaman cabai berdasarkan citra daun dan buah tanaman cabai dengan mengimplementasikannya ke dalam aplikasi *mobile* berbasis *Edge-AI*. Dengan solusi tersebut, diharapkan petani dapat lebih mudah mengklasifikasikan penyakit pada tanaman cabainya.

2. METODE

Penelitian ini menggabungkan metode *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV3, menggunakan TensorFlow Lite untuk *deployment* model, dan

membangun aplikasi Android agar dapat digunakan secara langsung oleh petani atau pengguna umum.

2.1. Pengumpulan Data dan Augmentasi

Pengumpulan data dilakukan dengan mencari lalu mengunduh dataset gambar yang akan digunakan pada *platform* Kaggle dan Roboflow, selain itu dilakukan juga observasi langsung pada perkebunan cabai di daerah Kampung Sayur Blederan, dan Desa Mlandi, Kecamatan Garung, Kabupaten Wonosobo. Selain itu, observasi juga dilakukan pada area perkebunan cabai di Desa Batusari, Kecamatan Kledung, Kabupaten Temanggung. Total data gambar yang didapatkan dari unduhan dan observasi berjumlah 2,690. Data asli yang digunakan berjumlah 2,690 data gambar yang terdiri dari empat kelas penyakit tanaman cabai yaitu penyakit Antraknosa, Bercak Daun, Virus Kuning dan Keriting Daun, serta satu kelas tanaman cabai Sehat dengan jumlah yang bervariasi. Untuk menjaga keseimbangan antar kelas pada proses pelatihan, dilakukan augmentasi data secara manual pada beberapa kelas penyakit, sehingga setiap kelas memiliki 500 data gambar dan menghasilkan total 2,500 data.

2.2. Data Preprocessing

Pada data preprocessing, dataset yang telah dikumpulkan kemudian dibagi menjadi 3 bagian, yaitu 80% untuk data training, 10% untuk data validasi dan 10% untuk data uji. Data tersebut kemudian diletakkan pada Google Drive dengan nama *dataset_split* yang akan digunakan sebagai data pelatihan, validasi, dan data uji model.

Normalisasi data juga dilakukan pada tahap ini dengan menggunakan *preprocess_input* yang sudah disediakan di dalam tensorflow untuk *preprocessing* arsitektur MobileNetV3 yang fungsinya sebagai pengubah piksel yang semula bernilai antara 0 hingga 255 menjadi nilai dalam rentang [-1,1] dengan perhitungan atau rumus:

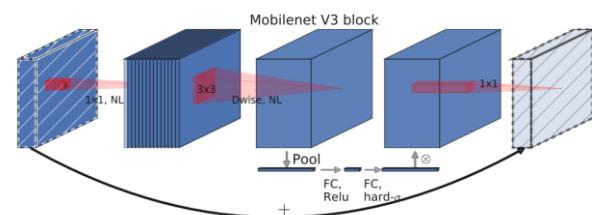
$$x_{normalized} = \left(\frac{x}{127.5} \right) - 1 \quad (1)$$

Setelah proses augmentasi, normalisasi, dan *training split* dilakukan, kemudian dataset yang telah dibagi untuk *training* dan *validation* lalu di-*resize* menjadi ukuran 224 x 224 piksel dengan pembagian per-*batch* pada *validation*

dan *training* masing-masing berjumlah 32 gambar. *Shuffle* (acak) diaktifkan pada saat training agar setiap kali model dilatih urutan data diacak, untuk mencegah model menghafal urutan data, karena jika data selalu datang dengan urutan yang sama, model bisa belajar urutan, bukan pola visual. Sedangkan pada *validation*, *shuffle* (acak) dinonaktifkan agar mendapatkan evaluasi yang akurat dan konsisten.

2.3. Modelling

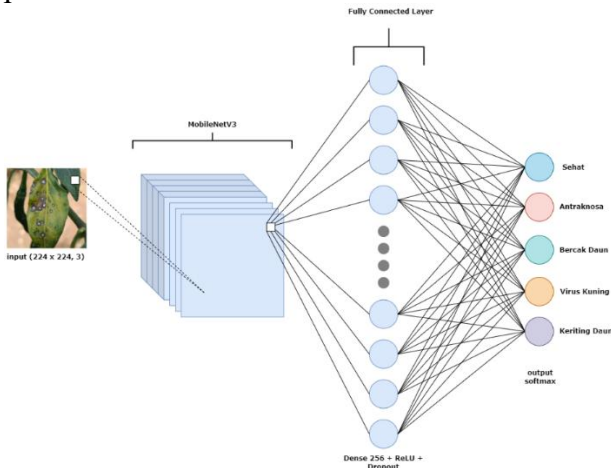
Setelah tahap preprocessing, dilakukan tahap modelling yang dimana peneliti memilih model arsitektur untuk algoritma Convolutional Neural Network (CNN). Arsitektur yang dipilih adalah MobileNetV3 yang merupakan pengembangan dari versi sebelumnya (V1 dan V2), yang menggabungkan teknik *depthwise separable convolution*, *squeeze-and-excitation*, serta pemanfaatan *Neural Architecture Search* (NAS) untuk arsitektur ringan namun akurat. MobileNetV3 menggunakan blok *bottleneck* dengan kombinasi *depthwise* dan *pointwise convolution*, serta aktivasi *h-swish* yang lebih efisien dari ReLU. Arsitekturnya terdiri dari sekitar 20 lapisan konvolusi, diakhiri dengan *global average pooling* dan *fully connected layer*. Hasil akhir diproses oleh *softmax* untuk menentukan kelas penyakit cabai berdasarkan probabilitas tertinggi.



Gambar 1. Arsitektur MobileNetV3 (Howard dkk, 2019)

Model yang dipilih oleh peneliti adalah MobileNetV3-Small karena struktur arsitektur yang ringan dan efisien, serta dioptimalkan untuk efisiensi di perangkat dengan keterbatasan sumber daya komputasi, sehingga sangat cocok untuk diterapkan pada aplikasi *mobile*. MobileNetV3-Small menggunakan teknik *depthwise separable convolution* yang memisahkan proses konvolusi menjadi *depthwise* (misalnya filter 3×3 atau 5×5) dan

pointwise (1×1), sehingga lebih efisien secara komputasi. Arsitektur ini juga memanfaatkan *bottleneck block* untuk menurunkan dan menyesuaikan dimensi fitur, serta menerapkan teknik *Squeeze-and-Excitation* (SE) untuk memperkuat fitur penting. Setiap layer umumnya disertai dengan *Batch Normalization* dan fungsi aktivasi *h-swish* untuk mempercepat dan menstabilkan proses pelatihan. Di bawah ini merupakan Arsitektur Jaringan yang digunakan pada model ini.



Gambar 2. Model Arsitektur MobileNet yang digunakan

Pada Gambar 2.2, arsitektur MobileNet digunakan untuk ekstraksi fitur dengan menerapkan *layer dense* berukuran 256. Proses ini diakhiri dengan layer output yang menggunakan fungsi *softmax* dengan 5 kelas untuk klasifikasi.

Tabel 1. Spesifikasi MobileNetV3 Small

Input	Operator	exp size	#out	SE	NL	s
224 ² × 3	conv2d, 3x3	-	16	-	HS	2
112 ² × 16	bneck, 3x3	16	16	✓	RE	2
56 ² × 16	bneck, 3x3	72	24	-	RE	2
28 ² × 24	bneck, 3x3	88	24	-	RE	1
28 ² × 24	bneck, 5x5	96	40	✓	HS	2
14 ² × 40	bneck, 5x5	240	40	✓	HS	1
14 ² × 40	bneck, 5x5	240	40	✓	HS	1
14 ² × 40	bneck, 5x5	120	48	✓	HS	1
14 ² × 48	bneck, 5x5	144	48	✓	HS	1
14 ² × 48	bneck, 5x5	288	96	✓	HS	2
7 ² × 96	bneck, 5x5	576	96	✓	HS	1
7 ² × 96	bneck, 5x5	576	96	✓	HS	1
7 ² × 96	conv2d, 1x1	-	576	✓	HS	1
7 ² × 576	pool, 7x7	-	-	-	-	1
1 ² × 576	conv2d 1x1, NBN	-	1280	-	HS	1
1 ² × 1280	conv2d 1x1, NBN	-	k	-	-	1

Model dilatih menggunakan platform Google Colab dengan framework TensorFlow,

menggunakan dataset gambar yang telah melalui tahap *preprocessing*. Data kemudian dibagi menjadi tiga *subset*, yaitu *training*, *validation*, dan *test*, guna mencegah *overfitting* sekaligus memastikan kemampuan generalisasi model. Berbagai konfigurasi pelatihan diuji, meliputi variasi arsitektur model, parameter optimisasi, dan teknik augmentasi data. Setelah serangkaian eksperimen, konfigurasi yang memberikan performa terbaik dari segi akurasi, konsistensi hasil, dan efisiensi waktu pelatihan dipilih sebagai konfigurasi utama penelitian ini. Konfigurasi pelatihan yang digunakan adalah sebagai berikut:

- *Epoch*: 100
- *Batch size*: 32
- *Learning rate*: 0,0001
- *Optimizer*: Adam
- *Early stopping*: diaktifkan untuk menghentikan proses pelatihan ketika *validation loss* tidak mengalami perbaikan.

Pemilihan konfigurasi ini didasarkan pada praktik umum pelatihan model CNN ringan yang efektif mempercepat proses sekaligus mempertahankan akurasi. Selanjutnya, model dilatih dengan berbagai kombinasi yang dikembangkan dari konfigurasi dasar tersebut. Menurut Howard dkk. (2019), MobileNetV3 secara signifikan meningkatkan keseimbangan antara latency dan akurasi pada aplikasi visi komputer di perangkat mobile, sehingga arsitektur ini sangat tepat untuk diimplementasikan dalam aplikasi klasifikasi penyakit tanaman berbasis Android.

2.4. Evaluation

Tahap *evaluation* adalah tahap terakhir dari pemodelan yang bertujuan untuk mengukur performa model *Convolutional Neural Network* (CNN) dalam mengklasifikasi jenis penyakit tanaman cabai berdasarkan citra daun dan buah. *Evaluation* dilakukan dengan menggunakan metrik klasifikasi berbasis *confusion matrix* serta perhitungan akurasi, presisi, *recall*, dan *F1-Score* terhadap data uji secara manual untuk

memberikan gambaran kinerja model secara lebih detail.

2.5. Deployment

Setelah seluruh tahapan pengembangan model CNN selesai, tahap selanjutnya adalah deployment model. Penelitian ini memilih aplikasi mobile berbasis Android sebagai media *deployment* karena teknologi Edge-AI memungkinkan model dijalankan secara langsung pada perangkat tanpa memerlukan koneksi internet. Dengan demikian, pengguna dapat mengakses aplikasi secara *offline* dan mendapatkan hasil klasifikasi secara langsung.

3. HASIL DAN PEMBAHASAN

Berdasarkan percobaan pada penelitian yang telah dilakukan, berikut adalah hasil dari pelatihan dengan berbagai macam kombinasi konfigurasi guna membandingkan kombinasi mana yang lebih optimal, dan juga hasil pengukuran menggunakan berbagai macam metrics evaluasi terhadap data uji.

3.1. Hasil Uji Konfigurasi Model

Serangkaian eksperimen konfigurasi dilakukan, untuk memperoleh performa model yang optimal pada tahap pelatihan *Convolutional Neural Network* (CNN). Konfigurasi yang diuji meliputi jumlah unit pada *Layer Dense*, penggunaan data augmentasi, variasi nilai *learning rate*, dan jumlah *epoch* pelatihan. Hasil dari berbagai kombinasi tersebut kemudian dibandingkan berdasarkan nilai akurasi dan validasi akurasi tertinggi yang dicapai. Berikut adalah hasil dari beberapa konfigurasi model yang telah dilakukan.

Tabel 2. Hasil Uji Konfigurasi

No.	Layer Dense	Augmentasi	Macam Augmentasi	Epoch	Akurasi	Val. Akurasi
1.	256	Ya	Rotate, zoom, shift & flip	10	0.8710	0.8640
2.	256	Ya	Rotate, zoom, shift & flip	20	0.9034	0.9200
4.	512, 256	Ya	Rotate, zoom, shift & flip	25	0.9072	0.9360
5.	512, 256	Ya	Rotate, zoom, shift & flip	48/50	0.9398	0.9280
6.	256	Ya	Rotate & flip	50	0.9402	0.9400
7.	256	Ya	Rotate & flip	63/100	0.9734	0.9400
8.	512, 256	Ya	Rotate & flip	32/100	0.9523	0.9640
9.	512	Ya	Rotate & flip	64/100	0.9809	0.9600
10.	256	Tidak	-	96/100	0.9819	0.9680

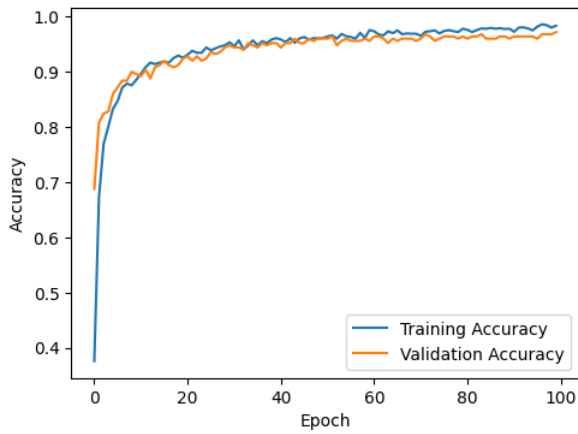
Setelah dilakukan pelatihan dengan berbagai kombinasi, diperoleh hasil terbaik pada konfigurasi berikut:

- *Epoch* :100
- *Batch size* : 32
- *Learning rate* : 1-4e (0,0001)
- *Optimizer* : Adam
- Jumlah unit *Dense* : 256
- Augmentasi Data :Tidak digunakan
- *Epoch* : 96/100
- Akurasi tertinggi : 0.9819
- Akurasi Validasi tertinggi: 0.9680

Konfigurasi tersebut menghasilkan model yang paling stabil dan akurat dalam mengklasifikasikan citra daun dan buah tanaman cabai ke dalam kelas-kelas penyakit yang telah ditentukan. Penggunaan *Layer Dense* 256 menunjukkan kemampuan generalisasi yang lebih baik dibandingkan konfigurasi lain. Selain itu, tidak digunakannya augmentasi justru memberikan hasil yang lebih optimal pada dataset ini, yang kemungkinan besar disebabkan oleh kualitas citra yang cukup merepresentasikan masing-masing kelas.

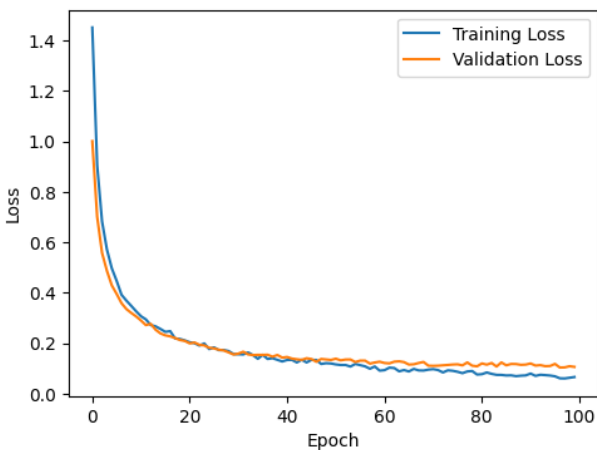
3.2. Hasil Uji Model Klasifikasi

Dengan hasil akhir *training* tepatnya pada *Epoch* ke-100 (Gambar 4.8) dengan nilai pada *accuracy* sebesar 0.9843 dan *loss* sebesar 0.0623, serta *validation accuracy* sebesar 0.9720 dan *validation loss* sebesar 0.1059. Berikut adalah gambar kurva *training* (*accuracy* dan *loss*) dan *validation*-nya (*validation training* dan *validation loss*).



Gambar 3. Kurva Training dan Validation Accuracy

Berdasarkan grafik *Training* dan *Validation Accuracy* pada Gambar 3. terlihat bahwa *Training Accuracy* (biru) terus meningkat hingga mencapai nilai 0.9843 atau 98%, sedangkan *Validation Accuracy* (jingga) juga mengalami peningkatan secara stabil hingga mencapai sekitar 0.9720 atau 97%. Selisih (*gap*) antara *Training* dan *Validation Accuracy* yang sangat kecil dan stabil menunjukkan bahwa model mampu belajar dengan baik tanpa *overfitting* yang signifikan.



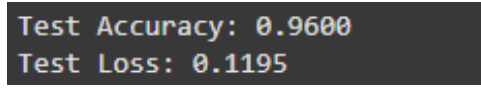
Gambar 4. Kurva Training dan Validation Loss

Sementara itu, pada Gambar 4. yang menampilkan grafik *Training* dan *Validation Loss*, terlihat bahwa *Training Loss* (biru) terus menurun hingga bernilai 0.0623. *Validation Loss* (jingga) juga menunjukkan tren penurunan meskipun hingga bernilai 0.1059. Tidak terdapat indikasi *Validation Loss* yang meningkat secara konsisten, sehingga dapat disimpulkan bahwa model tidak mengalami

overfitting yang parah. Penurunan kurva yang signifikan di awal pelatihan dan kestabilan di akhir pelatihan menunjukkan proses pembelajaran yang efektif.

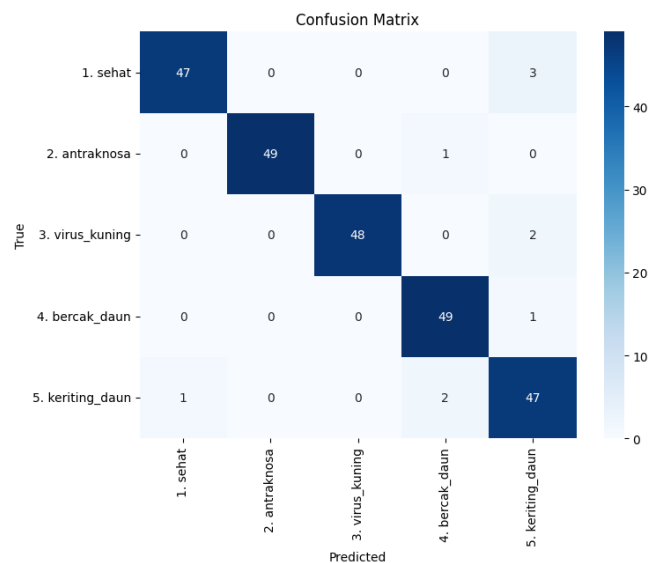
3.3. Hasil Evaluasi Model

Setelah dilakukan evaluasi model terhadap data uji hasil splitting sebesar 10% dari 2500 dataset, didapatkan hasil *test accuracy* sebesar 0.9600 atau dalam persentase adalah 96% dan *test loss* sebesar 0.1195 yang dibuktikan dengan gambar *test accuracy* pada gambar berikut.



Gambar 5. Evaluasi model

Setelah dilakukan evaluasi menggunakan *matrix accuracy* dan *loss* terhadap data uji, model juga dianalisis lebih lanjut menggunakan nilai presisi, *recall*, dan F1-Score. Nilai-nilai tersebut dihitung berdasarkan parameter dari *confusion matrix* berikut:



Gambar 6. Confusion Matrix Multiclass terhadap data uji

Dari *Confusion Matrix* hasil evaluasi terhadap data uji diatas, kemudian dihitung menggunakan metrik *precision*, *recall*, dan F1-Score, maka dapat ditemukan hasil perhitungan dari *metrics* evaluasi terhadap data uji (*test*) pada setiap kelas sebagai berikut:

Tabel 3. Precision

Kelas	Precision
Sehat	0.98 (98%)
Antraknosa	1.00 (100%)
Virus Kuning	1.00 (100%)
Bercak Daun	0.94 (94%)
Keriting Daun	0.89 (89%)

Tabel 4. Recall

Kelas	Recall
Sehat	0.94 (94%)
Antraknosa	0.98 (98%)
Virus Kuning	0.96 (96%)
Bercak Daun	0.98 (98%)
Keriting Daun	0.94 (94%)

Tabel 5. F1-Score

Kelas	F1-Score
Sehat	0.96 (96%)
Antraknosa	0.99 (99%)
Virus Kuning	0.98 (98%)
Bercak Daun	0.96 (96%)
Keriting Daun	0.91 (91%)

Setelah evaluasi dilakukan, hasil *training* model kemudian disimpan dengan format Keras di dalam Google Drive untuk dilakukan *deployment* setelahnya

3.4. Deployment Aplikasi

Deployment aplikasi merupakan tahap akhir dalam pengembangan sistem, dimana model yang telah dilatih dan diuji kemudian diterapkan ke dalam aplikasi untuk dijalankan pada perangkat nyata yang dalam hal ini adalah aplikasi *mobile* berbasis android. Model kemudian disimpan di dalam Google Drive dengan format keras, namun agar aplikasi dapat berjalan optimal pada perangkat dengan spesifikasi rendah, perlu dilakukan konversi ke dalam format tflite.

Model yang telah dikonversi menjadi tflite kemudian diunduh dan ditempatkan di dalam folder *assets* pada proyek Android Studio. Pendekatan ini termasuk dalam kategori *Edge-AI*, karena model tidak diakses melalui API atau

server eksternal, melainkan ditanamkan langsung ke dalam aplikasi sehingga dapat dijalankan secara lokal di perangkat pengguna.

3.5. Pengujian Teknis Aplikasi

Setelah aplikasi selesai dikembangkan, kemudian dilakukan *build app* agar aplikasi dapat dijalankan di perangkat Android. Kemudian dilakukan uji teknis aplikasi agar dapat diketahui apakah aplikasi dapat berjalan baik tanpa kendala atau masih terdapat *bug*. Berikut ini adalah hasil analisisnya.

Tabel 6. Hasil Uji Teknis Aplikasi

No.	Fitur yang Diuji	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
1.	Impor Gambar	User menekan tombol "Unggah Gambar" dan memilih gambar dari galeri	Gambar berhasil dipilih dan ditampilkan di <i>Result</i>	Berhasil
2.	Kamera Input	User menekan tombol "Ambil Foto" dan memvalidasi foto	Gambar berhasil diambil dan ditampilkan di <i>Result</i>	Berhasil
3.	Klasifikasi Gambar	Gambar yang telah di-input user diproses menggunakan model TFLite	Label dan <i>confidence</i> klasifikasi tampil di <i>Result</i>	Berhasil
4.	Navigasi Antar Halaman	Aplikasi berpindah dari Beranda ke <i>Result</i> setelah gambar dipilih	Halaman <i>Result</i> tampil dengan data hasil klasifikasi	Berhasil
5.	Performa Model	Model berjalan di perangkat tanpa internet	Klasifikasi tetap berjalan di mode <i>offline</i>	Berhasil
6.	Tampilan Hasil Klasifikasi	Label, <i>confidence</i> , dan gambar input tampil lengkap di halaman <i>Result</i>	Semua elemen UI tampil sesuai	Berhasil










Berdasarkan hasil pengujian teknis (Tabel 6) yang telah dilakukan terhadap aplikasi mobile dapat disimpulkan bahwa seluruh fungsi utama aplikasi berjalan dengan baik dan sesuai dengan harapan. Fitur-fitur seperti impor gambar dari galeri, ambil foto, klasifikasi gambar, navigasi antar halaman, serta tampilan hasil klasifikasi dapat berfungsi secara optimal tanpa terjadi kendala atau *crash*. Selain itu, model klasifikasi dapat dijalankan secara *offline* di perangkat Android, sehingga mendukung pendekatan *Edge-AI* dan membuat aplikasi lebih praktis

untuk digunakan di lapangan tanpa memerlukan koneksi internet.

3.6. Pengujian Model Terhadap Aplikasi









Dari pengujian hasil klasifikasi pada aplikasi menggunakan data baru dengan input dari galeri dan ambil gambar langsung dengan kamera adalah sebagai berikut.



Tabel 7. Pengujian Model Terhadap Aplikasi Sumber Impor Gambar

No.	Result Screen	Prediksi	Aktual	Confidence	Keterangan
1.		Virus Kuning	Virus Kuning	99,99%	Valid
2.		Keriting Daun	Keriting Daun	90,96%	Valid
3.		Bercak Daun	Bercak Daun	99,98%	Valid
4.		Bercak Daun	Bercak Daun	99,80%	Valid
5.		Keriting Daun	Virus Kuning	90,93%	Tidak Valid
6.		Virus Kuning	Virus Kuning	97,56%	Valid
7.		Bercak Daun	Bercak Daun	99,90%	Valid
8.		Antraknosa	Antraknosa	92,71%	Valid
9.		Keriting Daun	Keriting Daun	85,33%	Valid

10.		Bercak Daun	Bercak Daun	99,54%	Valid
-----	--	-------------	-------------	--------	-------

Tabel 8. Pengujian Model Terhadap Aplikasi Sumber Kamera

No.	Result Screen	Prediksi	Aktual	Confidence	Keterangan
1.		Antraknosa	Antraknosa	99,96%	Valid
2.		Antraknosa	Antraknosa	97,13%	Valid
3.		Keriting Daun	Keriting Daun	99,22%	Valid
4.		Keriting Daun	Bercak Daun	78,24%	Tidak Valid
5.		Virus Kuning	Virus Kuning	85,49%	Valid
6.		Antraknosa	Antraknosa	100%	Valid
7.		Keriting Daun	Keriting Daun	95,70%	Valid
8.		Virus Kuning	Virus Kuning	99,33%	Valid

9.	 Sehat	Sehat	Sehat	87,93%	Valid
10.	 Bercak Daun	Bercak Daun	Bercak Daun	92,65%	Valid

Berdasarkan 10 gambar baru yang diambil dari galeri dan 10 gambar yang diambil langsung menggunakan input kamera pada aplikasi, didapatkan hasil pengujian menggunakan data baru pada Tabel 7. dan Tabel 8. Dari hasil pengujian tersebut dapat disimpulkan akurasi pengujian aplikasi mobile klasifikasi penyakit tanaman cabai dengan data baru yaitu dapat dihitung sebagai berikut:

$$Acc = \frac{18}{20} \times 100 = 90\% \quad (1)$$

Hasil perhitungan di atas merupakan nilai *accuracy* aplikasi mobile klasifikasi penyakit tanaman cabai terhadap 20 gambar baru dengan input gambar dari galeri dan input gambar dari kamera mendapatkan nilai *accuracy* sebesar 90%. Ini membuktikan bahwa model yang telah diimplementasikan ke dalam aplikasi dapat mengklasifikasikan penyakit dengan optimal.

4. PENUTUP

4.1. Kesimpulan

Model klasifikasi penyakit tanaman cabai yang dikembangkan menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur MobileNet mampu memproses dan mengklasifikasikan citra secara optimal pada tahap pelatihan dan pengujian. Dari banyaknya pelatihan dengan berbagai kombinasi konfigurasi yang telah dilakukan, didapatkan konfigurasi pelatihan terbaik dengan jumlah unit Dense sebesar 256, tanpa augmentasi data dengan python, 100 *Epochs* dan dengan *Early Stopping* sebesar 15. Konfigurasi tersebut menghasilkan nilai akurasi tertinggi yaitu sebesar 0.9819 atau 98% dengan validasi akurasi tertingginya pada 0.9680 atau 96%. Kemudian, setelah dilakukan pelatihan ulang menggunakan kombinasi tersebut, didapatkan hasil baru dengan nilai *accuracy* sebesar 0.9843 atau 98% dan *validation accuracy* sebesar

0.9720 atau 97% serta nilai *loss* sebesar 0.0623 dan *loss accuracy* sebesar 0.1059. Hasil ini menunjukkan bahwa model yang dilatih mampu menghasilkan performa yang optimal untuk melakukan klasifikasi penyakit tanaman cabai berdasarkan citra daun dan buah, dibuktikan dengan hasil evaluasi menggunakan data uji mendapatkan nilai *test accuracy* sebesar 0.9600 atau 96% dan *test loss* sebesar 0.1195.

Model yang dirancang dapat mengenali pola penyakit dengan cukup baik, dengan tingkat akurasi model sebesar 98% terhadap 2000 data latih dan tingkat akurasi sebesar 97% terhadap 250 data validasi serta tingkat akurasi sebesar 96% terhadap 250 data uji. Pada penilaian menggunakan metrik evaluasi *Precision* untuk setiap kelas menghasilkan nilai 98% untuk kelas Sehat, 100% untuk kelas Antraknosa, 100% untuk kelas Virus Kuning, 94% untuk kelas Bercak Daun dan 89% untuk kelas Keriting Daun. Pada penilaian metrik evaluasi Recall untuk setiap kelas menghasilkan nilai 94% untuk kelas Sehat, 98% untuk kelas Antraknosa, 96% untuk kelas Virus Kuning, 98% untuk kelas Bercak Daun dan 94% untuk kelas Keriting Daun. Kemudian, pada penilaian metrik evaluasi *F1-Score* untuk setiap kelas menghasilkan nilai 96% untuk kelas Sehat, 99% untuk kelas Antraknosa, 98% untuk kelas Virus Kuning, 96% untuk kelas Bercak Daun dan 91% untuk kelas Keriting Daun. Model yang telah melalui tahap evaluasi kemudian dilakukan *test* terhadap 20 data baru di dalam aplikasi dan didapatkan hasil akurasi yang cukup baik yaitu 90%.

Hasil pengujian terhadap aplikasi klasifikasi penyakit tanaman cabai pada Tabel 6. menunjukkan bahwa seluruh fungsi utama aplikasi berjalan dengan baik dan sesuai dengan yang diharapkan. Implementasi *Edge-AI* pada aplikasi juga terbukti efisien dan responsif. Proses klasifikasi terhadap citra yang diinput berlangsung kurang dari 1 detik, serta dapat dilakukan secara *offline*. Hal ini mengindikasikan bahwa penerapan *Edge-AI* dalam pengembangan aplikasi klasifikasi penyakit tanaman cabai telah berhasil dilakukan dengan baik.

4.2. Saran

Dari penelitian yang telah dilakukan oleh penulis, telah mendapatkan hasil akurasi yang cukup baik menggunakan MobileNetV3-

Small. Adapun saran yang dapat dijadikan acuan dalam pengembangan penelitian di masa mendatang yaitu, tambahkan jumlah dan variasi penyakit lain untuk data citra daun serta buah cabai, uji akurasi dengan arsitektur lain seperti EfficientNet atau MobileNetV3-Large untuk mencari arsitektur mana yang lebih optimal dalam mengklasifikasikan penyakit tanaman cabai.

5. DAFTAR PUSTAKA

- Balai Penerapan Standar Instrumen Pertanian Banten (2025) *Teknologi budidaya tanaman cabai (Capsicum annum L)*. Kementerian Pertanian Republik Indonesia.
- Badan Pusat Statistik (2023) *Statistik Hortikultura 2023*. Jakarta: Badan Pusat Statistik.
- Sulistio, M.J. and Lubis, C. (2023) 'Implementasi CNN dan MobileNet untuk mendeteksi penyakit pneumonia dan COVID-19 dengan menggunakan aplikasi smartphone', *Nusantara Journal of Multidisciplinary Science*, 1(4), pp. 736–745.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., and Adam, H., 2019. *Searching for MobileNetV3*. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, pp.1314–1324.
- Gill, S.S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., Walia, G.K., Murugesan, S.S., Ali, B., Kumar, M., Ye, K., Verma, P., Kumar, S., Cuadrado, F. and Uhlig, S. (2024) 'Edge AI: A taxonomy, systematic review and future directions', *arXiv*.
- Kurniawan, M., Sari, A.I. and de Jong, P.A. (2024) 'Perbandingan kinerja metode CNN dalam mengklasifikasi orang merokok', *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 11(2), pp. 221–229.
- IBM (2021) *Apa itu convolutional neural network?* IBM. Available at: <https://www.ibm.com/id-id/think/topics/convolutional-neural-networks/> (Accessed: June 5th 2025)
- TVRI Yogyakarta News (2025) 'Terserang banyak penyakit, panen cabai turun drastis', *TVRI Yogyakarta News*, 17 Januari. Available at: <https://tvriyogyakartanews.com/2025/01/17/terserang-banyak-penyakit-panen-cabai-turun-drastis/> (Accessed: June 5th 2025)