Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 3 July 2025, pp: 841-850
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Implementation of Computer Vision for Detecting Diseases in Chili and Tomato Plants Using the Convolutional Neural Networks Method

Implementasi *Computer Vision* dalam Mendeteksi Penyakit pada Tanaman Cabai dan Tomat Menggunakan Algoritma Convolutional Neural Networks

Deris Pakiding[1*], Ahmad Selao[2], Wahyuddin[3]

[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik,
Universitas Muhammadiyah Parepare, Indonesia

E-Mail: [1]pakidingd86@gmail.com,
[2]ahmadselao@umpar.ac.id, [3]wahyuddin081090@gmail.com

Received Mar 03rd 2025; Revised May 20th 2025; Accepted Jun 06th 2025; Available Online Jun 23th 2025, Published Jun 23th 2025
Corresponding Author: Deris Pakiding

Abstract

Computer Vision is widely used in agriculture to detect plant diseases automatically. This study develops a web-based application using Convolutional Neural Networks (CNN) to detect diseases in chili and tomato plants. The application was built using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Python, JavaScript, and Flask, allowing users to upload plant images for classification. The CNN model achieved 91% training accuracy. When tested on a separate dataset, the model maintained a high accuracy of 91%. However, in real-world detection via the application, accuracy dropped to 75%, likely due to lighting, image quality, and dataset differences. Misclassifications were most common in Chili Leaf Curl 50% and Tomato Normal 40% from 10 test samples each. Performance evaluation using precision, recall, and F1-score showed that Chili Leaf Curl had the highest recall of 0.9619 and precision of 0.9711. In contrast, Tomato Late Blight had the lowest recall, 0.7594, and F1-score, 0.8510, indicating classification challenges. Heatmap analysis showed the model focused on specific image features, but not always on the disease areas. Further improvements in data preprocessing, image augmentation, and model architecture are needed to enhance accuracy and reduce misclassifications.

Keyword: Computer Vision, Convolutional Neural Networks, Disease, Plant, Python

Abstrak

Computer Vision banyak diterapkan dalam bidang pertanian untuk mendeteksi penyakit tanaman secara otomatis. Penelitian ini mengembangkan aplikasi berbasis web menggunakan Algoritma Convolutional Neural Networks (CNN) untuk mendeteksi penyakit pada tanaman cabai dan tomat. Aplikasi ini dibangun dengan HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Python, JavaScript, dan Flask, memungkinkan pengguna mengunggah gambar tanaman untuk diklasifikasikan. Model CNN mencapai akurasi pelatihan 91%. Saat diuji menggunakan dataset terpisah, model memperoleh akurasi tinggi. Namun, setelah diterapkan dalam aplikasi untuk deteksi nyata, akurasi menurun menjadi 75%, yang bisa disebabkan oleh kondisi pencahayaan, kualitas gambar, dan perbedaan dataset. Kesalahan untuk deteksi nyata terutama terjadi pada kelas Cabai Leaf Curl 50% dan Tomat Normal 40% dari masing-masing 10 kali uji deteksi sampel. Analisis kinerja model menggunakan precision, recall, dan F1-score menunjukkan bahwa Cabai Leaf Curl memiliki recall tertinggi sebesar 0.9619 dan memiliki precision tertinggi 0.9711. Sementara pada Tomat Late Blight mendapatkan skor terendah pada Recall 0.7594 dan F1-Score 0.8510, menunjukkan tantangan dalam klasifikasi. Analisis heatmap mengungkapkan bahwa model berfokus pada fitur gambar tertentu tetapi tidak selalu pada area penyakit. Peningkatan kinerja melalui optimasi preprocessing data, augmentasi gambar, dan arsitektur model yang lebih kompleks diperlukan untuk meningkatkan akurasi serta mengurangi kesalahan klasifikasi.

Kata Kunci: Computer Vision, Convolutional Neural Networks, Penyakit, *Python*, Tanaman

1. PENDAHULUAN

Computer Vision merupakan bidang yang memungkinkan mesin untuk "Melihat" dengan menggunakan kamera dan komputer sebagai pengganti mata manusia untuk mengidentifikasi, melacak, dan mengukur objek sebelum diproses lebih lanjut secara digital [1]. Kemajuan teknologi di bidang *Computer Vision* telah memberikan berbagai solusi inovatif dalam berbagai sektor, termasuk pertanian. Salah satu tantangan utama dalam pertanian adalah penyakit tanaman, yang dapat menurunkan produktivitas dan kualitas hasil panen secara signifikan. Cabai dan tomat merupakan komoditas pertanian yang memiliki nilai ekonomi tinggi, namun rentan terhadap berbagai penyakit, seperti *Leaf Curl* pada cabai, serta *Late Blight* dan *Early Blight* pada tomat. Jika tidak terdeteksi secara dini, penyakit-penyakit ini dapat menyebabkan kerugian besar bagi petani [2].

Deteksi penyakit tanaman secara manual sering kali memerlukan keahlian khusus dan membutuhkan waktu yang lama, sehingga kurang efisien jika diterapkan dalam skala luas. Oleh karena itu, diperlukan sistem otomatis yang mampu mengidentifikasi penyakit tanaman dengan akurasi tinggi. Dalam beberapa tahun terakhir, *Convolutional Neural Networks* (CNN) telah menjadi salah satu pendekatan *deep learning* yang banyak digunakan dalam analisis citra, termasuk dalam deteksi penyakit tanaman [1] [3]. CNN mampu mengenali pola visual pada gambar tanaman untuk membedakan antara kondisi sehat dan kondisi yang terinfeksi penyakit.

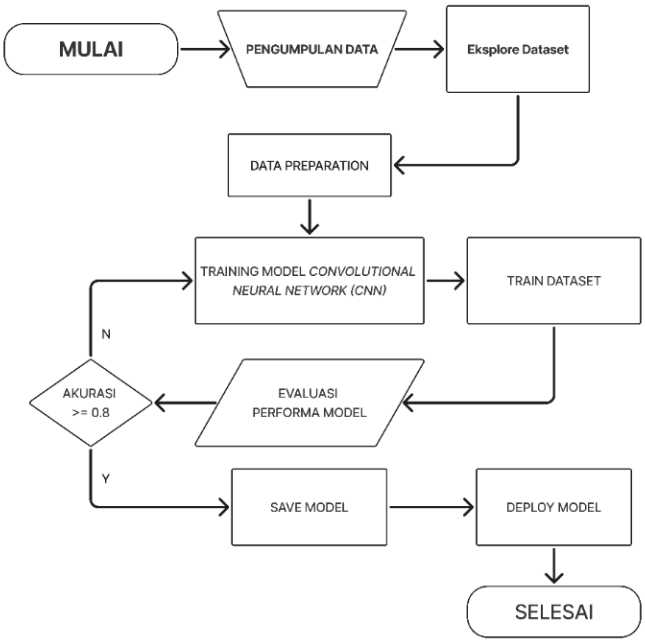
Implementasi CNN dalam bidang pertanian juga telah dilakukan oleh beberapa penelitian sebelumnya. Penelitian oleh Rasywir et al. (2020) mengembangkan sistem diagnosis penyakit kelapa sawit menggunakan Algoritma CNN dengan tujuan mempercepat proses identifikasi penyakit tanpa keterlibatan langsung tenaga ahli [4]. Model yang dilatih menggunakan citra dari Dinas Perkebunan Provinsi Jambi ini mampu mencapai akurasi rata-rata sebesar 87%, yang menunjukkan potensi besar CNN dalam membantu petani mengenali penyakit secara otomatis dan efisien. Selain itu, Sya'bani et al. (2022) berhasil menerapkan CNN untuk klasifikasi buah segar dan busuk (apel merah, pisang, dan jeruk mandarin), serta mengintegrasikan model ke dalam perangkat Android melalui TFLite, sehingga memberikan kemudahan akses dan implementasi secara mobile [5]. Penelitian lain oleh Lesmana et al. (2022) menunjukkan bahwa CNN juga efektif dalam mengidentifikasi penyakit pada daun kentang seperti *Early Blight* dan *Late Blight*, dengan tingkat akurasi validasi mencapai 99% dari 5400 citra yang digunakan [6]. Ketiga penelitian ini menunjukkan bahwa algoritma CNN sangat potensial untuk diterapkan dalam deteksi penyakit tanaman secara otomatis dan dapat menjadi solusi inovatif untuk permasalahan yang dihadapi petani di lapangan.

Penelitian ini mengembangkan aplikasi berbasis web yang tidak hanya berfokus pada satu jenis tanaman atau kondisi, tetapi mencakup dua komoditas sekaligus, yaitu cabai dan tomat, dengan masing-masing memiliki jenis penyakit yang berbeda (*Leaf Curl, Early Blight, Anthracnose, dan Late Blight*). Selain itu, aplikasi ini memungkinkan pengguna untuk mengunggah gambar tanaman secara langsung melalui antarmuka web, sehingga dapat digunakan oleh pengguna tanpa harus memasang aplikasi secara lokal.

Perbedaan lainnya terletak pada pengujian model dalam kondisi nyata. namun, berdasarkan hasil pengujian, akurasi deteksi penyakit pada aplikasi ini mencapai 75%, lebih rendah dibandingkan akurasi model saat pelatihan yang mencapai 91%. Disparitas ini kemungkinan disebabkan oleh faktor-faktor seperti perbedaan pencahayaan, kualitas gambar, sudut pengambilan gambar, serta distribusi data antara *dataset* pelatihan dan data uji. Pendekatan yang dapat diterapkan untuk meningkatkan akurasi dan mengurangi misklasifikasi antara lain adalah optimalisasi prapemrosesan data, teknik augmentasi gambar, serta penggunaan arsitektur model yang lebih kompleks. Dengan pengembangan lebih lanjut, diharapkan sistem ini dapat memberikan solusi yang lebih akurat dan efisien dalam mendeteksi penyakit tanaman cabai dan tomat di dunia nyata.

2. METODE PENELITIAN

Penelitian ini melakukan implementasi *Computer Vision* dalam mendeteksi penyakit tanaman melalui citra menggunakan Algoritma CNN. Algoritma ini dipilih karena kemampuannya melakukan ekstraksi fitur spasial dan tekstural secara otomatis tanpa rekayasa fitur manual yang sangat penting untuk mengenali beragam gejala penyakit pada sayuran seperti kentang, tomat, cabai, mentimun, dan kubis [7]. Selain itu, survei terhadap 100 publikasi terkini menyimpulkan bahwa Deep CNN (DCNN) yang dilatih pada dataset citra daun nyata mampu mendeteksi penyakit sejak tahap awal (*early detection*) dengan akurasi tertinggi, sehingga intervensi dapat dilakukan lebih cepat sebelum penyebaran penyakit meluas [8] [9]. Studi-studi tersebut juga menunjukkan bahwa CNN berskala baik dalam menangani kumpulan data besar seperti *PlantVillage* dengan metrik performa (akurasi, presisi, recall) yang konsisten tinggi [7]. Namun, kompleksitas komputasi, kebutuhan data pelatihan yang besar, dan potensi *overfitting* tetap menjadi tantangan utama. Berbagai pendekatan termasuk *augmentasi* data, regularisasi, dan *fine-tuning* arsitektur dipaparkan sebagai solusi untuk meningkatkan generalisasi dan mengurangi risiko tersebut [7] [8]. Dengan fondasi ini, dapat dikatakan bahwa CNN menjadi salah satu algoritma terbaik untuk kasus deteksi penyakit tanaman berbasis citra, baik dari segi akurasi klasifikasi maupun kemampuan deteksi dini. Penelitian ini bertujuan untuk melakukan implementasi computer vision dalam mendeteksi penyakit tanaman melalui citra menggunakan Algoritma CNN. Proses dan tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Pengumpulan Data (*Data Acquisition*)

pengumpulan data dalam penelitian ini dilakukan melalui dua metode utama, yaitu pemanfaatan dataset publik dan pengambilan gambar secara mandiri.

1. *Dataset* Publik (*PlantVillage - Kaggle*)
   Dataset ini digunakan sebagai sumber utama karena memiliki label yang jelas dan variasi gambar yang luas. Sebelum digunakan, dataset diproses dengan pengecekan kualitas, penyesuaian ukuran, dan pengelompokan ulang menjadi enam kelas utama, Cabai Antraknosa, Cabai Normal, Cabai Leaf Curl, Tomat Normal, Tomat Early Blight, dan Tomat Late Blight.

2. *Dataset* Mandiri
   Gambar diambil langsung menggunakan kamera dengan berbagai pencahayaan, sudut, dan latar belakang untuk meningkatkan performa model. Ini membantu mengidentifikasi tantangan seperti perbedaan kualitas gambar dan variasi lingkungan.

2.2. *Explore the Dataset*

Eksplorasi data dilakukan untuk mendapatkan wawasan lebih dalam mengenai karakteristik dataset yang digunakan [10]. Proses ini mencakup analisis distribusi kategori penyakit tanaman, jumlah sampel per kelas, serta pemeriksaan faktor-faktor yang dapat mempengaruhi performa model, seperti variasi pencahayaan, sudut pengambilan gambar, dan latar belakang gambar. Dengan memahami pola-pola ini, peneliti dapat mengidentifikasi potensi ketidakseimbangan data, menentukan langkah-langkah preprocessing yang diperlukan, serta menyesuaikan strategi augmentasi guna meningkatkan akurasi dan generalisasi model dalam mendeteksi penyakit tanaman.

2.3. Data *Preparation*

Data preparation berperan dalam memastikan bahwa data pelatihan memiliki kualitas yang baik dan bervariasi. Tahapan ini mencakup proses data *preprocessing* untuk menstandarkan input serta data augmentation untuk meningkatkan keragaman data [11]. Kedua proses ini saling melengkapi guna menghasilkan model yang akurat dan mampu melakukan generalisasi dengan baik.

1. Data *Preprocessing*
   Preprocessing menjadi tahap awal yang bertujuan menyiapkan gambar agar kompatibel dengan arsitektur model serta dapat diproses secara efisien. Pada penelitian ini dilakukan Resizing (Penyesuaian Ukuran), Seluruh gambar disesuaikan ke ukuran 256 × 256 piksel agar sesuai dengan spesifikasi input model. Proses ini dilakukan tanpa mengurangi kualitas gambar serta mempertahankan connected components, yaitu bagian gambar yang merepresentasikan objek utama. Penyesuaian ukuran penting untuk menjamin konsistensi format input selama pelatihan [12].

2. Data *Augmentation*
   Augmentasi data bertujuan untuk memperkaya variasi citra dalam data pelatihan tanpa harus menambah jumlah gambar secara manual. Teknik ini memungkinkan model belajar mengenali objek dalam berbagai kondisi yang berbeda, sehingga memperkuat kemampuan generalisasi. Beberapa teknik augmentasi yang digunakan dalam penelitian ini sebagai berikut:
   a. *Rotation* (Rotasi)
      Gambar diputar dalam berbagai sudut untuk mensimulasikan variasi orientasi objek. Teknik ini membantu model tetap akurat meskipun objek tidak selalu berada dalam posisi yang sama.

   b. *Flipping* (Pencerminan)
      Flipping dilakukan secara horizontal untuk menciptakan citra bayangan dari gambar asli. Teknik ini berguna dalam kasus di mana objek dapat muncul dengan arah yang berlawanan, sehingga model tidak menjadi bias terhadap arah tertentu.

   c. *Brightness and Contrast Adjustment* (Penyesuaian Kecerahan dan Kontras)
      Tingkat kecerahan dan kontras citra diubah untuk mengakomodasi berbagai kondisi pencahayaan. Penyesuaian ini membantu meningkatkan kejelasan objek serta mempertahankan detail penting dalam gambar [11].

2.4. Convolutional Neural Network (CNN)

CNN merupakan jenis arsitektur jaringan saraf tiruan atau turunan dari ilmu neural network. CNN telah merevolusi bidang ML dengan kemampuannya untuk secara otomatis mengekstraksi fitur-fitur yang relevan dari citra data. komponen utama CNN, termasuk lapisan-lapisan yang ada di dalamnya [13].

1. Lapisan Konvolusi (*Convolutional Layer*)

Digunakan untuk mengekstraksi fitur dari gambar dengan operasi konvolusi, menggantikan perkalian matriks yang biasa digunakan dalam jaringan saraf tiruan tradisional. Operasi konvolusi memungkinkan jaringan mempelajari hubungan antara input dan output dengan lebih efisien, sehingga dapat menangkap fitur penting dalam gambar [14]. Lapisan ini bekerja dengan menerapkan sejumlah filter konvolusi (kernel) yang digeser di seluruh citra input. Proses ini menghasilkan feature map, yang menyoroti keberadaan fitur tertentu seperti tepi, garis, tekstur, atau pola yang lebih kompleks. Setiap filter memiliki bobot (weight) yang dapat dipelajari, yang diperbarui selama pelatihan untuk mengoptimalkan pengenalan pola [10].Persamaan Output neuron dari convolutional layer dapat ditulis pada persamaan 1 dan 2.

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} . w_{u,v,k',k} \tag{1}$$

$$\text{dengan} = \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \tag{2}$$

2. Lapisan Aktivasi (*Activation Layer*)

Setelah lapisan konvolusi, fungsi aktivasi non-linear seperti Rectified Linear Unit (ReLU) diterapkan untuk memperkenalkan non-linearitas dan membantu jaringan mempelajari hubungan kompleks antara fitur. ReLU mengubah nilai negatif menjadi nol, sementara nilai positif tetap, sehingga memperkuat fitur penting dan mempercepat proses pembelajaran. Keunggulan utama ReLU adalah beban komputasi yang lebih rendah dibandingkan fungsi aktivasi lainnya [13].

$$f(x) = \max(0, x) \tag{3}$$

3. Lapisan *Pooling*

Lapisan *pooling* digunakan untuk mengurangi dimensi spasial dari fitur-fitur yang ditemukan oleh lapisan konvolusi. Metode pooling yang umum adalah max pooling. Pooling membantu mengurangi jumlah parameter dalam jaringan dan menghasilkan fitur representasi yang lebih tahan terhadap pergeseran dan deformasi kecil pada citra. Ini juga membantu dalam menghindari overfitting dan membuat model lebih umum [15].

4. Lapisan *Fully Connected*

Menghubungkan semua neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya, seperti pada jaringan saraf tiruan biasa. Lapisan ini bertugas menggabungkan fitur yang diekstraksi untuk mempelajari hubungan kompleks dan melakukan klasifikasi. Pada tahap akhir, fungsi aktivasi *Softmax* dapat digunakan untuk mengubah output menjadi distribusi probabilitas, sehingga model dapat menentukan kelas dengan probabilitas tertinggi. Softmax menghitung nilai probabilitas setiap kelas dengan persamaan 4 [16].

$$P(y_i) = \frac{e^{zi}}{\sum_j e^{zi}} \quad (4)$$

2.5. Evaluasi Performa Model

Evaluasi model pada CNN adalah proses untuk menilai kinerja dan akurasi model dalam mengklasifikasi objek atau matriks. Dilakukan melalui metode evaluation matrix yang menggunakan tabel Confusion Matrix. Confusion Matrix yakni metode visualisasi untuk hasil algoritma klasifikasi. Dalam kata lain ini merupakan tabel yang memecah jumlah instansi kebenaran dasar dari kelas tertentu terhadap jumlah instansi kelas yang diprediksi [17] [3]. Adapun evaluasi matriks yang dapat digunakan untuk menghitung nilai dari *Accuracy, Precision, Recall* dan *F-1 Score* dari performa model *Machine Learning* yang telah dilatih [18].

1. *Accuracy*
   Akurasi dari pengklasifikasi diukur dengan metrik ini. Jumlah data yang diklasifikasikan dengan benar dibagi oleh total jumlah data untuk menghitung akurasi. Rumus untuk menghitung akurasi adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2. *Precision*
   Presisi menunjukkan seberapa banyak data yang diprediksi sebagai positif yang diprediksi dengan benar. Dengan kata lain, presisi yang tinggi berarti lebih sedikit false positives. Rumus untuk menghitung presisi adalah sebagai berikut :

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

3. *Recall*
   Recall adalah metrik untuk menentukan kelengkapan dari pengklasifikasi. *Recall* yang lebih tinggi menunjukkan false negatives yang lebih rendah, sementara *recall* yang lebih rendah menunjukkan false negatives yang lebih tinggi. Presisi sering kali menurun dengan peningkatan recall. Rumus untuk menghitung *recall* adalah sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

4. *F1-Score*

Untuk mendapatkan *F1-Score*, hasil kali dari recall dan precision dibagi oleh jumlah dari *recall* dan *precision*. Rumus untuk menghitung *F1-Score*
adalah sebagai berikut:

$$F1 - Score = 2\,x\,\frac{Recall\ x\ Precision}{Recall+Precision} \quad (8)$$

2.6. *Deployment*

Deployment adalah proses menerapkan model yang telah dilatih ke dalam aplikasi agar dapat digunakan oleh pengguna. Dalam penelitian ini, model CNN untuk deteksi penyakit tanaman diterapkan dalam aplikasi berbasis web dengan menggunakan *Flask* sebagai *backend*. *Flask* berfungsi untuk menangani permintaan dari pengguna dan menjalankan model AI dalam menghasilkan prediksi [19]. *Python* digunakan sebagai bahasa pemrograman utama untuk membangun model, mengolah data, serta mengintegrasikan model dengan sistem [20]. Sementara itu, *JavaScript* berperan dalam meningkatkan interaktivitas pada *frontend*, seperti menampilkan hasil prediksi secara dinamis di halaman web, sehingga pengalaman pengguna menjadi lebih responsif [21].

3. HASIL DAN PEMBAHASAN

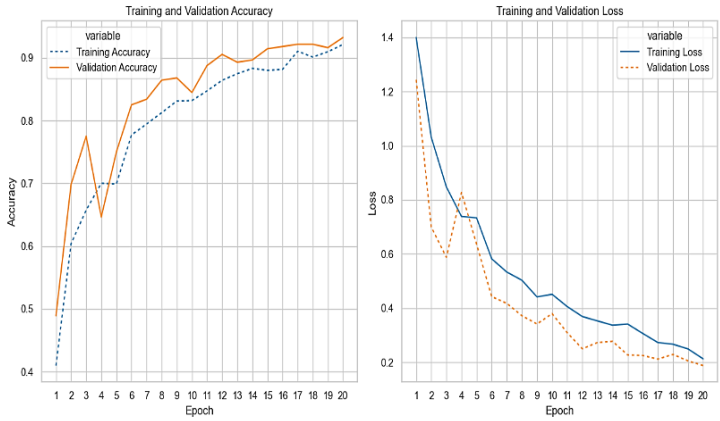
Pengumpulan data dilakukan dengan menggunakan dataset publik *Plant Village* dari *kaggle* dan *dataset* yang diambil secara mandiri menggunakan kamera gawai. *Dataset* yang digunakan dalam penelitian ini berjumlah 3.734 data terbagi menjadi 6 (enam) kelas dengan data yang terdiri dari citra daun tomat dan buah cabai dengan kondisi normal dan kondisi terserang penyakit, dengan distribusi data Pada Tabel 1 :

Tabel 1. Distribusi data

Kelas	Jumlah Data
Cabai Normal	600

Kelas	Jumlah Data
Cabai Antraknosa	620
Cabai *leaf curl*	621
Tomat Normal	670
Tomat *Early Blight*	612
Tomat *Late Blight*	611

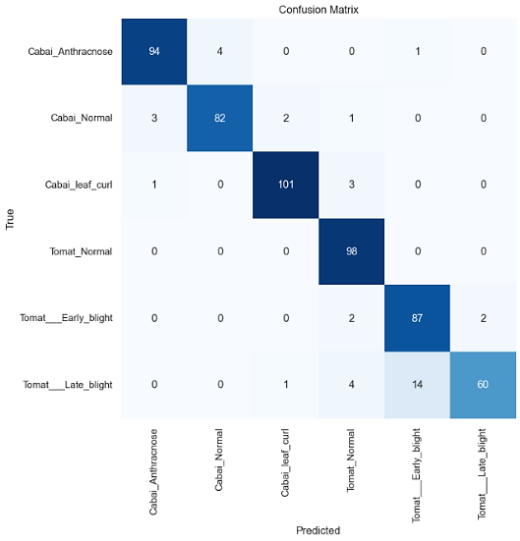
Dataset tersebut dibagi menjadi 3 bagian (*split dataset*) yakni *Train Dataset, Validation Dataset*, dan *Test Dataset* dengan rasio 70:15:15 dan juga dilakukan augmentasi data dengan teknik *flipping*, dan rotasi, *contrast* dan *brightness.*



Gambar 2. *Training Accuracy* dan *Loss*

Pada Gambar 2 merupakan Grafik akurasi yang menunjukkan tren positif di mana akurasi untuk data *training* dan validasi meningkat seiring bertambahnya *epoch.* Peningkatan ini mencerminkan bahwa model mampu mempelajari pola dalam data secara efektif. Pada tahap akhir *training*, terutama pada *epoch* ke-18 hingga ke-20 berada di kisaran 91%, akurasi training dan validasi berada pada tingkat yang hampir sama. Hal ini mengindikasikan bahwa model telah mencapai titik konvergensi, di mana kemampuan model untuk mempelajari data training dan generalisasi terhadap data validasi berada dalam keseimbangan. Menariknya, pada beberapa epoch awal, akurasi validasi terlihat lebih tinggi dibandingkan akurasi training. Ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data baru.

Grafik *loss* memperlihatkan penurunan nilai *loss* untuk data *training* dan validasi seiring bertambahnya *epoch*. Penurunan ini menunjukkan bahwa model berhasil meminimalkan error dengan baik selama proses *training*. Pada sebagian besar *epoch, loss* validasi sedikit lebih tinggi dibandingkan *loss training*, yang merupakan hal umum karena model diuji pada data yang belum pernah dilihat sebelumnya. Selain itu, pada *epoch-epoch* akhir, nilai loss validasi cenderung stabil dan tidak mengalami peningkatan yang signifikan. Pada Gambar 3 merupakan tampilan Confusion Matrix.



Gambar 3. Confusion Matrix

Pada Tabel 2 menunjukkan Nilai pada diagonal utama (dari kiri atas ke kanan bawah) menunjukkan jumlah deteksi yang benar untuk setiap kelas.

Tabel 2. Deteksi benar *dataset test*

Kelas Aktual	Deteksi Benar
Cabai Antraknosa	94
Cabai Normal	82
Cabai *Leaf Curl*	101
Tomat Normal	98
Tomat *Early Blight*	87
Tomat *Late Blight*	60

Tabel 3. Deteksi salah pada *dataset test*

Kelas Aktual	Kesalahan Kelas Deteksi	Jumlah Kesalahan	Keterangan
Cabai Antraknosa	Cabai Normal	4	Beberapa sampel salah diklasifikasikan sebagai Cabai_Normal.
Cabai Antraknosa	Tomat *Early Blight*	1	Satu sampel salah diklasifikasikan sebagai Tomat *Early Blight*
Cabai Normal	Cabai Antraknosa	3	Sebagian kecil prediksi salah diarahkan ke Cabai_*Anthracnose*.
Cabai Normal	Cabai *Leaf Curl*	2	Sebagian kecil kesalahan diarahkan ke Cabai *leaf curl*.
Cabai Normal	Tomat Normal	1	Satu kesalahan deteksi diarahkan ke Tomat Normal
Cabai *Leaf Curl*	Cabai Antraknosa	1	Kesalahan kecil kelas diarahkan ke Cabai *Anthracnose*
Cabai *Leaf Curl*	Tomat Normal	3	Beberapa kesalahan diarahkan ke Tomat Normal
Tomat *Early Blight*	Tomat Normal	2	kesalahan ke kelas Tomat Normal yang berbeda.
Tomat *Early Blight*	Tomat *Late Blight*	2	Kesalahan cukup besar ke kelas yang mirip secara visual.
Tomat *Late Blight*	Cabai *Leaf Curl*	1	Kesalahan diarahkan ke kelas Cabai *Leaf Curl*
Tomat *Late Blight*	Tomat Normal	4	Sebagian kecil Kesalahan diarahkan ke Tomat Normal
Tomat *Late Blight*	Tomat *Early Blight*	14	Kesalahan terbesar antara dua kelas yang mirip secara visual.

Kesalahan klasifikasi yang terjadi menunjukkan bahwa model mengalami kesulitan dalam membedakan beberapa kelas, terutama yang memiliki kemiripan visual. Pada tanaman cabai, kesalahan klasifikasi paling banyak terjadi pada Cabai Normal, di mana 3 sampel salah diklasifikasikan sebagai Cabai Antraknosa, 2 sampel sebagai Cabai Leaf Curl, dan 1 sampel sebagai Tomat Normal. Hal ini menunjukkan bahwa model mengalami kesulitan dalam membedakan kondisi cabai yang sehat dan yang mengalami penyakit tertentu, terutama yang memiliki gejala serupa seperti perubahan warna dan tekstur daun. Selain itu, Cabai Antraknosa juga mengalami beberapa kesalahan, dengan 4 sampel salah diklasifikasikan sebagai Cabai Normal dan 1 sampel salah terdeteksi sebagai Tomat Early Blight, yang menandakan adanya fitur visual yang tumpang tindih antara kelas tersebut. Cabai Leaf Curl juga mengalami kesalahan klasifikasi, di mana 1 sampel salah diklasifikasikan sebagai Cabai Antraknosa dan 3 sampel salah dikenali sebagai Tomat Normal, yang menunjukkan bahwa model kurang mampu membedakan ciri khas dari kondisi daun yang menggulung dengan kondisi normal tanaman tomat.

Pada tanaman tomat, kesalahan terbesar terjadi antara Tomat Early Blight dan Tomat Late Blight, di mana masing-masing memiliki kesalahan silang dengan 2 sampel salah diklasifikasikan ke kelas yang berlawanan. Hal ini menunjukkan bahwa gejala visual kedua penyakit ini cukup mirip, seperti bercak dan perubahan warna daun, yang membuat model kesulitan dalam membedakannya. Selain itu, Tomat Early Blight juga memiliki 2 kesalahan klasifikasi ke Tomat Normal, yang menunjukkan bahwa beberapa sampel mungkin memiliki gejala ringan sehingga terdeteksi sebagai kondisi sehat. Pada Tomat Late Blight, terdapat 1 sampel yang salah diklasifikasikan sebagai Cabai Leaf Curl, yang bisa terjadi akibat pola visual tertentu yang menyerupai gejala penyakit pada cabai. Selain itu, Tomat Late Blight juga memiliki kesalahan yang cukup signifikan, dengan 4 sampel salah diklasifikasikan sebagai Tomat Normal dan 14 sampel salah dikenali sebagai Tomat Early Blight, yang memperkuat indikasi bahwa model mengalami kesulitan dalam membedakan penyakit yang memiliki karakteristik visual serupa.Data yang diperoleh dari Confusion Matrix digunakan untuk mendapatkan nilai dari Precission, Recall, dan F1-Score dari tiap kelas pada Tabel 4:

Tabel 4. *Precision, Recall, F1-Score*

Kelas	*Precision*	*Recall*	*F1 - Score*
Cabai Antraknosa	0.9591	0.9494	0.9543
Cabai Normal	0.9534	0.9318	0.9425
Cabai *Leaf Curl*	0.9711	0.9619	0.9665
Tomat Normal	0.9074	0.95600	0.9514
Tomat *Early Blight*	0.8529	0.9560	0.9015
Tomat *Late Blight*	0.9677	0.7594	0.8510

Berdasarkan hasil evaluasi metrik *Precision*, *Recall*, dan *F1-Score*, model secara keseluruhan menunjukkan performa yang baik dalam mendeteksi kondisi penyakit pada tanaman cabai dan tomat, meski ada beberapa kelas yang perlu peningkatan. Pada kelas Cabai Antraknosa, model berhasil menyeimbangkan keakuratan dan sensitivitas Precision 0.9591 dan Recall 0.9494 sehingga F1-Score yang dihasilkan mencapai 0.9543. Ini menandakan bahwa model jarang melakukan kesalahan dalam memprediksi maupun melewatkan sampel yang sebenarnya. Demikian pula, kelas Cabai Leaf Curl tampil sangat baik dengan Precision 0.9711 dan Recall 0.9619, menghasilkan F1-Score 0.9665, yang menunjukkan keandalan tinggi dalam mengenali gejala keriting daun cabai.

Sementara itu, kelas Cabai Normal memiliki F1-Score yang sedikit lebih rendah, yakni 0.9425, karena Recall-nya 0.9318 masih di bawah kelas cabai lainnya. Hal ini menunjukkan bahwa model terkadang kesulitan membedakan cabai normal dengan kondisi lain mungkin disebabkan kesamaan tekstur atau warna daun. Pada kelompok tomat, kelas Tomat Normal memiliki Recall tinggi 0.9560 yang berarti hampir semua sampel benar-benar terdeteksi, namun Precision yang lebih rendah 0.9074 mengindikasikan adanya false positives; dengan demikian F1-Score-nya mencapai 0.9514, cukup kuat tapi masih dapat disempurnakan untuk mengurangi alarm palsu.

Untuk penyakit Tomat Early Blight, model juga menampilkan Recall 0.9560 sehingga sensitif terhadap keberadaan gejala, tetapi Precision yang lebih rendah 0.8529 mengisyaratkan bahwa banyak prediksi yang keliru. F1-Score sebesar 0.9015 menggaris bawahi bahwa model cukup baik, namun perlu peningkatan khususnya pada akurasi klasifikasi. Kondisi serupa terlihat pada Tomat Late Blight, meski Precision sangat tinggi 0.9677, nilai Recall yang hanya 0.7594 menunjukkan banyak kasus terlewatkan, sehingga F1-Scorenya berada di 0.8510. Ketidakseimbangan ini menandakan bahwa model sangat yakin ketika memprediksi Late Blight, tetapi belum cukup sensitif untuk menangkap semua sampel. Secara umum, fokus perbaikan sebaiknya diarahkan pada peningkatan Recall untuk kelas-kelas yang masih rendah, terutama Tomat Late Blight dan Cabai Normal, serta pengayaan data agar distribusi fitur antar kelas lebih merata

Gambar 4. Tampilan utama *website one-page*

Setelah proses deployment, dilakukan pengujian kembali untuk mengevaluasi performa aplikasi dalam mendeteksi penyakit tanaman cabai dan tomat. Pengujian ini bertujuan untuk memastikan bahwa model bekerja dengan baik dalam lingkungan nyata, menguji respons aplikasi terhadap berbagai kondisi input dapat dilihat pada Tabel 5.

Tabel 5. Pengujian deteksi aplikasi.

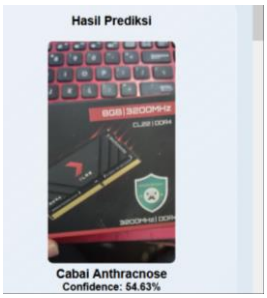
Kelas Aktual	Deteksi Benar	Deteksi Salah
Cabai Antraknosa	10	0
Cabai Normal	9	1
Cabai *Leaf Curl*	5	5
Tomat Normal	6	4
Tomat *Early Blight*	7	3
Tomat *Late Blight*	8	2

Hasil pengujian ini diperoleh dari percobaan deteksi menggunakan data nyata yang diambil langsung melalui kamera. Data ini mencakup enam kelas, yaitu Cabai Antraknosa, Cabai Normal, Cabai Leaf Curl, Tomat Normal, Tomat Early Blight, dan Tomat Late Blight. Setiap kelas memiliki jumlah sampel tertentu yang digunakan untuk mengukur kemampuan model dalam mengklasifikasikan kondisi tanaman dengan benar atau salah.

$$Akurasi = \frac{Jumlah\ deteksi\ benar}{Total\ sampel} \times 100\%$$

$$Akurasi = \frac{(10+9+5+6+7+8)}{(10)+(9+1)+(5+5)+(6+4)+(7+3)+(8+2)} = \frac{42}{60} \times 100\% = 75\%$$

Berdasarkan pengujian manual yang dilakukan setelah implementasi model ke dalam aplikasi berbasis web, akurasi deteksi penyakit tanaman cabai dan tomat mencapai 75%. Hasil ini menunjukkan adanya perbedaan dengan akurasi yang diperoleh selama pelatihan model, yang sebelumnya mencapai 91%. Perbedaan ini bisa disebabkan oleh beberapa faktor, seperti kondisi pencahayaan, kualitas gambar yang diambil melalui kamera, sudut pengambilan gambar, serta kemungkinan adanya perbedaan distribusi data, dapat dilihat pada Gambar 5.



Gambar 5. Gagal deteksi objek non kategori

Gambar 6. Visualisasi feature map layers

Gambar 6 merupakan visualisasi *feature* map yang menunjukkan bagaimana model CNN memberikan perhatian pada area tertentu dalam gambar yang tidak termasuk dalam kategori yang telah dilatih, namun tetap diklasifikasikan sebagai cabai antraknosa. Dari pola aktivasi yang terlihat, model tampaknya mengidentifikasi bagian dengan tekstur atau kontras tertentu yang menyerupai karakteristik penyakit pada cabai, sehingga terjadi salah klasifikasi. Aktivasi yang kuat pada area tertentu menunjukkan bahwa model memiliki pola spesifik yang dijadikan acuan untuk mengenali cabai antraknosa, yang mungkin juga ditemukan dalam gambar input ini. Kesalahan ini bisa disebabkan oleh keterbatasan dataset pelatihan yang tidak memiliki cukup variasi, sehingga model kurang mampu membedakan objek yang tidak relevan. Selain itu, model tidak dilatih untuk mengenali kategori di luar kelas (*out of distribution*), sehingga setiap input yang diberikan tetap akan dipetakan ke salah satu kelas yang ada.

4. KESIMPULAN

Penelitian ini mengimplementasikan teknologi *Computer Vision* pada aplikasi berbasis web untuk mendeteksi penyakit pada tanaman cabai dan tomat menggunakan Algoritma CNN. Aplikasi ini memungkinkan pengguna untuk mengunggah gambar tanaman melalui antarmuka berbasis HTML, CSS, *Python, JavaScript*, dan *Flask*, yang kemudian diproses oleh model CNN untuk mengidentifikasi jenis penyakit tanaman. Model yang dibangun mampu mengklasifikasikan enam kategori, yaitu cabai normal, cabai leaf curl, cabai antraknosa, tomat normal, tomat early blight, dan tomat late blight. Hasil evaluasi menggunakan dataset uji menunjukkan bahwa model memiliki performa yang cukup baik dengan akurasi 91% pada pelatihan. Namun, dalam tahap implementasi pada data nyata melalui kamera secara real-time, model mengalami sedikit penurunan performa dengan akurasi sebesar 75%.

Hasil pengujian lebih lanjut menunjukkan bahwa kesalahan prediksi cenderung terjadi pada kelas-kelas yang memiliki karakteristik visual yang mirip, seperti tomat early blight dan tomat late blight, serta cabai normal yang sering diklasifikasikan sebagai cabai leaf curl. Selain itu, model mengalami kesulitan dalam menangani gambar yang tidak termasuk dalam kategori pelatihan, sehingga beberapa objek di luar dataset dapat salah diklasifikasikan. Analisis menggunakan Confusion Matrix menunjukkan adanya misclassifications yang signifikan pada beberapa kelas tertentu. Visualisasi feature map juga memperlihatkan bahwa model lebih banyak fokus pada fitur tertentu dalam gambar, tetapi terkadang area yang diperhatikan kurang relevan dengan karakteristik penyakit yang sebenarnya. Hal ini menunjukkan bahwa meskipun model sudah cukup baik dalam mendeteksi penyakit tanaman, masih diperlukan peningkatan lebih lanjut agar model lebih robust terhadap data dunia nyata.

REFERENSI

[1] H. Tian, T. Wang, Y. Liu, X. Qiao and Y. Li, "Computer Vision Technology in Agricultural Automation," Information Processing In Agriculture vol.7, pp. 1-19, 2020.

[2] C. Wati, A. T. Karenina, R. Y. Nirwanto, I. Nurcahya, D. Meilani, D. Astuti, D. Septiarini, S. R. F. Purba, E. P. Ramdan and D. Nurul, Hama dan Penyakit Tanaman, bogor: Yayasan Kita Menulis, 2021.

[3] A. Taner, Y. B. Oztekin and H. Duran, "Performence Analysis of Deep Learning CNN Models for Variety Classification in Hazelnut," Sustainability, 2021.

[4] E. Rasywir and R. Sinaga, "Analisis dan Implementasi Diagnosa Penyakit Sawit Dengan Metode Convolutional Neural Network," Paradigma - Jurnal Informatika dan Komputer Vo.22 No.22, 2020.

[5] D. R. Sya'bani, A. Hamzah and E. Susanti, "Klasifikasi Buah Segar Dan Busuk Menggunakan Algoritma Convolutional Neural Network Dengan Tflite Sebagai Media Penerapan Model Machine Learning," 2022. [Online]. Available: https://ejournal.akprind.ac.id/index.php/snast/article/download/4180/2976/6861?.

[6] A. M. Lesmana, R. P. Fadhillah and C. Rozikin, "Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN)," Jurnal Sains dan Informatika Vol.8 No.1, 2022.

[7] M. M. u. Rehman, J. Liu, A. Nijabat, M. Faheem, W. Wang and S. Zhao, "Leveraging Convolutional Neural Networks for Disease Detection in Vegetables: A Comprehensive Review," Agronomy, vol. 14, 2024.

[8] B. Tugrul , E. Elfatimi and R. Eryigit , "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review," agriculture, vol. 12, 2022.

[9] A. Y. Ashurov, M. S. A. M. Al-Gaashani, N. A. Samee, R. Alkanhel, G. Atteia, H. A. Abdallah and M. S. A. Muthanna, "Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections," Frontiers in Plant Science, vol. 15, 2025.

[10] A. Geron, Hands-On Machine Learning With Scikit-Learn, Keras & TensorFlow, Canada: O'Reilly Media, 2019.

[11] T. Mulia, M. Kallista and P. D. Wibawa, "Preprocessing Gambar Sampah untuk Sistem Pemilah Sampah Otomatis menggunakan Roboflow," e-Proceeding of Engineering , vol. 11, pp. 6743-6247, 2024.

[12] A. Dubey, A. Lazarus and D. Mangal, "Handwritten Digit Recognition using Image Preprocessing and CNN," IJSRCSEIT, 2020.

[13] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Djuaili, Y. Duan, O. Al-Shama, S. J. and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, 2021.

[14] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate,dan Mixup," Jurnal Teknik Informatika dan Sistem Informasi, 2020.

[15] X. Zhao, L. Wang, Y. Zhang , X. Han, M. Deveci and M. Parmar, "A Review of Convolutional Neural Networks in computer Vision," Artificial Intelligence Review, vol. 57, 2024.

[16] Y. N. Fuadah, I. D. Ubaidullah, N. Ibrahim, F. F. Taliningsing, N. K. SY and M. A. Pramuditho, "Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma," ELKOMIKA, vol. 10, pp. 728-741, 2022.

[17] . J. Murel Ph.D. and E. Kavlakoglu, "What Is Confusion a Matrix," 19 01 2024. [Online]. Available: https://www.ibm.com/topics/confusion-matrix.

[18] A. Rahman, Brain Tumor Detection and Classification by Using CNN, Finland: UEF eRepository, 2024.

[19] A. Amanzadi and M. Karim, Comparison of Machine Learning Models Used for Swedish Text Classification in Chat Messaging, Stockholm: DiVa, 2022.

[20] S. Raschka, "Machine Learning in Python: Main Developments and Technology Trends In Data Science, Machine Learning and Artificial Intelligence," Information, 2020.

[21] Y. Supriadi, Semua Bisa Menjadi Programmer Javascript & Node.JS, Jakarta: PT Elex Media Komputindo, 2020.