

Klasifikasi Hama Pada Daun Sawi Menggunakan Convolutional Neural Network (CNN) Dengan Algoritma Xception dan Optimasi Adam

Saiful Bahri¹, Andi Sunyoto², Mei P. Kurniawan³

1,2,3 Magister Teknik Informatika
Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

Article Info

Article history:

Diterima 10 September 2024
Direvisi 19 September 2024
Diterbitkan 4 Oktober 2024

Keywords:

Convolutional Neural Network
Image Processing
Deep Learning
Klasifikasi
Hama Daun Sawi

ABSTRACT

Penelitian ini mengusulkan penggunaan Convolutional Neural Network (CNN) dengan arsitektur Xception dan optimasi Adam untuk klasifikasi hama pada daun sawi. Sawi (*Brassica rapa*) adalah sayuran yang populer di Indonesia, tetapi serangan hama seperti ulat grayak dapat merusak kualitas dan kuantitas hasil panen. Identifikasi hama yang akurat sangat penting untuk pengendalian yang efektif, namun metode konvensional yang menggunakan pengamatan visual sering kali kurang efisien dan rentan terhadap kesalahan manusia. Untuk mengatasi masalah ini, penelitian ini menerapkan teknik deep learning untuk meningkatkan akurasi identifikasi hama secara otomatis. Dataset yang digunakan dalam penelitian ini terdiri dari 1.000 gambar daun sawi, yang dikategorikan menjadi dua kelas: daun sawi dengan hama dan daun sawi tanpa hama. Dataset diperoleh dari situs publik Kaggle dan melalui tahap preprocessing untuk memastikan konsistensi data. Preprocessing termasuk perubahan ukuran gambar menjadi 512 x 512 piksel dan normalisasi nilai piksel untuk mempercepat waktu komputasi. Setelah itu, augmentasi data dilakukan untuk meningkatkan variasi data latih guna mengurangi risiko overfitting. Model CNN dengan arsitektur Xception dipilih karena keunggulannya dalam ekstraksi fitur yang mendalam, sementara algoritma optimasi Adam digunakan untuk mempercepat konvergensi model. Selama pelatihan, model menggunakan ukuran batch 64, 100 epoch, dan dropout sebesar 0,5 untuk mengurangi overfitting. Hasil pelatihan menunjukkan bahwa model mencapai akurasi yang sangat tinggi, baik dalam data pelatihan maupun validasi, yaitu sebesar 99,00%. Evaluasi kinerja model dilakukan menggunakan confusion matrix yang menunjukkan hasil klasifikasi yang sangat akurat. Penelitian ini memberikan kontribusi yang signifikan terhadap pengembangan teknologi pengendalian hama pada pertanian, khususnya dalam klasifikasi hama daun sawi. Hasil yang diperoleh menunjukkan bahwa penggunaan CNN dengan arsitektur Xception dan optimasi Adam dapat secara signifikan meningkatkan efisiensi dan akurasi identifikasi hama. Teknologi ini berpotensi membantu petani dalam mendeteksi serangan hama lebih dini, sehingga dapat meningkatkan produktivitas pertanian dan kualitas hasil panen. Implementasi teknologi ini juga membuka peluang untuk penggunaan lebih luas dalam klasifikasi hama pada tanaman lain.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Saiful Bahri
Universitas Amikom Yogyakarta,
Email: saifulbahri@students.amikom.ac.id

1. PENDAHULUAN

Sawi (*Brassica rapa*) adalah sayuran populer di Indonesia. Serangan hama sering menghambat produktivitasnya, mengurangi kualitas dan kuantitas panen. Identifikasi hama secara akurat penting untuk pengendalian efektif. Metode konvensional dengan pengamatan visual rentan terhadap kesalahan manusia,

sehingga perlunya teknologi untuk meningkatkan efisiensi dalam pengendalian hama. Namun metode konvensional yang melibatkan pengamatan visual sering kali kurang efisien dan rentan terhadap kesalahan manusia [1].

Dengan teknologi terbaru, metode pemrosesan citra dan machine learning digunakan untuk meningkatkan akurasi dan efisiensi dalam identifikasi hama di pertanian. *Convolutional Neural Network (CNN)* telah berhasil dalam tugas identifikasi penyakit dan hama tanaman [2]. Namun, algoritma pembelajaran harus lebih dioptimalkan untuk meningkatkan kinerja model *CNN* dalam klasifikasi hama daun sawi. Dua teknik yang dapat digunakan untuk meningkatkan akurasi dan efisiensi model *CNN* adalah Algoritma *Xception*, yang merupakan teknik pemrosesan citra yang inovatif, dapat membantu dalam ekstraksi fitur yang lebih mendalam [3]. Sementara itu, optimasi Adam menawarkan solusi adaptif yang dapat mempercepat konvergensi model dan mengurangi risiko *overfitting*[4].

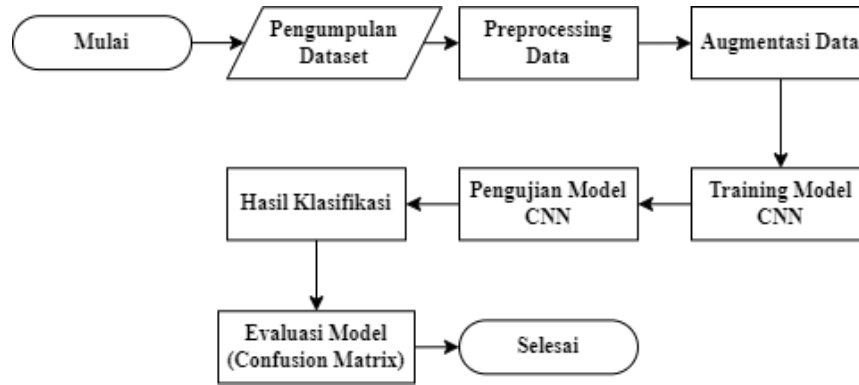
Salah satu sistem identifikasi hama pada daun sawi adalah menggunakan model *machine learning*. Penelitian tentang identifikasi penyakit pada daun sawi telah dilakukan oleh beberapa peneliti. Misalnya, Hridoy dkk menggunakan metode *CNN* dan membandingkan lima model, yaitu *MPNet* dan empat model *CNN* yang sudah terlatih sebelumnya, yaitu *MobileNetV2*, *DenseNet121*, *VGG19*, dan *ResNet50*. *MPNet* mampu mendeteksi penyakit tanaman sawi dengan akurasi terbaik sebesar 97.11%, sementara *MobileNetV2* mencapai akurasi 92.83%, *DenseNet121* sebesar 91.37%, *VGG19* sebesar 90.24%, dan *ResNet50* sebesar 88.02% [5]. Selain itu, Kukreja dkk menggunakan dua jenis model *Deep Learning (DL)*, yaitu *CNN* dan *Long Short-Term Memory (LSTM)*, untuk mendeteksi penyakit layu pada tanaman sawi yang disebabkan oleh jamur *Plasmopara brassicae* di India, dan mendapatkan akurasi sebesar 92.36% [6]. Sharma dkk menggunakan dua model *CNN* untuk mendeteksi penyakit pada tanaman sawi dengan tingkat keparahan berbeda. Model pertama, klasifikasi biner, memiliki akurasi sebesar 95.66%, sedangkan model kedua, klasifikasi ganda, menghasilkan akurasi sebesar 96.66% [6].

CNN juga digunakan oleh petani yang kesulitan membedakan atau mengklasifikasikan berbagai macam sawi, seperti sawi pakcoy, sawi putih, dan sawi caisin. Kurniadi dkk melaporkan bahwa akurasi yang didapatkan mencapai 84%, recall 80%, dan presisi 89% [7]. Singh dkk juga membandingkan akurasi *CNN* dengan model klasifikasi *Random Forest*, dan menemukan bahwa *CNN* dengan optimasi algoritma Adam memiliki akurasi sebesar 97.26%, jauh lebih baik dibandingkan dengan *Random Forest* yang hanya memiliki akurasi sebesar 73% [8]. Terakhir, Rasyid dan Wisudawati membahas tentang mengklasifikasi hama ulat pada citra daun sawi dengan model arsitektur *Xception* berbasis *Convolutional Neural Network*. Dataset daun sawi yang digunakan berjumlah 500 citra daun sawi tanpa hama dan 500 citra daun sawi yang terdapat hama ulat. Hasil dari klasifikasi dengan model arsitektur *Xception* dengan uji coba pada pelatihan data dengan 600 data latih, 200 data uji dan 200 data validasi menghasilkan nilai akurasi sebesar 96%, nilai sensitivitas 96% dan nilai spesifisitas 97% [9].

Penelitian sebelumnya yang menggunakan arsitektur *Xception* untuk klasifikasi hama ulat pada daun sawi berhasil mencapai akurasi sebesar 96%, dengan nilai sensitivitas 96% dan spesifisitas 97% [9]. Meskipun hasil ini sudah cukup baik, ada peluang untuk meningkatkan performa model dengan menggunakan optimasi yang lebih canggih dan dataset yang lebih beragam. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan performa klasifikasi hama pada citra tanaman dengan menggunakan *CNN* dengan arsitektur *Xception* yang dioptimasi menggunakan algoritma Adam optimizer dan menggunakan dataset yang berbeda dari penelitian sebelumnya. Dengan adanya perbaikan ini, diharapkan model yang dikembangkan dapat memberikan hasil klasifikasi yang lebih akurat dan efisien, sehingga dapat membantu petani dalam mendeteksi serangan hama lebih dini dan tepat.

2. METODE

Penelitian ini melibatkan beberapa langkah untuk mengumpulkan data, menganalisis data, dan mencapai tujuan. Untuk memastikan bahwa penelitian dilakukan secara sistematis dan dapat diandalkan, setiap tahap sangat penting. Hasil klasifikasi hama daun sawi menggunakan metode *Convolutional Neural Network* harus memenuhi ekspektasi [10]. Berikut adalah tahapan penelitian pada gambar 1.



Gambar 1. Tahapan Penelitian

2.1. Pengumpulan Dataset

Pada tahap ini, menggunakan dataset daun sawi dengan 2 kategori yaitu Daun Sawi Ada Hama dan Daun Sawi Tanpa Hama [11]. dataset dari penelitian ini dikumpulkan dari situs publik *Kaggle*. Sehingga sangat relevan untuk penelitian yang bertujuan mengembangkan model pembelajaran mesin dalam klasifikasi hama tanaman. Dengan menggunakan dataset ini, peneliti dapat melatih model untuk mendeteksi dan mengidentifikasi hama pada daun sawi secara akurat, yang diharapkan dapat membantu dalam pengendalian hama tanaman dan meningkatkan produktivitas pertanian.

Tabel 1. Kategori Daun Sawi

No	Kategori	Jumlah
1	Daun Sawi Ada Hama	500
2	Daun Sawi Tanpa Hama	500

Pada Tabel 1 di atas menunjukkan kategori daun sawi dalam Caisim beserta jumlah gambar yang termasuk dalam dua kategori.

1. Daun Sawi Ada Hama: Terdapat 500 gambar daun sawi yang dikategorikan sebagai daun sawi ada hama. Kondisi ini biasanya disebabkan oleh infeksi virus atau serangan hama, yang menyebabkan daun berlubang.
2. Daun Sawi Tanpa Hama: Terdapat 500 gambar daun sawi yang dikategorikan sebagai daun sawi tanpa hama, dalam kategori ini menunjukkan daun sawi tanpa tanda-tanda infeksi virus atau serangan hama.

2.2. Processing Data

Sebelum tahap pelatihan atau pengujian model, dilakukan *preprocessing* [12]. Sebelum data dibagi menjadi data pelatihan, pengujian, dan validasi, *preprocessing* dilakukan terlebih dahulu. Dimulai dengan memilih 1.000 gambar daun sawi untuk digunakan dalam penelitian. Ada beberapa gambar yang serupa tetapi memiliki label kelas yang berbeda, yang dapat mengganggu proses pembelajaran model *Convolutional Neural Network (CNN)* di kemudian hari. Oleh karena itu, keputusan ini sangat penting.

Preprocessing dilakukan untuk meningkatkan kualitas gambar, mengurangi suara, dan menyesuaikan karakteristik gambar untuk memenuhi persyaratan analisis atau pengolahan berikutnya. Untuk mengubah ukuran gambar menjadi 512 x 512 piksel, *preprocessing* digunakan untuk mengurangi ukurannya. Setelah itu, gambar diubah ke format *RGB* dan dinormalisasi dengan membagi setiap nilai piksel dengan 255, sehingga setiap nilai piksel memiliki rentang nilai antara 0 dan 1. Proses ini dilakukan untuk meningkatkan waktu komputasi [13].

2.3. Augmentasi Data

Dengan menggunakan transformasi pada gambar-gambar yang ada, *augmentasi data* dapat digunakan untuk membuat variasi baru dari data latih. Tujuan *augmentasi data* adalah untuk meningkatkan jumlah sampel data latih dan menambahkan variasi ke dalam dataset, sehingga model dapat digeneralisasi dengan lebih baik dan mengurangi risiko *overfitting* [14]. Parameter *augmentasi* yang digunakan dalam penelitian ini adalah *zoom_range* dengan nilai 0,2, *fill_mode* dengan nilai terdekat, *rotation_range* dengan nilai 30, *high_shift_range* dengan nilai 0,2, *height_shift_range* dengan nilai 0,2, *shear_range* dengan nilai 0,2, dan *horizontal_flip* dengan nilai benar.

2.4. Training Model CNN

Data training terdiri dari kumpulan data yang telah disesuaikan untuk diproses menggunakan *Convolutional Neural Network (CNN)* [15]. Total dataset daun sawi berjumlah 1.000 gambar, yang dibagi dalam dua kelas, yaitu daun sawi ada hama dan daun sawi tanpa hama, 800 data gambar digunakan untuk pelatihan, 100 untuk pengujian, dan 100 untuk validasi. *Xception* adalah arsitektur *CNN* yang digunakan dalam penelitian ini [16].

2.4.1. Xception

Dengan menggunakan pendekatan konvolusi yang dapat dipisahkan secara mendalam, *Xception* adalah struktur *Convolutional Neural Network* [17]. Struktur ini adalah pengembangan dari struktur *Inception* dan menggunakan *convolution depthwise* dan *pointwise* untuk membentuk blok residu. Ini memungkinkan aliran data melalui lapisan jaringan dengan lebih efisien. Dibandingkan dengan arsitektur *CNN* lainnya, *Xception* memiliki jumlah parameter dan komputasi yang lebih sedikit, yang merupakan keunggulan utamanya. Untuk aplikasi yang memiliki keterbatasan sumber daya komputasi, *Xception* adalah pilihan terbaik [18].

2.4.2. Convolution Layer

Convolution Layer adalah langkah penting dalam metode *CNN*, yang menerapkan proses konvolusi pada keluaran dari lapisan sebelumnya. Lapisan konvolusi menggunakan filter yang disebut kernel untuk mengekstraksi fitur atau objek dari gambar input; kernel terdiri dari bobot-bobot yang berfungsi untuk mendeteksi karakteristik khusus dari objek. Proses konvolusi kemudian dilakukan untuk menghasilkan transformasi *linier* dari gambar *input*, yang mencerminkan informasi spasial yang terkandung dalam data tersebut [19].

2.4.3. Pooling Layer

Pooling Layer berfungsi untuk menggunakan pengurangan sampel atau pengurangan spasial input, layer kolaborasi membantu mengurangi dimensi data selama proses konvolusi. Dengan menggunakan operasi pooling, kami dapat menyederhanakan representasi data, membuatnya lebih ringkas dan mudah dikelola, dan membantu mengontrol *overfitting*. *Max Pooling* mengambil nilai maksimum dari area tertentu; *Average Pooling* mengambil nilai rata-rata dari area yang sama. Ini adalah metode *pooling* yang paling umum digunakan [20].

2.4.4. Dropout

Dropout adalah metode untuk mengatasi *overfitting* pada jaringan saraf, *dropout* secara acak "mematikan" sejumlah *neuron* selama proses pelatihan. Hal ini mencegah model terlalu terfokus pada pola-pola tertentu dalam data pelatihan, yang mungkin tidak muncul dalam data pengujian atau data baru, dan memaksa model untuk belajar lebih banyak representasi yang berguna daripada bergantung [21].

2.4.5. Flattening

Flattening adalah proses di mana matriks dari lapisan pooling diubah menjadi satu baris atau vektor tunggal. Proses ini mengubah data dari matriks dua dimensi menjadi satu dimensi yang dapat digunakan sebagai input untuk lapisan berikutnya dalam jaringan saraf tiruan. Setelah fitur-fitur yang telah dipool dipetakan, setiap baris matriks digabungkan menjadi satu baris panjang dengan satu kolom. Untuk model berikutnya, vektor ini dimasukkan ke dalam lapisan input. Ini memungkinkan data dipindahkan ke lapisan jaringan saraf tiruan yang penuh dengan koneksi [22].

2.4.6. Dense

Ini adalah lapisan *fully connected (dense layer)* yang berfungsi untuk klasifikasi berdasarkan karakteristik dari lapisan sebelumnya. Metode *softmax* digunakan dalam proses klasifikasi untuk menentukan probabilitas setiap kelas target dibandingkan dengan seluruh kelas yang ada. Fungsi *softmax* mengubah *output* dari lapisan sebelumnya menjadi nilai probabilitas yang berkisar antara 0 dan 1. Jumlah keseluruhan probabilitas kelas target selalu sama dengan satu. Metode ini bekerja dengan mengambil nilai eksponensial dari setiap input dan kemudian menjumlahkan nilai eksponensial untuk setiap kelas. Hasil akhir dari metode ini adalah distribusi probabilitas, yang menunjukkan seberapa besar kemungkinan setiap kelas memiliki label yang tepat untuk input yang diberikan [22].

2.5. Pengujian Model CNN

Pengujian model *CNN* melibatkan evaluasi kinerja model setelah pelatihan, dengan tujuan utama untuk menilai kemampuan model untuk mengklasifikasikan data baru yang belum pernah dilihat sebelumnya. Setelah pelatihan menggunakan data latih, model diuji menggunakan data uji (*testing data*) dan data validasi, yang tidak digunakan selama proses pelatihan [23]. Pengujian ini sangat penting untuk menentukan seberapa

baik model dapat menggunakan informasi yang dikumpulkan dari data latih dalam situasi yang lebih umum. Pengujian tidak hanya mengukur akurasi model tetapi juga membantu menemukan masalah potensial seperti *overfitting*, di mana model mungkin terlalu spesifik pada pola dalam data latih sehingga kurang efektif ketika menghadapi data baru. Dengan pengujian yang tepat, kita dapat memastikan bahwa model tidak hanya mengenali pola yang spesifik dari data latih, sehingga dapat digunakan dalam aplikasi nyata.

2.6. Klasifikasi Citra

Klasifikasi citra pada *Convolutional Neural Network (CNN)* mengacu pada kemampuan model *CNN* untuk mengidentifikasi hama pada gambar dengan menggunakan pembelajaran yang diperolehnya selama proses pelatihan dengan menggunakan data latih. Diharapkan bahwa model *CNN* yang telah dilatih dapat mengenali pola dan fitur khusus yang menunjukkan adanya hama pada daun sawi, sehingga dapat melakukan klasifikasi yang sangat akurat pada data baru yang tidak pernah dilihat sebelumnya [24].

2.7. Confusion Matrix

Pada tahap ini, evaluasi kinerja model arsitektur transfer learning yang digunakan dilakukan. *Accuracy*, *precision*, *recall*, dan *F1 skor* adalah beberapa *metrik* yang digunakan untuk mengukur kinerja model. Setiap metrik dapat dihitung dengan menggunakan persamaan (1), (2), (3), dan (4). Dalam konteks ini, variabel TP (*True Positive*) mewakili jumlah prediksi benar yang diklasifikasikan sebagai positif, TN (*True Negative*) mewakili jumlah prediksi benar yang diklasifikasikan sebagai negatif, FP (*False Positive*) mewakili jumlah prediksi yang salah diklasifikasikan sebagai positif, dan FN (*False Negative*) menunjukkan jumlah prediksi yang salah diklasifikasikan sebagai negatif dalam konteks ini [5]. Sangat penting untuk melakukan evaluasi ini untuk memahami seberapa baik model menangani data yang belum pernah dilihat sebelumnya dan untuk menemukan area di mana perbaikan lebih lanjut diperlukan.

2.7.1. Accuracy

Accuracy adalah matriks yang digunakan untuk mengukur perbandingan keseluruhan *precision* yang benar terhadap total nilai kelas [25].

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

2.7.2. Precision

Precision adalah matriks yang dipergunakan untuk menghitung akurasi atau hasil *precision* yang positif dari keseluruhan hasil yang diprediksi sebagai positif [25].

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

2.7.3. Recall

Recall adalah sebuah matriks yang digunakan untuk menghitung jumlah *precision* yang relevan dengan kelas yang sama [26].

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

2.7.4. F1 Score

F1 Score adalah perbandingan rata-rata *precision* dan *recall* yang dibobotkan [26].

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

Keterangan dari formula diatas adalah:

<i>True Positive</i> (TP)	=Variabel TP mewakili jumlah prediksi benar positif
<i>True Negative</i> (TN)	=Variabel TN mewakili jumlah prediksi benar negatif
<i>False Positive</i> (FP)	=Variabel FP mewakili jumlah prediksi salah positif
<i>False Negative</i> (FN)	=Variabel FN mewakili jumlah prediksi salah negatif

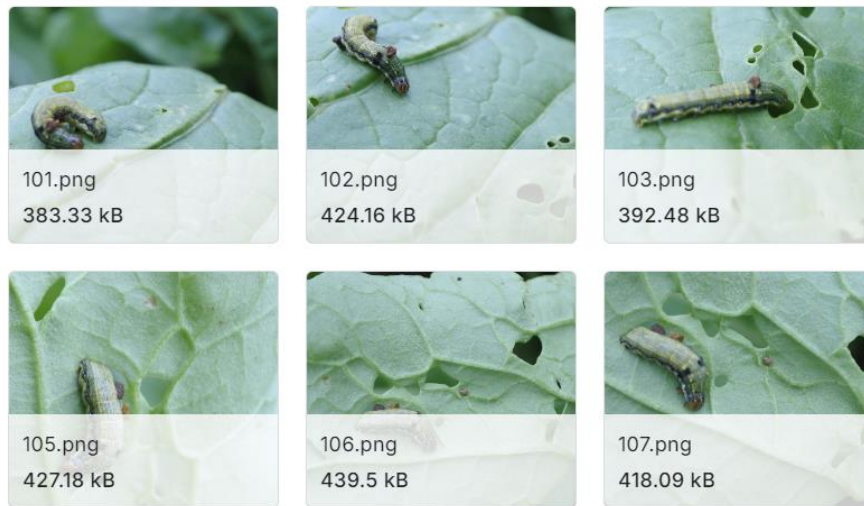
3. HASIL DAN PEMBAHASAN

Pada bagian hasil dan analisis dari pengujian klasifikasi hama pada daun sawi menggunakan metode *Convolutional Neural Network (CNN)* yang dimodifikasi dengan beberapa parameter inialisasi disajikan pada bagian hasil dan pembahasan. Penelitian ini bertujuan untuk menemukan kombinasi yang paling tepat dan paling akurat dari berbagai parameter, seperti ukuran *batch*, jumlah *epoch*, *dropout*, *rate* pembelajaran, dan *optimizer*. Hal ini disebabkan oleh berbagai faktor, seperti tidak menggunakan arsitektur *CNN* atau tidak menerapkan parameter *dropout*. Dalam pengujian ini ukuran *batch* yang digunakan dalam tes ini adalah 64. Jumlah *epoch* yang digunakan juga adalah 100. *dropout* disetel pada nilai 0,5, dengan *learning rate* 0,0001, dan *optimizer Adam*. Hasil pengujian dianalisis menggunakan *confucion matrix* untuk mengumpulkan nilai *Accuracy*, *precision*, *recall*, dan *F1 Score*, serta kinerja model dalam memprediksi gambar hama pada daun

sawi. Melalui hasil dan analisis ini, diharapkan dapat diperoleh pemahaman yang lebih mendalam tentang efektivitas model *CNN* dalam menangani masalah klasifikasi hama daun sawi dan untuk menemukan parameter yang relevan.

3.1. Hasil Pengumpulan Dataset

Pada langkah pengumpulan data, penulis memeriksa data dari berbagai sumber, termasuk dataset yang diunduh dari situs web publik seperti Kaggle yang menampilkan gambar daun sawi dengan dan tanpa hama ulat. Salah satu contoh distribusi data yang digunakan dalam penelitian ini digambarkan dalam Gambar 2. Preprocessing gambar uji adalah langkah pertama setelah dataset dikumpulkan. Pada titik ini, proses komputasi dipercepat dengan mengubah ukuran gambar menjadi 512 x 512 piksel. Selain itu, data augmentasi digunakan pada model yang dibangun untuk mengatasi overfitting. Augmentasi ini menghasilkan variasi gambar yang lebih besar selama proses pelatihan dengan melakukan pergeseran, flip, shear, dan perubahan skala pada gambar. Untuk membuat dataset lebih kuat, teknik ini penting untuk memperkayanya. Ini membuat model lebih tahan terhadap perubahan dalam kondisi gambar di dunia nyata. Penulis membagi dataset menjadi dua kelas utama gambar daun sawi yang terdapat hama ulat dan gambar daun sawi yang tidak terdapat hama ulat. Tujuan penulis adalah untuk membuat model lebih efisien dalam membedakan daun yang terinfeksi hama ulat dari daun yang sehat dengan memasukkan gambar daun yang menunjukkan kerusakan yang disebabkan oleh hama ulat. Setelah data hama ulat ini diproses dan dimasukkan ke dalam kedua kelas ini, diharapkan akurasi model akan meningkat.



Gambar 2. Dataset citra hama daun sawi

3.2. Hardware dan Software

Dalam pengujian ini menggunakan laptop Lenovo V14, yang memiliki prosesor *Intel Core i3* generasi 3, *RAM 4GB*, *SSD 256GB*, dan sistem operasi *Windows 11 Pro*. *Google Collab* adalah alat bantu untuk memulai program. Dengan menggunakan algoritma *Xception* dan optimasi *Adam*, penelitian ini melibatkan proses pelatihan dan validasi model. Hasil implementasi dan kinerja model setelah proses pelatihan dan validasi ditunjukkan dalam tabel berikut.

Tabel 2	Tabel 2. <i>Transfer Learning</i>	
	Nama	Parameter
	<i>Optimization</i>	<i>Adam</i>
	<i>Batch size</i>	64
	<i>Shape input</i>	7, 7, 2048
	<i>Learning Rate</i>	0.001
	<i>Activation</i>	<i>Softmax</i>

3.3. Hasil Pelatihan Model *CNN*

Hasil pelatihan model *Convolutional Neural Network (CNN)* menunjukkan kualitas dan kinerja model setelah melewati proses pelatihan dengan data yang diberikan. Parameter model seperti bobot dan bias pada setiap lapisan *CNN* disesuaikan selama proses pelatihan berdasarkan pola dan karakteristik yang ditemukan dalam data pelatihan. Untuk mengevaluasi hasil pelatihan model *CNN*, berbagai indikator dan metrik digunakan. Tujuan dari penyesuaian ini adalah agar model dapat memahami dan mengenali pola tersebut, sehingga dapat menggunakan data baru. Di bawah ini adalah gambar 4 yang menunjukkan hasil evaluasi dan kinerja model.

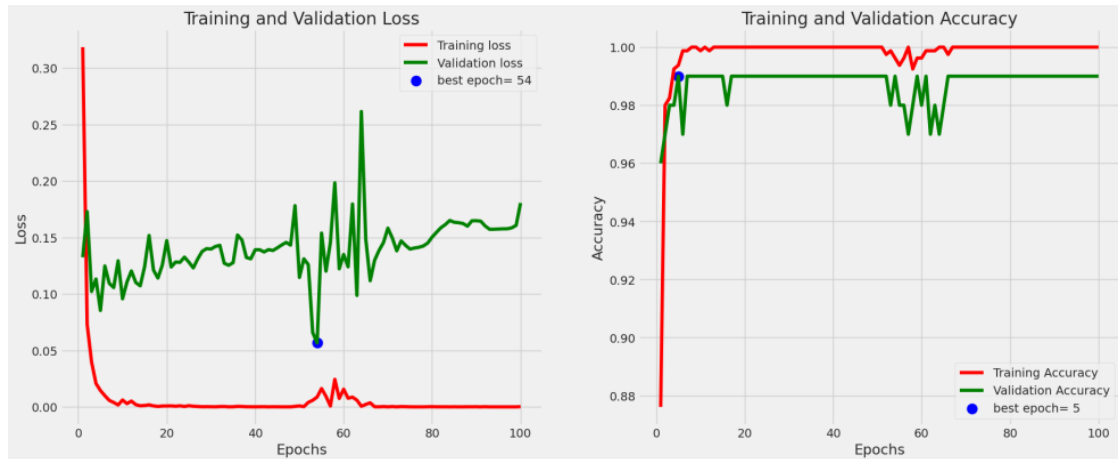
```

Epoch 1/100
13/13 [=====] - 257s 19s/step - loss: 0.3184 - accuracy: 0.8761 - val_loss: 0.1324 - val_accuracy: 0.9600
Epoch 2/100
13/13 [=====] - 247s 19s/step - loss: 0.0731 - accuracy: 0.9800 - val_loss: 0.1728 - val_accuracy: 0.9700
Epoch 3/100
13/13 [=====] - 251s 18s/step - loss: 0.0397 - accuracy: 0.9825 - val_loss: 0.1022 - val_accuracy: 0.9800
Epoch 4/100
13/13 [=====] - 250s 19s/step - loss: 0.0207 - accuracy: 0.9925 - val_loss: 0.1132 - val_accuracy: 0.9800
Epoch 5/100
13/13 [=====] - 262s 20s/step - loss: 0.0145 - accuracy: 0.9937 - val_loss: 0.0854 - val_accuracy: 0.9900
Epoch 6/100
13/13 [=====] - 232s 18s/step - loss: 0.0099 - accuracy: 0.9987 - val_loss: 0.1247 - val_accuracy: 0.9700
Epoch 7/100
13/13 [=====] - 246s 19s/step - loss: 0.0058 - accuracy: 0.9987 - val_loss: 0.1094 - val_accuracy: 0.9900
Epoch 8/100
13/13 [=====] - 253s 19s/step - loss: 0.0039 - accuracy: 1.0000 - val_loss: 0.1056 - val_accuracy: 0.9900
Epoch 9/100
13/13 [=====] - 241s 18s/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 0.1293 - val_accuracy: 0.9900
Epoch 10/100
13/13 [=====] - 246s 19s/step - loss: 0.0061 - accuracy: 0.9987 - val_loss: 0.0957 - val_accuracy: 0.9900
Epoch 11/100
13/13 [=====] - 251s 19s/step - loss: 0.0029 - accuracy: 1.0000 - val_loss: 0.1105 - val_accuracy: 0.9900
Epoch 12/100
13/13 [=====] - 246s 20s/step - loss: 0.0052 - accuracy: 0.9987 - val_loss: 0.1203 - val_accuracy: 0.9900
Epoch 13/100
13/13 [=====] - 242s 19s/step - loss: 0.0020 - accuracy: 1.0000 - val_loss: 0.1101 - val_accuracy: 0.9900
Epoch 14/100
13/13 [=====] - 236s 18s/step - loss: 9.9820e-04 - accuracy: 1.0000 - val_loss: 0.1072 - val_accuracy: 0.9900
Epoch 15/100
13/13 [=====] - 251s 19s/step - loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.1243 - val_accuracy: 0.9900
Epoch 16/100
13/13 [=====] - 233s 18s/step - loss: 0.0018 - accuracy: 1.0000 - val_loss: 0.1517 - val_accuracy: 0.9800
Epoch 17/100
13/13 [=====] - 240s 18s/step - loss: 9.0764e-04 - accuracy: 1.0000 - val_loss: 0.1211 - val_accuracy: 0.9900

```

Gambar 3. Hasil dari Pelatihan Model *CNN*

Upaya untuk menemukan nilai akurasi terbaik untuk model yang dikembangkan ditunjukkan oleh Gambar 3, yang menunjukkan hasil pelatihan model *Convolutional Neural Network (CNN)* dari seratus eksperimen ini. Variasi beberapa parameter, seperti ukuran *batch* dan jumlah *epoch*, dapat mempengaruhi performa model dalam proses ini. Dengan melakukan eksperimen dengan berbagai kombinasi parameter, penulis dapat memahami bagaimana model menanggapi perubahan tersebut. Pada setiap percobaan, model *CNN* dilatih dengan konfigurasi parameter yang berbeda. Setelah tahap pelatihan dan validasi selesai, akurasi diukur dan dicatat. Selanjutnya, hasil dari setiap percobaan dievaluasi dan dibandingkan untuk menentukan konfigurasi parameter yang paling akurat. Dalam penjelasan hasil pelatihan ini, analisis dari setiap percobaan biasanya disertakan, termasuk grafik yang menunjukkan *validation loss* dan *validation accuracy* selama *training*, serta perbandingan performa model pada setiap konfigurasi parameter. Analisis ini membantu memahami bagaimana perubahan nilai parameter mempengaruhi kinerja model, menentukan *overfitting* atau *underfitting*, dan menemukan nilai parameter yang dapat menghasilkan data validasi yang sangat akurat dan stabil. Dari 100 percobaan ini, percobaan pelatihan kelima paling akurat. Hasil penelitian ditunjukkan dalam grafik di bawah ini, yang menunjukkan bagaimana model *CNN* bekerja dengan baik.



Gambar 4. Grafik Akurasi dan *loss* terbaik pada pelatihan kelima

Pada gambar 4 di atas menunjukkan hasil pelatihan model *Convolutional Neural Network (CNN)* dengan arsitektur *Xception*, yang menunjukkan tingkat akurasi yang tinggi. Selama tahap pelatihan, model mencapai akurasi sebesar 99,00%, menunjukkan kemampuan yang sangat baik untuk memprediksi data yang digunakan selama proses pelatihan. Akurasi pelatihan yang tinggi ini menunjukkan bahwa model berhasil mengenali pola dan fitur pada data pelatihan dengan sangat efektif. Selain itu, pada tahap validasi, model juga mencapai akurasi 99,00%. Akurasi validasi ini mengukur kinerja model terhadap data validasi data yang belum pernah dilihat sebelumnya. Hasil akurasi validasi yang tinggi menunjukkan bahwa model mampu menggeneralisasi dengan baik ke data baru yang tidak ada dalam dataset pelatihan. Oleh karena itu, model ini dapat memprediksi klasifikasi hama daun sawi. Parameter *batch* sebesar 64 dan jumlah *epoch* sebanyak 100 digunakan untuk melatih model *CNN* dari seluruh dataset pelatihan. Ukuran *batch* 64 mengontrol jumlah sampel yang diproses oleh model sebelum memperbarui bobotnya pada setiap iterasi. Dalam hal ini, penggunaan *batch size* 64 terbukti efektif dalam mencapai akurasi yang tinggi. Jumlah *epoch* 100 menunjukkan bahwa model *CNN* dilatih selama 100 iterasi penuh. Model memiliki kesempatan yang cukup untuk mengidentifikasi dan menyesuaikan diri terhadap pola dan fitur dalam data pelatihan berkat jumlah *epoch* ini. Gambar yang digunakan memiliki ukuran 512 x 512 piksel dengan tiga *channel RGB*, yang berarti gambar memiliki resolusi tinggi dan terdiri dari tiga saluran warna (merah, hijau, dan biru). Memanfaatkan gambar dalam format ini memungkinkan model untuk memanfaatkan lebih banyak informasi warna dan fitur yang penting dalam proses pencarian. Menggabungkan ukuran *batch*, jumlah *epoch*, dan format gambar yang digunakan meningkatkan kemampuan model untuk mengklasifikasikan data dengan benar.

3.4. Hasil Klasifikasi Citra

Hasil pengujian menunjukkan bahwa pelatihan model *CNN Xception* yang menggunakan gambar 512 x 512 dengan parameter *batch size* 64 dan 100 *epoch* memiliki kinerja yang sangat baik. Pada tahap klasifikasi, gambar dievaluasi secara menyeluruh untuk setiap kelas dalam dataset. Sebagai bagian dari penggunaan pustaka Keras, prediksi acak digunakan. Teknik ini dimaksudkan untuk meningkatkan efisiensi analisis gambar. Hasil klasifikasi dari arsitektur *CNN Xception* menunjukkan bahwa konfigurasi parameter yang digunakan untuk memproses dan mengklasifikasikan gambar hama daun sawi sangat efektif.

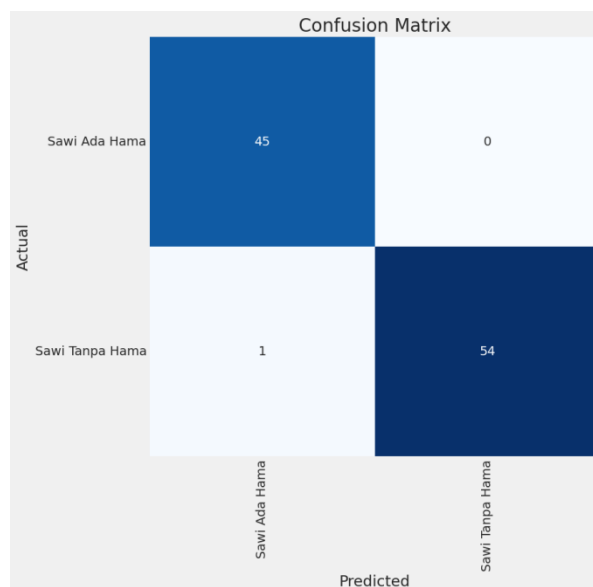


Gambar 5. Hasil klasifikasi pada percobaan pelatihan kelima

Pada gambar 5 menunjukkan hasil klasifikasi hama daun sawi menggunakan model *CNN Xception* menggunakan sampel acak. Dengan menggunakan citra yang ditampilkan, daun sawi ada hama dan daun sawi tanpa hama diidentifikasi dengan tepat sesuai kelasnya.

3.5. Evaluasi Model

Pengujian dalam penelitian ini menggunakan *Confusion Matrix* sebuah pendekatan yang efektif untuk mengevaluasi kinerja model dalam masalah klasifikasi. *Matrix* ini memungkinkan perhitungan *metrik* seperti *precision*, *recall*, *F1 score* dan *Accuracy*, serta informasi tentang jumlah prediksi yang benar atau salah untuk setiap kelas. Pengujian ini didasarkan pada model terbaik yang dibuat selama proses pelatihan; model ini dilatih dengan ukuran *batch* 64 dan *epoch* 100, dan mencapai akurasi pelatihan sebesar 99.00% dan akurasi validasi sebesar 99.00%. Hasil pengujian model yang cukup baik ini ditunjukkan pada Gambar 6 di bawah.



Gambar 6. Hasil dari confusion matrix

Gambar 6 menunjukkan hasil pengujian menggunakan *Confusion Matrix* untuk klasifikasi hama daun sawi dengan model *CNN Xception*. Model ini berhasil memprediksi 99 data tepat dari 100 data uji, dengan

tingkat akurasi rata-rata sebesar 99%, menunjukkan bahwa model ini berfungsi dengan sangat baik dan menunjukkan kemampuan untuk mengklasifikasikan hama daun sawi dengan akurat.

Hasil dari Classification Report hama daun sawi menunjukkan metrik evaluasi performa model klasifikasi ditunjukkan pada Gambar 7. Precision adalah rasio antara jumlah prediksi benar positif (*true positive* + *false positive*) dan total prediksi positif. *Precision* yang tinggi menunjukkan bahwa model cenderung memberikan sedikit prediksi positif yang salah (*false positive*), yaitu ketika model memprediksi positif padahal seharusnya negatif. Model menunjukkan kemampuan yang sangat baik untuk menemukan hama daun sawi dengan tingkat kesalahan yang sangat rendah. Gambar 8 di bawah ini menunjukkan tingkat *precision* tertinggi.

```

Classification Report:
-----
              precision    recall  f1-score   support

   Sawi Ada Hama         0.98      1.00      0.99         45
   Sawi Tanpa Hama       1.00      0.98      0.99         55

 accuracy                   0.99         100
 macro avg                   0.99      0.99      0.99         100
 weighted avg                0.99      0.99      0.99         100

```

Gambar 7. Hasil *Classification Report* Hama Daun Sawi

4. KESIMPULAN

Berdasarkan hasil yang telah diuraikan, metode *CNN* digunakan untuk mengklasifikasikan hama daun sawi dalam dua kelas daun sawi dengan hama dan daun sawi tanpa hama. Dataset ini terdiri dari 800 gambar untuk pelatihan, 100 gambar untuk pengujian, dan 100 gambar untuk validasi. *Preprocessing* dilakukan dengan mengubah ukuran setiap gambar menjadi 512 x 512 piksel dengan tiga *channel RGB*. Berbagai parameter digunakan untuk melatih model, tetapi parameter terbaik adalah 100 *epoch*, yang menunjukkan bahwa model dilatih selama 100 iterasi penuh dari seluruh dataset pelatihan. Untuk mengurangi *overfitting*, ukuran *batch* sebesar 64 dan nilai *dropout* 0,5 digunakan. Untuk mengoptimasi bobot model, tingkat pembelajaran (*learning rate*) sebesar 0.0001 digunakan. Sebaliknya, *optimizer Adam* dipilih karena kemanjurannya dalam menemukan bobot optimal. Hasil klasifikasi hama daun sawi dengan metode *CNN* ini menunjukkan bahwa model ini berhasil mencapai akurasi pelatihan sebesar 99,00% dan akurasi validasi sebesar 99,00% pada tahap pengujian, dengan prediksi yang tepat untuk setiap 100 data uji. Hasil ini menunjukkan bahwa model ini memiliki kemampuan yang luar biasa untuk mengenali dan mengklasifikasikan hama daun sawi, yang mungkin bermanfaat dalam pengendalian hama pertahanan. Namun, ada beberapa kekurangan penelitian yang dapat diperbaiki dalam penelitian berikutnya. Salah satu saran perbaikan adalah penggunaan dataset dengan jumlah kelas dan dataset yang lebih banyak, diharapkan bahwa ini akan meningkatkan nilai akurasi pengujian.

UCAPAN TERIMA KASIH

Kami menyampaikan penghargaan dan terima kasih yang sebesar-besarnya kepada pembimbing kami, Dr. Andi Sunyoto, M.Kom., dan Mei P. Kurniawan, M.Kom., atas bimbingan, dukungan, serta motivasi yang tak henti-hentinya beliau berikan selama berlangsungnya penelitian ini. Dedikasi dan pengetahuan yang dibagikan sangat berharga dalam menyelesaikan penelitian ini.

REFERESI

- [1] S. Paling, I. Inri, and L. Polona, "Identifikasi Jenis - Jenis Hama Yang Menginvasi Tanaman Sawi Hijau (*Brassica rapa* var. *parachinensis*) Di Lahan Pertanian Stkip Kristen Wamena," *STIGMA J. Mat. dan Ilmu Pengetah. Alam Unipa*, vol. 12, no. 01, pp. 34–40, 2019, doi: 10.36456/stigma.vol12.no01.a1857.
- [2] M. Yusuf, S. A. Talaohu, and J. Purnamasari, "Sistem Pakar Diagnosa Penyakit Pada Tanaman Sawi Menggunakan Metode Convolutional Neural Network Berbasis Android," vol. 5, no. 1, pp. 67–76, 2024, doi: 10.30865/klik.v5i1.2031.
- [3] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 1, pp. 58–65, 2022, doi: 10.29100/jipi.v7i1.2410.

- [4] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [5] R. H. Hridoy, A. D. Arni, and M. A. Hassan, "Recognition of Mustard Plant Diseases Based on Improved Deep Convolutional Neural Networks," *2022 IEEE Reg. 10 Symp. TENSYP 2022*, pp. 1–6, 2022, doi: 10.1109/TENSYP54529.2022.9864487.
- [6] R. Sharma, V. Kukreja, and Sakshi, "Mustard Downy Mildew Disease Severity Detection using Deep Learning Model," *2021 Int. Conf. Decis. Aid Sci. Appl. DASA 2021*, pp. 466–470, 2021, doi: 10.1109/DASA53625.2021.9682305.
- [7] A. Kurniadi, "Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras," *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 1, p. 25, 2020, doi: 10.25273/doubleclick.v4i1.5812.
- [8] P. P. Singh, R. Kaushik, H. Singh, N. Kumar, and P. S. Rana, "Convolutional neural networks based plant leaf diseases detection scheme," *2019 IEEE Globecom Work. GC Wkshps 2019 - Proc.*, 2019, doi: 10.1109/GCWkshps45667.2019.9024434.
- [9] M. I. Rasyid and L. M. Wisudawati, "Klasifikasi Hama Ulat Pada Citra Daun Sawi Berbasis Convolutional Neural Network Dengan Model Xception".
- [10] S. Hanifah, N. Apriliani, E. T. Suciarto, and E. S. Purwati, "Identifikasi Jamur Penyebab Penyakit pada Tanaman Sawi Putih (Brassica rapa L .) dan Persentase Penyakitnya di Desa Serang Kecamatan Karangreja, Kabupaten Purbalingga," *Bio Eksakta J. Ilm. Biol. Unsoed*, vol. 2, no. 2020, pp. 487–501, 2020.
- [11] C. L. Nazalia, P. Palupiningsih, B. Prayitno, and Y. S. Purwanto, "Implementation of Convolutional Neural Network Algorithm to Pest Detection in Caisim," *ICCoSITE 2023 - Int. Conf. Comput. Sci. Inf. Technol. Eng. Digit. Transform. Strateg. Facing VUCA TUNA Era*, pp. 609–614, 2023, doi: 10.1109/ICCoSITE57641.2023.10127792.
- [12] M. E. Prasetyo, M. R. Faza, R. Pratama, S. N. H. Alhabsy, H. Purwanti, and A. P. A. Masa, "Klasifikasi Ragam Kendaraan Menggunakan Metode Convolutional Neural Network (Cnn)," *Adopsi Teknol. dan Sist. Inf.*, vol. 2, no. 2, pp. 142–148, 2023, doi: 10.30872/atasi.v2i2.1156.
- [13] A. Akram, K. Fayakun, and H. Ramza, "Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network," *Build. Informatics, Technol. Sci.*, vol. 5, no. 2, pp. 397–406, 2023, doi: 10.47065/bits.v5i2.4063.
- [14] A. Ganang Pradana, D. Rosal, I. M. Setiadi, and A. R. Muslikh, "Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao Fine tuning Convolutional Neural Network model EfficientNet-B4 with data augmentation for cocoa disease classification," *J. Inf. Syst. Appl. Dev.*, vol. 2, no. 1, pp. 1–11, 2024, doi: 10.26905/jisad.v2i1.11899.
- [15] V. Atliha and D. Sesok, "Comparison of VGG and ResNet used as Encoders for Image Captioning," *2020 IEEE Open Conf. Electr. Electron. Inf. Sci. eStream 2020 - Proc.*, pp. 1–4, 2020, doi: 10.1109/eStream50540.2020.9108880.
- [16] R. L. Gaho, I. T. Ali, and E. Prakasa, "Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode Cnn Berbasis Arsitektur Xception," *INOVTEK Polbeng - Seri Inform.*, vol. 9, no. 1, pp. 354–365, 2024, doi: 10.35314/isi.v9i1.4213.
- [17] Darmasita, "Deteksi Penggunaan Masker Menggunakan Xception Transfer Learning," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 5, pp. 279–288, 2020.
- [18] R. Kurniawan, P. B. Wintoro, Y. Mulyani, and M. Komarudin, "Implementasi Arsitektur Xception Pada Model Machine Learning Klasifikasi Sampah Anorganik," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 2, pp. 233–236, 2023, doi: 10.23960/jitet.v11i2.3034.
- [19] D. Candra, G. Wibisono, M. Ayu, and M. Afrad, "Transfer Learning model Convolutional Neural Network menggunakan VGG-16 untuk Klasifikasi Tumor Otak pada Citra Hasil MRI," *LEDGER J. Inform. Inf. Technol.*, vol. 3, no. 1, pp. 11–18, 2024.
- [20] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritma.*, vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.
- [21] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021, doi: 10.1016/j.ymsp.2020.107398.
- [22] I. Suhardin, A. Patombongi, and A. M. Islah, "MENGIDENTIFIKASI JENIS TANAMAN BERDASARKAN CITRA DAUN MENGGUNAKAN AIGORITMA CONVOLUTIONAL NEURAL NETWORK," *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 6, no. 2, pp. 100–108, 2021, doi: 10.51876/simtek.v6i2.101.
- [23] A. B. Prakosa, Hendry, and R. Tanone, "Implementasi Model Deep Learning Convolutional Neural

- Network (CNN) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman,” *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 107–116, 2023.
- [24] E. D. Ratnasari, D. A. Rudira, and A. Surya Buana, “Klasifikasi Penyakit Daun Sawi Hijau Dengan Metode Cnn,” *Pros. Semin. Nas. Teknol. Dan Sains*, vol. 3, pp. 338–393, 2024.
- [25] K. Hameed, D. Chai, and A. Rassau, “A comprehensive review of fruit and vegetable classification techniques,” *Image Vis. Comput.*, pp. 1–35, 2018, doi: 10.1016/j.imavis.2018.09.016.
- [26] P. Romadloni, B. Adhi Kusuma, and W. Maulana Baihaqi, “Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 622–628, 2022, doi: 10.36040/jati.v6i2.5238.