

## Implementasi Deep Learning dengan Convolutional Neural Network untuk Tingkat Akurasi Citra Image Hama Sawi Hijau Menggunakan Google Colab

Nurliani<sup>1</sup>, Meitra Ulfi<sup>2</sup>, Saudah<sup>3</sup>, Tengku Savira Putri Ayu<sup>4</sup>, Adyanatalubis<sup>5</sup>

<sup>1234</sup> Program Studi, Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pasir Pengaraian

<sup>5</sup>Universitas Rokania, Riau, Indonesia

Email: [liannn2019@gmail.com](mailto:liannn2019@gmail.com)<sup>1</sup>, [meitraulfi@gmail.com](mailto:meitraulfi@gmail.com)<sup>2</sup>, [ssaudah116@gmail.com](mailto:ssaudah116@gmail.com)<sup>3</sup>,  
[psavira05@gmail.com](mailto:psavira05@gmail.com)<sup>4</sup>, [adyanata@gmail.com](mailto:adyanata@gmail.com)<sup>5</sup>

**Abstrak:** Penelitian ini membahas tentang penerapan metode *Convolutional Neural Network* (CNN) untuk mengidentifikasi hama tanaman sawi hijau berdasarkan gambar berwarna. Tujuan utama dari penelitian ini adalah untuk mengembangkan model yang dapat mendeteksi berbagai jenis penyakit hama daun sawi dengan akurasi tinggi. Dalam studi ini, Google Colab digunakan sebagai platform pemrosesan untuk model *Convolutional Neural Network* karena menyediakan lingkungan komputasi yang kuat dan akses gratis ke GPU, sehingga mempercepat pelatihan model. Dataset yang digunakan dalam penelitian ini diperoleh dari platform Kaggle yang menyediakan berbagai dataset berkualitas tinggi untuk penelitian dan pengembangan model pembelajaran mesin. Kaggle tidak hanya menawarkan beragam kumpulan data, tetapi juga kompetisi data, buku catatan interaktif untuk analisis data, dan forum diskusi aktif. Kumpulan data Kaggle biasanya diproses dan terstruktur dengan baik, sehingga memudahkan penggunaan langsung oleh peneliti dalam pengembangan model. Dalam konteks penelitian ini, Kaggle digunakan sebagai sumber kumpulan gambar tanaman sawi, yang kemudian diolah dan digunakan untuk melatih model *Convolutional Neural Network* di Google Colab. Hasil penelitian menunjukkan bahwa metode *Convolutional Neural Network* yang digunakan mampu mengidentifikasi berbagai jenis hama pada tanaman sawi dengan akurasi yang memuaskan. Temuan ini diharapkan dapat membantu petani mengidentifikasi dan mengelola penyakit hama dengan lebih efektif, sehingga meningkatkan hasil dan kualitas tanaman sawi

**Kata Kunci:** *Convolutional Neural Network* (CNN), Deteksi Penyakit Hama Sawi Hijau, Citra Warna, Google Colab, Akurasi

**Abstract :** *This study discusses the application of Convolutional Neural Network (CNN) methods to identify pests in green mustard plants based on color images. The main objective of this research is to develop a model that can detect various types of pest diseases in mustard leaves with high accuracy. In this study, Google Colab was used as the processing platform for the Convolutional Neural Network model because it provides a powerful computing environment and free access to GPUs, thereby accelerating the model training process. The dataset used in this research was obtained from the Kaggle platform, which provides various high-quality datasets for machine learning research and development. Kaggle not only offers a wide range of datasets but also data competitions, interactive notebooks for data analysis, and active discussion forums. Kaggle datasets are usually well-processed and structured, making it easy for researchers to use them directly in model development. In the context of this research, Kaggle was used as the source for the collection of mustard plant images, which were then processed and used to train the Convolutional Neural Network model in Google Colab. The results of the study show that the Convolutional Neural Network method used was able to identify various types of pests in mustard plants with satisfactory accuracy. These findings are expected to help farmers identify and manage pest diseases more effectively, thereby increasing the yield and quality of mustard plants.*

**Keywords:** *Convolutional Neural Network (CNN), Pest Disease Detection Green Mustard, Color Image, Google Colab, Accuracy,*

## 1. PENDAHULUAN

Sawi hijau, yang juga dikenal dengan nama *Brassica rapa subsp. chinensis* atau dalam bahasa Inggris sering disebut sebagai "Chinese mustard" atau "bok choy", adalah salah satu jenis sayuran daun yang populer dan banyak dibudidayakan di Asia, khususnya di Indonesia [1]. Sawi hijau memiliki ciri-ciri khas berupa daun yang lebar berwarna hijau tua dan tangkai daun yang putih tebal [2]. Tanaman ini dikenal karena kandungan nutrisinya yang tinggi serta manfaat kesehatannya [3].

Daun sawi memiliki beragam manfaat dan kegunaan dalam aktivitas sehari-hari. Selain digunakan sebagai bahan vegetarian, sawi juga dapat digunakan sebagai obat. Sawi mempunyai beberapa jenis hama, antara lain belalang kecil, ulat grayak (*Spodoptera litura*), kutu putih (*Plutella xylostella*), kutu putih (*Crociodolomia pavonana*), dan cacing potong (*Plusia sp.*). Jika kerusakan akibat hama ini 10-25 %, bahkan ada yang gagal panen [4].

Dalam bidang pertanian, identifikasi penyakit hama pada tanaman merupakan salah satu tantangan besar yang dihadapi oleh petani [5]. Penyakit dan hama dapat menurunkan produktivitas tanaman secara signifikan, sehingga diperlukan metode yang efektif dan efisien untuk mendeteksi dan mengelola penyakit ini [6] [7]. Dengan kemajuan teknologi, metode berbasis machine learning seperti *Convolutional Neural Network* (CNN) telah menjadi alat yang sangat berguna dalam menganalisis citra dan mendeteksi penyakit tanaman secara otomatis [8] [9].

*Convolutional Neural Network* adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data gambar [10] [11]. Metode ini bekerja dengan mengidentifikasi fitur-fitur penting dalam gambar melalui serangkaian lapisan konvolusi dan pooling. *Convolutional Neural Network* telah terbukti sangat efektif dalam berbagai aplikasi pengenalan pola, termasuk dalam bidang medis dan pertanian [12][13]. Rumus dasar yang digunakan dalam lapisan konvolusi adalah:

$$(I \cdot K)_{(x,y)} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x+i, y+j) \cdot K(i, j)$$

di mana  $I$  adalah input gambar,  $K$  adalah kernel (filter), dan  $*$  adalah operasi konvolusi

Kaggle adalah platform yang terkenal di kalangan peneliti data science dan machine learning. Platform ini menyediakan berbagai dataset berkualitas tinggi yang dapat digunakan untuk penelitian dan pengembangan model machine learning [14]. Selain dataset, Kaggle juga menawarkan berbagai kompetisi data science, notebook interaktif untuk analisis data, dan forum diskusi yang aktif. Dataset yang tersedia di Kaggle biasanya telah diproses dan disusun dengan baik, sehingga memudahkan peneliti dalam penggunaannya [15].

## 1. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen untuk mengembangkan dan menguji model *Convolutional Neural Network* (CNN) dalam mendeteksi penyakit hama pada tanaman sawi hijau berdasarkan citra warna [11] [16]. Proses penelitian melibatkan beberapa tahapan utama: pengumpulan dataset, preprocessing data, pengembangan model *Convolutional Neural Network*, pelatihan model, evaluasi model, dan interpretasi hasil.

### 2.1 Data Set Daun Sawi Hijau

Pengumpulan Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle, platform yang menyediakan berbagai dataset berkualitas tinggi untuk keperluan penelitian dan pengembangan model machine learning. Dataset terdiri dari berbagai gambar daun sawi hijau yang terinfeksi oleh berbagai jenis hama dan penyakit. Setiap gambar dilabeli sesuai dengan jenis hama atau penyakit yang teridentifikasi [15].

Dataset yang digunakan dalam penelitian ini terdiri dari gambar daun sawi hijau yang digunakan untuk melatih dan menguji model *Convolutional Neural Network* (CNN) dalam mendeteksi penyakit hama. Dataset mencakup gambar-gambar daun sawi hijau yang diambil dari berbagai kondisi, termasuk daun yang sehat dan daun yang terinfeksi hama [17]. Rincian dataset Jumlah Gambar Sampel 100 gambar sampel, Jumlah Gambar untuk Validasi 50 gambar yang terbagi dalam 2 kelas (sehat dan terinfeksi hama). Tujuan Validasi yaitu Menguji kemampuan aplikasi dalam memprediksi data baru yang belum pernah dilihat sebelumnya atau data di luar dari data training. Dengan Hasil gambar validasi 25 gambar sampel. pengujian: 10 langkah pengujian: 1 Ditemukan 80 nama file gambar tervalidasi yang termasuk dalam 2 kelas. Ditemukan 10 nama berkas gambar tervalidasi yang termasuk dalam 2 kelas. Ditemukan 10 nama berkas gambar tervalidasi yang termasuk dalam

2 kelas

Gambar 1 dibawah ini gambaran daun awal daun sawi hijau diserang hama. gambar daun sawi hijau dengan presisi daun bercak putih. Gambar menunjukkan daun sawi Keakuratan Alga Bercak Daun masih bersifat sementara.



Gambar 1. Daun sawi hijau gambaran awal terkena hama

Gambar 2 ini gambaran daun sawi hijau sudah parah terkena hama, hama sudah hamper merusak semua sisi daun. Pada daun sawi hijau dengan tingkat akurasi lightroast terlihat bahwa penyakit dari daun sawi hijau tersebut telah parah dan dimakan hama [9].



Gambar 2. Gambaran daun sawi hijau dengan hama sudah parah

### 2.3 Confusion Matrix (CM)

*Confusion Matrix* adalah alat penilaian visual yang biasa digunakan dalam pembelajaran mesin. Matriks konfusi dapat menghitung semua kemungkinan kasus masalah klasifikasi karena kolom matriks konfusi mewakili hasil kelas yang diprediksikan dan baris mewakili hasil kelas sebenarnya. Matriks konfusi menunjukkan beberapa jenis keluaran yang berbeda, termasuk skor presisi, akurasi, presisi, perolehan, dan f1. Persamaan berikut dapat membentuk matriks konfusi [12].

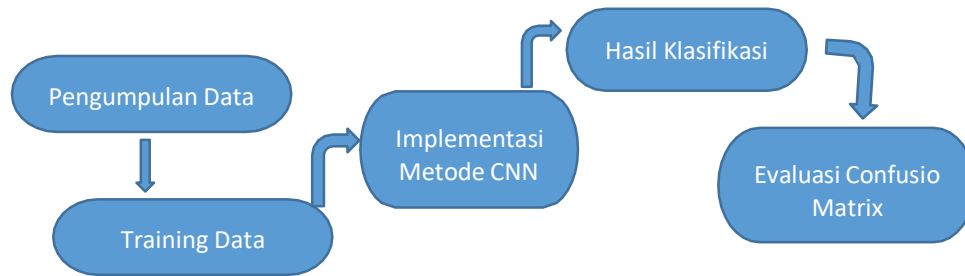
$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \dots\dots\dots(1)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(2)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

$$F1\ Score = \frac{2 \times Recall \times Presisi}{Recall+Presisi} \dots\dots\dots (4)$$

## 2.4 Flowchart/ Alur



Gambar Flowchart dari alur klasikasi hama daun sawi hijau

Penjelasan dari alur di atas:

1. Mengumpulkan dataset menggunakan situs kagle.
2. Dataset yang telah dikumpulkan dipecah menjadi data pelatihan (training data) dan data pengujian (testing data). Data pelatihan digunakan untuk melatih model *Convolutional Neural Network*.
3. Tahap ini mencakup pembangunan dan pelatihan model *Convolutional Neural Network* (CNN) menggunakan data pelatihan.
4. Setelah model *Convolutional Neural Network* (CNN) dilatih, tahap ini melibatkan penggunaan model untuk mengklasifikasikan data pengujian atau citra baru.
5. Mengevaluasi kinerja model dengan menggunakan metrik evaluasi yang dihasilkan dari *confusion matrix*.

## 3. HASIL DAN PEMBAHASAN

Implementasi ini dilakukan menggunakan Google Colab, platform yang memungkinkan eksekusi kode Python di cloud. Penggunaan Google Colab mempermudah proses komputasi yang berat, seperti pelatihan model *Convolutional Neural Network*, dengan memanfaatkan sumber daya GPU yang disediakan. Model *Convolutional Neural Network* yang diimplementasikan terbukti efektif dalam mendeteksi penyakit hama pada sawi hijau. Pengaturan parameter gambar dan ukuran batch yang tepat membantu dalam memastikan bahwa model menerima input yang konsisten dan diproses secara efisien. Hasil pengujian menunjukkan bahwa model memiliki tingkat akurasi yang tinggi dalam mengidentifikasi penyakit hama [14].

Persiapan lingkungan di Google Colab untuk memulai proyek ini, menggunakan Google Colab sebagai platform komputasi karena kemampuannya untuk mengakses GPU dan kemudahan penggunaannya. Berikut adalah langkah-langkah persiapan lingkungan

1. Membuka Google Colab di browser dan buat notebook baru
2. Mengimpor Paket yang Dibutuhkan:

Mengimpor library yang dibutuhkan seperti TensorFlow, Keras, NumPy, Pandas, Matplotlib, dan scikit-learn. Kode tersebut adalah serangkaian perintah impor dalam Python yang digunakan dalam proyek pembelajaran mendalam (deep learning) dan pembelajaran mesin (machine learning). Perintah-perintah ini mengimpor pustaka dan modul yang diperlukan, seperti TensorFlow dan Keras untuk membangun dan melatih model jaringan saraf, NumPy untuk komputasi ilmiah, pandas untuk manipulasi data, dan scikit-learn untuk membagi dataset.

```
[ ] import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv2D, MaxPoolin
from tensorflow.keras.optimizers import Adam, Adamax
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras import regularizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, load_model, Sequential
from keras.callbacks import ModelCheckpoint
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import os
import seaborn as sns
sns.set_style('darkgrid')
from sklearn.metrics import confusion_matrix, classification_report
from IPython.display import display, HTML
```

Gambar 4. Mengimpor library yang dibutuhkan

3. Selanjutnya kita extract terlebih dahulu file zip nya

```
▶ from zipfile import ZipFile

file_name = "/content/SAWI1.zip"
with ZipFile(file_name, 'r') as zip:
    zip.extractall()
print("Done")
```

➔ Done

Gambar 5. Ektrac file zip

4. Mengulang folder untuk membuat kerangka data dari label jalur file formulirImport Pandas.

Kode gambar dibawah ini memeriksa keberadaan direktori utama dan mengumpulkan jalur file serta label dari subdirektori di dalamnya. Hasilnya dengan data daun sawi hijau tanpa hama 50 dan daun sawi ada hama 50.

```
[ ] filepaths = []
labels = []

# Check if the directory exists
if os.path.isdir(sdir):
    classlist = os.listdir(sdir)
    for klass in classlist:
        classpath = os.path.join(sdir, klass)
        if os.path.isdir(classpath):
            flist = os.listdir(classpath)
            for f in flist:
                fpath = os.path.join(classpath, f)
                filepaths.append(fpath)
                labels.append(klass)
else:
    print(f"Directory {sdir} does not exist.")

# Create DataFrame
df = pd.DataFrame({'filepaths': filepaths, 'labels': labels})

# Display the first few rows and value counts of labels
print(df.head())
print(df['labels'].value_counts())
```

	filepaths	labels
0	/content/Dataset/Data Sawi Tanpa Hama/38.png	Data Sawi Tanpa Hama
1	/content/Dataset/Data Sawi Tanpa Hama/7.png	Data Sawi Tanpa Hama
2	/content/Dataset/Data Sawi Tanpa Hama/16.png	Data Sawi Tanpa Hama
3	/content/Dataset/Data Sawi Tanpa Hama/3.png	Data Sawi Tanpa Hama
4	/content/Dataset/Data Sawi Tanpa Hama/9.png	Data Sawi Tanpa Hama

```
labels
Data Sawi Tanpa Hama    50
Data Sawi Ada Hama      50
Name: count, dtype: int64
```

Gambar 6. Mengimpor pandas

5. Bagi df menjadi train\_df, test\_df dan valid\_df

Memastikan bahwa model dapat belajar dari data pelatihan, dioptimalkan berdasarkan data validasi, dan akhirnya dievaluasi pada data pengujian yang tidak terlihat selama proses pelatihan. Ini membantu dalam menghindari overfitting dan memberikan evaluasi yang lebih akurat tentang kinerja model di data yang belum pernah dilihat sebelumnya.

```
[ ] train_split=.8
test_split=.1
dummy_split=test_split/(1-train_split)
train_df, dummy_df=train_test_split(df, train_size=train_split, shuffle=True, random_state=123)
test_df, valid_df=train_test_split(dummy_df, train_size=dummy_split, shuffle=True, random_state=123)
print ('train_df length: ', len(train_df), ' test_df length: ', len(test_df), ' valid_df length: ', len(valid_df))
```

```
train_df length: 80 test_df length: 10 valid_df length: 10
```

Gambar 7. Bagi df menjadi train\_df, test\_df dan valid\_df

6. Membuat kereta, uji, generator yang valid, membuat ukuran gambar 64 isze. Kode tersebut mengatur parameter untuk gambar (tinggi, lebar, dan saluranwarna) dan menentukan ukuran batch untuk pengujian data, yaitu 64.Selanjutnya, kode menghitung ukuran batch dan jumlah langkah pengujian: 10 langkah pengujian:1

Ditemukan 80 nama file gambar tervalidasi yang termasuk dalam 2 kelas. Ditemukan 10 nama berkas gambar tervalidasi yang termasuk dalam 2 kelas.Ditemukan 10 nama berkas gambar tervalidasi yang termasuk dalam 2 kelas.

```
[ ] height=128
width=128
channels=3
batch_size=64

img_shape=(height, width, channels)
img_size=(height, width)
length=len(test_df)
test_batch_size=sorted([int(length/n) for n in range(1,length+1) if length % n ==0 and length/n<=64],reverse=True)[0]
test_steps=int(length/test_batch_size)
print (' test batch size: ',test_batch_size, ' test steps: ', test_steps)

gen=ImageDataGenerator(
    rescale=1./255,
)
train_gen=gen.flow_from_dataframe( train_df, x_col='filepaths', y_col='labels', target_size=img_size, class_mode='categorical',
    color_mode='rgb', shuffle=True, batch_size=batch_size)

valid_gen=ImageDataGenerator(rescale=1./255)
valid_gen=valid_gen.flow_from_dataframe( valid_df, x_col='filepaths', y_col='labels', target_size=img_size, class_mode='categorical',
    color_mode='rgb', shuffle=True, batch_size=batch_size)

test_gen=ImageDataGenerator(rescale=1./255)
test_gen=test_gen.flow_from_dataframe( test_df, x_col='filepaths', y_col='labels', target_size=img_size, class_mode='categorical',
    color_mode='rgb', shuffle=False, batch_size=test_batch_size)

classes=list(train_gen.class_indices.keys())
print (classes)
class_count=len(classes)
```

```
test batch size: 10 test steps: 1
Found 80 validated image filenames belonging to 2 classes.
Found 10 validated image filenames belonging to 2 classes.
Found 10 validated image filenames belonging to 2 classes.
['Data Sawi Ada Hama', 'Data Sawi Tanpa Hama']
```

Gambar 8. Membuat kereta, uji, generator yang valid

7. Membuat fungsi untuk menampilkan beberapa contoh gambar. Gambar daunsawi hijau seperti di bawah ini dengan menampilkan 25 gambar daun sawi hijau.



Gambar 9. Contoh gambar daun sawi hijau

8. Membuat model Convolutional Neural Network

Membuat atau pembangunan model *Convolutional Neural Network* (CNN) yang akan digunakan selama tahap pelatihan (training) dan pengujian (testing). Untuk membangun model ini, diperlukan beberapa lapisan (layers) agar klasifikasi yang dihasilkan dapat memenuhi nilai yang diharapkan [15].

```
[ ] base_model=tf.keras.applications.NASNetMobile(include_top=False, weights='imagenet', input_tensor=Input(shape=(128,128,3)))
↳ Mengunduh data dari https://storage.googleapis.com/tensorflow/keras-applications/nasnet/NASNet-Mobile-no-top.h5
19993432/19993432 [=====] - 0s 0us/langkah

[ ] base_model.summary()
↳ Model: "NASNet"
-----
Lapisan (tipe) Param Bentuk Keluaran * Terhubung ke
-----
input_1 (InputLayer) [(Tidak ada, 128, 128, 3)] 0 []
stem_conv1 (Konv2D) (Tidak ada, 63, 63, 32) 864 ['input_1[0][0]']
stem_bn1 (BatchNormalizati (Tidak Ada, 63, 63, 32) 128 ['stem_conv1[0][0]']
pada)
aktivasi (Aktivasi) (Tidak ada, 63, 63, 32) 0 ['stem_bn1[0][0]']
pengurangan_konv_1_batang_1 (C (Tidak Ada, 63, 63, 11) 352 ['aktivasi[0][0]']
pada2D)
reduksi_bn_1_stem_1 (Bat (Tidak Ada, 63, 63, 11) 44 ['reduction_conv_1_stem_1[0][0]
CINormalisasi)']
activation_1 (Aktivasi) (Tidak ada, 63, 63, 11) 0 ['reduction_bn_1_stem_1[0][0]'
[Bahasa Indonesia]
activation_3 (Aktivasi) (Tidak ada, 63, 63, 32) 0 ['stem_bn1[0][0]']
separable_conv_1_pad_reduc (Tidak Ada, 67, 67, 11) 0 ['activation_1[0][0]']
tion_left1_stem_1 (ZeroPad
ding2D)
separable_conv_1_pad_reduc (Tidak ada, 69, 69, 32) 0 ['aktivasi_3[0][0]']
```

Gambar 10. Membuat model CNN

9. Mendefinisikan fungsi untuk mencetak teks dalam warna latar depan dan latar belakang RGB, dan mendefinisikan fungsi untuk memplot data pelatihan.

```
+ Kode + Teks
def print_in_color(txt_msg,fore_tupple,back_tupple,):
    #prints the text_msg in the foreground color specified by fore_tupple with the background specified by back_tupple
    #text_msg is the text, fore_tupple is foreground color tuple (r,g,b), back_tupple is background tuple (r,g,b)
    rf,gf,bf=fore_tupple
    rb,gb,bb=back_tupple
    msg='{0}' + txt_msg
    mat='\33[38;2;' + str(rf) + ';' + str(gf) + ';' + str(bf) + ';' + str(rb) + ';' + str(gb) + ';' + str(bb) + 'm'
    print(msg .format(mat), flush=True)
    print('\33[0m', flush=True) # returns default print color to back to black
    return
```

Gambar 11. Mencetak teks dalam warna RGB

10. Selanjutnya tentukan fungsi untuk menghasilkan matriks kebingungan dan laporan klasifikasi

```
[ ] def print_info( test_gen, preds, print_code, save_dir, subject ):
    class_dict=test_gen.class_indices
    labels= test_gen.labels
    file_names= test_gen.file_names
    error_list=[]
    true_class=[]
    pred_class=[]
    prob_list=[]
    new_dict={}
    error_indices=[]
    y_pred=[]
    for key,value in class_dict.items():
        new_dict[value]=key # dictionary {integer of class number: string of class name)
    # store new_dict as a text file in the save_dir
    classes=list(new_dict.values()) # list of string of class names
    dict_as_text=str(new_dict)
    dict_name= subject + '.' + str(len(classes)) + '.txt'
    dict_path=os.path.join(save_dir,dict_name)
    with open(dict_path, 'w') as x_file:
        x_file.write(dict_as_text)
    errors=0
    for i, p in enumerate(preds):
        pred_index=np.argmax(p)
        true_index=labels[i] # labels are integer values
        if pred_index != true_index: # a misclassification has occurred
            error_list.append(file_names[i])
```

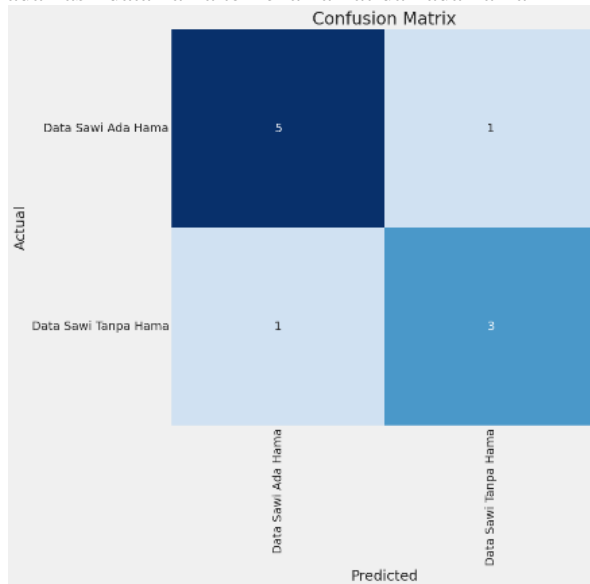
Gambar 12. Fungsi matriks kebingungan dan laporan klasifikasi

11. Evaluasi model pada set pengujian lalu simpan model Akurasi pada pengujian adalah 100 %



Gambar 13. Evaluasi model simpan

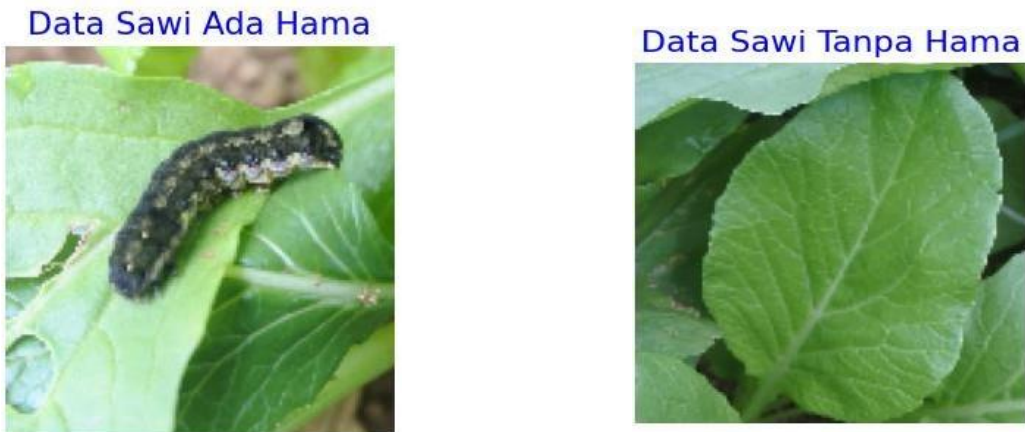
12. Membuat prediksi pada set pengujian dan menghasilkan matriks kebingungan (confusion matrix) dan laporan klasifikasi. Pada hasil data hama terkena hamatidak ada hama



Gambar 14. Hasil pengujian confusion matrix



13. Menampilkan hasil daun sawi hijau yang bagus dan terkena hama



Gambar 15. Daun sawi hijau ada hama dan tanpa hama

#### 4 KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan model deteksi hama pada tanaman sawi hijau menggunakan metode *Convolutional Neural Network* (CNN) yang diimplementasikan pada platform Google Colab. CNN dipilih karena kemampuannya dalam mendeteksi pola dan fitur dalam gambar digital. Penelitian ini menggunakan ribuan gambar daun sawi yang terinfestasi dan sehat sebagai data pelatihan. Hasil penelitian menunjukkan bahwa model *Convolutional Neural Network* (CNN) yang dikembangkan mampu mendeteksi hama dengan akurasi tinggi. Teknik augmentasi data dan validasi silang digunakan untuk meningkatkan akurasi dan generalisasi model. Implementasi pada Google Colab menyederhanakan proses pelatihan dengan memanfaatkan sumber daya komputasi awan yang canggih.

#### DAFTAR PUSTAKA

- [1] Nadjua Nuraini and Darmansyah, "PENGARUH DOSIS PUPUK GUANO TERHADAP PERTUMBUHAN DAN PRODUKSI TANAMAN SAWI HIJAU (*Brassica juncea* L.)," *HORTUSCOLER*, vol. 2, no. 02, 2021, doi: 10.32530/jh.v2i02.451.
- [2] J. Nurcholis, A. Vira, B. Buhaerah, and S. Syaifuddin, "PEMANFAATAN PUPUK ORGANIK CAIR (POC) KULIT PISANG KEPOK TERHADAP PERTUMBUHAN DAN PRODUKSI TANAMAN SAWI HIJAU (*Brassica rapa* var. *parachinensis* L.)," *Compos. J. Ilmu Pertan.*, vol. 3, no. 01, 2021, doi: 10.37577/composite.v3i01.307.
- [3] T. Suganda and D. Y. Wulandari, "Curvularia sp. Jamur Patogen Baru Penyebab Penyakit Bercak Daun pada Tanaman Sawi," *Agrikultura*, vol. 29, no. 3, 2019, doi: 10.24198/agrikultura.v29i3.22716.
- [4] N. K. Ayu Parweni, I. W. Getas, and S. Zaetun, "INFEKSI KECACINGAN NEMATODA USUS YANG DITULARKAN MELALUI TANAH (Soil Transmitted Helminth) PADA PETANI SAYUR SAWI HIJAU DI DESA BUG-BUG KECAMATAN LINGSAR KABUPATEN LOMBOK BARAT," *J. Anal. Med. Biosains*, vol. 5, no. 2, 2019, doi: 10.32807/jambs.v5i2.107.
- [5] L. Y. Rahayu, A. Mulyana, and U. Sunarya, "Perancangan Aplikasi Sistem Pemantauan Pertumbuhan Sawi Hijau Berbasis Web Dengan Computer Vision," *e-Proceeding Appl. Sci.*, vol. 4, no. 3, 2018.
- [6] R. Soekarta, N. Nurdjan, and A. Syah, "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN)," *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 2, 2023, doi: 10.33506/insect.v8i2.2356.
- [7] B. Yanto, Maria Angela Kartawidjaja, Ronald Sukwadi, and Marsellinus Bachtiar, "Implementation of Hue Saturation Intensity (Hsi) Color Space Transformation Algorithm With Red, Green, Blue (Rgb) Color Brightness in Assessing Tomato Fruit Maturity," *RJOCS (Riau J. Comput. Sci.*, vol. 9, no. 2, pp. 167–178, 2023, doi: 10.30606/rjocs.v9i2.2428.
- [8] B. Yanto, E. Rouza, L. Fimawahib, B. H. Hayadi, and R. R. Pratama, "Penerapan Algoritma Deep Learning Convolutional Neural Network Dalam Menentukan Kematangan Buah Jeruk Manis

- Berdasarkan Citra Red Green Blue (RGB),” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, 2023, doi: 10.25126/jtiik.20231015695.
- [9] B. Yanto, B. -, J. -, and B. H. Hayadi, “Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural Network (Cnn),” *JSAI (Journal Sci. Appl. Informatics)*, vol. 3, no. 3, pp. 106–114, 2020, doi: 10.36085/jsai.v3i3.1151.
- [10] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, “Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network,” *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 2, 2021, doi: 10.35314/isi.v6i2.2104.
- [11] M. A. Leonardi and A. Y. Chandra, “Analisis Perbandingan CNN dan Vision Transformer untuk Klasifikasi Biji Kopi Hasil Sangrai,” vol. 8, pp. 1398–1407, 2024, doi: 10.30865/mib.v8i3.7732.
- [12] B. Yanto, J. Jufri, A. Lubis, B. H. Hayadi, and E. Armita, NST, “KLARIFIKASI KEMATANGAN BUAH NANAS DENGAN RUANG WARNA HUE SATURATION INTENSITY (HSI),” *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 1, 2021, doi: 10.35314/isi.v6i1.1882.
- [13] N. A. Ayunda, E. Haryatmi, and T. A. Riyadi, “Classification of Tomato Ripeness Based on Convolutional Neural Network Methods,” *J. Inf. Syst. Informatics*, vol. 5, no. 4, 2023, doi: 10.51519/journalisi.v5i4.613.
- [14] Nengah Widya Utami and Joshael Juinor Soplantila, “USING SVM (SUPPORT VECTOR MACHINE) METHOD TO PREDICT BEST SELLER BOOK IN AMAZON,” *Smart Techno (Smart Technol. Informatics Technopreneurship)*, vol. 3, no. 1, 2021, doi: 10.59356/smart-techno.v3i1.28.
- [15] Y. H. Natbais and A. B. S. Umbu, “Aplikasi Deteksi Penyakit pada Daun Tomat Berbasis Android Menggunakan Model Terlatih Tensorflow Lite,” *TEKNOTAN*, vol. 17, no. 2, 2023, doi: 10.24198/jt.vol17n2.1.
- [16] K. P. Siwilopo and H. Marcos, “MEMBANDINGKAN KLASIFIKASI PADA BUAH JERUK MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN K-NEAREST NEIGHBOR,” *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 1, 2023, doi: 10.34010/komputa.v12i1.9068.
- [17] I. M. Z. Sukri and H. Rakhmad, “Penanganan Hama dan Penyakit Tanaman Jeruk Dalam Desain Sistem Pakar Diagnosis Penyakit Menggunakan Metode Euclidean Distance,” *Semin. Nas. Has. Penelit. dan Pengabd. Masy.*, 2016.