



## KLASIFIKASI PENYAKIT TANAMAN TOMAT MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DAN IMPLEMENTASI MODEL H5 PADA APLIKASI DESKTOP

Adhitya Jamalludin Bastari<sup>1</sup>, Anis Cherid<sup>2</sup>

<sup>1</sup>41519010032@student.mercubuana.ac.id, <sup>2</sup>anis.cherid@mercubuana.ac.id

<sup>1,2</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

### Abstrak

Penelitian ini mengembangkan metode budidaya tomat lebih efektif dengan analisis pola dan klasifikasi data citra menggunakan *Convolutional Neural Network* (CNN). Tujuan utamanya adalah mengklasifikasikan penyakit tomat dan jenis tanaman berdasarkan citra digital. Dengan menggunakan teknologi *Deep Learning*, informasi yang akurat dan mendalam tentang pertumbuhan dan kualitas tanaman tomat dapat diperoleh. Penelitian ini berhasil mencapai akurasi 99% dalam mengklasifikasikan citra daun tomat sehat, jamur septoria, jamur fulva, dan jamur target spot dengan model *Inception V3*. Perangkat lunak berbasis desktop yang dikembangkan mampu menampilkan hasil klasifikasi jenis daun secara spesifik.

**Kata kunci:** CNN, Klasifikasi, Tomat, InceptionV3

### Abstract

*This research develops a more effective method of tomato cultivation by pattern analysis and classification of image data using Convolutional Neural Network (CNN). The main objective is to classify tomato diseases and plant types based on digital images. By using Deep Learning technology, accurate and in-depth information about the growth and quality of tomato plants can be obtained. This research successfully achieved 99% accuracy in classifying images of healthy tomato leaves, septoria fungus, fulva fungus, and target spot fungus with the Inception V3 model. The desktop-based software developed is capable of displaying specific leaf type classification results.*

**Keywords:** CNN, Classification, Tomato, InceptionV3

### 1. Pendahuluan

Tomat (*Lycopersicon esculentum* Mill) merupakan tanaman hortikultura yang termasuk kedalam salah satu jenis tanaman sayuran sebagai sumber vitamin, mineral, serat serta memiliki kandungan senyawa lain seperti likopen yang berfungsi sebagai antioksidan berperan dalam menangkal radikal bebas yang bermanfaat bagi kesehatan manusia[1]. Tanaman tomat dapat tumbuh baik di tempat yang bersuhu panas, akan tetapi tomat memiliki suhu optimum untuk pertumbuhannya, sinar matahari yang berlebihan juga berpengaruh terhadap pertumbuhan dan hasil tanaman tomat. Salah satu bentuk modifikasi iklim mikro yang dapat membantu pertumbuhan dan hasil tanaman tomat yaitu dengan penggunaan naungan[1].

Data produksi tomat di Indonesia pada tahun 2018 sebesar 976.772 ton dan pada tahun 2019 meningkat menjadi 1.020.331 ton. Indonesia terus meningkat setiap tahun, konsumsi tomat oleh sektor rumah tangga tahun 2021 adalah mencapai 677,97 ribu ton, naik sebesar 6,93% (43,96 ribu ton) dari tahun 2020 (Badan Pusat Statistik, 2022).

Penyakit daun pada tanaman tomat sering menyebabkan penurunan hasil panen dan kerugian bagi petani. Penerapan teknologi *Deep Learning* untuk mendeteksi penyakit berpotensi menjadi solusi efektif. Namun, tantangan terkait kekurangan data latih, kompleksitas model, dan transferabilitas perlu diatasi agar teknologi ini dapat digunakan secara luas dan efisien dalam mendukung pertanian[2]. *Deep learning* merupakan suatu kelas dari teknik pembelajaran mesin yang memanfaatkan banyak lapisan dalam pemrosesan informasi untuk ekstraksi dan transformasi, serta untuk analisis pola dan klasifikasi terutama untuk memahami data seperti gambar, suara, dan teks[3].

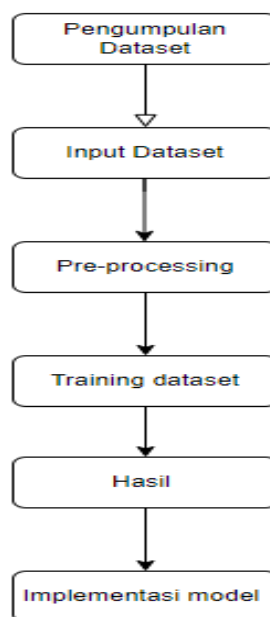
Penyakit tanaman adalah faktor penting dalam berbagai aspek kehidupan sehari-hari dan sektor-sektor seperti pertanian, kuliner, transportasi, dan kebutuhan di negara Indonesia. Dalam rangka meningkatkan pemahaman tentang ciri penyakit tanaman tomat dan kemampuan untuk melakukan klasifikasi pada ciri daun tomat kedepannya agar lebih baik, penulis akan berfokus pada penggunaan metode *Convolutional Neural Network* (CNN) dalam melakukan klasifikasi pada penyakit tanaman tomat dari ciri daun[4].

Pada penelitian ini juga membuat aplikasi klasifikasi berdasarkan citra digital untuk pencatatan aset menggunakan algoritma CNN dengan pustaka perangkat lunak *TensorFlow* dan mengetahui hasil akurasi yang didapatkan dari hasil klasifikasi menggunakan algoritma CNN dengan perangkat lunak *TensorFlow* [5]

*Convolutional Neural Network* (CNN) merupakan salah satu kelas dari *deep neural network*[6]. CNN adalah suatu metode *machine learning* yang dikembangkan dari *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data 2D[6]. Alasan memilih CNN untuk mendeteksi penyakit daun pada tanaman tomat yaitu karena CNN dirancang untuk data 2D, sehingga cocok untuk gambar daun seperti daun tomat, otomatis ekstraksi fitur dari gambar, struktur hierarkis untuk pengenalan pola bertahap, mengurangi *overfitting* dengan teknik seperti *dropout*, regularisasi, dan data augmentasi untuk mengatasi keterbatasan data latih.[7]

## 2. Metode

Pada penelitian ini menggunakan data citra berupa gambar daun tomat berjumlah 5632 data dari 4 *class* yang terdiri dari Daun Sehat, Daun Tomat Jamur Septoria, Daun Tomat Jamur Fulva, Daun Tomat Jamur Target Spot dan akan dilakukan klasifikasi menggunakan metode CNN. Hasil dari penelitian ini juga merupakan data numerik dengan menampilkan nilai akurasi dari algoritma tersebut. Berikut adalah alur dari penelitian yang dilakukan dapat dilihat pada Gambar 1.

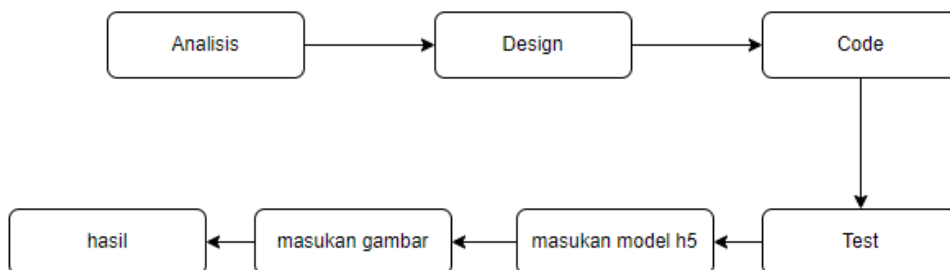


Gambar 1 Tahap Penelitian

Pada penelitian ini aplikasi dirancang dengan menggunakan antarmuka *Desktop GUI* lalu menggunakan modul python yaitu Tkinter, yang dimana modul Tkinter menyediakan sejumlah fungsi yang digunakan untuk membangun antarmuka grafis (GUI) seperti Tk, Frame, Label dan Button. Aplikasi pada penelitian ini nantinya akan menampilkan hasil gambar yang kita masukkan pada aplikasi tersebut

dan memprediksi apakah gambar daun yang dimasukkan masuk kedalam jenis daun tomat sehat, tomat jamur septoria, tomat jamur fulva atau daun tomat target spot [8].

Berikut merupakan alur pembuatan aplikasi desktop menggunakan model yang telah dilatih:

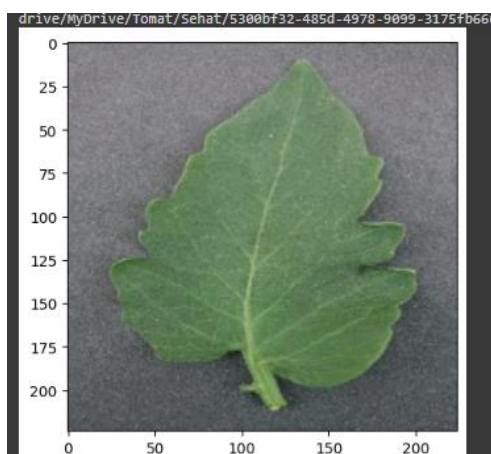


Gambar 2. Alur Penelitian dan Implementasi

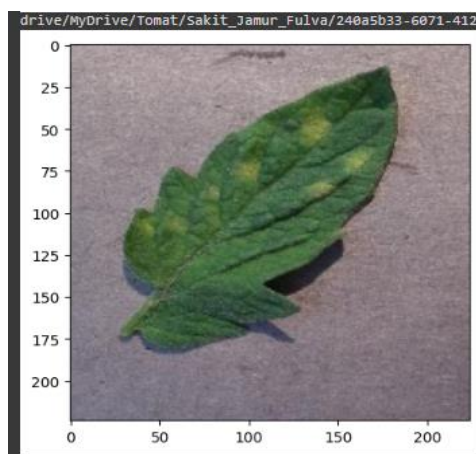
1. Latih model di Google colab menggunakan TensorFlow dan Keras dengan menggunakan dataset yang sudah dimiliki yaitu 5632 data dengan arsitektur model InceptionV3.
2. Setelah model selesai dilatih selanjutnya adalah menyimpan model dalam format .h5.
3. Setelah model berhasil disimpan dan diunduh selanjutnya adalah mentransfer model ke komputer lokal.
4. Lalu integrasi model dengan aplikasi desktop menggunakan Desktop GUI dan mengimpor *library* yang digunakan yaitu Keras untuk membuat antarmuka grafis pengguna.
5. Selanjutnya setelah model aplikasi jalan masukan model h5 di dalam aplikasi desktop untuk memuat model .h5 yang telah di transfer dari file menggunakan `load_model('./keras_model.h5')` untuk memuat model dari file h5.
6. Setelah berjalan load selesai masukan gambar dari file tomat yang digunakan sebelumnya untuk di klasifikasikan.
7. Lalu aplikasi akan memprediksi apakah gambar tersebut termasuk dari ciri daun sehat, atau terkena penyakit jamur.

## 2.1 Pengumpulan Data

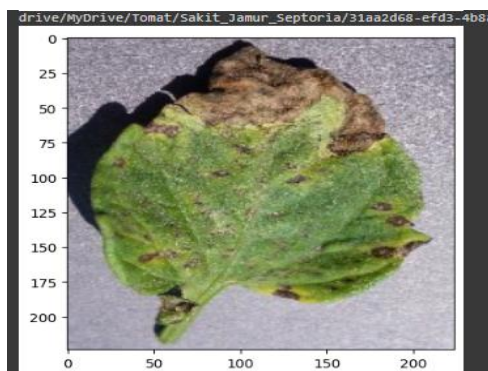
Pada Penelitian ini data yang digunakan berasal dari kaggle. Data yang terdiri dari berbagai macam daun tomat yang didalamnya berisikan Daun Sehat, Daun Tomat Jamur Septoria, Daun Tomat Jamur Fulva, Daun Tomat Jamur Target Spot dengan jumlah total data adalah 5632 data. Terbagi menjadi empat data yaitu Daun Tomat Sehat, Daun Tomat Jamur Septoria, Daun Tomat Jamur Fulva, Daun Tomat Jamur Target Spot, dapat dilihat pada gambar.



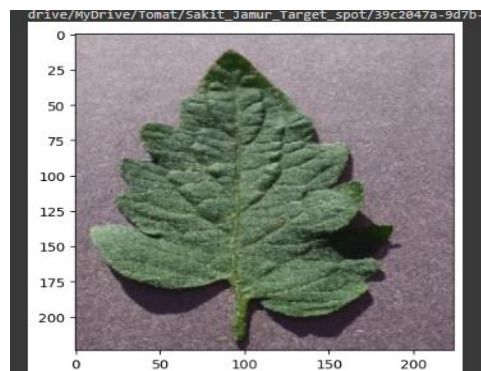
Gambar 3. Daun Sehat



Gambar 4. Sakit Jamur Fulva



Gambar 5. Sakit Jamur Septoria



Gambar 6. Sakit Jamur Target Spot

## 2.2 Data Preprocessing

Dataset di bagi menjadi *train* dan *test* biasanya dilakukan untuk memastikan bahwa model yang telah dibangun dapat bekerja dengan baik pada data yang belum pernah dilihat sebelumnya. Dengan memisahkan beberapa data ke dalam *train* dan *test*, kita dapat mengetahui seberapa baik model tersebut dapat memprediksi *output* untuk data baru. Dataset juga di *resize* agar dapat membantu meningkatkan akurasi model. Jika *dataset* memiliki ukuran yang berbeda-beda, maka model mungkin akan kesulitan untuk menemukan pola yang sama pada data yang berbeda ukuran. Dengan menyesuaikan ukuran data ke ukuran yang sama, model akan lebih mudah untuk mempelajari pola-pola yang ada pada data, sehingga dapat memberikan hasil yang lebih akurat. Pada penelitian ini *dataset* di *resize* agar menyesuaikan *input pre-trained* model *InceptionV3* menjadi 224x224[9].

## 2.3 Pembuatan Model

Model yang digunakan pada penelitian ini adalah arsitektur *InceptionV3* dari *library keras* yang merupakan salah satu arsitektur *Convolutional Neural Network (CNN)*.

Tabel 1. Code Pembuatan Model InceptionV3

```
import tensorflow as tf
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras import layers, models
import datetime

model = InceptionV3(input_shape=(tinggi_gambar, luas_gambar, warna), weights='imagenet',
include_top=False)
x = model.output
x = layers.Dropout(0.2)(x)
x = layers.Flatten()(x)
predictions = layers.Dense(len(nama_class), activation='softmax',
kernel_regularizer=tf.keras.regularizers.L2(0.01))(x)

model = models.Model(inputs=model.input, outputs=predictions)

# Set the layers to be trainable
for layer in model.layers[:151]:
    layer.trainable = False
for layer in model.layers[151:]:
    layer.trainable = True
```

---

Code tersebut merupakan implementasi penggunaan model InceptionV3 dari TensorFlow untuk melakukan klasifikasi gambar. Model ini digunakan untuk mengklasifikasikan jenis daun tomat berdasarkan citra digital. Tahapan code tersebut adalah sebagai berikut:

1. Mengimport *library* yang diperlukan, yaitu TensorFlow dan beberapa komponen seperti InceptionV3, *layers*, dan *models*.
2. Membangun model InceptionV3 dengan menentukan *input\_shape* gambar, bobot yang digunakan adalah "imagenet", dan tidak menyertakan layer teratas (*include\_top=False*).
3. Melakukan *pre-processing* pada output model InceptionV3 dengan menambahkan *Dropout* layer untuk mencegah *overfitting*, kemudian melakukan Flatten untuk mengubah output menjadi vektor 1 dimensi.
4. Menambahkan *Dense layer* sebagai output layer dengan menggunakan fungsi aktivasi softmax, yang menghasilkan probabilitas untuk masing-masing kelas daun tomat.
5. Membuat model baru yang mengambil input dari model InceptionV3 dan mengeluarkan output dari layer *Dense* yang telah ditambahkan.
6. Menetapkan *trainability* untuk lapisan model, dengan menyimpan lapisan-lapisan awal yang tidak dapat di *train* dan mengizinkan lapisan-lapisan berikutnya untuk di *train*.
7. Code ini dapat digunakan untuk melatih model dengan dataset gambar daun tomat yang telah dijelaskan sebelumnya.

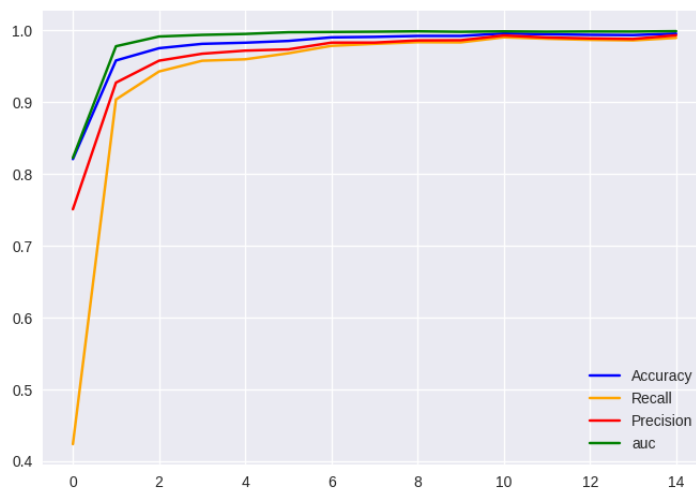
## 2.4 InceptionV3

Tahap selanjutnya adalah membuat arsitektur CNN untuk klasifikasi dataset citra daun tomat, pada tahap ini menggunakan *library TensorFlow keras* untuk memanggil *pre-trained* model yang bernama InceptionV3.

*InceptionV3* adalah arsitektur pertama dengan parameter yang lebih sedikit dan komputasi yang efisien [10]. Pada penelitian ini penulis menggunakan arsitektur *InceptionV3* pada pengerjaannya. *InceptionV3* adalah sebuah model *deep convolutional neural network* yang dikembangkan oleh Google untuk memenuhi *ImageNet Large Visual Recognition Challenge* pada tahun 2012. Inception sendiri memiliki empat versi, yaitu *InceptionV1*, *InceptionV2*, *InceptionV3*, dan *InceptionV4* [10].

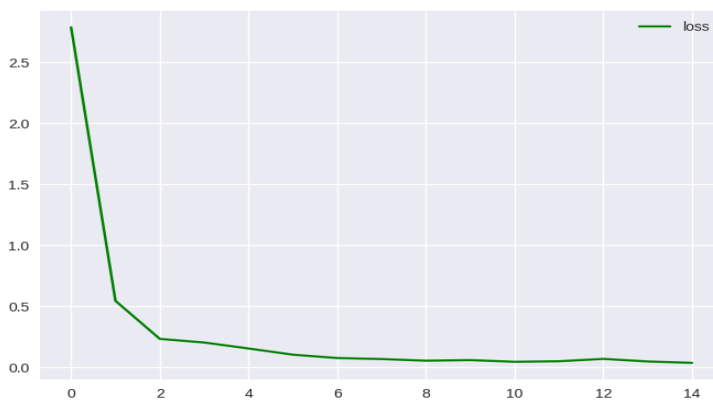
## 3. Hasil dan Pembahasan

Untuk data yang sudah dibagi menjadi data *train* dan data *test* kini mendapatkan hasil dengan menggunakan *epoch*. *Epoch* merupakan iterasi melalui seluruh dataset pelatihan yang digunakan untuk melatih model. Pada penelitian ini *epoch* yang digunakan adalah 15. Selama proses *fitting*, model akan melalui iterasi sebanyak 15 *epoch* melalui *dataset* pelatihan. Pada setiap *epoch*, model akan memperbarui parameter berdasarkan gradien yang dihitung dari *batch* sampel yang dievaluasi. *Callbacks* yang telah ditentukan akan dieksekusi pada setiap *epoch* sesuai dengan kebutuhan, seperti melakukan *EarlyStopping* jika terjadi penurunan signifikan dalam *loss* atau menyimpan model terbaik. Dengan menggunakan kode tersebut, model akan dilatih pada dataset pelatihan dengan jumlah *epoch* yang ditentukan, dan *callback* akan dijalankan pada setiap *epoch* sesuai dengan kebutuhan.



Gambar 7. Visualisasi Accuracy

Gambar 7 menunjukkan hasil dari *training* data dari iterasi 1 sampai 15 dan terjadi kenaikan grafik menuju nilai 1. Dari grafik di atas dapat disimpulkan bahwa konfigurasi model yang telah digunakan telah benar, semakin tinggi akurasi suatu model maka akan semakin baik.



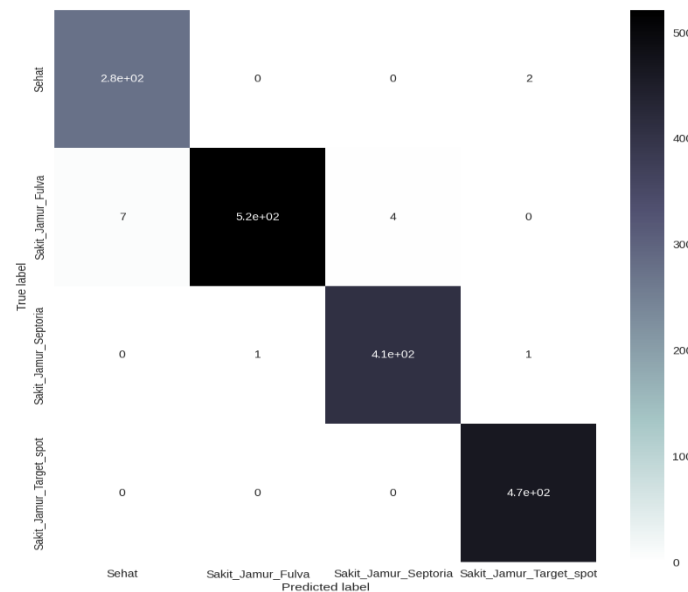
Gambar 8. Visualisasi Loss

Gambar 8 menunjukkan hasil training loss dari iterasi 1 sampai 15 dan terjadi penurunan yang sangat tajam dan akhirnya melandai. Dari grafik di atas dapat disimpulkan bahwa pada saat proses training data variable *loss* / kesalahan dalam klasifikasi data turun secara drastis.

Tabel 2. Hasil *Confusion Matrix*

	Sehat	Sakit_Jamur_Fulva	Sakit_Jamur_Septoria	Sakit-Jamur_Target_Spot
Sehat	278	0	0	2
Sakit_Jamur_Fulva	7	521	4	0
Sakit_Jamur_Septoria	0	1	408	1
Sakit_Jamur_Target_spot	0	0	0	466

Tabel 2 menunjukan hasil berbentuk *confusion matrix* yang di dapatkan dari hasil prediksi data *testing* yang telah dibagi sebelum melakukan proses *training* data. Sedangkan Gambar 9 dibawah adalah visualisasi *confusion matrix* menggunakan *library seaborn*.



Gambar 9. Confution Matrix Seaborn

Hasil dari *confusion matrix* pada Gambar 9 menunjukkan kinerja model klasifikasi terhadap empat kelas yang berbeda, yaitu "Sehat", "Sakit\_Jamur\_Fulva", "Sakit\_Jamur\_Septoria", dan "Sakit\_Jamur\_Target\_Spot". Berikut adalah interpretasi singkat dari hasil tersebut:

1. Sebanyak 278 data kelas "Sehat" berhasil diprediksi dengan benar sebagai "Sehat" oleh model (*True Positive*).
2. Ada 2 data yang sebenarnya "Sehat", namun model salah memprediksi sebagai "Sakit\_Jamur\_Fulva" (*False Negative*).
3. Sebanyak 521 data kelas "Sakit\_Jamur\_Fulva" berhasil diprediksi dengan benar sebagai "Sakit\_Jamur\_Fulva" oleh model (*True Positive*).
4. Ada 7 data yang sebenarnya "Sakit\_Jamur\_Fulva", namun model salah memprediksi sebagai "Sehat" (*False Positive*).
5. Sebanyak 408 data kelas "Sakit\_Jamur\_Septoria" berhasil diprediksi dengan benar sebagai "Sakit\_Jamur\_Septoria" oleh model (*True Positive*).
6. Ada 1 data yang sebenarnya "Sakit\_Jamur\_Septoria", namun model salah memprediksi sebagai "Sakit\_Jamur\_Fulva" (*False Negative*).
7. Sebanyak 466 data kelas "Sakit\_Jamur\_Target\_Spot" berhasil diprediksi dengan benar sebagai "Sakit\_Jamur\_Target\_Spot" oleh model (*True Positive*).

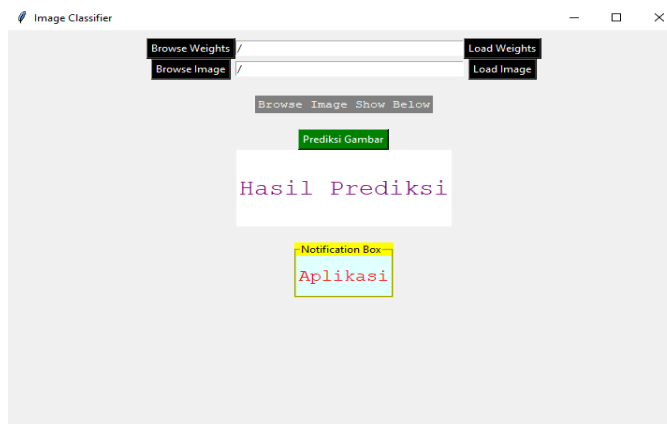
	precision	recall	f1-score	support
Sehat	0.99	0.98	0.98	285
Sakit_Jamur_Fulva	0.98	1.00	0.99	522
Sakit_Jamur_Septoria	1.00	0.99	0.99	412
Sakit_Jamur_Target_spot	1.00	0.99	1.00	469
accuracy			0.99	1688
macro avg	0.99	0.99	0.99	1688
weighted avg	0.99	0.99	0.99	1688

Gambar 10. Hasil Klasifikasi

Pada Gambar 10 di atas dapat juga disimpulkan bahwa hasil dari klasifikasi penyakit tanaman tomat dari ciri daun dengan jumlah *class* sebanyak 4 *class* menggunakan algoritma CNN inception v3. didapatkan hasil akurasi sebesar 99%, *Precision* 99%, *Recall* 99%, dan F1-score sebanyak 99%.

#### 4. Implementasi Model pada Aplikasi Berbasis Desktop

Berikut adalah aplikasi desktop yang digunakan untuk melakukan implementasi model h5 yang telah dibuat sebelumnya:



Gambar 11. Visualisasi Aplikasi Desktop

Aplikasi berikut memiliki beberapa fitur yang dapat digunakan untuk mengklasifikasi sebuah gambar penyakit dari ciri daun tomat. Berikut adalah penjelasan setiap fitur pada aplikasi:

1. Fitur Import Model: fitur ini digunakan untuk memasukkan model yang telah di buat sebelumnya dengan format h5.
2. Fitur Import Gambar: fitur ini digunakan untuk memilih gambar yang ingin di prediksi berdasarkan ciri daunnya.
3. Tombol Prediksi: fitur ini diguankan untuk melakukan prediksi pada gambar yang telah di inputkan kedalam aplikasi.
4. Display Box: digunakan untuk memvisualisasikan hasil dari gambar yang telah di prediksi.

Berikut adalah cara penggunaan aplikasi prediksi daun tomat. Hasil prediksi akan di visualisasikan pada *display box* yang ada di aplikasi.



Gambar 12. Hasil Klasifikasi Aplikasi Desktop

## 5. Kesimpulan Dan Saran

Penelitian ini mengembangkan sistem klasifikasi otomatis untuk mengidentifikasi jenis penyakit daun tomat berdasarkan ciri daun menggunakan algoritma CNN dengan arsitektur Inception V3. Model yang dikembangkan mencapai akurasi 99% dalam mengklasifikasikan jenis penyakit tomat. Implementasi aplikasi desktop dengan model ini dapat membantu mengurangi kesalahan identifikasi penyakit daun tomat dalam industri pertanian. Penelitian ini juga memberikan peluang pengembangan dan peningkatan model CNN untuk meningkatkan akurasi dalam mengklasifikasikan jenis penyakit tomat dari ciri daun.

Saran yang diusulkan meliputi pengembangan sistem klasifikasi otomatis untuk mengidentifikasi jenis penyakit tanaman tomat dari ciri daun sebagai solusi efektif dalam industri pertanian. Disarankan juga melakukan eksplorasi dan peningkatan model CNN, seperti Inception V3, untuk meningkatkan performa dan akurasi dalam mengklasifikasikan jenis penyakit tanaman tomat berdasarkan ciri daun.

## Daftar Pustaka

- [1] V.A.R.Barao, R.C.Coata, J.A.Shibli, M.Bertolini, and J.G.S.Souza, "Analisis struktural kovarians pada indikator yang berhubungan dengan kesehatan di antara para lansia di rumah, dengan fokus pada perasaan subjektif tentang kesehatan.Title," *Braz Dent J.*, vol. 33, no. 1, pp. 1–12, 2022.
- [2] N. C. P. Rahayu, "Perbedaan Tanaman Buah Tomat (*Lycopersiconesculentum*), Cabai (*Capsicumfrutencens L.*), dan Terong (*SolanummelongenaL.*) Pada Penyerapan Amonia (NH3), NITRIT (NO2) dan NITRAT (NO3) Air Budidaya Ikan Lele Dumbo (*Clariassp.*) Pada Sistem Akuaponik." Universitas Airlangga, 2019.
- [3] A. Peryanto, A. Yudhana, and R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Format : Jurnal Ilmiah Teknik Informatika*, vol. 8, no. 2, p. 138, 2020, doi: 10.22441/format.2019.v8i2.007.
- [4] M. Cosovic and R. Jankovic, "CNN Classification of the Cultural Heritage Images," *2020 19th International Symposium INFOTEH-JAHORINA, INFOTEH 2020 - Proceedings*, no. March, 2020, doi: 10.1109/INFOTEH48170.2020.9066300.
- [5] I. Maryati, "Website Perpustakaan 'Library HUB' dengan Pencarian Buku Berdasarkan Gambar Menggunakan Google MLKit," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 4, pp. 1821–1831, 2021, doi: 10.35957/jatisi.v8i4.1269.
- [6] M. Fasounaki, E. B. Yüce, S. Öncül, and G. Ince, "CNN-based Text-independent Automatic Speaker Identification Using Short Utterances," *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, vol. 01, pp. 413–418, 2021, doi: 10.1109/UBMK52708.2021.9559031.
- [7] S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, p. 76, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>
- [8] T. M. Kadarina and M. H. Ibnu Fajar, "Pengenalan Bahasa Pemrograman Python Menggunakan Aplikasi Games Untuk Siswa/I Di Wilayah Kembangan Utara," *Jurnal Abdi Masyarakat (JAM)*, vol. 5, no. 1, p. 11, 2019, doi: 10.22441/jam.2019.v5.i1.003.
- [9] Wahyutama Fitri Hidayat and Taufik Asra, "Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression," *Indonesian Journal on Software Engineering*, vol. 8, no. 2, pp. 173–179, 2022.
- [10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.