



## PENERAPAN *CONVOLUTIONAL NEURAL NETWORK* PADA KLASIFIKASI TANAMAN MENGGUNAKAN *RESNET50*

Umi Kulsum<sup>1</sup>, Anis Cherid<sup>2</sup>

<sup>1</sup>41519010021@student.mercubuana.ac.id, <sup>2</sup>anis.cherid@mercubuana.ac.id  
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

### Abstrak

Penelitian ini bertujuan untuk mengklasifikasikan tanaman apel menjadi dua kategori, yaitu sehat dan busuk, serta menciptakan perangkat lunak berbasis *Desktop* yang dapat mengklasifikasikan jenis tanaman berdasarkan citra digital menggunakan *Convolutional Neural Network* (CNN) dengan model *ResNet50*. Data yang digunakan untuk melatih model terdiri dari 1545 gambar, sedangkan data uji yang digunakan berjumlah 661 gambar. Proses klasifikasi dilakukan untuk dua kelas, yaitu citra daun apel sehat dan citra daun apel busuk. Hasil evaluasi model menggunakan *Confusion Matrix* menunjukkan akurasi sebesar 91% setelah proses pelatihan dilakukan selama 50 epoch pada data latih. Selanjutnya, aplikasi perangkat lunak berbasis *Desktop* dibuat untuk menampilkan hasil jenis daun apel yang dipilih, dengan menampilkan informasi apakah daun tersebut termasuk dalam kategori sehat atau busuk.

**Kata Kunci:** *Convolutional Neural Network, ResNet50, Daun Apel*

### Abstract

*This research aims to classify apple plants into two categories, which are healthy and diseased, and create a Desktop-based software that can classify plant types based on digital images using Convolutional Neural Network (CNN) with ResNet50 model. The training data consists of 1545 images, while the test data consists of 661 images. The classification process is performed for two classes, namely healthy apple leaf images and diseased apple leaf images. The evaluation results of the model using the Confusion Matrix show an accuracy of 91% after the training process with 50 epochs on the training data. Subsequently, a Desktop-based software application is developed to display the selected apple leaf type and provide information about whether the leaf belongs to the healthy or diseased category.*

**Keywords:** *Convolutional Neural Network, ResNet50, Apple Leaf*

### 1. Pendahuluan

Tanaman apel atau nama ilmiah *malus sylvestris* adalah tanaman tahunan yang dapat ditanam di wilayah *subtropis* dan di Indonesia. Buah apel adalah salah satu buah yang paling produktif di dunia karena nilai gizi dan manfaat obat yang tinggi. Menurut data yang dirilis oleh Kementerian Pertanian Republik Indonesia dari tahun 2013 hingga 2017, produksi apel di Indonesia mencapai 319.000 ton pada tahun 2017. Namun, produksi turun 3,3%, atau sekitar 10.780 ton, dari tahun sebelumnya, ketika produksi apel pada tahun 2016 mencapai 329.780 ton. Penurunan ini disebabkan oleh berbagai penyakit tanaman apel yang sering terjadi. Oleh karena itu, penemuan penyakit daun apel sangat penting untuk menjaga pertumbuhan apel sehat [1].

Untuk mengidentifikasi jenis penyakit pada daun apel, digunakan metode pengklasifikasian gambar menggunakan *Convolutional Neural Network* (CNN) [2]. CNN adalah kombinasi dari jaringan saraf tiruan dan metode *deep learning*. Struktur CNN terdiri dari satu atau lebih lapisan *konvolusional*, yang biasanya diikuti oleh lapisan *subsampling*, dan dilanjutkan dengan satu atau lebih lapisan terhubung penuh sebagai bagian dari jaringan saraf. Dalam prosesnya, CNN akan dilatih dan diuji menggunakan gambar daun apel sehingga akan menghasilkan model yang mampu mengklasifikasikan jenis penyakit pada daun apel [3].

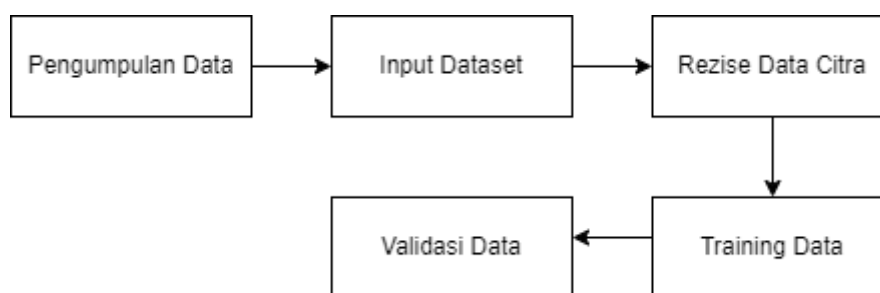
Berdasarkan penelitian sebelumnya yang berjudul "Klasifikasi Penyakit Daun Kentang Menggunakan Model *Regresi Logistik*," dapat disimpulkan bahwa dengan menggunakan ekstraksi fitur menggunakan *ResNet50* dan klasifikasi *Regresi Logistik*, sistem mampu mengidentifikasi penyakit yang terdapat pada daun kentang. Penelitian ini menggunakan dataset dalam bentuk citra dengan total 405 citra

daun kentang. Hasil pengujian mengelompokkan dataset ke dalam tiga kategori, yaitu *Earliy\_Blight*, *Healthy*, dan *Late\_Blight*. Tingkat akurasi yang berhasil dicapai dalam penelitian ini adalah sebesar 80% [4].

Pada penelitian ini, digunakan model *ResNet50* sebagai transfer *learning* yang sebelumnya telah dilatih dengan dataset besar dan beragam. Dengan memanfaatkan transfer *learning*, pengetahuan yang telah dipelajari oleh model tersebut dapat digunakan untuk tugas-tugas khusus pengenalan citra. Metode pembelajaran transfer memungkinkan penggunaan dataset yang lebih kecil untuk melatih model, menghemat waktu dan sumber daya yang dibutuhkan untuk melatih model sejak awal. Selain itu, penelitian ini mengembangkan aplikasi klasifikasi berdasarkan citra digital untuk mencatat aset. Ini melakukannya dengan menggunakan algoritma *Convolutional Neural Network* (CNN) yang tersedia dalam *TensorFlow*. Penelitian telah menunjukkan bahwa hasil klasifikasi CNN dapat dievaluasi dengan akurat [5].

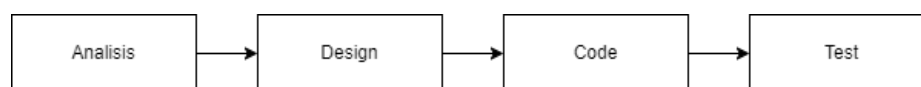
## 2. Metode

Penelitian ini akan menggunakan 2206 gambar daun apel untuk melakukan klasifikasi menggunakan metode *Convolutional Neural Network* (CNN). Hasil penelitian ini juga akan menunjukkan nilai akurasi algoritma yang digunakan. Semua data numerik yang dikumpulkan dari penelitian ini akan ditunjukkan. Gambar 1 menunjukkan rute penelitian ini.



Gambar 1. Metode Penelitian

Pada penelitian ini, telah dikembangkan sebuah aplikasi berbasis antarmuka grafis desktop (GUI) menggunakan modul Python yang disebut Tkinter. Modul Tkinter menyediakan berbagai fungsi yang umum digunakan untuk membangun antarmuka grafis, seperti Tk, Frame, Label, dan Button [6]. Aplikasi ini bertujuan untuk menampilkan hasil gambar yang dimasukkan oleh pengguna dan melakukan prediksi apakah gambar daun apel tersebut termasuk dalam jenis daun apel yang sehat atau daun apel yang busuk. Pengguna dapat memasukkan gambar daun apel ke dalam aplikasi, dan aplikasi akan memberikan hasil prediksi sesuai dengan model yang telah dilatih sebelumnya. Sebuah diagram yang menjelaskan proses pembuatan aplikasi desktop menggunakan model tersebut juga disajikan.



Gambar 2. Alur Pembuatan Aplikasi *Desktop*

1. Latih model menggunakan TensorFlow dan Keras di Google Colab menggunakan dataset yang terdiri dari 2206 data. Arsitektur model yang digunakan adalah *ResNet50*.
2. Setelah proses pelatihan selesai, model disimpan dalam format .h5 untuk dijadikan file model yang dapat digunakan kembali.
3. File model .h5 yang telah disimpan diunduh dari Google Colab dan selanjutnya ditransfer ke komputer lokal.
4. Model yang telah dilatih diintegrasikan ke dalam aplikasi *Desktop* menggunakan *Desktop GUI Library* yang digunakan, yaitu Keras, membantu dalam pembuatan antarmuka grafis yang akan digunakan oleh pengguna.

5. Model .h5 yang telah ditransfer digunakan dalam aplikasi *Desktop* dengan memuatnya menggunakan *Library Tkinter* GUI. Proses ini dilakukan dengan menggunakan fungsi `load_model('./keras_model.h5')` untuk memuat model dari file .h5.
6. Setelah model berhasil dimuat, model siap digunakan dalam aplikasi *Desktop*. Pengguna dapat memberikan input gambar daun apel ke model dan memperoleh prediksi atau hasil klasifikasi apakah gambar tersebut termasuk dalam jenis daun apel yang sehat atau daun apel yang busuk.

## 2.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari Kaggle, dan terdiri dari berbagai macam daun apel, termasuk daun sehat dan daun busuk. Total 2206 data dibagi menjadi dua bagian, yaitu daun apel sehat (gambar 2) dan daun apel busuk (gambar 3).



Gambar 2. Daun Apel Sehat



Gambar 3. Daun Apel Busuk

## 2.2. Data Preprocessing

Pembagian dataset menjadi data latih (*train*) dan data uji (*test*) dilakukan untuk memastikan bahwa model yang telah dibangun dapat menghasilkan prediksi yang baik pada data yang belum pernah dilihat sebelumnya. Dengan cara ini, kita dapat mengevaluasi kemampuan model untuk memprediksi output dengan akurasi yang tinggi pada data baru. Selain itu, untuk meningkatkan akurasi model, dataset mengalami perubahan. Ukuran dataset yang berbeda-beda dapat menyulitkan model untuk menemukan pola yang konsisten pada data dengan ukuran yang beragam. Dengan mengubah ukuran seluruh data menjadi 224 x 224 piksel, model akan lebih efektif dalam mengidentifikasi pola-pola yang ada pada data, sehingga memberikan hasil prediksi yang lebih akurat. Dataset penelitian telah disesuaikan dengan ukuran input model ResNet50 yang telah dilatih sebelumnya agar memiliki ukuran 224 x 224 piksel.

## 2.3. Pembuatan Model

Penelitian ini menggunakan model *ResNet50*, yang merupakan salah satu arsitektur *Convolutional Neural Network (CNN)* [7].

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras import layers, models
import datetime

model = ResNet50(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
```

Gambar 4. *ResNet50*

---

Perintah kode diatas memerintakan Impor *Library TensorFlow* dan modul *ResNet50* dari *'tensorflow.keras.applications'* *ResNet50* adalah model *pre-trained ResNet-50* yang telah disediakan oleh *TensorFlow*. Selanjutnya, objek model *ResNet50* dibuat dengan memberikan argumen *input\_shape* yang menentukan dimensi input gambar (*tinggi\_gambar*, *luas\_gambar*, *warna*), *weights='imagenet'* untuk menggunakan bobot *pre-trained* dari dataset *ImageNet*, dan *include\_top=False* untuk menghilangkan lapisan penuh (*fully connected*) di bagian akhir model. Output dari model *ResNet-50* kemudian disimpan dalam variabel *x*. Lalu dilakukan penambahan lapisan *Dropout* dengan tingkat *dropout* 0.2 untuk mengurangi *overfitting* pada model. Kemudian, lapisan *Flatten* digunakan untuk mengubah output dari model menjadi vektor satu dimensi. Lalu dilakukan penambahan lapisan *Dense* sebagai output layer dengan jumlah *neuron* yang sesuai dengan jumlah kelas (*len(nama\_class)*) dan fungsi aktivasi *softmax* untuk menghasilkan probabilitas prediksi kelas. Model akhir dibuat menggunakan *models*. Model dengan input dari model *ResNet-50* dan output dari lapisan prediksi yang telah ditambahkan. Selanjutnya, dilakukan pengaturan agar sebagian lapisan (*layer*) pada model tidak dapat dilatih (*trainable*) dengan mengatur atribut *trainable* pada lapisan-lapisan tersebut:

1. Lapisan-lapisan sebelum lapisan ke-151 (indeks 0 hingga 150) diatur menjadi tidak dapat dilatih dengan mengatur *layer.trainable = False*.
2. Lapisan-lapisan setelah lapisan ke-151 (indeks 151 dan seterusnya) diatur agar dapat dilatih dengan mengatur *layer.trainable = True*.

Dengan mengikuti langkah-langkah tersebut, kode di atas menghasilkan model yang menggunakan arsitektur *ResNet-50* dengan lapisan tambahan untuk tugas klasifikasi gambar. Lapisan-lapisan *ResNet-50* awal akan menggunakan bobot *pre-trained*, sedangkan lapisan-lapisan tambahan akan dilatih pada dataset khusus.

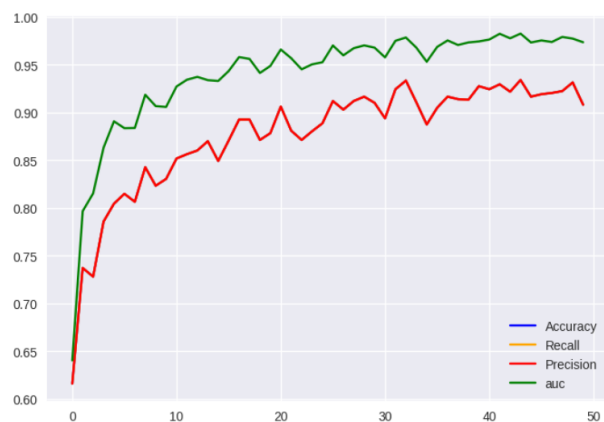
#### 2.4. *ResNet50*

Pada langkah berikutnya, arsitektur CNN untuk klasifikasi dataset citra daun apel dibuat. Pada tahap ini, *Library TensorFlow* yang kuat digunakan untuk memanggil model yang telah dilatih sebelumnya yang disebut *ResNet50*.

*ResNet-50* adalah salah satu model arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh *Microsoft Research* pada tahun 2015 [8]. Nama "*ResNet*" merupakan singkatan dari "*Residual Network*" yang mengacu pada penggunaan blok residu (*residual block*) dalam arsitektur ini. *ResNet-50* terdiri dari 50 lapisan (*depth*) dan merupakan versi tengah dari keluarga model *ResNet* yang lebih besar, seperti *ResNet-101* dan *ResNet-152* [9]. Model ini sangat populer dan sering digunakan dalam berbagai tugas penglihatan komputer, seperti klasifikasi gambar, deteksi objek, dan segmentasi citra. Arsitektur *ResNet-50* secara umum terdiri dari lapisan konvolusi awal, diikuti oleh beberapa blok residu. Setiap blok residu terdiri dari beberapa lapisan konvolusi dengan *shortcut connection*. Setelah blok residu terakhir, terdapat lapisan *pooling* global untuk mengurangi dimensi spasial fitur-fitur yang dihasilkan. Kemudian, fitur-fitur tersebut dihubungkan ke lapisan *fully connected* yang menghasilkan output klasifikasi [10].

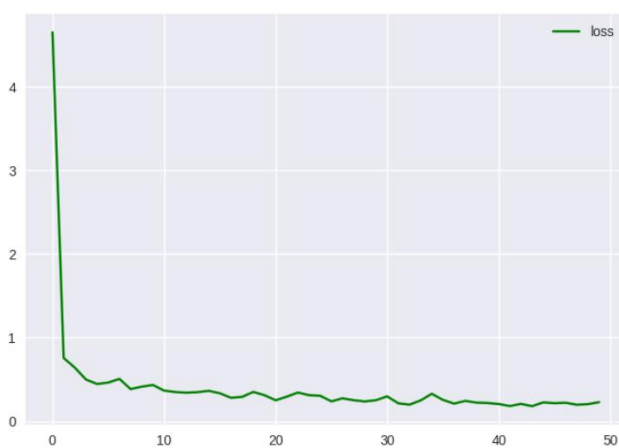
### 3. Hasil dan Pembahasan

Data yang telah dibagi menjadi data pelatihan dan data uji akan diolah dengan menggunakan epoch. Epoch merupakan satu kali proses pelatihan model dimana seluruh paket pelatihan yang digunakan untuk mengajar model akan diulang sebanyak periode yang telah ditentukan, dalam penelitian ini adalah lima puluh epoch [11]. Selama proses *fitting* (pelatihan), model akan mengulangi dataset pelatihan sebanyak lima puluh kali, dan pada setiap epoch, parameter model akan diperbarui berdasarkan gradien dari batch sampel yang dievaluasi. Dengan menggunakan pendekatan ini, model akan dilatih pada dataset pelatihan dengan jumlah epoch yang telah ditentukan, dan pada setiap epoch, callback yang telah ditentukan (misalnya, melakukan *Early Stopping* jika terjadi penurunan besar) akan dieksekusi.

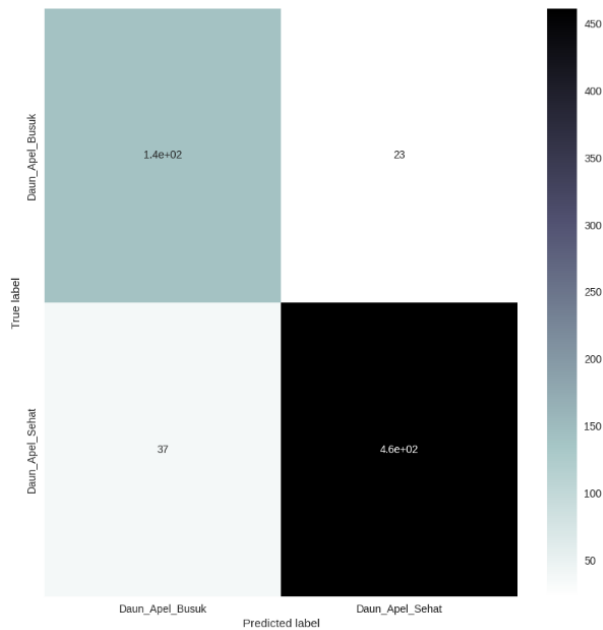


Gambar 5. Plot Nilai Akurasi

Grafik pada Gambar 5 merepresentasikan perhitungan *precision* pada setiap epoch selama proses *fitting*. Lalu menampilkan nilai akurasi auc (*Area Under the Curve*). Data auc ini merepresentasikan perhitungan auc pada setiap epoch selama proses *fitting*. Plot nilai *loss* dapat dilihat pada Gambar 6, dan *Confusion Matrix* dapat dilihat pada Gambar 7.



Gambar 6. Plot Nilai Loss

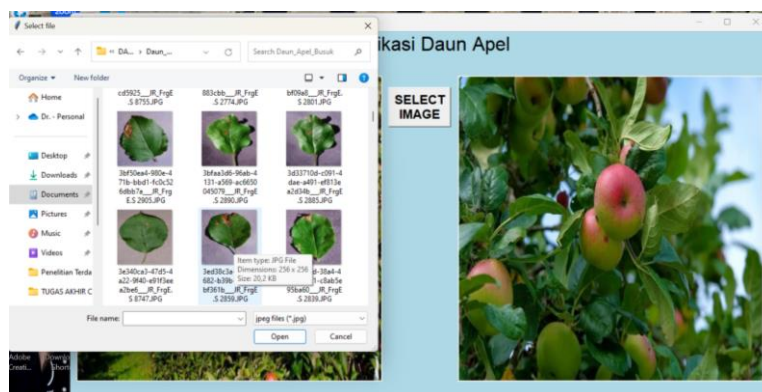


Gambar 7. Confusion Matrix

	precision	recall	f1-score	support
Daun_Apel_Busuk	0.86	0.79	0.82	177
Daun_Apel_Sehat	0.93	0.95	0.94	484
accuracy			0.91	661
macro avg	0.89	0.87	0.88	661
weighted avg	0.91	0.91	0.91	661

Gambar 8. Hasil Klasifikasi

Dari Gambar 8 dapat disimpulkan bahwa hasil dari klasifikasi daun apel dengan jumlah 2 kelas menggunakan algoritma CNN *ResNet50* mendapatkan hasil *akurasi*, *precision*, *recall* dan *f1-score* sebesar 91%.



Gambar 9. Pemilihan Gambar

Pada Gambar 9 diatas dilakukan pengujian untuk pemilihan daun untuk mengetahui hasil daun tersebut busuk atau sehat.



Gambar 10. Daun Dipilih

Telah dilakukan pemilihan gambar, selanjutnya mengenali apakah daun yang dimasukkan termasuk ke dalam jenis daun sehat atau busuk hasil dapat dilihat pada Gambar 11.



Gambar 11. Hasil Daun Busuk

Daun apel yang ditunjukkan pada Gambar 11 yang telah dipilih menunjukkan bahwa daun yang dipilih termasuk kedalam jenis daun busuk. Selanjutnya akan di uji coba memasukkan daun yang dianggap sehat lalu melihat hasil akurasi menurut aplikasi, dapat dilihat pada Gambar 12.



Gambar 12. Hasil Daun Sehat

#### 4. Kesimpulan dan Saran

Berdasarkan hasil penelitian yang telah dilakukan dengan melalui tahap *pre-processing* data dan pembuatan model CNN dengan menggunakan *ResNet50*. Maka dapat disimpulkan bahwa penggunaan CNN menggunakan model *ResNet50* sangat efektif karena dapat melakukan klasifikasi dan prediksi pada data citra daun apel dengan 2 jenis daun yaitu daun apel sehat dan daun apel busuk dengan menghasilkan nilai akurasi sebesar 91%. Hasil penelitian ini juga telah menghasilkan aplikasi *Desktop* yang dimana aplikasi tersebut dapat mengklasifikasi dan memprediksi jenis daun yang dimasukkan pada aplikasi tersebut.

Pada penelitian yang telah dilakukan diharapkan untuk penelitian selanjutnya untuk dilakukan pengembangan seperti mengumpulkan data lebih luas lagi, karena dataset citra yang digunakan pada penelitian ini hanya berjumlah 2206 data dan hanya terdapat 1 penyakit daun apel, sedangkan penyakit daun apel memiliki banyak variasinya, pada penelitian berikutnya sebaiknya ditambahkan lagi jenis penyakitnya agar jumlah kelas yang diklasifikasi menjadi lebih banyak atau *categorical*, dan mengembangkan aplikasi berbasis web yang serupa dan memperbaiki tampilan aplikasi saat ini, membuat tampilan lebih menarik.

#### Daftar Pustaka

- [1] C. Nisa', E. Y. Puspaningrum, and H. Maulana, "Penerapan Metode *Convolutional Neural Network* untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data," *Pros. Semin. Nas. Inform. Bela Negara*, vol. 1, pp. 169–175, 2020, doi: 10.33005/santika.v1i0.46.
- [2] R. Yohannes and M. E. Al Rivian, "Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM," *J. Algoritma*, vol. 2, no. 2, pp. 133–144, 2022, doi: 10.35957/algoritme.v2i2.2363.
- [3] H. Fonda, "Klasifikasi Batik Riau Dengan Menggunakan *Convolutional Neural Networks* (Cnn)," *J. Ilmu Komput.*, vol. 9, no. 1, pp. 7–10, 2020, doi: 10.33060/jik/2020/vol9.iss1.144.
- [4] Wahyutama Fitri Hidayat and Taufik Asra, "Klasifikasi Penyakit Daun Kentang Menggunakan Model Logistic Regression," *Indones. J. Softw. Eng.*, vol. 8, no. 2, pp. 173–179, 2022.
- [5] V. Ayumi, "Pengenalan Objek Bunga Berbasis Deep Learning Menggunakan Model *ResNet50* dan *MobileNet-v2*," *JSAI (Journal Sci. Appl. Informatics)*, vol. 5, no. 3, pp. 188–193, 2022, doi: 10.36085/jsai.v5i3.3613.
- [6] M. Fikry, "Pengembangan Aplikasi Klasifikasi Alat Transportasi Berdasarkan Citra Digital untuk Pencatatan Aset Studi Kasus : PT . Pulo Mas Jaya," vol. 9, no. 1, pp. 184–197, 2023.
- [7] N. D. Miranda, L. Novamizanti, and S. Rizal, "*Convolutional Neural Network* Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020, doi: 10.20884/1.jutif.2020.1.2.18.
- [8] F. H. Hawari, F. Fadillah, M. R. Alviandi, and T. Arifin, "Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritma Cnn (*Convolutional Neural Network*)," *J. Responsif Ris. Sains dan Inform.*, vol. 4, no. 2, pp. 184–189, 2022, doi: 10.51977/jti.v4i2.856.
- [9] E. Oktafanda, "Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode *Convolutional Neural Network* (CNN)," *J. Inform. Ekon. Bisnis*, vol. 4, no. 3, pp. 72–77, 2022, doi: 10.37034/infob.v4i3.143.
- [10] M. Khoiruddin, A. Junaidi, and W. A. Saputra, "Klasifikasi Penyakit Daun Padi Menggunakan *Convolutional Neural Network*," *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 2, no. 1, pp. 37–45, 2022, doi: 10.20895/dinda.v2i1.341.
- [11] E. Yuliani, A. N. Aini, and C. U. Khasanah, "Perbandingan Jumlah Epoch Dan Steps Per Epoch Pada *Convolutional Neural Network* Untuk Meningkatkan Akurasi Dalam Klasifikasi Gambar," *J. Inf. J. Penelit. dan Pengabd. Masy.*, vol. 5, no. 3, pp. 23–27, 2020, doi: 10.46808/informa.v5i3.140.