

# Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital

**Filach Akbar Arafat<sup>1</sup>, Muhammad Nur Ichsan<sup>2</sup>, Muhammad Firmandani Pramoedya<sup>3</sup>**

<sup>1,2</sup>Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: <sup>1</sup>[soackproo@gmail.com](mailto:soackproo@gmail.com), <sup>2</sup>[xsan69@gmail.com](mailto:xsan69@gmail.com), <sup>3</sup>[daniprmoedya2002@gmail.com](mailto:daniprmoedya2002@gmail.com)

**Abstrak** – Singkong merupakan salah satu tanaman yang banyak ditemukan di Indonesia dan merupakan tanaman dengan banyak manfaat. Salah satu manfaat singkong adalah sebagai bahan pokok pengganti nasi. Tanaman singkong atau biasa disebut ubi, memiliki banyak manfaat. Namun tanaman singkong juga tidak luput dari serangan penyakit. Salah satu faktor yang mempengaruhi rendahnya kualitas singkong adalah Adanya hama dan penyakit tanaman singkong yang sering kali mengakibatkan pertumbuhan umbi terganggu, bahkan dapat menggagalkan terwujudnya produksi yang maksimal. Salah satu cara yang dapat digunakan adalah memanfaatkan teknologi *artificial intelligence* (AI) yang sedang berkembang untuk dapat mengklasifikasikan penyakit pada singkong dengan mudah. Dirancanglah sebuah sistem yang dapat membantu dalam mengidentifikasi penyakit pada daun singkong menggunakan algoritma CNN dengan arsitektur *MobileNet*. Hasil akhir dari penelitian ini dengan melakukan pelatihan pada 30 *epoch* dengan rasio data latih dan data uji 80%:20% memberikan hasil akurasi *training* sebesar 88% dan *testing* sebesar 84%.

**Kata Kunci** — Penyakit Singkong, CNN, Klasifikasi Gambar, MobileNet

## 1. PENDAHULUAN

*Artificial Intelligence* (AI), atau dalam bahasa Indonesia dikenal sebagai Kecerdasan Buatan, adalah cabang ilmu komputer yang bertujuan untuk mengembangkan sistem dan mesin yang mampu melakukan tugas yang biasanya memerlukan kecerdasan manusia [1]. Singkong merupakan salah satu tanaman yang banyak ditemukan di Indonesia dan merupakan tanaman dengan banyak manfaat. Salah satu manfaat singkong adalah sebagai bahan pokok pengganti nasi [2]. Tanaman singkong atau biasa disebut ubi, memiliki banyak manfaat. Umbi yang diperoleh dari tanaman singkong memiliki kandungan karbohidrat yang tinggi dan dapat dimasak serta digunakan sebagai bahan. Selain umbi-umbian, daun ketela pohon juga bisa dijadikan sayuran [3]. Selain sebagai bahan makanan, ubi kayu juga dapat digunakan sebagai bahan baku industri dan pakan ternak. Oleh karena itu pengembangan ubi kayu sangat penting artinya di dalam upaya penyediaan bahan pangan karbohidrat non beras, diversifikasi/penganekaragaman konsumsi pangan lokal, pengembangan industri pengolahan hasil dan agro industri dan sebagai sumber devisa melalui ekspor serta upaya mendukung peningkatan ketahanan pangan dan kemandirian pangan [4]. Salah satu faktor yang mempengaruhi rendahnya kualitas singkong adalah Adanya hama dan penyakit tanaman singkong yang sering kali mengakibatkan pertumbuhan umbi terganggu, bahkan dapat menggagalkan terwujudnya produksi yang maksimal [5]. Dalam mengidentifikasi suatu penyakit dan hama pada tanaman singkong, dibutuhkan seorang pakar yang mengerti tentang penyakit pada tanaman singkong. Namun faktanya, tidak semua pakar dapat membantu para petani saat dibutuhkan, sehingga petani tidak dapat mengetahui secara cepat dan tepat permasalahan yang terjadi pada tanaman singkong [2]. Proses identifikasi yang ada selama ini masih dilakukan oleh para ahli pertanian, tetapi untuk mengatasi semua persoalan yang dihadapi petani terkendala oleh waktu dan banyaknya petani yang mempunyai masalah dengan tanamannya [5]. Salah satu cara yang dapat digunakan adalah memanfaatkan teknologi *artificial intelligence* (AI) yang sedang berkembang untuk dapat mengklasifikasikan penyakit pada singkong dengan mudah [6]. Sebuah sistem yang dapat digunakan untuk mengklasifikasikan penyakit tanaman singkong layaknya seorang pakar.

Penelitian yang pernah dilakukan oleh Iswantoro (2022) mengenai klasifikasi penyakit tanaman jagung menggunakan metode CNN, yang menunjukkan bahwa algoritma CNN sudah cukup baik dalam melakukan klasifikasi penyakit tanaman jagung ini. Dengan melakukan proses *training* dengan jumlah 100 *epoch*, jumlah *input* citra 150x150 dan didapatkan akurasi *training* sebesar 97,5% dan *validation* 100% [7]. Selain itu, penelitian yang dilakukan oleh Yuliany dan Rachman (2022) tentang penerapan *deep learning* pada sistem klasifikasi hama tanaman padi menunjukkan bahwa model dengan pembagian data 90% untuk pelatihan dan 10% untuk pengujian berhasil mencapai akurasi tertinggi sebesar 77,33%, dengan akurasi pelatihan mencapai 78,30%. Hasil penelitian menunjukkan bahwa penerapan metode CNN sangat sesuai dan mampu menghasilkan tingkat akurasi yang lebih tinggi [8]. Namun, penelitian yang berfokus pada penerapan metode CNN untuk mendeteksi penyakit daun singkong masih terbatas. Oleh karena itu, penulis merancang sebuah sistem yang digunakan untuk membantu

dalam mendeteksi penyakit pada daun singkong secara lebih cepat dan akurat. Dengan menerapkan teknik *deep learning* nantinya dapat menjadi alternatif untuk mempermudah para petani dalam mengatasi penyakit daun pada tanaman singkong.

## 2. METODE PENELITIAN

Pada penelitian ini merancang sebuah sistem yang dapat mendeteksi penyakit pada daun singkong dengan menggunakan algoritma *Convolutional Neural Network* (CNN) yang didasarkan pada arsitektur *MobileNet*. Pendekatan ini dipilih karena kemampuannya dalam mengenali pola visual pada gambar, yang sangat efektif untuk tugas klasifikasi gambar, seperti dalam mendeteksi penyakit pada tanaman. *Dataset* yang digunakan dalam penelitian ini diperoleh dari situs penyedia *dataset* *Kaggle*, yang terdiri dari gambar daun singkong yang menunjukkan berbagai kondisi, baik daun yang sehat maupun yang terinfeksi penyakit, seperti bercak daun, daun keriput, atau infeksi hama dan kutu. Data gambar tersebut merupakan koleksi gambar yang relevan dengan topik penelitian ini. Untuk menjadikan sistem dengan kualitas deteksi yang bagus, dilakukan *preprocessing* yang mencakup perubahan ukuran gambar agar seragam, augmentasi gambar untuk menambah variasi data, serta normalisasi. Dengan langkah-langkah *preprocessing* ini, *dataset* yang telah diproses dapat digunakan untuk melatih model CNN yang lebih efisien dan efektif. Untuk penjelasan lebih lanjut adalah sebagai berikut :

### 2.1 Persiapan Data

Data yang digunakan adalah data citra daun singkong. Data tersebut dibagi menjadi 5 kategori yaitu daun sehat, daun dengan bercak coklat, daun dengan bercak kuning, penyakit daun keriput, dan penyakit kutu daun. Data yang digunakan berjumlah 100 citra daun singkong. Tujuan dari studi literatur yang dilakukan adalah untuk mempelajari terkait teori dasar serta teori-teori pendukung yang berkaitan dengan klasifikasi, penyakit daun singkong, *Convolutional Neural Network* (CNN), dan *MobileNet*.

Tabel 1. *Dataset* Daun Singkong

Jenis Daun	Jumlah Data	Data Latih	Data Uji
Daun Sehat	100	80	20
Kutu Daun	100	80	20
Daun Keriput	100	80	20
Daun Bercak Kuning	100	80	20
Daun Bercak Coklat	100	80	20
Total	5000	400	100



Gambar 1. Daun Sehat



Gambar 2. Kutu Daun



Gambar 3. Daun Keriput



Gambar 4. Daun Bercak Kuning



Gambar 5. Daun Bercak Coklat

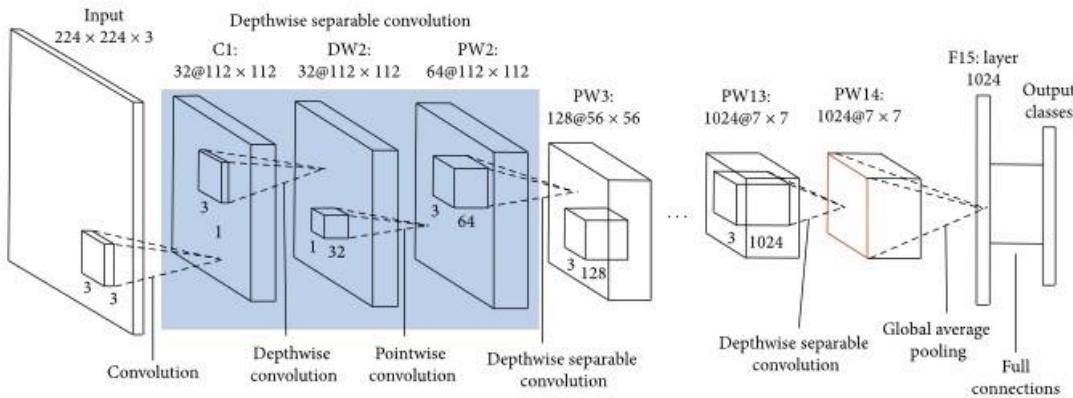
### 2.2 *Preprocessing* Data

Sebelum digunakan sebagai data dalam tahap pelatihan, *dataset* ini akan melalui proses *preprocessing* terlebih dahulu. Tahapan *preprocessing* yang digunakan adalah normalisasi dan *resize*. Tahap normalisasi digunakan untuk mengubah skala intensitas warna pada gambar yang semula 0 – 255 menjadi hanya 0 – 1. *Resize* digunakan untuk mengubah ukuran gambar menjadi 224×224 *pixels* [9].

### 2.3 Arsitektur Algoritma

Proses dilakukan dengan menggunakan arsitektur *MobileNet* pada CNN. CNN dipilih karena metode tersebut paling optimal dalam kasus klasifikasi citra, yang mana salah satu kelebihannya ialah ekstraksi fitur citra yang dilakukan secara otomatis, sehingga dapat menghemat waktu dan tenaga [10]. *Convolutional Neural Network* (CNN) adalah salah satu jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk menangani berbagai tugas, seperti klasifikasi gambar, deteksi objek, dan segmentasi. CNN merupakan

pengembangan dari *Multi-Layer Perceptron* (MLP) dan menjadi salah satu algoritma unggulan dalam bidang *deep learning*. Algoritma ini pertama kali diperkenalkan oleh Yann LeCun, Léon Bottou, Yoshua Bengio, dan Patrick Haffner pada tahun 1998 [11]. *MobileNet* merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan *computing* data dalam jumlah yang besar. Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN pada umumnya adalah penggunaan lapisan atau layer konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari *input image*. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution* [12].



Gambar 6. CNN *MobileNet Architecture*. (Sumber: Bacus & Linsangan, 2024) [13]

#### 2.4 Pelatihan Model

Pada proses ini, dilakukan pelatihan model menggunakan *dataset* citra yang telah dipersiapkan sebelumnya. Pelatihan ini menerapkan arsitektur *MobileNet*, yang dilaksanakan dengan lima kali percobaan pelatihan (*epoch*) dengan jumlah *epoch* yang bervariasi pada setiap percobaan. Setiap percobaan akan dianalisis untuk melihat dampak perbedaan jumlah *epoch* terhadap performa model yang dihasilkan.. *Epoch* merupakan satu kali proses pelatihan model dimana seluruh paket **pelatihan** yang digunakan untuk mengajar model akan diulang sebanyak periode yang telah ditentukan [14].

```
Epoch 1/10
16/16 45s 2s/step - accuracy: 0.3586 - loss: 1.6022 - val_accuracy: 0.5800 - val_loss: 1.0689
Epoch 2/10
16/16 40s 2s/step - accuracy: 0.5468 - loss: 1.1492 - val_accuracy: 0.6100 - val_loss: 0.9638
Epoch 3/10
16/16 36s 2s/step - accuracy: 0.5973 - loss: 1.0003 - val_accuracy: 0.6400 - val_loss: 0.8969
Epoch 4/10
16/16 37s 2s/step - accuracy: 0.6189 - loss: 0.9423 - val_accuracy: 0.6900 - val_loss: 0.8486
Epoch 5/10
16/16 40s 2s/step - accuracy: 0.6289 - loss: 0.9091 - val_accuracy: 0.6700 - val_loss: 0.8447
Epoch 6/10
16/16 41s 2s/step - accuracy: 0.6374 - loss: 0.8458 - val_accuracy: 0.7300 - val_loss: 0.7395
Epoch 7/10
16/16 41s 2s/step - accuracy: 0.6982 - loss: 0.8297 - val_accuracy: 0.7400 - val_loss: 0.6455
Epoch 8/10
16/16 36s 2s/step - accuracy: 0.7056 - loss: 0.8073 - val_accuracy: 0.7700 - val_loss: 0.6489
Epoch 9/10
16/16 37s 2s/step - accuracy: 0.6608 - loss: 0.8017 - val_accuracy: 0.7300 - val_loss: 0.6917
Epoch 10/10
16/16 37s 2s/step - accuracy: 0.7493 - loss: 0.7031 - val_accuracy: 0.7400 - val_loss: 0.5978
```

Gambar 7. Percobaan Pelatihan *Epoch 1*

```
Epoch 15/20
16/16 46s 2s/step - accuracy: 0.7596 - loss: 0.6890 - val_accuracy: 0.7900 - val_loss: 0.6226
Epoch 16/20
16/16 40s 2s/step - accuracy: 0.7552 - loss: 0.6816 - val_accuracy: 0.8300 - val_loss: 0.5824
Epoch 17/20
16/16 36s 2s/step - accuracy: 0.7858 - loss: 0.5452 - val_accuracy: 0.7800 - val_loss: 0.5801
Epoch 18/20
16/16 37s 2s/step - accuracy: 0.8221 - loss: 0.5288 - val_accuracy: 0.7600 - val_loss: 0.5721
Epoch 19/20
16/16 36s 2s/step - accuracy: 0.8815 - loss: 0.5329 - val_accuracy: 0.8300 - val_loss: 0.4637
Epoch 20/20
16/16 41s 2s/step - accuracy: 0.8299 - loss: 0.4727 - val_accuracy: 0.8100 - val_loss: 0.5197
```

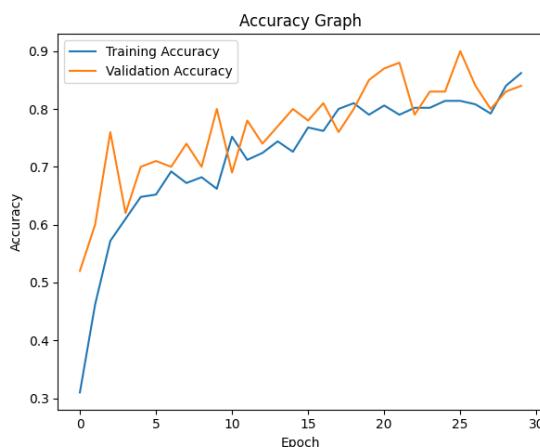
Gambar 8. Percobaan Pelatihan *Epoch 2*

Epoch	25/30
16/16	44s 2s/step - accuracy: 0.8066 - loss: 0.5446 - val_accuracy: 0.8300 - val_loss: 0.4204
Epoch 26/30	
16/16	36s 2s/step - accuracy: 0.8247 - loss: 0.5044 - val_accuracy: 0.9000 - val_loss: 0.3502
Epoch 27/30	
16/16	41s 2s/step - accuracy: 0.8269 - loss: 0.4636 - val_accuracy: 0.8400 - val_loss: 0.4375
Epoch 28/30	
16/16	46s 2s/step - accuracy: 0.7952 - loss: 0.5383 - val_accuracy: 0.8000 - val_loss: 0.5364
Epoch 29/30	
16/16	36s 2s/step - accuracy: 0.8570 - loss: 0.4054 - val_accuracy: 0.8300 - val_loss: 0.4535
Epoch 30/30	
16/16	38s 2s/step - accuracy: 0.8828 - loss: 0.3942 - val_accuracy: 0.8400 - val_loss: 0.4581

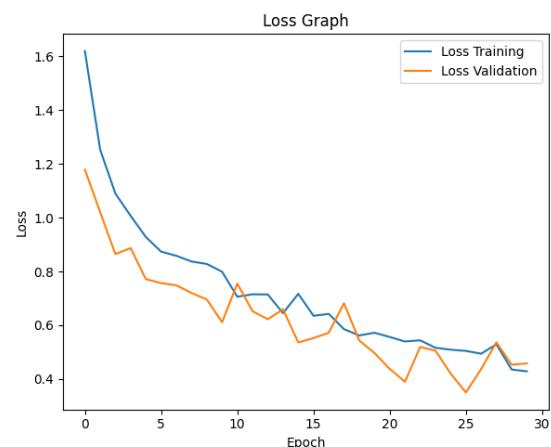
Gambar 9. Percobaan Pelatihan *Epoch* 3

### 3. HASIL DAN PEMBAHASAN

Pada penelitian ini, sistem deteksi penyakit daun pada tanaman singkong menggunakan algoritma (CNN) berhasil diimplementasikan pada sistem yang menggunakan klasifikasi citra. Setelah melalui proses *preprocessing* yang meliputi pengubahan ukuran menjadi 224 x 224, normalisasi, dan augmentasi data. Uji coba terhadap sistem ini dilakukan guna mencari performa maksimal dalam mengklasifikasikan citra penyakit daun singkong dengan menggunakan arsitektur *MobileNet*, dengan fokus pada *parameter accuracy* dan *loss*. Pada penelitian ini menggunakan *dataset* yang terdiri dari 100 gambar daun singkong yang terbagi dalam 5 kategori yaitu daun sehat, daun dengan bercak coklat, daun dengan bercak kuning, penyakit daun keriput, dan penyakit kutu daun. Pada eksperimen yang telah dilakukan ini, arsitektur *MobileNet* diuji untuk melakukan klasifikasi penyakit daun pada tanaman singkong. Berikut merupakan hasil grafik dengan jumlah pelatihan 30 epoch.



Gambar 10. Accuracy Graph



Gambar 11. Loss Graph

Pada grafik *accuracy* dan *loss* terdapat beberapa hal yang dapat diperhatikan. Hal tersebut antara lain sebagai berikut :

Grafik *accuracy* :

- Tren Umum : Baik akurasi pelatihan maupun validasi telah meningkat, seperti yang ditunjukkan pada grafik akurasi. Akurasi pelatihan mencapai puncaknya sekitar *epoch* 28, dan akurasi validasi mencapai puncaknya sekitar *epoch* 26.
- *Gap accuracy* : Tidak ada *overfitting* yang signifikan pada model ini, seperti yang ditunjukkan oleh perbedaan kecil antara akurasi pelatihan dan validasi, yang menunjukkan bahwa model dapat digeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.
- Perbedaan *accuracy* : Akurasi *training* dan *validation* cukup konsisten sepanjang proses, meskipun ada sedikit perbedaan. Ini menunjukkan bahwa model belajar baik dan mampu mempertahankan kinerja data *training* dan validasi.

Grafik *loss* :

- Tren Umum : Baik *loss training* maupun *loss validation* menunjukkan tren penurunan pada grafik *loss*. *Loss* mencapai *training* titik terendah sekitar *epoch* 28, dan *loss validation* mencapai titik terendah sekitar *epoch* 26.

- *Gap loss* : Terdapat gap kecil antara *loss training* dan validasi, yang menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan. Gap ini menunjukkan bahwa model dapat generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.
- Perbedaan *Loss*: *Loss training* dan *validation* cukup konsisten sepanjang proses *training*, meskipun terdapat fluktuasi kecil. Hal ini menunjukkan bahwa model belajar dengan baik dan mampu mempertahankan performa pada data *training* dan *validation*.

Tabel 2. Perbandingan *Epoch*

Epoch	Data Latih		Data Uji	
	Acc	Loss	Vall Acc	Vall Loss
10	0.7493	0.7031	0.7400	0.5978
20	0.8299	0.4727	0.8100	0.5197
30	0.8828	0.3942	0.8400	0.4581

Pada tabel 2 ditampilkan perbandingan 3 pelatihan *epoch*. Melihat dari hasil tersebut pada *epoch* ke-10, model menunjukkan akurasi sebesar 74.93% pada data latih dan 74.00% pada data uji, dengan *loss* pada data latih sebesar 0.7031 dan pada data uji 0.5978. Meskipun hasilnya cukup baik, akurasi dan *loss* pada data latih dan uji masih menunjukkan adanya ruang untuk perbaikan. Ketika jumlah *epoch* ditingkatkan menjadi 20, performa model mengalami peningkatan yang jelas. Akurasi pada data latih meningkat menjadi 82.99% dengan *loss* yang berkurang menjadi 0.4727, sementara pada data uji, akurasi mencapai 81.00% dan *loss* sedikit meningkat menjadi 0.5197. Ini menunjukkan bahwa model mulai lebih mampu mempelajari pola pada data latih, sementara di sisi lain, model juga berhasil mempertahankan akurasi yang cukup tinggi pada data uji meskipun ada sedikit peningkatan pada *loss*-nya. Pada *epoch* ke-30, hasilnya semakin menunjukkan kemajuan yang signifikan pada data latih, dengan akurasi mencapai 88.28% dan *loss* turun lebih jauh menjadi 0.3942. Namun, meskipun akurasi pada data uji juga meningkat menjadi 84.00%, *loss* pada data uji mengalami sedikit peningkatan menjadi 0.4581. Peningkatan ini mungkin mencerminkan bahwa model mulai mendekati batas kemampuan optimalnya dalam mempelajari data, yang tercermin dari melambatnya kenaikan akurasi pada data uji dan sedikitnya peningkatan *loss* setelah *epoch* ke-20. Ini bisa menunjukkan bahwa model mulai mengalami sedikit *overfitting* terhadap data latih, di mana model terlalu menyesuaikan diri dengan data latih sehingga tidak lagi memberikan peningkatan yang signifikan pada data uji.

#### 4. SIMPULAN

Berdasarkan temuan pada penelitian ini, dapat disimpulkan bahwa penggunaan model dengan arsitektur *MobileNet* yang dilatih dengan 3 percobaan *epoch* menunjukkan peningkatan kinerja yang konsisten seiring dengan jumlah *epoch* yang digunakan. Selama proses pelatihan, model secara bertahap meningkatkan akurasi yang signifikan dalam mengenali dan mengklasifikasikan berbagai jenis penyakit pada daun singkong. Hal ini menunjukkan bahwa kinerja model dipengaruhi secara langsung oleh jumlah *epoch* yang digunakan dalam pelatihan. Semakin banyak *epoch* yang diterapkan, lebih banyak kesempatan bagi model untuk melakukan iterasi dan mempelajari pola-pola penting dalam data, yang memungkinkan model untuk menyesuaikan parameter internalnya dengan lebih baik. Dengan kata lain, pelatihan yang lebih lama memberi model lebih banyak waktu untuk mengoptimalkan proses pembelajarannya, sehingga lebih akurat mendeteksi penyakit pada daun.

#### 5. SARAN

Berdasarkan dari penelitian yang telah dilakukan, saran yang dapat diberikan adalah untuk meningkatkan akurasi sistem deteksi dengan memperluas dan memperkaya *dataset*, agar mencakup lebih banyak variasi gambar daun singkong dari berbagai kondisi lapangan. Hal ini akan membantu model mengenali gejala penyakit dengan lebih baik, terutama untuk penyakit yang memiliki gejala serupa. Selain itu, disarankan untuk mengembangkan model dengan menggunakan arsitektur yang lebih kompleks untuk dapat meningkatkan kemampuan klasifikasi penyakit yang lebih sulit dibedakan. Penyempurnaan sistem juga diperlukan, dengan melakukan integrasi sistem deteksi penyakit daun pada sistem yang berbasis *mobile application* diharapkan dapat mempermudah pengguna dalam menggunakan sistem, serta memastikan sistem dapat berfungsi optimal dengan kondisi pencahayaan yang berbeda. Dengan langkah-langkah tersebut, diharapkan teknologi deteksi penyakit daun singkong dapat semakin berkembang dan memberikan dampak positif terhadap peningkatan produktivitas pertanian singkong.

## DAFTAR PUSTAKA

- [1] E. S. Eriana and M. D. A. Z. M. S. Kom., *ARTIFICIAL INTELLIGENCE (AI)*, CV. EUREKA MEDIA AKSARA, 2023.
- [2] A. J. M. S. K. Selviana Yunita, "Sistem Pakar Deteksi Penyakit Pada Tanaman Singkong Expert System to Detect the Disease of Cassava Plants," *Jurnal Ilmiah SISFOTENIKA*, vol. 9, no. 1, p. 25, 2019.
- [3] D. K. H. R. Hendra Rustantono, "Pelatihan Pemanfaatan Limbah Kulit Singkong Menjadi Keripik," *I-Com : Indonesian Community Journal*, vol. 2., no. 1, pp. 31-37, 2022.
- [4] M. Ir. Roch Widaningsih, *OUTLOOK KOMODITAS PERTANIAN TANAMAN PANGAN UBI KAYU*, Pusat Data dan Sistem Informasi Pertanian Kementerian Pertanian, 2016.
- [5] A. Isnadi, "Aplikasi Diagnosa Hama Dan Penyakit Tanaman Singkong Menggunakan Metode Certainty Factor Berbasis Web," *J-INTECH (Journal of Information and Technology)*, vol. 6, no. 2, pp. 177-180, 2018.
- [6] S. R. N. K. C. P. A. E. Nugraha, ""Klasifikasi Penyakit Pada Tanaman Singkong Menggunakan Arsitektur VGGNET Berbasis Deep Learning,"" *eProceedings of Engineering*, vol. 9, no. 6, 2022.
- [7] D. H. U. D. Iswantoro, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 22, no. 2, pp. 900-905, 2022.
- [8] A. N. R. S. Yuliany, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Buana Informatika*, vol. 13, no. 1, pp. 54-65, 2022.
- [9] S. R. d. N. K. C. P. A. E. Nugraha, "Klasifikasi Penyakit Pada Tanaman Singkong Menggunakan Arsitektur VGGNET Berbasis Deep Learning," *eProceedings of Engineering*, vol. 9, no. 6, 2022.
- [10] Y. P. I. Y. N. A. M. F. Firdaus, "Klasifikasi Tanaman Anggrek Menggunakan Metode CNN Berbasis Web Django," *Stains*, vol. 3, no. 1, pp. 394-403, 2024.
- [11] I. F. d. R. A. P. A. Nugroho, "Implementasi deep learning menggunakan convolutional neural network (CNN) pada ekspresi manusia," *Algor*, vol. 2, no. 1, pp. 12-20, 2020.
- [12] H. H. d. Y. Maulana, "Identification of Types of Wood using Convolutional Neural Network with Mobilenet Architecture," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 70-76, February 2020.
- [13] N. B. L. J. A. Bacus, "Detection and Identification with Analysis of Carica papaya Leaf Using Android - Scientific Figure".
- [14] A. S. H. A. F. A. T. Rahman, "Klasifikasi Penyakit Daun Apel Menggunakan Arsitektur CNN dengan Transfer Learning," *Jurnal SENOPATI: Sustainability, Ergonomics, Optimization, and Application of Industrial Engineering*, vol. 6, no. 1, 2024.