

# Penerapan Model *Convolutional Neural Network* (Cnn) Untuk Klasifikasi Penyakit Daun Pada Tanaman Singkong (*Manihot Esculenta Crantz*)

Gilang Surya<sup>1\*</sup>, Sufiatul Maryana<sup>2</sup>, Siska Andriani<sup>3</sup>

<sup>1,2,3</sup>Program Studi Ilmu Komputer, Universitas Pakuan, Indonesia

<sup>1\*</sup>suryaagilang02@email.com, <sup>2</sup>sufiatul.maryana@unpak.ac.id,

<sup>3</sup>siska.andriani@unpak.ac.id

Submitted	Accepted	Publish
14-February-2026	29-April-2026	15-June-2026

**Abstrak:** Penurunan produktivitas tanaman singkong di Indonesia sering disebabkan oleh penyakit daun seperti *Cassava Bacterial Blight*, *Brown Streak Disease*, *Green Mottle*, dan *Mosaic Disease* yang sulit diidentifikasi secara manual sehingga berpotensi menimbulkan kesalahan diagnosis. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis berbasis citra digital menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet152V2. Dataset yang digunakan berjumlah 21.397 citra daun singkong dari Kaggle yang terdiri dari lima kelas, dengan penanganan ketidakseimbangan data menggunakan teknik *Random Oversampling*. Tahapan penelitian meliputi *Exploratory Data Analysis* (EDA), pra-pemrosesan data (resize 32×32 piksel, normalisasi, dan augmentasi), pembagian data (80:20), pemodelan CNN, serta evaluasi menggunakan confusion matrix dengan metrik akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu mencapai akurasi sebesar 97,2%, yang menunjukkan kinerja sangat baik dalam mengklasifikasikan penyakit daun singkong. Dengan demikian, sistem ini dapat menjadi solusi efektif dalam mendukung deteksi dini penyakit tanaman secara cepat dan akurat serta berkontribusi pada pengembangan pertanian cerdas (*smart agriculture*).

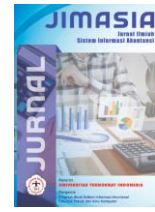
**Kata Kunci:** Penyakit daun singkong, CNN, ResNet152V2, Klasifikasi citra, *Deep Learning*, *Streamlit*.

**Abstract:** The decline in cassava productivity in Indonesia is often caused by leaf diseases such as *Cassava Bacterial Blight*, *Brown Streak Disease*, *Green Mottle*, and *Mosaic Disease*, which are difficult to identify manually and prone to misdiagnosis. Therefore, this study aims to develop an automatic classification system based on digital images using the *Convolutional Neural Network* (CNN) method with the ResNet152V2 architecture. The dataset consists of 21,397 cassava leaf images obtained from Kaggle, categorized into five classes, with class imbalance handled using the *Random Oversampling* technique. The research stages include *Exploratory Data Analysis* (EDA), data preprocessing (image resizing to 32×32 pixels, normalization, and augmentation), data splitting (80:20), CNN modeling, and evaluation using a confusion matrix with accuracy, precision, recall, and F1-score metrics. The results show that the proposed

Sufiatul Maryana: \* Penulis Korespondensi



Copyright © 2026, Gilang Surya, Sufiatul Maryana, Siska Andriani.



model achieves an accuracy of 97.2%, indicating excellent performance in classifying cassava leaf diseases. Thus, this system can serve as an effective solution for early detection of plant diseases quickly and accurately and contribute to the development of smart agriculture systems.

**Keywords:** Cassava leaf disease, CNN, ResNet152V2, Image classification, Deep Learning, Streamlit.

## 1. PENDAHULUAN

Singkong (*Manihot esculenta Crantz*) merupakan salah satu komoditas pangan penting di Indonesia yang memiliki nilai ekonomi tinggi, dengan produksi nasional mencapai lebih dari 18 juta ton per tahun [1]. Meskipun demikian, produktivitas tanaman singkong sering mengalami penurunan akibat serangan penyakit daun yang menghambat proses fotosintesis. Akibat penurunan tersebut, banyak para petani sulit untuk mengidentifikasi penyakit tanaman singkong secara manual sehingga menimbulkan kesalahan diagnosis untuk penanganan penyakit menyebar ke seluruh lahan. Empat penyakit utama yang umum menyerang daun singkong adalah *Cassava Bacterial Blight*, *Cassava Brown Streak Disease*, *Cassava Green Mottle*, dan *Cassava Mosaic Disease* [2]. Dataset citra daun singkong yang tersedia di Kaggle berjumlah 13.158 citra menunjukkan adanya ketidakseimbangan kelas, dengan *Cassava Mosaic Disease* sebagai kelas mayoritas. Oleh karena itu, dilakukan penyeimbangan data menggunakan metode *Random Oversampling* (ROS) pada kelas minoritas [3]. Mengingat proses identifikasi penyakit masih dilakukan secara manual dan rentan terhadap kesalahan [4], maka diperlukan sistem otomatis yang mampu mendeteksi penyakit daun singkong secara cepat dan akurat.

Setelah proses pengolahan citra, analisis data dapat dilakukan menggunakan pendekatan *Machine Learning*, yaitu metode pembelajaran otomatis yang memungkinkan komputer mengenali pola dari data tanpa pemrograman eksplisit. Perkembangan lebih lanjut dari *Machine Learning* adalah *Deep Learning*, yang memanfaatkan jaringan saraf tiruan berlapis banyak (*multi-layered artificial neural network*) untuk menghasilkan representasi fitur yang lebih kompleks dan abstrak [5]. Salah satu model *Deep Learning* yang paling banyak digunakan dalam pengolahan citra adalah *Convolutional Neural Network* (CNN). CNN dirancang untuk memproses data berbentuk gambar melalui tiga lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer* [6]. Lapisan konvolusi berfungsi mengekstraksi fitur penting, *pooling layer* menyederhanakan data tanpa kehilangan informasi signifikan, sedangkan *fully connected layer* berperan dalam proses klasifikasi. Dengan kemampuan tersebut, CNN mampu mengenali pola visual seperti warna, tepi, tekstur, dan bentuk daun singkong yang menunjukkan gejala penyakit tertentu.

Dalam perkembangannya, berbagai arsitektur CNN telah dikembangkan untuk meningkatkan akurasi dan efisiensi, salah satunya adalah ResNet (*Residual Network*). ResNet, khususnya varian ResNet152V2, menerapkan konsep *residual learning* melalui mekanisme *skip connection* atau *shortcut path* yang memungkinkan aliran informasi lintas lapisan [7]. Mekanisme ini efektif dalam mengatasi permasalahan *vanishing gradient* pada jaringan yang sangat dalam, sehingga proses pembelajaran menjadi lebih stabil dan mampu menghasilkan ekstraksi fitur yang lebih mendalam. Karakteristik ini menjadikan ResNet152V2 sangat sesuai untuk tugas klasifikasi citra yang kompleks, seperti identifikasi penyakit daun singkong.

Beberapa penelitian terdahulu telah menerapkan algoritma *Deep Learning* dalam klasifikasi citra penyakit daun tanaman, di antaranya [8] dengan judul "Penerapan metode residual network (resnet) dalam klasifikasi penyakit pada daun gandum", [9] dengan judul "Implementasi Arsitektural Resnet-34 Dalam Klasifikasi Gambar Penyakit Pada Daun Kentang". [4] dengan judul "Deteksi Penyakit Pada Daun Singkong Dengan Pengolahan Citra Menggunakan Metode *Convolutional Neural Network* (CNN)". [10] dengan judul

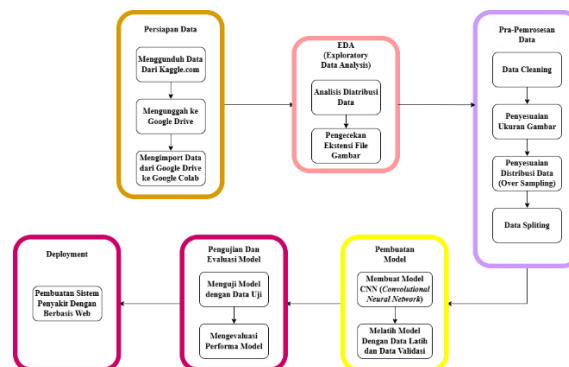


"Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital". [11] dengan judul "Klasifikasi Penyakit Daun Singkong Menggunakan *Convolutional Neural Network* (CNN) Dengan Arsitektur VGG16 Berbasis Android". Berdasarkan penelitian-penelitian tersebut, terlihat bahwa pemanfaatan arsitektur CNN yang berbeda dapat memberikan variasi performa dalam klasifikasi citra penyakit tanaman.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan model CNN berbasis arsitektur ResNet152V2 yang mampu melakukan klasifikasi penyakit daun singkong secara otomatis, cepat, dan akurat berdasarkan citra digital. Melalui pendekatan ini, diharapkan penelitian dapat berkontribusi dalam pengembangan sistem *smart agriculture* berbasis kecerdasan buatan yang membantu petani dalam mendeteksi penyakit secara dini, meningkatkan efisiensi produksi, serta mengoptimalkan kualitas hasil panen di Indonesia.

## 2. METODE PENELITIAN

Penelitian ini secara garis besar terdiri dari enam tahapan utama yang dilalui secara sistematis. Setiap tahapan dirancang untuk mendukung proses pengolahan dan analisis citra daun singkong secara optimal, mulai dari pengumpulan data hingga evaluasi model. Secara umum, alur penelitian mencakup proses persiapan data, *exploratory data analysis* (EDA), pra-pemrosesan data, pembagian data, pembuatan model menggunakan metode CNN (*Convolution Neural Network*), evaluasi model, serta *deployment* menggunakan metrik klasifikasi. Diagram alur penelitian secara keseluruhan dapat dilihat pada Gambar 1.



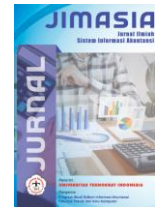
Gambar 1. Diagram Alur Penelitian

### 2.1 PERSIAPAN DATA

Dataset yang digunakan pada penelitian ini adalah dataset sekunder yang diambil dari kaggle.com. Dataset ini memiliki total 21.397 data gambar yang terdiri atas 5 katagori yaitu *healthy*, *bacterial blight*, *brown streak*, *green mottle*, dan *mosaic disease*. Pada tahap ini, dataset diunduh dan unggah kedalam *Google Drive* sebagai basis data untuk membuat model. Untuk menggunakan dataset tersebut, dataset yang berada dalam *Google Drive* dimuat kedalam lingkungan proyek di *Google Colab*.

### 2.2 Exploratory Data Analysis (EDA)

*Exploratory Data Analysis* (EDA) dilakukan untuk memahami struktur dataset, mengidentifikasi ketidakseimbangan kelas, serta memastikan konsistensi format file gambar. Hasil analisis menunjukkan bahwa terdapat ketimpangan signifikan antar kelas, di mana kelas *Mosaic Disease* mendominasi dengan jumlah gambar jauh lebih banyak dibandingkan kelas lain. Seluruh gambar menggunakan format .jpg, sehingga tidak diperlukan konversi format. Temuan ini menjadi dasar untuk menerapkan strategi



*augmentasi* dan penyeimbangan data dalam tahap pra-pemrosesan agar performa model lebih seimbang [12].

### 2.3 PRA-PEMROSESAN DATA

Tahap pra-pemrosesan data dilakukan untuk menyiapkan dataset agar sesuai dengan kebutuhan model sebelum masuk ke proses *modelling*. Seluruh langkah pra-pemrosesan ini bertujuan untuk meningkatkan kualitas data input dan memastikan bahwa model memiliki kesempatan yang adil untuk mempelajari karakteristik dari semua kelas secara seimbang. Dalam penelitian ini menggunakan beberapa tahapan *preprocessing* yaitu :

#### 2.3.1 Data Cleaning

Teknik *data cleaning* adalah suatu teknik yang digunakan untuk menangani data yang tidak lengkap. Proses data cleaning ini juga mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak [13]. Untuk tahapan ini data tidak terdapat kesalahan/kosong seluruh file gambar memiliki ekstensi .jpg, sehingga tidak ditemukan ketidaksesuaian format yang perlu diperbaiki. Oleh karena itu, penyeragaman ekstensi file tidak diperlukan lebih lanjut.

#### 2.3.2 Resize

Pada penelitian ini, ukuran input citra ditetapkan menjadi  $32 \times 32$  piksel untuk mempercepat komputasi dan mengurangi beban pemrosesan. Meskipun ResNet152V2 umumnya menggunakan input  $224 \times 224$  piksel, arsitektur *residual learning* memungkinkannya tetap bekerja efektif pada resolusi lebih kecil. Pemilihan ukuran  $32 \times 32$  juga sesuai dengan jumlah dataset yang besar karena dapat mempercepat proses pelatihan, menurunkan kebutuhan memori, serta membantu mengurangi risiko *overfitting*.

#### 2.3.3 Random Oversampling

*Random oversampling* bekerja dengan menambah jumlah data pada kelas minoritas melalui proses duplikasi sampel secara acak. Dengan cara ini, jumlah data pada kelas minoritas akan diseimbangkan hingga setara dengan kelas mayoritas. Teknik ini membantu mencegah model menjadi bias terhadap kelas yang jumlah datanya lebih besar, sehingga proses pelatihan menjadi lebih stabil dan model dapat mengenali pola dari semua kelas dengan lebih baik [14]. Dalam dataset membuat data menjadi seimbang dengan memperbanyak dataset yang kurang dari kelas terbanyak sehingga dataset menjadi data yang seimbang untuk diolah.

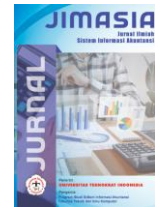
#### 2.3.4 Data Splitting

Split data merupakan proses pembagian data menjadi data uji coba dan data latih secara acak. Proses ini dilakukan untuk mengevaluasi kinerja dari model yang dibangun menggunakan data yang dilatih, data uji dengan *split validation* digunakan sebagai validasi. Data yang tidak pernah terlibat selama proses pembangunan model tetapi digunakan untuk menilai apakah penelitian berhasil atau tidak disebut sebagai data pengujian. Sementara itu, data pelatihan adalah data yang dipakai untuk melatih atau membangun model dalam penelitian tersebut [15].

### 2.4 Modeling

Penelitian ini menggunakan pendekatan deep learning berbasis *Convolutional Neural Network* (CNN) dengan teknik *transfer learning* menggunakan arsitektur ResNet152V2. Arsitektur ini dipilih karena kemampuannya dalam mengenali fitur kompleks dan mengolah citra resolusi tinggi melalui mekanisme *skip connection*. Model dikembangkan dengan menambahkan lapisan seperti *batch normalization*, *dropout*, dan *dense layer* agar sesuai dengan karakteristik citra daun singkong. Input berupa gambar RGB berukuran  $32 \times 32$





*piksel* diproses melalui jaringan ResNet untuk ekstraksi fitur, dilanjutkan dengan lapisan-lapisan tambahan hingga klasifikasi akhir ke dalam lima kategori penyakit.

## 2.5 Evaluation

Evaluasi ini digunakan untuk mengukur ketepatan atau jumlah *error* yang ada pada model yang sudah di bangun. Dalam penelitian ini *evaluation* menggunakan *Confusion Matrix* yang digunakan untuk mengukur performa model dalam mengklasifikasikan citra daun singkong. Penelitian ini menggunakan empat metrik utama, yaitu akurasi, presisi, *recall*, dan *f1-score*. Metrik-metrik ini memberikan gambaran menyeluruh, baik dari segi ketepatan prediksi maupun kemampuan model mengenali masing-masing kelas secara adil khususnya penting saat data tidak seimbang.

## 2.6 Deployment

Tahap *deployment* dilakukan dengan mengimplementasikan model ResNet152V2 yang telah dilatih ke dalam *Google Colab* dan aplikasi web berbasis *Streamlit*. Model disimpan dalam format. *keras* lalu diintegrasikan ke dalam antarmuka yang memungkinkan pengguna mengunggah gambar daun singkong. Gambar tersebut diproses melalui tahapan pra-proses, ekstraksi fitur oleh ResNet152V2, dan klasifikasi pada lapisan Dense akhir. Hasil prediksi ditampilkan secara real-time sehingga model dapat digunakan secara praktis dan mudah oleh pengguna di luar lingkungan pengembangan.

# 3. HASIL DAN PEMBAHASAN

## 3.1 Hasil

Tahapan perancangan dan implementasi yang telah dijelaskan sebelumnya menjadi dasar dalam pembuatan program. Pada bagian ini disajikan hasil dari program yang telah dikembangkan beserta pembahasannya.

### 3.1.1 Persiapan Data

Dalam penelitian ini digunakan dataset berupa data sekunder yang berasal dari sumber *open source* di Kaggle.com dan dipublikasikan oleh pengguna bernama Nirmal Sankalana pada tahun 2023. Secara keseluruhan, dataset memuat 21.397 citra daun singkong dengan lima kategori utama, yaitu *Cassava Bacterial Blight*, *Cassava Brown Streak Disease*, *Cassava Green Mottle*, *Cassava Mosaic Disease*, dan *Healthy Leaf*. Dataset diunduh ke penyimpanan lokal terlebih dahulu, kemudian diunggah ke *Google Drive* sebagai basis penyimpanan. Selanjutnya, dataset dimuat ke dalam *Google Colab* dengan membuat direktori khusus.

### 3.1.2 Exploratory Data Analysis

Dataset yang telah dimuat ke *Google Colab* kemudian dianalisis untuk mendapatkan informasi penting mengenai isi data. Tahap eksplorasi ini mencakup penampilan contoh gambar dari tiap kategori, perhitungan jumlah data pada masing-masing kelas, visualisasi distribusi dataset, serta pemeriksaan berbagai ekstensi file gambar yang digunakan. Langkah-langkah ini dilakukan untuk memastikan data siap diproses pada tahap pelatihan model selanjutnya.

```
Folder: Cassava_healthy
Format: jpg, Jumlah: 2577

Folder: Cassava_mosaic_disease
Format: jpg, Jumlah: 13158

Folder: Cassava_brown_streak_disease
Format: jpg, Jumlah: 2189

Folder: Cassava_bacterial_blight
Format: jpg, Jumlah: 1087

Folder: Cassava_green_mottle
Format: jpg, Jumlah: 2386
```

**Gambar 2.** Ekstensi file dan jumlah datanya

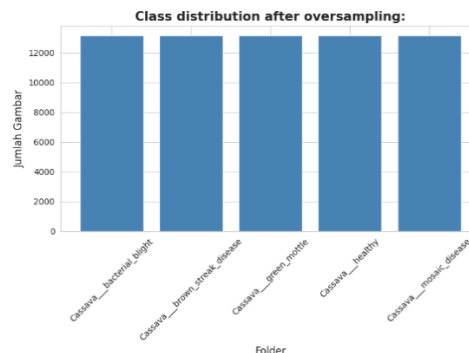
Berdasarkan gambar 2 terlihat bahwa jumlah data pada setiap kategori tidak seimbang. Meskipun terdapat perbedaan jumlah data pada masing-masing kelas, penyeragaman distribusi data tetap direncanakan. Ketidakeimbangan ini dapat diatasi menggunakan teknik random oversampling pada kelas yang jumlah datanya sedikit, sehingga setiap kelas memiliki jumlah data yang setara dan model tidak mengalami bias saat pelatihan. Hasil eksplorasi pada Gambar juga menunjukkan bahwa seluruh file gambar memiliki ekstensi yang sama, yaitu .jpg atau .jpeg. Format ini umum digunakan karena menghasilkan kualitas gambar yang baik dengan ukuran file yang relatif kecil. Karena semua file sudah menggunakan ekstensi yang seragam, proses normalisasi atau penyamaan ekstensi tidak perlu dilakukan.

### 3.1.3 Pra-Pemrosesan Data

Setelah memperoleh berbagai informasi dari tahap *Exploratory Data Analysis* (EDA), langkah selanjutnya adalah melakukan penyesuaian terhadap dataset agar siap digunakan dalam proses training dan testing. Tahapan penyesuaian ini meliputi perubahan ukuran citra, penetapan nilai *batch size*, serta pengacakan data sebelum digunakan pada proses pemodelan.

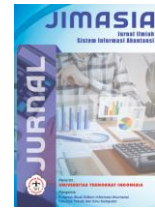
#### 3.1.3.1 Random Oversampling

Pada tahap ini meratakan dataset minoritas agar sama dengan dataset terbanyak atau mayoritas. *Random Oversampling* dipilih karena lebih stabil, aman terhadap distorsi visual, dan cocok untuk dataset citra, terutama pada kasus klasifikasi penyakit daun singkong yang mengandalkan pola visual detail sebagai dasar prediksi.



**Gambar 3.** Grafik Distribusi Dataset Setelah *Random OverSampling*

Pada gambar 3 adalah hasil *random oversampling* untuk meningkatkan data dari minoritas menjadi setara dengan mayoritas.



### 3.1.3.2 Resize

Pada penelitian ini, ukuran input citra ditetapkan sebesar 32×32 piksel sebelum dimasukkan ke arsitektur ResNet152V2. Pemilihan ukuran tersebut bertujuan untuk mempercepat proses komputasi dan mengurangi beban pemrosesan tanpa menghilangkan informasi penting yang diperlukan dalam klasifikasi. Meskipun ResNet152V2 pada umumnya menggunakan ukuran input lebih besar seperti 224×224 piksel, model ini tetap mampu beradaptasi dengan resolusi yang lebih kecil karena karakteristik *residual learning* yang memungkinkan jaringan mempelajari fitur secara mendalam dan stabil. Penggunaan shape 32×32 juga dinilai efektif mengingat dataset yang digunakan berjumlah besar, sehingga ukuran yang lebih kecil dapat mempercepat *training*, menurunkan kebutuhan memori GPU, serta mengurangi risiko *overfitting*.

### 3.1.3.3 Pembagian Data

Proses pembagian dataset dilakukan dengan memisahkan data ke dalam dua kelompok, yaitu data pelatihan dan data uji. Data pelatihan digunakan selama proses pelatihan model, sedangkan data uji berfungsi untuk mengukur performa model secara objektif. Pembagian dilakukan secara acak menggunakan rasio tertentu antara kedua jenis data tersebut. Pada tahap pertama, dataset dibagi menjadi data pelatihan dan data uji dengan rasio 80:20. Pembagian ini dirancang untuk memastikan bahwa model memperoleh jumlah data yang memadai untuk belajar dan divalidasi, sekaligus menyediakan data terpisah yang belum pernah dilihat sebelumnya untuk tahap pengujian akhir. Hasil pembagian dataset dapat dilihat pada tabel 1.

**Tabel 1.** Dataset Dengan Pembagian 80:20

No	Katagori	Data Latih	Data Uji
1.	<i>Bacterial Blight</i>	10.526	2.632
2.	<i>Brown Streak</i>	10.526	2.632
3.	<i>Green Mottle</i>	10.526	2.632
4.	<i>Healthy</i>	10.526	2.632
5.	<i>Mosaic Disease</i>	10.526	2.632
Total		52.630	13.160

Pada table 1 ini merupakan pembagian data dibuat dengan rasio 80% data pelatihan dan 20% data uji.

### 3.1.4 Modeling

Pada tahap pelatihan model, digunakan sejumlah parameter seperti data pelatihan, jumlah *epoch*, data validasi, *batch size*, serta *callbacks*. Data pelatihan berfungsi sebagai sumber utama bagi model untuk mempelajari pola. Parameter *epoch* menentukan berapa kali keseluruhan data pelatihan diproses, dan dalam penelitian ini jumlah maksimalnya ditetapkan sebanyak 30 *epoch*. Data validasi merupakan data yang telah dipisahkan sebelumnya dan digunakan untuk mengevaluasi performa model selama proses pelatihan berlangsung. *Batch size* yang digunakan adalah 64. Sementara itu, *Callbacks* digunakan untuk menghentikan proses pelatihan apabila tidak terjadi peningkatan akurasi pada data validasi selama lima *epoch* berturut-turut. Pada tahap ini menggunakan model dengan arsitektur ResNet152V2.

Pada tahap selanjutnya tabel 2 membangun arsitektur model CNN dengan berbasis ResNet152V2. Berikut ini pada table arsitektur model.



**Tabel 2.** Membangun Arsitektur Model CNN dengan berbasis ResNet152V2

No	Keterangan	Output shape
1.	<i>input</i>	32, 32, 3
2.	<i>True divide</i>	32, 32, 3
3.	<i>Subtract</i>	32, 32, 3
4.	<i>ResNet152V2</i>	1, 1, 2048
5.	<i>Flatten</i>	2048
6.	<i>Dropout</i>	2048
7.	<i>Dense+softmax</i>	5

Tahap selanjutnya menjalankan pelatihan model dengan *ReduceLRonPlateau* untuk menurunkan *learning rate* secara otomatis jika model mengalami pemberhentian peningkatan pada *validasi loss*, dan menggunakan *EarlyStoping* untuk menghentikan pelatihan lebih awal jika tidak ada peningkatan pada *validasi loss*. Pada gambar 27 *sourcecode* pelatihan model.

Table 3 merupakan hasil dari pelatihan model dengan *epoch 30*, *batch\_size* yang digunakan sebesar 64.

**Tabel 3.** Hasil Pelatihan Model CNN Berbasis ResNet152V2

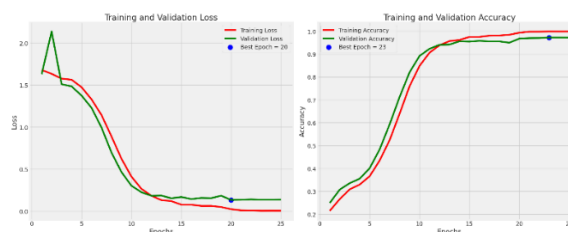
Epoch	Accuracy	Validation Accuracy	Loss	Validation Loss
1	0.2019	0.2496	1.6845	1.6273
2	0.2496	0.3079	1.6497	2.1356
3	0.3010	0.3358	1.5857	1.5092
4	0.3262	0.3554	1.5600	1.4828
5	0.3499	0.4014	1.5124	1.3738
6	0.4211	0.4834	1.3643	1.2280
7	0.5018	0.5929	1.1817	0.9934
8	0.6149	0.7126	0.9430	0.6967
9	0.7368	0.8205	0.6747	0.4651
10	0.8360	0.8922	0.4432	0.3019
11	0.9009	0.9232	0.2776	0.2251
12	0.9349	0.9409	0.1938	0.1826
13	0.9586	0.9419	0.1252	0.1853
14	0.9620	0.9558	0.1225	0.1529
15	0.9755	0.9546	0.0766	0.1681
16	0.9754	0.9580	0.0735	0.1441

17	0.9808	0.9556	0.0604	0.1572
18	0.9826	0.9555	0.0556	0.1546
19	0.9848	0.9498	0.0487	0.1834
20	0.9906	0.9676	0.0312	0.1332
21	0.9978	0.9698	0.0091	0.1346
22	0.9977	0.9702	0.0080	0.1398
23	0.9990	0.9725	0.0052	0.1349
24	0.9988	0.9725	0.0050	0.1360
25	0.9988	0.9722	0.0056	0.1375

Dilihat pada Tabel 3, terlihat bahwa model mencapai akurasi pelatihan tertinggi pada *epoch* ke-25 dengan nilai 0,9988 atau setara 99,8%. Pada *epoch* ke-19 dan 23, mekanisme *callbacks ReduceLRonPlateau* mulai bekerja karena tidak terjadi peningkatan pada data validasi selama tiga *epoch* berturut-turut. Nilai *loss* pelatihan yang dicapai berada pada angka yang relatif rendah, yaitu 0,0056. Pada *epoch* yang sama, akurasi validasi tercatat sebesar 0,9722 atau 97,2%, dengan nilai *loss* validasi sebesar 0,1375. Pada *epoch* ke-25 *callbacks EarlyStopping* aktif dikarenakan tidak terjadi penurunan nilai *loss* pada dataset validasi selama lima *epoch*. Perbedaan akurasi antara data pelatihan dan data validasi yang kurang dari 5% menunjukkan bahwa model tidak mengalami *overfitting*.

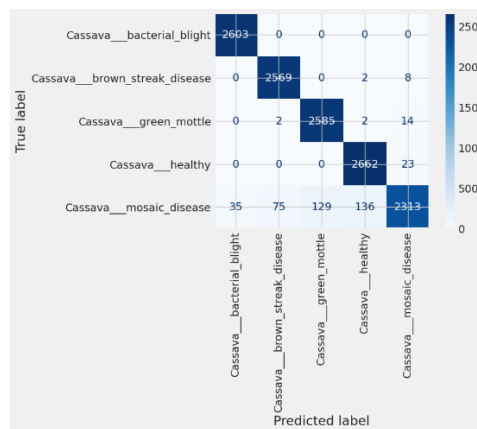
### 3.1.5 Evaluasi Model

Berdasarkan hasil pelatihan yang tercantum pada tabel 3, dilakukan proses visualisasi berupa grafik yang menampilkan perkembangan akurasi dan nilai kerugian (*loss*) selama pelatihan maupun validasi. Visualisasi ini berfungsi untuk membandingkan kinerja model pada data pelatihan dan data validasi. Grafik hasil visualisasi tersebut ditampilkan pada gambar 4.



**Gambar 4.** Grafik hasil visualisasi

Dari grafik tersebut terlihat bahwa peningkatan akurasi pada data pelatihan berlangsung lebih stabil, sedangkan akurasi pada data validasi menunjukkan perubahan yang naik-turun. Pada grafik *loss*, terlihat tren yang positif karena nilai kerugian terus menurun, dengan kurva pelatihan tampak lebih konsisten. Perbedaan stabilitas antara kurva pelatihan dan validasi merupakan hal yang umum terjadi akibat perbedaan distribusi data pada kedua dataset tersebut. Dataset pelatihan yang lebih besar membuat model lebih mudah mengenali pola, sementara data validasi dapat menghasilkan nilai yang lebih bervariasi karena memiliki keragaman lebih tinggi atau pola yang kurang terwakili. Meski demikian, perbedaan tersebut dapat meningkatkan risiko terjadinya *overfitting*.



**Gambar 5.** Confusion Matrix

Berdasarkan *confusion matrix* yang ditampilkan, terlihat bahwa model mampu mengklasifikasikan 12.642 dari total 13.158 citra pada dataset pengujian yang sebelumnya tidak pernah digunakan dalam pelatihan. Secara rinci, model berhasil mengidentifikasi dengan benar 2.603 citra kelas *Cassava Bacterial Blight*, 2.560 citra kelas *Cassava Brown Streak Disease*, 2.575 citra kelas *Cassava Green Mottle*, 2.638 citra kelas *Healthy*, dan 2.266 citra kelas *Cassava Mosaic Disease*. Hasil ini juga menunjukkan bahwa model masih mengalami tantangan dalam membedakan beberapa kelas yang memiliki kemiripan karakteristik, terutama pada kelas *Cassava Brown Streak Disease*, *Cassava Green Mottle*, *Healthy*, dan *Cassava Mosaic Disease*. Meski demikian, performa keseluruhan model tergolong baik dan menunjukkan kemampuan generalisasi yang cukup kuat terhadap data baru, sebagaimana didukung oleh nilai akurasi pada tabel 4.

**Tabel 4.** Matriks Evaluasi Model

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Cassava Bacterial Blight</i>	0.99	1.00	0.99	2.603
<i>Cassava Brown Streak Disease</i>	0.97	1.00	0.98	2.579
<i>Cassava Green Mottle</i>	0.95	0.99	0.97	2.603
<i>Healthy</i>	0.95	0.99	0.97	2.685
<i>Cassava Mosaic Disease</i>	0.98	0.86	0.92	2.688
<i>Accuracy</i>			0.97	13.158
<i>Macro Avarage</i>	0.97	0.97	0.97	13.158
<i>Weighted Avarage</i>	0.97	0.97	0.97	13.158

Berdasarkan tabel 4, model mencapai akurasi pengujian sebesar 97%. Performa tiap kelas juga tinggi, dengan *precision*, *recall*, dan *f1-score* berkisar antara 92% hingga 100%. Kelas *Cassava Bacterial Blight* dan *Cassava Brown Streak Disease* menunjukkan hasil terbaik, sedangkan *Cassava Mosaic Disease* memiliki *f1-score* terendah (92%) karena *recall* yang lebih rendah. Nilai *macro average* dan *weighted average* seluruhnya mencapai *precision* 97%, *recall* 97%, dan *f1-score* 97%, menunjukkan performa yang konsisten di semua kelas. Secara keseluruhan, model memiliki kemampuan generalisasi yang baik

tanpa indikasi *overfitting*, berdasarkan kesesuaian kinerja antara data pelatihan dan pengujian.

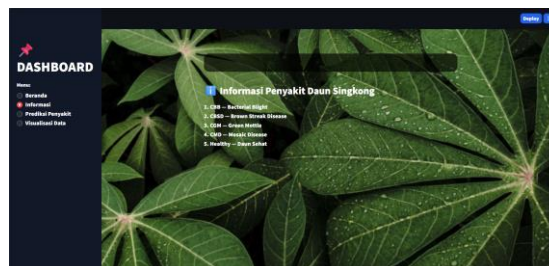
### 3.1.6 Deployment

Setelah model CNN dengan arsitektur ResNet152V2 serta skenario pembagian data 80% untuk pelatihan dan 20% untuk pengujian ditetapkan sebagai model terbaik, tahap berikutnya adalah mengembangkan aplikasi berbasis web menggunakan *framework Streamlit* melalui *editor Visual Studio Code*. Aplikasi ini dibuat agar proses deteksi penyakit daun singkong dapat diakses dengan lebih praktis.



Gambar 6. Halaman Beranda

Pada halaman beranda ini terdapat informasi mengenai penjelasan aplikasi dan singkong yang telah dibuat. Pada pilihan halaman terdapat 3 halaman selain beranda, yaitu halaman informasi, halaman prediksi penyakit, dan halaman visualisasi data. Tampilan halaman informasi seperti gambar 7.



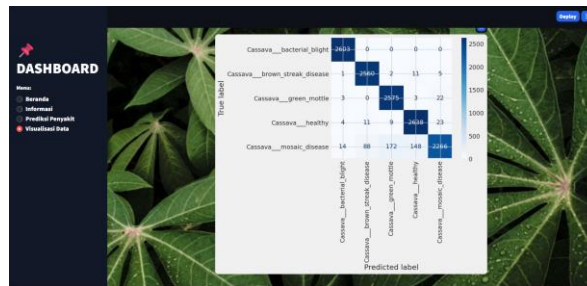
Gambar 7. Halaman Informasi

Pada halaman informasi menjelaskan tentang penyakit daun singkong tandanya. Tampilan halaman prediksi penyakit seperti gambar 8.



Gambar 8. Halaman Prediksi Penyakit

Pada halaman prediksi penyakit ini dapat mengupload data gambar singkong dan mendapat hasil prediksi dengan menampilkan penyakit dan akurasinya. Tampilan halaman visualisasi data seperti gambar 9.



**Gambar 9.** Visualisasi Data

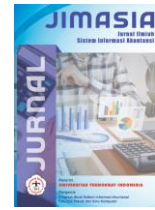
Pada halaman visualisasi data ini menampilkan hasil *confusion matrix*s dan grafik dari hasil dari pelatihan model yang sudah dilakukan dengan pembagian data 80% data latih dan 20% data uji.

### 3.1.7 Uji Coba

Pengujian dilakukan untuk menilai seberapa efektif model bekerja ketika diterapkan pada website. Pada penelitian ini, proses uji coba bertujuan memastikan bahwa model mampu memberikan keluaran yang sesuai dan tetap berfungsi dengan baik saat dijalankan melalui platform web. Uji coba menggunakan 3 gambar dari berbagai kelas dataset penyakit daun singkong secara acak dilihat dari nilai prediksinya. Hasil uji coba dapat dilihat dalam tabel 5.

**Tabel 5.** Hasil Uji Coba

Kategori	Presiksi yang diinginkan	Hasil prediksi website
<i>Cassava Bacterial Blight 1</i>	CBB	CBB (97%)
<i>Cassava Bacterial Blight 2</i>	CBB	CBB (97%)
<i>Cassava Bacterial Blight 3</i>	CBB	CBB (85%)
<i>Cassava Brown Streak Disease 1</i>	CBSD	CBSD (100%)
<i>Cassava Brown Streak Disease 2</i>	CBSD	CBSD (100%)
<i>Cassava Brown Streak Disease 3</i>	CBSD	CBB (79%)
<i>Cassava Green Mottle 1</i>	CGM	CGM (100%)
<i>Cassava Green Mottle 2</i>	CGM	H (48%)
<i>Cassava Green Mottle 3</i>	CGM	CGM (62%)
<i>Healthy 1</i>	H	H (43%)
<i>Healthy 2</i>	H	H (75%)



Healthy 3	H	CMD (95%)
<i>Cassava Mosaic Disease 1</i>	CMD	CMD (97%)
<i>Cassava Mosaic Disease 2</i>	CMD	CMD (98%)
<i>Cassava Mosaic Disease 3</i>	CMD	CBSD (99%)

Berdasarkan hasil uji coba tersebut ada beberapa prediksi yang tidak tepat dengan dataset yang digunakan.

### 3.1.8 Pembahasan

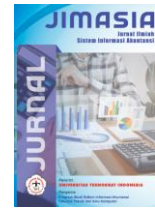
Penelitian ini mengangkat topik mengenai Penerapan Model *Convolutional Neural Network* (CNN) Untuk Klasifikasi Penyakit Daun Pada Tanaman Singkong (*Manihot Esculenta Crantz*). Tujuan utama penelitian adalah memprediksi penyakit tanaman singkong. Model yang digunakan dalam proses ini adalah algoritma *Convolutional Neural Network* (CNN) arsitektur ResNet152V2. Pada pengujian ini menggunakan beberapa kali pengujian dengan pembagian data yaitu 60:40, 70:30, dan 80:20. Dengan pembagian data 60% data latih dan 40% data uji mendapatkan hasil akurasi 0,9378 atau 93,78 dengan loss 0,2527, dan pembagian data 70% data latih dan 30% data uji mendapatkan hasil akurasi 0,8810 tau 88,10% dengan loss 0,3582.

Berdasarkan hasil pada tabel eval model, dengan skenario pembagian data 80% data latih dan 20% data uji memberikan performa terbaik dengan akurasi sebesar 0,9722 atau 97,2%, sehingga dipilih untuk tahap *deployment*. Pada penelitian ini, nilai akurasi tersebut dijadikan dasar dalam pemilihan model yang akan diterapkan. Tahap *deployment* bertujuan untuk mengintegrasikan model yang telah dilatih ke dalam sistem atau aplikasi yang dapat digunakan untuk memprediksi penyakit daun singkong. Fitur ini diharapkan dapat dimanfaatkan oleh pihak terkait, seperti petani dan masyarakat yang masih kurangnya pengetahuan tentang penyakit daun singkong.

## 4. KESIMPULAN

Penelitian ini menunjukkan bahwa model *Convolutional Neural Network* (CNN) dengan arsitektur ResNet152V2 mampu mengklasifikasikan penyakit daun singkong secara efektif. Dataset yang digunakan berasal dari *Kaggle* (Nirmal Sankalana, 2023) dengan total 21.397 citra daun pada lima kategori: *Cassava Bacterial Blight*, *Brown Streak Disease*, *Green Mottle*, *Mosaic Disease*, dan *Healthy*. Proses klasifikasi meliputi persiapan data, *exploratory data analysis*, *prapemrosesan* dengan *batch size* 64, ukuran citra 32x32, serta *random oversampling* untuk mengatasi ketidakseimbangan data. Dengan pembagian data latih dan uji 80:20, model mencapai akurasi 97,2%. Sebelum mendapatkan hasil yang lebih baik, penelitian ini telah melakukan menjalankan pelatihan model tanpa menggunakan *Random Oversampling* untuk meratakan dataset perkelas mendapatkan hasil yang kurang baik dengan akurasi sebesar 0,4639 atau 46,4% dengan *loss* 1,2707 dan validasi akurasi sebesar 0,4771 atau 48% dengan *loss* validasi 1,2284. Model ResNet152V2 diimplementasikan pada aplikasi web berbasis *Streamlit*, dan dapat berjalan sesuai fungsi meskipun beberapa prediksi masih keliru akibat kemiripan citra pada *confusion matrix*. Hasil penelitian ini bermanfaat bagi petani dan masyarakat dalam mendeteksi penyakit daun singkong secara dini, sehingga penyebaran penyakit dapat dicegah dan produktivitas panen dapat meningkat. Penelitian ini tidak hanya berkontribusi pada pengembangan metode klasifikasi, tetapi juga mendukung peningkatan hasil pertanian bagi petani singkong.





## 5. REFERENCES

- [1] M. Faturrachman, I. Yustiana, and . S., "Sistem Pendeteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Deep Learning Dan Tensorflow Berbasis Android," *IJIS - Indones. J. Inf. Syst.*, vol. 7, no. 2, p. 176, 2022, doi: 10.36549/ijis.v7i2.225.
- [2] A. I. Ramadhan, J. T. Hardinata, and Y. P. Purba, "Analisa Jaringan Saraf Tiruan Backpropagation Untuk Memprediksi Prestasi Siswa SMA Muhammadiyah Serbelawan," *Brahmana J. Penerapan Kecerdasan Buatan*, vol. 3, no. 1, pp. 18–26, 2021, doi: 10.30645/brahmana.v3i1.88.
- [3] E. R. Bawana, E. Seniwati, and P. F. Kristami, "Studi Kinerja Algoritma K-Nearest Neighbors ( KNN ) untuk Klasifikasi Pasien Diabetes," vol. 4, no. 2, pp. 169–176, 2025.
- [4] Desvita Fitri Amalia, "Deteksi Penyakit Pada Daun Singkong Dengan Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)," vol. 1, pp. 1–8, 2024.
- [5] M. Liu, H. Liang, and M. Hou, "Research on cassava disease classification using the multi-scale fusion model based on EfficientNet and attention mechanism," no. December, pp. 1–11, 2022, doi: 10.3389/fpls.2022.1088531.
- [6] F. Marpaung, F. Aulia, and R. C. Nabila, *Computer Vision Dan Pengolahan Citra Digital*. 2022. [Online]. Available: [www.pustakaaksara.co.id](http://www.pustakaaksara.co.id)
- [7] J. D. P. I Komang Tri Aditya Pariyana , Anak Agung Ngurah Mahendra Adhi Putra, I Gusti Gede Sumartana, "Identifikasi Penyakit Buah Kakao Jembrana Menggunakan Metode Resnet152v2," vol. 05, no. 1, pp. 12–20, 2024.
- [8] A. S. Andreanov Ridhovan, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," vol. 07, pp. 58–65, 2022.
- [9] V. No, J. Hal, M. Doddy, A. Pranatha, M. A. Maricar, and G. H. Setiawan, "Implementasi Arsitektural Resnet-34 Dalam Klasifikasi Gambar Penyakit Pada Daun Kentang," vol. 6, no. 3, pp. 575–580, 2024.
- [10] F. A. Arafat, M. N. Ichsan, and M. F. Pramoedya, "Pemanfaatan Arsitektur MOBILENET-CNN Untuk Mendiagnosis Penyakit Pada Daun Singkong Melalui Teknologi Citra Digital," *Pros. Semin. Nas. Teknol. Dan Sains*, vol. 4, pp. 73–78, 2025.
- [11] I. M. N. Annisa Mustika Anggraeni, Teguh Iman Hermanto, "Klasifikasi Penyakit Daun Singkong Menggunakan Convolutional Neural Network (CNN) Dengan Arsitektur VGG16 Berbasis Android," vol. 11, no. 1, pp. 53–59, 2025.
- [12] W. M. Rahmawati, R. Edelani, P. Studi, T. Informatika, P. Elektronika, and N. Surabaya, "Deteksi PCOS pada Wanita Menggunakan Explanatory Data Analysis ( EDA ) dan Support Vector Machine ( SVM )," pp. 165–172, 2018.
- [13] J. Jasmir, "Implementasi Teknik Data Cleaning Dan Teknik Roughset Pada Data Tidak Lengkap Dalam Data Mining," *Semin. Nas. APTIKOM*, pp. 99–106, 2020.
- [14] J. Prasetya, "Penerapan Klasifikasi Naive Bayes dengan Algoritma Random Oversampling dan Random Undersampling pada Data Tidak Seimbang Cervical Cancer Risk Factors," *Leibniz J. Mat.*, vol. 2, no. 2, pp. 11–22, 2022, doi: 10.59632/leibniz.v2i2.173.
- [15] D. Septhya, K. Rahayu, S. Rabbani, and V. Fitria, "Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru," vol. 3, no. April, pp. 15–19, 2023.

