

# **Sistem Pakar Diagnosa Penyakit Pada Tanaman Sawi Menggunakan Metode Convolutional Neural Network Berbasis Android**

**Muhammad Yusuf<sup>\*</sup>, Syamsudin Aliphadji Talaohu, Jumria Purnamasari**

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong, Sorong, Indonesia

Email: <sup>1</sup>\*yusuf@um-sorong.ac.id, <sup>2</sup>syamsudintalaohu@gmail.com, <sup>3</sup>jumriapurnamasari15@gmail.com

Email Penulis Korespondensi: yusuf@um-sorong.ac.id

**Abstrak**—Sawi merupakan salah satu tanaman sayuran yang sangat mudah di budidayakan, karena mampu tumbuh di dataran tinggi maupun dataran rendah, tanaman sawi juga rentan terhadap penyakit yang dapat menurunkan hasil dan kualitasnya, identifikasi penyakit sawi secara manual merupakan hal yang sulit dan membutuhkan pengetahuan mendalam tentang gejala dan penyebab penyakit pada tanaman sawi. Tujuan penelitian ini adalah mengembangkan sistem yang mampu digunakan dalam mengidentifikasi penyakit tanaman sawi. Penyakit yang digunakan terdiri dari 3 jenis yaitu, leaf miner, busuk daun, dan ulat grayak. Dengan perkembangan teknologi saat ini, petani dapat dibantu untuk mengurangi kesalahan dalam mengidentifikasi penyakit tanaman sawi, Deep Learning adalah bidang dalam machine learning yang memanfaatkan artificial neural networks untuk menyelesaikan masalah dengan dataset yang besar. Salah satu algoritma yang sering digunakan dalam sistem deep learning adalah Convolutional Neural Network (CNN). Hasil yang didapatkan dengan dataset yang digunakan yaitu dengan accuracy tertinggi 100% train accuracy dan 97,78% validation accuracy, serta memperoleh hasil akurasi tertinggi dengan menggunakan f1-score yaitu 95,56%.

**Kata Kunci:** Sawi; Sistem Pakar; Deep Learning; CNN; Android

**Abstract**—Mustard greens are one of the vegetable crops that are very easy to cultivate, because they are able to grow in the highlands and lowlands, mustard greens are also susceptible to diseases that can reduce yields and quality, identifying mustard diseases manually is difficult and requires in-depth knowledge of the symptoms. and causes of disease in mustard plants. The aim of this research is to build a system that can be used to identify mustard plant diseases. The diseases used consist of 3 types, namely, leaf miner, leaf rot, and armyworm. With current technological developments that can help farmers minimize errors in determining diseases in mustard plants, Deep Learning is a field of machine learning that utilizes artificial neural networks to solve problems with large datasets. One algorithm that is often used in deep learning systems is Convolutional Neural Network (CNN). The results obtained with the dataset used were with the highest accuracy of 100% train accuracy and 97.78% validation accuracy, and obtained the highest accuracy results using f1-score, namely 95.56%.

**Keywords:** Mustard; Expert System; Deep Learning; CNN; Android

## **1. PENDAHULUAN**

Sayuran merupakan salah satu bagian dari menu makanan sehat, maka dari itu, tak heran jika permintaan terhadap sayuran terus bertambah seiring dengan kesadaran akan pentingnya mengonsumsi makanan sehat[1]. Diantara berbagai jenis sayuran yang bisa ditanam, sawi ialah salah satu sayuran yang memiliki nilai ekonomi dan prospek yang tinggi. Tanaman sawi adalah salah satu sayuran yang memiliki nilai ekonomi tinggi karena kaya akan serat, kandungan gizi yang tinggi, dan juga tanaman ini diyakini memiliki khasiat obat dan mudah untuk ditanam di berbagai daerah [2].

Tanaman sawi sendiri bisa ditanam baik di kawasan dataran tinggi ataupun di dataran rendah. Sawi juga masuk dalam kategori tanaman sayuran yang mampu bertahan saat hujan turun. Oleh karena itu, bisa ditanam sepanjang tahun, dengan catatan bahwa selama musim kemarau harus ada penyediaan air yang mencukupi untuk proses penyiraman. Proses budi daya tanaman sawi terbilang sederhana, sehingga dapat dijalankan dengan mudah oleh petani atau orang awam yang mau terlibat dalam bisnis tanaman ini[3][4]. Namun, tanaman sawi rentan terhadap penyakit yang dapat menurunkan hasil dan kualitasnya. Identifikasi penyakit sawi secara manual merupakan tugas yang sulit dan membutuhkan pengetahuan mendalam tentang gejala dan penyebab penyakit pada tanaman sawi[5].

Karena itu dalam kasus ini, perlu menggunakan pengetahuan seorang pakar yang akan dituangkan kedalam bidang komputer agar bisa menghasilkan suatu sistem yang bisa membantu identifikasi penyakit pada tanaman sawi. Dalam konteks ini, implementasi sistem pakar diagnosa penyakit pada tanaman sawi menggunakan algoritma CNN[6][7] berbasis android bertujuan untuk memberikan solusi yang cepat dan tepat akurat dalam mengidentifikasi penyakit dan memberikan saran kepada petani atau individu yang tertarik dengan budidaya tanaman sawi.

Penelitian yang sudah dilakukan sebelumnya berkaitan dengan aplikasi sistem pakar diagnose penyakit pada tanaman sawi diantaranya yaitu “Identifikasi Derris Tanaman Invasif Mangrove Trifoliasi Menggunakan Gambar UAV dan Algoritma deep learning”[8]. Hasil penelitian menunjukkan bahwa model-model pembelajaran mendalam yang diusulkan, Dibandingkan dengan LVG, LDN memiliki memperoleh akurasi tertinggi sebesar 93% pada identifikasi. Perbedaan antara penelitian ini dengan penelitian peneliti adalah dataset yang ada dalam penelitian ini hanya 400 dataset, sedangkan peneliti menggunakan 450 dataset. Selain itu, penelitian ini juga mengimplementasikan sistem pakar tersebut ke dalam aplikasi berbasis Android, yang memungkinkan pengguna untuk mendiagnosis penyakit tanaman sawi secara langsung melalui perangkat mobile mereka, suatu aspek yang belum diteliti sebelumnya.

Penelitian lain yang berjudul “Eksplorasi spektroskopi fluoresensi sinkron total dikombinasikan dengan convolutional neural network terlatih dalam identifikasi dan kuantifikasi minyak nabati”[9]. Hasil penelitian menunjukkan bahwa penggunaan jaringan saraf konvolusi (CNN), terutama VGG-16 sebagai ekstraktor fitur, dalam kombinasi dengan metode TSyF dapat membedakan jenis minyak nabati, mengidentifikasi minyak nabati yang

teradulterasi dan palsu dengan tingkat kesalahan validasi sebesar 0.81%. Terdapat perbandingan antara penelitian ini dengan penelitian peneliti yakni dataset yang terdapat pada penelitian ini hanya 344 dataset, sedangkan peneliti menggunakan 450 dataset. Dengan demikian, penelitian ini memperluas jumlah dataset yang digunakan dan memperkenalkan inovasi baru dalam bentuk aplikasi mobile untuk memudahkan akses dan penggunaan sistem pakar oleh para petani dan pengguna lainnya.

Adapun penelitian lainnya yaitu “Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Mengklasifikasi Hama Pada Citra Daun Tanaman Kopi”[10]. Hasil penelitian ini yaitu, arsitektur CNN Alexnet menghasilkan akurasi antara 69,44% hingga 80,56%. Selain itu terdapat perbandingan antara penelitian ini dengan penelitian peneliti yaitu pada dataset, dimana dataset yang terdapat pada penelitian ini hanya 300 dataset, sedangkan peneliti menggunakan 450 dataset. Perbedaan pada penelitian ini bertujuan untuk memperluas dan meningkatkan hasil yang telah dicapai sebelumnya dengan menggunakan 450 dataset, yang diharapkan dapat meningkatkan akurasi dan kinerja model dalam diagnosis penyakit pada tanaman sawi.

Terdapat juga penelitian dengan judul “Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network”[11]. Hasil penelitian ini menunjukkan bahwa *Convolutional Neural Network* dengan metode Transfer Learning dapat mengidentifikasi jamur yang bisa dimakan atau beracun dan didapatkan akurasi lebih dari 86%. Selain itu, terdapat perbandingan antara penelitian ini dalam hal pengumpulan dataset, dimana pada penelitian ini masih menggunakan dataset dari situs Kaggle sedangkan peneliti menggunakan dataset secara primer yaitu dengan mengambil gambar langsung di lapangan. Perbedaan ini diharapkan bisa memberikan hasil yang lebih representatif serta akurat.

Penelitian terakhir yang dijadikan referensi yaitu “Deteksi hama pada daun apel menggunakan algoritma Convolutional Neural Network”[12]. Hasil penelitian mengungkapkan bahwasanya ukuran citra, augmentasi data, dan jumlah epoch (iterasi) memiliki pengaruh yang signifikan terhadap akurasi model deteksi hama. Berdasarkan uji coba, model CNN pada parameter ukuran citra 256x256, augmentasi data horizontal flip, vertical flip, dan random rotation, serta jumlah epoch ke-60, didapatkan akurasi dengan data asli yakni 94,85%. Perbedaan yang terdapat pada penelitian ini dalam hal pengumpulan dataset, dimana pada penelitian ini masih menggunakan dataset dari situs Kaggle sedangkan peneliti menggunakan dataset secara primer yaitu dengan mengambil gambar langsung di lapangan. Perbedaan ini diharapkan mampu memberikan hasil yang lebih representatif serta akurat dalam mengidentifikasi penyakit pada tanaman sawi.

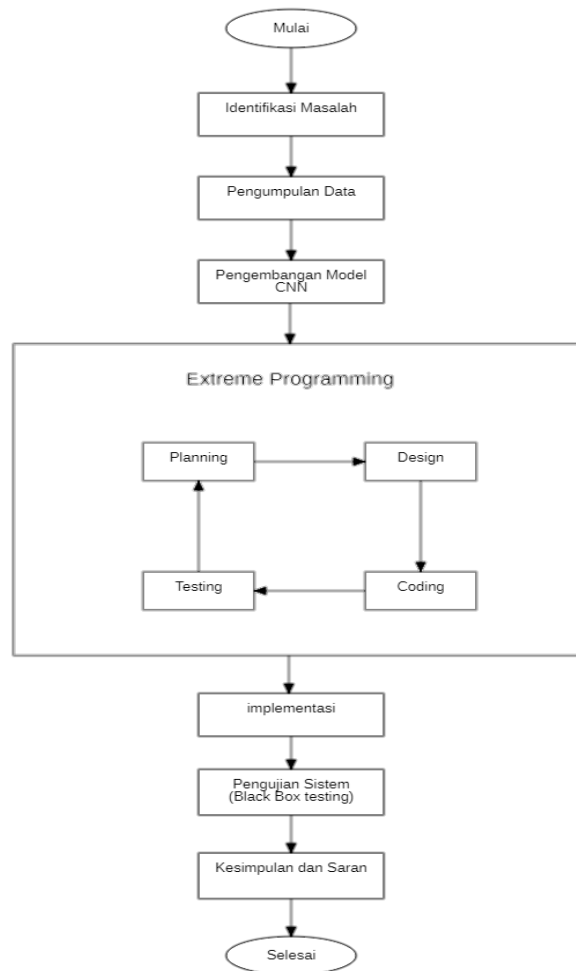
Dalam konteks ini, implementasi sistem pakar diagnosa penyakit pada tanaman sawi menggunakan CNN berbasis android bertujuan untuk memberikan solusi yang cepat dan tepat akurat dalam mengidentifikasi penyakit dan memberikan saran pengobatan kepada petani atau individu yang tertarik dengan budidaya tanaman sawi, harapannya penelitian ini bisa meningkatkan produktivitas serta kualitas hasil pertanian, serta mengurangi kerugian yang diakibatkan oleh penyakit tanaman[13][14]. Metode yang diterapkan pada implementasi ini yaitu *Convolutinal Neural Network* (CNN). CNN merupakan salah satu metode yang paling efektif dalam pengolahan citra. CNN mampu mengekstraksi fitur secara otomatis dari citra tanaman sawi yang terserang penyakit dan mampu mengklasifikasikan jenis penyakit dengan tingkat akurasi yang tinggi[15].

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini melibatkan beberapa tahapan yaitu, identifikasi masalah yaitu menentukan masalah utama yang dihadapi petani sawi dalam mengidentifikasi penyakit tanaman, pengumpulan data yaitu mengumpulkan data atau memberi label kelas terkait penyakit pada tanaman sawi dari seorang pakar, pengembangan Model *CNN* dengan menggunakan data yang terkumpul untuk melatih model *Convolutional Neural Network* (*CNN*) dalam mengidentifikasi penyakit tanaman sawi, pengembangan sistem dengan *Extreme Programming* untuk membangun sistem berbasis Android yang mengimplementasikan model *CNN* untuk digunakan oleh petani, selanjutnya implementasi hasil dari pengembangan aplikasi, dan menguji sistem yang dibangun dalam kondisi nyata untuk memastikan fungsionalitasnya, terakhir yaitu kesimpulan dan saran dalam mengevaluasi hasil penelitian dan memberikan rekomendasi untuk pengembangan lebih lanjut.

Pada gambar 1 pada alur penelitian merupakan tahapan atau alur yang digunakan pada penelitian. Dalam tahapan ini peneliti menggunakan tahapan metode *Extreme Programming* (XP) sebagai pengembangan perangkat lunak. Ada beberapa tahap pada metode *Extreme Programming*, pada tahap pertama, yaitu perencanaan, langkah yang dilakukan adalah menganalisis rencana kebutuhan yang diperlukan untuk pengembangan sistem. Selanjutnya, pada tahap desain, beberapa visualisasi dibuat yaitu flowchart, usecase, diagram aktivitas, serta antarmuka pengguna agar alur penelitian yang sedang diterapkan bisa dijelaskan. Kemudian, pada fase pengkodean, sesudah model *tensorflow* terbuat, file ini akan diganti formatnya menjadi *.tf lite*, ini nantinya digunakan dalam pembuatan aplikasi android. Terakhir, tahap pengujian dilaksanakan agar bisa memastikan dengan teliti semua fitur fungsional *software*.



Gambar 1. Alur Penelitian

## 2.2 Pengembangan Model CNN

Pada fase ini model *Convolutional Neural Network* akan dikembangkan agar bisa menerapkan Deep Learning dalam mengidentifikasi jenis penyakit sawi berdasarkan dengan rancangan sebelumnya, model ini dikembangkan dengan algoritma Convolutional Neural Network, memakai bahasa pemrograman Python, library TensorFlow Lite dalam membangun model deteksi, serta untuk menjalankan program menggunakan notebook dari Google Colab [16].

### a. Dataset

Dataset yang digunakan peneliti yaitu sawi dan terdiri atas 3 kelas yaitu leaf miner, ulat grayak, dan busuk daun. Dataset yang sudah didapat lalu dikumpulkan dengan cara gambar-gambar diambil langsung di lokasi studi kasus. Setelah itu, dataset di letakkan ke Google Drive yang memiliki fungsi sebagai tempat penyimpanan di Google Colab.

### b. Pelabelan Dataset

Pada fase ini dataset yang sudah terkumpul akan diberi label sesuai kelas yaitu leaf miner, ulat grayak, dan busuk daun.

### c. Split Data

Pada fase ini dilakukan pembagian dataset yang sudah terkumpul di dalam 3 kelas dengan pembagian 70% data latih, 20% data uji dan 10% data validasi.

### d. Processing Data

Selanjutnya, pada langkah ini peneliti menerapkan pemrosesan data dengan mengganti dimensi yaitu height 128, width 128, channels 3, dan batch size 64. Pemrosesan data ini memanfaatkan image data generator yang tersedia di TensorFlow Lite.

### e. Membangun Model CNN

Di fase ini, peneliti merancang model agar bisa menjalankan aplikasi sistem pakar diagnosa penyakit sawi dengan menerapkan metode *Convolutional Neural Network* (CNN). Disamping hal tersebut, peneliti juga menerapkan arsitektur model yaitu VGG-16. Untuk membangun aplikasi, pada tahap pertama convolution layer akan dilakukan pengurangan informasi citra dari inputan yang dimasukkan untuk mengekstraksi fitur-fitur penting, kemudian pooling dilakukan untuk

mereduksi ukuran fitur yang dihasilkan oleh lapisan pertama untuk memperkecil resolusi dari fitur dan mempertahankan informasi penting tersebut untuk mempercepat komputasi, kemudian masuk ke fully connected layer untuk mengklasifikasikan hasil yang didapat dari tahap pooling, selanjutnya dataset akan di resize menjadi 224x224 piksel kemudian akan dibagi menjadi dua data yaitu data latih untuk menjadi inputan dari pelatihan CNN dengan arsiteksur VGG-16, dan data uji. Setelah model dibangun data uji akan digunakan untuk uji kinerja sistem pada model yang dibangun.

f. Training Model

Pada tahap ini dilakukan pelatihan data dengan jumlah epoch 30 dan melakukan pengujian terhadap model VGG-16 untuk sistem yang telah dikembangkan.

g. Pengujian Model

Di langkah ini, peneliti menerapkan pengujian terhadap model yang sudah dikembangkan dalam menyusun skrip evaluasi agar memverifikasi keakuratan prediksi. Data yang dipakai pada *testing* tentunya berbeda pada data pelatihan serta validasi.

h. Saving Model

Di langkah ini, peneliti akan melakukan penyimpanan model agar nantinya bisa diterapkan pada aplikasi. Hasil pada proses ini akan menghasilkan file dengan format .h5. Selanjutnya, data latihan akan dikonversi menjadi TensorFlow Lite untuk mempermudah pemodelan aplikasi di Android. Setelah konversi selesai, aplikasi akan dimodelkan dengan memakai Android Studio.

### 2.3 Pengembangan Sistem

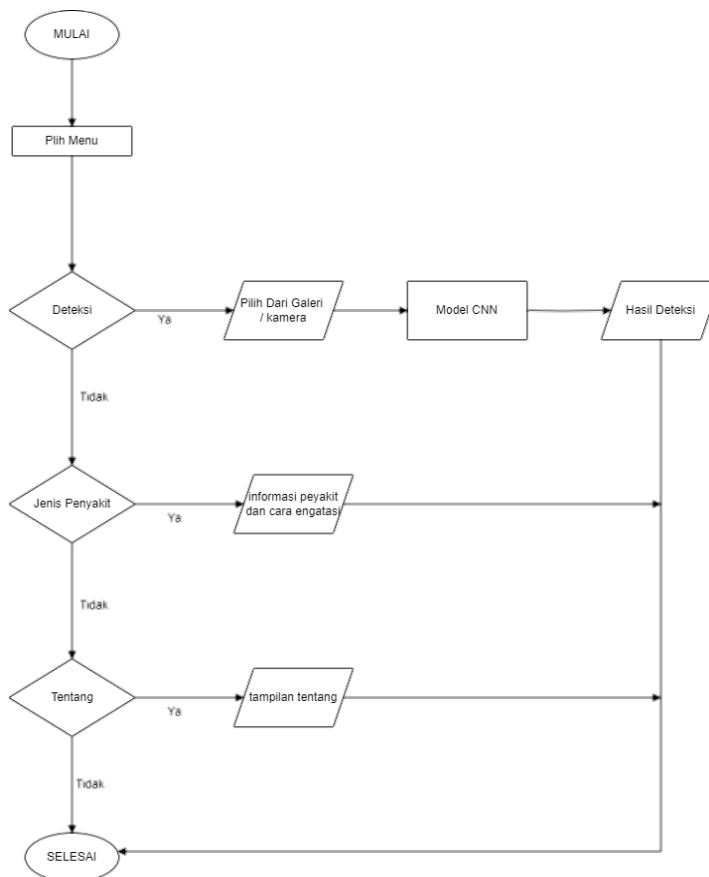
Dalam tahap pengembangan sistem, peneliti merancang serta membangun sesuatu aplikasi yang nantinya diterapkan dalam pemecahan masalah yang sudah diuraikan Latar Belakang yaitu Sistem Pakar Diagnosa Penyakit Tanaman Sawi Menggunakan metode CNN Berikut adalah tahapan-tahapan pada metode metode *Extreme Programming*[17][18]:

a. Planning

Fase ini mencakup penentuan konteks bisnis aplikasi, definisi output, fitur-fitur yang tersedia, fungsi aplikasi yang dikembangkan, penentuan waktu hingga biaya, serta alur kinerja aplikasi.

b. Design

Pada fase desain ini, dibuat visualisasi flowchart yang bertujuan dalam menjelaskan visualisasi proses kinerja aplikasi yang ingin dibuat. Hal ini bisa terlihat di gambar 2 berikut:



Gambar 2. Flowchart Sistem

Pada gambar 2 di atas terdapat 3 menu yaitu deteksi, jenis penyakit dan tentang. Pada halaman deteksi pengguna dapat melakukan proses deteksi menggunakan kamera ataupun dari gallery dan akan mendapat hasil deteksi, lalu pada halaman jenis penyakit pengguna bisa mendapat informasi penyakit dan cara penanganannya, dan halaman tentang untuk melihat informasi tentang aplikasi.

c. Coding

Di fase ini sesudah pembuatan model tensorflow lite, ekstensi file nantinya diubah menjadi file dengan ekstensi *.tflite* dimana ini nantinya diterapkan pada pembuatan aplikasi android. Pengembangan aplikasi ini akan dikerjakan dengan memanfaatkan perangkat lunak Android Studio dan bahasa pemrograman Java.

d. Testing

Tahap pengujian dilakukan melalui pengujian *blackbox*, yang berarti diperiksa dalam keseluruhan semua fitur fungsional pada *software*. Pada pengujian awal dilihat apakah aplikasi telah terpenuhi persyaratan yang diinginkan. Kemudian, penguji memeriksa sebuah input apakah valid atau tidak untuk dipastikan kalau proses berjalan secara baik dan benar. Sesudah *testing* selesai, penguji memberikan semua bug yang ditemui untuk developer bisa memperbaikinya.

### 2.4 Convolutional Neural Network

Convolutional Neural network (CNN) merupakan bagian dari kelompok jaringan saraf tiruan yang diterapkan dalam mengenali dan memproses gambar. CNN meniru cara interaksi neuron dalam otak kita, dan memiliki arsitektur serupa. Secara prinsip CNN tidak berbeda jauh dengan jenis Neural Network lainnya, terdiri dari neuron dengan bobot, bias, dan fungsi aktivasi[19].

### 2.5 Android

Operasi Android terutama dirancang untuk layar sentuh layar sentuh seperti ponsel, tablet, dan smartphone. OS Android adalah berdasarkan kernel Linux dan software dari sumber terbuka lainnya. Kita tahu bahwa Android adalah sumber terbuka, sehingga menjadi sistem operasi yang paling cepat berkembang untuk ponsel. Sistem operasi seluler juga disebut sebagai OS seluler yang mencakup layar sentuh, seluler, Bluetooth, Wi-Fi, GPS navigasi seluler, kamera HD, kamera video, kamera video, pengenalan suara pengenalan suara, pemutar musik, perekam suara, browser[20].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengumpulan Data

Pada penelitian ini, peneliti menggunakan dataset sawi yang terdiri atas berbagai macam penyakit, lalu terbagi menjadi 3 kelas yaitu leaf miner, busuk daun, dan ulat grayak. Dataset ini telah terkumpul selama peneliti mengadakan observasi

Tabel 1. Dataset penyakit

No	Label	Train	Validation	Test
1.	Busuk Daun	109	19	34
2.	Leaf Miner	103	13	28
3.	Ulat Grayak	103	13	28
	Jumlah	350	45	90

Dari tabel 1, terlihat bahwa dataset terbagi menjadi 3 kelas yang terbagi dalam 3 folder yakni train, validation, serta test. Pada folder train terbagi atas 3 kelas dengan jumlah 350 dataset. Pada folder validation dengan 3 kelas berjumlah 45 dataset. Dan pada folder test dengan 3 kelas berjumlah 90 dataset. Total seluruh dataset yang dipakai di penelitian ini yaitu 450 dataset.

### 3.2 Pengembangan Model CNN

a. Dataset

Dataset yang digunakan peneliti yaitu sawi yang terbagi menjadi 3 kelas, yaitu leaf miner, ulat grayak, dan busuk daun. Dataset yang sudah didapat lalu dikumpulkan dengan mengambil gambar langsung di lokasi studi kasus. Contoh dataset bisa dilihat di gambar 3.



Gambar 3. Visualisasi Dataset

Pada gambar 3 di atas ialah contoh dataset dari penyakit sawi yaitu Ulat Grayak, Busuk Daun, dan *Leaf Miner*.

b. Pelebelan dataset

Di langkah ini data yang sudah terkumpul akan diberi label sesuai kelas yaitu leaf miner, ulat grayak, dan busuk daun.

c. Split data

Di langkah ini peneliti melakukan pembagian dataset yang telah terkumpul ke dalam 3 kelas dengan pembagian 70% data latih, 20% data uji dan 10% data validasi

d. Processing data

Di langkah ini peneliti melakukan pemrosesan data dengan cara mengubah dimensi yaitu height 128, width 128, channels 3, dan batch size 64. . Pemrosesan data ini memanfaatkan image data generator yang tersedia di TensorFlow Lite.

e. Membangun model CNN

Di langkah ini peneliti menerapkan arsitektur model VGG-16 Untuk membangun sistem VGG-16 terdiri dari 16 lapisan total, termasuk 13 lapisan konvolusional dan 3 lapisan yang benar-benar terhubung. Untuk membentuk lapisan konvolusional, digunakan kernel 3x3 melalui strides 1. Di setiap blok akhir, max pooling dilakukan melalui strides 2, serta ukuran pooling adalah 2x2. Berikut kode yang digunakan pada google colabatory:

```
[ ] model_name='syam'
print("Building model with", base_model)
model = tf.keras.Sequential([
    # Note the input shape is the desired size of the image 128x128 with 3 bytes color
    # This is the first convolution
    base_model,
    tf.keras.layers.Conv2D(filters=32, padding='same', kernel_size=3, activation='relu', strides=1),
    tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
    tf.keras.layers.Dropout(rate=0.5),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(3, activation='softmax')
])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001), loss='categorical_crossentropy', metrics='accuracy')
```

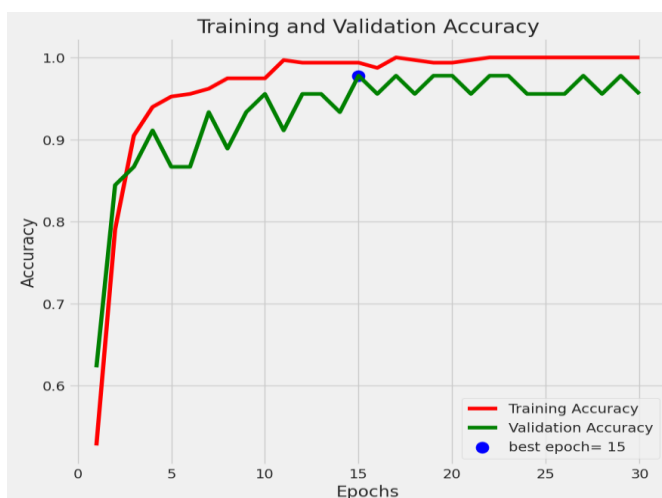
Building model with <keras.src.engine.functional.Functional object at 0x78d9750b5540>

Gambar 4. Pembuatan Model

Pada gambar 4 di atas ialah pembuatan Model VGG-16 dengan memakai generator gambar library Tensorflow menggunakan `img_shape` 128 x 128.

f. Training Model

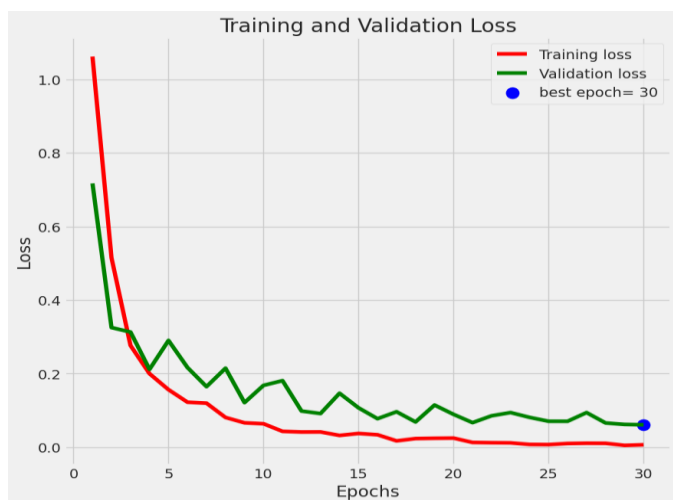
Setelah melalui proses pembuatan model, selanjutnya pada fase ini peneliti melakukan pelatihan data dengan jumlah epoch 30 dan menguji model VGG-16 untuk sistem yang dibangun. Waktu pelatihan yang dilakukan yaitu kurang lebih 1 jam. Berikut grafik dari hasil training yang dilakukan:



Gambar 5. Grafik Akurasi Training dan Validasi

Pada gambar 5 di atas menampilkan grafik akurasi train yang hampir mencapai angka 1 seiring bertambahnya epoch, baik di dataset training maupun validation. Sehingga di epoch ke-15 didapatkan setiap nilai akurasi training dan

validation yaitu 0,9937 dan 0,9778. Garis merah pada grafik menunjukkan accuracy data training, sementara grafik warna hijau menunjukkan accuracy data validation.

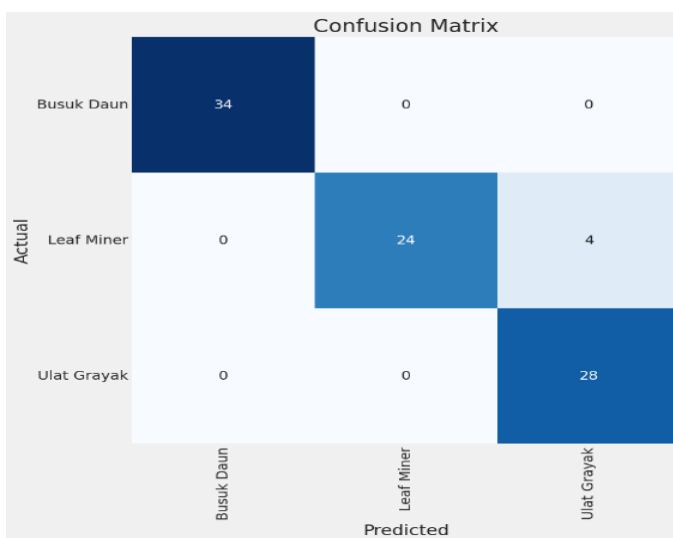


Gambar 6. Grafik Loss Training dan Validation

Gambar 6 di atas, menampilkan grafik *loss training* yang mendekati angka 0 sejalan dengan bertambahnya epoch untuk dataset *training* dan *validation*. Nilai *loss training* adalah 0,0065, dan nilai *loss validation* yaitu 0,0609. Perbedaan antara grafik loss dapat dilihat berdasarkan warna pada garis grafik. Pada garis grafik menampilkan *loss* yang mendekati nilai 0, oleh karena itu bisa disimpulkan kalau *training* serta *validation* sudah berhasil mengklasifikasi secara baik. Disamping hal tersebut, *loss training* dan *validation* terletak di posisi yang cukup sama atau paralel sepanjang waktu yang dilakukan, maka dari itu grafik ini bisa disebut grafik *goodfit*, yang menunjukkan bahwa kegagalan pelatihan pada dataset yang sama daripada yang ditunjukkan pada grafik.

#### g. Pengujian Model

Di langkah ini untuk mengukur tingkat akurasi pada arsitektur VGG-16 serta pengujian pada label prediksi maka perlu dilakukan pengujian model. Berikut hasil confusion matrix pada gambar 7.



Gambar 7. Confusion Matrix

Berdasarkan gambar 7 tersebut, prediksi menggunakan arsitektur VGG-16 berdasarkan dengan menggunakan confusion matrix. Dari pengujian kelas Busuk Daun, Precision bisa tercapai di angka 100%, Recall 100% dan f1-score 100%. Lalu dari kelas Leaf Miner pengujiannya didapatkan Precision tercapai di angka 100%, Recall 86% dan f1-score 92%. Dan dari pengujian kelas Ulat Grayak, Precision mendapat hasil di angka 88%, Recall 100% dan f1-score 93%. Dengan demikian, penggunaan model VGG-16 dapat menghasilkan tingkat akurasi 96%.




#### h. Saving Model

Di langkah ini, untuk menerapkan model pada aplikasi maka perlu dilakukan saving model. Hasil pada pemrosesan nantinya dapat menghasilkan file dengan format .h5. Selanjutnya, data latihan akan dikonversi menjadi TensorFlow Lite untuk mempermudah pembuatan aplikasi pada Android. Setelah konversi selesai, akan dilakukan pembuatan aplikasi dengan memakai software Android Studio.

### 3.3 Implementasi Aplikasi

Berikut merupakan tampilan implementasi aplikasi yang telah dibangun, dapat dilihat pada tabel 2.

**Tabel 2.** Implementasi aplikasi

Tampilan Halaman	Keterangan
	Halaman <i>splash</i> digunakan untuk halaman loading dari aplikasi. Pengguna akan menunggu selama 4 detik pada halaman ini sebelum masuk pada halaman utama aplikasi.
<p><b>Gambar 8.</b> Halaman <i>Splash</i></p> 	Halaman utama ini digunakan untuk memilih fitur, terdapat 3 fitur dalam halaman ini yaitu deteksi, jenis penyakit dan tentang.
<p><b>Gambar 9.</b> Halaman utama</p> 	Halaman ini digunakan untuk mendeteksi penyakit tanaman sawi, dengan 2 tombol pada menu yakni ambil gambar yaitu bisa pengambilan gambar dengan langsung dan galeri untuk pengambilan gambar pada galeri. Terdapat juga solusi berdasarkan penyakit yang terdeteksi.



Gambar 10. Halaman deteksi

Halaman ini digunakan untuk melihat informasi penyakit tanaman sawi dan cara mengatasinya.



Gambar 11. Halaman jenis penyakit

Halaman ini memiliki fungsi dalam melihat informasi cara penggunaan aplikasi, informasi tentang pakar, dan informasi pembuat aplikasi.



Gambar 12. Halaman tentang

## 4. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan aplikasi sistem pakar untuk mengidentifikasi penyakit pada tanaman sawi secara akurat dan cepat menggunakan algoritma CNN telah berhasil. Algoritma CNN, yang merupakan metode dalam bidang kecerdasan buatan, telah diterapkan secara efektif dalam sistem pakar ini. Aplikasi ini berhasil diimplementasikan pada sistem pakar berbasis android dengan menggunakan bahasa

pemrograman Java dan Python. Hasil dari pengujian usability menunjukkan skor sebesar 80%, yang mengindikasikan bahwa penelitian ini berhasil dilaksanakan sesuai dengan perancangan sistem yang telah dibuat. Implementasi deep learning dalam mengidentifikasi jenis penyakit pada tanaman sawi menggunakan Convolutional Neural Network, khususnya model VGG-16, menghasilkan tingkat akurasi yang sangat tinggi pada F1-Score, yaitu sebesar 96%. Pada pengujian kelas Busuk Daun, terdapat 34 prediksi berhasil dan tidak ada prediksi yang gagal. Pada pengujian kelas Leaf Miner, terdapat 24 prediksi berhasil dan 4 prediksi gagal. Sementara itu, pada pengujian kelas Ulat Grayak, terdapat 28 prediksi berhasil dan tidak ada prediksi yang gagal. Hasil ini menunjukkan bahwa proses identifikasi jenis penyakit pada tanaman sawi berhasil dilakukan dengan sangat baik.

## REFERENCES

- [1] M. D. Sartika, W. D. Andika, and S. Sumarni, "Literature Review : Motivasi yang Diberikan Kepada Anak Dalam Mengonsumsi Sayuran," *J. Pendidik. Anak*, Vol. 11 (1), 2022, Hal. 30-39, vol. 11, no. 1, pp. 30–39, 2022.
- [2] N. S. Damayanti, D. W. Widjanto, and S. Sutarno, "Pertumbuhan dan produksi tanaman sawi Pakcoy (*Brassica rapa* L.) akibat dibudidayakan pada berbagai media tanam dan dosis pupuk organik," *J. Agro Complex*, vol. 3, no. 3, p. 142, 2019, doi: 10.14710/joac.3.3.142-150.
- [3] M. Delfiya and N. Ariska, "Pengaruh Kombinasi Media Tanam Terhadap Pertumbuhan dan Hasil Tanaman Sawi (*Brassica Juncea* L.)," *COMSERVA Indones. J. Community Serv. Dev.*, vol. 1, no. 9, pp. 614–622, 2022, doi: 10.36418/comserva.v1i9.124.
- [4] Yuli Ataribaba, Petrus Selestinus Peten, and Carolina Diana Mual, "Pengaruh Pupuk Hayati terhadap Pertumbuhan Tanaman Sawi (*Brassica juncea* L.) di Kampung Sidomulyo, Distrik Oransbari, Kabupaten Manokawari Selatan, Provinsi Papua Barat," *J. Trit.*, vol. 12, no. 2, pp. 66–78, 2021, doi: 10.47687/jt.v12i2.215.
- [5] H. Sastypratiwi and R. D. Nyoto, "Analisis Data Artikel Sistem Pakar Menggunakan Metode Systematic Review," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 2, p. 250, 2020, doi: 10.26418/jp.v6i2.40914.
- [6] M. Krichen, "Convolutional Neural Networks: A Survey," *Computers*, vol. 12, no. 8, pp. 1–41, 2023, doi: 10.3390/computers12080151.
- [7] A. Permana and K. Budayawan, "Aplikasi Android Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah," *Voteteknika (Vocational Tek. Elektron. dan Inform.)*, vol. 8, no. 4, p. 128, 2020, doi: 10.24036/voteteknika.v8i4.110349.
- [8] M. Liu, H. Deng, and W. Dong, "Identification of Mangrove Invasive Plant *Derris trifoliata* Using UAV Images and Deep Learning Algorithms," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 10017–10026, 2022, doi: 10.1109/JSTARS.2022.3223227.
- [9] X. Wu, Z. Zhao, R. Tian, S. Gao, Y. Niu, and H. Liu, "Exploration of total synchronous fluorescence spectroscopy combined with pre-trained convolutional neural network in the identification and quantification of vegetable oil," *Food Chem.*, vol. 335, no. July 2020, p. 127640, 2021, doi: 10.1016/j.foodchem.2020.127640.
- [10] A. Irfansyah, D., Mustikasari, M., & Suroso, "Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi," *J. Inform. J. Pengemb. IT*, vol. 6, no. 2, pp. 87–92, 2021.
- [11] E. I. Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," *J. ELTIKOM*, vol. 5, no. 2, pp. 81–91, 2021, doi: 10.31961/eltikom.v5i2.428.
- [12] D. Husen, K. Kusri, and K. Kusnawi, "Deteksi Hama Pada Daun Apel Menggunakan Algoritma Convolutional Neural Network," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2103, 2022, doi: 10.30865/mib.v6i4.4667.
- [13] M. Megawaty and N. Huda, "Pembaharuan Sistem Penentuan Untuk Klasifikasi Jenis Penyakit pada RSUD Sekayu Menggunakan Pendekatan Extreme Programming," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 1, p. 66, Jan. 2021, doi: 10.30865/mib.v5i1.2273.
- [14] H. Fonda, "KLASIFIKASI BATIK RIAU DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS (CNN): KLASIFIKASI BATIK RIAU DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS (CNN)," *J. Ilmu Komput.*, vol. 9, no. 1 SE-Articles, pp. 7–10, May 2020, doi: 10.33060/JIK/2020/Vol9.Iss1.144.
- [15] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *Sistemasi*, vol. 10, no. 3, p. 618, 2021, doi: 10.32520/stmsi.v10i3.1248.
- [16] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.
- [17] F. Mahardika, M. Khoiri, and M. Al 'Amin, "Implementasi Extreme Programming pada Sistem Informasi Penggajian untuk Peningkatan Pelayanan kepada Karyawan," *Hello World J. Ilmu Komput.*, vol. 2, no. 2, pp. 74–84, 2023, doi: 10.56211/helloworld.v2i2.274.
- [18] I. Tarsini and R. Anggraeni, "Explore flowchart and pseudocode concepts in algorithms and programming," *Indones. J. Multidiscip. Sci.*, vol. 3, no. 5, 2024, doi: 10.55324/ijoms.v3i5.807.
- [19] R. Soekarta, S. Aras, and Ahmad Nur Aswad, "Hyperparameter Optimization of CNN Classifier for Music Genre Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 5, pp. 1205–1210, 2023, doi: 10.29207/resti.v7i5.5319.
- [20] P. S. Uttarwar, R. P. Tidke, D. S. Dandwate, and U. J. Tupe, "A Literature Review on Android -A Mobile Operating System," *Int. Res. J. Eng. Technol.*, vol. 8, no. 1, pp. 1–6, 2021.