

**KLASIFIKASI HAMA DARI CITRA DAUN BROKOLI MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK EFFICIENTNET-B0***

**SKRIPSI**

**A. RAIHAN MAULANA GS  
201401108**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**KLASIFIKASI HAMA DARI CITRA DAUN BROKOLI MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK EFFICIENTNET-B0***

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer**

**A. RAIHAN MAULANA GS  
201401108**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**PERSETUJUAN**

Judul : KLASIFIKASI HAMA DARI CITRA  
DAUN BROKOLI MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK*  
*EFFICIENTNET-B0*

Kategori : SKRIPSI

Nama : A. RAIHAN MAULANA GS

Nomor Induk Mahasiswa : 201401108

Program Studi : SARJANA (S1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERSITAS SUMATERA  
UTARA

Telah diuji dan dinyatakan lulus di Medan, 08 Juli 2024

Komisi Pembimbing:

Pembimbing II

Pembimbing I

Hayatunnufus, S.Kom, M.Cs.

NIP. 199207192020012001

Dr. Eng Ade Candra, ST., M.Kom.

NIP. 197909042009121002

Diketahui/disetujui oleh

Program Studi S1 Ilmu Komputer

Ketua,

Dr. Amalia ST., M.T

NIP. 197812212014042001

**UNIVERSITAS SUMATERA UTARA**

**PERNYATAAN****KLASIFIKASI HAMA DARI CITRA DAUN BROKOLI MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK EFFICIENTNET-B0*****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 06 Juni 2024



A. Raihan Maulana GS

201401108

## PENGHARGAAN

Syukur Alhamdulillah Rabbil 'alamin, dengan menyebut nama Allah *Subhanahu Wa Ta'ala* yang Maha Pengasih lagi Maha Penyayang, sehingga penulis dapat menyelesaikan skripsi berjudul “Klasifikasi Hama dari Citra Daun Brokoli Menggunakan Convolutional Neural Network *Efficientnet-B0*”.

Skripsi ini merupakan hasil dari kontribusi dan dukungan dari berbagai individu untuk itu saya mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi di Universitas Sumatera Utara.
3. Ibu Dr. Amalia, S.T., M.T. sebagai Ketua Program Studi S-1 Ilmu Komputer yang juga membimbing penulis sebagai Dosen Penguji I dengan berbagai masukan dan kritik yang konstruktif selama penyusunan skripsi ini.
4. Ibu Dian Rachmawati S.Si., M.Kom. sebagai Dosen Pembimbing Akademik yang telah memberikan bimbingan, nasihat, dan dorongan kepada penulis dalam melalui tahap-tahap akademik dari masa kuliah hingga penulisan skripsi.
5. Bapak Dr. Eng Ade Candra, S.T., M.Kom. sebagai Dosen Pembimbing I yang telah memberikan bimbingan dan masukan yang sangat berharga selama penulisan skripsi.
6. Ibu Hayatunnufus, S.Kom., M.Cs. sebagai Dosen Pembimbing II yang memberikan nasihat dan kritik yang membantu dalam penyusunan skripsi ini.
7. Ibu Desilia Selvida S.Kom., M.Kom. sebagai Dosen Penguji II yang telah memberikan saran dan kritik yang konstruktif dalam penyelesaian skripsi ini.
8. Seluruh dosen Program Studi S-1 Ilmu Komputer yang telah meluangkan waktu dan energi untuk mengajar dan membimbing.
9. Teristimewa Saya ingin menyampaikan penghargaan kepada kedua orang

tua saya, Ibu Suriati Sembiring S.E. dan Bapak M. Maulana Gurusinga yang selalu memberikan dukungan, doa, serta menjadi pendorong utama bagi saya.

10. Orang terdekat, sahabat, serta teman penulis yaitu atas nama Amru, Habib, Alwan, Arifandi, Uga, Kevin, Zahra, Chindy, Sonia, dan teman-teman lainnya yang belum dapat disebutkan oleh penulis.
11. Teman-teman mahasiswa/i program studi S-1 Ilmu Komputer Stambuk 2020, khususnya KOM – C.
12. Dan semua pihak yang terlibat, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu.
13. Terakhir kepada saya sendiri, A. Raihan Maulana GS. Terima kasih sudah bertahan sampai di titik ini. Sesulit apapun proses penyusunan skripsi ini kamu telah menyelesaikan sebaik dan semaksimal mungkin. Apapun kurang dan lebihmu mari merayakan diri sendiri.

Medan, 06 Juni 2024



A.Raihan Maulana GS

201401108

## ABSTRAK

Brokoli (*Brassica oleracea L.*) merupakan sayuran yang tergolong dalam keluarga kubis-kubisan. Sayuran ini terkenal karena kandungan nutrisi dan gizinya yang melimpah, termasuk vitamin C, vitamin K, zat besi, dan senyawa antioksidan yang berguna bagi kesehatan. Setiap tahun banyak brokoli yang gagal panen dikarenakan serangan hama dan cara utama untuk menanganinya adalah dengan menyemprotkan pestisida. Dalam mengendalikan hama sebagian besar petani menyemprotkan pestisida tanpa memperhatikan dosis, waktu, metode, dan target yang tepat. Tindakan tersebut menyebabkan dampak negatif seperti membunuh organisme yang bukan merupakan target hama. Untuk mengatasi permasalahan tersebut, penelitian ini memanfaatkan metode CNN untuk melakukan klasifikasi terhadap hama yang menyerang tanaman brokoli. Dataset yang dimanfaatkan pada penelitian ini berupa gambar yang telah diambil secara langsung dari lahan pertanian brokoli di daerah Berastagi. Dari hasil pengujian menunjukkan bahwa model CNN menghasilkan tingkat akurasi tertinggi yaitu *training accuracy* sebesar 95,69 % dengan *training loss* 0.12 dan *validation accuracy* sebesar 98,96 % dengan *validation loss* 0.06 dengan jumlah 75 *epoch* pada pelatihan model. Pada evaluasi model menggunakan *confusion matrix* mendapatkan nilai akurasi 96,56 %, nilai *precision* 96,55 %, nilai *recall* 96,61 %, dan 96,57% *F1-score* dalam klasifikasi hama. Penelitian ini memberikan kontribusi dalam pengembangan teknologi untuk mendukung petani brokoli dalam mengatasi masalah serangan hama yang dapat mengancam hasil panen dan untuk memudahkan pengguna, sistem ini dirancang untuk perangkat Android secara langsung.

**Kata kunci:** Brokoli , Hama, Klasifikasi, *Convolutional Neural Network*

## ABSTRACT

Broccoli (*Brassica oleracea* L.) is a vegetable that belongs to the cabbage family. This vegetable is well known for its abundance of nutrients and nutrients, including vitamin C, vitamin K, iron, and antioxidant compounds that are useful for health. Every year many broccoli crops fail due to pests and the main way to deal with them is by spraying pesticides. In controlling pests, most farmers spray pesticides without paying attention to the right dose, time, method, and target. This action causes negative impacts such as killing organisms that are not the target of pests. To overcome these problems, this research utilizes the CNN method to classify pests that attack broccoli plants. The dataset utilized in this research is in the form of images that have been taken directly from broccoli farms in the Berastagi area. The test results show that the CNN model produces the highest level of accuracy, namely training accuracy of 95.69% with a training loss of 0.12 and validation accuracy of 98.96% with a validation loss of 0.06 with a total of 75 epochs in model training. In evaluating the model using confusion matrix, the accuracy value is 96.56%, precision value is 96.55%, recall value is 96.61%, and 96.57% F1-score in pest classification. This research contributes to the development of technology to support broccoli farmers in overcoming the problem of pest attacks that can threaten crop yields and to facilitate users, this system is designed for Android devices directly.

**Keyword:** Broccoli, Pest, Classification, Convolutional Neural Network



## DAFTAR ISI

PERSETUJUAN .....	ii
PERNYATAAN.....	iii
PENGHARGAAN .....	iv
ABSTRAK .....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB I    PENDAHULUAN.....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan Penelitian.....	3
1.5    Manfaat Penelitian.....	3
1.6    Metodologi Penelitian .....	3
1.7    Penelitian Relevan .....	4
1.8    Sistematika Penulisan.....	6
BAB II LANDASAN TEORI .....	8
2.1 <i>Artificial Intelligence</i> .....	8
2.2 <i>Machine Learning</i> .....	8
2.3 <i>Deep Learning</i> .....	8
2.4 <i>Convolutional Neural Network (CNN)</i> .....	8
2.4.1 Lapisan CNN.....	9
2.5 <i>Batch Normalization</i> .....	10
2.6 <i>Dropout Regularization</i> .....	10
2.7 <i>EfficientNet-B0</i> .....	11
2.8 <i>Confusion Matrix</i> .....	12
2.9    Tensorflow.....	13
2.10    Tensorflow Lite.....	14

2.11	Android Studio.....	14
2.12	Kutu Daun ( <i>Thrips</i> ).....	14
2.13	Ulat Daun.....	15
2.14	Bercak Daun <i>Altenaria</i> .....	15
2.15	Sehat.....	16
BAB III ANALISIS DAN PERANCANGAN .....		17
3.1	Analisis Sistem .....	17
3.1.1	Analisis Masalah .....	17
3.1.2	Analisis Kebutuhan .....	17
3.1.3	Arsitektur Sistem.....	18
3.1.4	Pengumpulan Data .....	19
3.1.5	Pre-Processing Data .....	20
3.1.6	Arsitektur <i>EfficientNet-B0</i> .....	21
3.2	Pemodelan Sistem .....	24
3.2.1	<i>Use Case Diagram</i> .....	24
3.2.2	<i>Activity Diagram</i> .....	25
3.3	<i>Flowchart</i> .....	25
3.3.1	<i>Flowchart</i> Pelatihan Model CNN <i>EfficientNet-B0</i> .....	26
3.3.2	<i>Flowchart</i> Sistem .....	27
3.4	Perancangan <i>Interface</i> .....	27
3.4.1	Halaman Tampilan Sistem .....	28
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....		29
4.1	Implementasi Sistem .....	29
4.1.1	Perangkat Keras ( <i>Hardware</i> ) .....	29
4.1.2	Perangkat Lunak ( <i>Software</i> ).....	29
4.1.3	Halaman Utama.....	30
4.1.4	Halaman Hasil.....	30
4.2	Pelatihan Model.....	31
4.2.1	Import Library Tensorflow .....	31
4.2.2	Proses Mengakses Dataset .....	32
4.2.3	Memuat Model <i>EfficientNet-B0</i> .....	32
4.2.4	Proses Pelatihan .....	33
4.2.5	Evaluasi Model.....	34

4.4	Pengujian Sistem .....	38
BAB 5 KESIMPULAN DAN SARAN .....		43
5.1	Kesimpulan.....	43
5.2	Saran .....	43
DAFTAR PUSTAKA .....		45

## DAFTAR TABEL

<b>Tabel 2.1</b> Perbandingan Arsitektur EfficientNet .....	11
<b>Tabel 2.2</b> Confusion Matrix .....	12
<b>Tabel 3.1</b> Jumlah Data per Kelas.....	19
<b>Tabel 4.1</b> Spesifikasi Perangkat Keras .....	29
<b>Tabel 4.2</b> Spesifikasi Perangkat Lunak. ....	29
<b>Tabel 4.3</b> Percobaan dengan beberapa Jumlah Epoch.....	33
<b>Tabel 4.4</b> Pengujian Sistem .....	38
<b>Tabel 4.5</b> Pengujian Berdasarkan Mayoritas Hama yang Ada Pada Daun.....	40

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Arsitektur CNN .....	9
<b>Gambar 2.2</b> Fully Connected.....	10
<b>Gambar 2.3</b> Contoh daun brokoli yang terkena Thrips .....	15
<b>Gambar 2.4</b> Contoh daun brokoli yang terkena Ulat Daun .....	15
<b>Gambar 2.5</b> Contoh daun yang terkena Bercak Daun Alternaria .....	16
<b>Gambar 2.6</b> Contoh daun brokoli yang Sehat.....	16
<b>Gambar 3.1</b> Arsitektur Sistem .....	19
<b>Gambar 3.2</b> Gambar di resize menjadi 224 x 224 .....	21
<b>Gambar 3.3</b> Arsitektur EfficientNet-B0 .....	21
<b>Gambar 3.4</b> Use Case Diagram .....	24
<b>Gambar 3.5</b> Activity Diagram .....	25
<b>Gambar 3.6</b> Flowchart pelatihan model CNN .....	26
<b>Gambar 3.7</b> Flowchart Sistem .....	27
<b>Gambar 3.8</b> Desain Halaman Tampilan Sistem.....	28
<b>Gambar 4.1</b> Tampilan Halaman Utama .....	30
<b>Gambar 4.2</b> Tampilan Halaman Hasil .....	31
<b>Gambar 4.3</b> Kode untuk Import Tensorflow .....	31
<b>Gambar 4.4</b> kode untuk mengakses dan menampilkan dataset .....	32
<b>Gambar 4.5</b> Kode Untuk Memuat Model EfficientNet-B0 .....	32
<b>Gambar 4.6</b> Proses Pelatihan .....	33
<b>Gambar 4.7</b> Grafik Akurasi dan Loss Model saat proses pelatihan .....	34
<b>Gambar 4.8</b> Confusion Matrix.....	35

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Brokoli (*Brassica oleracea L.*) merupakan sayuran yang tergolong dalam keluarga kubis-kubisan. Sayuran ini terkenal karena kandungan nutrisi dan gizinya yang melimpah, termasuk vitamin C, vitamin K, zat besi, dan senyawa antioksidan yang berguna bagi kesehatan (Widiwurjani et al., 2019). Setiap tahun banyak brokoli yang gagal panen dikarenakan serangan hama dan cara utama untuk menanganinya adalah dengan menyemprotkan pestisida (Zou et al., 2021). Dampak utama serangan hama adalah terjadinya penyakit yang menyebabkan berkurangnya fungsi daun brokoli.

Daun brokoli berfungsi untuk fotosintesis, penyimpanan energi, penguatan struktur tanaman, regulasi air, suhu, produksi hormon, perlindungan terhadap hama penyakit, serta interaksi ekologi. Ini menunjukkan peran penting daun dalam pertumbuhan brokoli secara menyeluruh. Beberapa hama yang sering menyerang daun brokoli adalah ulat daun kubis (*Plutella xylostella Linn*), kutu daun (*Aphis brassicae L.*), dan bercak daun alternaria (*Alternaria brassicae*) (Septariani et al., 2020).

Dalam mengendalikan hama sebagian besar petani menyemprotkan pestisida tanpa memperhatikan dosis, waktu, metode, dan target yang tepat. Tindakan tersebut menyebabkan dampak negatif seperti membunuh organisme yang bukan merupakan target hama (Zou et al., 2021).

Oleh karena itu dibutuhkan suatu sistem klasifikasi jenis hama pada daun brokoli agar petani dapat lebih efektif menangani hama tersebut dengan mempertimbangkan jenisnya. Klasifikasi sendiri merupakan suatu proses pembelajaran terarah (*supervised learning*) dalam Machine Learning terhadap sebuah kumpulan data latih sehingga model dapat digunakan untuk data pengujian (Azhari et al., 2021).

Penelitian ini menggunakan metode CNN untuk mengembangkan suatu model yang mampu mengklasifikasikan data citra. CNN adalah salah satu teknik jaringan saraf tiruan yang terkenal dan banyak diminati. CNN digunakan untuk

memproses data yang tersusun dalam bentuk grid, seperti citra dua dimensi, serta untuk mengolah data dengan dimensi tinggi seperti video (Jinan & Hayadi, 2022).

Terdapat beberapa arsitektur pada CNN, salah satunya adalah *EfficientNet-B0* yang dikembangkan untuk meningkatkan kinerja model CNN dalam pelatihan data. Peneliti menemukan bahwa menyeimbangkan kedalaman, keluasan, dan resolusi model penting untuk meningkatkan kinerja berbeda dengan pendekatan sebelumnya yang hanya memperbesar satu aspek dari model. *EfficientNet* menggunakan skala faktor konstan untuk memperbesar ketiga aspek tersebut secara proporsional dan menghasilkan model dengan parameter lebih sedikit dan akurasi yang tinggi (Tan & Le, 2019).

Berdasarkan uraian di awal maka penelitian ini akan menggunakan metode CNN *Efficientnet-B0* untuk mengklasifikasi hama dari citra daun brokoli. Dataset yang akan dipakai pada penelitian ini dikumpulkan langsung oleh peneliti.

## **1.2 Rumusan Masalah**

Petani brokoli kesulitan dalam membedakan serangan hama yang menyebabkan brokoli menjadi rusak/gagal panen. Oleh karena itu dibutuhkan suatu sistem yang mampu mengklasifikasikan hama dari citra daun brokoli guna mempermudah petani dalam membedakan hama yang menyerang daun brokoli seperti ulat daun kubis, kutu daun, dan bercak daun *Alternaria*.

## **1.3 Batasan Masalah**

Ada beberapa aspek yang membatasi penelitian ini.

1. Jumlah kelas diidentifikasi oleh model ada 4 yaitu ulat daun kubis, kutu daun, bercak daun *Alternaria*, dan sehat. Dengan membatasi jumlah kelas diharapkan dapat membantu meningkatkan akurasi dan Interpretabilitas model.
2. Penelitian hanya menghasilkan output klasifikasi hama pada brokoli dan penanganan nya.
3. Data yang digunakan berupa gambar yang diambil langsung di perkebunan brokoli wilayah Berastagi.
4. Data yang diambil pada penelitian menggunakan kamera *Smartphone* dengan resolusi 64 MP dengan format JPG
5. Pengambilan data dilakukan saat brokoli usia 1 bulan / 40% siap panen.

6. Jumlah data yang dipakai berjumlah 2.120 data.

#### **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah merancang sebuah sistem yang dapat mengklasifikasikan hama yang menyerang daun brokoli serta memberikan dukungan kepada petani dalam proses pemilihan dan penentuan pestisida yang tepat untuk mengatasi hama dengan efektif.

#### **1.5 Manfaat Penelitian**

Diharapkan bahwa penelitian ini akan memberikan manfaat sebagai berikut:

Bagi Peneliti:

1. Penelitian menghasilkan sebuah model yang mampu mengklasifikasikan jenis hama yang menyerang daun brokoli.
2. Penelitian memberikan informasi dan menjadi bahan referensi mengenai penerapan *Machine Learning* dengan pendekatan CNN arsitektur *EfficientNet-B0*.

Bagi Petani Brokoli:

1. Penelitian membantu petani mengidentifikasi hama pada daun brokoli dengan cepat dan akurat memungkinkan mereka segera mengambil langkah-langkah pencegahan yang efektif.
2. Penelitian membantu petani mengurangi biaya yang terkait dengan penggantian tanaman yang rusak atau penggunaan pestisida yang berlebihan sehingga dapat meningkatkan efisiensi biaya dalam produksi brokoli.

#### **1.6 Metodologi Penelitian**

Berikut beberapa metode penelitian yang akan dilakukan, yaitu:

1. Studi Pustaka

Peneliti membaca, mencari, dan mengumpulkan referensi dari berbagai sumber yang dapat dipercaya melalui beberapa media seperti buku, jurnal, ebook, atau situs web yang relevan dengan CNN *EfficientNet-B0*.

2. Pengumpulan Data Set

Mengumpulkan kumpulan citra yang mencakup berbagai jenis hama. Data diperoleh pengambilan citra langsung.



### 3. Analisis dan Perancangan

Penulis menganalisis semua komponen yang diperlukan untuk penelitian yang akan dikembangkan dengan mengacu pada perumusan dan ruang lingkup masalah dalam penelitian ini.

### 4. Implementasi

Pada bagian ini, *Android Studio* akan digunakan untuk mengembangkan sistem sesuai dengan diagram yang ditentukan.

### 5. Pengujian Sistem

Pengujian akan dilakukan pada sistem yang dikembangkan untuk memeriksa fungsionalitas perangkat lunak dan melihat seberapa baik pendekatan *Convolutional Neural Network* terhadap data yang diklasifikasi.

### 6. Dokumentasi

Penelitian menyimpan proses dokumentasi mulai dari tahap analisis hingga tahap pengujian sebagai kelengkapan penelitian agar lebih optimal.

## 1.7 Penelitian Relevan

Berikut adalah penelitian-penelitian terdahulu yang relevan, yaitu:

1. Penelitian berjudul "*Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network*" mengusulkan metode berbasis pembelajaran mendalam untuk mengidentifikasi dan mengklasifikasikan penyakit daun tanaman. Pendekatan ini memanfaatkan jaringan saraf konvolusional yang ditingkatkan dengan menggabungkan modul inception, modul squeeze-and-excitation (SE), dan lapisan pooling global. Tujuannya adalah untuk mengatasi masalah waktu konvergensi pelatihan yang lama dan parameter model yang terlalu besar. Struktur inception menggabungkan data fitur pada berbagai skala untuk meningkatkan akurasi pada dataset penyakit daun. Sementara itu, lapisan pooling global menggantikan lapisan fully connected untuk mengurangi jumlah parameter model. Eksperimen menunjukkan bahwa model ini mencapai akurasi 91,7% pada dataset uji, mengungguli beberapa jaringan saraf konvolusional tradisional baik dalam akurasi maupun efisiensi parameter dan waktu pelatihan. Studi ini menegaskan bahwa metode yang diusulkan efektif dan dapat diandalkan dalam mengklasifikasikan penyakit daun tanaman (Hang et al., 2019).

2. Penelitian berjudul *“Detection of the Affected Area and Classification of Pests Using Convolutional Neural Networks From the Leaf Images”* membahas penggunaan jaringan saraf konvolusi (CNN) untuk deteksi dan klasifikasi hama pada tanaman sayuran di Distrik Batticaloa, Sri Lanka. Sistem yang diusulkan mendeteksi area yang terkena hama dan mengklasifikasikan jenis hama berdasarkan citra daun yang terpengaruh. Data citra dikumpulkan dari kamera dan sumber daring, kemudian diproses untuk mengidentifikasi hama. Hasil pengujian menunjukkan tingkat klasifikasi yang bervariasi, dengan akurasi 76,65% untuk kumbang kutu dan 65,17% untuk kumbang tanaman tarnish (Suthakaran & Premaratne, 2020).
3. Penelitian berjudul *“Classification of Olive Leaf Diseases Using Deep Convolutional Neural Networks”* mengkaji penggunaan jaringan saraf konvolusional dalam mengidentifikasi penyakit daun zaitun, yakni *Aculus olearius* dan *peacock spot*. Penelitian ini menggunakan dataset 3.400 gambar daun zaitun yang mencakup daun sehat dan berpenyakit dari Turki. Model CNN yang diusulkan dibandingkan dengan model VGG16 dan VGG19 menggunakan metode pembelajaran transfer, dan kinerja algoritma optimisasi seperti Adam, AdaGrad, SGD, dan RMSProp diuji. Eksperimen menunjukkan bahwa augmentasi data meningkatkan akurasi klasifikasi dari 88% menjadi 95%. Studi ini menekankan pentingnya pengenalan dini penyakit pada daun zaitun untuk meningkatkan produktivitas dan menyediakan aplikasi berbasis web sebagai solusi praktis bagi petani (Uğuz & Uysal, 2021).
4. Penelitian berjudul *“Classifying Barako Coffee Leaf Diseases Using Deep Convolutional Models”* menganalisis 4667 citra daun kopi yang terdiri dari empat kelas, yaitu Coffee Leaf Rust, *Cercospora* Leaf Spots, Sooty Molds, dan Healthy Leaves. Hasil pengujian menunjukkan bahwa tingkat akurasi tertinggi mencapai 97,20% menggunakan arsitektur VGG16, sementara tingkat akurasi terendah adalah 90,83% dengan arsitektur ResNetV2-152 (Montalbo & Hernandez, 2020).
5. Penelitian berjudul *“Plant disease classification using deep learning”* mengklasifikasikan hama tanaman anggur menggunakan 4.062 citra dengan

empat kelas yang terdiri dari healthy, black rot, Esca (Black Measles), dan Leaf blight (Isariopsis Leaf Spot). Parameter yang digunakan dalam studi ini adalah ukuran gambar 224 x 224 piksel, augmentasi data seperti rotasi, zoom, dan shift. Hasil pengujian menunjukkan akurasi tertinggi mencapai 98,31% menggunakan arsitektur DenseNet dan tingkat akurasi terendah adalah 95,54% dengan arsitektur VGG (Akshai & Anitha, 2021).

6. Penelitian berjudul "*Klasifikasi Penyakit pada Daun Jagung Menggunakan Metode CNN AlexNet*" Dataset yang dipakai terdiri dari 4198 gambar daun jagung yang mencakup empat kelas yaitu Blight, Healthy, Gray Leaf Spot, dan Common Rust. Dataset dibagi menjadi data pelatihan, data pengujian, dan data validasi berdasarkan kategori. Hasil menunjukkan bahwa sistem dapat mengklasifikasikan penyakit pada daun jagung dengan akurasi mencapai 90% menggunakan arsitektur AlexNet (Azizah, 2023).
7. Penelitian berjudul "*Klasifikasi Penyakit pada Daun Padi Menggunakan Metode CNN dengan Multilayer Perceptron*" peneliti memakai tiga jenis penyakit daun padi yakni Bacterial Leaf Blight, Brown Spot, dan Leaf Spot. Hasil penelitian diperoleh dengan menggunakan arsitektur CNN dengan Multilayer Perceptron. Parameter yang digunakan untuk mencapai hasil tersebut adalah ukuran gambar 100 x 100 piksel, kernel 3 x 3, learning rate 0,01, Adam optimizer, 150 epoch, batch size 30, dan perbandingan dataset 90 banding 10 untuk gambar berwarna RGB. Dengan menggunakan parameter tersebut, penelitian ini mampu mencapai akurasi sebesar 91,7% pada data pengujian (Jinan & Hayadi, 2022).
8. Penelitian dengan judul "*Mendeteksi Penyakit pada Daun Kentang Menggunakan Metode CNN*" menggunakan 922 citra daun kentang yang mencakup 3 kelas yaitu Later Bright, Early Bright, dan Healty. Percobaan menunjukkan tingkat akurasi sebesar 95% pada uji dan 94% pada validasi (Rozaqi et al., 2021).

## **1.8 Sistematika Penulisan**

Struktur skripsi ini terdiri dari lima bab yang mencakup:

**BAB I            PENDAHULUAN**

Bab ini memuat uraian mengenai latar belakang penelitian, perumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan susunan penulisan.

**BAB II           LANDASAN TEORI**

Bab ini menguraikan teori yang relevan terkait dengan penerapan klasifikasi hama dari citra daun brokoli menggunakan CNN EfficientNet-B0.

**BAB III          ANALISIS DAN PERANCANGAN**

Merupakan tahap menganalisis syarat yang diperlukan untuk pembangunan sistem dan metode yang akan digunakan untuk merancangnya.

**BAB IV          IMPLEMENTASI DAN PENGUJIAN**

Bab ini menjelaskan penerapan dan pengujian aplikasi sistem sesuai dengan rancangan yang telah dibuat.

**BAB V           KESIMPULAN DAN SARAN**

Bab ini memaparkan ringkasan hasil pengujian dari penelitian yang dilakukan dan saran untuk penelitian mendatang.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1    *Artificial Intelligence***

*Artificial intelligence* merupakan kecerdasan yang diperkenalkan kepada sistem yang dikontrol dalam lingkungan ilmiah. Kecerdasan buatan dapat diperoleh dari rangkaian pembelajaran pada sistem dengan menggunakan data yang telah di input dalam sistem. Dari pembelajaran tersebut akan mencapai tujuan untuk memahami data eksternal (Boulent et al., 2019).

#### **2.2    *Machine Learning***

*Machine Learning* adalah salah satu bidang dalam kecerdasan buatan yang memungkinkan komputer untuk belajar dari data tanpa memerlukan pemrograman eksplisit. Ini memungkinkan komputer untuk mengidentifikasi pola dalam data, membuat prediksi, dan mengambil keputusan tanpa intervensi manusia (Zou et al., 2021).

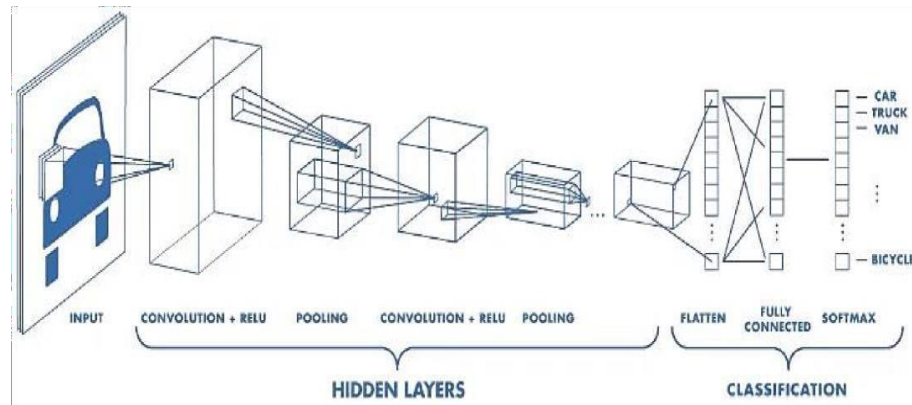
#### **2.3    *Deep Learning***

*Deep learning* adalah bagian dari pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan yang berguna untuk mengolah dan memahami fitur-fitur kompleks dari data dalam jumlah banyak. Teknik ini meniru cara kerja otak manusia dalam pemrosesan informasi melalui lapisan-lapisan yang saling berhubungan, yang memungkinkan komputer mengenali pola yang rumit dan tersembunyi dalam data (Sarker, 2021).

#### **2.4    *Convolutional Neural Network (CNN)***

CNN adalah komponen dari struktur jaringan saraf yang umumnya terkait dengan analisis gambar. CNN memiliki tiga komponen utama yaitu lapisan konvolusi, lapisan penggabungan (pooling), dan lapisan yang terhubung sepenuhnya. Lapisan konvolusi berfungsi untuk mengekstraksi fitur-fitur penting dari gambar dengan menerapkan operasi konvolusi pada gambar input. Lalu lapisan penggabungan membantu mengurangi ukuran spasial dari fitur yang dihasilkan lapisan konvolusi. Kemudian lapisan yang terhubung sepenuhnya bertugas untuk menyatukan fitur-fitur tersebut dan membuat prediksi akhir berdasarkan fitur-fitur yang telah dipelajari. CNN telah menjadi metode yang sangat efisien dalam berbagai jenis

tugas pemrosesan seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar (Boulent et al., 2019).



**Gambar 2.1** Arsitektur CNN (Ahmed et al., 2023)

### 2.4.1 Lapisan CNN

#### a. *Convolution Layer*

Konvolusi adalah operasi matematis yang menggunakan fungsi berulang kepada sebuah output. Terdapat beberapa parameter yang dapat digunakan pada lapisan ini, yaitu kernel, stride, padding dan lainnya. Kernel memiliki nama lain filter yang berguna untuk mengekstraksi fitur yang ada pada citra (Gajjar et al., 2022).

#### b. *Pooling Layer*

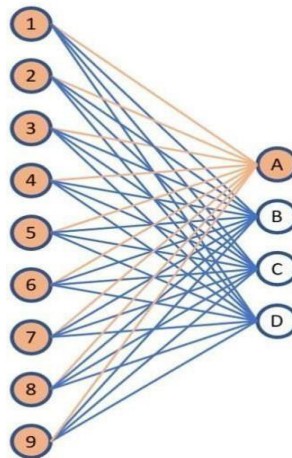
*Pooling* layer memiliki tujuan mengurangi kompleksitas layer berikutnya, atau dapat juga disebut mengurangi resolusi. Ukuran pooling layer yang biasa digunakan adalah 2x2. Beberapa metode pooling yang yang biasa dipakai dalam deep learning adalah *Max Pooling*, *Average Pooling*, *Sum Pooling* dan sebagainya (Jinan & Hayadi, 2022). Jenis pooling layer yang digunakan pada penelitian ini adalah *Max Pooling*, di mana metode tersebut akan mencari nilai maksimum dari bagian yang sudah di partisi dengan ukuran 2x2.

#### c. *ReLU (Rectified Linear Unit)*

Rectified Linear Unit digunakan untuk menyesuaikan hasil dari konvolusi yang akan dipersiapkan ke layer berikutnya dengan tujuan mengubah nilai negatif dari hasil konvolusi menjadi nol.

d. *Fully Connected Layer*

Setiap neuron dalam lapisan *fully connected* memiliki koneksi ke setiap aktivasi dalam lapisan di atasnya. Data diubah menjadi vektor satu dimensi menggunakan metode konvolusi dan pooling kemudian menjadi input bagi jaringan *fully connected*. Struktur tersebut mengandung satu atau lebih lapisan tersembunyi (Sarigül et al., 2019).



**Gambar 2.2** *Fully Connected*

## 2.5 *Batch Normalization*

*Batch normalization* adalah teknik yang sering digunakan untuk mengurangi *overfitting* pada model. Teknik ini melibatkan normalisasi setiap lapisan dalam model, memungkinkan pembelajaran yang lebih efektif dengan bobot yang sudah dinormalisasi. Ketika model melatih data, distribusi input pada setiap lapisan disesuaikan berdasarkan parameter dari lapisan sebelumnya. Meskipun ini dapat menyebabkan penurunan laju pembelajaran dan memerlukan inisialisasi parameter yang cermat, metode ini membantu mengurangi *overfitting* dan dapat mempercepat waktu pemrosesan (Falaschetti et al., 2022).

## 2.6 *Dropout Regularization*

*Dropout* adalah teknik regularisasi yang bertujuan untuk mencegah *overfitting* pada model. *Overfitting* terjadi ketika model terlalu sesuai atau terlalu kompleks sehingga mampu mempelajari detail - detail yang tidak relevan dari data pelatihan, yang mengakibatkan kinerja yang buruk pada data uji yang baru. Masalah *overfitting* dapat diidentifikasi dengan melihat perbedaan antara akurasi data

pelatihan dan data pengujian. Jika model menunjukkan performa yang lebih baik pada data pelatihan dibandingkan data pengujian, maka model tersebut dianggap mengalami overfitting (Tobias et al., 2020).

## 2.7 *EfficientNet-B0*

*EfficientNet-B0* merupakan sebuah arsitektur CNN yang dikembangkan untuk meningkatkan kinerja model CNN dalam pelatihan data. Peneliti menemukan bahwa keseimbangan antara kedalaman, keluasan, dan resolusi model sangat penting untuk meningkatkan kinerja, berbeda dengan pendekatan sebelumnya yang hanya memperbesar satu aspek dari model. *EfficientNet* memperbesar ketiga aspek tersebut secara proporsional dengan menggunakan skala faktor konstan, menghasilkan model dengan jumlah parameter yang lebih sedikit tetapi tetap memiliki akurasi yang tinggi (Tan & Le, 2019).

**Tabel 2.1** Perbandingan Arsitektur *EfficientNet* (Tan & Le, 2019)

Model	Top-1 Acc.	Top-5 Acc.	#Params
<b>EfficientNet-B0</b>	<b>77.3 %</b>	<b>93.5 %</b>	<b>5.3 M</b>
ResNet-50	76.0 %	93.0 %	26 M
DenseNet-169	76.2 %	76.2 %	14 M
<b>EfficientNet-B1</b>	<b>79.2 %</b>	<b>94.5 %</b>	<b>7.8 M</b>
ResNet-152	77.8 %	93.8 %	60 M
DenseNet-264	77.9 %	93.9 %	34 M
Inception-v3	78.8 %	94.4 %	24 M
Xception	79.0 %	94.5 %	23 M
<b>EfficientNet-B2</b>	<b>80.3 %</b>	<b>95.0 %</b>	<b>9.2 M</b>
Inception-v4	80.0 %	95.0 %	48 M
Inception-resnet-v2	80.1 %	95.1 %	56 M
<b>EfficientNet-B3</b>	<b>81.7 %</b>	<b>95.6 %</b>	<b>12 M</b>
ResNeXt-101	80.9 %	95.6 %	84 M
PolyNet	81.3 %	95.8 %	92 M
<b>EfficientNet-B4</b>	<b>83.0 %</b>	<b>96.3 %</b>	<b>19 M</b>
SENet	82.7 %	96.2 %	146 M
NASNet-A	82.7 %	96.2 %	89 M
AmoebaNet-A	82.8 %	96.1 %	87 M
PNASNet	82.9 %	96.2 %	86 M



<b>EfficientNet-B5</b>	<b>83.7 %</b>	<b>96.7 %</b>	<b>30 M</b>
AmoebaNet-C	83.5 %	96.5 %	155 M
<b>EfficientNet-B6</b>	<b>84.2 %</b>	<b>96.8 %</b>	<b>43 M</b>
<b>EfficientNet-B7</b>	<b>84.4 %</b>	<b>97.1 %</b>	<b>66 M</b>
Gpipe	84.3 %	97.0 %	557 M

*EfficientNet* telah mengungguli semua arsitektur CNN pada periode sebelumnya. Salah satu contohnya adalah EfficientNet-B7, yang berhasil mencapai tingkat akurasi tertinggi dengan akurasi sebesar 84,3%, yang melebihi performa beberapa arsitektur CNN lainnya (Tan & Le, 2019).

## 2.8 Confusion Matrix

*Confusion matrix* adalah suatu tabel yang dipakai untuk menilai efektivitas model klasifikasi. Tabel ini menunjukkan perbandingan hasil klasifikasi aktual dengan hasil prediksi yang dihasilkan oleh model. Saat menggunakan confusion matrix untuk mengukur kinerja, ada empat komponen utama: *True Negative* yang menunjukkan jumlah data negatif yang diidentifikasi dengan benar. Sebaliknya, *False Positive* mengacu pada data negatif yang salah diidentifikasi sebagai data positif. *True Positive* mengindikasikan data positif yang terdeteksi dengan benar. Terakhir, *False Negative* adalah kebalikan dari *True Positive*, yaitu data positif yang salah diidentifikasi sebagai data negatif (Hadianto et al., 2019).

**Tabel 2.2** *Confusion Matrix*

Aktual	Klasifikasi	
	+	-
+	<i>True Positif (TP)</i>	<i>False Negative (FN)</i>
-	<i>False Positif (FP)</i>	<i>True Negative (TN)</i>

Berdasarkan hasil dari confusion matrix, kita dapat mengevaluasi kinerja dan menentukan kualitas sistem yang dibuat. Parameter yang dianalisis dalam penelitian ini yaitu akurasi, presisi, *recall*, dan *F1-Score*. Rumus akurasi dihitung memakai persamaan berikut ini:

$$\text{Akurasi} = \frac{TP}{\text{Total Data}}$$

Presisi (*Precision*) adalah perbandingan antara item relevan yang terpilih dengan total item yang terpilih. Ini mengukur sejauh mana jawaban yang diberikan sesuai dengan permintaan informasi. Nilai *Precision* dihitung dengan persamaan berikut ini:

$$\text{Precision} = \frac{TP}{TP + FP}$$

*Recall* adalah rasio item yang relevan yang berhasil terpilih dibandingkan dengan total jumlah item yang relevan yang ada. Ini mengukur sejauh mana semua item relevan berhasil diidentifikasi. Rumus *recall* dihitung dengan persamaan berikut ini:

$$\text{Recall} = \frac{TP}{TP + FN}$$

*F1-Score* adalah pengukuran yang menggabungkan presisi dan *recall* secara rata-rata. Ini memberikan gambaran keseluruhan tentang seberapa baik jawaban relevan dan sejauh mana semua item relevan teridentifikasi dengan benar. Nilai *F1-Score* dapat dihitung dengan persamaan berikut ini:

$$\text{F1 - Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

## 2.9 Tensorflow

Salah satu perangkat lunak sumber terbuka yang digunakan untuk pemrosesan data aliran dalam komputasi numerik. Node dalam grafik merepresentasikan operasi

matematika, sementara pengiriman data multidimensi (tensor) antara node-node tersebut direpresentasikan oleh tepi grafik. Tensorflow terdiri dari empat komponen utama dalam grafik komputasi: operasi, tensor, variabel, dan sesi. Keunggulan Tensorflow yaitu kecepatan, kemudahan adaptasi, dan kesiapan untuk produksi (Tobias et al., 2020).

### **2.10 Tensorflow Lite**

Tensorflow Lite adalah library yang dikembangkan khusus untuk perangkat mobile. Hal ini memungkinkan penggunaan machine learning pada perangkat dengan ukuran biner yang kecil. Tensorflow Lite juga kompatibel dengan berbagai sistem operasi, termasuk Linux (melalui API Python/Java/C++), iOS (melalui API C++), dan Android (melalui API Java/C++) (Tobias et al., 2020).

### **2.11 Android Studio**

*Android Studio* diciptakan dengan tujuan untuk memfasilitasi pengembangan aplikasi *Android*. Selain itu, Android Studio dilengkapi dengan beberapa fitur penting, seperti emulator yang responsif, integrasi kode dengan GitHub, serta kemudahan dalam menghubungkan Google Cloud Platform dengan App Engine berkat dukungan untuk C++, NDK, dan Google Cloud Messaging.

### **2.12 Kutu Daun (*Thrips*)**

Kutu daun adalah serangga kecil yang termasuk dalam keluarga *Aphididae* dan dikenal sebagai hama pada tanaman brokoli. Kutu daun merusak tanaman dengan menyerap cairan sel daun yang menyebabkan daun mengering, menguning, dan akhirnya layu. Pada tanaman brokoli, kutu daun sering ditemukan bergerombol saat masa berbunga, menyebabkan daun yang terinfeksi menunjukkan bintik-bintik kotor.



**Gambar 2.3** Contoh daun brokoli yang terkena *Thrips*

### 2.13 Ulat Daun

Ulat daun adalah fase larva dari serangga yang dikenal dengan nama *Plutella xylostella*. Serangga ini merupakan hama utama pada tanaman kubis seperti brokoli. Mereka dapat menyebabkan kerusakan yang signifikan pada tanaman budidaya karena mampu memakan daun tanaman secara massal.



**Gambar 2.4** Contoh daun brokoli yang terkena Ulat Daun

### 2.14 Bercak Daun Alternaria

Bercak-bercak berwarna coklat gelap yang tumbuh dengan cepat di daun merupakan gejala dari serangan jamur *Alternaria brassicae*. Hama ini dapat

mengakibatkan penurunan kualitas daun dan merugikan produksi tanaman brokoli, terutama jika serangan terjadi pada tahap pertumbuhan yang penting.



**Gambar 2.5** Contoh daun yang terkena Bercak Daun Alternaria

### **2.15 Sehat**

Daun brokoli yang sehat umumnya memiliki warna hijau yang cerah menandakan kandungan klorofil yang tinggi, Ujung daun brokoli yang sehat harus tampak segar, tidak layu, tidak terdapat tanda-tanda serangga atau hama.



**Gambar 2.6** Contoh daun brokoli yang Sehat

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Analisis sistem merupakan proses evaluasi dan memahami sistem serta mengidentifikasi semua komponen yang dibutuhkan. Tujuan utamanya adalah untuk memfasilitasi perbaikan sistem di masa depan sehingga sistem yang dirancang mampu beroperasi dengan optimal.

##### **3.1.1 Analisis Masalah**

Analisis masalah merupakan langkah yang diambil untuk memahami dan merumuskan permasalahan yang timbul dalam pengembangan sistem, dengan fokus pada penciptaan sistem yang optimal dalam kinerjanya. Dalam penelitian ini, fokus analisis masalah adalah bagaimana mengidentifikasi dan menampilkan jenis hama yang menyerang brokoli kepada petani. Pendekatan yang diusulkan adalah pengembangan model klasifikasi menggunakan Convolutional Neural Network EfficientNet-B0, model ini diharapkan mampu mengidentifikasi jenis hama dari citra daun brokoli dengan akurat sehingga memberikan informasi yang tepat kepada petani.

##### **3.1.2 Analisis Kebutuhan**

Analisis kebutuhan merupakan langkah dalam pengembangan sistem yang ditujukan untuk mengidentifikasi dan mengevaluasi kebutuhan serta persyaratan sistem. Proses ini melibatkan dua aspek utama, yaitu kebutuhan yang berhubungan dengan fungsi sistem (kebutuhan fungsional) dan kebutuhan yang berhubungan dengan kualitas sistem (kebutuhan non-fungsional).

#### **1. Fungsional**

Dalam pembuatan sistem ini, kebutuhan fungsional yang diperlukan adalah sebagai berikut:

- a. Sistem mampu untuk mengklasifikasikan serangan hama pada tanaman brokoli dengan menggunakan gambar yang dimasukkan melalui metode Convolutional Neural Network.
- b. Sistem mampu menampilkan hasil dari klasifikasi tersebut sesuai dengan gambar yang telah diinput.

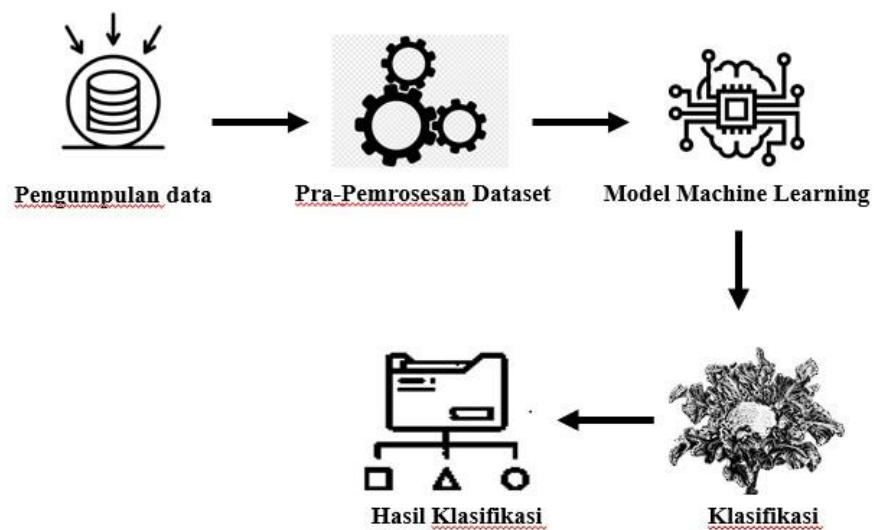
## 2. Non-Fungsional

Kebutuhan non-fungsional menggambarkan bagaimana suatu sistem beroperasi sesuai dengan ekspektasi yang telah ditetapkan. Kebutuhan tersebut mencakup:

- a. Sistem yang dibangun menggunakan user interface (UI) yang mudah dipahami dan digunakan oleh pengguna.
- b. Sistem tidak memerlukan perangkat tambahan sehingga lebih efisien.
- c. Sistem yang dikembangkan dapat dijalankan tanpa memerlukan koneksi internet.

### 3.1.3 Arsitektur Sistem

Arsitektur umum sistem dalam penelitian ini yaitu melakukan Pengumpulan data berupa gambar yang diambil secara langsung dari lahan pertanian brokoli di daerah Berastagi dengan menggunakan kamera *smatrphone*. Gambar yang diambil pada penelitian ini adalah gambar daun brokoli saat usia brokoli 1 bulan / 40 % siap panen, dikarenakan mengambil gambar pada usia 1 bulan memberikan data yang relvan dari kondisi tanaman yang sehat dan terinfeksi pada tahap pertumbuhan yang signifikan. Data ini membantu dalam melatih model CNN untuk mengenali perbedaan antara daun yang sehat dan yang terkena hama dengan lebih baik. Setelah itu data di pra-proses yang mencakup normalisasi data yang bertujuan agar data memiliki ukuran dan skala yang seragam. Data yang telah di pra-proses dijadikan input untuk membangun model machine learning dengan pendekatan CNN arsitektur *EfficientNet-B0*. Model *machine learning* digunakan untuk mengklasifikasikan hama yang menyerang daun brokoli. Hasil klasifikasi hama dari citra daun brokoli dapat diketahui setelah melewati alur arsitektur umum sistem, yang terdiri atas 4 kelas yaitu ulat daun kubis, kutu daun, bercak daun *Alternaria*, dan sehat.




**Gambar 3.1** Arsitektur Sistem




### 3.1.4 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini berupa gambar yang diambil secara langsung dari lahan pertanian brokoli di daerah Berastagi dengan format .jpg. Dari 2120 data gambar tersebut dibagi atas empat kelas yaitu jamur, thrips, ulat, dan sehat. Setiap kelas memiliki jumlah data sama yaitu 530 data dan dibagi menjadi data pelatihan 70 %, data validasi 20%, dan data pengujian 10%.

**Tabel 3.1** Jumlah Data per Kelas

No	Kelas	Gambar Sampel	Jumlah
1	Jamur		530



2	Thrips		530
3	Ulat		530
4	Sehat		530

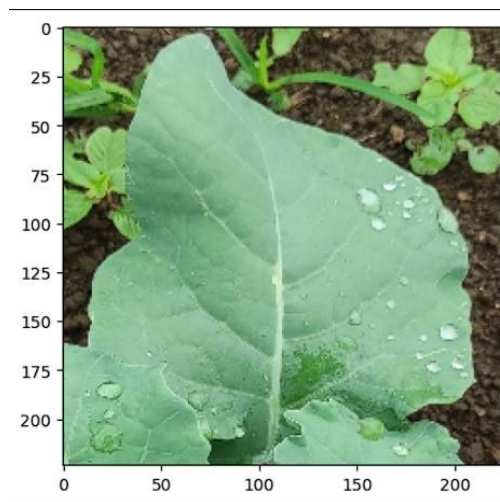
### 3.1.5 Pre-Processing Data

Pre-processing merupakan proses persiapan dan penyiapan data sebelum data tersebut dapat digunakan untuk analisis atau diproses lebih lanjut. Pada penelitian ini menggunakan resize dan rescale.

#### 1. Resize

Resize adalah proses mengubah ukuran gambar menjadi ukuran yang ditentukan. Ini sering digunakan untuk memastikan bahwa semua gambar dalam dataset memiliki ukuran yang konsisten sebelum dimasukkan ke

dalam model CNN. Pada Penelitian ini ukuran gambar di ubah menjadi 224 x 224.



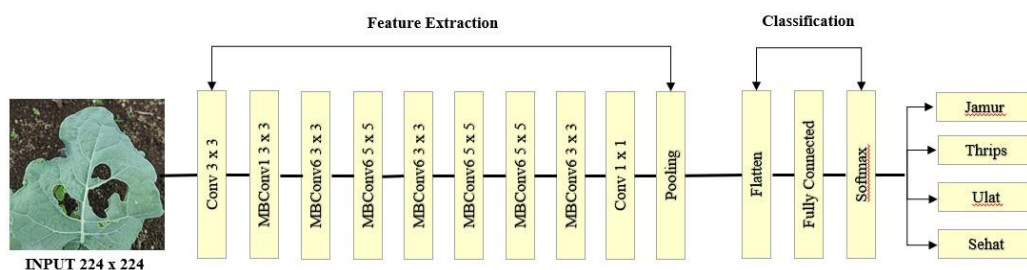
**Gambar 3.2** Gambar di resize menjadi 224 x 224

## 2. Rescale

Rescale adalah proses normalisasi intensitas piksel dalam gambar. Ini melibatkan pengurangan rentang nilai piksel lebih kecil, seperti dari 0-255 menjadi 0-1 atau -1 hingga 1. Rescale membantu dalam mempercepat konvergensi selama pelatihan model karena menghindari masalah dengan skala fitur yang berbeda dalam data.

### 3.1.6 Arsitektur *EfficientNet-B0*

Setelah melakukan pre-processing data, langkah selanjutnya adalah membangun model machine learning dengan pendekatan CNN arsitektur *EfficientNet-B0*. Secara umum CNN memiliki dua tahapan yaitu tahap feature learning dan classification. Berikut adalah desain arsitektur CNN dalam penelitian ini:



**Gambar 3.3** Arsitektur *EfficientNet-B0*

Pada gambar 3.3 terlihat tahapan arsitektur *EfficientNet-B0* yang akan diterapkan pada penelitian ini untuk klasifikasi data hama brokoli. Berikut tahapan-tahapan arsitektur *EfficientNet-B0* dalam melakukan klasifikasi gambar:

1. Input Layer

Model menerima gambar dengan ukuran 224 x 224 piksel dan 3 saluran warna (RGB) sebagai input. Gambar ini kemudian diproses oleh lapisan-lapisan berikutnya untuk ekstraksi fitur dan klasifikasi.

2. *Feature Extraction*

Bagian ini terdiri dari beberapa lapisan konvolusi, MBConv, dan *pooling* yang bertujuan untuk mengekstraksi fitur penting dari gambar input.

a. *Conv 3x3*

Lapisan konvolusi pertama menggunakan filter berukuran 3 x 3 untuk melakukan operasi konvolusi pada gambar input. Proses ini menghasilkan sebuah feature map dengan dimensi 224 x 224 x 64. Feature map ini berperan penting dalam proses deteksi fitur dasar seperti tepi dan tekstur pada gambar. Dengan menerapkan filter konvolusi, lapisan ini dapat mengekstrak informasi visual yang relevan dari gambar asli, membantu model untuk mengenali pola yang lebih kompleks pada tahap-tahap berikutnya dalam jaringan saraf.

b. *MBConv Layers*

Lapisan MBConv menggunakan *depthwise separable convolution*, terdiri dari *depthwise convolution* (filter 3 x 3 atau 5 x 5 pada setiap saluran input) dan *pointwise convolution* (1 x 1 *convolution* untuk mengintegrasikan hasilnya). Ini dirancang untuk memperluas dan mereduksi dimensi saluran fitur, menangkap informasi kontekstual kompleks dari data input seperti gambar, dan meningkatkan kemampuan jaringan untuk mengenali pola yang rumit.

c. *Pooling Layer*

*Max Pooling* menggunakan filter 2 x 2 dengan stride 2 untuk mengurangi dimensi spasial feature map sebelumnya. Setiap area 2 x 2 dari feature map

diproses untuk menghasilkan nilai maksimum, mengubahnya menjadi feature map yang lebih kecil seperti dari  $224 \times 224 \times 128$  menjadi  $112 \times 112 \times 128$ . Prosedur ini mengurangi beban komputasi dan mempertahankan informasi penting dari data asli.

### 3. Classification

Bagian ini mengubah fitur yang diekstraksi menjadi prediksi kelas akhir melalui beberapa lapisan *fully connected* dan softmax.

#### a. Flatten Layer

Lapisan Flatten mengubah feature map  $112 \times 112 \times 128$  menjadi vektor satu dimensi dengan panjang 1,605,632 elemen, yang kemudian diproses oleh lapisan *fully connected* dalam jaringan neural. Ini memungkinkan representasi spasial dari fitur menjadi tersedia dalam bentuk vektor untuk analisis lebih lanjut dan pengambilan keputusan.

#### b. Fully Connected Layer

Lapisan Fully Connected dalam jaringan neural menghubungkan setiap neuron dengan semua neuron dari lapisan sebelumnya, menciptakan output yang sangat terhubung untuk integrasi dan pengambilan keputusan model. Jumlah parameter dalam lapisan ini berbanding langsung dengan jumlah neuronnya, misalnya dengan 1000 neuron akan ada 1,606,632,000 parameter yang terlibat.

#### c. Softmax Layer

Lapisan Softmax dalam jaringan *neural networks* menerapkan fungsi softmax pada output yang dihasilkan oleh lapisan *fully connected*. Fungsi ini mengubah nilai-nilai output menjadi probabilitas untuk setiap kelas seperti jamur, thrips, ulat, dan sehat. Probabilitas ini mengindikasikan seberapa besar kemungkinan setiap kelas mewakili prediksi yang tepat. Kelas dengan probabilitas tertinggi dianggap sebagai prediksi akhir dari model, memungkinkan untuk pengklasifikasi yang akurat berdasarkan distribusi probabilitas dari output terakhir.

#### 4. Output Layer

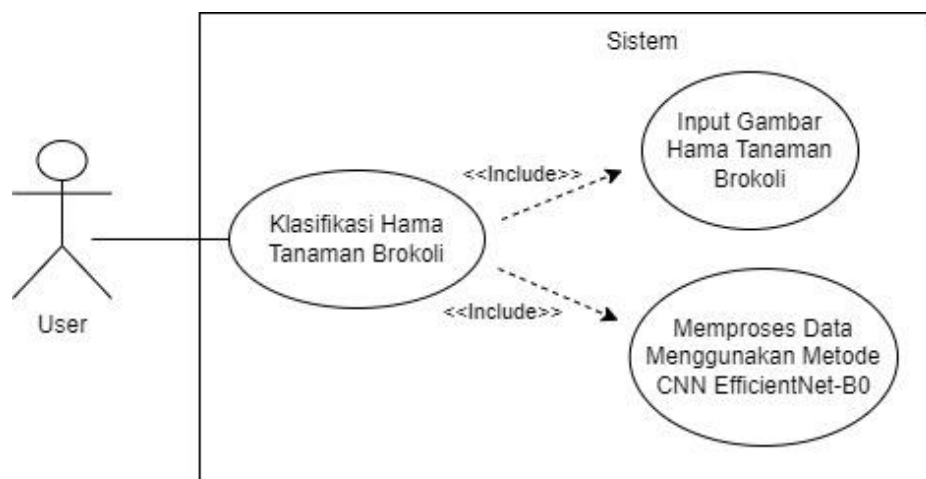
Setelah melalui lapisan Softmax, model klasifikasi memanfaatkan probabilitas yang dihasilkan untuk mengklasifikasi gambar daun ke dalam salah satu dari empat kelas yang telah ditetapkan yaitu jamur, thrips, ulat, atau sehat. Ini memungkinkan model untuk membuat prediksi akhir berdasarkan distribusi probabilitas dari output terakhir, yang terbentuk melalui proses pelatihan dan pembelajaran menggunakan data yang ada.

### 3.2 Pemodelan Sistem

Pemodelan sistem merupakan metode untuk menetapkan fungsi dan tujuan inti dari suatu sistem, dengan tujuan memperoleh pemahaman yang mendalam tentang kebutuhan pengguna serta gambaran menyeluruh tentang struktur sistem.

#### 3.2.1 Use Case Diagram

Diagram ini menggambarkan hubungan antara sistem dan pengguna.

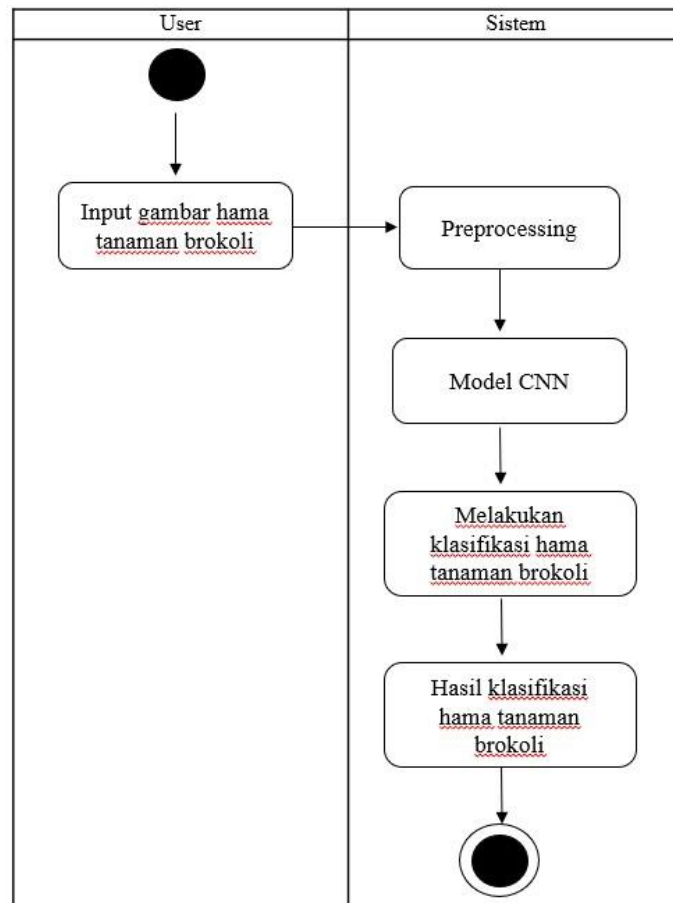


**Gambar 3.4** Use Case Diagram

Pada Gambar 3.4 terlihat skenario pemanfaatan sistem ini dimulai dengan pengguna membuka aplikasi sistem. Setelah itu, pengguna menginputkan gambar hama pada tanaman brokoli yang akan dikelola oleh sistem untuk melakukan klasifikasi. Selanjutnya, sistem akan menjalankan proses klasifikasi dengan menggunakan model yang telah disiapkan dan dibangun dengan metode CNN EfficientNet-B0. Setelah itu, sistem akan menampilkan hasil klasifikasi.

### 3.2.2 Activity Diagram

Diagram ini menjelaskan representasi visual dari aktivitas yang terjadi dalam sistem. Diagram ini menggambarkan urutan langkah-langkah dalam proses kerja dari tahap awal hingga akhir.



**Gambar 3.5** Activity Diagram

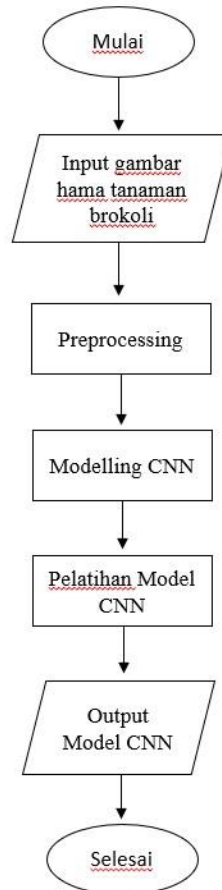
Dari gambar 3.5 langkah awal penggunaan sistem dimulai ketika pengguna menekan tombol "Ambil Gambar" untuk mengunggah gambar hama pada tanaman brokoli. Setelah itu sistem akan melakukan proses klasifikasi pada gambar yang telah di input dan menampilkan hasil klasifikasi kepada pengguna.

### 3.3 Flowchart

Flowchart adalah representasi visual yang digunakan untuk mengilustrasikan urutan langkah dan keputusan yang diperlukan dalam menjalankan suatu proses. Flowchart memiliki manfaat dalam mendokumentasikan, mempelajari,

merencanakan, dan mengoptimalkan proses yang kompleks dalam bentuk diagram yang mudah dimengerti.

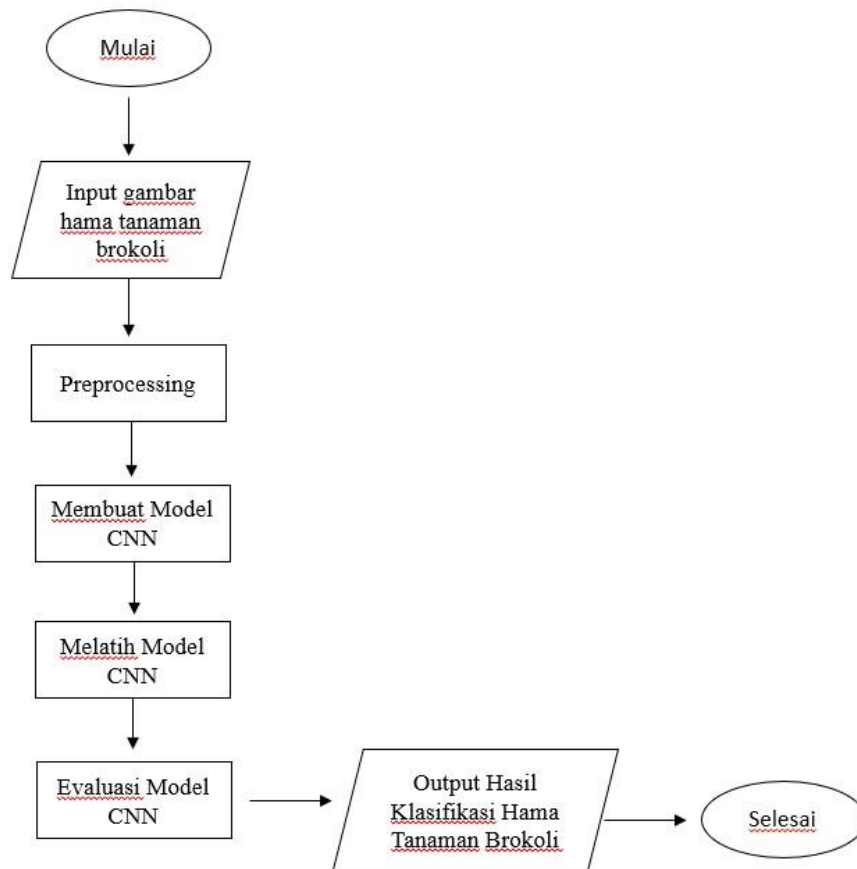
### 3.3.1 Flowchart Pelatihan Model CNN *EfficientNet-B0*



**Gambar 3.6** Flowchart pelatihan model CNN

Pada gambar 3.6 pertama pengguna memasukkan gambar hama pada tanaman brokoli. Gambar tersebut kemudian akan mengalami tahap pre-processing. Pada tahap pre-processing gambar akan dilakukan resizing, scalling, pembagian data training, data testing, dan validation. Selanjutnya dataset yang telah dilakukan pre-processing akan dilakukan proses pembuatan model CNN untuk dilatih sehingga menghasilkan sebuah model CNN *EfficientNet-B0*.

### 3.3.2 Flowchart Sistem



**Gambar 3.7** Flowchart Sistem

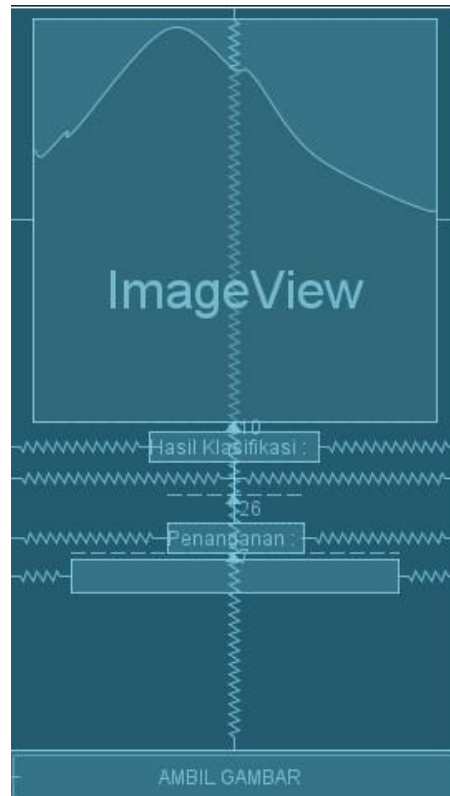
Pertama pengguna menginputkan gambar hama pada tanaman brokoli. Gambar tersebut akan mengalami pre-processing. Setelah itu membuat model CNN *EfficientNet-B0* yang akan digunakan untuk proses pelatihan / melakukan klasifikasi. Setelah mendapatkan model CNN *EfficientNet-B0* yang sudah terlatih model akan dievaluasi sehingga akan menghasilkan output hasil klasifikasi hama pada tanaman brokoli.

### 3.4 Perancangan Interface

Perancangan interface adalah bagian penting dari proses pengembangan aplikasi yang bertujuan untuk menciptakan tampilan yang memfasilitasi interaksi antara pengguna dan sistem. Tujuan dari perancangan interface adalah menghasilkan representasi visual yang memungkinkan pengguna berinteraksi dengan aplikasi secara efektif, memastikan bahwa aplikasi dapat berfungsi sesuai kebutuhan.



### 3.4.1 Halaman Tampilan Sistem



**Gambar 3.8** Desain Halaman Tampilan Sistem

Gambar diatas merupakan desain halaman tampilan sistem sederhana. Pada tampilannya terdapat tombol “ambil gambar”, label “hasil klasifikasi”, label “penanganan”, dan kotak gambar (*Image View*). Pada bagian menu “ambil gambar”, pengguna akan diminta untuk mengambil gambar dengan kamera *smartphone* untuk melakukan klasifikasi. Setelah itu sistem akan menampilkan gambar daun brokoli yang telah diambil pada kotak gambar (*Image View*) serta hasil klasifikasi dan penanganan pada label “hasil klasifikasi” dan label “penanganan”.

## BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

### 4.1 Implementasi Sistem

Setelah tahap desain dan analisis selesai, sistem yang dikembangkan kemudian diterapkan dan diuji untuk memastikan bahwa sistem tersebut memenuhi harapan dan dapat mengidentifikasi hama pada tanaman brokoli.

#### 4.1.1 Perangkat Keras (*Hardware*)

Berikut ini merupakan spesifikasi perangkat keras pada penelitian ini.

**Tabel 4.1** Spesifikasi Perangkat Keras

Perangkat Keras	Komponen	
ASUS TUF Gaming A15 FA506IH_FX506IH	CPU	: AMD Ryzen 5 4600H with Radeon Graphics (12 CPUs), ~3.0GHz
	GPU	: NVIDIA GeForce GTX 1650 4 GB
	RAM	: 16 GB
	SSD	: 512 GB
Xiaomi Redmi Note 9 Pro	Memori Internal : 64 GB	
	RAM	: 6 GB
	Camera	: 64 MP
	Battery	: 5020 mAh

#### 4.1.2 Perangkat Lunak (*Software*)

Berikut ini merupakan spesifikasi perangkat lunak pada penelitian ini

**Tabel 4.2** Spesifikasi Perangkat Lunak

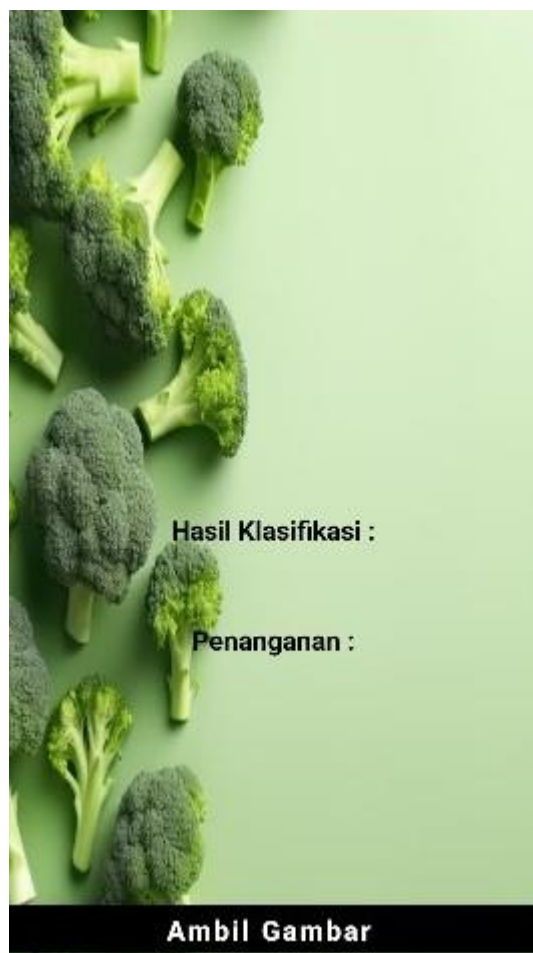
Perangkat	Software
ASUS TUF Gaming A15 FA506IH_FX506IH	OS : Windows 11 Home Single Language
	Notebook editor : Jupyter Lab
	Android Studio Electric Eel   2022.1.1 Patch 2
	Bahasa Pemrograman Java Script, Python

Xiaomi Redmi Note 9 Pro

OS : Android 10, MIUI 11

#### 4.1.3 Halaman Utama

Berikut merupakan tampilan awal yang akan muncul kepada pengguna saat memulai aplikasi, dimana terdapat tombol “ambil gambar” yang digunakan untuk mengambil gambar daun brokoli yang akan diklasifikasi. Pada tampilan ini juga terdapat label hasil klasifikasi dan penanganan.



**Gambar 4.1** Tampilan Halaman Utama

#### 4.1.4 Halaman Hasil

Berikut adalah tampilan hasil yang akan muncul kepada pengguna setelah mengambil gambar daun brokoli. Gambar yang telah di ambil akan ditampilkan

pada halaman ini beserta hasil klasifikasi dan penanganan sesuai hama yang terklasifikasi.



**Gambar 4.2** Tampilan Halaman Hasil

## 4.2 Pelatihan Model

Berikut tahapan – tahapan yang akan dilakukan dalam melakukan pelatihan model.

### 4.2.1 Import Library Tensorflow

```
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
from IPython.display import HTML
from tensorflow.keras.applications import EfficientNetB0
```

**Gambar 4. 3** Kode untuk *Import* Tensorflow

Gambar 4.3 diatas merupakan proses mengimpor tensorflow untuk membangun model neural network dengan *EfficientNet-B0* sebagai arsitektur dasar. Ini melibatkan impor library tensorflow untuk operasi machine learning, termasuk modul keras untuk definisi model dan lapisan-lapisannya. Matplotlib diimpor untuk visualisasi data atau hasil, dan IPython.display.HTML untuk kemungkinan integrasi interaktif dalam notebook.

#### 4.2.2 Proses Mengakses Dataset

```
dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "C:\\Users\\Asus\\Documents\\Skripsi\\Brokoli3",
    Found 2120 files belonging to 4 classes.
```

**Gambar 4.4** kode untuk mengakses dan menampilkan dataset

Gambar 4.4 diatas merupakan pengaksesan dataset gambar dari sebuah direktori pada sistem file ke dalam format yang dapat langsung digunakan untuk melatih model *neural network* menggunakan TensorFlow. Path "C:\\Users\\Asus\\Documents\\Skripsi\\Brokoli3" menunjukkan direktori tempat dataset gambar disimpan.

#### 4.2.3 Memuat Model *EfficientNet-B0*

```
from tensorflow.keras import models, layers
from tensorflow.keras.applications import EfficientNetB0
BATCH_SIZE = 64
IMAGE_SIZE = 224
CHANNELS = 3
n_classes = 4
base_model = EfficientNetB0(include_top=False, weights='imagenet', input_shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS))
base_model.trainable = False
```

**Gambar 4.5** Kode Untuk Memuat Model *EfficientNet-B0*

Gambar 4.5 diatas merupakan proses tensorflow mengimpor model dan lapisan-lapisan dari Keras serta memuat *EfficientNet-B0* dari tensorflow.keras.applications. Batch size didefinisikan sebagai 64, image\_size sebagai 224, channels sebagai 3, dan n\_classes sebagai 4, yang mencerminkan jumlah kelas yang akan digunakan. Objek base\_model dibuat dengan menggunakan *EfficientNet-B0* yang telah dilatih pada dataset ImageNet (weights='imagenet') dan konfigurasi include\_top=False, yang menghapus lapisan fully connected di bagian atas model untuk memungkinkan penggunaan output fitur dari lapisan terakhir.

Parameter `input_shape=(image_size, image_size, channels)` menentukan dimensi gambar yang akan diterima sebagai input oleh model, dengan resolusi 224 x 224 piksel dan tiga saluran warna untuk gambar RGB. Selanjutnya, `base_model.trainable` diatur ke `False`, yang mengunci semua layer dalam model *EfficientNet-B0* sehingga tidak akan dilatih ulang selama pelatihan lanjutan. Penggunaan batch size 64 merupakan pilihan yang umum dalam CNN.

#### 4.2.4 Proses Pelatihan

Proses pelatihan pada sistem menggunakan metode CNN *EfficientNet-B0*. Penulis melakukan beberapa percobaan dengan jumlah *epoch* yang bervariasi, bertujuan untuk mendapatkan hasil yang optimal dalam tingkat akurasi dan loss. Pengaturan optimizer pada pelatihan ini menggunakan optimizer *Adam* serta penggunaan fungsi loss yaitu menggunakan *categorical\_crossentropy (multi class)* untuk tugas klasifikasi hama pada tanaman brokoli dan menggunakan akurasi sebagai metrik evaluasi untuk menilai kinerja model selama pelatihan dan pengujian. Berikut adalah hasil dari proses pelatihan yang dilakukan.

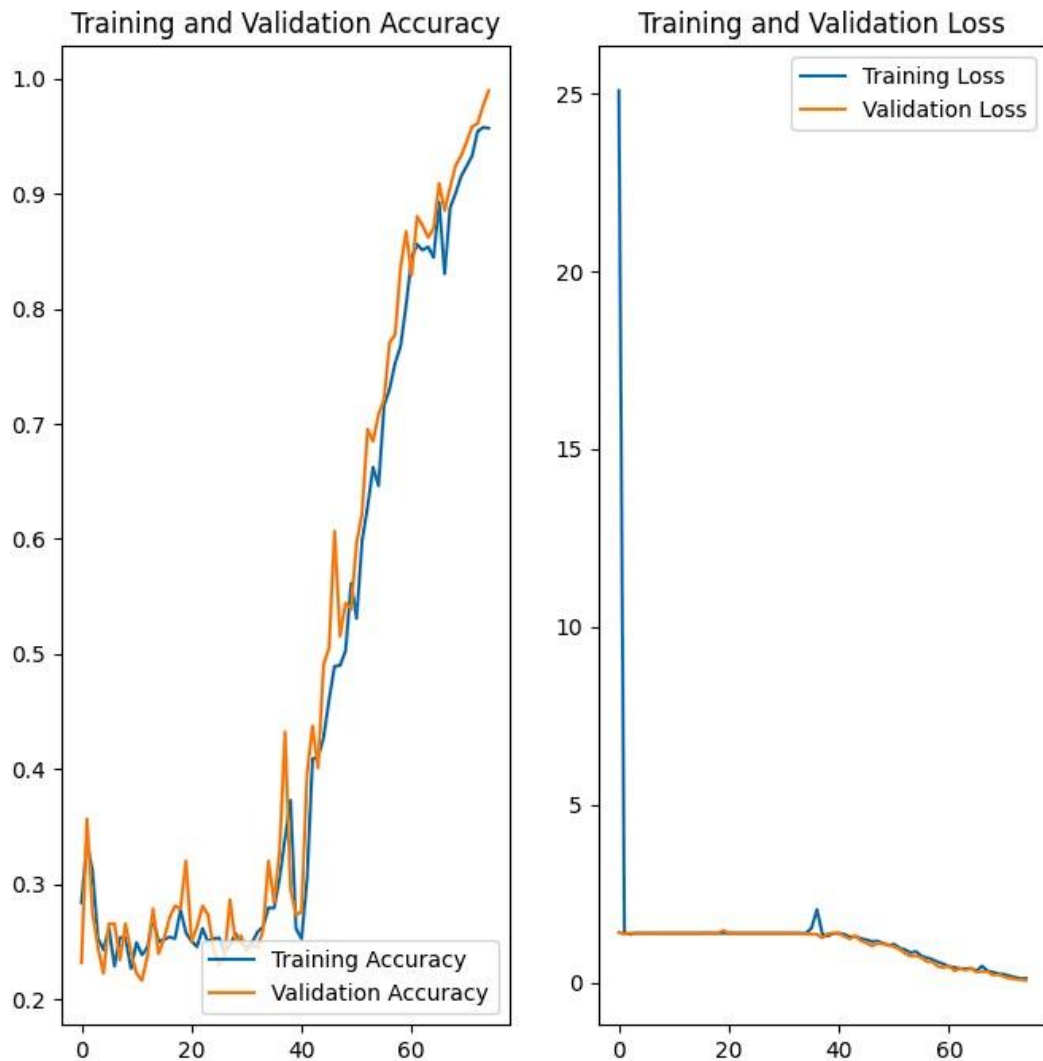
**Tabel 4.3** Percobaan dengan beberapa Jumlah Epoch

Percobaan	Epoch	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
1	25	76,20 %	0,58	75,91 %	0,57
2	50	92,66 %	0,19	95,12 %	0,12
3	75	95,69 %	0,12	98,96 %	0,06

Dari hasil percobaan dapat disimpulkan bahwa pada percobaan ke-3 dengan jumlah *epoch* 75 mendapatkan *training accuracy* sebesar 95,69 % dengan *training loss* 0.12 dan *validation accuracy* sebesar 98,96 % dengan *validation loss* 0.06. Dengan ini penulis menggunakan model tersebut untuk proses pengklasifikasian. Berikut tampilan proses pelatihan.

```
Epoch 70/75
23/23 [=====] - 41s 1s/step - loss: 0.2464 - accuracy: 0.9151 - val_loss: 0.2243 - val_accuracy: 0.9329
Epoch 71/75
23/23 [=====] - 41s 1s/step - loss: 0.2318 - accuracy: 0.9239 - val_loss: 0.1689 - val_accuracy: 0.9451
Epoch 72/75
23/23 [=====] - 40s 1s/step - loss: 0.1878 - accuracy: 0.9329 - val_loss: 0.1167 - val_accuracy: 0.9583
Epoch 73/75
23/23 [=====] - 40s 1s/step - loss: 0.1487 - accuracy: 0.9541 - val_loss: 0.0992 - val_accuracy: 0.9609
Epoch 74/75
23/23 [=====] - 41s 1s/step - loss: 0.1133 - accuracy: 0.9576 - val_loss: 0.0841 - val_accuracy: 0.9766
Epoch 75/75
23/23 [=====] - 40s 1s/step - loss: 0.1210 - accuracy: 0.9569 - val_loss: 0.0618 - val_accuracy: 0.9896
```

**Gambar 4.6** Proses Pelatihan

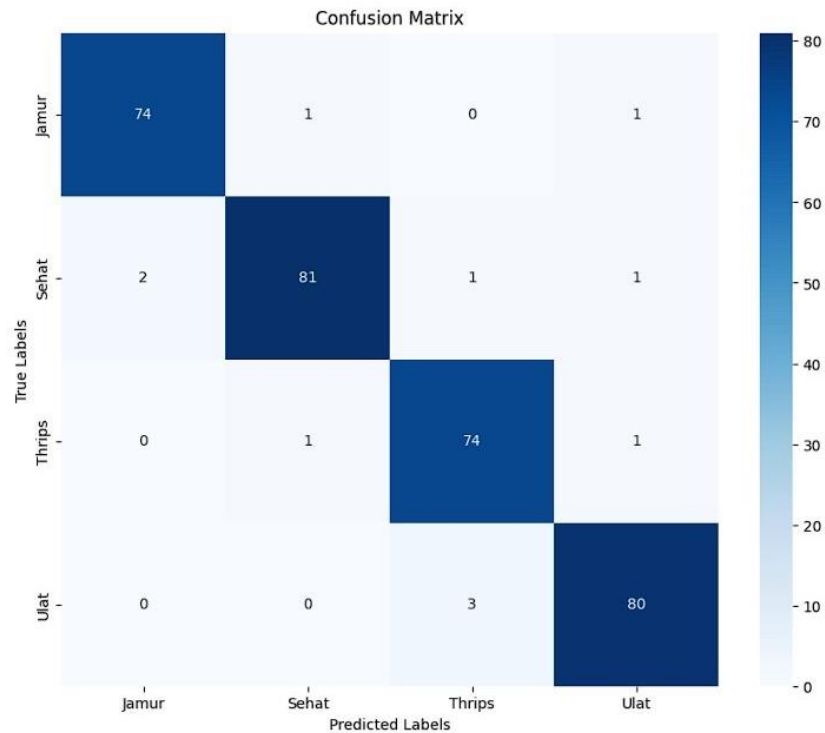


**Gambar 4.7** Grafik Akurasi dan Loss Model saat proses pelatihan

Pada gambar 4.7 memperlihatkan grafik model loss dan akurasi yang dialami selama proses pelatihan. Hasil pada gambar di atas diketahui bahwa model ini cukup baik untuk digunakan karena akurasi pada data training dan validation cukup akurat.

#### 4.2.5 Evaluasi Model

Untuk mengevaluasi model, penelitian ini menggunakan *confusion matrix* yang memberikan gambaran tentang bagaimana prediksi model sesuai dengan hasil sebenarnya.



**Gambar 4.8** Confusion Matrix

Gambar 4.8 diatas merupakan *confusion matrix* dari model. Evaluasi kinerja menggunakan *confusion matrix* memanfaatkan empat komponen utama yaitu:

*True Positive* (TP) mengindikasikan data positif yang benar diklasifikasikan sebagai positif oleh model.

*True Negative* (TN) mengindikasikan data negatif yang benar diklasifikasikan sebagai negatif oleh model.

*False Positive* (FP) mengindikasikan data negatif yang salah diklasifikasikan sebagai positif oleh model.

*False Negative* (FN) mengindikasikan data positif yang salah diklasifikasikan sebagai negatif oleh model.

Akurasi

Akurasi merupakan persentase dari keseluruhan data yang berhasil diidentifikasi dan diklasifikasikan dengan benar. Akurasi dihitung dengan menggunakan rumus berikut:



$$\text{Akurasi} = \frac{\text{TP}}{\text{Total Data}}$$

$$\text{Akurasi} = \frac{74 + 81 + 74 + 80}{320} = \frac{309}{320} = 0,9656 = 96,56 \%$$

### *Precision*

*Precision* adalah sebuah metrik yang mengevaluasi seberapa baik model dapat menghasilkan hasil positif yang benar atau tepat, berdasarkan pada informasi yang tidak akurat atau kurang tepat. Ini merupakan ukuran keakuratan dari hasil positif yang dihasilkan oleh model. *Precision* dihitung dengan menggunakan rumus berikut:

Kelas	TP	FP
Jamur	74	2
Sehat	81	2
Thrips	74	4
Ulat	80	3

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision Jamur} = \frac{74}{76} = 0.9737 = 97,37 \%$$

$$\text{Precision Sehat} = \frac{81}{83} = 0.9759 = 97,59 \%$$

$$\text{Precision Thrips} = \frac{74}{78} = 0.9487 = 94,87 \%$$

$$\text{Precision Ulat} = \frac{80}{83} = 0.9639 = 96,39 \%$$

$$\text{Mean Precision} = \frac{97.37 + 97.59 + 94.87 + 96.39}{4} = \frac{386.22}{4} = 96,55 \%$$

### *Recall*

*Recall* merupakan data yang tidak mampu dihitung dengan benar di mana dibutuhkan nilai *False Negative* untuk menghitung nilai *recall*. *Recall* dihitung dengan menggunakan rumus berikut:

Kelas	TP	FN
Jamur	74	2
Sehat	81	4
Thrips	74	2
Ulat	80	3

$$Recall = \frac{TP}{TP + FN}$$

$$Recall \text{ Jamur} = \frac{74}{76} = 0.9737 = 97,37 \%$$

$$Recall \text{ Sehat} = \frac{81}{84} = 0.9529 = 95,29 \%$$

$$Recall \text{ Thrips} = \frac{74}{76} = 0.9737 = 97,37 \%$$

$$Recall \text{ Ulat} = \frac{80}{83} = 0.9639 = 96,39 \%$$

$$Mean Recall = \frac{97.37 + 95.29 + 97.37 + 96.39}{4} = \frac{386.42}{4} = 96,61 \%$$

#### *F1-Score*

*F1-score* merupakan *harmonic mean* yang digunakan untuk menunjukkan perbandingan antara *precision* dan *recall*. Berikut adalah rumus untuk mendapatkan hasil *f1-score*:

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$





$$F1 - Score = 2 \times \frac{(0,9655 \times 0,9661)}{(0,9655 + 0,9661)} = 2 \times \frac{0,9327}{1,9316} = 0,9657 = 96,57 \%$$







Berdasarkan perhitungan diatas kesimpulan yang didapat adalah metode CNN *EfficientNet-B0* dalam melakukan klasifikasi mendapat akurasi sebesar 96,56%, *precision* sebesar 96,55%, *recall* sebesar 96,61%, dan *F1-Score* sebesar 96,57%.

#### 4.4 Pengujian Sistem

Setelah melewati beberapa tahapan seperti pelatihan model dan evaluasi model berikut akan dilakukan pengujian sistem untuk mengetahui kemampuan dari sistem yang telah dibangun untuk melakukan klasifikasi dari citra daun brokoli menggunakan CNN *EfficientNet-B0*.





**Tabel 4.4** Pengujian Sistem





No	Gambar	Kelas	Hasil Klasifikasi	Output
1		Jamur	 <p>My Brokoli</p> <p>Hasil Klasifikasi : <b>Jamur</b></p> <p>Penanganan : Melakukan penyemprotan fungisida berbahan aktif Mankozeb &amp; Klorotanil, Membersihkan gulma di sekitar tanaman agar tanah tidak lembab</p> <p>Ambil Gambar</p>	Jamur
2		Sehat	 <p>My Brokoli</p> <p>Hasil Klasifikasi : <b>Sehat</b></p> <p>Penanganan :</p> <p>Ambil Gambar</p>	Sehat

No	Gambar	Kelas	Hasil Klasifikasi	Output
3		Thrips	<p>My Brokoli</p>  <p>Hasil Klasifikasi : <b>Thrips</b></p> <p>Penanganan : Melakukan penyemprotan insektisida berbahan aktif Amebectin &amp; Sprimetrin</p> <p>Ambil Gambar</p>	Thrips
4		Ulat	<p>My Brokoli</p>  <p>Hasil Klasifikasi : <b>Ulat</b></p> <p>Penanganan : Melakukan penyemprotan insektisida berbahan aktif Asefat &amp; Amebectin</p> <p>Ambil Gambar</p>	Ulat
5		Thrips	<p>My Brokoli</p>  <p>Hasil Klasifikasi : <b>Ulat</b></p> <p>Penanganan : Melakukan penyemprotan insektisida berbahan aktif Asefat &amp; Amebectin</p> <p>Ambil Gambar</p>	Ulat

Hasil yang didapat pada Tabel 4.4 dapat dilihat bahwa klasifikasi berjalan dengan baik pada Sistem.

**Tabel 4.5** Pengujian Berdasarkan Mayoritas Hama yang Ada Pada Daun

No	Gambar	Kelas	Hasil Klasifikasi	Output
1		Jamur dan Ulat	 <p>My Brokoli</p> <p>Hasil Klasifikasi : <b>Jamur</b></p> <p>Penanganan : Melakukan penyemprotan fungisida berbahan aktif Mankozeb &amp; Klorotanil, Membersihkan gulma di sekitar tanaman agar tanah tidak lembab</p> <p>Ambil Gambar</p>	Jamur
2		Thrips dan Ulat	 <p>My Brokoli</p> <p>Hasil Klasifikasi : <b>Thrips</b></p> <p>Penanganan : Melakukan penyemprotan insektisida berbahan aktif Anebectin &amp; Sprimetrin</p> <p>Ambil Gambar</p>	Thrips

No	Gambar	Kelas	Hasil Klasifikasi	Output
3		Thrips dan Sehat		Thrips
4		Jamur dan Ulat		Jamur

Dari tabel 4.5 diatas, penggunaan *CNN EffecientNet-B0* dalam melakukan klasifikasi hama pada tanaman brokoli menghasilkan kinerja yang baik. Model CNN memberikan hasil klasifikasi dengan akurat pada setiap gambar yang diuji. Untuk penanganan hama pada sistem yang diberikan sudah benar berdasarkan kelas hasil klasifikasi berikut rincian dari masing-masing penanganan hama:

#### Hama Jamur

Melakukan penyemprotan berbahan aktif *Mankozebe* dan *klorotalonil*. *Mankozebe* digunakan untuk mengontrol penyakit seperti *Alternaria*, *antraknosa*, dan *botrytis* sedangkan *klorotalonil* merupakan *fungisida* spektrum luas yang digunakan untuk mengontrol berbagai penyakit tanaman serta mengurangi kelembapan tanah dengan

membersihkan gulma sangat penting untuk mencegah perkembangan jamur, karena kelembapan tinggi dapat memicu pertumbuhan jamur.

#### Hama *Thrips*

Melakukan penyemprotan berbahan aktif *amebectin* dan *spinosad*. *Amebectin* adalah insektisida yang efektif melawan thrips dengan mempengaruhi sistem saraf mereka dan *spinosad* digunakan secara luas untuk mengendalikan *thrips* melalui mekanisme kerja yang berbeda, yakni sebagai racun saraf yang mempengaruhi reseptor asetilkolin pada serangga.

#### Hama Ulat

Melakukan penyemprotan berbahan aktif *asefat* dan *amebectin*. Asefat adalah insektisida *organofosfat* yang bekerja dengan menghambat enzim *kolinesterase* pada ulat, menyebabkan akumulasi *asetilkolin* yang mengakibatkan kematian ulat dan *amebectin*, juga efektif melawan ulat, bekerja dengan cara yang berbeda, yaitu mempengaruhi saluran klorida dalam sistem saraf ulat.

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil penelitian dan pengujian yang telah dilakukan mengenai klasifikasi hama tanaman brokoli menggunakan Convolutional Neural Network EfficientNet-B0, diperoleh beberapa kesimpulan:

1. Penelitian ini bertujuan untuk mengklasifikasikan hama yang menyerang tanaman brokoli dengan menggunakan CNN *EfficientNet-B0*. Temuan dari penelitian menunjukkan bahwa penerapan CNN *EfficientNet-B0* terbukti efektif dalam mengidentifikasi hama pada tanaman brokoli. Dengan menggunakan model CNN *EfficientNet-B0* yang telah dikembangkan, penulis berhasil mencapai tingkat akurasi tertinggi yaitu *training accuracy* sebesar 95,69 % dengan *training loss* 0.12 dan *validation accuracy* sebesar 98,96 % dengan *validation loss* 0.06 dengan jumlah 75 *epoch* pada pelatihan model. Pada evaluasi model menggunakan *confusion matrix* mendapatkan nilai akurasi 96,56 %, nilai *precision* 96,55 %, nilai *recall* 96,61 %, dan 96,57% *F1-score*.
2. Penerapan CNN dalam bidang pertanian, khususnya pada tanaman brokoli, dapat membantu petani dan peneliti dalam mengidentifikasi gejala penyerangan hama dengan cepat dan akurat. Dengan demikian, tindakan pencegahan dan pengendalian yang tepat waktu dapat diambil untuk mengurangi kerugian hasil panen. Penelitian ini memberikan kontribusi positif dalam upaya meningkatkan produktivitas pertanian dan keberlanjutan pertanian secara keseluruhan.

### 5.2 Saran

Dari pembahasan yang telah diuraikan, peneliti memberikan beberapa saran untuk dipertimbangkan dalam penelitian berikutnya:

1. Kepada peneliti selanjutnya diharapkan memanfaatkan perkembangan terbaru dalam arsitektur jaringan saraf, seperti *Vision Transformers* (ViTs) atau hybrid models yang menggabungkan CNN dengan ViTs untuk meningkatkan akurasi dan efisiensi.



2. Kepada peneliti selanjutnya diharapkan dapat meningkatkan klasifikasi dengan mempertimbangkan faktor-faktor tambahan seperti ketinggian dari pengambilan data hama tanaman.

## DAFTAR PUSTAKA

- Ahmed, M. I. B., Saraireh, L., Rahman, A., Al-Qarawi, S., Mhran, A., Al-Jalaoud, J., Al-Mudaifer, D., Al-Haidar, F., AlKhulaifi, D., Youldash, M., & Gollapalli, M. (2023). Personal Protective Equipment Detection: A Deep-Learning-Based Sustainable Approach. *Sustainability (Switzerland)*, 15(18). <https://doi.org/10.3390/su151813990>
- Akshai, K. P., & Anitha, J. (2021). Plant disease classification using deep learning. *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, 407–411. <https://doi.org/10.1109/ICSPC51351.2021.9451696>
- Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(2), 640. <https://doi.org/10.30865/mib.v5i2.2937>
- Azizah, Q. N. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet. *Sudo Jurnal Teknik Informatika*, 2(1), 28–33. <https://doi.org/10.56211/sudo.v2i1.227>
- Boulent, J., Foucher, S., Théau, J., & St-Charles, P. L. (2019). Convolutional Neural Networks for the Automatic Identification of Plant Diseases. In *Frontiers in Plant Science* (Vol. 10). Frontiers Media S.A. <https://doi.org/10.3389/fpls.2019.00941>
- Falaschetti, L., Manoni, L., Leo, D. Di, Pau, D., Tomaselli, V., & Turchetti, C. (2022). A CNN-based image detector for plant leaf diseases classification. *HardwareX*, 12, e00363. <https://doi.org/10.17605/OSF.IO/UCM8D>
- Gajjar, R., Gajjar, N., Thakor, V. J., Patel, N. P., & Ruparelia, S. (2022). Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. *Visual Computer*, 38(8), 2923–2938. <https://doi.org/10.1007/s00371-021-02164-9>
- Hadianto, N., Novitasari, H. B., & Rahmawati, A. (2019). KLASIFIKASI PEMINJAMAN NASABAH BANK MENGGUNAKAN METODE NEURAL NETWORK. *Jurnal Pilar Nusa Mandiri*, 15(2), 163–170. <https://doi.org/10.33480/pilar.v15i2.658>
- Hang, J., Zhang, D., Chen, P., Zhang, J., & Wang, B. (2019). Classification of plant leaf diseases based on improved convolutional neural network. *Sensors (Switzerland)*, 19(19). <https://doi.org/10.3390/s19194161>

- Jinan, A., & Hayadi, B. H. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). In *Journal of Computer and Engineering Science* (Vol. 1, Issue 2).
- Montalbo, F. J. P., & Hernandez, A. A. (2020). Classifying barako coffee leaf diseases using deep convolutional models. *International Journal of Advances in Intelligent Informatics*, 6(2), 197–209. <https://doi.org/10.26555/ijain.v6i2.495>
- Rozaqi, A. J., Sunyoto, A., & Arief, R. (2021). Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network Detection of Potato Leaves Disease Using Image Processing with Convolutional Neural Network Methods. *Creative Information Technology Journal*, 8(1), 22-31.
- Sarigül, M., Ozyildirim, B. M., & Avci, M. (2019). Differential convolutional neural network. *Neural Networks*, 116, 279–287. <https://doi.org/10.1016/j.neunet.2019.04.025>
- Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 6). Springer. <https://doi.org/10.1007/s42979-021-00815-1>
- Septariani, D. N., Mei Liana, I., & Adi Cahyono, S. (2020). Pengendalian OPT Ramah Lingkungan pada Brokoli Mendukung Good Agricultural Practices. *Jurnal Ilmiah Mahasiswa Agribisnis UNS*, 2020, Pp. 584-596., 4(1).
- Suthakaran, A., & Premaratne, S. (2020). Detection of the affected area and classification of pests using convolutional neural networks from the leaf images. *International Journal of Computer Science Engineering (IJCSE)*, 9(1).
- Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning* (Pp. 6105-6114). PMLR.
- Tobias, R. R., Carlo De Jesus, L., Mital, M. E., Lauguico, S., Guillermo, M., Sybingco, E., Bandala, A., & Dadios, E. (2020). Android Application for Chest X-ray Health Classification from a CNN Deep Learning TensorFlow Model. *LifeTech 2020 - 2020 IEEE 2nd Global Conference on Life Sciences and Technologies*, 255–259. <https://doi.org/10.1109/LifeTech48969.2020.1570619189>

- Uğuz, S., & Uysal, N. (2021). Classification of olive leaf diseases using deep convolutional neural networks. *Neural Computing and Applications*, 33(9), 4133–4149. <https://doi.org/10.1007/s00521-020-05235-5>
- Widiwurjani, Guniarti, & Andansari, P. (2019). Status Kandungan Sulforaphane Microgreens Tanaman Brokoli (*Brassica oleracea* L.) Pada Berbagai Media Tanam Dengan Pemberian Air Kelapa Sebagai Nutrisi. *Jurnal Ilmiah Hijau Cendekia*, ISSN 2548-9372.
- Zou, K., Ge, L., Zhou, H., Zhang, C., & Li, W. (2021). Broccoli seedling pest damage degree evaluation based on machine learning combined with color and shape features. *Information Processing in Agriculture*, 8(4), 505–514. <https://doi.org/10.1016/j.inpa.2020.12.003>