

**KLASIFIKASI PENYAKIT PADA DAUN KELAPA SAWIT  
MENGUNAKAN EFFICIENTNETV2 DAN VGG19 BERBASIS WEBSITE**

**SKRIPSI**

**MARCHELLA STEPHANIE PUTRI AGASKA PARAPAT**

**191401104**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**2024**

**KLASIFIKASI PENYAKIT PADA DAUN KELAPA SAWIT  
MENGUNAKAN EFFICIENTNETV2 DAN VGG19 BERBASIS WEBSITE**

**SKRIPSI**

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

**MARCHELLA STEPHANIE PUTRI AGASKA PARAPAT**

**191401104**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**2024**

**PERSETUJUAN**

Judul : KLASIFIKASI PENYAKIT PADA DAUN KELAPA  
SAWIT MENGGUNAKAN EFFICIENTNETV2 DAN  
VGG19 BERBASIS WEBSITE  
Kategori : SKRIPSI  
Nama : MARCHELLA STEPHANIE PUTRI AGASKA  
PARAPAT  
Nomor Induk Mahasiswa : 191401104  
Program Studi : SARJANA (S-1) ILMU KOMPUTER  
Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERSITAS SUMATERA  
UTARA  
Komisi Pembimbing :

Dosen Pembimbing I

Dosen Pembimbing II

Prof. Dr. Syahril Efendi S.Si., M.I.T.  
NIP. 196711101996021001

Dr. T. Henny Febriana Harumy S.Kom., M.Kom.  
NIP. 198802192019032016

Diketahui/disetujui oleh  
Program Studi SI Ilmu Komputer  
Ketua,

Dr. Amalia ST., M.T.  
NIP. 197812212014042001

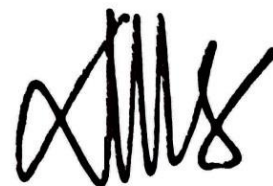
## **PERNYATAAN**

### **KLASIFIKASI PENYAKIT PADA DAUN KELAPA SAWIT MENGGUNAKAN EFFICIENTNETV2 DAN VGG19 BERBASIS WEBSITE**

## **SKRIPSI**

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

Medan, 12 Juni 2024



Marchella Stephanie Putri Agaska Parapat  
191401104

## KATA PENGANTAR

Syukur kepada Tuhan Yang Maha Esa atas rahmat-Nya yang memungkinkan penulis menyelesaikan skripsi ini sebagai salah satu syarat untuk meraih kelulusan di Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu penulis dalam bentuk doa, bimbingan, semangat, dan ilmu selama pembuatan dan penyelesaian skripsi ini.

1. Bapak 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 Universitas Sumatera Utara.
3. Bapak Dr. Mohammad Andri Budiman, S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan 1 Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.
5. Ibu Sri Melvani Hardi, S.Kom., M.Kom. sebagai Sekretaris Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.
6. Bapak Prof. Dr. Syahril Effendi S.Si., M.I.T. selaku Dosen Pembimbing 1 yang telah memberikan banyak motivasi, saran, dan bimbingan kepada penulis hingga skripsi ini dapat diselesaikan.
7. Ibu Dr. T. Henny Febriana Harumy S.Kom., M.Kom. selaku Dosen Pembimbing 2 yang telah memberikan banyak motivasi, saran, dan bimbingan kepada penulis hingga skripsi ini dapat terselesaikan.
8. Seluruh dosen, staf pengajar, dan pegawai di Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
9. Terkhusus kepada ayah terkasih, Agaphe Roma Parapat S.T. dan ibu tercinta, Friska Devita br. Manalu S.T., yang telah mendukung dan memotivasi penulis baik secara moral maupun materi.

10. Teruntuk adik penulis Jeremia Janeigh Genrado Parapat, terima kasih atas dukungan dan semangat juga doa kepada penulis selama pengerjaan skripsi ini.
11. Saudari-saudari penulis, Dian Putri Angelia Sitorus dan Belintawati Zelda Br. Ginting yang selalu mendukung, memotivasi dan menemani penulis.
12. Leonardo Star Berutu, yang selalu mendukung, memotivasi, dan menemani penulis selama pengerjaan skripsi ini.
13. Sahabat-sahabat penulis, Natasya Pakpahan, Jeremi Hutagalung, Meyra Okta Puri, Beatric Sihalohe dan Monica Pasaribu yang selalu mendukung penulis.
14. Teman-teman penulis di perkuliahan, Clara Hulu, Friska Pegrisentia br Sirait, Maristella, Adnia Chiolita, Wahyu Francesco Toldo Hutabarat, Ridho Anshory Nainggolan, Febriando Manik, Eric Martin, Victor Hutapea, dan seluruh mahasiswa Ilmu Komputer USU stambuk 2019.
15. Rekan-rekan kerja penulis di IMILKOM terkhusus Departemen Wawasan Kontemporer IMILKOM yang memberikan penulis pengalaman berorganisasi dalam perkuliahan selama dua periode.
16. Teman-teman sepelayanan penulis di KMKI terkhusus Badan Pengurus Harian KMKI yang selalu melayani Tuhan bersama-sama dengan penulis selama perkuliahan.
17. *Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting.*

Semoga Tuhan Yang Maha Esa memberikan berkah kepada semua pihak yang telah mendukung dalam pengerjaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat bagi individu, keluarga, masyarakat dan negara, serta kemajuan dunia pendidikan.

## ABSTRAK

Kelapa sawit merupakan tanaman komersial serbaguna yang banyak digunakan dalam produksi minyak nabati, minyak industri, dan bahan bakar nabati. Indonesia merupakan produsen kelapa sawit terbesar di dunia, sehingga menjadikannya sebagai peserta penting dalam bisnis kelapa sawit. Meskipun demikian, masih terdapat kesalahan dalam mengidentifikasi penyakit daun kelapa sawit tersebut khususnya bagi masyarakat yang belum memiliki pengetahuan tentang tanaman tersebut. Penelitian dilakukan dengan memanfaatkan metode deep learning dengan model *EfficientNetV2* untuk mengidentifikasikan penyakit daun tanaman kelapa sawit. Penelitian ini dilakukan dengan melakukan beberapa tahapan antara lain pengumpulan dataset, augmentasi data, pembuatan model, pelatihan data, dan pengujian. Hasil penelitian menunjukkan bahwa model yang digunakan mencapai tingkat akurasi pelatihan sebesar 99.74% dan tingkat akurasi validasi sebesar 98.75% pada epoch ke-30. Dengan demikian, sistem klasifikasi penyakit daun kelapa sawit berbasis website yang dikembangkan dalam penelitian ini diharapkan dapat menjadi alat yang bermanfaat dalam mendukung pemahaman terhadap penyakit daun kelapa sawit.

**Kata Kunci :** Klasifikasi Penyakit Daun Kelapa Sawit, *EfficientNetV2*, *VGG19*, Aplikasi Berbasis Web, *Deep Learning*.

**CLASSIFICATION OF OIL PALM DISEASE USING EFFICIENTNETV2 AND  
VGG19 BASED ON WEBSITE**

**ABSTRACT**

*Oil palm is a versatile cash crop that is widely used in the production of edible oils, industrial oils, and biofuels. Indonesia is the world's largest producer of oil palm, making it an important participant in the palm oil business. Nonetheless, there are still errors in identifying oil palm leaf diseases, especially for people who do not have knowledge about the plant. Research was conducted by utilizing the deep learning method with the EfficientNetV2 model to identify leaf diseases of oil palm plants. This research was conducted by performing several stages including dataset collection, data augmentation, model building, data training, and testing. The results showed that the model used achieved a training accuracy rate of 99.74% and a validation accuracy rate of 98.75% at the 30th epoch. Thus, the web-based palm leaf disease classification system developed in this study is expected to be a useful tool in supporting the understanding of palm leaf disease.*

**Keywords:** *Palm Leaf Disease Classification, EfficientNetV2, VGG19, Web-based Application, Deep Learning.*



## DAFTAR ISI

PERSETUJUAN .....	iii
PERNYATAAN .....	iv
KATA PENGANTAR .....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiv
BAB 1 PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah .....	3
1.4    Tujuan Penelitian .....	4
1.5    Manfaat Penelitian .....	4
1.6    Penelitian Relevan .....	4
1.7    Metodologi Penelitian.....	5
1.8    Sistematika Penulisan .....	6
BAB 2 LANDASAN TEORI.....	7
2.1    Daun Kelapa Sawit .....	7
2.2 <i>Machine Learning</i> .....	11
2.3 <i>Deep Learning</i> .....	12
2.4 <i>Convolutional Neural Network (CNN)</i> .....	12
2.4.1    Convolutional Layer.....	13
2.4.2    Pooling Layer .....	13
2.4.3    Fully Connected Layer .....	14

2.4.4	Softmax .....	14
2.4.5	ReLU .....	15
2.5	<i>Transfer Learning</i> .....	15
2.6	<i>EfficientNetV2</i> .....	15
2.7	<i>VGG19</i> .....	18
2.8	<i>Website</i> .....	18
2.9	<i>Application Programming Interface (API)</i> .....	19
BAB 3 ANALISIS DAN PERANCANGAN SISTEM .....		20
3.1	Analisis Sistem.....	20
3.1.1	Analisis Masalah .....	20
3.1.2	Analisis Kebutuhan .....	21
3.1.3	Arsitektur Umum Sistem.....	22
3.2	Pengambilan Dataset.....	23
3.3	Pra-pemrosesan Data .....	23
3.3.1	Augmentasi Data .....	23
3.3.2	Pemisahan Data Training, Validasi dan Testing .....	24
3.4	Pemodelan Sistem.....	24
3.4.1	Use Case Diagram.....	24
3.4.2	Activity Diagram.....	25
3.4.3	Sequence Diagram.....	26
3.5	Pelatihan Model .....	27
3.6	Evaluasi dan Pengujian Model.....	27
3.7	<i>Flowchart</i> .....	28
3.7.1	Flowchart Sistem.....	28
3.7.2	Flowchart Arsitektur EfficientNetV2.....	29
3.7.3	Flowchart Arsitektur VGG19.....	30
3.8	Perancangan Antarmuka .....	31

3.8.1	Halaman Identifikasi .....	31
3.8.2	Halaman Hasil Identifikasi .....	31
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM.....		33
4.1	Implementasi Sistem .....	33
4.1.1	Perangkat Keras.....	33
4.1.2	Perangkat Lunak.....	33
4.2	Implementasi Dataset .....	34
4.3	Implementasi Model .....	34
4.3.1	Implementasi Model EfficientNetV2 .....	35
4.3.2	Implementasi Model VGG19 .....	36
4.4	Implementasi Tahap Evaluasi dan Pengujian Model .....	39
4.5	Hasil Perbandingan Arsitektur .....	44
4.6	Hasil Pengujian Sistem .....	45
4.7	Implementasi Pada Website .....	47
BAB 5 PENUTUP .....		48
5.1	Kesimpulan .....	48
5.2	Saran .....	48
DAFTAR PUSTAKA .....		xv

## DAFTAR GAMBAR

<b>Gambar 1.1</b> Metodologi Penelitian .....	5
<b>Gambar 2.1</b> Daun Kelapa Sawit Sehat .....	7
<b>Gambar 2.2</b> Penyakit bercak daun <i>brown spot</i> .....	8
<b>Gambar 2.3</b> Penyakit bercak daun <i>white scale</i> .....	10
<b>Gambar 2.4</b> Arsitektur CNN (Sumber: <a href="http://www.mathworks.com">www.mathworks.com</a> ) .....	12
<b>Gambar 2.5</b> Proses Konvolusi (Nanang Kasim, 2021) .....	13
<b>Gambar 2.6</b> <i>Max Pooling</i> dan <i>Average Pooling</i> (Rinku Datta Rakshit, 2021) .....	13
<b>Gambar 2.7</b> <i>Fully Connected Layer</i> (Arvind Sreenivas, 2020) .....	14
<b>Gambar 2.8</b> Softmax (Sumber: <a href="http://towardsdatascience.com">towardsdatascience.com</a> ) .....	14
<b>Gambar 2.9</b> Fungsi Aktivasi ReLU (Tahirou Djara, 2024) .....	15
<b>Gambar 2.10</b> Arsitektur <i>EfficientNetV2</i> .....	16
<b>Gambar 2.11</b> Arsitektur VGG19 (Imen Chebbi, 2021) .....	18
<b>Gambar 2.12</b> Ilustrasi API (Sumber: <a href="http://www.leadmine.net">www.leadmine.net</a> ) .....	19
<b>Gambar 3.1</b> Analisis Sistem .....	20
<b>Gambar 3.2</b> <i>Fishbone Diagram</i> .....	21
<b>Gambar 3.3</b> Arsitektur Umum Sistem .....	22
<b>Gambar 3.4</b> <i>Use Case Diagram</i> .....	25
<b>Gambar 3.5</b> <i>Activity Diagram</i> .....	26
<b>Gambar 3.6</b> <i>Sequence Diagram</i> .....	26
<b>Gambar 3.7</b> <i>Flowchart</i> / Proses Kerja Sistem .....	28
<b>Gambar 3.8</b> <i>Flowchart</i> Arsitektur <i>EfficientNetV2</i> .....	29
<b>Gambar 3.9</b> <i>Flowchart</i> Arsitektur VGG19 .....	30
<b>Gambar 3.10</b> Gambar Rangka Halaman Klasifikasi .....	31
<b>Gambar 3.11</b> Gambar Rangka Halaman Hasil Klasifikasi .....	32
<b>Gambar 4.1</b> Jumlah <i>Dataset</i> Tiap Folder .....	34
<b>Gambar 4.2</b> <i>Summary Model</i> <i>EfficientNetV2</i> .....	35
<b>Gambar 4.3</b> <i>Summary Model</i> VGG19 .....	36
<b>Gambar 4.4</b> Grafik <i>Training dan Validation Accuracy EfficientNetV2</i> .....	40
<b>Gambar 4.5</b> Grafik <i>Training and Validation Accuracy VGG19</i> .....	40
<b>Gambar 4.6</b> Grafik <i>Training and Validation Loss EfficientNetV2</i> .....	41
<b>Gambar 4.7</b> Grafik <i>Training and Validation Loss VGG19</i> .....	41

<b>Gambar 4.8</b> <i>Confusion Matrix EfficientNetV2</i> .....	42
<b>Gambar 4.9</b> <i>Confusion Matrix VGG19</i> .....	43
<b>Gambar 4.10</b> Tampilan keluaran model <i>EfficientNetV2</i> .....	46
<b>Gambar 4.11</b> Tampilan keluaran model <i>VGG19</i> .....	46
<b>Gambar 4.12</b> Halaman Identifikasi.....	47
<b>Gambar 4.13</b> Halaman Hasil Klasifikasi dengan gambar .....	47

**DAFTAR TABEL**

<b>Tabel 3.1</b> Tabel Pelatihan Model <i>EfficientNetV2</i> .....	27
<b>Tabel 3.2</b> Tabel Pelatihan Model <i>VGG19</i> .....	27
<b>Tabel 4.1</b> Metrik Evaluasi Model.....	44
<b>Tabel 4.2</b> Perbandingan <i>EfficientNetV2</i> dan <i>VGG19</i> .....	44
<b>Tabel 4.3</b> Daftar Hasil Percobaan .....	45

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kelapa sawit merupakan tanaman komersial serbaguna yang banyak digunakan dalam produksi minyak nabati, minyak industri, dan bahan bakar nabati. Indonesia merupakan produsen kelapa sawit terbesar di dunia, sehingga menjadikannya sebagai peserta penting dalam bisnis kelapa sawit. Kelapa sawit merupakan tanaman serbaguna dengan berbagai macam kegunaan. Buah kelapa sawit dapat diolah menjadi minyak yang dapat dimakan dan tidak dapat dimakan, sementara sabut, cangkang, pelepah, dan daunnya juga memiliki berbagai kegunaan.

Di Indonesia, produksi minyak kelapa sawit telah mengalami pertumbuhan yang signifikan, mencapai total produksi 48,68 juta ton pada tahun 2018, yang mencakup minyak kelapa sawit dan inti sawit. Perkebunan kelapa sawit di Indonesia tersebar di 26 provinsi, dengan Riau sebagai provinsi yang memiliki perkebunan terluas, yaitu 2,89 juta hektar. Sektor kelapa sawit di Indonesia secara efektif mengelola interaksi antara faktor sosial, ekonomi, dan lingkungan, dan telah muncul sebagai komoditas pertanian yang penting dalam pertumbuhan ekonomi negara.

Tanaman kelapa sawit merupakan tanaman yang sangat rentan terserang Organisme Pengganggu Tumbuhan (OPT), salah satu penghalang yang signifikan bagi tanaman untuk mencapai potensi produksi maksimumnya adalah adanya penyakit daun kelapa sawit. Penyakit daun kelapa sawit merupakan masalah yang umum terjadi di perkebunan kelapa sawit. Penyakit-penyakit ini memiliki potensi untuk menyebabkan kerusakan pada bibit yang dibudidayakan, yang mengakibatkan kerusakan yang signifikan dan potensi kematian pada tanaman. Metode standar yang biasa digunakan oleh petani untuk mengidentifikasi jenis penyakit daun tertentu pada pohon kelapa sawit adalah dengan melakukan pengamatan langsung terhadap gejala yang ditunjukkan oleh daun. Pendekatan ini kurang efisien karena tidak semua orang memiliki kemampuan untuk mengidentifikasi jenis penyakit tertentu yang menyerang daun kelapa sawit

melalui pengamatan langsung (Pribadi & Kurniawan, 2022). Oleh karena itu, sangat penting untuk mengidentifikasi dan mengkategorikan penyakit tanaman secara akurat.

Untuk mengatasi masalah ini, peneliti membangun sebuah sistem yang bertujuan untuk mendeteksi dan mengidentifikasi penyakit pada daun kelapa sawit dengan cepat, yang merupakan langkah penting untuk pengendalian tanaman yang efisien. Penelitian ini mengusulkan model kerja yang dibangun dengan arsitektur deep transfer learning *EfficientNetV2* dan *VGG19* untuk mengklasifikasikan dan mendeteksi penyakit pada daun kelapa sawit. Aplikasi deep learning berbasis website yang dapat mendeteksi daun yang sakit melalui citra gambar lebih cerdas dan akurat serta menghemat waktu jika dibandingkan dengan deteksi penyakit secara manual di setiap tanaman. Dengan menggunakan sistem deteksi penyakit yang cepat dan tepat, memungkinkan petani untuk menerapkan tindakan pencegahan yang sesuai dan berfungsi sebagai dasar untuk merumuskan kebijakan pengendalian penyakit yang efektif.

*EfficientNet* adalah kumpulan struktur jaringan saraf yang dibuat dengan tujuan untuk mencapai keseimbangan ideal antara ukuran model dan kinerjanya. *EfficientNetV2* adalah salah satu arsitektur Convolutional Neural Network (CNN) yang menunjukkan waktu pelatihan dan efisiensi parameter yang unggul dibandingkan dengan model *EfficientNetV1* sebelumnya. Penelitian ini menggunakan model *EfficientNetV2* dengan menggunakan *TensorFlow* (Kusumastuti & Sunyoto, 2023). *EfficientNetV2* ini juga telah banyak diterapkan dalam klasifikasi dan mendeteksi penyakit pada citra daun seperti penelitian yang dilakukan pada tahun 2023 yang berjudul “*EfficientNetV2 Model for Plant Disease Classification and Pest Recognition*” oleh R. S. Sandhya Devi<sup>1</sup>, V. R. Vijay Kumar dan P. Sivakumar dengan hasil akurasi yang baik yaitu sebesar 99.5%, 97.5%, dan 80.1% pada tiga dataset yang berbeda.

*VGG19* salah satu arsitektur *Convolutional Neural Network* (CNN) yang terdiri dari 19 lapisan. Jaringan ini memiliki kemampuan untuk mengklasifikasikan foto ke dalam 1000 kategori objek yang berbeda. Model *VGG19* adalah versi modifikasi dari model *VGG* yang terdiri dari 19 lapisan, termasuk 16 lapisan konvolusi, 3 lapisan yang terhubung penuh, 5 lapisan penyatuan maksimal, dan 1 lapisan softmax. Varian tambahan dari model *VGG* termasuk *VGG11*, *VGG16*, dan lainnya (Kobra et al., 2022). *VGG19* ini juga telah banyak digunakan untuk klasifikasi dan mendeteksi penyakit pada citra daun seperti penelitian yang dilakukan oleh Sri Adi Pavan Naidu Kavala dan



Rajarajeswari Pothuraju pada tahun 2022 yang berjudul “*Detection Of Grape Leaf Disease Using Transfer Learning Methods: VGG16 & VGG19*” menunjukkan akurasi sebesar 98%.

Berdasarkan penelitian terdahulu, pada penelitian ini peneliti melakukan klasifikasi dan deteksi penyakit kelapa sawit melalui citra daun dengan pengimplementasian arsitektur *deep transfer learning* yaitu *EfficientNetV2* dan *VGG19*. Yang mana pada sistem ini nanti, pengguna dapat menggunakan website dengan gambar yang sudah diambil terlebih dahulu.

## **1.2 Rumusan Masalah**

Kelapa sawit merupakan salah satu tanaman yang penting bagi sebagian masyarakat Indonesia. Petani kelapa sawit akan mengalami kerugian apabila tanaman kelapa sawit sakit dan masa panen terhambat. Jika petani melakukan pengecekan daun kelapa sawit satu per satu secara manual untuk melihat penyakitnya maka itu akan membutuhkan waktu yang sangat lama. Oleh karena itu, petani membutuhkan aplikasi yang dapat mengidentifikasi penyakit tanaman kelapa sawit sehingga berpotensi menyelamatkan tanaman kelapa sawit sebelum mengalami kerusakan yang tidak dapat diperbaiki.

## **1.3 Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. Penyakit yang diangkat adalah penyakit bercak daun berupa *white scale* dan *brown spot*.
- b. Bahasa yang digunakan dalam membangun program dan model adalah bahasa pemrograman *Python*.
- c. Model yang digunakan yaitu *EfficientNetV2* dan *VGG19* dengan membandingkan kinerja dari dua model tersebut.
- d. Sistem diimplementasikan pada aplikasi berbasis *website*.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun sebuah aplikasi berbasis *website* yang dapat melakukan pendeteksian penyakit pada daun kelapa sawit melalui citra daun kelapa sawit dengan menggunakan model *EfficientNetV2* dan *VGG19*.

## 1.5 Manfaat Penelitian

Pada penelitian ini diharapkan dapat bermanfaat bagi pengguna/petani kelapa sawit agar dapat lebih mengenal jenis penyakit pada daun kelapa sawit dengan lebih baik dan dapat menentukan prosedur penyembuhan dari penyakit tersebut dengan lebih cepat.

## 1.6 Penelitian Relevan

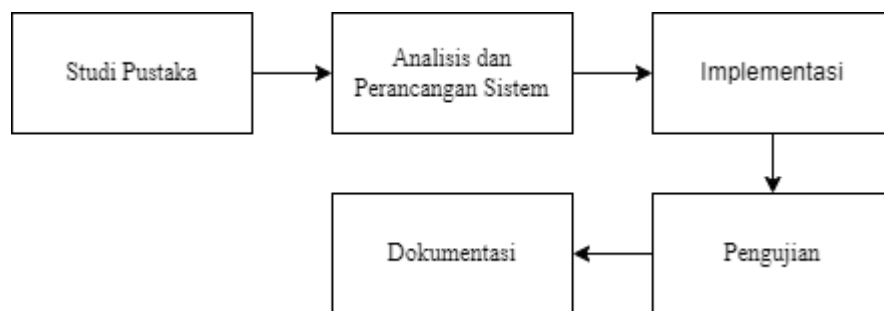
Berikut adalah penelitian-penelitian terdahulu yang relevan dengan yang dilakukan antara lain:

1. Penelitian (R. S. Sandhya Devi, V. R. Vijay Kumar dan P. Sivakumar, 2023) “*EfficientNetV2 Model for Plant Disease Classification and Pest Recognition*”. Temuan penelitian menunjukkan kecepatan dan akurasi pelatihan *EfficientNetV2* sebesar 99,5%, 97,5% dan 80,1% pada 3 dataset berbeda.
2. Penelitian (Zhanhao Shi, Cui Wang dan Lin Zhao, 2022) “*Tomato Disease Identification Application Based on EfficientNetV2*”. Temuan penelitian mengidentifikasi 9 penyakit tomat secara akurat dengan nilai akurasi mencapai 0,98 atau 98%.
3. Penelitian (Sunil C. K., Jaidhar C. D., dan Nagamma Patil, 2021) “*Cardamom Plant Disease Detection Approach Using EfficientNetV2*”. Temuan penelitian menunjukkan model *EfficientNetV2-L* mencapai akurasi deteksi maksimum sebesar 98,26% dan 96,45% pada 2 dataset berbeda.
4. Penelitian (Sri Adi Pavan Naidu Kavala dan Rajarajeswari Pothuraju, 2022) “*Detection Of Grape Leaf Disease Using Transfer Learning Methods: VGG16 & VGG19*”. Temuan Penelitian menunjukkan akurasi yang tinggi pada dataset validasi dan pengujian, dengan *VGG16* mencapai 98,09% pada validasi dan 98,75% pada

data pengujian, dan *VGG19* mencapai 97,41% pada validasi dan 97,25% pada data pengujian.

5. Penelitian (Abhishek Rajput dan Unmukh Datta, 2023) yang berjudul “*A Novel Deep Learning Design of Plant Disease Recognition and Detection Using VGG19, and DenseNet121*”. Temuan penelitian menunjukkan DenseNet-121 mencapai akurasi pelatihan yang lebih tinggi yaitu 97,57% dan akurasi validasi sebesar 99,29%. Sedangkan *VGG19* yang mencapai akurasi pelatihan sebesar 85,06% dan akurasi validasi sebesar 88,06%.

### 1.7 Metodologi Penelitian



**Gambar 1.1** Metodologi Penelitian

#### 1. Studi Pustaka

Pada tahap ini penelitian dimulai dengan mencari referensi dari berbagai sumber terpercaya dan melakukan peninjauan pustaka melalui buku-buku, jurnal, *e-book*, artikel ilmiah, makalah ataupun situs internet yang berhubungan dengan tanaman kelapa sawit, *EfficientNetV2* dan *VGG19*.

#### 2. Analisa dan Perancangan Sistem

Berdasarkan ruang lingkup penelitian, penulis melakukan analisa terhadap apa saja yang dibutuhkan dalam penelitian untuk segera dirancang dalam sebuah diagram alir (*flowchart*).

#### 3. Implementasi

Pada tahap ini, penulis menggunakan model *EfficientNetV2* dan *VGG19* yang menggunakan bahasa pemrograman *Python* sesuai dengan *flowchart* yang telah dirancang.

#### 4. Pengujian

Pada tahap ini, sistem yang telah dirancang dilakukan uji coba dan evaluasi dalam mendeteksi penyakit terhadap *dataset* daun kelapa sawit yang telah dikumpulkan untuk mengetahui apakah sistem berjalan sesuai tujuan.

#### 5. Dokumentasi

Pada tahap ini dilakukan dokumentasi mulai dari tahap analisis sistem sampai kepada pengujian sistem dalam format penulisan berbentuk skripsi.

### 1.8 Sistematika Penulisan

Berikut merupakan sistematika penulisan skripsi ini, yang terbagi ke dalam beberapa bagian utama, sebagaimana dijelaskan berikut ini.

#### **BAB 1        PENDAHULUAN**

Bab ini memuat uraian tentang segala latar belakang informasi yang berkenaan dengan pokok bahasan yang akan diteliti, meliputi rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

#### **BAB 2        LANDASAN TEORI**

Tinjauan teoritis yang berkaitan dengan *EfficientNetV2*, *VGG19* dan penyakit Daun Kelapa Sawit dan juga yang berhubungan dengan penelitian dibahas pada bab ini.

#### **BAB 3        ANALISIS PERANCANGAN**

Melakukan analisis terhadap masalah dan sistem yang dibangun akan dibahas pada bab ini. Setelah itu, kita akan melanjutkan dengan desain sistem, dengan menggunakan model *EfficientNetV2* dan *VGG19*.

#### **BAB 4        IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab ini membahas implementasi dan pengujian sistem menggunakan tahapan analisis dan desain yang telah dibahas sebelumnya.

#### **BAB 5        PENUTUP**

Kesimpulan dan Saran dari penelitian yang telah selesai dimasukkan dalam bab ini untuk kemajuan penelitian di masa depan.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Daun Kelapa Sawit**

Kelapa sawit adalah anggota dari keluarga Arecaceae. Budidaya kelapa sawit berasal dari Amerika Selatan. Produksi kelapa sawit merupakan komponen utama dari bisnis petani, karena tanaman ini berfungsi sebagai komoditas utama. Kelapa sawit sangat bernilai dalam bisnis perkebunan sebagai alternatif kelapa dalam produksi minyak. Perkebunan kelapa sawit biasanya terletak di lokasi hutan terpencil karena kebutuhan lahan yang cukup besar, yang mengharuskan adanya jarak dengan aktivitas desa. Karena keuntungan ekonomi yang terkait dengan produk turunan kelapa sawit, para petani telah beralih untuk membudidayakan komoditas kelapa sawit di beberapa pulau di Indonesia, seperti Kalimantan, Sumatra, dan Sulawesi (Rahmawati, 2023).

Minyak kelapa sawit masih menjadi komoditas utama untuk memenuhi kebutuhan minyak nabati global. Minyak kelapa sawit menyumbang 36% dari produksi minyak nabati global, melampaui komoditas lain seperti kedelai, bunga matahari, dan rapa (OECD/FAO, 2021). Indonesia, negara dengan perkebunan kelapa sawit terbesar di dunia, memproduksi sekitar 55% dari total minyak kelapa sawit. Proporsi ini dapat ditingkatkan lebih lanjut dengan meminimalkan kerugian dalam produksi minyak kelapa sawit, terutama di perkebunan yang dikelola oleh masyarakat (Priwiratama et al., 2023).



**Gambar 2.1** Daun Kelapa Sawit Sehat

Daun kelapa sawit yang sehat memiliki beberapa ciri fisik yang dapat diidentifikasi. Berikut adalah beberapa ciri-ciri daun kelapa sawit yang sehat:

1. Daunnya hijau berkilat, tidak ada warna kusam yang terjangkit pada daunnya. Tidak ada bercak-bercak kuning atau zat lain di permukaan daun-daun pohonnya.
2. Setiap bulan keluar pucuk pelepah. Tahap ini terjadi secara berkala. Jika tidak ada pelepah yang muncul, kemungkinan pohon kelapa sawit mengalami gangguan. Kelenturan pelepah lebih baik, daun muda yang keluar lebih besar daripada daun tua dan tidak pernah mengalami daun tombak.
3. Lebar dan panjang pelepah normal dan tidak mengecil. Pelepah sering menjadi tanda ciri fisik tanaman yang sehat. Jika bentuknya berukuran biasa dan tidak ada pengecilan ukuran maka dapat dipastikan tanaman kelapa sawit adalah sehat dan tidak kekurangan nutrisi.
4. Pembentukan bunga dan buah sebanyak 12-24 janjang per tahun. Jika kurang dari jumlah yang sudah disebutkan maka sudah pasti ada yang salah dengan kesuburan tanamannya.
5. Pohon yang sehat adalah yang terbebas dari serangan hama penyakit.

Terdapat dua penyakit pada daun kelapa sawit yang diklasifikasikan dalam penelitian ini:



**Gambar 2.2** Penyakit bercak daun *brown spot*

Penyakit bercak coklat (*brown spot*) pada daun kelapa sawit adalah salah satu penyakit yang disebabkan oleh jamur. Penyakit ini dapat mengganggu pertumbuhan dan produktivitas tanaman kelapa sawit jika tidak ditangani dengan baik. Berikut penjelasan lebih lanjut mengenai penyakit ini:

1. Penyakit bercak coklat pada daun kelapa sawit umumnya disebabkan oleh jamur *Curvularia spp.* dan *Helminthosporium spp.* Jamur ini tumbuh subur pada kondisi yang lembab dan seringkali menyerang daun-daun yang lebih tua.
2. Gejala
  - Munculnya bercak-bercak kecil berwarna coklat pada daun kelapa sawit. Bercak ini dapat berdiameter beberapa milimeter hingga beberapa sentimeter.

- Bercak biasanya berbentuk bulat atau oval dan dapat menyatu membentuk bercak yang lebih besar.
- Bercak dapat menyebar dengan cepat pada kondisi cuaca yang lembab dan basah.
- Daun yang terinfeksi berat dapat mengalami nekrosis atau kematian jaringan, yang menyebabkan bagian daun mengering dan mati.

### 3. Dampak

- Daun yang terkena penyakit ini akan mengalami penurunan kemampuan fotosintesis, sehingga mengganggu pertumbuhan tanaman secara keseluruhan.
- Jika serangan jamur cukup parah, produksi buah kelapa sawit dapat menurun secara signifikan.
- Tanaman yang terinfeksi berat akan terlihat kurang sehat dan lebih rentan terhadap serangan penyakit lain serta hama.

### 4. Pengendalian

- Menjaga kebersihan area sekitar tanaman dengan menghilangkan daun-daun yang terinfeksi serta sisa-sisa tanaman yang dapat menjadi sumber infeksi.
- Penggunaan fungisida yang sesuai dapat membantu mengendalikan penyebaran jamur penyebab penyakit ini. Fungisida yang mengandung bahan aktif seperti *mancozeb*, *chlorothalonil*, atau *tridemorph* sering digunakan.
- Mengatur kelembaban dan memastikan sirkulasi udara yang baik di sekitar tanaman untuk mengurangi kondisi yang mendukung pertumbuhan jamur.
- Pemberian pupuk yang seimbang dapat membantu meningkatkan kekebalan tanaman terhadap penyakit.

### 5. Pencegahan

- Menanam varietas kelapa sawit yang lebih tahan terhadap serangan penyakit bercak coklat.
- Melakukan pemangkasan daun secara berkala untuk mencegah penumpukan kelembaban yang dapat memicu pertumbuhan jamur.

Dengan langkah-langkah pengendalian dan pencegahan yang tepat, serangan penyakit bercak coklat pada daun kelapa sawit dapat diminimalisir, sehingga produktivitas tanaman tetap optimal.



**Gambar 2.3** Penyakit bercak daun *white scale*

Penyakit sisik putih (*white scale*) pada daun kelapa sawit adalah masalah umum yang disebabkan oleh serangan serangga dari keluarga *Diaspididae*, khususnya spesies seperti *Aspidiotus destructor*. Serangga ini dikenal sebagai kutu sisik putih dan dapat menyebabkan kerusakan serius pada tanaman kelapa sawit. Berikut penjelasan lebih lanjut mengenai penyakit ini:

1. Penyakit sisik putih disebabkan oleh serangan kutu sisik putih. Serangga ini memiliki tubuh kecil, pipih, dan biasanya berwarna putih atau kekuningan. Mereka menempel pada daun dan batang tanaman, menghisap getah tanaman, dan mengeluarkan zat lilin yang melindungi tubuh mereka.
2. Gejala
  - Munculnya bercak-bercak putih pada permukaan daun dan batang kelapa sawit. Bercak ini sebenarnya adalah kutu sisik putih yang menempel pada tanaman.
  - Daun yang terserang akan menunjukkan gejala menguning dan dapat mengering jika serangan kutu sisik putih cukup parah.
  - Pertumbuhan tanaman dapat terhambat akibat serangan kutu yang menghisap nutrisi dari tanaman.
  - Tanaman yang terserang berat mungkin mengalami penurunan produksi buah.
3. Dampak
  - Serangan kutu sisik putih menyebabkan stres pada tanaman, mengurangi kemampuan fotosintesis, dan mengurangi vigor tanaman.
  - Infestasi yang tidak ditandangi dapat menyebabkan kerusakan jangka panjang, membuat tanaman lebih rentan terhadap penyakit dan hama lain.
  - Serangan kutu sisik putih yang parah dapat mengakibatkan penurunan hasil panen secara signifikan.



#### 4. Pengendalian

- Melakukan pengamatan rutin untuk mendeteksi dini adanya serangan kutu sisik putih sehingga tindakan pengendalian dapat segera diambil.
- Aplikasi insektisida yang tepat dapat membantu mengendalikan populasi kutu sisik putih. Insektisida yang mengandung bahan aktif seperti *imidacloprid*, *chlorpyrifos*, atau *buprofezin* sering digunakan.
- Menggunakan musuh alami kutu sisik putih seperti kumbang pemangsa (*ladybugs*) dan parasit tawon yang dapat membantu mengurangi populasi kutu secara alami.
- Menjaga kebersihan kebun dengan memotong dan membuang bagian tanaman yang terinfeksi kutu sisik putih.

#### 5. Pencegahan

- Memilih varietas kelapa sawit yang lebih tahan terhadap serangan kutu sisik putih.
- Mengatur jarak tanam dan melakukan pemangkasan untuk memastikan sirkulasi udara yang baik dan mengurangi kelembaban yang mendukung pertumbuhan kutu.
- Rotasi penggunaan pestisida untuk mencegah resistensi kutu terhadap insektisida.

Dengan penerapan langkah-langkah pengendalian dan pencegahan yang tepat, infestasi kutu sisik putih pada daun kelapa sawit dapat dikendalikan, sehingga produktivitas tanaman tetap terjaga dan hasil panen dapat dioptimalkan.

## 2.2 *Machine Learning*

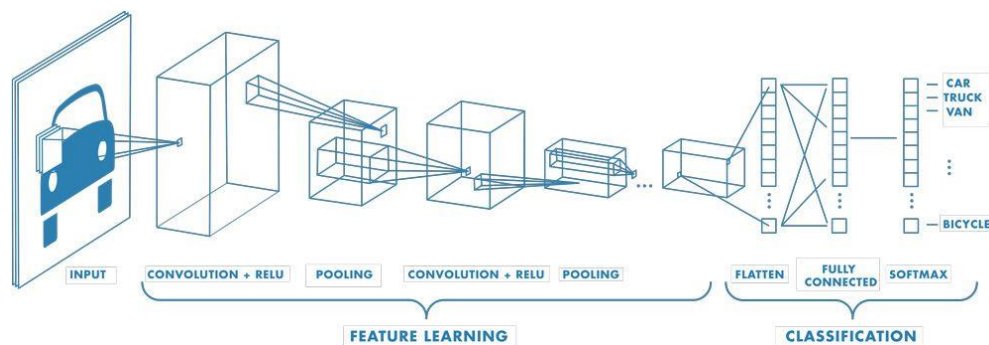
*Machine learning* adalah salah satu bagian dari *artificial intelligence* yang memungkinkan mesin belajar dari data atau pengalaman masa lalu (data historis) sehingga tidak perlu diprogram secara manual untuk melakukan seluruh pekerjaan. Dengan kemampuan belajar secara otomatis, sistem dapat secara bertahap terus belajar meningkatkan akurasi. Data menjadi objek penting dalam membuat machine learning. Tanpa data, machine tidak mendapatkan pengetahuan apa pun yang dapat dipelajari (Taufiq Dimas, 2023).

### 2.3 Deep Learning

Salah satu terobosan terbesar dalam dunia *machine learning* adalah deep learning. Deep learning adalah metode dalam kecerdasan buatan (AI) yang mengajarkan komputer untuk memproses data dengan cara yang terinspirasi oleh otak manusia. Model pembelajaran mendalam dapat mendeteksi pola kompleks dalam gambar, teks, audio, dan data lainnya untuk menghasilkan wawasan dan prediksi yang akurat (Taufiq Dimas, 2023).

### 2.4 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) adalah salah satu kelas *deep feed-forward artificial neural network* yang banyak diaplikasikan pada analisis citra. CNN terdiri atas satu lapis masukan (*input layer*), suatu lapis keluaran (*output layer*), dan sejumlah lapis tersembunyi (*hidden layer*) (Suyanto, 2022).

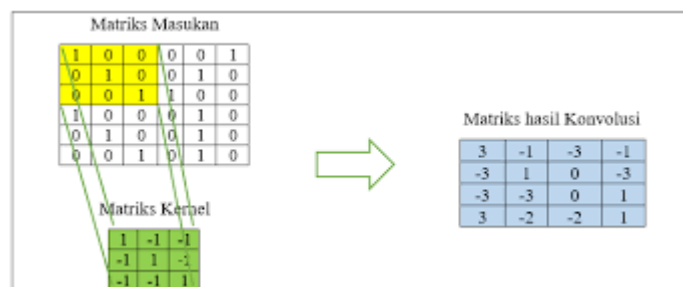


**Gambar 2.4** Arsitektur CNN (Sumber: [www.mathworks.com](http://www.mathworks.com))

Berdasarkan Gambar 2.4 struktur CNN terdiri dari masukan, proses ekstraksi fitur, proses klasifikasi dan keluaran. Proses ekstraksi dalam CNN terdiri dari beberapa *hidden layer* yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan *pooling*. CNN bekerja secara hierarki, sehingga *output* pada lapisan konvolusi pertama akan digunakan sebagai input pada konvolusi selanjutnya. Pada proses klasifikasi terdiri dari *fully connected layer* dan *softmax* yang keluarannya berupa hasil klasifikasi (Melisa, 2021).

### 2.4.1 Convolutional Layer

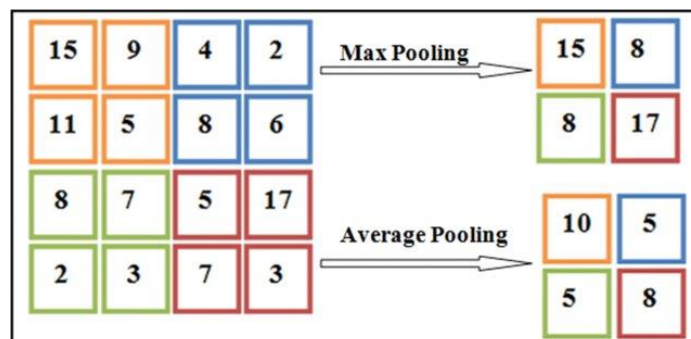
*Convolutional layer* adalah blok utama CNN. *Convolutional layer* berisi satu set *filter* (kernel) dan parameter yang harus dipelajari selama pelatihan. Ukuran *filter* biasanya lebih kecil dari gambar sebenarnya. Setiap *filter* menyatu dengan gambar dan membuat peta aktivasi. Untuk konvolusi, *filter* meluncue melintasi tinggi dan lebar gambar dan *dot product* antara setiap elemen filter dan input dihitung pada setiap posisi spasial (Mostafa & Wu, 2021).



**Gambar 2.5** Proses Konvolusi (Nanang Kasim, 2021)

### 2.4.2 Pooling Layer

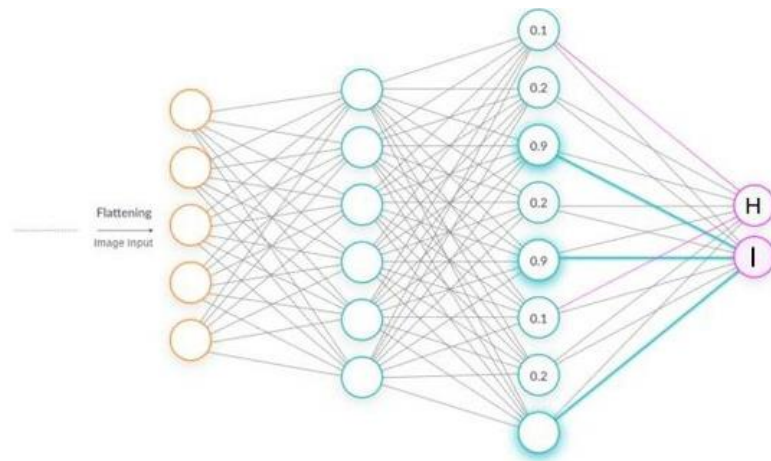
*Pooling layer* beroperasi dengan premis yang sama dengan lapisan konvolusi. Perbedaannya terletak pada fakta bahwa filter di *pooling layer* tidak digunakan untuk deteksi fitur, tetapi lebih untuk mengeksekusi tindakan seperti mengidentifikasi nilai maksimum (*Max Pooling*) atau nilai rata-rata (*Average Pooling*) di wilayah tertentu.



**Gambar 2.6** *Max Pooling* dan *Average Pooling* (Rinku Datta Rakshit, 2021)

### 2.4.3 Fully Connected Layer

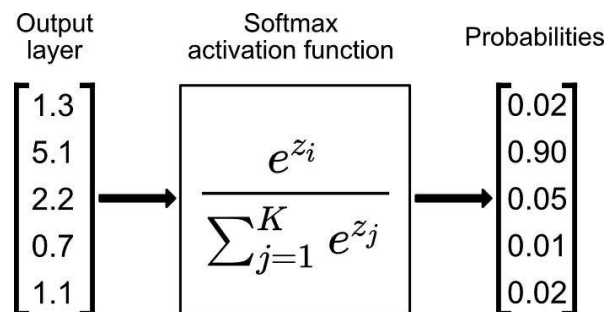
*Fully connected layer* adalah lapisan di mana setiap neuron pada lapisan sebelumnya terhubung dengan setiap neuron pada lapisan berikutnya. Sebelum terhubung ke semua neuron di *fully connected layer*, aktivitas apa pun dari lapisan sebelumnya perlu diubah menjadi data satu dimensi. *Fully connected layer* adalah lapisan yang terdiri dari beberapa node.



**Gambar 2.7** *Fully Connected Layer* (Arvind Sreenivas, 2020)

### 2.4.4 Softmax

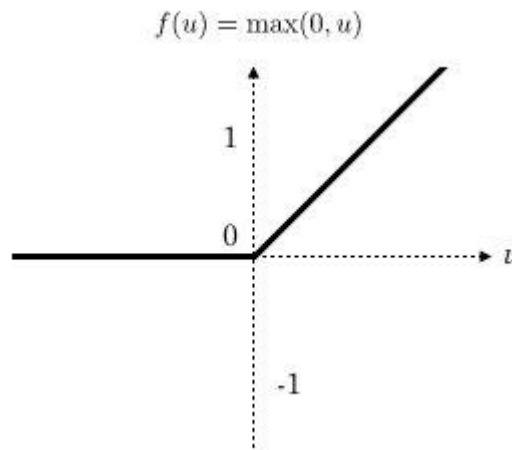
Fungsi aktivasi *softmax* mengubah output dari perhitungan *fully connected layer* menjadi distribusi probabilitas yang mewakili kemungkinan setiap kelas. Fungsi aktivasi *softmax* digunakan dalam tugas klasifikasi yang melibatkan beberapa kelas.



**Gambar 2.8** Softmax (Sumber: towardsdatascience.com)

### 2.4.5 ReLU

*Rectified Linear Units* (ReLU) layer memfasilitasi pelatihan data yang lebih cepat dan lebih efisien dengan mengubah nilai negatif menjadi nol dan mempertahankan nilai positif. Lapisan ini mengimplementasikan fungsi aktivasi.



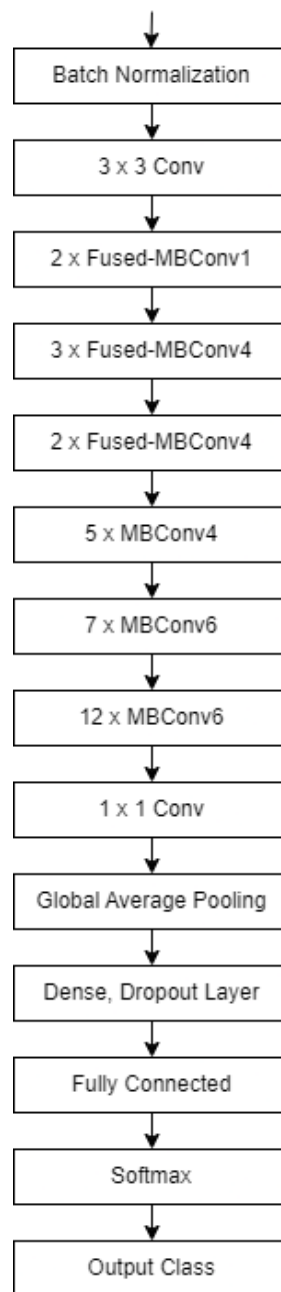
**Gambar 2.9** Fungsi Aktivasi ReLU (Tahirou Djara, 2024)

## 2.5 Transfer Learning

*Transfer learning* merupakan teknik yang memanfaatkan model yang sudah dilatih sebelumnya (*pretrained model*) dalam hal ini adalah model CNN untuk digunakan mengklasifikasikan *dataset* yang baru (Rochman & Junaedi, 2020). *Pretrained model* yang digunakan pada penelitian ini adalah *EfficientNetV2* dan *VGG19*. *Pretrained model* digunakan untuk melakukan *feature learning* pada *dataset* sebelum dimasukkan ke lapisan klasifikasi.

## 2.6 EfficientNetV2

*EfficientNetV2* adalah model CNN dengan kecepatan pelatihan yang lebih cepat dan efisiensi parameter yang lebih baik daripada model sebelumnya yaitu *EfficientNet* yang mana menggunakan *training-aware neural architecture search* dan *scaling*. Arsitektur utama dari model *EfficientNetV2* dapat dilihat pada Gambar 2.10.



**Gambar 2.10** Arsitektur *EfficientNetV2*

Pada Gambar 2.10 terlihat bahwa model memperkenalkan beberapa blok arsitektur baru yang memberikan peningkatan efisiensi dan kinerja dibandingkan *EfficientNet*, seperti blok MBConv (*Mobile Inverted Residual Bottleneck Convolution*), Fused-MBConv (*Fused Mobile Inverted Residual Bottleneck Convolution*), dan SE (*Squeeze-and-Excitation*) yang dibuat untuk meningkatkan efisiensi dan representasi fitur yang lebih baik (Sidik et al., 2023).

MBConv (*Mobile Inverted Residual Bottleneck Convolution*) adalah komponen kunci dalam arsitektur jaringan saraf dalam model *EfficientNetV2*. Dalam *EfficientNetV2*, MBConv digunakan sebagai blok dasar untuk membangun jaringan dengan cara yang efisien dan terukur. Dengan menggunakan MBConv, *EfficientNetV2* dapat mencapai kinerja yang tinggi dengan efisiensi komputasi yang lebih baik dibandingkan model lain. Penggunaan *depthwise* dan *pointwise convolution* dalam MBConv memungkinkan pengurangan jumlah parameter dan operasi, sehingga menghemat sumber daya komputasi dan memori.

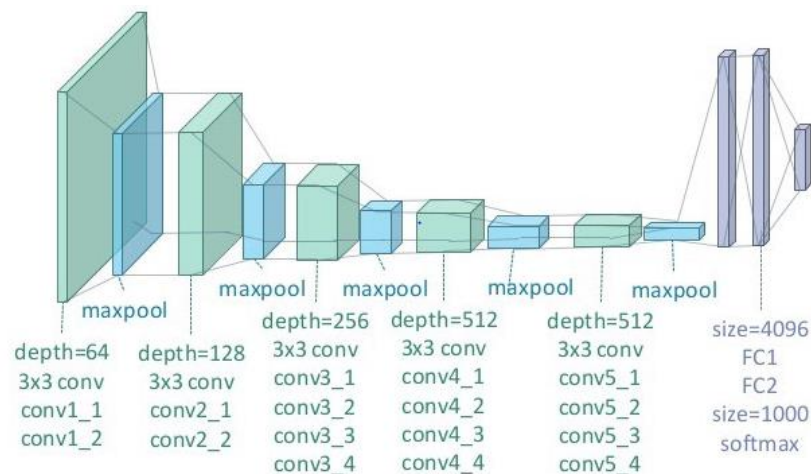
Fused-MBConv (*Fused Mobile Inverted Residual Bottleneck Convolution*) adalah varian dari blok MBConv yang digunakan dalam arsitektur model *EfficientNetV2*. Ini merupakan modifikasi dari blok MBConv yang dirancang untuk meningkatkan efisiensi dan kinerja pada perangkat keras modern. Dalam *EfficientNetV2*, Fused-MBConv digunakan sebagai alternatif atau pengganti MBConv di beberapa bagian jaringan untuk meningkatkan efisiensi dan kinerja. Dengan menggabungkan konvolusi, Fused-MBConv mengurangi jumlah operasi yang harus dilakukan, sehingga mempercepat inferensi dan pelatihan, terutama pada perangkat keras modern seperti GPU dan TPU.

Blok *Squeeze-and-Excitation* (SE) adalah komponen penting yang digunakan dalam berbagai model jaringan saraf termasuk *EfficientNetV2*. Blok SE dirancang untuk meningkatkan representasi jaringan dengan menimbang channel secara adaptif berdasarkan relevansinya terhadap tugas tertentu. Dalam *EfficientNetV2*, blok SE digunakan untuk meningkatkan kapasitas representasi model dengan memberikan perhatian lebih pada channel yang relevan dan mengurangi perhatian pada channel yang kurang penting. Hal ini memungkinkan model untuk lebih efisien dalam menangkap fitur yang penting untuk tugas tertentu, seperti pengenalan gambar atau klasifikasi.

*EfficientNetV2* adalah model pembelajaran progresif yang menyesuaikan nilai regularisasi berdasarkan ukuran gambar. Regularisasi adalah metode untuk mencapai kecocokan yang seimbang dan karenanya merupakan kunci untuk mengurangi fungsi kerugian (Sandhya Devi et al., 2023).

## 2.7 VGG19

*VGG19* adalah jaringan saraf convolutional yang terdiri dari 19 lapisan. Model *VGG19* adalah variasi dari model *VGG* dengan 19 lapisan (16 lapisan konvolusi, 3 lapisan yang terhubung penuh, 5 lapisan *MaxPool* dan 1 lapisan *SoftMax*). Variasi *VGG* lainnya termasuk *VGG11*, *VGG16*, dan banyak lagi. Terdapat 19,6 miliar FLOP dalam *VGG19*. Dalam istilah awam, *VGG* adalah *deep CNN* yang digunakan untuk mengklasifikasikan gambar. Model *VGG19* memiliki lapisan berikut: Conv3x3 (64), Conv3x3 (64), MaxPool, Conv3x3 (128), Conv3x3 (128), MaxPool, Conv3x3 (256), Conv3x3 (256), Conv3x3 (256), Conv3x3 (256), Conv3x3 (256), MaxPool, Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), MaxPool, Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), Conv3x3 (512), MaxPool, Fully Connected (4096), Fully Connected (4096), Fully Connected (1000), SoftMax (Kobra et al., 2022). Gambar 2.11 menunjukkan arsitektur *VGG19*.



**Gambar 2.11** Arsitektur VGG19 (Imen Chebbi, 2021)

## 2.8 Website

Situs web (*website*) adalah kumpulan halaman web yang berisi informasi dalam domain tertentu dan dapat diakses melalui internet. Situs web dapat mencakup beragam bentuk konten, termasuk gambar, video, teks, atau format lainnya, dan dimaksudkan untuk memberikan informasi atau berinteraksi dengan pengunjung. Biasanya, situs web di-*host* dan dapat diakses oleh siapa saja yang memiliki koneksi internet. Fungsi situs web mencakup penyebaran informasi, mendukung produk atau layanan, memfasilitasi

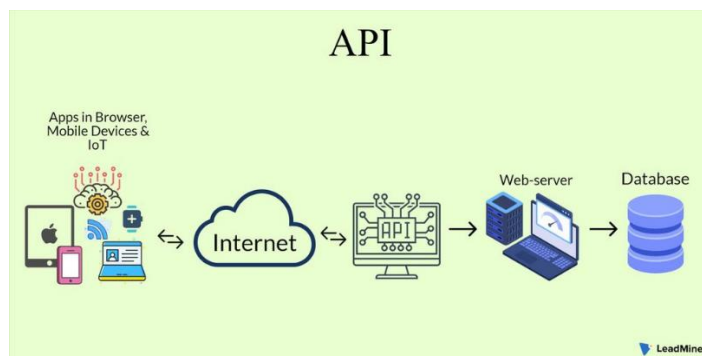


komunikasi, atau memberikan hiburan kepada pengguna. Situs web adalah instrumen penting bagi individu dan bisnis, memungkinkan pengguna untuk membangun kehadiran online dan membangun kredibilitas di ranah digital.

## 2.9 *Application Programming Interface (API)*

*Application Programming Interface (API)* adalah sekumpulan protokol dan instruksi yang memungkinkan sistem perangkat lunak untuk berkomunikasi dan berinteraksi satu sama lain. API berfungsi sebagai perantara untuk memfasilitasi pertukaran data antara dua atau lebih aplikasi perangkat lunak.

API berfungsi sebagai metode untuk menyebarluaskan *Open Data*. API dapat digunakan ketika seseorang ingin mengedit data tanpa perlu memodifikasi halaman web mereka, sebagai ilustrasi. Pengembang mendelegasikan komponen tertentu dari situs web ke API. Tidak ada modifikasi halaman web pengembang yang diperlukan untuk menyajikan data ini.



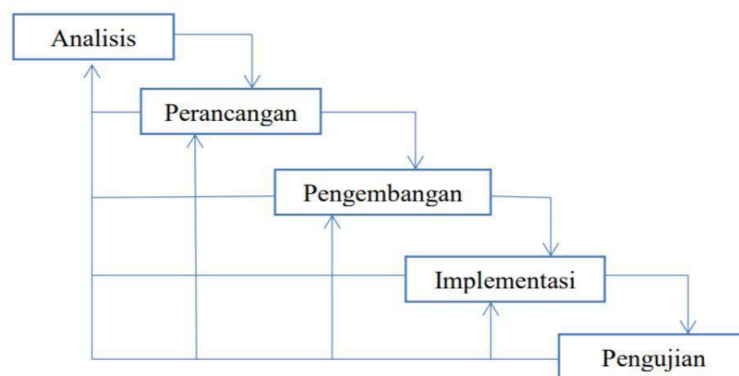
**Gambar 2.12** Ilustrasi API (Sumber: [www.leadmine.net](http://www.leadmine.net))

## BAB 3 ANALISIS DAN PERANCANGAN SISTEM

### 3.1 Analisis Sistem

Analisis sistem juga dianggap sebagai pendekatan pemecahan masalah yang melibatkan pembongkaran sistem menjadi bagian-bagian penyusunnya. Hal ini dilakukan untuk memantau seberapa baik kinerja masing-masing komponen serta bagaimana mereka berinteraksi untuk membantu sistem secara keseluruhan mencapai tujuannya.

Biasanya, analisis sistem dilakukan pada tingkat perencanaan sistem (*system design*). Salah satu dari sekian banyak proses dalam proses rekayasa pemecahan masalah adalah perancangan sistem, yang menggabungkan berbagai komponen pembentuk untuk menciptakan suatu sistem secara keseluruhan.

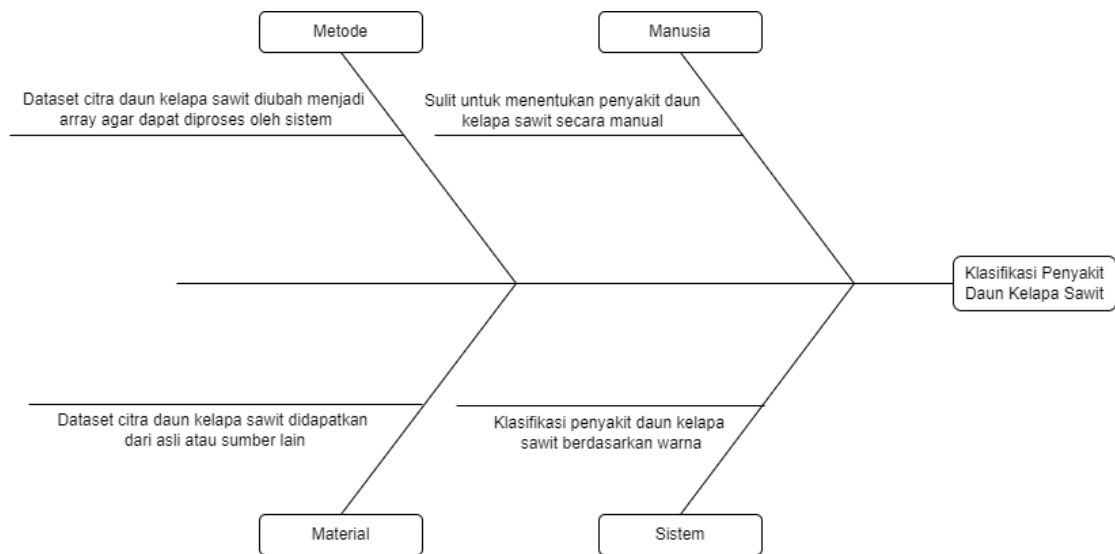


**Gambar 3.1** Analisis Sistem

#### 3.1.1 Analisis Masalah

Dalam penelitian, analisis masalah bertujuan untuk mengetahui penyebab dari permasalahan kemudian dilakukan pengkajian lebih dalam pada masalah yang akan diselesaikan sehingga tercipta sistem yang efektif. Langkah ini adalah poin kritis dalam upaya mengidentifikasi akar penyebab dari masalah yang dihadapi. Pada penelitian ini, untuk menjelaskan permasalahan yang ada, digunakanlah Diagram Ishikawa (*Fishbone*

*Diagram*) untuk menggambarkan akar masalah dalam 4 kategori yaitu, manusia, sistem, metode, dan material yang digambarkan pada Gambar 3.2 berikut.



**Gambar 3.2** *Fishbone Diagram*

### 3.1.2 Analisis Kebutuhan

Analisis kebutuhan adalah langkah penting untuk mengenali data dan prosedur yang diperlukan dalam merancang sistem. Dalam proses perancangan, analisis kebutuhan dapat mencakup persyaratan fungsional dan non-fungsional yang perlu dibuat agar sistem yang dikembangkan dapat mencapai tujuannya.

#### 1. Kebutuhan Fungsional

Tindakan yang harus dilakukan sistem untuk mencapai tujuannya semuanya dianggap sebagai persyaratan fungsional. Kebutuhan Fungsional yang diterapkan dalam penelitian dan sistem ini adalah :

- a. Sistem dapat membaca input gambar dari pengguna.
- b. Sistem dapat mengenali jenis penyakit daun kelapa sawit ketika dilakukan proses klasifikasi oleh model.
- c. Sistem dapat memberikan hasil yaitu penyakit daun kelapa sawit dalam kehidupan pengguna untuk pemahaman pengguna lebih lanjut.

#### 2. Kebutuhan Non-Fungsional

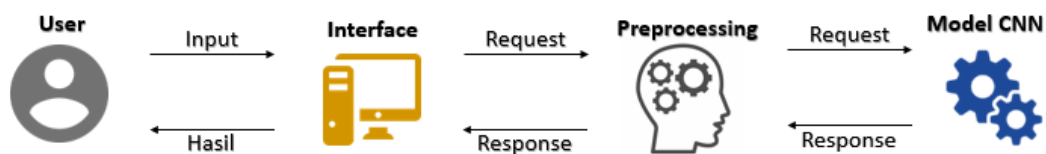
Kebutuhan non-fungsional adalah persyaratan yang mencakup fitur, atribut, pembatasan layanan, atau aspek-aspek lain dari sistem seperti batasan waktu,

kendala proses pengembangan, dan standar sistem yang berfungsi sebagai pelengkap. Di bawah ini terdapat daftar kebutuhan fungsional yang diperlukan dalam sistem ini:

- a. Batasan ekstensi gambar yang dapat digunakan sebagai input adalah .JPG, .PNG, dan .JPEG.
- b. Batasan penyakit daun kelapa sawit yang digunakan adalah *brown spot* dan *white scale*.
- c. User Interface pada sistem akan mudah dipahami oleh pengguna.
- d. Tidak membutuhkan biaya yang besar dan tidak memerlukan perangkat tambahan.
- e. Internet diperlukan untuk terhubung dengan sistem.

### 3.1.3 Arsitektur Umum Sistem

Arsitektur sistem umum merupakan gambaran visual tentang bagaimana sistem akan dibangun.



**Gambar 3.3** Arsitektur Umum Sistem

- a. Pengguna mengunggah gambar daun kelapa sawit ke aplikasi klasifikasi penyakit daun kelapa sawit.
- b. Gambar yang diunggah diterima oleh modul pengelola file dalam sistem.
- c. Modul *preprocessing* menerima gambar dan melakukan tahap preprocessing, seperti resize.
- d. Data gambar yang telah di-*preprocessing* diteruskan ke modul *EfficientNetV2* dan *VGG19*.
- e. Modul *EfficientNetV2* dan *VGG19* akan melakukan proses *training* dan akan menghasilkan file model.keras yang akan digunakan pada proses klasifikasi penyakit daun kelapa sawit.

- f. Modul klasifikasi menghasilkan identifikasi penyakit daun kelapa sawit berdasarkan analisis fitur dari gambar.
- g. Hasil klasifikasi dikirim kembali ke aplikasi dan ditampilkan kepada pengguna.
- h. Pengguna dapat melihat hasil klasifikasi pada antarmuka aplikasi, yang menampilkan informasi tentang penyakit daun kelapa sawit berdasarkan gambar yang diunggah.

Arsitektur ini mencakup modul pengelola file, *preprocessing*, model *EfficientNetV2* dan *VGG19*, klasifikasi, dan antarmuka aplikasi yang bekerja bersama-sama untuk menerima gambar daun kelapa sawit, mengolahnya, menganalisis fitur, melakukan klasifikasi, dan menampilkan hasil kepada pengguna.

### 3.2 Pengambilan Dataset

Tahap awal pada penelitian ini dengan melakukan pengambilan *dataset* yang digunakan. Dataset diambil dari *Kaggle* dan pengambilan langsung ke lapangan yang berupa daun kelapa sawit sehat dan dua jenis penyakit daun kelapa sawit, yaitu *white scale* dan *brown spot*.

### 3.3 Pra-pemrosesan Data

Tahap selanjutnya dengan melakukan pra-pemrosesan data yang bertujuan agar model yang dibangun dapat mempelajari data dengan baik. Beberapa sub-bagian dalam tahap pra-pemrosesan data meliputi, Augmentasi data dan Pembagian Data: *Training*, *Validation*, dan *Testing*.

#### 3.3.1 Augmentasi Data

Augmentasi data adalah teknik untuk memperbesar jumlah dataset dengan membuat modifikasi dari dataset yang sudah ada. Hal ini berguna untuk mengatasi masalah *overfitting*, terutama ketika dataset asli berukuran kecil. Pada program kali ini augmentasi dilakukan dengan memutar gambar sebesar 45 derajat dan menyimpan hasilnya di mana dalam setiap folder, kode menghitung jumlah gambar yang ada dan menambahkan gambar baru hingga mencapai target jumlah gambar yang diinginkan.

### 3.3.2 Pemisahan Data Training, Validasi dan Testing

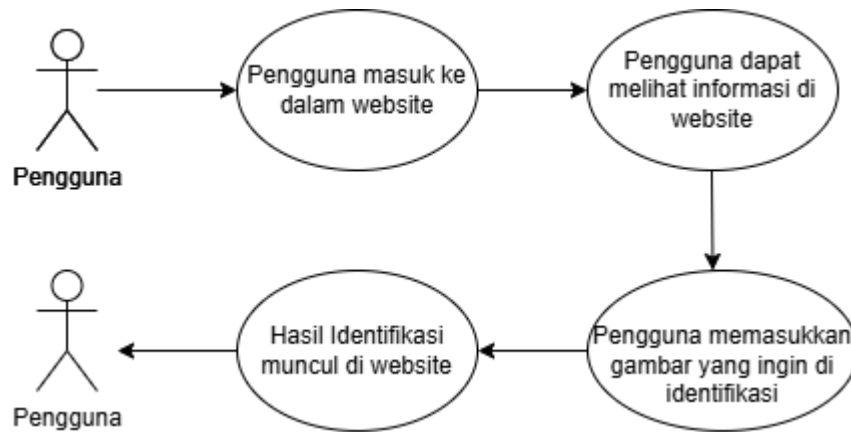
Setelah augmentasi, langkah selanjutnya adalah memisahkan data menjadi set pelatihan, validasi, dan pengujian. Ini dilakukan menggunakan *ImageDataGenerator* dari *Keras*. Di sini, penulis menggunakan beberapa parameter untuk melakukan augmentasi seperti *rescaling*, *shear range*, *zoom range*, *rotation range*, *width shift range*, *height shift range*, *horizontal flip*, *vertical flip*, dan *validation split*. Generator ini digunakan untuk memuat gambar dari direktori, melakukan augmentasi, dan membagi data menjadi batch untuk pelatihan, validasi, dan pengujian.

## 3.4 Pemodelan Sistem

Pemodelan sistem adalah proses menguraikan langkah-langkah interaksi antara pengguna dengan aplikasi yang dibangun, sehingga sistem dapat beroperasi secara efisien. Biasanya, pemodelan sistem diwujudkan dalam format *Unified Modelling Language* (UML). UML adalah bahasa pemodelan yang umum digunakan untuk menggambarkan hubungan antara komponen-komponen dalam sistem, sehingga memungkinkan interaksi melalui pengguna. Dalam penelitian ini, diagram *use case*, diagram aktivitas, dan diagram urutan adalah tiga jenis model UML yang digunakan.

### 3.4.1 Use Case Diagram

Menurut *Unified Modeling Language* (UML), diagram *use case* adalah jenis diagram yang digunakan untuk menggambarkan bagaimana aktor berinteraksi dengan suatu sistem atau aplikasi. Aktor adalah entitas luar yang berperan ketika berinteraksi dengan perangkat keras, sistem lain, atau pengguna manusia. Diagram *use case* yang digunakan dapat dilihat pada Gambar 3.4.

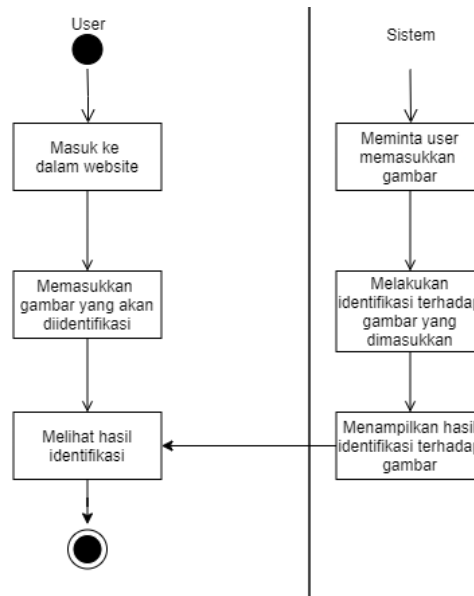


**Gambar 3.4** *Use Case Diagram*

Gambar 3.4 menunjukkan interaksi pengguna dengan sistem. Pengguna akan memulai interaksi dengan masuk aplikasi dan dapat langsung masuk ke inti program. Pengguna dapat memilih Input Gambar di mana dengan memasukkan sebuah gambar ke dalam aplikasi maka akan dilakukan identifikasi terhadap gambar dan dalam beberapa saat akan ditampilkan hasilnya. User dapat melihat apakah daun tersebut sehat atau sakit dan apa nama penyakitnya berdasarkan gambar yang diberikan.

### 3.4.2 *Activity Diagram*

*Activity Diagram* adalah representasi visual dari urutan alur kerja atau proses kegiatan dalam sistem, dimulai dari langkah awal hingga mencapai langkah akhir. *Activity Diagram* juga berfungsi untuk menggambarkan bagian-bagian komponen yang ada dalam *Use Case Diagram*. Berikut *Activity Diagram* yang dipakai dalam sistem:

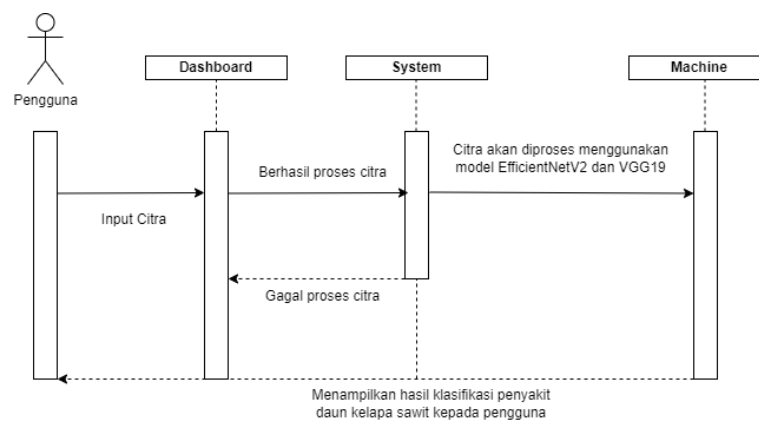


**Gambar 3.5** *Activity Diagram*

Gambar 3.5 menunjukkan *activity diagram* sistem. Pertama *user* membuka halaman aplikasi dan sistem, dan *User* akan diminta memasukkan gambar. *User* kemudian melakukan *input* gambar dan sistem akan menampilkan hasil identifikasi tersebut kepada *User*.

### 3.4.3 Sequence Diagram

*Sequence diagram* adalah suatu bentuk diagram yang digunakan dalam rekayasa perangkat lunak dan desain sistem untuk menggambarkan cara objek atau komponen berinteraksi dalam suatu sistem. Dalam Gambar 3.6, terlihat penggunaan *sequence diagram* dalam mengklasifikasikan penyakit daun kelapa sawit.



**Gambar 3.6** *Sequence Diagram*



### 3.5 Pelatihan Model

Setelah melakukan augmentasi data dan memisahkan data menjadi data *training*, *validation*, dan *testing*, langkah berikutnya adalah melatih model *deep learning* menggunakan data yang telah diproses.

Model ini menggunakan *EfficientNetV2* dan *VGG19* yang telah dilatih sebelumnya di dataset *ImageNet*. *EfficientNetV2* dan *VGG19* merupakan arsitektur *deep learning* yang sangat kuat dan efisien untuk klasifikasi gambar. Model dilatih selama 50 *epoch*. Satu *epoch* adalah satu kali iterasi penuh melalui *training dataset*.

**Tabel 3.1** Tabel Pelatihan Model *EfficientNetV2*

<i>Epoch</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Val_Acc</i>	<i>Val_Loss</i>	<i>Time (s)</i>
45	0.8337	0.4122	0.9190	0.2375	26
46	0.8533	0.3949	0.9190	0.2297	26
47	0.8450	0.4127	0.9190	0.2262	32
48	0.8498	0.3923	0.9190	0.2220	27
49	0.8658	0.3669	0.9214	0.2176	29
50	0.8456	0.3904	0.9238	0.2114	30

**Tabel 3.2** Tabel Pelatihan Model *VGG19*

<i>Epoch</i>	<i>Accuracy</i>	<i>Loss</i>	<i>Val_Acc</i>	<i>Val_Loss</i>	<i>Time (s)</i>
5	0.9774	0.0686	0.9833	0.0362	12
6	0.9828	0.0530	0.9881	0.0385	14
7	0.9881	0.0384	0.9881	0.0436	12
8	0.9852	0.0411	0.9833	0.0399	12
9	0.9917	0.0244	0.9881	0.0376	12
10	0.9887	0.0319	0.9881	0.0372	12

### 3.6 Evaluasi dan Pengujian Model

Setelah proses pelatihan model selesai dilakukan, kemudian dilakukan tahap evaluasi dan pengujian model. Pada penelitian ini, evaluasi dilakukan pada saat proses pelatihan

dengan data latih (*train set*) dan data validasi (*validation set*). Sedangkan tahap pengujian model dilakukan menggunakan data uji (*test set*) dengan cara memprediksi data uji.

### 3.7 Flowchart

*Flowchart* (Diagram Alir) adalah representasi visual dari serangkaian langkah-langkah yang dijalankan oleh sebuah algoritma secara teratur. Setiap tahapan prosesnya direpresentasikan oleh simbol-simbol yang berbeda dan dilengkapi dengan penjelasan untuk setiap tahapannya.

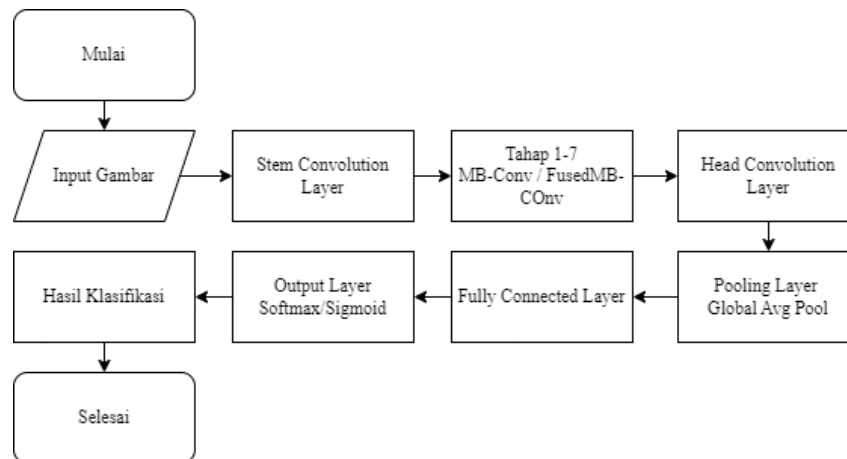
#### 3.7.1 Flowchart Sistem



**Gambar 3.7** Flowchart / Proses Kerja Sistem

Pada Gambar 3.7 dapat dilihat alur kerja dari sistem yang pertama yakni user memasukkan gambar daun kelapa sawit yang akan diprediksi. Kemudian model akan melakukan prediksi dan akan mengembalikan respon berdasarkan hasil klasifikasi kepada *user*, kemudian menampilkan hasil dan informasi relevan terhadap klasifikasi yang dilakukan oleh model.

### 3.7.2 Flowchart Arsitektur *EfficientNetV2*



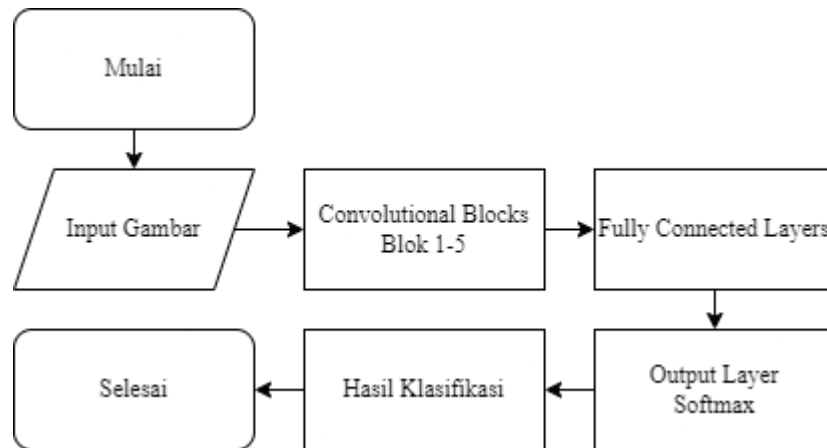
**Gambar 3.8** Flowchart Arsitektur *EfficientNetV2*

Pada Gambar 3.8 dapat dilihat bagaimana cara kerja arsitektur *EfficientNetV2*.

1. *Mulai*: Proses dimulai
2. *Input Gambar*: Gambar dimasukkan sebagai *input* ke dalam model. Menerima gambar berukuran 224x224 piksel dengan 3 saluran warna (RGB).
3. *Stem Convolution Layer*: Lapisan awal yang berisi beberapa lapisan konvolusi dasar untuk memproses input gambar.
4. *Tahap 1-7 MB-Conv/FusedMB-Conv*: Berisi beberapa blok MBConv dan Fused-MBConv, yang merupakan varian dari *convolutional blocks*. Setiap *stage* berisi beberapa blok dengan berbagai konfigurasi *filter*, *kernel size*, dan *stride*.
5. *Head Convolution Layer*: Lapisan akhir yang berisi konvolusi, *batch normalization*, dan aktivasi untuk menghasilkan fitur akhir.
6. *Pooling Layer*: *Global Average Pooling* untuk mereduksi dimensi fitur menjadi satu vektor.
7. *Fully Connected Layer*: Lapisan *dense* yang menghubungkan fitur yang telah diproses ke *output*.
8. *Output Layer*: Lapisan *softmax* atau *sigmoid* untuk menghasilkan probabilitas kelas untuk tugas klasifikasi.
9. *Hasil Identifikasi*: Model menghasilkan prediksi akhir berdasarkan fitur-fitur yang telah diproses dan dapat mengenali jenis daun kelapa sawit.

10. Selesai: Proses berakhir

### 3.7.3 Flowchart Arsitektur VGG19



**Gambar 3.9** Flowchart Arsitektur VGG19

Pada Gambar 3.9 dapat dilihat bagaimana cara kerja arsitektur *EfficientNetV2*.

1. Mulai: Proses dimulai
2. Input Gambar: Gambar dimasukkan sebagai *input* ke dalam model. Menerima gambar berukuran 224x224 piksel dengan 3 saluran warna (RGB).
3. *Convolutional Blocks*:
  - Block 1: Dua lapisan konvolusi (masing-masing dengan 64 filter) diikuti oleh satu lapisan *max pooling*.
  - Block 2: Dua lapisan konvolusi (masing-masing dengan 128 filter) diikuti oleh satu lapisan *max pooling*.
  - Block 3: Empat lapisan konvolusi (masing-masing dengan 256 filter) diikuti oleh satu lapisan *max pooling*.
  - Block 4: Empat lapisan konvolusi (masing-masing dengan 512 filter) diikuti oleh satu lapisan *max pooling*.
  - Block 5: Empat lapisan konvolusi (masing-masing dengan 512 filter) diikuti oleh satu lapisan *max pooling*.
4. *Fully Connected Layer*: Lapisan *dense* yang menghubungkan fitur yang telah diproses ke *output*. Tiga lapisan *fully connected* dengan jumlah unit 4096, 4096, dan 1000, berturut-turut.

5. *Output Layer*: Lapisan *softmax* untuk menghasilkan distribusi probabilitas kelas untuk tugas klasifikasi.
6. Hasil Identifikasi: Model menghasilkan prediksi akhir berdasarkan fitur-fitur yang telah diproses dan dapat mengenali jenis daun kelapa sawit.
7. Selesai: Proses berakhir

### 3.8 Perancangan Antarmuka

Perancangan Antarmuka adalah proses menciptakan kerangka desain sistem yang akan dikonstruksi. Perancangan Antarmuka adalah langkah yang diperlukan untuk memastikan bahwa proses pembuatan sistem dapat berfungsi lebih efektif berdasarkan panduan yang telah didefinisikan dalam desain antarmuka tersebut.

#### 3.8.1 Halaman Identifikasi

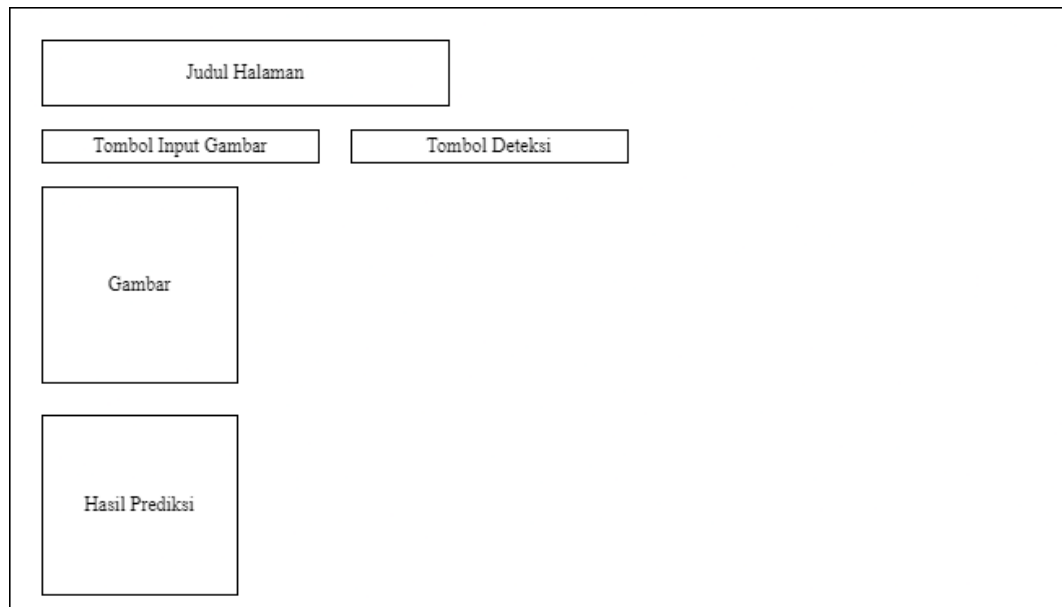
Halaman Identifikasi *website* berisi tampilan input gambar pengguna terkait daun kelapa sawit.



**Gambar 3.10** Gambar Rangka Halaman Klasifikasi

#### 3.8.2 Halaman Hasil Identifikasi

Halaman Hasil Identifikasi *website* berisi tampilan yang memunculkan gambar yang di-*input* dan hasil prediksi.



**Gambar 3.11** Gambar Rangka Halaman Hasil Klasifikasi

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

#### **4.1 Implementasi Sistem**

Pada penelitian ini, sistem dibangun dengan memanfaatkan model *EfficientNetV2* dan *VGG19*. Sistem dibangun dengan dua komponen yaitu *front-end* dan *back-end*. Bagian *front-end* dibangun dengan HTML dan *back-end* dibangun menggunakan bantuan bahasa pemrograman *Python*. Sistem juga dibantu menggunakan aplikasi seperti *visual studio code*.

##### *4.1.1 Perangkat Keras*

1. Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 2.50 GHz
2. 8 GB RAM

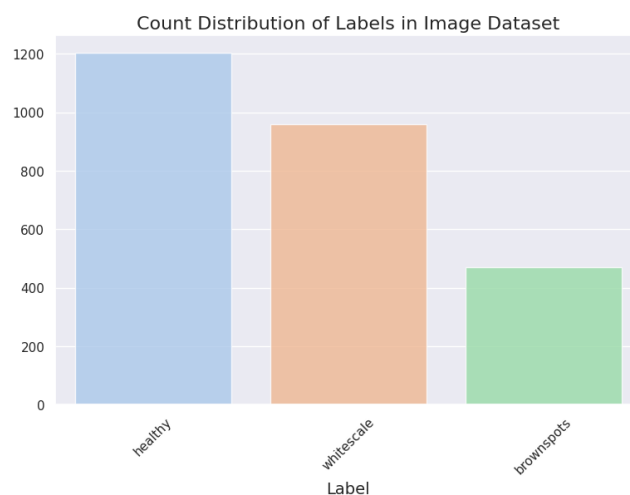
##### *4.1.2 Perangkat Lunak*

1. Implementasi Pengambilan Dataset
  - Kaggle
2. Implementasi pra-pemrosesan dataset
  - Google Colab
  - Library
3. Implementasi membangun, melatih, dan mengevaluasi model
  - Google Colab
  - Library
4. Implementasi model ke aplikasi berbasis website
  - Visual Studio Code
  - Bahasa : HTML, CSS, JavaScript, Python
  - Framework : Bootstrap, Flask

## 4.2 Implementasi Dataset

*Dataset* yang digunakan diambil dari *Kaggle* di mana terdapat 3 kelas yaitu *healthy*, *brown spot* dan *white scale*. *Dataset* berjumlah 2.631 citra penyakit daun sawit dengan 1203 citra daun kelapa sawit pada kelas *healthy*, 470 citra daun kelapa sawit pada kelas *white scale*, dan 958 citra daun kelapa sawit pada kelas *brown spot*.

```
There are 3 directories and 0 images in '/content/drive/MyDrive/DataSawit'.  
There are 0 directories and 1203 images in '/content/drive/MyDrive/DataSawit/healthy'.  
There are 0 directories and 470 images in '/content/drive/MyDrive/DataSawit/brownspots'.  
There are 0 directories and 958 images in '/content/drive/MyDrive/DataSawit/whitescale'.
```



**Gambar 4.1** Jumlah *Dataset* Tiap Folder

Gambar 4.1 menunjukkan distribusi jumlah *dataset* pada setiap folder, dan hasil nya ditampilkan dalam bentuk grafik batang.

## 4.3 Implementasi Model

Setelah tahap pra-pemrosesan data, selanjutnya adalah tahap membangun model yaitu menggunakan *EfficientNetV2* dan *VGG19*. Tahap ini dilakukan menggunakan library TensorFlow dan Keras dengan arsitektur umum jaringan saraf tiruan yang terdiri dari tiga lapisan: lapisan masukan, lapisan keluaran, dan lapisan tersembunyi. Arsitektur model terdiri dari sebuah lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*) dengan 3 buah *dense layer* dan juga 3 buah *dropout layer* masing-masing setelah *dense layer*, serta sebuah lapisan keluaran (*output layer*).



### 4.3.1 Implementasi Model EfficientNetV2

Model: "model\_16"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[None, 224, 224, 3]	0	[]
rescaling_3 (Rescaling)	(None, 224, 224, 3)	0	['input_2[0][0]']
stem_conv (Conv2D)	(None, 112, 112, 32)	864	['rescaling_3[0][0]']
stem_bn (BatchNormalization)	(None, 112, 112, 32)	128	['stem_conv[0][0]']
stem_activation (Activation)	(None, 112, 112, 32)	0	['stem_bn[0][0]']
block1a_project_conv (Conv2D)	(None, 112, 112, 32)	9216	['stem_activation[0][0]']
block1a_project_bn (BatchNormalization)	(None, 112, 112, 32)	128	['block1a_project_conv[0][0]']
block1a_project_activation (Activation)	(None, 112, 112, 32)	0	['block1a_project_bn[0][0]']
block1a_add (Add)	(None, 112, 112, 32)	0	['block1a_project_activation[0][0]', 'stem_activation[0][0]']
block1b_project_conv (Conv2D)	(None, 112, 112, 32)	9216	['block1a_add[0][0]']
block1b_project_bn (BatchNormalization)	(None, 112, 112, 32)	128	['block1b_project_conv[0][0]']
block1b_project_activation (Activation)	(None, 112, 112, 32)	0	['block1b_project_bn[0][0]']
block1b_drop (Dropout)	(None, 112, 112, 32)	0	['block1b_project_activation[0][0]']
block1b_add (Add)	(None, 112, 112, 32)	0	['block1b_drop[0][0]', 'block1a_add[0][0]']
...			
block7g_se_expand (Conv2D)	(None, 1, 1, 3840)	618240	['block7g_se_reduce[0][0]']
block7g_se_excite (Multiply)	(None, 7, 7, 3840)	0	['block7g_activation[0][0]', 'block7g_se_expand[0][0]']
block7g_project_conv (Conv2D)	(None, 7, 7, 640)	2457600	['block7g_se_excite[0][0]']
block7g_project_bn (BatchNormalization)	(None, 7, 7, 640)	2560	['block7g_project_conv[0][0]']
block7g_drop (Dropout)	(None, 7, 7, 640)	0	['block7g_project_bn[0][0]']
block7g_add (Add)	(None, 7, 7, 640)	0	['block7g_drop[0][0]', 'block7f_add[0][0]']
top_conv (Conv2D)	(None, 7, 7, 1280)	819200	['block7g_add[0][0]']
top_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	['top_conv[0][0]']
top_activation (Activation)	(None, 7, 7, 1280)	0	['top_bn[0][0]']
max_pool (GlobalMaxPooling2D)	(None, 1280)	0	['top_activation[0][0]']
dense_3 (Dense)	(None, 128)	163968	['max_pool[0][0]']
dropout_2 (Dropout)	(None, 128)	0	['dense_3[0][0]']
dense_4 (Dense)	(None, 256)	33024	['dropout_2[0][0]']
dropout_3 (Dropout)	(None, 256)	0	['dense_4[0][0]']
dense_5 (Dense)	(None, 3)	771	['dropout_3[0][0]']
=====			
Total params: 117944611 (449.92 MB)			
Trainable params: 197763 (772.51 KB)			
Non-trainable params: 117746848 (449.17 MB)			

**Gambar 4.2** Summary Model EfficientNetV2

Dari Gambar 4.2 dapat dilihat bahwa model memiliki parameter di mana total parameter = 117944611, jumlah parameter yang dapat dilatih = 197763 dan jumlah parameter yang tidak dapat dilatih = 117746848.

### 4.3.2 Implementasi Model VGG19

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590880
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590880
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_max_pooling2d (GlobalMaxPooling2D)	(None, 512)	0
dense (Dense)	(None, 128)	65664
batch_normalization (Batch Normalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 256)	33024
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 3)	771

=====  
 Total params: 20125379 (76.77 MB)  
 Trainable params: 100227 (391.51 KB)  
 Non-trainable params: 20025152 (76.39 MB)

**Gambar 4.3** Summary Model VGG19

Dari Gambar 4.3 dapat dilihat bahwa model memiliki parameter di mana total parameter = 20125379, jumlah parameter yang dapat dilatih = 100227 dan jumlah parameter yang tidak dapat dilatih = 20025152. Penjelasan model sebagai berikut:

#### 1. Input Layer

Menerima input gambar dengan ukuran (224, 224, 3), di mana 224 x 224 adalah resolusi gambar dan 3 menunjukkan jumlah kanal warna (RGB).

#### 2. Block 1

- a. *Convolutional Layer* (Conv2D) : Lapisan pertama adalah *Convolutional Layer* dengan 64 *filter* menggunakan *kernel* 3x3. Ini diikuti oleh lapisan Conv2D kedua yang serupa.
- b. *Max Pooling Layer* (MaxPooling2D) : Setelah dua lapisan Conv2D, selanjutnya MaxPooling2D memiliki ukuran *pool* 2x2 untuk mengurangi dimensi gambar menjadi (112, 112, 64).
- c. Jumlah *layer* di *Block* 1: 3.

### 3. *Block* 2

- a. *Convolutional Layer* (Conv2D) : Blok kedua terdiri dari dua *Convolutional Layer* dengan 128 *filter* menggunakan *kernel* 3x3.
- b. *Max Pooling Layer* (MaxPooling2D) : Setelah dua lapisan Conv2D, selanjutnya MaxPooling2D memiliki ukuran *pool* 2x2 untuk mengurangi dimensi gambar menjadi (56, 56, 128).
- c. Jumlah *layer* di *Block* 2: 3.

### 4. *Block* 3

- a. *Convolutional Layer* (Conv2D) : Blok ketiga terdiri dari empat *Convolutional Layer* dengan 256 *filter* menggunakan *kernel* 3x3.
- b. *Max Pooling Layer* (MaxPooling2D) : Setelah empat lapisan Conv2D, selanjutnya MaxPooling2D memiliki ukuran *pool* 2x2 untuk mengurangi dimensi gambar menjadi (28, 28, 256).
- c. Jumlah *layer* di *Block* 3: 5.

### 5. *Block* 4

- a. *Convolutional Layer* (Conv2D) : Blok keempat terdiri dari empat *Convolutional Layer* dengan 512 *filter* menggunakan *kernel* 3x3.
- b. *Max Pooling Layer* (MaxPooling2D) : Setelah empat lapisan Conv2D, selanjutnya MaxPooling2D memiliki ukuran *pool* 2x2 untuk mengurangi dimensi gambar menjadi (14, 14, 512).
- c. Jumlah *layer* di *Block* 4: 5.

### 6. *Block* 5

- a. *Convolutional Layer* (Conv2D) : Blok keempat terdiri dari empat *Convolutional Layer* dengan 512 *filter* menggunakan *kernel* 3x3.

- b. *Max Pooling Layer* (MaxPooling2D) : Setelah empat lapisan Conv2D, selanjutnya MaxPooling2D memiliki ukuran pool 2x2 untuk mengurangi dimensi gambar menjadi (7, 7, 512).
- c. Jumlah *layer* di *Block 5*: 5.

#### 7. *Global Max Pooling*

GlobalMaxPooling2D mengambil nilai maksimum dari setiap filter, mengubah dimensi gambar menjadi (None, 512)

#### 8. *Fully Connected Layers (Dense Layers)*

Setelah lima blok konvolusi, selanjutnya pada bagian *fully connected layers* dari model ini, terdapat tiga lapisan utama yang dapat dihitung sebagai *fully connected layers*, yaitu dua *Dense layers* dan satu *output Dense layers*. Jumlah *layer* di *Fully Connected Layers*: 7.

- a. *Dense Layer* (Dense) 1 : Lapisan pertama *dense* dengan 128 unit diikuti oleh aktivasi ReLU lalu dilakukan *Batch Normalization* dan *Dropout*.
- b. *Dense Layer* (Dense) 2 : Lapisan kedua *dense* dengan 256 unit diikuti oleh aktivasi ReLU lalu dilakukan *Batch Normalization* dan *Dropout*.

#### 9. *Output Layer*

Lapisan *output* terakhir adalah lapisan *Dense* dengan jumlah unit yang sesuai dengan jumlah kelas yang diinginkan untuk melakukan prediksi yaitu 3 dan diikuti oleh aktivasi *Softmax* untuk klasifikasi 3 kelas. Jumlah *layer* di *Output Layer*: 1.

Jumlah total *layer* pada model ini adalah 3 (*Block 1*) + 3 (*Block 2*) + 5 (*Block 3*) + 5 (*Block 4*) + 5 (*Block 5*) + 7 (*Fully Connected*) + 1 (*Output*) = 29 *layer*. VGG19 seharusnya memiliki 19 *layer* sesuai namanya namun diberikan modifikasi sebagai berikut:

##### 1. *Pre-trained VGG19 Model*

Dimulai dengan model *VGG19 pre-trained* tanpa lapisan atas dan dengan lapisan *max pooling*. Lapisan konvolusi dari *VGG19* dipertahankan, tetapi bagian *fully connected* aslinya dihilangkan. Menetapkan semua *layer* dalam *pretrained\_model* tidak dapat dilatih dan bobotnya tidak akan berubah selama pelatihan.

##### 2. *Augmentation*

Diasumsikan bahwa *augment* adalah fungsi augmentasi yang diterapkan pada input. Fungsi augmentasi ini tidak ditampilkan di ringkasan model.

### 3. *Fully Connected Layers* tambahan

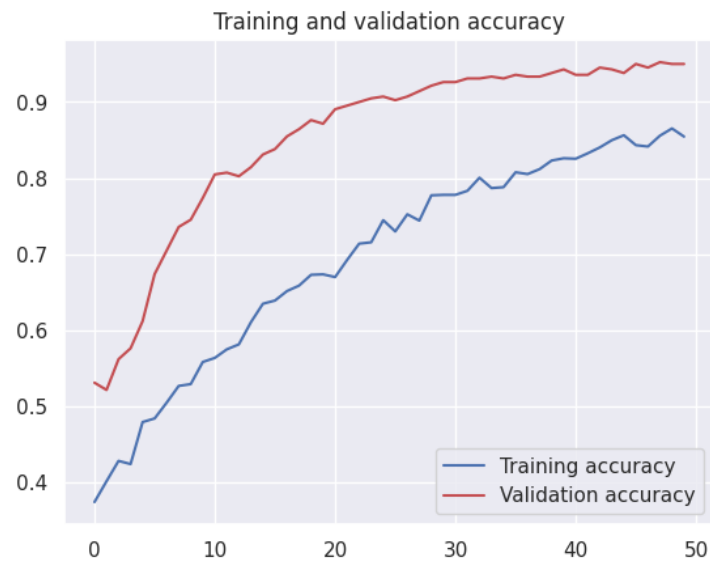
- Menambahkan *layer Dense* dengan 128 unit dengan aktivasi ReLU.
- Menambahkan *layer BatchNormalization*.
- Menambahkan *layer Dropout*.
- Menambahkan *layer Dense* dengan 256 unit dengan aktivasi ReLU.
- Menambahkan *layer BatchNormalization*.
- Menambahkan *layer Dropout*.
- Menambahkan *layer Dense* dengan 3 unit untuk *output* dengan aktivasi *softmax*.

Sehingga kesimpulannya jumlah lapisan pada *pre-trained VGG19* (tanpa bagian atas) ada 22 *layer* dan jumlah *layer* pada *Fully Connected Layers* tambahan ada 7 *layer*. Sehingga, model memiliki 29 *layer* meskipun dimulai dengan model *VGG19* yang asli memiliki 19 *layer* berisi bobot.

## 4.4 Implementasi Tahap Evaluasi dan Pengujian Model

Implementasi Tahap Evaluasi dan Pengujian Model adalah tahapan yang dilakukan setelah tahapan implementasi model berhasil dilakukan. Tujuannya adalah untuk menunjukkan bahwa sistem yang dikembangkan dapat berfungsi dengan baik, dalam mengenali jenis penyakit daun kelapa sawit. Pada tahap ini, terdapat tiga kelas jenis citra daun kelapa sawit yaitu *healthy*, *brown spot* dan *white scale*. Berikut adalah tabel yang menampilkan jenis-jenis citra daun kelapa sawit.

<i>Healthy</i>	<i>Brown Spot</i>	<i>White Scale</i>
		

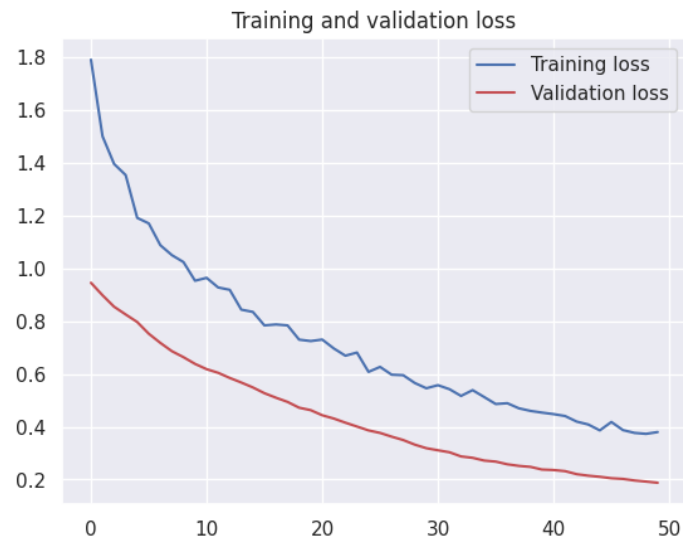


**Gambar 4.4** Grafik *Training dan Validation Accuracy EfficientNetV2*

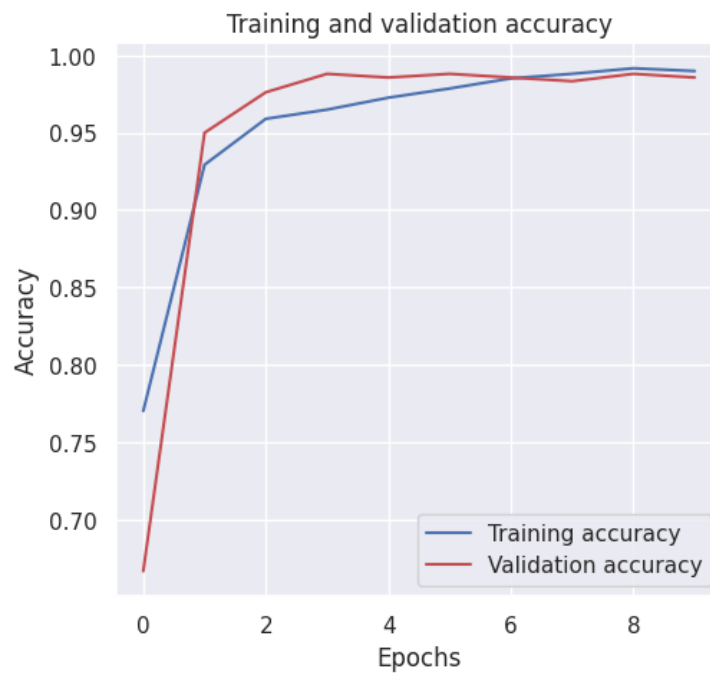


**Gambar 4.5** Grafik *Training and Validation Accuracy VGG19*

Pada Gambar 4.4 dan 4.5 terdapat grafik yang dihasilkan berdasarkan akurasi latihan dan validasi data. Kedua garis terlihat saling berdampingan pada bagian atas grafik. Hal ini menunjukkan bahwa model dapat melakukan identifikasi yang baik dengan data latihan serta data yang baru pertama kali diperlihatkan.

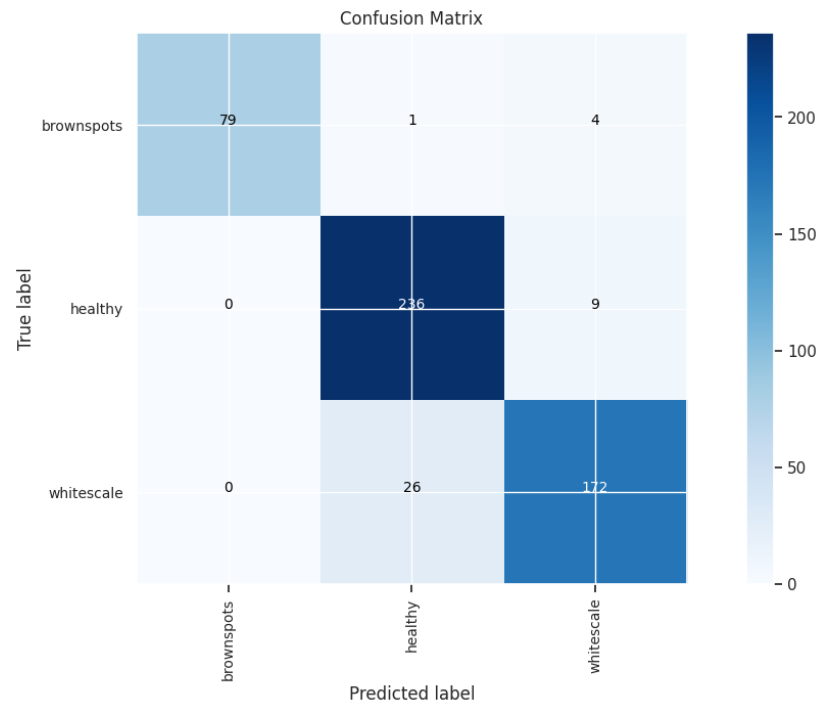


**Gambar 4.6** Grafik *Training and Validation Loss EfficientNetV2*



**Gambar 4.7** Grafik *Training and Validation Loss VGG19*

Pada gambar 4.6 dan 4.7 terdapat grafik yang dihasilkan berdasarkan loss latihan dan validasi. Hal ini mengindikasikan bahwa model mempelajari pola data dengan baik pada saat latihan dan dapat mempelajari data baru dengan baik pula pada validasi.

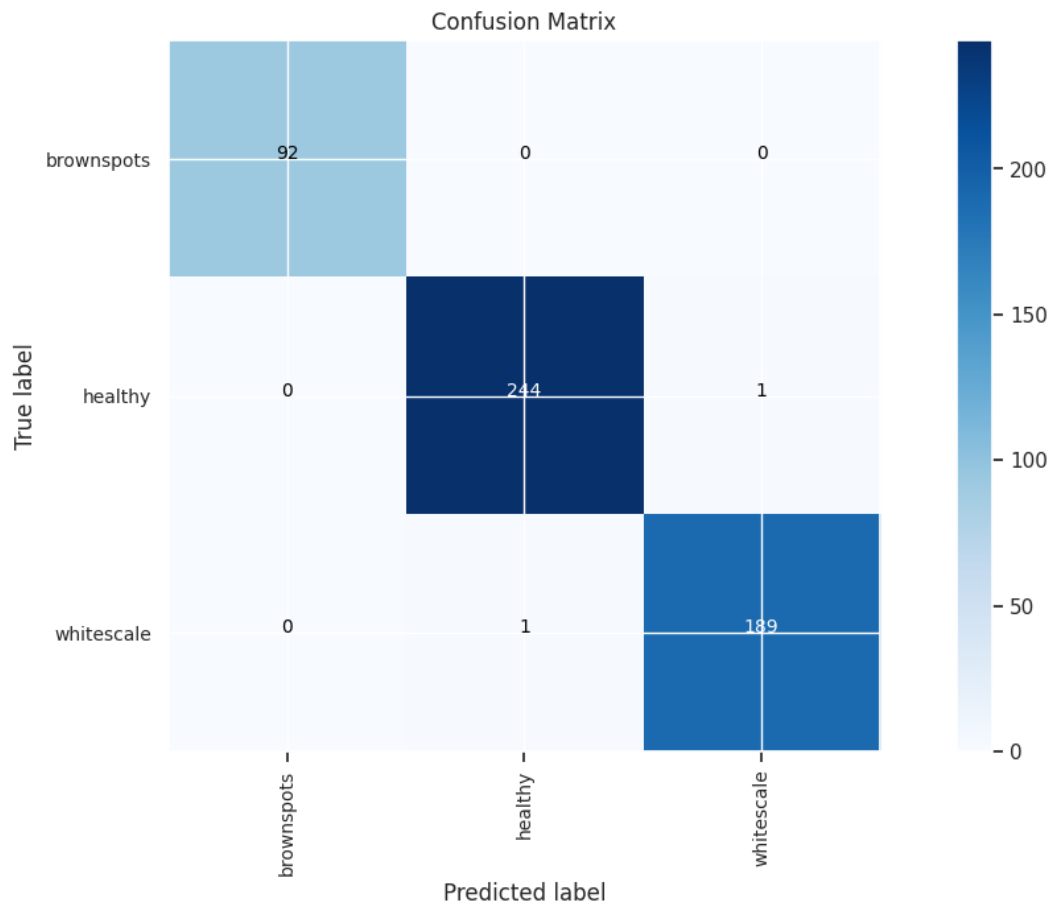


**Gambar 4.8** *Confusion Matrix EfficientNetV2*

Pada Gambar 4.8 terdapat *confusion matrix* model *EfficientNetV2* yang dihasilkan dari prediksi terhadap data *testing* dapat disimpulkan:

- Dari 84 gambar pada kelas *brownspots*. Sebanyak 79 gambar diprediksi dengan benar, 1 gambar diprediksi sebagai *healthy*, dan 4 gambar diprediksi sebagai *whitescale*,
- Dari 245 gambar pada kelas *healthy*. Tidak ada gambar yang diprediksi sebagai *brownspots*, sebanyak 236 gambar diprediksi dengan benar, dan 9 gambar diprediksi sebagai *whitescale*,
- Dari 198 gambar pada kelas *whitescale*. Tidak ada gambar yang diprediksi sebagai *brownspots*, sebanyak 26 gambar diprediksi sebagai *healthy*, dan 172 gambar diprediksi dengan benar.





**Gambar 4.9** *Confusion Matrix VGG19*

Pada gambar 4.9 terdapat *confusion matrix* model *VGG19* yang dihasilkan dari prediksi terhadap data *testing* dapat disimpulkan:

- Dari 92 gambar pada kelas *brownspots*. Sebanyak 92 gambar diprediksi dengan benar, tidak ada gambar diprediksi sebagai *healthy*, dan tidak ada gambar diprediksi sebagai *whitescale*,
- Dari 245 gambar pada kelas *healthy*. Tidak ada gambar yang diprediksi sebagai *brownspots*, sebanyak 244 gambar diprediksi dengan benar, dan 1 gambar diprediksi sebagai *whitescale*,
- Dari 190 gambar pada kelas *whitescale*. Tidak ada gambar yang diprediksi sebagai *brownspots*, sebanyak 1 gambar diprediksi sebagai *healthy*, dan 189 gambar diprediksi dengan benar.

**Tabel 4.1** Metrik Evaluasi Model

<b>Model</b>	<b><i>Accuracy</i></b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>F1-Score</i></b>
<i>EfficientNetV2</i>	93,92%	95,70%	94,50%	94,94%
<i>VGG19</i>	99,43%	99,51%	99,55%	99,53%

Berdasarkan confusion matrix maka langkah selanjutnya yang dilakukan adalah mencari nilai *Recall*, *Precision*, *F1 Score*, dan *Accuracy*. Nilai *Recall* yang tinggi menunjukkan model dapat mengenali kelas dengan baik. Pada tabel 4.1 terlihat bahwa model *VGG19* memiliki nilai *recall* 99,55% di mana nilai ini lebih tinggi dibandingkan model *EfficientNetV2* dengan nilai *recall* 94,50%.

*Precision* adalah nilai yang menunjukkan ketepatan model dalam memprediksi sebuah kelas dengan benar. Pada tabel 4.1 terlihat bahwa model *VGG19* memiliki nilai *precision* sebesar 99,51% di mana nilai ini lebih tinggi dibandingkan model *EfficientNetV2* dengan nilai 95,70%. Sedangkan, *f1 score* merupakan *Harmonic Mean* antara *precision* dan *recall*. Nilai *f1 score* yang tinggi mengindikasikan model memiliki *precision* dan *recall* yang baik. Pada tabel 4.1 terlihat bahwa model *VGG19* mempunyai nilai *f1 score* lebih tinggi sebesar 99,53% sedangkan model *EfficientNetV2* memiliki nilai *f1 score* sebesar 94,94%.

Selanjutnya adalah nilai *accuracy*. *Accuracy* dapat menunjukkan seberapa akurat model dapat melakukan pendeteksian dengan benar terhadap keseluruhan data. *VGG19* mempunyai nilai *accuracy* lebih tinggi yaitu 99,43% % sedangkan *EfficientNetV2* memiliki nilai *accuracy* 93,92%.

#### 4.5 Hasil Perbandingan Arsitektur

Pada tahap ini merupakan hasil perbandingan dari arsitektur *EfficientNetV2* dan *VGG19* dari hasil pengujian data dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Perbandingan *EfficientNetV2* dan *VGG19*




<b>Arsitektur</b>	<b><i>EfficientNetV2</i></b>	<b><i>VGG19</i></b>
<i>Running Time</i>	25 menit 15 detik	2 menit 21 detik
<i>Train loss</i>	0.3903	0.0319

<i>Train accuracy</i>	0.8450	0.9887
<i>Validation loss</i>	0.2113	0.0372
<i>Validation accuracy</i>	0.9238	0.9881
<i>Test loss</i>	0.1967	0.0180
<i>Test accuracy</i>	0.9393	0.9943
<i>Trainable parameter</i>	197763	100227
Kapasitas penyimpanan	449.92 MB	76.77 MB

#### 4.6 Hasil Pengujian Sistem

Setelah dilakukan pengujian sistem, berikut hasil yang didapat dengan mencoba semua daun yang telah dipaparkan di atas:

**Tabel 4.3** Daftar Hasil Percobaan

<i>Input</i>	<i>Output</i>
	<b>Prediction EfficientNetV2</b> <ul style="list-style-type: none"> <li>• healthy: 93%</li> </ul> <b>Prediction VGG16</b> <ul style="list-style-type: none"> <li>• healthy: 99%</li> </ul>
	<b>Prediction EfficientNetV2</b> <ul style="list-style-type: none"> <li>• healthy: 95%</li> </ul> <b>Prediction VGG16</b> <ul style="list-style-type: none"> <li>• brown spots: 99%</li> </ul>
	<b>Prediction EfficientNetV2</b> <ul style="list-style-type: none"> <li>• healthy: 95%</li> </ul> <b>Prediction VGG16</b> <ul style="list-style-type: none"> <li>• white scale: 99%</li> </ul>

Selain dilakukan pengujian pada *website*, ada dilakukan pengujian pada model juga yang menghasilkan hasil sebagai berikut:



**Gambar 4.10** Tampilan keluaran model *EfficientNetV2*



**Gambar 4.11** Tampilan keluaran model *VGG19*

Pada Gambar 4.10 dan 4.11 terdapat hasil keluaran prediksi gambar yang dilakukan. Dapat dilihat, model berhasil mengidentifikasi data baru yang diberikan dengan akurasi yang tinggi. Hal ini berarti model telah dilatih dengan baik dan siap melakukan identifikasi untuk data baru yang belum pernah dilihat lainnya.

## 4.7 Implementasi Pada Website

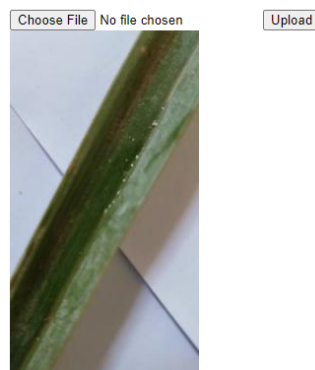
---

### Upload an Image for Classification

No file chosen

**Gambar 4.12** Halaman Identifikasi

### Upload an Image for Classification



#### Prediction EfficientNetV2

- healthy: 94%

#### Prediction VGG16

- healthy: 99%

**Gambar 4.13** Halaman Hasil Klasifikasi dengan gambar

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian dan implementasi yang dijabarkan dalam Bab 4, Ada beberapa kesimpulan yang dapat penulis buat, khususnya :

1. Pada penelitian ini, akurasi model *VGG19* terus meningkat seiring dengan bertambahnya jumlah *epoch*, mencapai akurasi tertinggi sebesar 98,81% dan akurasi model *EfficientNetV2* mencapai akurasi tertinggi sebesar 84,50%
2. Model mencapai akurasi yang sangat tinggi. Hal ini menunjukkan bahwa model *EfficientNetV2* dan *VGG19* sangat efektif dalam mengidentifikasi penyakit pada daun kelapa sawit.
3. Performa yang baik dari model ini menunjukkan potensi besar dalam aplikasi praktis untuk identifikasi penyakit pada daun kelapa sawit secara otomatis, yang dapat bermanfaat bagi petani kelapa sawit.

#### **5.2 Saran**

Berikut beberapa saran dan rekomendasi yang penulis dapat berikan dari hasil penelitian ini untuk pengembangan sistem selanjutnya, yaitu:

1. Disarankan untuk memperluas *dataset* yang digunakan dalam pelatihan model dengan lebih banyak jenis penyakit pada daun kelapa sawit dan variasi gambar. Hal ini akan membantu meningkatkan generalisasi model dan akurasinya dalam situasi nyata.
2. Melakukan optimisasi lebih lanjut pada model untuk melihat dampaknya terhadap akurasi dan waktu pelatihan serta mendapatkan hasil yang lebih baik.
3. Menambahkan fitur tambahan seperti bentuk pencegahan dan penanganan penyakit serta analisis data pengguna untuk memperkaya fungsi aplikasi.

4. Mengembangkan versi *mobile* dari aplikasi untuk meningkatkan aksesibilitas dan kenyamanan penggunaan oleh masyarakat luas, terutama bagi pengguna yang membutuhkan klasifikasi penyakit daun sawit di lapangan.

## DAFTAR PUSTAKA

- Hadjer Hamaidi. (2019). *Date Palm Data*. Kaggle.Com.  
<https://www.kaggle.com/datasets/hadjerhamaidi/date-palm-data>
- Kobra, K.-T., Suham, R. R., & Fairouz, M. (2022). *Plant Disease Diagnosis Using Deep Transfer Learning Architectures- VGG19, MobileNetV2 and Inception-V3* [Thesis]. Brac University.
- Kusumastuti, R., & Sunyoto, A. (2023). Skin Cancer Classification Using EfficientNetV2 and ViT B16. *2023 6th International Conference on Information and Communications Technology, ICOIACT 2023*, 395–400.  
<https://doi.org/10.1109/ICOIACT59844.2023.10455921>
- OECD/FAO. (2021). Global vegetable oils: Oilseeds and oilseed products. *OECD-FAO Agricultural Outlook 2021-2030*, 138–149.
- Pribadi, A., & Kurniawan, A. (2022). Deteksi Penyakit Sawit Menggunakan Metode Deep Learning. In *Jurnal Sains dan Ilmu Terapan* (Vol. 5, Issue 2).
- Priwiratama, H., Eris, D. D., Pradana, M. G., & Rozziansha, T. A. P. (2023). STATUS TERKINI PENYAKIT BERCAK DAUN KELAPA SAWIT DI SUMATERA DAN KALIMANTAN. *WARTA Pusat Penelitian Kelapa Sawit*, 28(1), 27–38.  
<https://doi.org/10.22302/iopri.war.warta.v28i1.101>
- Rahmawati, A. (2023). Keragaman Genetik Varietas Kelapa Sawit (*Elaeis guineensis* Jacq.). *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, 5.
- Sandhya Devi, R. S., Vijay Kumar, V. R., & Sivakumar, P. (2023). EfficientNetV2 Model for Plant Disease Classification and Pest Recognition. *Computer Systems Science and Engineering*, 45(2), 2249–2263. <https://doi.org/10.32604/csse.2023.032231>
- Sidik, D. P., Utaminingrum, F., & Muflikhah, L. (2023). *Penggunaan Variasi Model pada Arsitektur EfficientNetV2 untuk Prediksi Sel Kanker Serviks* (Vol. 7, Issue 5). <http://j-ptiik.ub.ac.id>