

**KLASIFIKASI PENYAKIT PADA TUMBUHAN PADI  
MELALUI CITRA DAUN DENGAN PEMANFAATAN  
MODEL VGG16**

**SKRIPSI**

**ARI RAMA GUNANDA**

**198160051**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2025**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 13/3/26

Access From (repositori.uma.ac.id)13/3/26

**KLASIFIKASI PENYAKIT PADA TUMBUHAN PADI  
MELALUI CITRA DAUN DENGAN PEMANFAATAN  
MODEL VGG16**

**SKRIPSI**

Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area

Oleh:

ARI RAMA GUNANDA

198160051

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2025**

## LEMBAR PENGESAHAN

Judul Skripsi : **Klasifikasi Penyakit pada Tumbuhan Padi Melalui Citra Daun Dengan Pemanfaatan Model VGG16**  
Nama : **Ari Rama Gunanda**  
NPM : **198160051**  
Fakultas : **Teknik Informatika**

Disetujui Oleh  
Komisi  
Pembimbing



Susilawati, S.kom, M.kom  
Pembimbing



Tanggal Lulus : 17 September 2025

## HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagianbagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya dengan jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku apabila dikemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan 17 September 2025  
Penulis



Ari Rama Gunanda

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan dibawah ini :

Nama : Ari Rama Gunanda

NPM : 198160051

Fakultas : Teknik

Program Study : Teknik Informatika

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul : **"Klasifikasi Penyakit Pada Tumbuhan Padi Melalui Citra Daun Dengan Pemanfaatan Model VGG-16"**. Bersama dengan perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihkan media/format, mengelola dalam bentuk pangkalan database, memelihara, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan

Pada Tanggal : 17 September 2025

Yang menyatakan



Ari Rama Gunanda  
198160051

## ABSTRAK

Penyakit pada tanaman padi merupakan ancaman serius bagi ketahanan pangan global, dengan contoh historis seperti kelaparan besar di Bangladesh pada tahun 1940-an yang menunjukkan dampak buruk dari penyakit tanaman yang tidak terkendali. Meskipun teknologi telah berkembang, banyak petani masih kesulitan dalam mengidentifikasi dan menangani penyakit padi secara akurat, sering kali hanya mengandalkan saran dari sesama petani yang belum tentu tepat. Penelitian ini bertujuan untuk mengembangkan sistem otomatis untuk klasifikasi penyakit tanaman padi melalui citra daun menggunakan model deep learning VGG-16. Dataset yang digunakan terdiri dari 15.000 gambar latih dan 3.400 gambar validasi yang mencakup delapan jenis penyakit padi yang umum: Bacterial Leaf Blight, Brown Spot, Leaf Blast, Leaf Scald, Narrow Brown Spot, Rice Hispa, Sheath Blight, dan Tungro. Sebanyak delapan model VGG-16 dilatih dengan variasi parameter seperti batch size, jumlah epoch, dan jenis optimizer (Adam dan AdamW) untuk menentukan konfigurasi terbaik. Evaluasi model dilakukan menggunakan confusion matrix dan classification report. Hasil penelitian menunjukkan bahwa VGG-16 mampu mengklasifikasikan penyakit padi dengan akurasi tinggi, serta menunjukkan performa yang lebih baik dibandingkan model seperti AlexNet. Penelitian ini tidak hanya berkontribusi pada pengembangan teknologi pertanian, tetapi juga mendukung pembuatan aplikasi berbasis mobile yang memungkinkan petani untuk mendiagnosis penyakit secara cepat dan akurat. Pada akhirnya, sistem ini diharapkan dapat menjadi dasar pengembangan alat bantu keputusan dalam manajemen penyakit padi yang lebih efisien dan efektif.

**Kata Kunci:** VGG-16, CNN, Klasifikasi penyakit tanaman padi

## ABSTRACT

*Rice plant diseases pose a significant threat to global food security, with historical examples such as the 1940s famine in Bangladesh highlighting the devastating effects of uncontrolled plant diseases. Despite technological advances, many farmers still struggle to accurately identify and manage rice diseases, often relying on peer suggestions that may not be scientifically accurate. To address this issue, this study aims to develop an automated system for classifying rice plant diseases through leaf images using the VGG-16 deep learning model. The dataset used consists of 15,000 training and 3,400 validation images across eight common rice diseases: Bacterial Leaf Blight, Brown Spot, Leaf Blast, Leaf Scald, Narrow Brown Spot, Rice Hispa, Sheath Blight, and Tungro. Eight VGG-16 models were trained with variations in batch size, number of epochs, and optimizer types (Adam and AdamW) to identify the best-performing configuration. Model evaluation was conducted using confusion matrices and classification reports. The results show that VGG-16 is capable of classifying rice diseases with high accuracy, outperforming other models such as AlexNet. This research not only contributes to the field of agricultural technology but also supports the development of mobile-based applications that enable farmers to diagnose diseases quickly and accurately. Ultimately, the proposed system can serve as a foundation for a more efficient and effective decision support tool in rice disease management.*

**Keywords: VGG-16, CNN, Classification of Diseases in Rice Plants**

## KATA PENGANTAR

Puji dan Syukur penulis panjatkan kepada Allah SWT atas segala limpaham rahmat dan kasih sayang-Nya sehingga penulis dapat melaksanakan penelitian dan menyelesaikan skripsi ini dengan baik. Adapun judul skripsi ini adalah “**Klasifikasi Penyakit Pada Tumbuhan Padi Melalui Citra Daun Dengan Pemanfaatan Model VGG-16**”. Skripsi ini merupakan salah satu syarat untuk dapat memperoleh gelar Sarjana di Program Studi Informatika, Fakultas Teknik, Universitas Medan Area.

Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Kepada orang tua tercinta, Bapak Suparmin dan Ibunda Suprianti yang senantiasa memberikan kasih sayang, semangat, motivasi, nasihat dan doa yang tiada putusnya serta dukungan baik secara moril dan materil yang tiada hentinya hingga penulis dapat menyelesaikan pendidikan sarjana di program Studi Informatika, Fakultas Teknik, Universitas Medan Area.
2. Kepada Ibu Susilawati S.kom,M.kom. selaku Pembimbing yang telah meluangkan waktunya untuk memberikan bimbingan, nasihat, arahan, dan motivasi sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
3. Kepada seluruh Dosen fakultas Teknik khususnya Teknik Informatika, Universitas Medan Area yang telah memberikan ilmu yang bermanfaat kepada penulis selama masa perkuliahan.
4. Kepada teman seperjuangan saya Anjeli Septiyana yang telah memberikan doa, dukungan, motivasi dan semangat kepada penulis untuk dapat menyelesaikan skripsi ini.

5. Kepada teman-teman penulis di program studi Informatika khususnya angkatan 2019 untuk doa dan dukunganya yang telah menemani dan memberikan semangat kepada penulis selama perkuliahan.

Penulis menyadari skripsi ini masih jauh dari kata sempurna. Untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun dari semua pihak untuk kesempurnaan skripsi ini. Penulis berharap skripsi ini bermanfaat bagi kita semua. Akhir kata penulis mengucapkan terima kasih.



Medan, 17 September 2025

Penulis

## RIWAYAT HIDUP

ARI RAMA GUNANDA lahir di Medan, 05 Desember 2000 anak dari Bapak Suparmin dan Ibu Suprianti yang merupakan anak pertama dari tiga bersaudara.

Pendidikan Formal yang pernah ditempuh adalah sebagai berikut:

1. Lulus dari Sekolah Dasar Negri 050577 pada tahun 2013.
2. Pada tahun 2013 diterima di Sekolah Menengah Pertama di SMP Negri 1 Kecamatan Binjai, Kabupaten Langkat dan lulus pada tahun 2016.
3. Pada tahun 2016 di terima Sekolah Menengah Kejuruan di SMK Tunas Pelita Binjai, dan lulus pada tahun 2019.
4. Pada tahun 2019 menempuh pendidikan sarjana (S1) di Program Studi Informatika, Fakultas Teknik Informatika, Universitas Medan Area.

## DAFTAR ISI

|                             |      |
|-----------------------------|------|
| <b>ABSTRACT</b> .....       | vi   |
| <b>KATA PENGANTAR</b> ..... | viii |
| <b>RIWAYAT HIDUP</b> .....  | x    |
| <b>DAFTAR ISI</b> .....     | xi   |
| <b>DAFTAR GAMBAR</b> .....  | xiv  |
| <b>DAFTAR TABEL</b> .....   | xvi  |

|                                              |    |
|----------------------------------------------|----|
| <b>BAB I PENDAHULUAN</b> .....               | 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 Sistematis Penulisan.....                | 5  |
| <b>BAB II TINJAUAN PUSTAKA</b> .....         | 7  |
| 2.1 Tanaman Padi.....                        | 7  |
| 2.2 Penyakit Padi.....                       | 7  |
| 2.2.1 <i>Bacterial leaf blight</i> .....     | 7  |
| 2.2.2 <i>Brown spot</i> .....                | 8  |
| 2.2.3 <i>Leaf blast and neck blast</i> ..... | 9  |
| 2.2.4 <i>Leaf scald</i> .....                | 9  |
| 2.2.5 <i>Narrow brown Spot</i> .....         | 10 |
| 2.2.6 <i>Rice Hispa</i> .....                | 11 |
| 2.2.7 <i>Sheat Blight</i> .....              | 12 |
| 2.2.8 <i>Tungro</i> .....                    | 12 |
| 2.3 CNN .....                                | 13 |
| 2.3.1 <i>Convolutional Layer</i> .....       | 13 |
| 2.3.2 <i>Activation Function</i> .....       | 15 |
| 2.3.3 <i>Pooling Layer</i> .....             | 16 |
| 2.3.4 <i>Flatten Layer</i> .....             | 17 |

|                                                  |    |
|--------------------------------------------------|----|
| 2.3.5 <i>Fully-Connected Layer</i> .....         | 18 |
| 2.4 VGG16 .....                                  | 18 |
| 2.5 <i>Python</i> .....                          | 19 |
| 2.5.1 <i>Tensorflow</i> .....                    | 19 |
| 2.5.2 <i>Keras</i> .....                         | 20 |
| 2.6 Android Studio .....                         | 20 |
| 2.7 Citra Digital .....                          | 21 |
| 2.8 Penelitian Terdahulu .....                   | 21 |
| <b>BAB III METODOLOGI PENELITIAN</b> .....       | 24 |
| 3.1 Waktu dan Tempat Penelitian .....            | 24 |
| 3.2 Alat dan Bahan Penelitian .....              | 24 |
| 3.3 Dataset .....                                | 25 |
| 3.4 Tahapan Penelitian .....                     | 26 |
| 3.4.1 Data Collection .....                      | 27 |
| 3.4.2 Data Preparation .....                     | 27 |
| 3.4.3 Model Training .....                       | 27 |
| 3.4.4 Pembagian data .....                       | 30 |
| 3.4.5 Augmentasi Dataset .....                   | 31 |
| 3.4.6 Pemodelan Arsitektur VGG-16 .....          | 34 |
| 3.4.7 Analisa Model Dengan Peforma Terbaik ..... | 36 |
| 3.4.8 Deploy Mobile App for the Model .....      | 37 |
| <b>BAB IV HASIL DAN PEMBAHASAN</b> .....         | 40 |
| 4.1 Hasil .....                                  | 40 |
| 4.2 Pembahasan .....                             | 50 |
| <b>BAB V KESIMPULAN DAN SARAN</b> .....          | 68 |
| 5.1 Kesimpulan .....                             | 68 |
| 5.2 Saran .....                                  | 69 |

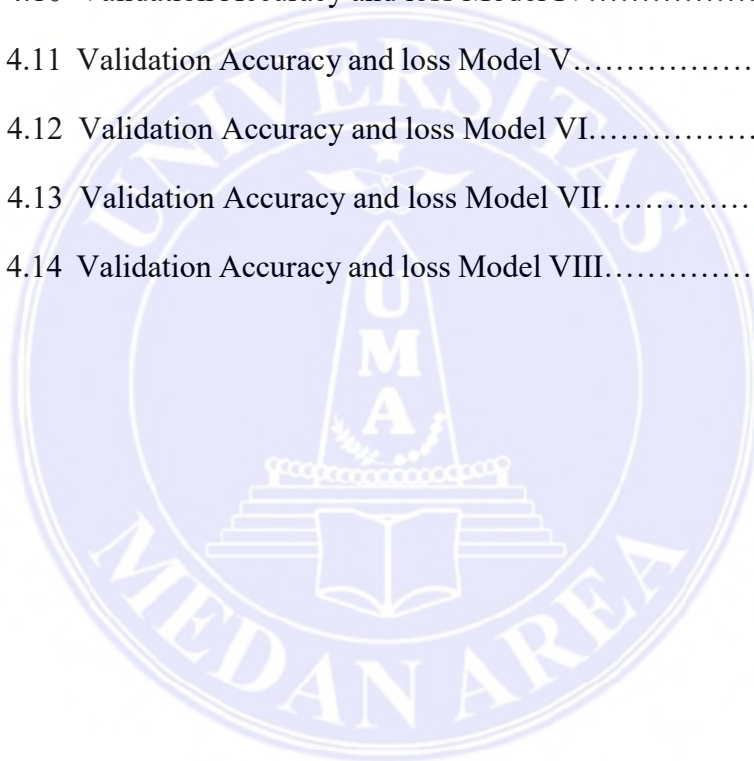
|                      |           |
|----------------------|-----------|
| <b>LAMPIRAN.....</b> | <b>72</b> |
|----------------------|-----------|



## DAFTAR GAMBAR

|             |                                                |    |
|-------------|------------------------------------------------|----|
| Gambar 2.1  | Bacterial leaf blight .....                    | 8  |
| Gambar 2.2  | Brown Spot .....                               | 8  |
| Gambar 2.3  | Leaf Blast .....                               | 9  |
| Gambar 2.4  | Leaf Scald .....                               | 10 |
| Gambar 2.5  | Narrow Brown Spot .....                        | 11 |
| Gambar 2.6  | Rice Hispa .....                               | 11 |
| Gambar 2.7  | Sheat Blight .....                             | 12 |
| Gambar 2.8  | Tungro .....                                   | 13 |
| Gambar 2.9  | Convolutional Layer .....                      | 14 |
| Gambar 2.10 | Relu .....                                     | 16 |
| Gambar 2.11 | Ilustrasi Maxpooling dan Avarage pooling ..... | 17 |
| Gambar 2.12 | Ilustrasi Flatten Layer .....                  | 18 |
| Gambar 2.13 | VGG-16 .....                                   | 19 |
| Gambar 3.1  | Dataset Daun Padi .....                        | 26 |
| Gambar 3.2  | Tahapan Penelitian .....                       | 27 |
| Gambar 3.3  | Augmentasi Horizontal Flip .....               | 32 |
| Gambar 3.4  | Augmentasi Rotation Range .....                | 32 |
| Gambar 3.5  | Augmentasi Shear Range .....                   | 33 |
| Gambar 3.6  | Augmentasi Zoom Range .....                    | 33 |
| Gambar 3.7  | Rancangan Arsitektur VGG-16 .....              | 34 |
| Gambar 3.8  | Tahapan Analisa .....                          | 37 |
| Gambar 3.9  | Flowchard Aplikasi.....                        | 38 |
| Gambar 3.10 | Perancangan UI/UX.....                         | 38 |
| Gambar 4.1  | Confusion matrix Model 5.....                  | 41 |
| Gambar 4.2  | Confusion Matrix Model 6.....                  | 43 |

|             |                                              |    |
|-------------|----------------------------------------------|----|
| Gambar 4.3  | Confusion matrix Model 7.....                | 45 |
| Gambar 4.4  | Confusion matrix Model 8.....                | 46 |
| Gambar 4.5  | Confusion matrix Model 6.....                | 49 |
| Gambar 4.6  | Gambar Tampilan utama aplikasi.....          | 50 |
| Gambar 4.7  | Validation Accuracy and loss Model I.....    | 52 |
| Gambar 4.8  | Validation Accuracy and loss Model II.....   | 54 |
| Gambar 4.9  | Validation Accuracy and loss Model III.....  | 56 |
| Gambar 4.10 | Validation Accuracy and loss Model IV.....   | 58 |
| Gambar 4.11 | Validation Accuracy and loss Model V.....    | 60 |
| Gambar 4.12 | Validation Accuracy and loss Model VI.....   | 62 |
| Gambar 4.13 | Validation Accuracy and loss Model VII.....  | 64 |
| Gambar 4.14 | Validation Accuracy and loss Model VIII..... | 66 |



## DAFTAR TABEL

|            |                                            |    |
|------------|--------------------------------------------|----|
| Tabel 2.1  | Penelitian Terdahulu .....                 | 21 |
| Tabel 3.1  | Kebutuhan Perangkat Keras Komputer .....   | 24 |
| Tabel 3.2  | Kebutuhan Perangkat Keras Smartphone ..... | 24 |
| Tabel 3.3  | Kebutuhan Perangkat Lunak .....            | 25 |
| Tabel 3.4  | Hyperparameter .....                       | 28 |
| Tabel 3.5  | Pembagian dataset .....                    | 30 |
| Tabel 3.6  | Konfigurasi Augmentasi .....               | 31 |
| Tabel 4.1  | Perbandingan akursi model .....            | 40 |
| Tabel 4.2  | Classification Report Model 5.....         | 42 |
| Tabel 4.3  | Classification Report Model 6.....         | 43 |
| Tabel 4.4  | Classification Report Model 7.....         | 45 |
| Tabel 4.5  | Classification Report Model 8.....         | 47 |
| Tabel 4.6  | Tabel Tes Accuracy model.....              | 47 |
| Tabel 4.7  | Confusion matrix Model 6.....              | 48 |
| Tabel 4.8  | Confusion matrix Model 6 TFLite.....       | 49 |
| Tabel 4.9  | Hyperparameter I.....                      | 51 |
| Tabel 4.10 | training performance Model I.....          | 51 |
| Tabel 4.11 | Hyperparameter II.....                     | 53 |
| Tabel 4.12 | training performance Model II.....         | 53 |
| Tabel 4.13 | Hyperparameter III.....                    | 55 |
| Tabel 4.14 | training performance Model III.....        | 55 |
| Tabel 4.15 | Hyperparameter IV.....                     | 56 |
| Tabel 4.16 | training performance Model IV.....         | 57 |
| Tabel 4.17 | Hyperparameter V.....                      | 58 |
| Tabel 4.18 | training performance Model V.....          | 59 |

|                                                           |    |
|-----------------------------------------------------------|----|
| Tabel 4.19 Hyperparameter VI.....                         | 60 |
| Tabel 4.20 training performance Model VI.....             | 61 |
| Tabel 4.21 Hyperparameter VII.....                        | 63 |
| Tabel 4.22 training performance Model VII.....            | 63 |
| Tabel 4.23 Hyperparameter VII.....                        | 65 |
| Tabel 4.24 training performance Model VIII.....           | 65 |
| Tabel 4.25 Perbandingan akurasi penelitian terdahulu..... | 67 |



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Padi (*Oryza sativa* L.) merupakan tanaman pangan yang memiliki nilai ekonomi penting di banyak negara berkembang khususnya di kawasan Asia Selatan, Sekitar 50% dari total lahan pertanian di Asia Selatan digunakan untuk sistem tanam berbasis padi. Padi merupakan salah satu tanaman pangan pokok terpenting di India dan banyak negara lain di Asia. Potensi hasil panen padi sangat dipengaruhi oleh berbagai tekanan biotik, terutama oleh penyakit seperti *bakanae*, *rice blast*, *bacterial leaf blight* (BLB), *sheath blight*, dan *brown spot*, yang merupakan penyakit penting di wilayah penghasil padi di seluruh dunia (M, Kumar, & Kumar, 2024).

Identifikasi penyakit tanaman padi menggunakan bantuan *Artificial Intelligence* (AI) dapat menjadi solusi dalam mencegah dan memberikan tindakan terhadap penyakit atau hama tanaman padi dengan memberikan informasi tentang penyakit dan hama yang menyerang tanaman padi tersebut, metode *Deep Learning CNN* dapat diimplementasikan untuk identifikasi citra daun yang berpenyakit (Hawari dkk., 2025).

Salah satu model CNN yang dapat digunakan untuk mengidentifikasi penyakit tanaman padi adalah VGG-16 model CNN ini adalah arsitektur jaringan saraf tiruan yang terdiri dari 16 lapisan konvolusi yang terhubung penuh. Model ini menggunakan serangkaian lapisan konvolusi dengan filter kecil (3x3) yang diikuti oleh lapisan aktivasi ReLU untuk mengekstraksi fitur dalam hierarki yang semakin tinggi, Kemudian *subsampling* menggunakan lapisan *max-pooling* untuk

mengurangi dimensi fitur, Setelah beberapa lapisan konvolusi dan *max-pooling*, komponen diubah menjadi vektor satu dimensi dan digabungkan melalui beberapa lapisan yang terhubung penuh. Akhirnya lapisan output melakukan klasifikasi menggunakan fungsi aktivasi *softmax*.

VGG-16 dikenal karena kedalamannya yang besar dan kemampuannya untuk melakukan pengenalan objek secara akurat dalam berbagai tugas pemrosesan gambar (Cipta dkk., 2023).

Penggunaan VGG16 dibanding *Alexnet* juga didasari oleh performa dari model VGG16 yang lebih baik dibandingkan *Alexnet*, Metode VGG16 memiliki tingkat keberhasilan tinggi dibandingkan dengan *Alexnet*, dimana VGG 16 memiliki tingkat keberhasilan 85% dalam mendeteksi gambar, sedangkan *Alexnet* dengan tingkat keberhasilan 70% (Cipta dkk., 2023).

Menurut GoodStats (2023), jumlah pengguna smartphone di Indonesia mencapai 209,3 juta orang pada tahun 2023. Selain itu, data Statista (2023) menunjukkan bahwa sekitar 67,29% populasi nasional memiliki ponsel, bahkan di daerah rural sekitar 59,29% penduduk sudah menggunakan mobile phone. Berdasarkan fakta tersebut, smartphone bukanlah barang langka di masyarakat, termasuk di kalangan petani, sehingga pemanfaatan perangkat Android dalam identifikasi penyakit padi merupakan pilihan yang realistis sekaligus relevan. Dengan memanfaatkan keunggulan teknologi *deep learning* yang dikombinasikan dengan *mobile device* diharapkan penelitian ini dapat memberikan kontribusi dalam percepatan proses identifikasi penyakit pada tanaman padi dan menjadi opsi dalam identifikasi penyakit padi.

Selain itu hasil penelitian ini juga diharapkan dapat menjadi dasar bagi pengembangan sistem pendukung keputusan untuk manajemen penyakit tanaman padi yang lebih efisien dan efektif.

## 1.2 Rumusan Masalah

Berdasarkan dari latar belakang diatas, maka terdapat beberapa rumusan masalah yang dikaji dalam penelitian ini yaitu “

1. Bagaimana Hasil Klasifikasi penyakit pada tumbuhan padi melalui citra daun dengan pemanfaatan model VGG16 pada perangkat *mobile*”,
2. Bagaimana tingkat akurasi yang dihasilkan arsitektur Deep Learning VGG-16 dalam mengklasifikasi jenis penyakit tanaman padi.

## 1.3 Batasan Masalah

Agar dalam penelitian dan penulisan tidak keluar dari tema permasalahan, maka dibutuhkan batasan-batasan masalah sebagai berikut:

1. Sistem dapat mengenali 8 jenis penyakit padi yang umum dijumpai antara lain: *Bacterial leaf blight, brown spot, leaf blast, leaf scald, narrow brown spot, rice hispa, sheath blight, dan tungro*.
2. Model dilatih menggunakan dataset yang berasal dari Kaggle dengan total 15,000 gambar latih dan 3,400 gambar validasi dengan 8 kelas penyakit padi
3. Terdapat 8 model yang akan dilatih dengan parameter yang berbeda yaitu *Batch size* (16 dan 32), jumlah Epoch (20 dan 30), *Optimizer* (Adam dan AadamW) serta *learning rate* statis 0,00005.
4. Uji akurasi dilakukan menggunakan *confusion matrix* dan *classification report* untuk menghitung tingkat akurasi yang didapatkan oleh masingmasing

skenario model, sehingga dapat ditentukan performa model terbaik yang selanjutnya akan dikembangkan dalam aplikasi berbasis Android.

#### 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem otomatis untuk mengidentifikasi penyakit pada tanaman padi melalui analisis citra daun menggunakan model VGG-16, Menganalisa performa dari tingkat akurasi model dari arsitektur *Deep Learning* VGG-16 sehingga dapat menghasilkan klasifikasi jenis penyakit padi secara akurat.

#### 1.5 Manfaat Penelitian

Adapun beberapa manfaat yang dapat diambil dari hasil penelitian ini yaitu sebagai berikut

1. Manfaat Teoritis
  - a) Memberikan kontribusi terhadap ilmu pengetahuan, khususnya dalam bidang terkait untuk meningkatkan proses pembelajaran yang baik.
  - b) Hasil penelitian ini dapat menjadi acuan bagi penelitian lain yang fokus pada topik yang sama.
2. Manfaat Praktis

Manfaat bagi Peneliti dan Masyarakat Umum adalah dapat menambah wawasan tentang model VGG-16, dan bagi Institusi Melalui penulisan ini diharapkan nantinya dapat digunakan sebagai tambahan referensi kepustakaan terutama dalam upaya pengimplementasian model VGG-16 dalam mengenali penyakit padi.

## 1.6 Sistematis Penulisan

### **BAB I            PENDAHULUAN**

Bab ini membahas beberapa hal umum dalam penelitian, seperti latar belakang dari sebuah permasalahan yang diangkat, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, metodologi penelitian, dan sistematis penulisan.

### **BAB II           TINJAUAN PUSTAKA**

Bab ini menjelaskan beberapa materi dan teori yang di butuhkan dalam melaksanakan penelitian ini.

### **BAB III          METODE PENELITIAN**

Bab ini menjelaskan tentang metode yang di gunakan untuk mendapatkan data dan metode untuk pengembangan sistem yang telah di buat serta kerangka berfikir tugas akhir.

### **BAB IV          HASIL DAN PEMBAHASAN**

Bab ini berisi hasil penelitian serta pembahasan dari penelitian yang telah dilaksanakan. Dalam bab ini disajikan gambar, tabel, serta grafik dari hasil penelitian yang telah dilakukan.

### **BAB V           KESIMPULAN DAN SARAN**

Bab ini merupakan kesimpulan dari keseluruhan pembahasan terutama hasil penelitian yang telah dijelaskan pada bab sebelumnya. Bab ini juga berisi saran untuk penelitian

selanjutnya yang akan mengangkat tema yang sama dengan penelitian ini.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tanaman Padi

Padi (*Oryza sativa*) merupakan tanaman pangan pokok yang menyediakan pangan utama bagi hampir setengah populasi dunia. Tanaman ini sangat bergantung pada kondisi lingkungan yang mendukung seperti cahaya, suhu, kelembapan, dan ketersediaan air. (Jin & Ding, 2025).

Selain itu, adaptasi daun seperti perkembangan daun, pembentukan primordia daun, dan pertumbuhan daun yang dipengaruhi oleh polaritas dan ekspansi daun merupakan mekanisme penting yang mempengaruhi produktivitas biji padi. (Gong dkk., 2024).

#### 2.2 Penyakit Padi

Padi sebagai makanan pokok didarat Asia menjadikannya komoditas yang sangat penting bagi masyarakat Asia, namun tanaman yang cukup penting ini juga tidak bebas dari berbagai hama dan penyakit, penyakit padi di antaranya ialah:

##### 2.2.1 *Bacterial leaf blight*

*Bacterial Leaf Blight (BLB)* adalah penyakit padi yang disebabkan oleh bakteri *Xanthomonas oryzae* pv. *Oryzae* (Xoo) yang menyerang daun dan dapat menurunkan hasil panen. Gejala penyakit ini ditandai dengan munculnya bercak berbentuk garis atau bercak air pada daun, yang lama-kelamaan berubah menjadi kuning hingga coklat, dan daun yang terinfeksi akhirnya mengering (Subburaj dkk., 2024). gambar daun yang terjangkit Bacterial leaf bligh dapat di lihat pada Gambar 2.1 di bawah.



**Gambar 2.1** *Bacterial leaf blight*

### **2.2.2 Brown spot**

Penyakit ini disebabkan oleh jamur *Cochliobolus miyabeanus*, dengan gejala awal berupa bercak-bercak kecil berwarna coklat keabu-abuan pada daun padi. Bercak ini biasanya muncul dari bagian bawah daun dan dapat menyebar ke atas. Bercak-bercak tersebut memiliki tepi gelap di sekelilingnya, sering kali dengan pusat bercak yang lebih terang. Daun yang terinfeksi dapat mengalami kekeringan dan akhirnya mengering, yang dapat mengurangi kemampuan fotosintesis (Bhutia dkk., 2025), tanaman gambar daun yang terjangkit *Brown spot* dapat di lihat pada Gambar 2.2 di bawah.



**Gambar 2.2** *Brown spot*

### 2.2.3 Leaf blast and neck blast

penyakit blast pada padi disebabkan oleh jamur *Magnaporthe oryzae* yang dikenal sebagai salah satu patogen udara paling merusak pada tanaman padi. Patogen ini mampu menginfeksi hampir semua bagian tanaman, mulai dari daun hingga malai, dengan membentuk struktur khusus berupa *appressorium* yang dapat menembus kutikula daun melalui tekanan turgor tinggi. Infeksi tersebut menghasilkan gejala bercak berbentuk spindle pada daun dan dapat mengganggu proses fotosintesis sehingga menurunkan produktivitas tanaman (Tan dkk., 2023), gambar daun yang terjangkit *Leaf blast* dapat di lihat pada Gambar 2.3 di bawah.



**Gambar 2.3** Leaf blast dan neck blast

### 2.2.4 Leaf scald

*Leaf Scald* adalah salah satu penyakit penting pada padi yang disebabkan oleh jamur *Monographella albescens* (sinonim *Microdochium oryzae*). Penyakit ini menyerang daun pada fase pertumbuhan lanjut, dengan gejala berupa bercak memanjang dari ujung atau tepi daun yang menunjukkan pola bergaris antara warna terang dan coklat gelap. Seiring perkembangan penyakit, bercak-bercak tersebut meluas dan dapat menyebabkan bilah daun mengering serta tampak seperti terbakar, sehingga

berpotensi menurunkan produktivitas tanaman padi (Murudkar dkk., 2023). Contoh daun yang terjangkit *Leaf scald* dapat di lihat pada Gambar 2.4 di bawah.



**Gambar 2.4** Leaf scald

#### **2.2.5 Narrow brown Spot**

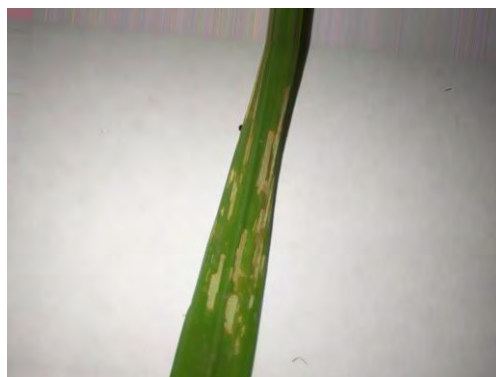
*Narrow Brown Leaf Spot* (NBLS) adalah penyakit padi yang disebabkan oleh jamur *Cercospora janseana* (syn. *Passalora janseana*) yang menyerang daun, pelepah daun, dan bagian atas tanaman padi pada fase mendekati generatif. Gejalanya berupa lesi linear memanjang sejajar tulang daun, berwarna coklat sampai coklat tua, biasanya kira-kira 2-10 mm panjang dan 1-1,5 mm lebar, lesi ini bisa melebar atau menyatu pada varietas yang rentan sehingga menyebabkan *nekrosis linear* dan pengeringan daun (Sofyaningrum dkk., 2024). Contoh daun yang terjangkit *Narrow brown spot* dapat di lihat pada Gambar 2.5 di bawah.



**Gambar 2.5** Narrow Brown Spot

### 2.2.6 Rice Hispa

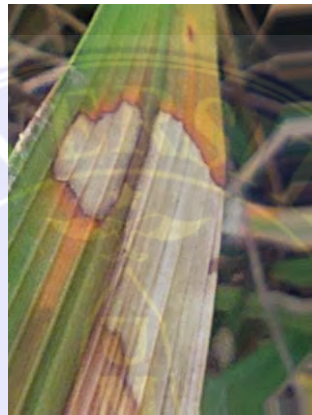
*Rice hispa* (*Dicladispa armigera*) merupakan hama penting padi yang menyerang fase vegetatif dan dapat menurunkan hasil sebesar 35–65%, bahkan hingga 100% pada kondisi tertentu, sehingga menjadi faktor pembatas utama dalam produksi padi. Upaya pengendalian hayati dengan jamur entomopatogen *Beauveria bassiana* terbukti efektif, di mana media sekam padi dengan tambahan dekstrosa serta kaldu kentang yang diperkaya kitosan mampu meningkatkan produksi spora dan *patogenisitas* terhadap imago *D. armigera* (Das dkk., 2021). Contoh daun yang terjangkit *Rice hispa* dapat dilihat pada Gambar 2.6 di bawah.



**Gambar 2.6** Rice Hispa

### 2.2.7 Sheat Blight

*Sheat Blight* penyakit utama pada tanaman padi yang disebabkan oleh jamur *Rhizoctonia solani*. Penyakit ini dapat menurunkan hasil panen secara signifikan, terutama di daerah yang lembap dan bersuhu tinggi (Li, Li, Wei, & Sun, 2021). Contoh daun yang terjangkit Sheat Blight dapat di lihat pada Gambar 2.7 di bawah.



**Gambar 2.7** Sheat Blight

### 2.2.8 Tungro

*Tungro* merupakan salah satu penyakit tanaman padi yang sangat merugikan. *Tungro* sendiri merupakan penyakit yang disebabkan oleh kombinasi dari dua virus yaitu *Rice Tungro Bacilliform Virus* (RTBV) dan *Rice Tungro Spherical Virus* (RTSV), penyakit ini biasa ditularkan oleh Wareng hijau (Widiarta dkk., 2025). Gambar penyakit yang terjangkit *Tungro* dapat dilihat pada Gambar 2.8 di bawah.



**Gambar 2.8** Tungro

## 2.3 CNN

Convolutional Neural Network (CNN) adalah jenis algoritma deep learning yang merupakan pengembangan dari multilayer perceptron. CNN merupakan jenis algoritma yang umum digunakan untuk melakukan klasifikasi ataupun pengenalan object pada data berupa gambar (Kotta dkk., 2022). Algoritma CNN sering digunakan dalam pengenalan citra karena akurasi yang tinggi dan sangat baik dalam pengenalan gambar visual. Algoritma CNN dapat menerima input seperti gambar kemudian mengenali objek-objek yang ada pada sebuah gambar, melalui objek-objek tersebut mesin belajar mengenali gambar dan membedakan antara setiap gambar dengan gambar yang lainnya (Asseweth, 2024).

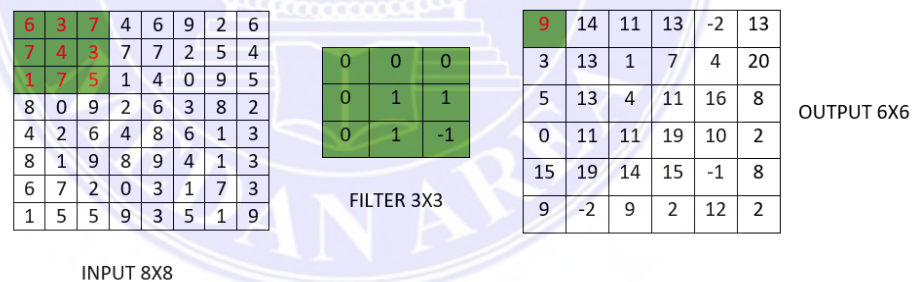
### 2.3.1 Convolutional Layer

Lapisan Konvolusional (*Convolutional layer*) merupakan inti lapisan dari sebuah CNN, sebagian besar proses komputasi dilakukan pada lapisan ini, tujuan lapisan ini ialah untuk mengekstraksi fitur yang ada pada gambar yang di gunakan, (Radana, 2022), Proses ekstraksi fitur ini adalah dengan

menggeser filter pada atas gambar secara bertahap, setiap filter memiliki bobot dan nilai bias yang sama diseluruh gambar, setelah filter ini melakukan skaning pada keseluruhan gambar, lapisan konvolusi akan menghasilkan gambar baru yang di sebut peta fitur (feature map), (Radana, 2022), Operasi konvolusi pada citra dengan lebih dari satu dimensi, dapat digunakan rumus pada persamaan 2.1 berikut.

$$S_{(i,j)} = (I * K)_{(i,j)} = \sum_m \sum_n I_{(i-m,j-n)} K_{(m,n)} \quad (2.1)$$

Untuk lebih mudah memahami proses penghitungan konvolusi menggunakan filter 3 x 3 yang dilakukan dengan menggunakan persamaan diatas dapat diilustrasikan pada Gambar 2.9



**Gambar 2.9** Convolutional Layer

Pada Gambar 2.9 citra berukuran 8x8 dikonvolusikan dengan filter berukuran 3x3, hasil dari proses konvolusi ini ialah citra berukuran 6x6, proses ini dilakukan dengan melakukan perkalian antara nilai filter atau kernel dengan citra yang difilter, hasil perkalian akan menjadi nilai dari citra output, setelah perkalian citra pertama selesai, maka filter akan bergerak ke

kanan dan melakukan perkalian ulang hingga seluruh nilai citra pada input 8x8 selesai dikalikan seutuhnya dan menghasilkan citra baru yang berukuran 6x6.

### 2.3.2 Activation Function

Agar hasil akhir lebih akurat maka akan diperlukan fungsi aktivasi (*Activation Function*), fungsi aktivasi merupakan node yang diletakan diakhir atau di antara *neural network*, fungsi aktivasi yang paling banyak digunakan dalam metode CNN adalah fungsi aktivasi ReLU dan fungsi aktivasi *Softmax*. (Syuhada, 2023).

#### 1. *Function Rectified Linear Unit (ReLU)*

ReLU merupakan fungsi yang sangat populer digunakan, ReLU berfungsi untuk mengubah bilangan *negative* menjadi 0, artinya jika feature map yang dihasilkan oleh convolution layer memiliki *value* dibawah 0, maka ReLU akan mengubah nilai tersebut menjadi 0, berikut merupakan persamaan ReLU.

$$f(x) = \max(0, x) \quad (2.2)$$

Fungsi aktivasi ReLU dibandingkan dengan fungsi *Sigmoid* atau fungsi aktivasi serupa, memungkinkan pelatihan yang lebih cepat dan efektif untuk arsitektur syaraf pada kumpulan data yang besar dan kompleks, contoh bagaimana ReLu mengubah bilangan yang tadinya negative menjadi 0 dapat dilihat pada Gambar 2.10.

|    |   |    |    |
|----|---|----|----|
| 1  | 0 | -1 | 4  |
| 4  | 0 | -4 | -7 |
| 1  | 3 | -2 | 0  |
| -4 | 2 | 0  | 0  |

**ReLU**

|   |   |   |   |
|---|---|---|---|
| 1 | 0 | 0 | 4 |
| 4 | 0 | 0 | 0 |
| 1 | 3 | 0 | 0 |
| 0 | 2 | 0 | 0 |

Gambar 2.10 ReLU

Gambar 2.10 di atas menunjukan bagaimana ReLU mengubah *value negative* ke 0.

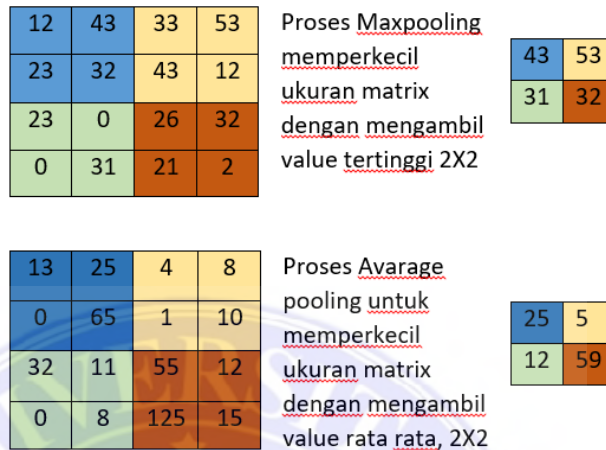
2. Fungsi aktivasi *Softmax* Fungsi *Softmax* pada CNN biasanya digunakan pada *fully connected layer* untuk klasifikasi. Secara umum, fungsi ini akan menghitung probabilitas masing-masing kelas target diatas semua kelas sasaran yang mungkin ada. Kemudian probabilitas yang dihitung akan sangat membantu untuk menentukan kelas target untuk input yang diberikan. (Syuhada, 2023), berikut merupakan persamaan untuk *Softmax*.

$$f(x) = \frac{e^{x_i}}{\sum_{j=0}^k e^{x_j}} \quad (2.3)$$

### 2.3.3 Pooling Layer

*Feature map* yang telah melalui proses konvolusi dan Aktivasi akan memasuki proses *pooling* (Radana, 2022), *Pooling layer* berfungsi untuk memperkecil ukuran pada saat konvolusi, dan melakukan *down sampling*, sehingga data dikembangkan menjadi bagian yang lebih kecil, mudah dikelola, dan mudah dalam mengatur *overfitting* (Syuhada, 2023), ilustrasi

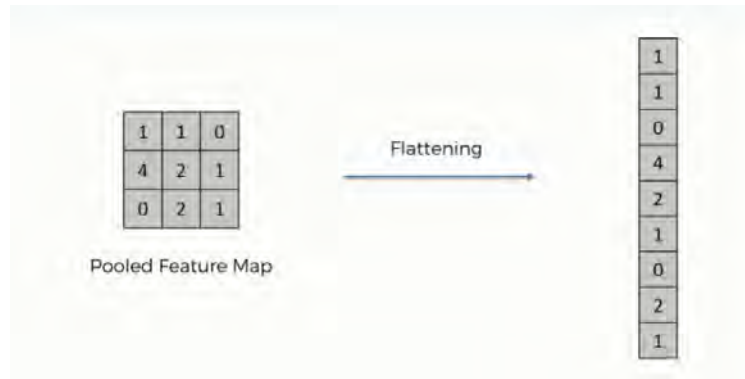
dari proses *Maxpooling* dan *Average pooling* dapat dilihat pada Gambar 2.11 di bawah.



Gambar 2.11 Ilustrasi *Maxpooling* dan *Average pooling*

### 2.3.4 Flatten Layer

*Flatten layer* pada *Convolutional Neural Network* (CNN) berfungsi untuk merubah output berdimensi banyak dari lapisan konvolusi / pooling menjadi vektor satu dimensi agar dapat dihubungkan ke lapisan *fully connected*. Selain sebagai jembatan antara ekstraksi fitur spasial dan klasifikasi, flatten layer juga sering menjadi titik krusial dalam optimasi arsitektur, terutama untuk mengurangi jumlah parameter yang sangat besar yang muncul saat banyak fitur diekstraksi. Penelitian terbaru menunjukkan bahwa dengan melakukan *pruning* pada flattening layer—mengeliminasi bagian yang dinilai kurang berdampak terhadap hasil akhir—dapat mengurangi ukuran model secara signifikan sambil menjaga akurasi mendekati model penuh (Jeczminek & Kowalski, 2021), ilustrasi dari Flatten layer dapat dilihat pada Gambar 2.12 di bawah.



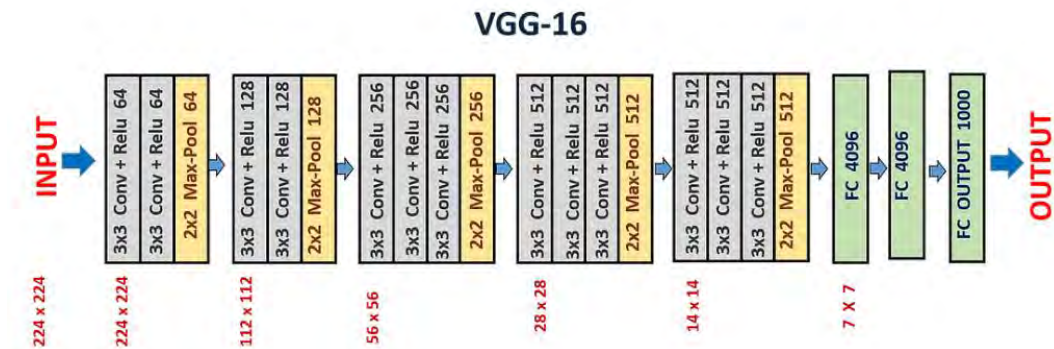
Gambar 2.12 Ilustrasi Flatten Layer

### 2.3.5 Fully-Connected Layer

*Fully connected layer* adalah lapisan dalam jaringan syaraf tiruan di mana semua neuron dalam lapisan tersebut terhubung (*connected*) ke semua neuron di lapisan sebelumnya. Lapisan ini biasanya ditempatkan di bagian akhir model CNN sebagai pengklasifikasi yang menggabungkan fitur-fitur yang telah diekstrak oleh lapisan-convolution dan pooling untuk menghasilkan prediksi (Matsumura dkk.,2021).

## 2.4 VGG16

Jaringan saraf konvolusional atau juga dikenal sebagai *ConvNet*, yang merupakan sejenis jaringan saraf tiruan. Jaringan saraf konvolusional memiliki lapisan masukan, lapisan keluaran, dan berbagai lapisan tersembunyi, VGG-16 adalah jenis CNN, VGG-16 adalah arsitektur jaringan saraf konvolusional yang biasa dimanfaatkan untuk pengenalan gambar, VGG-16 menggunakan 16 layer dengan bobot dan dianggap sebagai salah satu arsitektur model yang cukup baik saat ini, VGG-16 terbukti memiliki akurasi lebih baik dengan *size* yang cukup kecil, Hal ini menunjukkan bahwa VGG-16 lebih unggul dibandingkan dengan algoritma VGG-19 dalam kasus ini (noorizky & kusumawati, 2023).



Gambar 2.13 VGG-16  
Sumber : Abhishek Jain, 2024

## 2.5 Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat open source dan multiparadigma, mendukung pemrograman berorientasi objek, prosedural, maupun fungsional. Python memiliki sintaks yang sederhana, mudah dipahami, serta dilengkapi dengan pustaka standar yang sangat luas, menjadikannya populer di kalangan peneliti, akademisi, maupun industri. Bahasa ini banyak digunakan untuk pengembangan aplikasi berbasis web, otomasi, ilmu data, kecerdasan buatan, hingga pengolahan citra. Salah satu keunggulan Python adalah ekosistem pustaka pihak ketiga yang sangat kaya, seperti *NumPy*, *TensorFlow*, dan *PyTorch*, yang mendukung penerapan pembelajaran mesin dan *deep learning* (Van Rossum & Drake, 2022).

### 2.5.1 Tensorflow

*TensorFlow* adalah *framework open-source* untuk *machine learning* dan *deep learning* yang mendukung pembangunan dan *deployment* model dalam berbagai skala, mulai dari komputer pribadi hingga *cluster* besar dengan GPU dan TPU. Salah satu keunggulan *TensorFlow* adalah

fleksibilitas arsitekturnya, di mana pengguna dapat memilih antara eksekusi graf statis ataupun *eager execution* untuk memudahkan optimasi (El Aoun dkk., 2022).

### 2.5.2 Keras

Keras adalah API *high-level* dalam *Python* yang berjalan di atas backend seperti *TensorFlow*, yang memudahkan pembangunan, pelatihan, dan *deployment model-deep learning* dengan sintaks yang sederhana dan *modular*. Karena sifatnya yang lebih abstrak dibanding menulis model dari nol, pengguna bisa lebih cepat menguji ide, membuat prototipe, dan mengimplementasikan berbagai jenis arsitektur (Lafta, 2023).

### 2.6 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (*Integrated Development Environment*, IDE) resmi dari Google untuk pembuatan aplikasi Android. IDE ini menyediakan berbagai fitur yang memudahkan proses pembuatan aplikasi, seperti editor kode dengan bantuan *auto-completion*, *debugger*, *layout editor visual*, emulator untuk uji coba aplikasi, dan manajemen *dependency* menggunakan *Gradle*. Dalam konteks pendidikan dan pengembangan perangkat lunak, *Android Studio* memungkinkan pengembang atau mahasiswa untuk merancang aplikasi dengan metode yang terstruktur (seperti UML atau waterfall), melakukan testing, dan mengintegrasikan komponen modern seperti pembayaran digital atau fitur real-time (Aminuddin & Nirsal, 2024). Selain itu Android Studio juga mendukung deployment ke perangkat Android nyata, sehingga

aplikasi bisa diuji langsung dalam lingkungan pengguna sesungguhnya (Sulistiani dkk., 2022).

2.7 Citra Digital

Menurut Zahid Mahmood didalam jurnalnya yang berjudul *Digital Image Processing: Advanced Technologies and Applications*, ia menggambarkan citra digital sebagai representasi data visual, dimana setiap pixel mengandung nilai spesifik yang mencerminkan karakteristik seperti warna, atau intensitas. Proses citra digital ini dapat dimanfaatkan dalam beberapa tugas diantaranya seperti mendeteksi objek pada gambar, pengenalan objek pada gambar, dan bahkan memanipulasi gambar. Bidang ini telah berkembang pesat, terutama dengan integrasi *deep learning*. (Zahid, 2024).

2.8 Penelitian Terdahulu

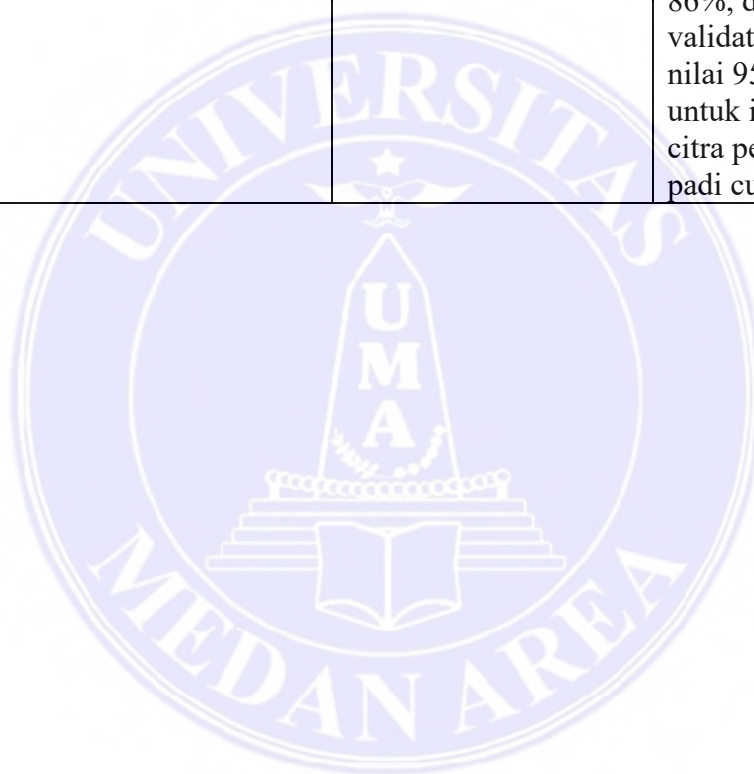
Penelitian terdahulu memaparkan tentang penelitian yang telah dilakukan oleh peneliti lain tentang kasus-kasus yang berkaitan dengan penelitian saat ini. Penelitian terdahulu menjadi bahan referensi dan kajian dalam melakukan penelitian ini. Berikut adalah penelitian terdahulu yang berkaitan dengan penelitian ini.

Tabel 2.1 Penelitian Terdahulu

| no | Nama Peneliti dan tahun | Judul Penelitian                                                                                            | Hasil Penelitian                                                                                                                                |
|----|-------------------------|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Devina Margarita,       | Klasifikasi penyakit daun padi menggunakan support vector machine berdasarkan fitur mendalam (Deep Feature) | pada pembagian data 80:10:10 dengan menggunakan parameter C =0.1, dan gamma = 0.031. Untuk kernel linear, juga memiliki hasil akurasi yang sama |

|   |                                                        |                                                                                |                                                                                                                                                                                                                                                                                                                                                                                             |
|---|--------------------------------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                                                        |                                                                                | dengan kernel RBF yaitu 93.94%,                                                                                                                                                                                                                                                                                                                                                             |
| 2 | Primatua Sitompul, Harly Okprana, Annas Prasetyo, 2022 | Identifikasi penyakit tanaman padi melalui Citra daun emnggunakan DenseNet 201 | Pada percobaan yang telah dilakukan untuk mengidentifikasi penyakit tanaman padi berdasarkan citra daun menggunakan model DenseNet201, diperoleh nilai akurasi sebesar 92.59% pada data training dan nilai akurasi sebesar 82.99% pada data testing. Evaluasi model dengan menggunakan confusion matriks diperoleh nilai precision sebesar 0.85%, recall 0.83% dan f1- score sebesar 0.83%. |
| 3 | Muhammad Yazid Abud Asseweth, 2024                     | Klasifikasi penyakit pada daun tanaman padi Menggunakan Arsitektur DensNet-169 | Arsitektur DenseNet-169 memperoleh hasil terbaik dengan menggunakan model hasil tuning grid search dengan menggunakan nilai hyperparameter epoch sebesar 25, batch size 32, optimizer Adam, dan learning rate sebesar 0.001. Hasil pengujian dari model ini mendapatkan nilai precision 93%, recall 93%, f1-score 93%, dan accuracy 93%.                                                    |

|   |                                                                                 |                                                     |                                                                                                                                                                                                                                                                                                                                                               |
|---|---------------------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Fakhri Habib Hawari, Faslah Fadillah, Muhamad Rifqi Alviandi, Toni Arifin, 2022 | Klasifikasi penyakit padi emnggunakan Algoritma CNN | Dari hasil penelitian ini dapat disimpulkan bahwa metode Deep Learning CNN dapat diimplementasikan untuk identifikasi citra daun padi yang berpenyakit. Nilai tertinggi dari akurasi data training mencapai nilai 85%, untuk data testing 86%, dan untuk data validation mencapai nilai 95%. Sehingga untuk identifikasi citra penyakit daun padi cukup baik. |
|---|---------------------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan sejak dikeluarkannya izin penelitian pada 11 April 2025, Tempat yang digunakan dalam melakukan penelitian ini adalah pada laboratorium Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.

#### 3.2 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan dalam penelitian ini adalah sebuah unit laptop dan sebuah perangkat android sebagai *hardware* (perangkat keras) yang didukung dengan beberapa *software* (perangkat lunak), adapun beberapa deskripsi spesifikasi kebutuhan perangkat keras yang digunakan sebagai berikut.

Tabel 3.1 Kebutuhan Perangkat Keras Komputer

| No. | Nama perangkat | Deskripsi                                    |
|-----|----------------|----------------------------------------------|
| 1.  | Laptop         | Lenovo Flex 5                                |
| 2.  | Processor      | 12th Gen Intel(R) Core(TM) i3-1215U 1.20 GHz |
| 3.  | RAM            | 8.00 GB                                      |
| 4.  | Harddisk       | 500 GB                                       |

Tabel 3.2 Kebutuhan Perangkat Keras *Smartphone*

| N0. | Nama perangkat | Deskripsi     |
|-----|----------------|---------------|
| 1.  | Smart phone    | Galaxy A35 5G |
| 2.  | processor      | Octa-core     |
| 3.  | RAM            | 8.00 GB       |
| 4.  | Harddisk       | 256 GB        |

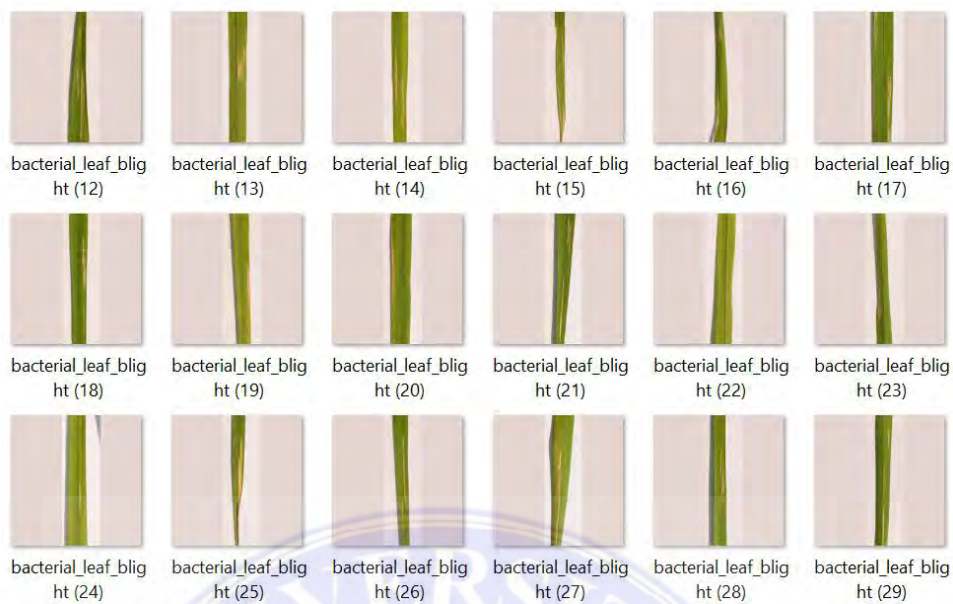
Untuk spesifikasi kebutuhan perangkat lunak (*Software*) yang digunakan antara lain:

Tabel 3.3 Kebutuhan Perangkat Lunak

| No. | Nama                            | Deskripsi                                                                  |
|-----|---------------------------------|----------------------------------------------------------------------------|
| 1.  | Windows 11 Home Single Language | Operating system                                                           |
| 2.  | Python 3.12.4                   | Bahasa pemrograman yang digunakan untuk melakukan coding dan melatih model |
| 3.  | Visual Studio Code              | Tools untuk menulis dan menjalankan coding dari model yang akan di latih   |
| 4.  | Microsoft Office 2013           | Tools yang di gunakan untuk menulis laporan penelitian                     |
| 5.  | Android Studio                  | Tools yang di gunakan untuk mendeploy model ke dalam aplikasi android      |

### 3.3 Dataset

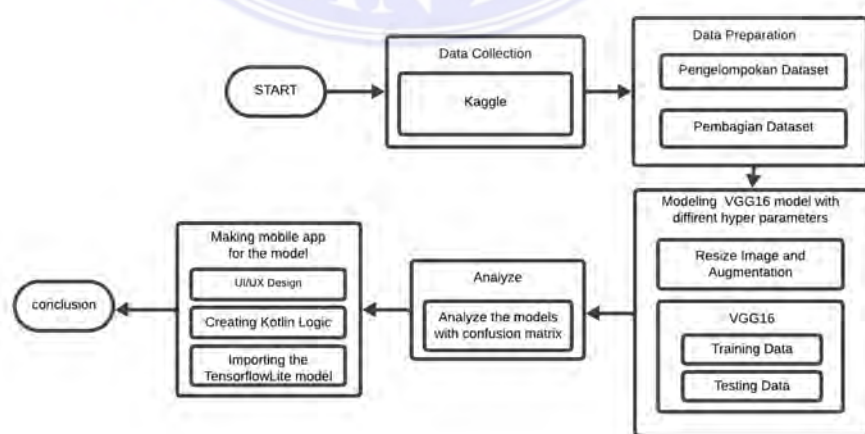
Pengumpulan data menggunakan dataset yang tersedia di *Kaggle*, yang bernama *Rice Leaf Disease* dengan jumlah 15,000 data latih dan 3,400 data tes yang terbagi menjadi 9 penyakit padi dan 1 folder yang berisi citra atau gambar daun padi sehat yaitu *Bacterial Leaf Blight*, *Brown Spot*, *Healthy*, *Leaf Blast*, *Leaf Scald*, *Narrow Brown Spot*, *Neck Blast*, *Rice Hispa*, *Sheath blight*, dan *Tungro*, dataset ini berupa macam macam gambar dari tanaman padi yang terjagkit hama atau penyakit.



**Gambar 3.1** Dataset Daun Padi

### 3.4 Tahapan Penelitian

Untuk mendukung penelitian ini, diperlukan suatu rangkaian kerangka kerja yang sistematis dalam tahapan-tahapannya. Rangka kerja ini akan menguraikan langkah-langkah yang akan dijalankan untuk mengatasi permasalahan yang telah diidentifikasi, garis besar dari kerangka kerja yang akan diterapkan dapat di lihat pada Gambar 3.2 di bawah.



**Gambar 3.2** Tahapan Penelitian

### 3.4.1 Data Collection

Pengumpulan data dilakukan untuk mendapatkan informasi yang dibutuhkan guna melakukan pelatihan model VGG16 yang akan digunakan identifikasi penyakit padi, data yang dibutuhkan iyalah data berupa gambar dari tanaman padi yang terjangkit penyakit atau hama, disini data diperoleh melalui *Kaggle*, hingga mendapatkan total data sebanyak kurang lebih 18,000 data berupa gambar tanaman padi yang terjangkit penyakit.

### 3.4.2 Data Preparation

Dalam tahap ini data yang telah dikumpulkan sebelumnya akan dipersiapkan lebih lanjut agar mempermudah malakukan pelatihan model persiapan data ini antara lain:

- a) Pengelompokan dataset, pada proses ini data yang telah didapat akan dikelompokkan sesuai penyakit atau kelasnya masing masing, guna mempermudah dalam proses pelatihan.
- b) Pembagian dataset, pada proses ini data yang telah didapat dan dikelompokkan akan dibagi menjadi data latih dan data validasi.

### 3.4.3 Model Training

Pada tahap ini peneliti melakukan training pada beberapa model dengan *Hyperparameter* yang berbeda untuk menemukan model dengan akurasi terbaik. Sebelumnya dari 10 kelas data, peneliti akan mengecualikan satu kelas data dari data latih dan data uji, yaitu *Neck Blast* dikarnakan isi dari kelas ini merupakan kumpulan citra atau foto dari buah padi atau *grain*,

dibandingkan kelas lain yang berisi data foto daun padi yang terkena penyakit.

Pengecualian kelas *Neck Blast* ini juga di terapkan pada data uji, sehingga dari 3400 data uji yang di gunakan adalah 3099 data, dan dikarenakan adanya perbedaan jumlah data pada tiap kelas, Penulis juga akan membatasi data yang digunakan pada setiap kelas, peneliti akan menggunakan 1000 data dari tiap kelas. Peneliti akan melatih masing masing model dan melakukan analisa untuk mengetahui model mana yang mendapatkan akurasi tertinggi, setelah model dengan akurasi terbaik didapat maka model tersebut akan digunakan dan dikonversi dalam bentuk *TensorflowLite* dan kembali dianalisa keakuratan model tersebut dalam format *TensorflowLite*, scenario *hyperparameter* yang akan digunakan dalam penelitian ini dapat di lihat pada Tabel 3.4 di bawah.

Tabel 3.4 Hyperparameter

| model | Epoch | Bach Size | Optimizer | Learning Rates |
|-------|-------|-----------|-----------|----------------|
| 1     | 20    | 16        | AdamW     | 0,00005        |
| 2     | 20    | 16        | Adam      | 0,00005        |
| 3     | 20    | 32        | AdamW     | 0,00005        |
| 4     | 20    | 32        | Adam      | 0,00005        |
| 5     | 30    | 16        | AdamW     | 0,00005        |
| 6     | 30    | 16        | Adam      | 0,00005        |
| 7     | 30    | 32        | AdamW     | 0,00005        |
| 8     | 30    | 32        | Adam      | 0,00005        |

*Hyperparameter* adalah parameter yang ditentukan sebelum model memulai pelatihan. *Hyperparameter* mempengaruhi bagaimana sebuah model *machine learning* mempelajari data. Penggunaan *hyperparameter*

yang tepat dapat meningkatkan kinerja model *machine learning* baik dari akurasi model ataupun waktu pembelajaran model.

Pemilihan nilai *learning rate* yang kecil, yaitu 0.00005, didasari oleh penelitian dari Leslie N. Smith (2017) dengan judul “*Cyclical Learning Rates for Training Neural Networks*”. Penelitian tersebut menjelaskan bahwa *learning rate* yang terlalu besar dapat menyebabkan model cepat konvergen namun berisiko mengalami *overfitting* atau tidak stabil. Sebaliknya, *learning rate* yang lebih kecil cenderung membuat proses pelatihan lebih stabil serta membantu model mencapai kemampuan generalisasi yang lebih baik.

Sementara itu, untuk *optimizer* digunakan Adam dan AdamW. Pemilihan ini didukung oleh penelitian (Zhou dkk., 2024) dalam “*Towards Understanding Convergence and Generalization of AdamW*”, yang menunjukkan bahwa AdamW dengan mekanisme *decoupled weight decay* memiliki kesalahan generalisasi yang lebih kecil. Selain itu, penelitian (Kumar dkk., 2022) dalam “*How to Fine-Tune Vision Models with SGD*” juga menemukan bahwa AdamW sering kali memberikan performa lebih baik pada proses *fine-tuning* model visi, terutama dalam kondisi adanya perubahan distribusi data (*distribution shift*). Dengan demikian, penggunaan Adam dan AdamW dalam penelitian ini dipilih untuk memberikan keseimbangan antara kecepatan konvergensi serta peningkatan kemampuan generalisasi model.

### 3.4.4 Pembagian data

Dalam proses klasifikasi penyakit tanaman padi ini data yang didapat ialah 2 jenis data yaitu data latih dan data uji, namun peneliti masih memerlukan data validasi untuk melatih model VGG-16 yang akan diteliti, oleh karna itu data latih akan dibagi menjadi dua yaitu data latih dan data validasi. Dikarnakan jumlah data pada setiap kelas tidak sama maka peneliti perlu membatasi data yang akan di gunakan untuk melatih model agar menghindari bias pada proses pelatihan.

Peneliti menentukan jumlah data yang akan digunakan, dalam penelitian ini data yang digunakan ialah sebanyak 9000 data yang mencakup 9 kelas, masing masing kelas akan menggunakan 1000 data latih yang kemudian akan dibagi sekitar 0.2-nya menjadi data Validasi, adapun deskripsi dari pembagian data dapat dilihat pada Table 3.5 dibawah:

Tabel 3.5 Pembagian dataset

| No.               | Nama Kelas            | Data Latih  | Data Validasi | Data Tes    |
|-------------------|-----------------------|-------------|---------------|-------------|
| 1                 | Bacterial leaf blight | 800         | 200           | 375         |
| 2                 | Brown Spot            | 800         | 200           | 380         |
| 3                 | Healthy               | 800         | 200           | 391         |
| 4                 | Leaf blast            | 800         | 200           | 362         |
| 5                 | Leaf scald            | 800         | 200           | 386         |
| 6                 | Narrow Brown Spot     | 800         | 200           | 382         |
| 7                 | Rice Hispa            | 800         | 200           | 225         |
| 8                 | Sheath Blight         | 800         | 200           | 288         |
| 9                 | Tungro                | 800         | 200           | 310         |
| <b>TOTAL</b>      |                       | <b>7200</b> | <b>1800</b>   | <b>3099</b> |
| <b>PERSENTASE</b> |                       | <b>80%</b>  | <b>20%</b>    | <b>-</b>    |

### 3.4.5 Augmentasi Dataset

Augmentasi data adalah proses untuk memperluas dataset dengan cara memanipulasi data citra sedemikian rupa untuk membuat variasi lebih dari citra yang ada, augmentasi dilakukan dengan cara mengubah sudut, memutar, memperbesar dan memperkecil gambar untuk mendapatkan lebih banyak variasi dari satu citra gambar, proses ini akan memperkaya dan menambah keberagaman gambar yang kita miliki, semua model dalam penelitian ini memiliki konfigurasi Augmentasi data yang sama yaitu:

Tabel 3.6 konfigurasi Augmentasi

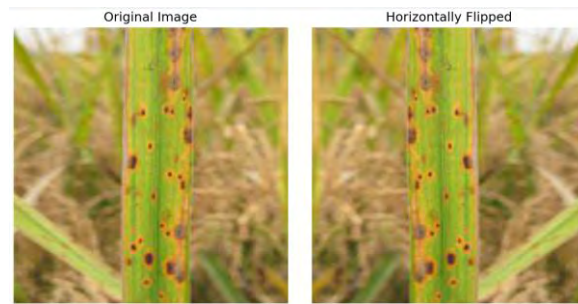
|                  |         |
|------------------|---------|
| Rescale          | 1.0/255 |
| Horizontal Flip  | True    |
| Rotation Range   | 20      |
| Shear Range      | 0.2     |
| Zoom Range       | 0.2     |
| Validation Split | 0.2     |

#### 1. *Rescale*

*Rescaling* merupakan proses untuk merubah ukuran gambar yang awalnya tidak beraturan atau random, menjadi ukuran yang di perlukan, dalam kasus ini ukuran gambar yang di perlukan ialah 255 pixel, yang merupakan input untuk model VGG16 itu sendiri.

#### 2. *Horizontal Flip*

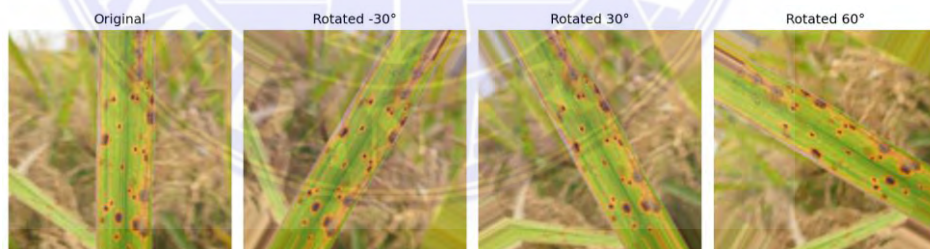
*Horizontal Flip* digunakan untuk memutarbalikan gambar secara horizontal guna menambah variasi dari data gambar training yang digunakan. Contoh Horizntal Flip dapat dilihat pada Gambar 3.3 di bawah.



Gambar 3.3 Augmentasi Horizontal Flip

### 3. *Rotation Range*

*Rotation Range* digunakan untuk merotasi gambar, dalam kasus ini gambar akan dirotasi secara *random* sebesar 20 derajat pada setiap batch, hal ini bertujuan untuk menambah variasi dari rotasi atau sudut dalam pelatihan model, agar model nantinya mampu untuk mengenali gambar dengan berbagai macam sudut rotasi, seperti yang terlihat pada Gambar 3.4 di bawah.



Gambar 3.4 Augmentasi Rotation Range

### 4. *Shear Range*

*Shear Range* digunakan untuk memberikan variasi distorsi gambar pada model, konfigurasi *Shear Range* akan memiringkan gambar 20% ke segala arah, atas, kanan, kiri, atau bawah, konfigurasi ini akan membantu

model mengenali objek walau memiliki perspektif yang berbeda, seperti pada Gambar 3.5 di bawah.



Gambar 3.5 Augmentasi Shear Range

#### 5. *Zoom Range*

*Zoom Range* digunakan untuk melakukan zooming secara acak pada data gambar, konfigurasi ini digunakan untuk memberikan variasi ukuran objek pada gambar dengan cara melakukan *zooming* pada data gambar, seperti yang terlihat pada Gambar 3.6 di bawah.



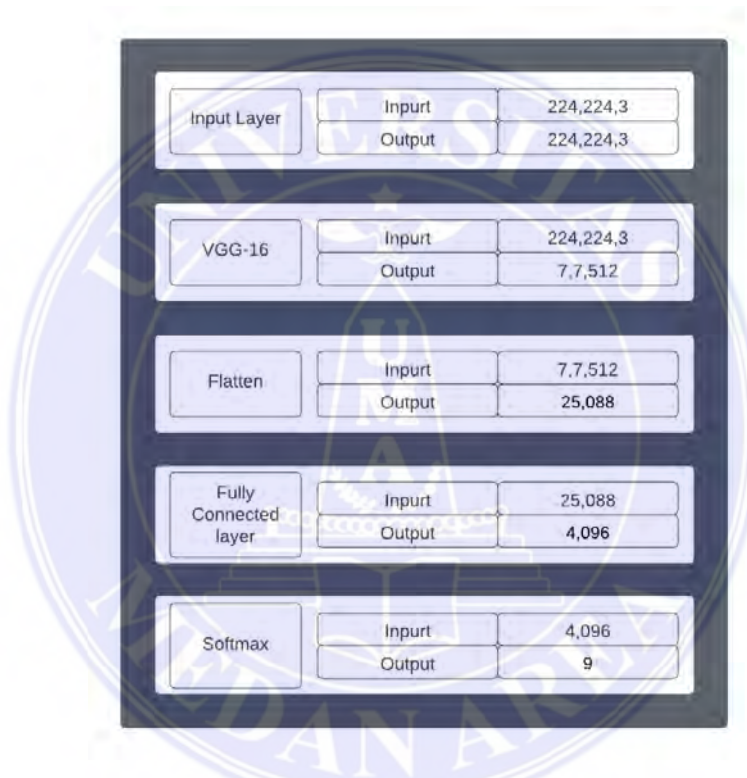
Gambar 3.6 Augmentasi Zoom Range

#### 6. *Falidation Split*

*Falidation Split* ini adalah proses pembagian sebagian data latih untuk tujuan validasi model saat proses pelatihan, Dalam kasus ini peneliti membagi data training sebesar 80% dan data validasi sebesar 20%, hal ini berfungsi sebagai pengukur peforma model saat training,

### 3.4.6 Pemodelan Arsitektur VGG-16

Penelitian ini akan menggunakan Arsitektur VGG-16 dikarenakan model ini memiliki lapisan convulsi yang lebih kecil di banding VGG-19 menjadikan VGG-16 akan memiliki ukuran model yang sedikit lebih kecil untuk digunakan pada aplikasi berbasis Android, berikut merupakan pemodelan dari arsitektur VGG-16.



Gambar 3.7 Rancangan Arsitektur VGG-16

1. Pada bagian Input Layer, citra yang akan digunakan berukuran 224x224x3. Angka 3 pada ukuran citra tersebut menandakan channel image (RGB) yang digunakan merupakan citra warna (*true color*). Citra yang menjadi masukan (*input*) dalam pelatihan model merupakan citra yang telah diaugmentasi pada proses sebelumnya.

2. Citra yang sudah diinputkan kemudian masuk pada tahap *feature extractor* untuk mengekstraksi fitur dari citra dataset menggunakan *pre-trained* model dengan arsitektur VGG-16. terdapat beberapa parameter dari arsitektur VGG-16 yang disesuaikan dalam penelitian ini, yaitu:

- a) `include_top=False`

Befungsi untuk menghilangkan layer terakhir (*layer output*) pada model VGG-16. Layer output default dari MobileNetV2 adalah 1000 (sesuai jumlah class VGG-16). Pada penelitian ini, jumlah *class* dari output-nya adalah 9. Oleh karena itu, menggunakan parameter `include_top=False` untuk menghilangkan layer terakhir pada model VGG-16.

- b) `weights='imagenet'`

Berfungsi agar dapat menggunakan *weights* model dari VGG-16 yang sudah dilatih pada dataset imagenet.

- c) `trainable=False`

Berfungsi agar semua layer pada VGG-16 tidak dapat di-training ulang. Hal ini karena hanya ingin menggunakan model VGG-16 sebagai *feature extractor* untuk mengekstraksi fitur dari citra dataset, dan tidak ingin mengubah bobot-bobot (*weights*) yang sudah dilatih pada model VGG-16.

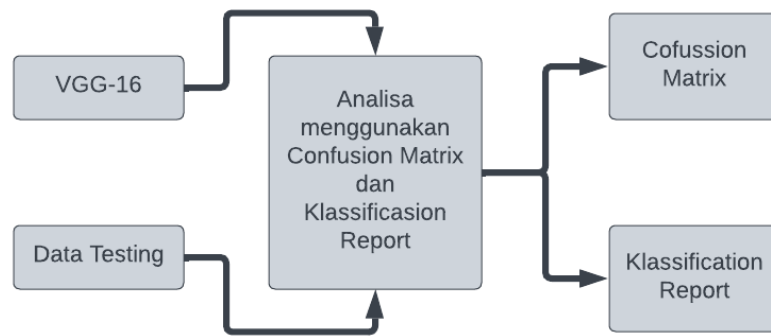
3. Output dari VGG-16 kemudian dilakukan proses proses *Flatten*. *Flatten layer* merupakan langkah yang dilakukan sebelum masuk ke *fully connected layer*. *Feature map* dari tahap *feature extractor* masih

berbentuk *array* multidimensi, sehingga perlu dilakukan proses *Flatten* untuk mengubah ulang fitur (*reshapre feature map*) menjadi vektor satu dimensi agar dapat digunakan sebagai *input* untuk tahap klasifikasi atau *fully connected*.

4. Selanjutnya, model akan dihubungkan dengan dense layer yang berfungsi untuk melakukan klasifikasi ke dalam 9 kelas yang terdapat dalam dataset. Dense layer ini akan menambahkan lapisan *fully connected* yang berperan sebagai *classifier*. *Fully connected layer* berfungsi sebagai hidden layer yang menentukan ke kelas mana citra akan dikelompokkan berdasarkan fitur yang telah diekstraksi oleh lapisan sebelumnya.

#### 3.4.7 Analisa Model Dengan Peforma Terbaik

Mengevaluasi kinerja model menggunakan data uji yang independen. Mengukur metrik evaluasi seperti akurasi, precision, recall, dan f1-score dengan menggunakan *confusion matrix*, dan *Classification Report* untuk mengevaluasi seberapa baik model dalam mengklasifikasi jenis penyakit. Model yang dihasilkan akan diuji menggunakan data testing untuk mengevaluasi hasilnya lalu memilih model terbaik dengan akurasi yang paling tinggi, dan tahap yang terakhir adalah melakukan *export* model kedalam aplikasi siap pakai.



Gambar 3.8 Tahapan Analisa

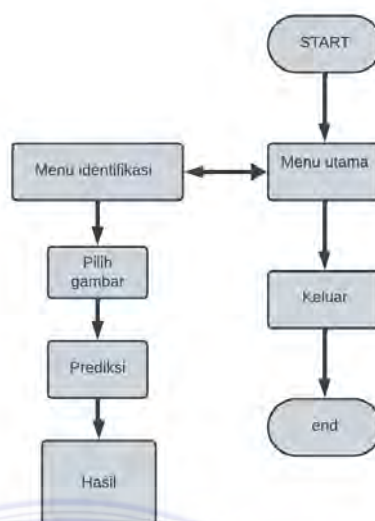
### 3.4.8 Deploy Mobile App for the Model

Tahapan ini adalah proses pembuatan aplikasi mobile yang akan digunakan sebagai Tes *platform* untuk model VGG-16 yang telah dilatih dan dipilih, berikut merupakan tahapan yang dilakukan:

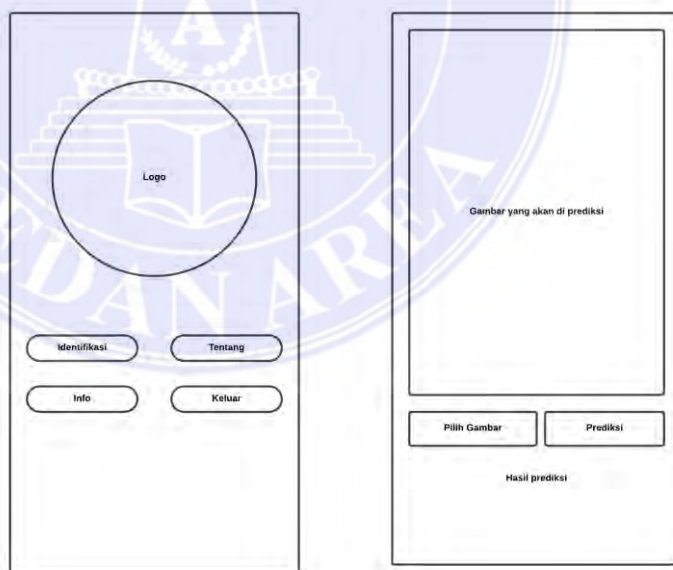
#### a) UI/UX Design

Tahap ini adalah proses pembuatan *interface* yang akan digunakan sebagai tempat interaksi bagi user, dan tempat dimana user dapat melakukan input untuk berintraksi dengan model yang digunakan.

Flowchard dan perancangan UI/UX dari aplikasi dapat dilihat pada Gambar di bawah. Flowchard dari software klasifikasi penyakit daun padi dapat dilihat pada Gambar 3.9 di bawah.



Gambar 3.9 Flowchart Aplikasi perancangan UI/UX dari aplikasi sederhana yang akan di bangun dapat dilihat pada Gambar 3.10 di bawah.



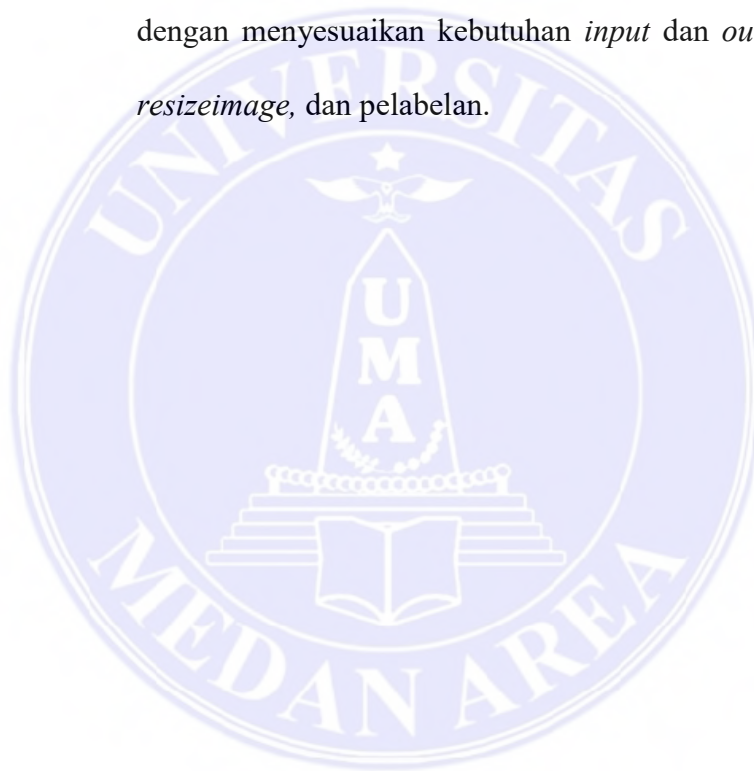
Gambar 3.10 perancangan UI/UX

b) Creating Kotlin Logic

Tahap ini adalah tahap pembuatan kotlin file dimana input dan interaksi dari *user* akan diproses untuk menghasilkan sebuah output yang diinginkan.

c) Importing the TFlite Model

Dalam tahap ini model yang sebelumnya dilatih akan diimport dan integrasikan kedalam *system* yang sebelumnya telah dibuat, dengan menyesuaikan kebutuhan *input* dan *outputnya*, seperti *resizeimage*, dan pelabelan.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil analisis dalam melakukan klasifikasi jenis penyakit padi dengan menggunakan arsitektur VGG-16 adalah sebagai berikut:

1. Dari 8 model yang dilatih terdapat 4 model dengan performa yang tinggi yaitu model 5, 6, 7, dan 8, model 5 menggunakan parameter 0.00005 *Learning rates*, 16 *batch size*, *AdamW Optimizer*, 0.5 *Dropout rate*, dan 30 *Epoch*, dengan parameter ini menghasilkan akurasi validasi training sebesar 91%, model 6 menggunakan parameter : 0.00005 *Learning rates*, 16 *batch size*, *Adam Optimizer*, 0.5 *Dropout rate*, dan 30 *Epoch* dengan parameter ini menghasilkan akurasi validasi training sebesar 90%, model 7 dengan parameter : 0.00005 *Learning rates*, 32 *batch size*, *AdamW Optimizer*, 0.5 *Dropout rate*, dan 30 *Epoch*, dengan parameter ini menghasilkan akurasi validasi training sebesar 90%, dan model 8 dengan parameter 0.00005 *Learning rates*, 32 *batch size*, *Adam Optimizer*, 0.5 *Dropout rate*, dan 30 *Epoch* dengan parameter ini menghasilkan akurasi validasi training sebesar 90%, keempat model ini kemudain diuji dan dievaluasi menggunakan *confusion matrix* dan *classification report*. Model juga menunjukkan peningkatan akurasi bergantung dari jumlah *Epoch* yang digunakan, model dengan *Epoch* 20 menunjukkan akurasi yang lebih rendah di banding *Epoch* 30 yang mencapai 90%.

2. Dari 4 skenario model yang diuji menggunakan data testing dan dievaluasi menggunakan *confusion matrix* dan *classification report*, maka performa model yang paling optimal adalah model 6 dengan Persentasi akurasi sebesar 94.29%.
3. Setelah model 6 dikonversi ke format TFLite model mengalami penurunan akurasi sebesar 1.06%, dari 94% menjadi 93%.
4. Setelah melakukan penelitian ini, peneliti dapat menyimpulkan bahwa model VGG-16 dengan format TFLite dapat digunakan dalam perangkat mobile dengan akurasi model TFLite sebesar 93% dan dapat dijalankan di perangkat mobile dengan baik.

## 5.2 Saran

Saran yang dapat penulis berikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menggunakan dataset yang diperluas dengan jumlah yang lebih banyak, dan meningkatkan nilai Epoch yang lebih tinggi sehingga dapat menghasilkan performa model yang lebih baik, atau bisa dengan menggunakan data *primer* untuk pengujiannya.
2. Penelitian selanjutnya dapat menerapkan *quantization* pada saat melakukan Konversi dari format .h5 keras menjadi TFLite untuk memperkecil file model
3. Penelitian selanjutnya dapat menambah variasi model untuk mendapatkan model dengan performance yang lebih baik.

## DAFTAR PUSTAKA

- Cipta, G. S., Putra, S. U., Darma, A. E., & Ambara, D. G. (2023). Smart Mobile Application for Detecting Balinese Masks to Introduce Balinese Culture to World Tourism. *Jurnal Info Sains : Informatikan dan Sains*, 356.
- Hawari, F. H., Fadillah, F., Alviandi, M. R., & Arifin, T. (2022). Klasifikasi penyakit padi menggunakan algoritma CNN. *JURNAL RESPONSIF*, 188.
- Li, D., Li, S., Wei, S., & Sun, W. (2021). Strategies to Manage Rice Sheath Blight: Lessons from Interactions between Rice and *Rhizoctonia solani*.
- M, K. K., Kumar, R., & Kumar, A. (2024). Impact of IPM Pratices on Microbial population and disease development in transplanted and direct-seeded rice. *Frontier in microbiology*, 02.
- Margarita, D. (2024). Klasifikasi penyakit daun padi menggunakan Support Vector Machine. *repository.upnjatim*.
- Nivolli, C., Baccarin, A. C., & Pedrozo, R. (2024). Leaf and Neck Blast of Rice. *Agriculture and Natural Resources*.
- noorizky, a. z., & kusumawati, w. i. (2023). Perbandingan Performa Algoritma VGG16 dan. *Journal of Computer, Electronic, and Telecommunication*.
- Radana, J. D. (2022). Implementasi Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN) untuk Image Classification Varietas Tanaman Tembakau. 8.
- Syuhada, R. (2023). Analisis arsitektur deep learning VGG untuk Klasifikasi jenis Jamur. *repository.uma.ac.id*, 12.
- Widiarta, N., Firmansyah, F., Yunush, M., & Apriana, A. (2025). Rice virus disease in Indonesia: epidemiology and varietal resistance. *Phytopathology Research*, 3.
- Zahid, M. (2024). Digital Image Processing: Advanced Technologies and Applications. *Department of Electrical and Computer Engineering*, 1-6.
- Jin, M. L., & Ding, D. Y. (2025). Research insights into photosynthetic gene regulation in rice for yield improvement. *Rice Genomics and Genetics*, 36-49.

- Gong, X., Chen, J., Chen, Y., He, Y., & Jiang, D. (2024). *Advancements in rice leaf development research*. *Plants*, 1–15.
- GoodStats. (2023). *Jumlah pengguna smartphone di Indonesia*. Diakses dari <https://www.goodstats.com/id/smartphone>
- Statista. (2023). *Penetrasi kepemilikan ponsel di Indonesia*. Diakses dari <https://www.statista.com/statistics/1084089/indonesia-share-of-rural-population-owning-a-mobile-phone/>
- Subburaj, S., Thulasinathan, T., Sakthivel, V., Ayyenar, B., Kambale, R., Rajagopalan, V. R., Manickam, S., Rajasekaran, R., Chellappan, G., Thiagarajan, K., Swaminathan, M., & Muthurajan, R. (2024). *Genetic enhancement of blast and bacterial leaf blight resistance in rice variety CO 51 through marker-assisted selection*. *Agriculture*, 14(5), 693.
- Brown Spot of Rice: Worldwide Disease Impact, Phenotypic and Genotypic Diversity, and Management Strategies*. (2023). *Plant Pathology*, 72(6), 1605–1620.
- Murudkar, A. C., Kadam, J. J., Pawar, H. D., Shinde, P. U., Chavan, S. A., & Dhemre, V. N. (2023). *Exploration of botanicals against Microdochium oryzae, causal agent of leaf scald disease of rice*. *Journal of Plant Disease Sciences*, 18(1), 16–21.
- Priyadarshani, T. D. C., Herath, U. S., Wickramasinghe, M. D. M., Egodawatta, W. C. P., Beneragama, D. I. D. S., & Sirisena, U. G. A. I. (2022). *Factors affecting incidence of narrow brown leaf spot of rice caused by Passalora janseana*. *Ceylon Journal of Science*, 51(1), 83–87.
- Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Zatu Kusuma Frisky, A., & Qazi Mazhar ul Haq. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748.
- Shen, W., Wei, Z., Huang, S., Zhang, B., Fan, J., Zhao, P., & Zhang, Q. (2021). Interpretable compositional convolutional neural networks. *arXiv preprint*.
- Zafar, A., Aamir, M., Mohd Naw, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17), 8643.

PAPER NAME AUTHOR

**ARI RAMA GUNANDA\_KLASIFIKASI PENYAKIT PADA TUMBUHAN PADI \_\_MELALUI CITRA DAUN DENGAN PEMANFAATAN\_MODEL VGG16\_HYMcC49o3QJHMazIsKLsHiGWrmI24yQpqzPHswIV.pdf**

WORD COUNT 10402 Words

CHARACTER COUNT 55079 Characters

PAGE COUNT 69 Pages

FILE SIZE  
2.0 MB

SUBMISSION DATE

Oct 17, 2025 2:28 PM GMT+7

REPORT DATE

Oct 17, 2025 2:30 PM GMT+7

### 15 % Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 15% Internet database 4 % Publications database
- Crossref database Crossref Posted Content database

### Excluded from Similarity Report

- Submitted Works database Bibliographic material
- Cited material Abstract
- Small Matches (Less than 15 words)

## 1. Surat Keputusan Pembimbing Tugas Akhir



### UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate / Jalan Gedung PBSI, Medan 20223  
Kampus II : Jalan Sei Serayu Nomor 70 A / Jalan Setia Budi Nomor 79 B, Medan 20112 Telepon : (061) 8225602, 8201994  
Fax : (061) 8226334 HP : 0811 607 259 website: [www.uma.ac.id](http://www.uma.ac.id) Email : [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

Nomor : 806/FT/01.10/IV/2025  
Lampiran : -  
Hal : Pembimbing Tugas Akhir

11 April 2025

Yth. Pembimbing Tugas Akhir  
**SUSILAWATI S.Kom, M.Kom ( Sebagai Pembimbing )**  
di Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : ARI RAMA GUNANDA  
NIM : 198160051  
Jurusan : TEKNIK INFORMATIKA

Maka dengan hormat kami mengharapkan kesediaan saudara :

**SUSILAWATI S.Kom, M.Kom ( Sebagai Pembimbing )**

Adapun Tugas Akhir Skripsi berjudul :

**"Klasifikasi Penyakit pada Tumbuhan Padi Melalui Citra Daun dengan Pemanfaatan Model VGG16"**

Perpanjangan SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,



Dr Eng. Supriatno.ST, MT.

## 2. Surat Izin Melakukan Penelitian dan Pengambilan Data Tugas Akhir

**UNIVERSITAS MEDAN AREA**  
**FAKULTAS TEKNIK**

Kampus I : Jalan Kolam Rencong 1 Medan 20132  
Kampus II : Jalan Sebelah Nomor 70 / Jalan Sei Selayat Nomor 70 A 20122  
Website [www.teknik.uma.ac.id](http://www.teknik.uma.ac.id) E-mail [umt@uma.ac.id](mailto:umt@uma.ac.id)

Nomor : 83/FT.6/01.10/IV/2025  
Lamp : -  
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

11 April 2025

Yth. Wakil Rektor Bidang Mutu Sumber Daya Manusia dan Perekonomian  
Jln. Kolam No,1  
Di  
Medan

Dengan hormat, kami mohon kesediaan bapak kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

| NO | N A M A          | N P M     | PRODI              |
|----|------------------|-----------|--------------------|
| 1. | Ari Rama Gunanda | 198160051 | Teknik Informatika |

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area**.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan Ilmiah dan Skripsi, yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

**Klasifikasi Penyakit pada Tumbuhan Padi Melalui Citra Daun dengan Pemanfaatan Model VGG16.**

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya Surat ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

  
Dr. Eng. Supriatno, ST, MT

Tembusan :  
1. Ka. BIMPP  
2. Mahasiswa  
3. File

### 3. Surat Selesai Melakukan Penelitian



## SOURCE CODE

### Proses pelatihan model:

```
import os
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Flatten, Dropout
from tensorflow.keras.optimizers import Adam

# DIREKTORI DATA SET
train_dir = r"C:\Users\lenovo\latihan\myenv\Rice_Leaf_Disease\train"
test_dir = r"C:\Users\lenovo\latihan\myenv\Rice_Leaf_Disease\test"
save_model_path=r"C:\Users\lenovo\latihan\finish
model\VGG16_FINAL_MODEL_8.h5"

# Target classes
classes = [
    "bacterial_leaf_blight", "brown_spot", "healthy", "leaf_blast",
    "leaf_scald", "narrow_brown_spot", "rice_hispa", "sheath_blight", "tungro"
]

# Preprocess data
train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    horizontal_flip=True,
    rotation_range=20,
    shear_range=0.2,
    zoom_range=0.2,
    validation_split=0.2
)

test_datagen = ImageDataGenerator(rescale=1.0/255)

# Load data latih dan validasi
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    classes=classes,
    subset='training'
```

```

)

val_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    classes=classes,
    subset='validation'
)

# Load data tes
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    classes=classes
)

# Load VGG16 model (without the top layers)
base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Freeze base model layers
for layer in base_model.layers:
    layer.trainable = False

# Add custom layers
x = Flatten()(base_model.output)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(len(classes), activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=output)

# Compile the model
learning_rate = 0.00005
model.compile(optimizer=Adam(learning_rate=learning_rate),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
epochs = 30
history = model.fit(
    train_generator,
    epochs=epochs,

```

```

        validation_data=val_generator
    )

# Evaluate the model
test_loss, test_accuracy = model.evaluate(test_generator)
print(f'Test Accuracy: {test_accuracy:.4f}')
print(f'Test Loss: {test_loss:.4f}')

# Save the model
model.save(save_model_path)
print(f'Model saved at {save_model_path}')

```

### **Source Code Confusion Matrix:**

```

import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import os

# === Paths ===
model_path=r"C:\Users\lenovo\latihan\finish
model\VGG16_FINAL_MODEL_8.h5"
test_data_dir = r"C:\Users\lenovo\latihan\myenv\Rice_Leaf_Disease\test"
output_matrix_path=r"C:\Users\lenovo\Desktop\everything
thesis\SKRIPSI\Berkas Semhas\source\confussion matrix\CMR-8.png"

# === Load Model ===
model = load_model(model_path)
print("✓ Model loaded successfully.")

# === Prepare Test Data ===
test_datagen = ImageDataGenerator(rescale=1.0 / 255)
test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(224, 224),
    batch_size=16,
    class_mode='categorical',
    shuffle=False
)

# === Predict ===

```

```

print("🔍 Making predictions...")
predictions = model.predict(test_generator, verbose=1)
y_pred = predictions.argmax(axis=1)
y_true = test_generator.classes
class_labels = list(test_generator.class_indices.keys())

# === Classification Report ===
report = classification_report(y_true, y_pred, target_names=class_labels,
                              digits=4)
print("\n📊 Classification Report:\n")
print(report)

# === Confusion Matrix ===
cm = confusion_matrix(y_true, y_pred)
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=class_labels, yticklabels=class_labels)
plt.title("Confusion Matrix - VGG16_FINAL_MODEL_8")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.tight_layout()

# === Ensure output folder exists ===
os.makedirs(os.path.dirname(output_matrix_path), exist_ok=True)

# === Save Confusion Matrix ===
plt.savefig(output_matrix_path)
print(f"\n✅ Confusion matrix saved to: {output_matrix_path}")
plt.show()

```

### Source Code Classification Report:

```

import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import classification_report
import os

# Paths to model and test data
model_path=r"C:\\Users\\lenovo\\latihan\\finish
model\\VGG16_FINAL_MODEL_8.h5"
test_data_dir = r"C:\\Users\\lenovo\\latihan\\myenv\\Rice_Leaf_Disease\\test"

# Load the model
model = load_model(model_path)
print("Model loaded successfully.")

```

```
# Prepare the test data generator
test_datagen = ImageDataGenerator(rescale=1.0 / 255)
test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(224, 224), # Ensure this matches your model's input size
    batch_size=16,          # Batch size for testing
    class_mode='categorical',
    shuffle=False           # No shuffling for evaluation
)

# Get the class labels
class_labels = list(test_generator.class_indices.keys())

# Predict on the test data
print("Making predictions on the test data...")
predictions = model.predict(test_generator, verbose=1)
y_pred = predictions.argmax(axis=1)
y_true = test_generator.classes

# Generate the classification report
report = classification_report(
    y_true, y_pred, target_names=class_labels, digits=4
)

print("Classification Report:")
print(report)

# Save the report to a file
output_path=os.path.join(os.path.dirname(model_path),
    "classification_report.txt")
with open(output_path, "w") as f:
    f.write(report)

print(f'Classification report saved to: {output_path}')
```

### **Source Code Converting H5 to TFLite:**

```
import tensorflow as tf

# Load your old .h5 model
model=tf.keras.models.load_model(r"C:\Users\lenovo\latihan\finish
model\VGG16_FINAL_MODEL_6.h5")

# Export the model in SavedModel format (suitable for TFLite)
model.export("converted_model_tf") # This is the correct API for newer TF
versions
```

```
# Load the SavedModel directory
converter = tf.lite.TFLiteConverter.from_saved_model("converted_model_tf")
tflite_model = converter.convert()

# Save the TFLite model
tflite_path=r"C:\Users\lenovo\latihan\finish
model\rice_leaf_disease_model.tflite"
with open(tflite_path, "wb") as f:
    f.write(tflite_model)

print("🎉 Successfully converted to TFLite without quantization!")
```

