

Sistem Deteksi Penyakit Tanaman Padi Dan Rekomendasi Pupuk Menggunakan Metode Transfer Learning Dengan Model VGG19

Alvin Noor Hidayah¹, Asmunin²

¹alvin.21016@mhs.unesa.ac.id

²asmunin@unesa.ac.id

Abstrak— Penyakit tanaman padi adalah salah satu masalah terbesar dalam produktivitas pertanian global, terutama di negara-negara agraris seperti Indonesia. Untuk mengatasi masalah ini, penelitian ini menggunakan metode transfer learning dengan model VGG-19, salah satu arsitektur Convolutional Neural Network (CNN) yang paling diandalkan untuk klasifikasi gambar. Dengan menggunakan metode transfer learning digunakan untuk melatih dataset gambar daun padi yang digunakan. Ini meningkatkan akurasi dan efektivitas proses pelatihan. Metode ini mempercepat dan meningkatkan kinerja model dengan memanfaatkan bobot pra-latih dari dataset besar seperti ImageNet. Hasil penelitian menunjukkan bahwa sistem dengan model VGG-19 memiliki akurasi akhir sebesar 93% dan akurasi validasi sebesar 92%. Metode black-box juga digunakan untuk menguji sistem dari sudut pandang pengguna akhir dan melibatkan petani. Melalui deteksi cepat dan rekomendasi pengelolaan yang akurat. Penelitian ini tidak hanya membantu dalam penerapan kecerdasan buatan dalam pertanian, tetapi juga menawarkan solusi praktis untuk mengelola penyakit tanaman padi dengan lebih baik..

Kata kunci— Sistem Deteksi Penyakit Tanaman Padi, VGG19, Transfer Learning, CNN, Blackbox.

Abstrak— Rice plant diseases are one of the biggest problems in global agricultural productivity, especially in agrarian countries like Indonesia. To solve this problem, this study uses transfer learning method with VGG-19 model, one of the most reliable Convolutional Neural Network (CNN) architectures for image classification. By using the transfer learning method is used to train the rice leaf image dataset used. It improves the accuracy and effectiveness of the training process. This method accelerates and improves model performance by utilizing pre-trained weights from large datasets such as ImageNet. The results show that the system with the VGG-19 model has a final accuracy of 93% and a validation accuracy of 92%. The black-box method was also used to test the system from the end user's point of view to evaluate. Through rapid detection and accurate management recommendations. This research not only helps in the application of artificial intelligence in agriculture, but also offers practical solutions to better manage rice plant diseases..

Kata kunci— Rice Plant Disease Detection System, VGG19, Transfer Learning, CNN, Blackbox.

I. PENDAHULUAN

Padi merupakan tanaman sereal yang memiliki peran vital dalam ketahanan pangan dunia, menempati peringkat kedua

setelah jagung. Pada tahun panen terakhir, produksi beras global tercatat mencapai lebih dari 513 juta metrik ton, sebagian besar berasal dari negara-negara Asia [1]. Tiongkok, India, dan Bangladesh menjadi tiga negara dengan produksi terbesar, di mana Tiongkok memimpin dengan lebih dari 208 juta ton pada 2022 [2]. Produksi beras yang melimpah ini menunjukkan betapa strategisnya padi sebagai komoditas pangan. Indonesia, sebagai negara agraris juga memiliki peran penting dalam produksi padi di tingkat global karena sebagian besar penduduk Indonesia menjadikan beras sebagai makanan pokok sehari-hari [3]. Oleh karena itu, pengelolaan produksi padi menjadi salah satu prioritas pemerintah untuk memastikan kebutuhan pangan masyarakat terpenuhi.

Produksi padi Indonesia pada 2023 mengalami penurunan sebesar 1.12 juta ton GKG dibandingkan sebelumnya. Penurunan ini disebabkan oleh berbagai faktor, termasuk serangan penyakit pada tanaman dan distribusi pupuk bersubsidi yang tidak merata [4]. Masalah distribusi pupuk juga semakin diperburuk oleh transisi ke sistem digital yang belum berjalan optimal. Akibatnya, banyak petani terpaksa membeli pupuk nonsubsidi dengan harga tinggi, yang berdampak pada biaya produksi. Kondisi ini menjadi tantangan besar dalam upaya menjaga stabilitas produksi padi di Indonesia.

Mengatasi permasalahan tersebut, memerlukan inovasi teknologi dalam bidang pertanian. Salah satu teknologi yang dapat diandalkan adalah kecerdasan buatan Artificial Intelligence (AI) yang mampu meningkatkan efisiensi produksi [5]. Salah satu cabang AI yang relevan dalam bidang ini adalah computer vision, yang fokus pada pengolahan data visual seperti foto dan video. Teknologi ini memberikan peluang besar untuk membantu petani dalam mendiagnosis dan mengelola tanaman mereka dengan lebih baik.

Computer vision bekerja berdasarkan metode deep learning, cabang dari machine learning yang menggunakan jaringan saraf tiruan atau artificial neural networks (ANN). Salah satu metode populer dalam deep learning adalah Convolutional Neural Network (CNN), yang dirancang untuk menganalisis data visual. CNN memiliki kemampuan untuk mengenali pola pada gambar dan mengelompokkannya ke dalam kategori tertentu. Dalam konteks pertanian, CNN dapat digunakan untuk mendeteksi apakah tanaman padi terinfeksi penyakit atau tidak. Hal ini memungkinkan identifikasi dini yang dapat membantu mencegah kerugian lebih lanjut.

Penerapan CNN menghadapi beberapa tantangan meliputi kebutuhan data dalam jumlah besar dan daya komputasi yang tinggi. Untuk mengatasi tantangan ini, teknik transfer learning menjadi solusi yang efektif. Transfer learning menggunakan model yang telah dilatih sebelumnya untuk menyelesaikan tugas baru dengan adaptasi minimal. Salah satu model yang sering digunakan adalah VGG-19, jaringan saraf convolutional yang memiliki 19 lapisan. Model ini dikenal dengan kemampuannya dalam mendeteksi fitur visual secara akurat dan mudah diimplementasikan.

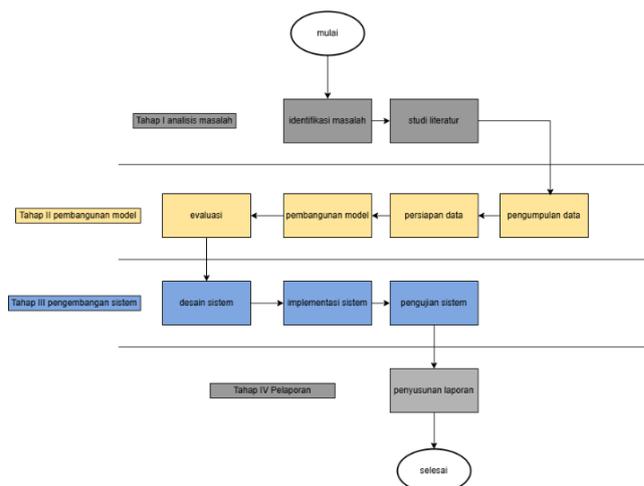
Efektivitas VGG-19 telah terbukti dalam berbagai penelitian sebelumnya di berbagai bidang. Misalnya, [6] berhasil mendeteksi penyakit mata katarak dengan akurasi mencapai 91,66% menggunakan VGG-19. Penelitian lain oleh [7] menunjukkan bahwa model ini efektif untuk klasifikasi citra buah salak, dengan akurasi 94,58%. [8] juga menemukan bahwa VGG-19 unggul dalam klasifikasi kanker kulit dibandingkan model lainnya. Bahkan, [9] melaporkan akurasi sempurna 100% untuk klasifikasi tingkat kematangan buah pisang menggunakan VGG-19.

Melihat berbagai keberhasilan penelitian tersebut, VGG19 memiliki potensi besar untuk diterapkan dalam deteksi penyakit tanaman padi. Teknologi ini memungkinkan petani untuk mengidentifikasi penyakit pada tanaman secara cepat dan akurat. Hal ini akan membantu dalam pengambilan tindakan pencegahan lebih awal, sehingga kerugian produksi dapat diminimalkan. Selain itu, integrasi teknologi ini dalam sektor pertanian dapat mendorong efisiensi dan keberlanjutan sistem pertanian. Potensi ini menjadikan VGG-19 sebagai alat yang sangat relevan untuk mendukung produktivitas pangan di Indonesia.

Penelitian ini berfokus pada pengembangan sistem deteksi penyakit tanaman padi berbasis VGG19, Pengujian blackbox dilakukan untuk mengevaluasi fungsi sistem deteksi dan rekomendasi penyakit tanaman padi tanpa memeriksa kode internalnya. Diharapkan teknologi ini mampu membantu petani dalam mendiagnosis penyakit tanaman secara efisien, mengurangi biaya produksi, dan meningkatkan hasil panen. Selain itu, penerapan teknologi kecerdasan buatan ini juga mendukung upaya pemerintah dalam menjaga ketahanan pangan nasional. Dengan demikian, penelitian ini diharapkan tidak hanya memberikan solusi inovatif tetapi juga berkontribusi terhadap pengembangan pertanian yang berkelanjutan di Indonesia.

II. METODOLOGI PENELITIAN

Adapun alur tahapan yang dilakukan selama proses penelitian berlangsung dapat dilihat pada gambar di bawah ini.



Gambar 1 Tahap Perancangan Penelitian

A. Tahap Analisis Masalah

Dalam tahap analisis masalah, peneliti melakukan identifikasi masalah utama seperti serangan hama atau penyakit yang menyebabkan tanaman padi mati dan gagal panen, yang kemudian mendasari pendekatan penelitian dengan memanfaatkan berbagai sumber terkait seperti jurnal, buku, dan informasi internet. Selanjutnya, penelitian dilanjutkan dengan studi literatur melalui jurnal dan observasi lapangan untuk identifikasi masalah lebih mendalam, sehingga peneliti dapat membuat interpretasi awal mengenai kebutuhan pengguna terhadap sistem deteksi penyakit tanaman padi yang akan dikembangkan..

B. Tahap Pengembangan Model

1) *Data Collection*: Penelitian ini mengumpulkan data dalam dua tahap, yaitu dengan mengumpulkan gambar tanaman padi sebagai bahan percobaan sistem dan mengumpulkan data set untuk melakukan pelatihan model. Pelatihan model ini dapat diakses di kaggle.com dengan judul "Paddy Doctor: Paddy Disease Classification, Author (Pandarasamy Arjunan)", dan berikut detail dataset yang digunakan.

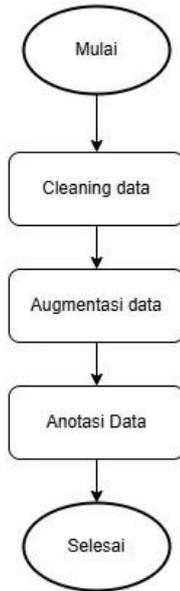
TABEL I
RINCIAN DATASET YANG DIGUNAKAN

Class	Jumlah Gambar
<i>Brown Spot</i>	965
<i>Healthy</i>	1764
<i>Hispa</i>	1594
<i>Dead Heart</i>	1442
<i>Blast</i>	1738
<i>Bacterial Panicle</i>	337
<i>Blight</i>	
<i>Bacterial Leaf Streak</i>	380
<i>Downy Mildew</i>	620
<i>Bacterial Leaf Blight</i>	479
<i>Tungro</i>	1088
Total	10.407

Dalam penelitian ini terdapat jenis pupuk yang digunakan sebagai penanganan penyakit padi, dan beberapa jenis pupuk untuk tanaman padi diantaranya yaitu pupuk organik yang terbuat dari makhluk hidup, urea yang mengandung zat hara,

SP-36 yang dapat menyuburkan tanaman padat dan cair, dan pupuk KCL yang merupakan pupuk organik yang dapat diserap dengan mudah oleh tanaman.

2) *Data Preparation*: Proses preprocessing secara keseluruhan adalah langkah penting dalam mempersiapkan dataset untuk pelatihan model. Hal ini membantu model mengidentifikasi lokasi objek Tanaman padi dalam foto. Setelah diproses, gambar disimpan di direktori output yang telah ditentukan sebelumnya.

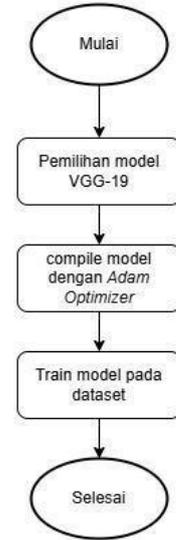


Gambar 2 Alur Pre-processing Data

Tahapana data preparation didalamnya termasuk tahapan pembersihan data yang melibatkan beberapa tahap seperti cropping dan peningkatan kontras. Dilanjutkan dengan augmentasi data yang menghasilkan variasi tambahan pada dataset pelatihan dengan melakukan transformasi seperti rotasi, pergeseran, shear, zoom, dan flip horizontal. Fungsi yang akan digunakan yaitu 'ImageDataGenerator' dari TensorFlow untuk melakukan augmentasi data tersebut. Setelah itu, tahapan anotasi data yang merupakan fase dimana label dan informasi lainnya diberikan pada benda atau area tertentu dalam ilustrasi.

3) *Modelling*: Tahap pertama yang dilakukan dalam training data adalah pemilihan model pre-trained yaitu menggunakan model VGG-19 yang telah dilatih sebelumnya sebagai dasar untuk pelatihan pada dataset ImageNet. Langkah selanjutnya yaitu dengan mengoptimasi dengan algoritma Adam Optimizer dimana langkah ini berguna untuk melatih model dengan kebutuhan yang menyesuaikan parameter seperti laju pembelajaran (learning rate). Lalu model dilatih dengan data train sebelumnya. Kemudian mengatur jumlah epoch, ukuran, dan hiperparameter seperti tingkat pembelajaran. Dan selanjutnya file disimpan yang kemudian bisa digunakan dengan mobile flutter. Dalam pengajaran mesin, dataset dibagi menjadi 90 persen dan 10 persen untuk pelatihan dan pengujian. Pembagian sepuluh persen data digunakan untuk menguji kinerja model setelah proses pelatihan, memastikan bahwa model menghafal data pelatihan dan mampu membuat prediksi

yang akurat. Ini membuat penilaian kemampuan model menjadi lebih objektif.



Gambar 3 Alur Training Data

4) *Evaluasi Data*: Setelah tahap training data selesai, dilakukan evaluasi performa model menggunakan confusion matrix dan classification report. Dengan menggunakan confusion matrix dan classification report, menghasilkan informasi lebih lanjut tentang sejauh mana model dapat membedakan kelas-kelas dan performa model secara keseluruhan serta memberikan wawasan tentang trade-off antara mengidentifikasi positif dan menghindari false positive atau false negative. Hasil ini dapat membantu memahami kekuatan dan kelemahan model dalam klasifikasi. Misalnya, akurasi yang dihitung dengan menggunakan persamaan (1), Nilai precision dihitung menggunakan formula(2), recall dihitung dengan menggunakan formula (3), dan f1-score dihitung menggunakan formula (4).

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

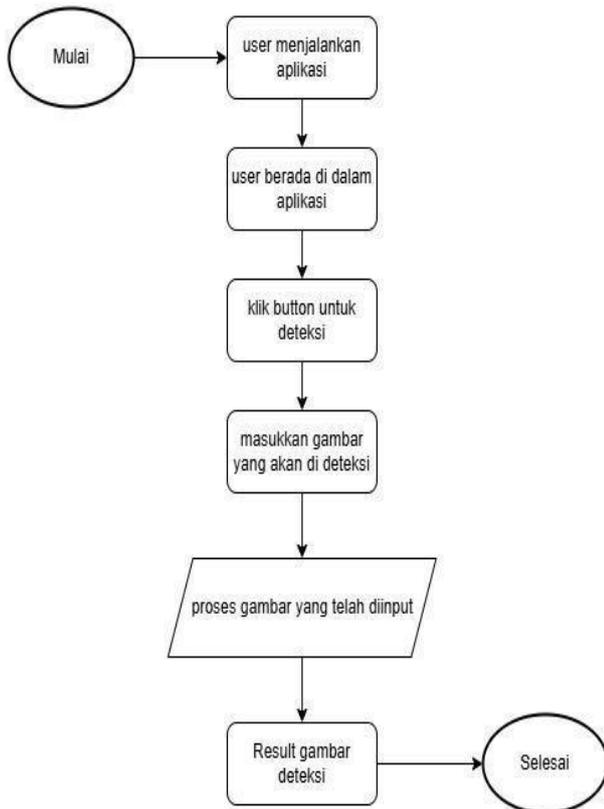
$$\text{F1 - Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

C. Tahap Pengembangan Sistem

Analisis sistem diperlukan untuk mengetahui kebutuhan dan konfigurasi yang digunakan dalam pengembangan produk penelitian, meliputi spesifikasi komputer minimal seperti CPU Intel Core i5 11th Gen @2.70GHz, RAM 8GB, storage 512GB, OS (Windows/Linux/macOS), dan GPU NVIDIA GeForce RTX 3060; aplikasi dan library pendukung seperti Google Colab, Visual Studio Code, Python, Pandas, TensorFlow,

NumPy, Matplotlib, Keras, dan Flutter; serta input berupa gambar padi dan output berupa identifikasi penyakit tanaman padi beserta rekomendasi pupuk yang diperlukan.

1) *Desain Sistem*: Tahap pertama pada pengembangan sistem berbasis mobile adalah melakukan desain sistem. Dengan menggunakan metode ini, sistem deteksi penyakit tanaman padi dapat dikembangkan secara sistematis dan terdokumentasi. Desain sistem dibuat menggunakan diagram alur dan desain prototype menggunakan figma. Berikut diagram alur kerja sistem deteksi penyakit tanaman padi.



Gambar 4 User Aplikasi Diagram

Selain menggunakan diagram alur, desain sistem juga dirancang menggunakan desain wireframe aplikasi deteksi penyakit tanaman padi berbasis mobile.

Sistem membuat user dapat memilih opsi dalam penggunaan fitur deteksi yaitu melalui ambil gambar langsung menggunakan kamera secara real-time ataupun menggunakan gambar yang sudah tersedia di galeri user untuk langsung di upload dan selanjutnya dideteksi.

2) *Implementasi Sistem*: Setelah melakukan desain sistem, kegiatan yang dilakukan untuk Pengembangan Sistem Deteksi Penyakit Tanaman padi adalah membuat sebuah program berbasis mobile yang dapat diakses dengan internet melalui sistem unduh di Playstore berdasarkan rancangan sistem yang telah dibuat.

3) *Pengujian Sistem*: Pengujian produk deteksi penyakit tanaman padi adalah langkah kritis dalam memastikan bahwa sistem tersebut dapat berfungsi dengan baik dan dapat

diandalkan saat digunakan di lingkungan dunia nyata. Produk sistem deteksi penyakit tanaman padi ini akan dilakukan pengujian dengan menggunakan metode black-box, dimana pengujiannya berfokus pada fitur program yang telah dibuat menggunakan bahasa pemrograman python untuk menguji seberapa akurat hasil pelatihan data pada sistem ini. Pengujian dilakukan dengan memanfaatkan dataset berupa gambar daun padi yang terkena penyakit yang telah diekstrak menjadi frame sebelumnya, sehingga menghasilkan output deteksi penyakit tanaman padi.

D. Tahap Pelaporan

Tahap terakhir dalam pengembangan sistem deteksi penyakit tanaman padi adalah pelaporan, yang merupakan tahapan penting untuk menyampaikan informasi secara efektif kepada pengguna, pemangku kepentingan, dan pihak terkait.

III. HASIL DAN PEMBAHASAN

A. Tahap Pengembangan Model

Tahap pengembangan model adalah proses di mana menerapkan algoritma pada data untuk melatih mesin, sehingga mesin dapat belajar dan menghasilkan model yang berguna. Model ini akan digunakan untuk mengidentifikasi penyakit pada tanaman padi. Proses ini terdiri dari empat langkah berurutan, yaitu:

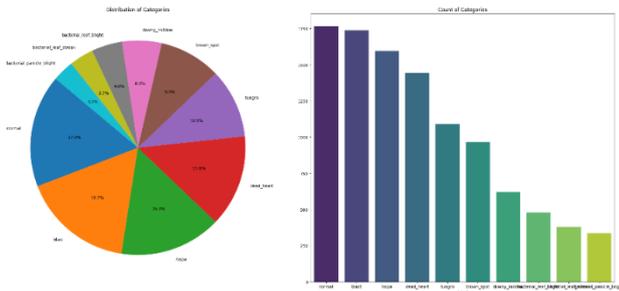
1) *Pengumpulan Data (Data Collection)*: Penelitian ini menggunakan dataset dengan judul “Paddy Disease Classification” oleh Mehak Iftikhar (2023) di situs kaggle <https://www.kaggle.com/competitions/paddy-disease-classification>.



Gambar 5 Dataset

Dataset ini mencakup sepuluh (10) jenis penyakit yang dapat menyerang tanaman padi, antara lain: *Blast*, *hispa*, *dead heart*, *tungro*, *normal*, *downy mildew*, *brownspot*, *bacterial leaf blight*, *bacterial leaf streak*, *bacterial panicle blight*.

2) *Persiapan Data (Data Preparation)*: Pada tahap ini peneliti membagi 3 proses tahapan yaitu, cleaning, augmentasi, dan anotasi. Tahap ini dapat membuat visualisasi distribusi kategori penyakit dengan menggunakan dua jenis grafik berbeda sebagai berikut.



Gambar 6 Hasil Visualisasi Kategori Penyakit

Dataset terdiri dari 10.407 gambar dengan ukuran total sekitar 1,9 GB. Sebelum dataset ini dapat digunakan, harus melalui tahap persiapan data yang sistematis diawali dengan cleaning data. Tahapan cleaning data bertujuan untuk meningkatkan kualitas data sebelum digunakan untuk pemodelan atau analisis dengan output sebagai berikut.

```
Issue checks completed. 388 issues found in the dataset. To
see a detailed report of issues found, use imagelab.report().
  issue_type num_images
0 near_duplicates      242
1 exact_duplicates    146
2 dark                  0
3 light                0
4 odd_aspect_ratio     0
5 blurry               0
6 low_information      0
7 odd_size             0
8 grayscale            0
```

Kode Program 1 Hasil Cleaning Data

Setelah data bersih, pada tahap berikutnya yaitu augmentasi data. Peneliti menggunakan image Datagenerator dari TensorFlow agar dapat menghasilkan generator data untuk pelatihan model. Proses augmentasi data dilakukan untuk meningkatkan variasi gambar dalam dataset pelatihan dengan flipping gambar secara horizontal. Berikut total data setelah augmentasi yang menghasilkan output beberapa contoh gambar.

```
Found 10407 images belonging to 10 classes.
Jumlah data asli      : 10407
Jumlah parameter augmentasi : 3
Estimasi data hasil augmentasi : 31221
Total estimasi data dilihat model: 41628
```

Kode Program 2 Total Data Setelah Augmentasi



Gambar 7 Hasil Augmentasi Data

Selanjutnya adalah tahapan anotasi data atau proses pemberian label yang menghasilkan gambar dengan label yang sesuai dengan masing-masing classnya.

3) *Modeling*: Dalam penelitian ini peneliti menggunakan model VGG-19 yang menerapkan optimasi Adam untuk meningkatkan kinerja. Langkah-langkahnya diawali dengan pemilihan pre-trained model VGG-19 dan melakukan instalasi library untuk membangun model.

```
0 | # Import library utama
1 | import os
2 | import cv2
3 | import warnings
4 | import numpy as np
5 | import pandas as pd
6 | import seaborn as sns
7 | import matplotlib.pyplot as plt
8 | import matplotlib.image as mpimg
9 |
10| # TensorFlow dan Keras
11| import tensorflow as tf
12| from keras import Sequential
13| from keras.layers import *
14| from keras.models import Model
15| from keras.callbacks import
EarlyStopping
16| from keras.preprocessing import
image
17| from tensorflow.keras.optimizers
import Adam
18| from tensorflow.keras.losses import
BinaryCrossentropy
19| from tensorflow.keras.applications
import VGG19
20| from
tensorflow.keras.preprocessing import
image dataset from directory
```

Kode Program 3 Import Library

Selanjutnya melakukan tahap feature extraction model dengan model VGG-19 yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet. Dengan parameter `include_top=False`, lapisan klasifikasi akhir dihilangkan sehingga model hanya digunakan sebagai ekstraktor fitur.

Setelah melalui tahap feature extraction maka langkah selanjutnya yaitu menambahkan fully connected layers untuk melakukan klasifikasi menggunakan Keras dengan arsitektur berlapis berurutan (Sequential). Untuk mencegah overfitting, lapisan Dropout dengan probabilitas 0,35 secara acak menonaktifkan 35% neuron selama pelatihan. Selanjutnya, pola non-linear ditangkap melalui lapisan Dense dengan 128 neuron dan aktivasi ReLU. Pola output Dense dengan 10 neuron dan aktivasi softmax menghasilkan probabilitas untuk setiap kelas.

Tahap selanjutnya adalah compile model dengan Adam Optimizer dengan learning rate 0.001, fungsi loss sparse_categorical_crossentropy, dan metrik akurasi untuk mengevaluasi performa. Dilanjutkan dengan train model pada dataset, dilakukan sebanyak 100 epoch yang berlangsung 4 jam 27 menit. Hasil terbaik didapatkan pada epoch 36 dengan nilai akurasi sebesar val_accuracy: 0.9335 dan nilai loss di angka val_loss: 0.3126.

Setelah selesai pelatihan model, yaitu menyimpan model default dengan model.h5 dan ubah format file menjadi .tflite.

4) *Evaluasi Data*: Untuk mengetahui performa model yang telah dilatih sebelumnya, dilakukan evaluasi model dalam beberapa tahap dengan hasil sebagai berikut.

33/33 ————— 7s
 224ms/step - accuracy: 0.9286 - loss: 0.2294
 Validation Loss: 0.2533901035785675
 Validation Accuracy: 0.925000011920929

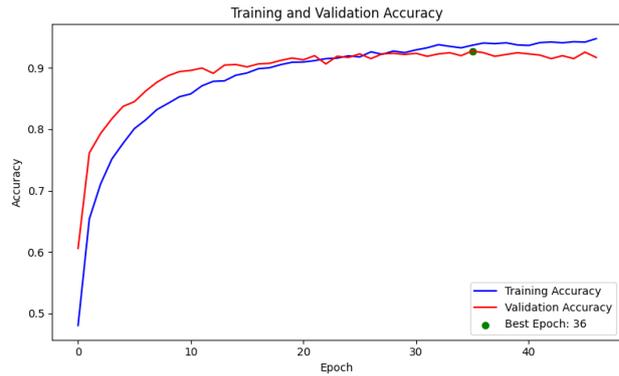
Kode Program 4 Hasil Evaluasi Model

Selanjutnya menampilkan classification report untuk menghitung evaluasi matriks seperti accuracy, precision, recall, dan f1-score. Hasil klasifikasi menunjukkan bahwa model memiliki kinerja dengan akurasi keseluruhan 93% yang artinya model memprediksi benar dengan tingkat keakuratan 93% dengan hasil keluaran sebagai berikut.

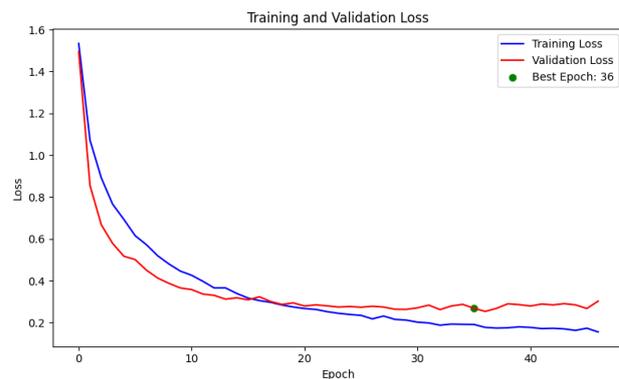
	precision	recall	f1-score	support
bacterial_leaf_blight	0.95	0.91	0.93	44
bacterial_leaf_streak	0.98	0.92	0.91	38
bacterial_panicle_blight	0.79	0.88	0.84	26
blast	0.96	0.96	0.96	197
brown_spot	0.94	0.89	0.91	106
dead_heart	0.97	0.95	0.96	146
downy_mildew	0.88	0.93	0.90	54
hispa	0.93	0.91	0.92	160
normal	0.98	0.92	0.91	168
tungro	0.89	0.91	0.90	101
accuracy			0.93	1040
macro avg	0.91	0.92	0.91	1040
weighted avg	0.93	0.93	0.93	1040

Gambar 8 Hasil Classification Report

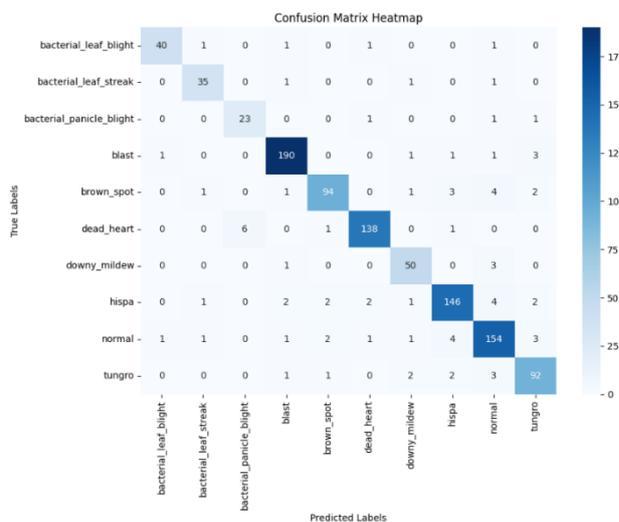
Grafik performa model secara keseluruhan dan visualisasi data pelatihan model juga ditampilkan dalam bentuk training dan akurasi validasi pada grafik berikut.



Gambar 9 Accuracy Data Training Dan Validasi



Gambar 10 Data Training Dan Validasi Loss



Gambar 11 Hasil Confusion Matrix

Tabel spesifikasi dari hasil confusion matrix dengan F1-Score tertinggi dan terendah memiliki penjelasan sebagai berikut.

TABEL III
 SPESIFIKASI MATRIX

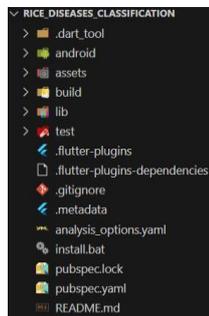
Class	Best F1-Score	Data	Penjelasan
Blast	0.96	197	Model bekerja dengan sangat baik, 190 dari 197 data

			diklasifikasi dengan benar
Dead Heart	0.96	146	Model bekerja dengan sangat baik, 138 dari 146 data diklasifikasi dengan benar
Normal	0.94	163	Model bekerja dengan sangat baik, 154 dari 163 data diklasifikasi dengan benar
Hispa	0.93	160	Model bekerja dengan sangat baik, 146 dari 160 data diklasifikasi dengan benar
Bacterial_Panicle_Blight	0.85	29	Terdapat kesalahan umum, 6 data bacterial panicle blight diklasifikasi sebagai dead heart

B. Tahap Pengembangan Sistem

Tahap pengembangan sistem informasi berbasis mobile pada penelitian ini memiliki 2 tahap yaitu seperti berikut.

1) *Implementasi Sistem*: Pengembangan aplikasi menggunakan framework flutter, kemudian instalasi flutter SDK, dan text editor seperti VS code dengan struktur folder sebagai berikut.



Gambar 12 Struktur Folder Mobile Sistem Deteksi

Implementasi sistem termasuk didalamnya berbagai logika kode program untuk integrasi model ke dalam flutter, penambahan logika dan widget ke mainpage, akses kamera, dan tampilan hasil deteksi.

2) *Pengujian Sistem*: Pengujian Black Box dilakukan dengan melakukan pengamatan pada hasil input dan output oleh petani secara langsung dengan hasil sebagai berikut.

TABEL IIIII
PENGUJIAN BLACKBOX

Fitur	Input	Output	Hasil
Home	Klik "Mulai Deteksi", klik "Riwayat Deteksi"	Navigasi ke halaman upload gambar / navigasi ke halaman Riwayat	Berhasil
Button Kamera	Klik icon kamera di	Aplikasi membawa ke	Berhasil

(Floating Action Button)	tengah bawah, ambil foto, klik OK	halaman hasil deteksi	
Halaman Hasil Deteksi	Menampilkan gambar padi, nama penyakit, confidence, rekomendasi pupuk	Data tampil sesuai hasil model, button "Ulangi Deteksi" berfungsi	Berhasil
Button Ulangi Deteksi	Klik button "Ulangi Deteksi", pilih gambar baru dari galeri	Gambar dan hasil update dengan data baru	Berhasil
Halaman Panduan Penyakit Padi	Klik accordion salah satu penyakit	Detail penyakit (definisi, gejala, penanganan, pupuk) tampil	Berhasil
Riwayat Deteksi - Tampil	Lihat daftar riwayat	Daftar riwayat tampil sesuai scan yang pernah dilakukan	Berhasil
Riwayat Deteksi - Hapus Riwayat	Klik button "Hapus Riwayat"	Data riwayat terhapus	Berhasil

Setelah dilakukan percobaan uji sistem oleh petani langsung dengan 4/5 gambar yang diinput gambar terdeteksi dengan benar. Kemudian dapat dihitung dengan formula blackbox ($\frac{\text{berhasil}}{\text{total}} \times 100\% = 100\%$). Jadi setelah dilakukan perhitungan ($\frac{4}{5} \times 100\% = 80\%$) dan menunjukkan bahwa sistem berhasil/valid. Skenario dan langkah-langkah penggunaan sistem deteksi dimulai ketika user membuka aplikasi, sistem menampilkan layout sebagai pengenalan aplikasi dengan tampilan berikut.



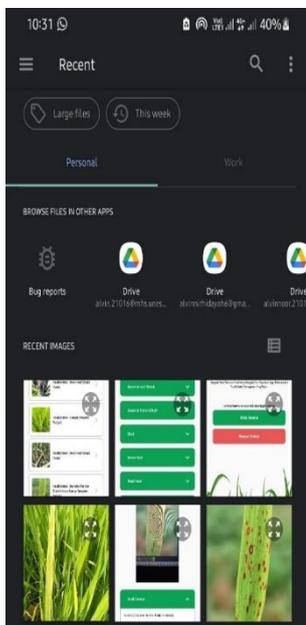
Gambar 13 Home Page

Selanjutnya user bisa memilih scanning dari galeri atau melihat riwayat deteksi yang sudah pernah dilakukan dengan tampilan halaman sebagai berikut.



Gambar 14 Main Page

Saat user memilih mulai deteksi dan mengambil gambar dari galeri akan otomatis ke direct local foto untuk memilih gambar kondisi padi untuk dideteksi dengan tampilan sebagai berikut.



Gambar 15 Local Photo Untuk Deteksi

Sedangkan jika user memilih untuk klik tombol icon kamera otomatis akan muncul deteksi langsung melalui kamera secara real-time. Fitur ini bisa digunakan jika user ingin melakukan scanning objek secara langsung.



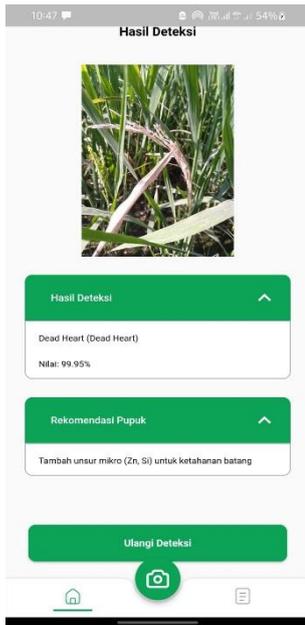
Gambar 16 Screen Kamera Deteksi

sedangkan dari halaman utama jika user klik opsi riwayat deteksi akan muncul hasil deteksi yang sudah pernah dilakukan sebelumnya dengan tampilan seperti berikut.



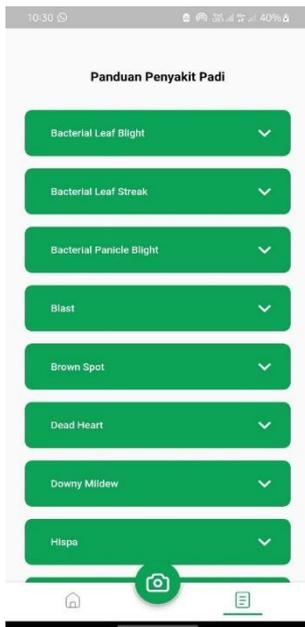
Gambar 17 Riwayat Deteksi

Hasil pemindaian akan mengidentifikasi jenis penyakit pada tanaman padi. Jenis penyakit, tingkat akurasi deteksi persentase, dan rekomendasi pupuk pencegahan disimpan dalam aplikasi. Pengguna juga dapat melakukan deteksi ulang jika diperlukan. Halaman hasil deteksi seperti berikut.



Gambar 18 Hasil Deteksi

Ketika user memilih menu “guide” pengguna akan diarahkan ke halaman panduan penyakit berikut.



Gambar 19 Halaman Panduan Penyakit Tanaman Padi

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melalui semua tahapan penelitian, berikut kesimpulan yang didapatkan.

- 1) Penelitian ini mengembangkan sistem deteksi penyakit tanaman padi berbasis mobile menggunakan model VGG-19. Deteksi dini penyakit sangat penting untuk mencegah kehilangan hasil panen. Dengan metode transfer learning, sistem ini mencapai akurasi 93% dalam mengidentifikasi berbagai penyakit pada

tanaman padi. Selain mendeteksi penyakit, sistem ini juga memberikan rekomendasi nitrogen yang sesuai, membantu petani merawat tanaman dengan lebih baik.

- 2) VGG-19 terbukti efektif dalam mendeteksi penyakit tanaman padi. Analisis menggunakan metrik akurasi, presisi, recall, dan F1-score menunjukkan bahwa model ini menghasilkan tingkat akurasi yang tinggi dan keseimbangan yang baik antara presisi dan recall. Temuan ini menegaskan bahwa VGG-19 dapat menjadi alat yang berguna untuk diagnosis penyakit tanaman padi, mendukung pengelolaan pertanian yang lebih baik.
- 3) Pembangunan sistem deteksi penyakit tanaman padi ini dilakukan dengan menggunakan framework flutter berbasis mobile. Proses ini melibatkan tahapan perancangan, dimana model yang sudah dilatih akan disimpan menjadi .tflite.

B. Saran

Dari semua hasil yang dilakukan dan tahap yang di kerjakan ada beberapa saran yang perlu dilakukan pda penelitian berikutnya.

- 1) Untuk pengembangan lebih lanjut, disarankan untuk menguji sistem ini di lapangan dengan melibatkan lebih banyak petani.
- 2) Untuk pengembangan lebih lanjut agar flexibel aplikasi bisa dipublish di playstore untuk digunakan multi device sehingga penggunaan lebih efisien.
- 3) Menggunakan real time deteksi ketika selain padi maka output yang dihasilkan akan menghasilkan scan gagal.

REFERENSI

- [1] Marwanti Marwanti, Setyono Hari Adi, Hendri Sosiawan, Muhrizal Sarwani, Gatot Irianto, and Mohammad Ismail Wahab, “Disrupsi Sistem Produksi Padi Nasional: Mampukah Indonesia Memenuhi Kebutuhan Beras di Tahun 2045?,” *J. Trit.*, vol. 14, no. 2, pp. 403–421, 2023, doi: 10.47687/jt.v14i2.588.
- [2] A. Siddharta, “Food industry in Malaysia - statistics & facts,” *statista.com*, 2024. <https://www.statista.com/topics/11001/food-industry-in-malaysia/#topicOverview> (accessed Nov. 14, 2024).
- [3] D. Purnomo and P. N. Utami, “Analisis Produksi Padi Di Indonesia,” *Proceeding of The URECOL*, vol. 4, no. 3, pp. 224–230, 2024.
- [4] M. Rijal, A. M. Yani, and A. Rahman, “Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan Deep Learning dengan Model CNN,” *J. Teknol. Terpadu*, vol. 10, no. 1, pp. 56–62, 2024, doi: 10.54914/jtt.v10i1.1224.
- [5] L. Messeri and M. J. Crockett, “Artificial intelligence and illusions of understanding in scientific research,” *Nature*, vol. 627, no. 8002, pp. 49–58, 2024, doi: 10.1038/s41586-024-07146-0.
- [6] A. A. SHELEMO, “No Title,” *Nucl. Phys.*, vol. 13, no. 1, pp. 104–116, 2023.
- [7] I. I. A. Widyan, “Implementasi Convolutional Neural Network Menggunakan Arsitektur Visual Geometry Group -19 (Vgg-19) Untuk,” vol. 19, 2024.
- [8] I. A. Kandhro *et al.*, “Performance evaluation of E-VGG19 model: Enhancing real-time skin cancer detection and classification,” *Heliyon*, vol. 10, no. 10, p. e31488, 2024, doi: 10.1016/j.heliyon.2024.e31488.
- [9] F. A. H. M. K. B. S. A. W. P. S. Alala, “Jurnal tika,” *J. TIKFA Fak. Ilmu ...*, vol. xx, no. 1, pp. 85–91, 2024, [Online]. Available: <http://download.garuda.kemdikbud.go.id/article.php?article=3285185&val=21707&title=Game Edukasi Pengenalan Nama-Nama Sayuran Menggunakan Metode Development Life Cycle>