

Performance Comparison of VGG-19 and DenseNet-121 Architectures for Rice Plant Disease

Perbandingan Performa Arsitektur VGG-19 dan DenseNet-121 Untuk Klasifikasi Penyakit Tanaman Padi

Istimewa Megahaztuti¹, Bambang Yuwono², Bagus Muhammad Akbar³, Heru Cahya Rustamaji⁵, Simon Pulung Nugroho⁵

^{1,2,3,4,5} Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

¹123190001@student.upnyk.ac.id, ^{2*}bambangy@upnyk.ac.id,

^{3*}bagusmuhammadakbar@upnyk.ac.id ^{4*}herucr@upnyk.ac.id ^{5*}simon.pulung@upnyk.ac.id

Informasi Artikel

Received: January 2025

Revised: February 2025

Accepted: April 2025

Published: June 2025

Abstract

Rice (Oryza sativa L.) is a major food source that often faces the challenge of crop failure due to various plant diseases. These diseases not only reduce productivity, but are also exacerbated by farmers' limited knowledge in recognizing symptoms and reliance on manual diagnosis that takes a long time. This study aims to compare the performance of two Convolutional Neural Network (CNN) architectures, namely VGG-19 and DenseNet-121, in classifying rice plant diseases based on image processing. Low accuracy and overfitting are problems that are often observed when small datasets are used to train deep learning models, such as Convolutional Neural Networks (CNN). In this study, modifications were made to the VGG-19 and DenseNet-121 architectures so that the model can achieve good accuracy and reduce the risk of overfitting despite using small datasets. The dataset consists of 11,790 images in 9 classes, which are divided into 7545 training data, 1887 validation data, and 2358 testing data. After the training data is segmented, the total number of images in the dataset is 23,580. Before modification, the DenseNet-121 model achieved the highest accuracy of 50.45% and F1-score of 44.83%, while VGG-19 achieved the highest accuracy of 13.84% and F1-score of 7.39%. After making modifications to both models, the test results show that DenseNet-121 achieved an accuracy of 97.76% and F1-score of 96.31%, while VGG-19 achieved an accuracy of 84.82% and F1-score of 87.52%. The advantage of DenseNet-121 lies in its ability to process features more efficiently, resulting in more accurate predictions than VGG-19. This research contributes to the selection of the best model architecture to

support automatic diagnosis of rice plant diseases, which is relevant to the agricultural sector in Indonesia.

Keywords: Classification, Rice, CNN, VGG-19, DenseNet-121, Architecture Modification

Kata kunci: Klasifikasi, Padi, CNN, VGG-19, DenseNet-121, Modifikasi Arsitektur

Abstrak

Tanaman padi (*Oryza sativa L.*) merupakan sumber pangan utama yang sering menghadapi tantangan kegagalan panen akibat berbagai penyakit tanaman. Penyakit ini tidak hanya menurunkan produktivitas, tetapi juga diperburuk oleh keterbatasan pengetahuan petani dalam mengenali gejala serta ketergantungan pada diagnosis manual yang membutuhkan waktu lama. Penelitian ini bertujuan membandingkan performa dua arsitektur *Convolutional Neural Network* (CNN), yaitu VGG-19 dan DenseNet-121, dalam mengklasifikasi penyakit tanaman padi berbasis pengolahan citra. Akurasi yang rendah dan *overfitting* adalah masalah yang sering diamati saat dataset kecil digunakan untuk melatih model *deep learning*, seperti *Convolutional Neural Networks* (CNN). Pada penelitian ini dilakukan modifikasi pada arsitektur VGG-19 dan DenseNet-121 agar model dapat mencapai akurasi yang baik dan mengurangi risiko *overfitting* meskipun menggunakan dataset kecil. Dataset terdiri dari 11.790 gambar dalam 9 kelas, yang dibagi menjadi 7545 data *training*, 1887 data validasi, dan 2358 data pengujian. Setelah data *training* diaugmentasi total gambar dalam dataset menjadi 23.580. Sebelum modifikasi, model DenseNet-121 mencapai akurasi tertinggi sebesar 50.45% dan F1-score 44.83%, sedangkan VGG-19 mencapai akurasi tertinggi sebesar 13.84% dan F1-score 7.39%. Setelah dilakukan modifikasi pada kedua model, hasil pengujian menunjukkan bahwa DenseNet-121 mencapai akurasi 97.76% dan F1-score 96.31%, sedangkan VGG-19 mencapai akurasi 84.82% dan F1-score 87.52%. Keunggulan DenseNet-121 terletak pada kemampuannya mengolah fitur secara lebih efisien, menghasilkan prediksi yang lebih akurat dibandingkan VGG-19. Penelitian ini berkontribusi dalam pemilihan arsitektur model terbaik untuk mendukung diagnosis penyakit tanaman padi secara otomatis, yang relevan bagi sektor pertanian di Indonesia.

1. Pendahuluan

Indonesia sebagai negara agraris memiliki mayoritas penduduk yang bekerja dalam sektor pertanian, salah satunya adalah pertanian padi yang menghasilkan beras dan menjadi salah satu makanan pokok penduduk Indonesia. Meskipun demikian, hasil panen padi seringkali terhambat

oleh berbagai penyakit tanaman yang menyerang [1]. Serangan organisme pengganggu tanaman (OPT), terutama pada bagian daun padi, sering kali menjadi penyebab utama gagal panen. Hambatan ini diperburuk oleh keterbatasan pengetahuan petani dalam mengenali gejala penyakit dan ketergantungan pada proses diagnose manual yang memerlukan waktu lebih lama [2]. Oleh karena itu, pengamatan pada daun padi dapat menjadi langkah awal dalam mendeteksi penyakit padi dan menerapkan teknologi pertanian yang mendukung intensifikasi guna meningkatkan hasil panen. VGG-19 dan DenseNet-121 merupakan dua arsitektur CNN yang cukup populer dan sering digunakan untuk tugas klasifikasi gambar, terutama dalam aplikasi seperti deteksi penyakit tanaman. Keduanya dianggap sebagai "backbone" yang kuat dalam tugas-tugas visi komputer karena kemampuannya untuk menangani data gambar dengan baik[3].

VGG-19 terkenal sebagai model klasik yang mudah dipahami dan diterapkan, dengan arsitektur yang konsisten (menggunakan blok convolusi 3x3 yang berulang) [4]. Sementara DenseNet121 hadir sebagai arsitektur lebih modern yang memperkenalkan mekanisme "dense connections" yang lebih efisien dalam penanganan fitur [5]. Sehingga dengan reputasi kedua model ini dalam klasifikasi gambar, membandingkan keduanya cukup relevan, khususnya untuk mengevaluasi performa keduanya dalam deteksi penyakit tanaman yang berbasis pada tekstur, pola, dan warna daun.

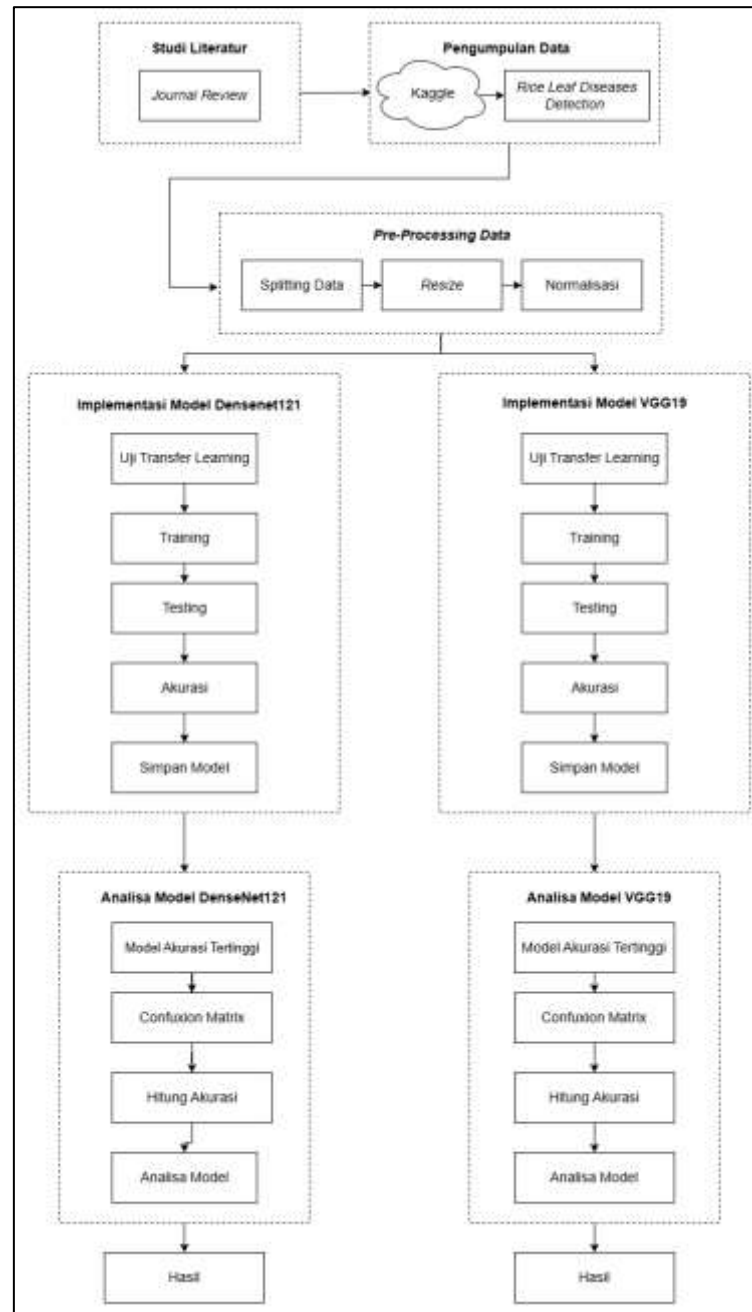
Akurasi yang rendah dan *overfitting* adalah masalah yang sering diamati saat dataset kecil digunakan untuk melatih model *deep learning*, seperti *Convolutional Neural Networks* (CNN). Model besar dan kompleks seperti ResNet dan VGG memiliki kesulitan berkonvergensi pada dataset kecil, terutama ketika menggunakan metode *deep learning* standar [6]. Dataset yang digunakan pada penelitian ini terdiri dari 23.580 data hasil augmentasi, yang tergolong kecil untuk melatih CNN yang sangat dalam. Hal ini didukung oleh penelitian [7] yang menyatakan bahwa dataset CIFAR-10 dengan 60.000 gambar dianggap kecil untuk melatih CNN yang sangat dalam. Oleh karena itu, pada penelitian ini dilakukan modifikasi pada arsitektur VGG-19 dan DenseNet-121 agar model dapat mencapai akurasi yang baik dan mengurangi risiko *overfitting* meskipun menggunakan dataset kecil. Setelah dimodifikasi, performa kedua model akan dievaluasi untuk menentukan model terbaik dalam mengklasifikasikan penyakit tanaman padi. Hasil penelitian ini tidak hanya relevan bagi sektor pertanian, tetapi juga dapat menjadi referensi dalam penggunaan model CNN untuk tugas klasifikasi gambar di bidang lain.

2. Metode/Perancangan

Penelitian ini menggunakan metodologi kuantitatif tipe eksperimental dengan tahapan yaitu pengumpulan data, *pre-processing* data, implementasi model, pelatihan dan validasi model, dan pengujian model. Dataset yang digunakan yaitu dataset sekunder dari Kaggle (Yuji Itadori) digunakan, terdiri dari 11.801 citra dengan format .jpg, .jpeg, dan .png. Dataset dibagi menjadi 7.552 data pelatihan, 1.888 data validasi, dan 2.361 data pengujian. *Pre-processing* data meliputi penyesuaian ukuran citra dan normalisasi nilai piksel untuk meningkatkan efisiensi pelatihan model. Implementasi model menggunakan arsitektur 19 lapisan konvolusi dengan pengaturan pooling layer untuk mentransformasi fitur secara bertahap. Implementasi model DenseNet121 menggunakan koneksi langsung antar lapisan (*dense connections*) untuk mempertahankan informasi fitur di setiap lapisan. Pelatihan dan validasi model dilakukan secara terpisah pada kedua model dengan hyperparameter yang disesuaikan, seperti learning rate,

optimizer, dan *epoch* sebanyak 30 iterasi. Pengujian model menggunakan data pengujian dengan evaluasi berdasarkan akurasi, dan F1-score.

Alur dalam penelitian ini dapat dilihat pada **Gambar 1**.



Gambar 1. Alur Penelitian

Gambar di atas menunjukkan pedoman yang diterapkan didalam melakukan penelitian ini. Ada beberapa tahapan yaitu studi literatur, pengumpulan data *pre-processing* yang di dalamnya terdapat *resize*, *normalisasi* dan *splitting data*.

2.1. Pengumpulan Data

Proses pengumpulan data pada penelitian ini menggunakan *dataset* sekunder yang bersumber dari website Kaggle dengan nama *Rice leaf Diseases Detection* yang menyediakan sumber data terbuka (*open source*) yang kemudian *dataset* tersebut disimpan dan diakses dari *google drive* untuk kemudahan pemrosesan. Pada *dataset* dari situs kaggle ini menyajikan 11.801 data citra yang terbagi menjadi sembilan kategori penyakit tanaman padi, dengan rincian yaitu *Sheath Blight* 1629 citra, *Rice Hispa* 1299 citra, *Neck Blast* 1000 citra, *Narrow Brown Leaf Spot* 954 citra, *Leaf Scald* 1332 citra, *Healthy Rice Leaf* 1095 citra, *Leaf Blast* 1749 citra, *Bacterial Leaf Blight* 1197 citra, dan *Brown Spot* 1546 citra.

2.2. VGG-19

Arsitektur VGG-Net, yang dikembangkan oleh tim *Visual Geometry Group* dari Universitas Oxford pada tahun 2014, menjuarai *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) dengan meraih posisi kedua dalam kategori klasifikasi gambar. Pada tahun 2015, Simonyan dan Zisserman dari Universitas Oxford memperkenalkan model jaringan saraf dengan 19 lapisan, terdiri dari 16 lapisan konvolusi dan 3 lapisan *fully connected*. Model ini menggunakan filter 3x3 dengan stride dan padding sebesar 1, dan dilengkapi dengan lapisan max-pooling 2x2 dengan stride sebesar 2. Arsitektur tersebut dikenal sebagai VGG-19 [4].

VGGNet dirancang dengan arsitektur sederhana yang terdiri dari komponen-komponen seragam, yaitu lapisan konvolusi dan pooling. Inovasi ini merupakan penyempurnaan dari AlexNet, di mana filter berukuran besar dengan kernel 11 dan 5 pada lapisan konvolusi pertama dan kedua, secara berturut-turut, digantikan oleh beberapa filter berukuran 3x3. VGG-19 memiliki struktur yang terdiri dari serangkaian blok konvolusi yang konsisten, diikuti oleh lapisan pooling yang juga konsisten. Setiap lapisan konvolusi menggunakan filter berukuran 3x3 dengan stride 1 dan padding "same", sementara lapisan pooling menggunakan ukuran pool 2x2 dengan stride 2, yang menjaga ukuran output tetap sama dengan ukuran input.

2.3. DenseNet-121

DenseNet-121 adalah jenis CNN di mana setiap lapisan terhubung langsung dengan semua lapisan berikutnya, sehingga keluaran dari setiap lapisan digunakan kembali sebagai input untuk lapisan-lapisan berikutnya [5].

Arsitektur ini terdiri dari empat *Dense Block*. Setiap *Dense Block* terdiri dari beberapa *convolutional layers* yang terhubung secara padat (*dense connections*). Output dari setiap layer dalam *Dense Block* menjadi input untuk semua layer berikutnya di dalam block yang sama. *Dense Block* Mengandung beberapa *convolutional layers* dengan *batch normalization* dan ReLU activation. Setiap layer di *dense block* menerima input dari semua layer sebelumnya, termasuk layer input *dense block* tersebut. Ini memastikan bahwa informasi dari layer-layer awal tetap tersedia untuk layer-layer selanjutnya. Lalu Output dari setiap layer digabungkan (*concatenate*) dengan input *dense block* untuk menghasilkan input bagi layer berikutnya.

Setelah *dense block*, di dalam arsitektur DenseNet juga terdapat *transition layer*. *Transition layer* ini terletak di antara *Dense Blocks* untuk mengurangi dimensi dan jumlah fitur, menjaga efisiensi komputasi. Disini ada sebanyak tiga *transition layer* yang mana setiap *transition layer* terdiri dari beberapa komponen, pertama yaitu *Convolutional Layer* (Conv): Menggunakan 1x1

convolutions untuk mengurangi jumlah fitur. Kedua yaitu *Pooling Layer*: Menggunakan 2x2 *average pooling* untuk mengurangi dimensi spasial dari fitur. Bagian selanjutnya yaitu *softmax layer*. Ini merupakan layer terakhir dalam jaringan yang melakukan klasifikasi. Layer ini menghasilkan probabilitas untuk setiap kelas yang mungkin.

3. Hasil dan Pembahasan

3.1. Hasil Training Model VGG-19 Setelah Modifikasi

Tabel 1. Hasil Proses Train Model Fine Tuning Model VGG-19

| <i>Epoch</i> | <i>Training Loss</i> | <i>Training Accuracy</i> | <i>Validation Loss</i> | <i>Validation Accuracy</i> |
|--------------|----------------------|--------------------------|------------------------|----------------------------|
| 1 | 7.7124 | 0.2839 | 7.1798 | 0.4509 |
| 2 | 6.9784 | 0.4248 | 6.5854 | 0.4777 |
| 3 | 6.6651 | 0.4597 | 6.3395 | 0.5670 |
| 4 | 6.4637 | 0.5064 | 6.0804 | 0.6116 |
| 5 | 6.3466 | 0.5169 | 5.9235 | 0.6071 |
| 6 | 6.1950 | 0.5551 | 5.8043 | 0.6696 |
| 7 | 6.0018 | 0.5678 | 5.5955 | 0.7143 |
| 8 | 5.8660 | 0.5932 | 5.4637 | 0.7054 |
| 9 | 5.8744 | 0.5763 | 5.6408 | 0.6429 |
| 10 | 5.7446 | 0.6070 | 5.4680 | 0.6830 |
| 11 | 5.6018 | 0.6345 | 5.2633 | 0.6875 |
| 12 | 5.4706 | 0.6483 | 5.1742 | 0.7277 |
| 13 | 5.4279 | 0.6462 | 5.1336 | 0.7143 |
| 14 | 5.4103 | 0.6398 | 5.0998 | 0.7500 |
| 15 | 5.2156 | 0.6557 | 4.9925 | 0.7768 |
| 16 | 5.2582 | 0.6320 | 4.8063 | 0.7723 |
| 17 | 5.1467 | 0.6684 | 4.7996 | 0.7679 |
| 18 | 5.0162 | 0.6663 | 4.6010 | 0.7991 |
| 19 | 4.9317 | 0.6896 | 4.6825 | 0.7679 |
| 20 | 4.8547 | 0.6864 | 4.5837 | 0.7634 |
| 21 | 4.8380 | 0.6910 | 4.6079 | 0.7500 |
| 22 | 4.8157 | 0.6737 | 4.4902 | 0.7545 |
| 23 | 4.5929 | 0.7267 | 4.2434 | 0.8482 |
| 24 | 4.5700 | 0.6939 | 4.3338 | 0.7946 |
| 25 | 4.4767 | 0.7267 | 4.2976 | 0.7946 |
| 26 | 4.4063 | 0.7076 | 4.0801 | 0.8438 |

Tabel 2. Hasil Proses Train Model Fine Tuning Model VGG-19 (Lanjutan 1)

| <i>Epoch</i> | <i>Training Loss</i> | <i>Training Accuracy</i> | <i>Validation Loss</i> | <i>Validation Accuracy</i> |
|--------------|----------------------|--------------------------|------------------------|----------------------------|
| 27 | 4.4668 | 0.6981 | 4.0050 | 0.8348 |
| 28 | 4.3171 | 0.7278 | 4.0924 | 0.7902 |
| 29 | 4.3458 | 0.7267 | 4.0599 | 0.7768 |
| 30 | 4.1236 | 0.7383 | 4.0248 | 0.7902 |

3.2. Hasil Training Model DenseNet-121 Setelah Modifikasi

Tabel 3. Hasil Proses Train Model Fine Tuning Model DenseNet-121

| <i>Epoch</i> | <i>Training Loss</i> | <i>Training Accuracy</i> | <i>Validation Loss</i> | <i>Validation Accuracy</i> |
|--------------|----------------------|--------------------------|------------------------|----------------------------|
| 1 | 6.1790 | 0.5169 | 5.6345 | 0.6295 |
| 2 | 5.4861 | 0.6197 | 4.8649 | 0.7366 |
| 3 | 5.0079 | 0.7013 | 4.7101 | 0.7232 |
| 4 | 4.7330 | 0.7066 | 4.4406 | 0.8036 |
| 5 | 4.5207 | 0.7161 | 4.2364 | 0.8259 |
| 6 | 4.3510 | 0.7479 | 4.0581 | 0.8393 |
| 7 | 4.0974 | 0.7786 | 3.8372 | 0.8571 |
| 8 | 3.8979 | 0.8231 | 3.6313 | 0.8661 |
| 9 | 3.8013 | 0.8051 | 3.5556 | 0.8616 |
| 10 | 3.6843 | 0.8072 | 3.3978 | 0.8884 |
| 11 | 3.5026 | 0.8358 | 3.3787 | 0.8438 |
| 12 | 3.4072 | 0.8347 | 3.1504 | 0.9107 |
| 13 | 3.2165 | 0.8475 | 3.2437 | 0.8616 |
| 14 | 3.1099 | 0.8538 | 2.9130 | 0.9196 |
| 15 | 3.0169 | 0.8697 | 2.8715 | 0.9062 |
| 16 | 2.9349 | 0.8623 | 2.7456 | 0.9152 |
| 17 | 2.7991 | 0.8856 | 2.6154 | 0.9107 |
| 18 | 2.6797 | 0.9036 | 2.4903 | 0.9509 |
| 19 | 2.6200 | 0.8867 | 2.4711 | 0.9152 |
| 20 | 2.5904 | 0.8739 | 2.4587 | 0.8973 |
| 21 | 2.5092 | 0.8845 | 2.2601 | 0.9286 |
| 22 | 2.3343 | 0.9216 | 2.2638 | 0.9062 |
| 23 | 2.3232 | 0.9078 | 2.1671 | 0.9196 |
| 24 | 2.2112 | 0.9036 | 2.0263 | 0.9554 |

Tabel 4. Hasil Proses Train Model Fine Tuning Model DenseNet-121

| <i>Epoch</i> | <i>Training Loss</i> | <i>Training Accuracy</i> | <i>Validation Loss</i> | <i>Validation Accuracy</i> |
|--------------|----------------------|--------------------------|------------------------|----------------------------|
| 25 | 2.1562 | 0.9057 | 2.0722 | 0.9241 |
| 26 | 2.1461 | 0.8919 | 1.9580 | 0.9375 |
| 27 | 2.0254 | 0.9089 | 1.8839 | 0.9464 |
| 28 | 1.9517 | 0.9163 | 1.8411 | 0.9330 |
| 29 | 1.8400 | 0.9292 | 1.7359 | 0.9643 |
| 30 | 1.8078 | 0.9301 | 1.6477 | 0.9777 |

3.3. Hasil Evaluasi Model Sebelum dan Setelah Modifikasi

Pada **Tabel 5.** menunjukkan hasil pengujian untuk model terbaik dari setiap arsitektur sebelum dilakukan modifikasi.

Tabel 5. Hasil Pengujian Model Sebelum Modifikasi

| Model | <i>Epoch</i> | <i>Learning Rate</i> | <i>Validation Accuracy</i> | <i>Validation Loss</i> | <i>F1-Score</i> |
|--------------|--------------|----------------------|----------------------------|------------------------|-----------------|
| VGG-19 | 6 | 0.0001 | 13.84% | 2.1975 | 7.39% |
| DenseNet-121 | 16 | 0.0001 | 50.45% | 2.9534 | 44.83% |

Berdasarkan pengujian sebelum modifikasi, model VGG-19 menunjukkan performa optimal pada *epoch* ke-6 dengan akurasi validasi sebesar 13.84% dan F1-score 7.39%. Model DenseNet-121 mencapai performa terbaik pada *epoch* ke-16 dengan akurasi validasi 50.45% dan F1-score 44.83%.

Pada **Tabel 6.** menunjukkan hasil pengujian untuk model terbaik dari setiap arsitektur setelah dilakukan modifikasi.

Tabel 6. Hasil Pengujian Model Setelah Modifikasi

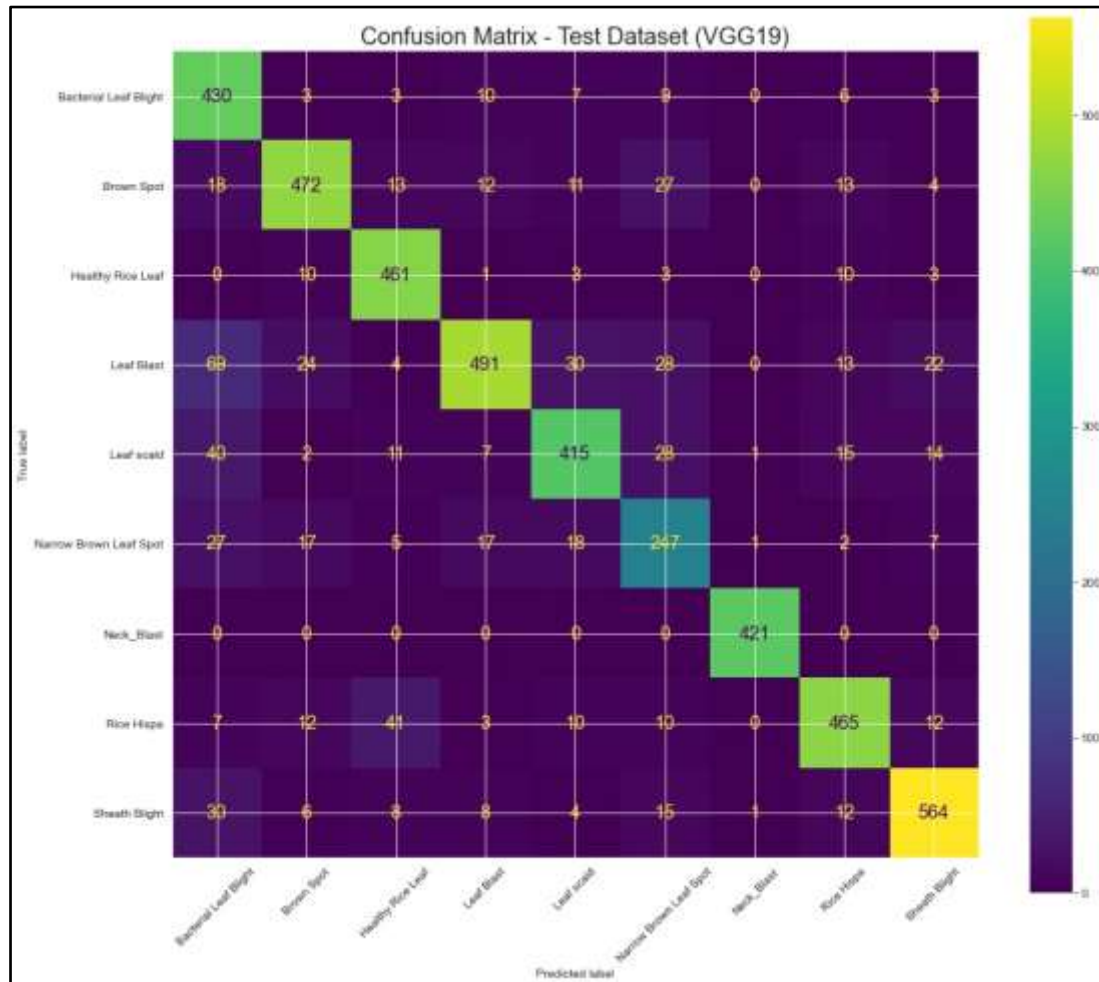
| Model | <i>Epoch</i> | <i>Learning Rate</i> | <i>Validation Accuracy</i> | <i>Validation Loss</i> | <i>F1-Score</i> |
|--------------|--------------|----------------------|----------------------------|------------------------|-----------------|
| VGG-19 | 23 | 0.0001 | 84.82% | 4.2434 | 87.52% |
| DenseNet-121 | 30 | 0.0001 | 97.76% | 1.6477 | 96.31% |

Berdasarkan pengujian setelah modifikasi, model VGG-19 menunjukkan performa optimal pada *epoch* ke-23 dengan akurasi validasi sebesar 84.82% dan F1-score 87.52%. Model DenseNet-121 mencapai performa terbaik pada *epoch* ke-30 dengan akurasi validasi 97.76% dan F1-score 96.31%. Dari hasil ini, dapat disimpulkan bahwa DenseNet-121 lebih unggul dibandingkan VGG-19 dalam klasifikasi penyakit tanaman padi. Pada pengujian dengan data validasi, model DenseNet-121 akurasi 97.76% dengan F1-score 96.31%, sedangkan VGG-19 mencapai akurasi

84.82% dengan F1-score 87.52%. Berdasarkan hasil ini, DenseNet-121 menunjukkan performa yang lebih baik dalam klasifikasi penyakit tanaman padi.

Berdasarkan hasil penelitian, dengan akurasi yang meningkat dan F1-score yang tinggi menunjukkan bahwa modifikasi pada model VGG-19 dan DenseNet-121 berhasil mengatasi masalah dataset kecil dan mengurangi resiko *overfitting*.

Confusion matrix dari pengujian model optimal pada data test disajikan pada Gambar 2 dan Gambar 3



Gambar 2. Confusion matrix VGG-19

Pada Gambar 2. *confusion matrix* diperoleh menggunakan data test yaitu data baru yang tidak melalui proses train. Hasil *confusion matrix* pada dataset uji menunjukkan bahwa model VGG-19 mampu mengklasifikasikan sebagian besar kelas dengan benar, yang terlihat dari dominasi nilai pada diagonal utama. Kelas *Leaf Blast* dan *Sheath Blight* memiliki jumlah prediksi benar tertinggi, masing-masing sebanyak 491 dan 564, menunjukkan bahwa model sangat andal dalam mengidentifikasi kedua kelas ini. Namun, beberapa kesalahan klasifikasi masih terjadi, seperti kelas *Brown Spot* yang sering salah diprediksi sebagai *Narrow Brown Leaf Spot* (27 kesalahan). Kesalahan model dalam memprediksi *Brown Spot* sebagai *Narrow Brown Leaf Spot* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan warna dominan bercak (coklat

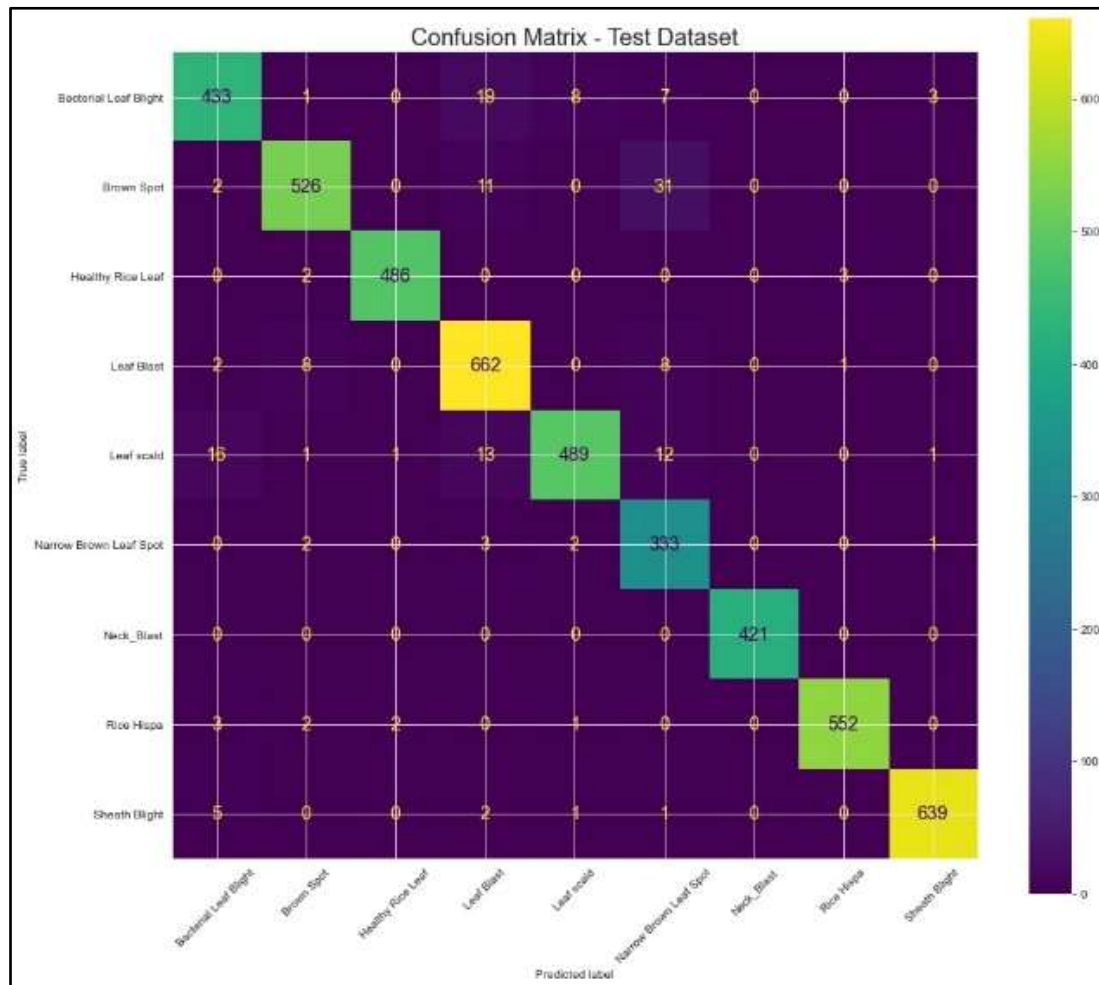
tua) yang menjadi ciri utama kedua penyakit. Selain itu, *Narrow Brown Leaf Spot* memiliki bercak kecil yang tersebar merata di seluruh helai daun, dan pola ini dapat terlihat seperti bercak-bercak yang sering juga muncul pada *Brown Spot*. Ukuran bercak pada *Brown Spot* lebih besar daripada *Narrow Brown Leaf Spot*, namun jika dalam gambar ukuran bercak tidak ditangkap dengan baik karena jarak pengambilan gambar atau resolusi gambar maka ini bisa menjadi sulit dikenali oleh model.

Selanjutnya yaitu *Leaf Scald* sering salah diprediksi sebagai *Bacterial Leaf Blight* (40 kesalahan). Kesalahan model dalam memprediksi *Leaf Scald* sebagai *Bacterial Leaf Blight* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan yaitu warna daun yang berubah menjadi kelabu, abu-abu, atau kuning pucat, Lesi atau kerusakan sering dimulai dari ujung daun, dan daun tampak seperti terbakar dan kering. Selain itu, *Leaf Blast* yang sering salah diprediksi sebagai *Bacterial Leaf Blight* (69 kesalahan). Kesalahan model dalam memprediksi *Leaf Blast* sebagai *Bacterial Leaf Blight* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan yaitu kedua penyakit menunjukkan warna bercak yang serupa, yaitu kelabu atau abu-abu dan kuning pucat. Pada fase infeksi lanjut, *Leaf Blast* dan *Bacterial Leaf Blight* sama-sama menyebabkan kerusakan yang terlihat seperti pembakaran atau pengeringan daun.

Selain penyakit tersebut, *Rice Hispa* juga sering salah diprediksi sebagai *Healthy Rice Leaf* (41 kesalahan). Kesalahan model dalam memprediksi *Rice Hispa* sebagai *Healthy Rice Leaf* bisa disebabkan karena kedua penyakit tersebut sama-sama memiliki warna dasar hijau. Meski *Rice Hispa* memiliki bercak putih kecil akibat kikisan pada permukaan daun, warna hijau dominan masih terlihat, sehingga membuat model sulit membedakan antara daun sehat dan daun yang terinfeksi. Selain itu, kerusakan akibat *Rice Hispa* cenderung berupa kikisan tipis yang menghasilkan bercak putih kecil. Pola kerusakan ini tidak terlalu kontras terhadap latar hijau daun, terutama jika gambar memiliki resolusi rendah atau pencahayaan yang kurang ideal. Hal ini membuat kerusakan tersebut terlihat seperti tekstur alami daun. Bercak putih akibat kikisan sering kali terlihat mirip dengan pantulan cahaya pada daun sehat dalam kondisi pencahayaan tertentu. Hal ini dapat membingungkan model dalam mengidentifikasi ciri khas kerusakan *Rice Hispa*. Tidak seperti penyakit lain yang menunjukkan pola bercak besar atau lesi yang jelas, kerusakan *Rice Hispa* lebih menyerupai garis-garis atau bercak kecil yang tersebar secara acak. Pola ini mungkin tidak cukup signifikan untuk dianggap sebagai kerusakan oleh model.

Penyakit *Sheath Blight* juga sering salah diprediksi sebagai *Bacterial Leaf Blight* (30 kesalahan). Kesalahan model dalam memprediksi *Sheath Blight* sebagai *Bacterial Leaf Blight* bisa disebabkan karena perubahan warna kuning hingga coklat pada *Sheath Blight* (terutama saat lesi meluas ke daun) sering kali menyerupai warna daun yang terinfeksi *Bacterial Leaf Blight*, terutama dalam kondisi pencahayaan tertentu pada gambar. Selain itu, Pola lesi memanjang yang dihasilkan oleh *Sheath Blight* dan *Bacterial Leaf Blight* dapat terlihat sangat mirip, terutama pada gambar yang menunjukkan tahap infeksi yang parah. Dalam kasus infeksi berat, kedua penyakit tersebut dapat menghasilkan daun yang kering dan melipat, sehingga membuat model sulit membedakan antara keduanya.

Selain kelas penyakit yang sudah disebutkan tersebut, walaupun beberapa kelas lain juga melakukan kesalahan dalam prediksi, tetapi jumlah kesalahan yang terjadi relatif kecil dibandingkan prediksi yang benar. Secara keseluruhan, hasil ini mencerminkan kemampuan model yang baik dalam mengklasifikasikan penyakit tanaman padi pada data uji, dengan tingkat kesalahan yang minimal dan tidak signifikan.



Gambar 3. Confusion matrix DenseNet-121

Pada Gambar 3. *confusion matrix* diperoleh menggunakan data test yaitu data baru yang tidak melalui proses train. Hasil *confusion matrix* pada dataset uji menunjukkan bahwa model DenseNet-121 mampu mengklasifikasikan sebagian besar kelas dengan benar, yang terlihat dari dominasi nilai pada diagonal utama. Kelas *Leaf Blast* dan *Sheath Blight* memiliki jumlah prediksi benar tertinggi, masing-masing sebanyak 662 dan 639, menunjukkan bahwa model sangat andal dalam mengidentifikasi kedua kelas ini. Namun, beberapa kesalahan klasifikasi masih terjadi, seperti kelas *Bacterial Leaf Blight* yang sering salah diprediksi sebagai *Leaf Blast* (19 kesalahan). Kesalahan model dalam memprediksi *Bacterial Leaf Blight* sebagai *Leaf Blast* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan yaitu kedua penyakit menunjukkan warna bercak yang serupa, yaitu kelabu atau abu-abu dan kuning pucat. Pada fase infeksi lanjut, *Leaf Blast* dan *Bacterial Leaf Blight* sama-sama menyebabkan kerusakan yang terlihat seperti pembakaran atau pengeringan daun.

Selanjutnya yaitu *Brown Spot* yang sering salah diprediksi sebagai *Narrow Brown Leaf Spot* (31 kesalahan). kesalahan model dalam memprediksi *Brown Spot* sebagai *Narrow Brown Leaf Spot* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan warna dominan bercak

(coklat tua) yang menjadi ciri utama kedua penyakit. Selain itu, *Narrow Brown Leaf Spot* memiliki bercak kecil yang tersebar merata di seluruh helai daun, dan pola ini dapat terlihat seperti bercak-bercak yang sering juga muncul pada *Brown Spot*. Ukuran bercak pada *Brown Spot* lebih besar daripada *Narrow Brown Leaf Spot*, namun jika dalam gambar ukuran bercak tidak ditangkap dengan baik karena jarak pengambilan gambar atau resolusi gambar maka ini bisa menjadi sulit dikenali oleh model. Selanjutnya, *Leaf Scald* yang sering salah diprediksi sebagai *Bacterial Leaf Blight* (16 kesalahan). Kesalahan model dalam memprediksi *Leaf Scald* sebagai *Bacterial Leaf Blight* bisa disebabkan karena kedua penyakit tersebut memiliki kemiripan yaitu warna daun yang berubah menjadi kelabu, abu-abu, atau kuning pucat, lesi atau kerusakan sering dimulai dari ujung daun, dan daun tampak seperti terbakar dan kering.

Selain penyakit-penyakit tersebut, penyakit yang lain memiliki jumlah prediksi salah relatif kecil dan tidak signifikan. Secara keseluruhan, hasil ini mencerminkan kemampuan model yang baik dalam mengklasifikasikan penyakit tanaman padi pada data uji dengan tingkat kesalahan yang minimal.

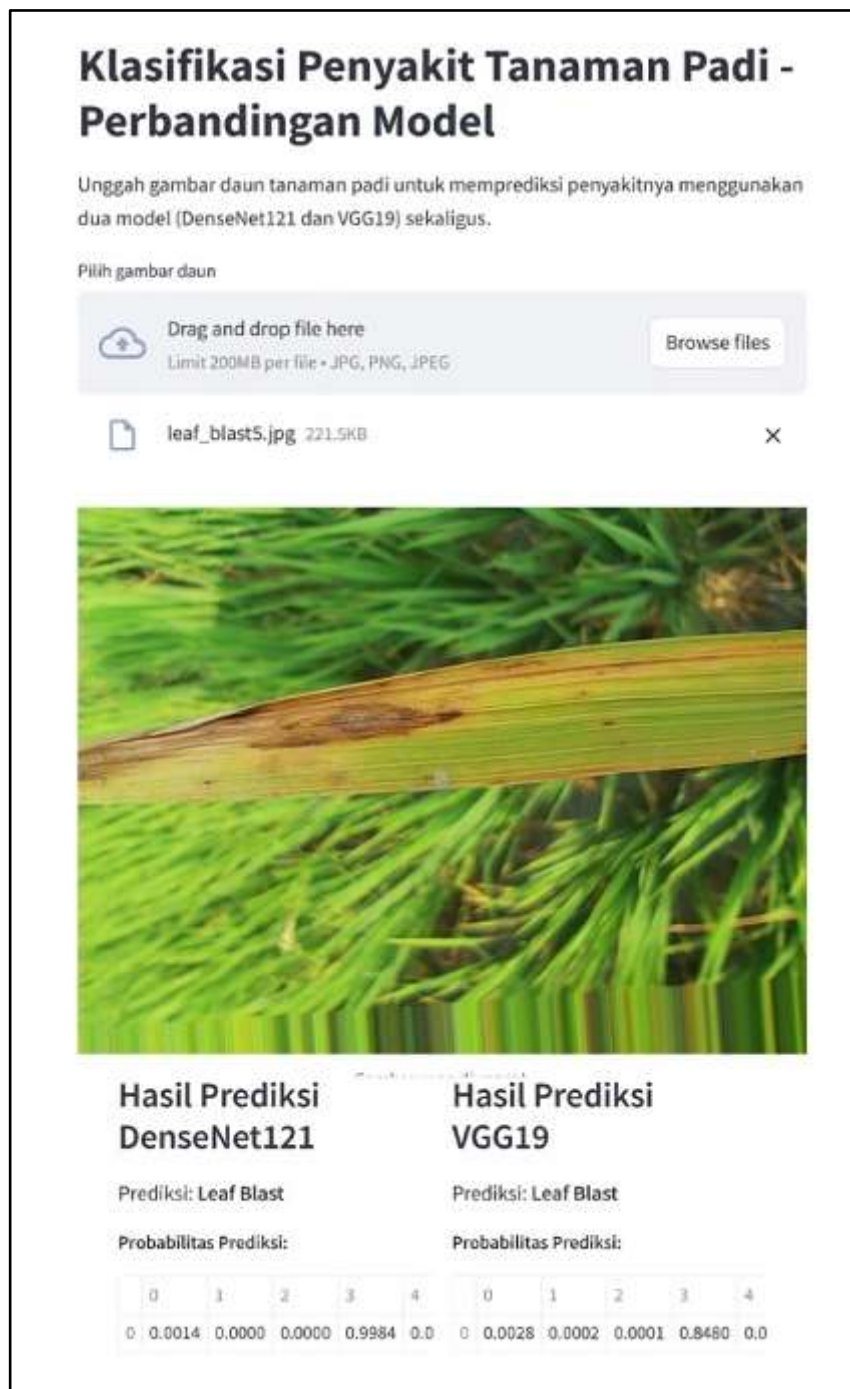
3.4. Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna dirancang menggunakan Streamlit, sebuah framework Python yang memungkinkan pembuatan aplikasi web dengan mudah. Aplikasi ini menerima input berupa gambar daun padi, melakukan prediksi menggunakan dua model (DenseNet-121 dan VGG-19), dan menampilkan hasil prediksi beserta probabilitasnya.

Antarmuka ini dirancang agar mudah digunakan, memungkinkan pengguna untuk membandingkan hasil prediksi dari dua model secara langsung. Dengan adanya visualisasi probabilitas, pengguna dapat memahami tingkat keyakinan model terhadap prediksi yang diberikan.

Saat *source code* dijalankan, perangkat lunak web untuk klasifikasi penyakit tanaman padi menampilkan antarmuka seperti pada Gambar 4. Antarmuka ini memungkinkan pengguna untuk mengunggah gambar daun tanaman padi, yang kemudian diproses oleh dua model (DenseNet-121 dan VGG-19) untuk menghasilkan prediksi. Hasil prediksi berupa nama kelas penyakit dan probabilitasnya ditampilkan dalam dua kolom secara berdampingan.

Gambar 4. menunjukkan tata letak antarmuka sistem yang dirancang menggunakan Streamlit. Area unggahan gambar terletak di bagian atas, untuk mengunggah gambar caranya dengan menekan tombol “browse files”, kemudian akan diarahkan ke gambar yang akan dipilih dari komputer. Setelah selesai di unggah maka gambar akan muncul dibawah, sementara hasil prediksi ditampilkan di bawahnya dalam dua kolom. Prediksi dari DenseNet-121 ditampilkan di kolom kiri, sedangkan prediksi dari VGG-19 ditampilkan di kolom kanan. Berikut ini adalah Gambar 4.



Gambar 4. Hasil klasifikasi Penyakit Tanaman Padi

4. Kesimpulan dan Saran

Penelitian ini menunjukkan bahwa modifikasi pada model VGG-19 dan DenseNet-121 berhasil mengatasi masalah dataset kecil dan mengurangi resiko *overfitting*. Hal ini ditunjukkan dari hasil akurasi yang meningkat setelah modifikasi. Sebelum modifikasi model VGG-19 hanya mencapai akurasi tertinggi sebesar 13.84% dan F1-score 7.39%, sedangkan model DenseNet-121

hanya mencapai akurasi tertinggi sebesar 50.45% dan F1-score 44.83%. Setelah modifikasi, kedua model menunjukkan peningkatan akurasi, dan setelah perbandingan, model DenseNet-121 menunjukkan performa yang lebih unggul dibandingkan dengan VGG-19 dalam klasifikasi penyakit tanaman padi. VGG-19 mencapai akurasi 84.82% dengan F1-score 87.52%, sedangkan DenseNet-121 mencapai akurasi validasi tertinggi sebesar 97.76% dan F1-score 96.31%, yang mencerminkan kemampuannya dalam menangkap pola visual yang kompleks melalui *dense connections*. Sebaliknya, model VGG-19 menghadapi tantangan dalam membedakan beberapa kelas penyakit, seperti *Leaf Scald* dan *Bacterial Leaf Blight*, akibat kemiripan karakteristik visual antar kelas dan distribusi data yang tidak merata. Parameter *fine-tuning* dan pelatihan selama 30 *epoch* pada DenseNet-121 menghasilkan model terbaik.

Berdasarkan hasil analisis yang telah disampaikan tentang perbandingan performa arsitektur VGG-19 dan DenseNet-121 terdapat banyak kekurangan. Oleh karena itu, beberapa saran yang dapat diterapkan untuk pengembangan penelitian selanjutnya adalah sebagai berikut :

1. Menggunakan model *deep learning* lain yang lebih kompleks dengan dataset kecil untuk dimodifikasi, disertai dokumentasi percobaan yang lengkap mengenai variasi angka pada *layer* tambahan yang diterapkan, seperti *dropout*.
2. Penelitian lebih lanjut dapat mempertimbangkan pengembangan model *hybrid* yang menggabungkan keunggulan VGG-19 dalam menyederhanakan arsitektur dengan kekuatan DenseNet-121 dalam menangkap fitur representatif.
3. Mengimplementasikan model DenseNet-121 dalam aplikasi mobile sehingga memudahkan petani untuk mengakses diagnosis penyakit secara langsung di lapangan.

Daftar Pustaka

- [1] Agustiani, S., Tajul Arifin, Y., Junaidi, A., Khotimatul Wildah, S., & Mustopa, A. (2022). Klasifikasi Penyakit Daun Padi menggunakan Random Forest dan Color Histogram. *Jurnal Komputasi*, 10(1). <https://doi.org/10.23960/komputasi.v10i1.2961>
- [2] Dwi Puspa, I., Wicaksono, A., Tri Samiha, Y., Falahudin, I., Putri Anggun, D., & Oktiansyah, R. (2018). Serangga hama sebagai prganisme pengganggu tanaman (opt) terhadap produktivitas padi (Oryza sativa L.). *UINRadenFatahPalembang*, 1–6. <http://proceedings.radenfatah.ac.id/index.php/semnaspbio>
- [3] Sharma, N.; Gupta, S.; Koundal, D.; Alyami, S.; Alshahrani, H.; Asiri, Y.; Shaikh, A. (2023). U-Net Model with Transfer Learning Model as a Backbone for Segmentation of Gastrointestinal Tract. *Bioengineering*, 10(1), 119. <https://doi.org/10.3390/bioengineering10010119>.
- [4] Zheng, Y., Yang, C. K., & Merkulov, A. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. *Proceedings of SPIE--The International Society for Optical Engineering*, 10574, 105740G. <https://doi.org/10.1117/12.2304564>.
- [5] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>.

- [6] Khobragade, V., Nirmal, J., & Chedda, S. (2022). Revaluating pretraining in small size training sample regime. *International Journal of Electrical and Electronics Research*, 10(3), 694–704. <https://doi.org/10.37391/IJEER.100346>
- [7] Liu, S., & Deng, W. (2015). *Very deep convolutional neural network based image classification using small training sample size*. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). <https://doi.org/10.1109/ACPR.2015.7486599>