



Klasifikasi Citra Penyakit Daun Tanaman Padi Menggunakan CNN dengan Arsitektur VGG-19

Rahma Shinta^a, Jasril^b, Muhammad Irsyad^c, Febi Yanto^d, Suwanto Sanjaya^e

^aTeknik Informatika, Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, 11950125168@students.uin-suska.ac.id

^bTeknik Informatika, Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, jasril@uin-suska.ac.id

^cTeknik Informatika, Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, irsyadtech@uin-suska.ac.id

^dTeknik Informatika, Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, febiyanto@uin-suska.ac.id

^eTeknik Informatika, Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, suwantosanjaya@uin-suska.ac.id

Submitted: 05-04-2023, Reviewed: 03-05-2023, Accepted 04-05-2023

<http://doi.org/10.22216/jsi.v9i1.2175>

Abstract

The decline in rice production is caused by pests and diseases that are common on the leaves. Many studies related to the classification of types of rice leaf disease have been carried out. This study applies the Convolutional Neural Network (CNN) method with VGG-19 architecture for image classification of rice leaf disease. The purpose of this study was to compare the results of testing accuracy from models that use data augmentation and without augmentation. The data in this study were divided into 4 classes, namely blast, brown spot, leaf smut, and healthy with a total of 440 original data and 1320 augmentation data. The test results showed that the highest accuracy using data augmentation was 94.31%, while the highest accuracy without data augmentation was 93.18%. The results of the research show that augmentation can increase the accuracy of results. The use of the Nadam optimizer produces a higher accuracy value than Adamax. The Hyper Parameters used also affect the accuracy of the test results.

Keywords: Adamax Optimizer, CNN VGG-19, Data Augmentation, Nadam Optimizer, Rice Leaf Classification

Abstrak

Penurunan produksi padi disebabkan oleh serangan hama dan penyakit yang biasa terdapat pada bagian daun. Penelitian terkait klasifikasi jenis penyakit daun padi telah banyak dilakukan. Penelitian ini menerapkan metode Convolutional Neural Network (CNN) dengan arsitektur VGG-19 untuk klasifikasi citra penyakit daun tanaman padi. Tujuan penelitian ini adalah untuk membandingkan hasil akurasi pengujian dari model yang menggunakan augmentasi dan tanpa augmentasi data. Data pada penelitian ini terbagi atas 4 kelas, yaitu blast, brown spot, leaf smut, dan healthy dengan jumlah data asli sebanyak 440 dan data augmentasi sebanyak 1320 citra. Hasil pengujian menunjukkan bahwa akurasi tertinggi menggunakan augmentasi data yang diperoleh sebesar 94.31%, sedangkan akurasi tertinggi tanpa augmentasi data yang diperoleh sebesar 93.18%. Hasil penelitian menunjukkan bahwa augmentasi dapat meningkatkan hasil akurasi. Penggunaan *optimizer Nadam* menghasilkan nilai akurasi yang lebih tinggi dibandingkan *Adamax*. *Hyper Parameter* yang digunakan juga berpengaruh terhadap hasil akurasi pengujian.

Kata kunci: Augmentasi Data, Klasifikasi Daun Padi,, Optimizer Adamax, Optimizer Nadam, CNN VGG-19

© 2023 Jurnal Sains dan Informatika

1. Pendahuluan

Padi memegang peranan penting dalam kontribusi ekonomi di Indonesia di bidang sektor pertanian. Beras yang dihasilkan oleh padi merupakan salah satu sumber pangan yang menjadi makanan pokok dengan konsumsi terbesar di Indonesia. Tingkat keberhasilan panen padi sangat berpengaruh karena kegagalan produksi padi dapat menyebabkan terjadi gejolak ekonomi dan politik. Berdasarkan hasil survey Kerangka Sampel Area (KSA) produksi padi menurun pada tahun 2021. Luas panen padi menurun sebanyak 245.47 hektar atau

2.30 persen dibandingkan tahun 2020. Jumlah produksi padi juga menurun yang apabila dikonversikan menjadi produksi beras terjadi penurunan sebanyak 140,73 ton atau 0,45% dibandingkan tahun 2020 [1].

Produksi padi menurun karena hama atau penyakit yang menyerang tanaman padi. Pada tahun 2022, padi di Kabupaten Madiun, Jawa Timur menurun karena ada serangan Organisme Pengganggu Tanaman (OPT) [2]. Penyakit padi timbul disebabkan oleh bakteri dan jamur yang menyerang padi, terutama di bagian daun[3]. Masyarakat awam terkadang kesulitan dalam

membedakan jenis penyakit yang menyerang daun tanaman padi [4]. Gejala penyakit padi yang sudah lanjut akan sulit untuk dibedakan [5]. Penyakit pada daun tanaman padi sulit dibedakan menggunakan penglihatan manual serta memiliki banyak kelemahan. Hal tersebut disebabkan karena penyakit yang menyerang daun tanaman padi memiliki bentuk yang hampir sama. Penurunan hasil produksi padi dapat dicegah apabila teknik pengendalian penyakitnya dilakukan dengan tepat [3]. Setiap penyakit memiliki cara pengendalian yang berbeda-beda. Oleh karena itu dibutuhkan teknologi yang memanfaatkan teknik pengolahan citra untuk membedakan penyakit pada daun tanaman padi.

Salah satu teknik pengolahan citra yaitu *machine learning*. Machine learning merupakan suatu metode komputasi yang menggabungkan statistik dan ilmu komputer untuk menghasilkan suatu algoritma yang dapat mengidentifikasi data *inputan* [6]. Selain *machine learning* terdapat teknik pengolahan citra lain yaitu *deep learning* yang lebih baik dibandingkan *machine learning*. Pada penelitian menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG-19 untuk klasifikasi citra penyakit daun padi. Penelitian ini menerapkan augmentasi data karena keterbatasan jumlah data yang diperoleh. Augmentasi data yang digunakan yaitu *height shift*, *width shift*, dan *shear range*. Penelitian ini juga menggunakan *Hyper Parameter Optimization* yang bertujuan untuk mengoptimalkan kinerja algoritma yang digunakan.

Tujuan penelitian ini yaitu untuk membandingkan tinggi tingkat akurasi menggunakan CNN arsitektur VGG-19 serta *hyperparameter* untuk klasifikasi penyakit pada daun tanaman padi dengan augmentasi data atau tanpa augmentasi data.

2. Tinjauan Pustaka/ Penelitian Sebelumnya

Salah satu teknik pengolahan citra yaitu *machine learning*. Beberapa penelitian terkait membedakan penyakit daun tanaman padi yang memanfaatkan pengolahan citra telah dilakukan diantaranya klasifikasi menggunakan *K-Nearest Neighbor* (KNN) [7]. Raj Kumar dan kawan kawan juga melakukan penelitian yang menggunakan metode *Support Vector Machine* (SVM), *Naïve Bayes*, *Logistic Regressin*, *Random Forest*, *Decission Tree*, *Linear Discriminant Analysis* (LDA), dan *Principal Component Analysis* (PCA) [8]. Serta klasifikasi peyakit daun pada padi dengan implementasi Gray Level Co-Occurance Matrix (GLCM) [9]. Penelitian tersebut dilakukan oleh Arif Akbar Huda, Bayu Setiaji, dan Fajar Rosyid Hidayat pada tahun 2022. Data citra yang digunakan pada penelitian tersebut sebanyak 240 citra.

Selain *machine leraning* juga terdapat metode lain yaitu *deep learning* yang lebih baik dalam pengenalan citra dibandingkan *machine learning*. Penelitian

tentang perbandingan kedua metode tersebut telah dilakukan oleh Yudicy Amelia dan kawan-kawan (2018) dengan ujicoba pada data penyakit kanker payudara [10]. Penelitian tersebut menunjukkan bahwa hasil akurasi *deep learning* lebih tinggi daripada *machine learning*. Hasil penelitian Alex Sander Simbolon dan kawan kawan menunjukkan bahwa hasil akurasi terbaik didapatkan oleh algoritma *Convolutional Neural Network* (CNN) dengan *f-measure* sebesar 86% [11]. Algoritma CNN juga berhasil mengungguli SVM dan KNN dengan akurasi sebesar 0.942 berdasarkan penelitian Mohammad Farid Naufal tahun 2021 [12]. Penelitian-penelitian tersebut menunjukkan bahwa pengolahan citra menggunakan *deep learning* menghasilkan tingkat akurasi yang lebih tinggi dibandingkan *machine learning*.

Pada CNN terdapat berbagai model arsitektur diantaranya *EfficienNet*, *MobileNeT*, *AlexNet*, *VGGNet*, *ResNet*, *NasNet*, dan arsitektur lainnya. Shivam, Surya Pratap Singh (2021) melakukan penelitian dengan judul "*Rice Infection Recognition using Deep Neural Network Systems*". Hasil penelitian tersebut menunjukkan bahwa Arsitektur VGG-19 berhasil mengungguli arsitektur *LeNet5* dan *MobileNet-V2* dengan akurasi sebesar 77,09% [13]. Djarot Hindarto dan Handri Santoso juga telah melakukan penelitian untuk membandingkan beberapa algoritma *deep learning* pada tahun 2021. Hasil penelitan menunjukkan bahwa VGG-16 dan VGG-19 menghasilkan akurasi yang lebih bagus dibandingkan *DenseNet121* dan *NasNetLarge*. VGG-19 juga berhasil mengungguli *DenseNet121*, *NasNetLarge*, dan VGG-16 dengan akurasi sebesar 85,05% [14]. Penelitian-penelitian tersebut menunjukkan bahwa tingkat akurasi yang dihasilkan oleh VGG-19 berhasil mengungguli beberapa algoritma *deep learning* lainnya. Pada penelitian ini akan menerapkan algoritma VGG-19 untuk membangun model klasifikasi.

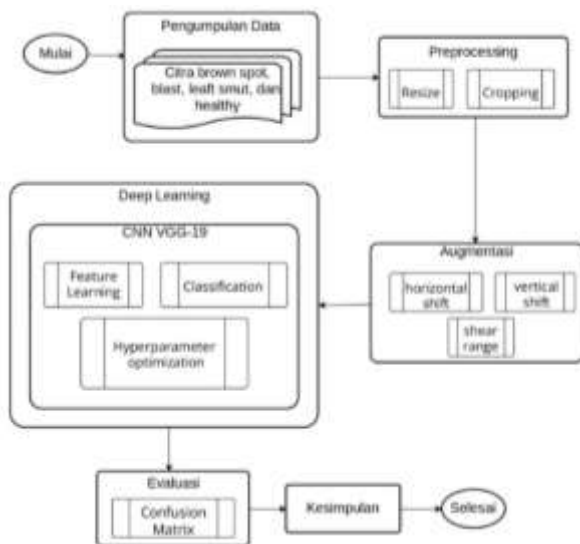
Penelitian yang dilakukan Shivam [13] memberikan saran penelitian untuk menerapkan teknik *preprocessing image* sebelum dilakukan *training* untuk mendapatkan nilai akurasi yang lebih baik. Augmentasi merupakan suatu cara yang dilakukan untuk memperbanyak jumlah citra yang tujuannya meningkatkan akurasi penelitian yang dilakukan. Penelitian augmentasi telah dilakukan oleh Victor dan kawan kawan pada tahun 2021. Penelitian tersebut menggunakan augmentasi berupa *horizontal flip*, *horizontal shift*, *vertical shift*, *shear range*, dan *zoom range*. Hasil penelitian menunjukkan bahwa akurasi tertinggi yang diperoleh sebesar 93,6% [15]. Augmentasi juga telah digunakan dalam meneliti penyakit padi pada tahun 2022 [16]. Jenis augmentasi yang digunakan pada penelitian tersebut yaitu *translation*, *arbitrary rotation*, *shearing*, *scaling*, dan *flipping*. Hasil akurasi sebesar 92,61% juga diperoleh dengan penggunaan augmentasi pada penelitian. Penelitian yang menerapkan augmentasi *horizontal flip*,

width shift, rotation, zoom range, shear range, dan fill mode menunjukkan hasil akurasi pelatihan terbaik sebesar 96,37% [17]. Augmentasi juga digunakan pada penelitian lain yaitu *shift, shear range, rescale, rotation range, dan fill mode* [18]. Hasil penelitian tersebut menunjukkan bahwa akurasi model yang menggunakan augmentasi data lebih tinggi dibandingkan tidak menggunakan augmentasi data. Pada penelitian ini akan menggunakan augmentasi data dan tanpa augmentasi data untuk membandingkan hasil tingkat akurasi yang didapatkan.

Pada penelitian ini akan menggunakan metode CNN arsitektur VGG-19 dengan augmentasi berupa *height shift, width shift, dan shear range* serta kombinasi *Hyper Parameter Optimization*. Penggunaan *Hyper Parameter Optimization* bertujuan untuk mengoptimalkan kinerja dari suatu algoritma. Penelitian terkait penggunaan *hyperparameter optimization* telah dilakukan oleh Jia Wu dan kawan kawan pada tahun 2019. Hasil penelitian menunjukkan penggunaan *hyperparameter optimization* dapat mencapai akurasi yang tinggi dalam beberapa sampel [19]. Parameter yang akan digunakan yaitu *batch size, dense, learning rate*, serta penggunaan optimizer Adamax dan Nadam untuk meningkatkan hasil akurasi.

3. Metodologi Penelitian

Gambar 1 merupakan tahapan penelitian yang akan dilakukan untuk klasifikasi citra penyakit daun tanaman padi pada penelitian ini.



Gambar 1. Tahapan Penelitian

Pada penelitian ini terdapat dua proses klasifikasi yaitu proses yang menggunakan augmentasi data dan tanpa augmentasi data. Pada proses yang menggunakan data asli, tahapan augmentasi tidak dilakukan. Setelah tahap *preprocessing* maka lanjut ke tahap *deep learning*. Pada proses yang menggunakan data augmentasi, tahap

augmentasi dilakukan sesuai dengan tahapan penelitian pada gambar 1.

3.1 Pengumpulan Data

Data yang dikumpulkan yaitu citra penyakit daun tanaman padi yang terdiri atas empat kelas yaitu *brown spot, blast, leaf smut, dan healthy*. Data tersebut berupa data sekunder yang didapatkan dari *website* penyedia dataset yaitu *Kaggle* [20], [21] dan *UCI Machine Learning* [22]. Data citra yang digunakan sebanyak 440 citra.

3.2 Preprocessing

Setelah dilakukan tahapan pengumpulan data maka tahapan selanjutnya yaitu tahap *preprocessing*. Tujuan tahap ini untuk menghasilkan citra yang dapat diproses dengan mudah. Jenis *Preprocessing* yang dilakukan yaitu *cropping* dan *resize*.

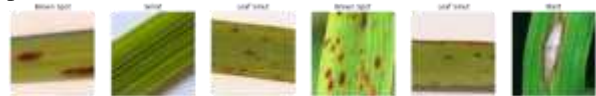
a. Cropping

Cropping bertujuan untuk menghilangkan sudut citra yang tidak menggambarkan pola dari citra. Tahap ini menggunakan kodingan *python* untuk menjalankannya. Citra yang terlalu jauh pengambilan gambarnya dan tidak terletak dibagian tengah akan di-*crop* secara manual terlebih dahulu sehingga pola penyakit pada citra tidak hilang. Kemudian keseluruhan citra di-*crop* menggunakan *python*.

b. Resize

Resize bertujuan untuk menyamakan ukuran citra sehingga memudahkan dan mempercepat proses perhitungan dan pengenalan citra [23]. Ukuran citra diperkecil menjadi 224 x 224 piksel. Ukuran tersebut sesuai dengan ukuran input dari algoritma VGG-19. Tahap *resize* menggunakan kodingan *python* untuk menjalankannya.

Berikut ini merupakan sampel data citra asli pada penelitian.



Gambar 2. Sampel Citra Asli

3.3 Augmentasi

Tahap augmentasi tidak dilakukan untuk proses klasifikasi yang menggunakan data asli. Augmentasi merupakan suatu cara yang dilakukan untuk memperbanyak jumlah citra karena keterbatasan jumlah citra. Tujuan augmentasi untuk meningkatkan akurasi penelitian yang dilakukan. Proses augmentasi dilakukan dengan memanfaatkan fungsi *Image Data Generator* dari *Tensorflow*.

Adapun teknik augmentasi yang digunakan sebagai berikut:

- a. *Width Shift*, merupakan pergeseran lebar pada gambar. Pergeseran tersebut bergeser sepanjang garis vertikal pada gambar. Pada penelitian ini menggunakan augmentasi *width_shift_range = 0.2*. Sehingga gambar akan digeser secara vertikal sepanjang 20%.
- b. *Height Shift*, yaitu pergeseran panjang pada gambar. Pergeseran tersebut bergeser sepanjang garis horizontal pada gambar. Pada penelitian ini menggunakan augmentasi *height_shift_range = 0.2*. Sehingga gambar akan digeser secara vertikal sepanjang 20%.
- c. *Shear range*, yaitu proses pemiringan pada lebar dan tinggi pada gambar. *Shear range* mengakibatkan terjadinya perubahan gambar menjadi bentuk jajar genjang. Pada penelitian ini menggunakan *augmentasi shehar_range = 0.3*.

Berikut ini adalah sampel data citra augmentasi pada penelitian.



Gambar 3. Sampel Citra Augmentasi

3.4 Deep Learning

Setelah tahap augmentasi maka selanjutnya adalah tahap pembagian data dari 440 citra asli dan 1320 citra augmentasi. Data tersebut dibagi menjadi data latih (*train*) dan data uji (*test*). Perbandingan yang digunakan untuk data *train* dan *testing* yaitu 70:30, 80:20, dan 90:10.

Tabel 1. Pembagian Data Citra

Data	Jenis Citra	Jenis Data	Pembagian Data		
			90:10	80:20	70:30
Blast	Asli	Latih	99	88	77
		Uji	11	22	33
	Augmentasi	Latih	297	264	231
		Uji	33	66	99
Brownspot	Asli	Latih	99	88	77
		Uji	11	22	33
	Augmentasi	Latih	297	264	231
		Uji	33	66	99
Leafsmut	Asli	Latih	99	88	77
		Uji	11	22	33
	Augmentasi	Latih	297	264	231
		Uji	33	66	99
Healthy	Asli	Latih	99	88	77
		Uji	11	22	33
	Augmentasi	Latih	297	264	231
		Uji	33	66	99

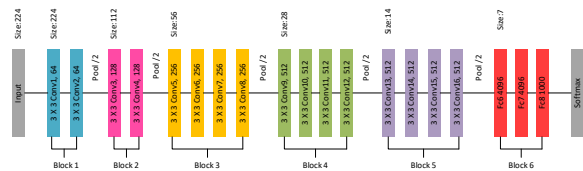
a. Convolutional Neural Network (CNN)

CNN adalah algoritma yang bekerja dengan cara menerima masukan berupa gambar. CNN merupakan metode *machine learning* yang dikembangkan dari *MultiLayer Perceptron (MLP)* yang dirancang dapat mengolah data 2D [24]. CNN tergolong *Deep Neural Network* karena banyak digunakan pada data citra dan memiliki kedalaman jaringan yang tinggi [23]. Cara kerja CNN mirip dengan jaringan syaraf, tetapi perbedaannya adalah bahwa setiap lapisan pada CNN memiliki dua dimensi yang kemudian dikonvolusi dengan masukan dari lapisan tersebut.[25]. Lapisan CNN terdiri atas *convolutional layer*, *pooling layer*, dan *fully connected layer*.

b. VGG-19

VGG (Visual Geometry Group-19) merupakan perkembangan dari arsitektur VGG-16 yang memiliki 19 lapisan mendalam sehingga kinerja yang dihasilkan menjadi lebih baik. VGG-19 merupakan arsitektur CNN yang terdiri atas 16 lapisan *convolutional layer* beserta aktivasi ReLU dan 3 lapisan *fully connected layer* beserta aktivasi ReLU. Pada setiap lapisan terakhir *convolutional* terdapat lapisan *max pooling* untuk mengurangi ukuran spasial dari *convolutional layer* sebelumnya. Lapisan input menggunakan ukuran 224 x 224 piksel, dan lapisan output diganti dengan fungsi aktivasi *softmax* yang menunjukkan hasil klasifikasi.

Berikut ini merupakan arsitektur VGG-19 yang digunakan pada penelitian ini



Gambar 4. Arsitektur VGG-19

c. Hyper Parameter Optimization

Pada tahap ini bertujuan untuk menentukan kombinasi parameter yang paling bagus untuk mendapatkan hasil akurasi maksimum dari penelitian. Kombinasi *Hyper Parameter Optimization* yang akan digunakan berupa *batch size*, *dense*, *learning rate*, serta *optimizer Adamax* dan *Nadam*.

3.4 Deep Learning

Setelah eksperimen selesai dilakukan dengan berbagai skenario pengujian, maka hasil eksperimen tersebut kemudian dilakukan tahap evaluasi. Tahap ini bertujuan untuk mengukur performa model yang telah dihasilkan dari tahap sebelumnya. *Confusion matrix* merupakan matriks berukuran N x N dengan N adalah

jumlah kelas target yang akan dievaluasi. Cara kerja *Confusion matrix* yaitu melakukan perbandingan antara nilai prediksi dengan nilai sebenarnya. Berikut nilai untuk mengevaluasi model klasifikasi.

a. *Accuracy*

Accuracy merupakan matrix yang digunakan untuk menghitung perbandingan keseluruhan prediksi benar terhadap keseluruhan nilai kelas.

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

b. *Precision*

Precision merupakan matrix yang digunakan untuk menghitung presisi atau data prediksi positif dari keseluruhan hasil yang diprediksi positif.

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

c. *Recall*

Recall merupakan matrix yang digunakan untuk menghitung jumlah prediksi pada kelas yang sama.

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

d. *F1-Score*

F1-Score merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Rentang pembobotan *F1-score* adalah 0 sampai 1 dengan 0 adalah nilai terburuk dan 1 adalah nilai terbaik.

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

4. Hasil dan Pembahasan

Proses eksperimen pada penelitian ini menggunakan *Google Colab* dalam menjalankan modelnya. Bahasa pemrograman yang digunakan yaitu *python* serta penggunaan *library* Keras dan *Tensorflow*. Jumlah eksperimen keseluruhan sebanyak 192 eksperimen.

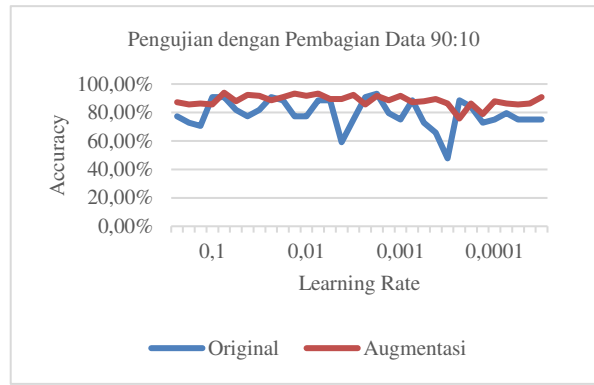
Tabel 2.Skenario Pengujian

Split Data	Learnin g Rate	Den se	Batch Size	Optimize r	Accuracy	
					Asli	Aug
90:10	0,1	32	64	Adamax	90.90	87.12
90:10	0,1	32	64	Nadam	72.72	85.60
90:10	0,1	32	128	Adamax	70.45	86,36
90:10	0,1	32	128	Nadam	90.90	85.60
90:10	0,1	128	64	Adamax	79.54	93.93
90:10	0,1	128	64	Nadam	81.81	87.87
90:10	0,1	128	128	Adamax	77.27	92.42
90:10	0,1	128	128	Nadam	81.81	91.66
90:10	0,01	32	64	Adamax	90.90	88.63
90:10	0,01	32	64	Nadam	88.63	90.90

Split Data	Learnin g Rate	Den se	Batch Size	Optimize r	Accuracy	
					Asli	Aug
90:10	0,01	32	128	Adamax	77.27	93.18
90:10	0,01	32	128	Nadam	77.27	91.66
90:10	0,01	128	64	Adamax	88.63	93.18
90:10	0,01	128	64	Nadam	88.63	89.39
90:10	0,01	128	128	Adamax	59.09	89.39
90:10	0,01	128	128	Nadam	75.00	92.42
90:10	0,001	32	64	Adamax	90.90	85.60
90:10	0,001	32	64	Nadam	93.18	91.66
90:10	0,001	32	128	Adamax	79.54	88.63
90:10	0,001	32	128	Nadam	75.00	91.66
90:10	0,001	128	64	Adamax	88.63	87.12
90:10	0,001	128	64	Nadam	72.72	87.87
90:10	0,001	128	128	Adamax	65.90	89.39
90:10	0,001	128	128	Nadam	47.72	86.36
90:10	0,0001	32	64	Adamax	88.63	75.75
90:10	0,0001	32	64	Nadam	84.09	86.36
90:10	0,0001	32	128	Adamax	72.72	78.78
90:10	0,0001	32	128	Nadam	75.00	87.87
90:10	0,0001	128	64	Adamax	79.54	86.36
90:10	0,0001	128	64	Nadam	75.00	85.60
90:10	0,0001	128	128	Adamax	75.00	86.36
90:10	0,0001	128	128	Nadam	75.00	90.90
80:20	0,1	32	64	Adamax	69.31	94.31
80:20	0,1	32	64	Nadam	85.22	92.42
80:20	0,1	32	128	Adamax	73.86	92.04
80:20	0,1	32	128	Nadam	72.72	92.42
80:20	0,1	128	64	Adamax	78.40	93.56
80:20	0,1	128	64	Nadam	81.81	92.42
80:20	0,1	128	128	Adamax	69.31	90.90
80:20	0,1	128	128	Nadam	71.59	93,18
80:20	0,01	32	64	Adamax	82.95	92.80
80:20	0,01	32	64	Nadam	80.68	91.66
80:20	0,01	32	128	Adamax	78.40	91.66
80:20	0,01	32	128	Nadam	76.13	93.18
80:20	0,01	128	64	Adamax	84.09	92.04
80:20	0,01	128	64	Nadam	84.09	91.28
80:20	0,01	128	128	Adamax	71.59	93.56
80:20	0,01	128	128	Nadam	81.81	93.93
80:20	0,001	32	64	Adamax	84.09	90.90
80:20	0,001	32	64	Nadam	82.95	92.42
80:20	0,001	32	128	Adamax	77.27	92.04
80:20	0,001	32	128	Nadam	78.40	94.31
80:20	0,001	128	64	Adamax	82.95	93.93
80:20	0,001	128	64	Nadam	78.40	93.18
80:20	0,001	128	128	Adamax	76.13	93.56
80:20	0,001	128	128	Nadam	73.86	92.04
80:20	0,0001	32	64	Adamax	71.59	77.65
80:20	0,0001	32	64	Nadam	70.45	89.77
80:20	0,0001	32	128	Adamax	60.22	78.40
80:20	0,0001	32	128	Nadam	73.86	83.71
80:20	0,0001	128	64	Adamax	73.86	86.36

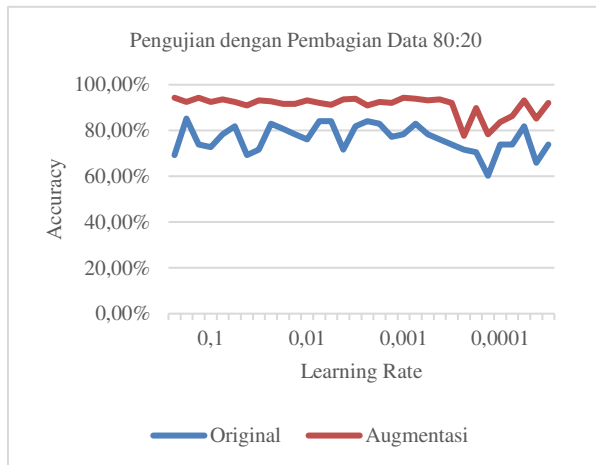
Split Data	Learning Rate	Dense	Batch Size	Optimizer	Accuracy	
					Asli	Aug
80:20	0,0001	128	64	Nadam	81.81	93.18
80:20	0,0001	128	128	Adamax	65.90	85.22
80:20	0,0001	128	128	Nadam	73.86	92.04
70:30	0,1	32	64	Adamax	62.87	93.43
70:30	0,1	32	64	Nadam	75.00	91.91
70:30	0,1	32	128	Adamax	46.96	86.66
70:30	0,1	32	128	Nadam	59.09	82.32
70:30	0,1	128	64	Adamax	69.69	92.42
70:30	0,1	128	64	Nadam	75.00	90.40
70:30	0,1	128	128	Adamax	57.57	83.08
70:30	0,1	128	128	Nadam	68.93	88.38
70:30	0,01	32	64	Adamax	77.27	88.63
70:30	0,01	32	64	Nadam	71.21	93.43
70:30	0,01	32	128	Adamax	43.93	86.61
70:30	0,01	32	128	Nadam	65.15	90.90
70:30	0,01	128	64	Adamax	75.75	90.65
70:30	0,01	128	64	Nadam	75.00	91.16
70:30	0,01	128	128	Adamax	42.42	88.13
70:30	0,01	128	128	Nadam	56.81	83.58
70:30	0,001	32	64	Adamax	74.24	87.87
70:30	0,001	32	64	Nadam	78.78	93.18
70:30	0,001	32	128	Adamax	43.93	82.57
70:30	0,001	32	128	Nadam	68.93	84.34
70:30	0,001	128	64	Adamax	75.75	92.17
70:30	0,001	128	64	Nadam	56.06	88.13
70:30	0,001	128	128	Adamax	43.93	83.83
70:30	0,001	128	128	Nadam	57.57	90.65
70:30	0,0001	32	64	Adamax	62.12	78.28
70:30	0,0001	32	64	Nadam	78.78	88.13
70:30	0,0001	32	128	Adamax	59.09	70.20
70:30	0,0001	32	128	Nadam	46.96	80.05
70:30	0,0001	128	64	Adamax	75.75	82.57
70:30	0,0001	128	64	Nadam	75.00	89.14
70:30	0,0001	128	128	Adamax	47.72	78.78
70:30	0,0001	128	128	Nadam	35.60	87.37

Tabel 2 menunjukkan hasil pengujian menggunakan citra asli memiliki akurasi tertinggi diperoleh pada eksperimen ke-18 dengan pembagian data 90:10 serta parameter berupa *learning rate* 0.001, *dense* 32, *batch size* 64, dan *optimizer* Nadam. Pengujian tersebut menghasilkan *accuracy* 93.18%, *precision* 100%, *recall* 81.81%, dan *fi score* 89.99%. Sedangkan hasil pengujian menggunakan data citra augmentasi akurasi tertingginya diperoleh pada eksperimen ke-3 yang pembagian datanya sebesar 80:20 serta parameter berupa *learning rate* 0.1, *dense* 32, *batch size* 128, dan *optimizer* Adamax. Pengujian tersebut menghasilkan *accuracy* 94.31%, *precision* 91.66%, *recall* 100%, dan *fi score* 95.64%



Gambar 5. Grafik Hasil Pengujian dengan Pembagian Data 90:10

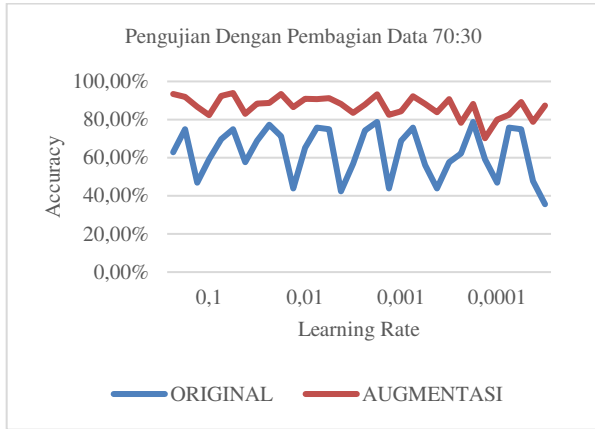
Gambar 5 menunjukkan bahwa pengujian yang menggunakan citra asli menghasilkan akurasi tertinggi pada eksperimen ke-18 dengan parameter berupa *learning rate* 0.001, *dense* 32, *batch size* 64, dan *optimizer* Nadam. Pengujian tersebut menghasilkan *accuracy* 93.18%, *precision* 100%, *recall* 81.81%, dan *fi score* 89.99%. Sedangkan pengujian menggunakan data citra augmentasi akurasi tertingginya diperoleh pada eksperimen ke-5 dengan parameter berupa *learning rate* 0.1, *dense* 128, *batch size* 64, dan *optimizer* Adamax. Pengujian tersebut menghasilkan *accuracy* 93.93%, *precision* 97.05%, *recall* 100.0%, dan *fi score* 98.50%. Pada grafik tersebut dapat dilihat bahwa hasil akurasi yang menggunakan augmentasi data lebih tinggi dibandingkan hasil akurasi tanpa augmentasi data. Hasil akurasi menggunakan augmentasi data meningkat sebesar 0.65% dibandingkan data asli



Gambar 6. Grafik Hasil Pengujian dengan Pembagian Data 80:20

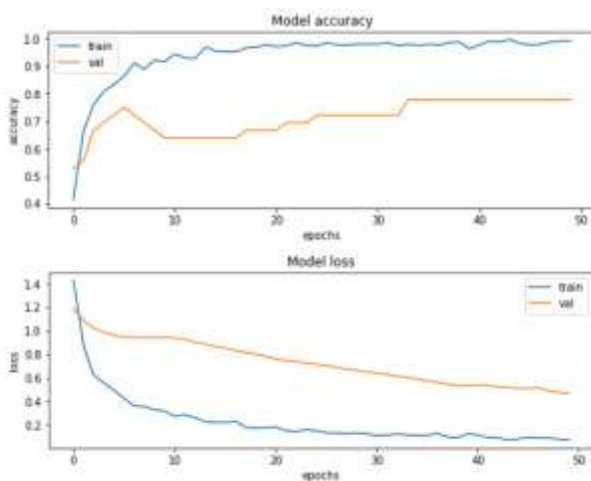
Gambar 6 menunjukkan bahwa pengujian yang menggunakan citra asli memiliki akurasi tertinggi pada eksperimen ke-2 dengan parameter berupa *learning rate* 0.1, *dense* 32, *batch size* 64, dan *optimizer* Nadam. Pengujian tersebut menghasilkan *accuracy* 85.22%, *precision* 88.00%, *recall* 100%, dan *fi score* 93.61%. Pengujian menggunakan data citra augmentasi memiliki akurasi tertinggi pada eksperimen ke-1 dengan parameter berupa *learning rate* 0.1, *dense* 32, *batch size* 64, dan *optimizer* Adamax. Pengujian

tersebut menghasilkan *accuracy* 94.31%, *precision* 88.00%, *recall* 100.0%, dan *f1 score* 93.61%. Pada grafik tersebut dapat dilihat bahwa hasil akurasi yang menggunakan augmentasi data lebih tinggi dibandingkan hasil akurasi tanpa augmentasi data. Hasil akurasi menggunakan augmentasi data meningkat sebesar 9,09% dibandingkan data asli



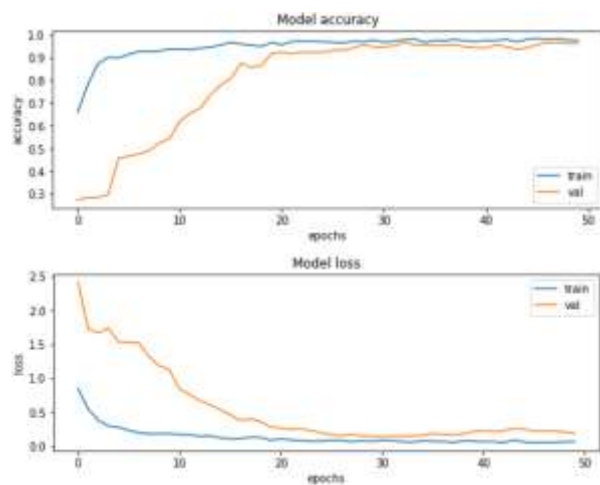
Gambar 7. Grafik Hasil Penguujian dengan Pembagian Data 70:30

Grafik 7 menunjukkan bahwa penguujian yang menggunakan citra asli memiliki akurasi tertinggi pada eksperimen ke-18 dengan parameter berupa *learning rate* 0.001, *dense* 32 *batch size* 64, dan *optimizer* Nadam. Penguujian yang menggunakan citra asli tersebut menghasilkan *accuracy* 78.78%, *precision* 88.23%, *recall* 90.32%, dan *f1 score* 89.54%. Penguujian menggunakan data citra augmentasi memiliki akurasi tertinggi pada eksperimen ke-1 dengan parameter berupa *learning rate* 0.1, *dense* 32, *batch size* 64, dan *optimizer* Adamax. Penguujian yang menggunakan citra augmentasi tersebut menghasilkan *accuracy* 93.43%, *precision* 95.05%, *recall* 97.97%, dan *f1 score* 96.48%. Pada grafik tersebut dapat dilihat bahwa hasil akurasi yang menggunakan augmentasi data lebih tinggi dibandingkan hasil akurasi tanpa augmentasi data. Hasil akurasi menggunakan augmentasi data meningkat sebesar 14.65% dibandingkan data asli



Gambar 8. Grafik Model Accuracy dan Loss pada Training Data Citra Asli

Gambar 8 menunjukkan hasil *accuracy* dan *loss* dari data *train* dan *valid* pada penguujian citra asli. Garis berwarna biru menunjukkan pergerakan dari data *train*, sedangkan garis berwarna orange menunjukkan pergerakan dari data *valid*. Hasil tersebut diperoleh pada eksperimen ke-18 dengan parameter berupa *learning rate* 0.001, *dense* 32, *batch size* 64, dan *optimizer* Nadam. Grafik tersebut menunjukkan bahwa nilai *accuracy* yang diperoleh pada *training* data citra asli naik hingga *epoch* 50 yaitu sebesar 0.7778. Nilai akurasi tersebut tidak terlalu bagus karena nilainya belum mendekati angka 1. Grafik tersebut juga menunjukkan bahwa nilai *loss* yang diperoleh pada *training* data citra augmentasi terus menurun hingga *epoch* 50 yaitu sebesar 0.4634. Nilai *loss* tersebut tidak terlalu bagus karena nilainya belum mendekati angka 0.



Gambar 9. Grafik Model Accuracy dan Loss pada Training Data Citra Augmentasi

Gambar 9 menunjukkan hasil *accuracy* dan *loss* dari data *train* dan *valid* pada penguujian citra augmentasi. Garis berwarna biru menunjukkan pergerakan dari data *train*, sedangkan garis berwarna orange menunjukkan pergerakan dari data *valid*. Hasil tersebut diperoleh pada eksperimen ke-1 dengan parameter berupa *learning rate* 0.1, *dense* 32, *batch size* 64, dan *optimizer* Adamax. Grafik tersebut menunjukkan bahwa nilai *accuracy* yang diperoleh pada *training* data citra asli naik hingga *epoch* 50 yaitu sebesar 0.96635. Nilai akurasi tersebut bagus karena nilainya sudah mendekati angka 1. Grafik tersebut juga menunjukkan bahwa nilai *loss* yang diperoleh pada *training* data citra augmentasi terus menurun hingga *epoch* 50 yaitu sebesar 0.1875. Nilai *loss* tersebut juga bagus karena nilainya sudah mendekati angka 0.

Berdasarkan eksperimen yang telah dilakukan, diperoleh hasil bahwa parameter mempengaruhi nilai akurasi. Penguujian yang menggunakan citra data asli mencapai hasil yang tinggi pada saat menggunakan *optimizer* Nadam, *dense* 32, *batch size* 64. Penguujian yang menggunakan citra data augmentasi mencapai hasil yang tinggi pada saat menggunakan *optimizer* Adamax, *batch size* 64, *learning rate* 0.1. Nilai akurasi yang dihasilkan pada penggunaan *dense* 32 lebih tinggi dibandingkan *dense* 128. Pada *batch size* 64 nilai

akurasi juga lebih tinggi daripada *batch size* 128. Nilai akurasi *learning rate* dengan nilai 0.1 lebih tinggi dibandingkan *learning rate* lainnya. Hasil penelitian menunjukkan bahwa penggunaan augmentasi dapat meningkatkan hasil akurasi pengujian. Penggunaan *hyperparameter optimization* juga mempengaruhi hasil akurasi penelitian. Semakin kecil parameter *dense* dan *batch size* yang digunakan, maka semakin tinggi nilai akurasi. Semakin besar *learning rate* yang digunakan, maka semakin tinggi nilai akurasi.

5. Kesimpulan

Penelitian klasifikasi citra penyakit daun tanaman padi ini menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG-19 serta penggunaan augmentasi berupa *height shift*, *width shift*, dan *shear range*. Data citra yang digunakan berupa data asli sebanyak 440 citra dan data augmentasi sebanyak 1320 citra yang terdiri atas empat kelas yaitu *blast*, *brown spot*, *leaf smut*, dan *healthy*. Nilai akurasi menggunakan citra augmentasi tertinggi yang diperoleh yaitu sebesar 94.31%, *precision* 88.00%, *recall* 100.0%, dan *f1 score* 93.61%. Nilai akurasi tanpa augmentasi tertinggi yang diperoleh yaitu *accuracy* 93.18%, *precision* 100%, *recall* 81.81%, dan *f1 score* 89.99%. Hasil penelitian menunjukkan bahwa augmentasi dapat meningkatkan hasil akurasi. Penggunaan *hyperparameter optimization* juga mempengaruhi hasil akurasi penelitian. Untuk penelitian selanjutnya, direkomendasikan menggunakan data citra yang lebih banyak agar akurasi yang dihasilkan lebih tinggi, serta menggunakan augmentasi lain yang lebih bagus. Semakin kecil parameter *dense* dan *batch size* yang digunakan, semakin tinggi hasil akurasi yang diperoleh. Semakin besar *learning rate* yang digunakan, semakin tinggi hasil akurasi yang didapatkan.

6. Daftar Rujukan

- [1] Badan Pusat Statistik, "Luas Panen dan Produksi Padi di Indonesia 2021," *Badan Pusat Statistik*, Jul. 12, 2022. <https://www.bps.go.id/publication/2022/07/12/c52d5cebe530c363d0ea4198/luas-panen-dan-produksi-padi-di-indonesia-2021.html> (accessed Oct. 22, 2022).
- [2] L. R. Stevani, "Produksi padi di Madiun turun akibat serangan OPT," *ANTARAJATIM*, Aug. 10, 2022.
- [3] A. Walascha, A. Febriana, D. Saputri, D. Sri Nur Haryanti, R. Tsania, and Y. Sanjaya, "Inventarisasi Jenis Penyakit yang Menyerang Daun Tanaman Padi (*Oryza sativa* L.)."
- [4] A. Jinan and B. H. Hayadi, "Klasifikasi Penyakit Tanaman Padi Menggunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron)," *Journal of Computer and Engineering Science*, vol. 1, no. 2, 2022.
- [5] E. Anggiratih, S. Siswanti, S. K. Octaviani, and A. Sari, "Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 dengan Transfer Learning," *Jurnal Ilmiah SINUS*, vol. 19, no. 1, p. 75, Jan. 2021, doi: 10.30646/sinus.v19i1.526.
- [6] L. Hernando, Ikhsan, A. Avaldo, and Ismael, "Implementasi Fuzzy Logic pada Alat Pemisah Buah Tomat," *Jurnal Sains dan Informatika*, vol. 08, pp. 55–61, 2022, doi: 10.22216/jsi.v8i2.1637.
- [7] R. A. Saputra, Suharyanto, S. Wasiyanti, D. F. Saefudin, A. Supriyatna, and A. Wibowo, "Rice Leaf Disease Image Classifications Using KNN Based on GLCM Feature Extraction," *J Phys Conf Ser*, vol. 1641, no. 1, Nov. 2020, doi: 10.1088/1742-6596/1641/1/012080.
- [8] R. Kumar, G. Baloch, A. Baseer Buriro, and J. Bhatti, "Fungal Blast Disease Detection in Rice Seed using Machine Learning," *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, 2021, [Online]. Available: www.ijacsa.thesai.org
- [9] A. A. Huda, B. Setiaji, and F. R. Hidayat, "Implementasi Gray Level Co-Occurrence Matrix (GLCM) untuk Klasifikasi Penyakit Daun Padi," *Jurnal Pseudocode*, vol. 9, 2022, [Online]. Available: www.ejournal.unib.ac.id/index.php/pseudocode
- [10] Y. Amelia, P. Eosina, and F. A. Setiawan, "Perbandingan Metode Deep Learning dan Machine Learning untuk Klasifikasi (Ujicoba pada Data Penyakit Kanker Payudara)," *INOVA-TIF*, vol. 1, no. 2, 2018, doi: 10.32832/inova-tif.v1i2.5449.
- [11] A. S. Simbolon, N. I. Pangaribuan, and N. M. Aruan, "Analisis Sentimen Aplikasi E-Learning Selama Pandemi Covid-19 dengan Menggunakan Metode Support Vector Machine dan Convolutional Neural Network," *SEMINASTIKA*, vol. 3, no. 1, pp. 16–25, Nov. 2021, doi: 10.47002/seminastika.v3i1.236.
- [12] M. F. Naufal, "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 2, 2021, doi: 10.25126/jtiik.2021824553.
- [13] S. Pratap Singh and I. Kumar, "Rice Plant Infection Recognition using Deep Neural Network Systems." [Online]. Available: <https://www.ifis.uni->
- [14] D. Hindarto and H. Santoso, "Plat Nomor Kendaraan dengan Convolution Neural Network," *JII: Jurnal Inovasi Informatika Universitas Pradita*, vol. 6, Sep. 2021.
- [15] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs

- training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 375–381, Nov. 2021, doi: 10.1016/j.gltp.2021.08.027.
- [16] N. N. Malvade, R. Yakkundimath, G. Saunshi, M. C. Elemmi, and P. Baraki, “A comparative analysis of paddy crop biotic stress classification using pre-trained deep neural networks,” *Artificial Intelligence in Agriculture*, vol. 6, pp. 167–175, 2022, doi: 10.1016/j.aiia.2022.09.001.
- [17] C. Nisa’, E. Y. Puspaningrum, and H. Maulana, “Penerapan Metode Convolutional Neural Network untuk Klasifikasi Penyakit Daun Apel pada Imbalanced Data,” *Seminar Nasional Informatika Bela Negara*, vol. 1, 2020.
- [18] W. M. Pradnya D and A. P. Kusumaningtyas, “Analisis Pengaruh Data Augmentasi Pada Klasifikasi Bumbu Dapur Menggunakan Convolutional Neural Network,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, p. 2022, Oct. 2022, doi: 10.30865/mib.v6i4.4201.
- [19] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, “Hyperparameter optimization for machine learning models based on Bayesian optimization,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.
- [20] Vbookshelf, “Rice Leaf Disease Dataset,” *Kaggle*, 2020. <https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases> (accessed Oct. 19, 2022).
- [21] Riyan Shayaz, “Rice Leaf,” *Kaggle*, 2020. <https://www.kaggle.com/datasets/shayanriyaz/riceleaves> (accessed Oct. 19, 2022).
- [22] H. B. P. and V. K. D. J. P. Shah, “Rice Leaf Diseases Data Set,” *UCI Machine Learning*, 2017. <https://archive.ics.uci.edu/ml/datasets/rice+leaf+diseases> (accessed Oct. 19, 2022).
- [23] F. Nurona Cahya *et al.*, “Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN),” *Jurnal Sistem Informasi*, vol. 10, pp. 618–626, 2021, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [24] F. Habib Hawari, F. Fadillah, M. Rifqi Alviandi, and T. Arifin, “Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (Convolutional Neural Network),” *JURNAL RESPONSIF*, vol. 4, no. 2, pp. 184–189, 2022, [Online]. Available: <https://ejurnal.ars.ac.id/index.php/jti>
- [25] G. Y. Alhafis, J. Jasril, S. Sanjaya, F. Syafria, and E. Budianita, “Klasifikasi Citra Daging Sapi dan Daging Babi Menggunakan Ekstraksi Ciri dan Convolutional Neural Network,” *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 3, p. 653, Jun. 2022, doi: 10.30865/jurikom.v9i3.4175.