

## IMPLEMENTASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* (CNN) UNTUK KLASIFIKASI PENYAKIT DAUN PADA TANAMAN PADI

Wahyu Ismawan<sup>1</sup>, Abd. Charis Fauzan<sup>2</sup>

<sup>1,2</sup>Program Studi Ilmu Komputer, Universitas Nahdlatul Ulama Blitar, Indonesia

Email: [ismawanwahyu426@gmail.com](mailto:ismawanwahyu426@gmail.com), [abdcharis@unublitar.ac.id](mailto:abdcharis@unublitar.ac.id)

### Abstrak

Tanaman padi merupakan salah satu produk unggulan petani yang menjadi sumber utama makanan pokok sebagian besar masyarakat Indonesia. Salah satu tantangan utama yang dihadapi adalah serangan penyakit seperti *bacterial leaf blight*, *brown spot*, dan *leaf smut* yang dapat mempengaruhi produktivitas hasil panen jika tidak segera diatasi. Untuk mengatasi masalah ini, penelitian dilakukan dengan menggunakan algoritma *convolutional neural network* (CNN) untuk mengklasifikasi penyakit daun padi. Dalam proses ini melibatkan lima tahap yaitu *convolutional layer*, *activation function (ReLU)*, *fully connected*, dan *dropout* dengan menggunakan dataset 120 citra yang dibagi menjadi tiga kelas yang terdiri 40 citra. Proses pelatihan model menggunakan data pelatihan dengan 50 *epoch*, dimana hasil pelatihan menunjukkan akurasi sebesar 81% dengan nilai loss sebesar 49%, sedangkan evaluasi menggunakan *confusion matrix* pada data uji menunjukkan nilai akurasi sebesar 79%. Hasil penelitian ini menunjukkan bahwa model CNN mampu mengenali penyakit daun dengan baik, sehingga berpotensi penting dalam mendeteksi penyakit sejak dini dan pemantauan secara efektif untuk meningkatkan hasil produktivitas tanaman padi.

**Kata kunci:** *Tanaman Padi, Penyakit Daun, Convolutional Neural Network, Klasifikasi Citra, Confusion Matrix*

## IMPLEMENTATION OF *CONVOLUTIONAL NEURAL NETWORK* (CNN) ALGORITHM FOR CLASIFICATION OF LEAF DISEASES IN RICE PLANTS

### Abstract

Rice is one of the main products of farmers who are the main source of staple food for most Indonesians. One of the main challenges faced is the attack of diseases such as *bacterial leaf blight*, *brown spot*, and *leaf smut* which can affect crop productivity if not immediately addressed. To overcome this problem, research was conducted using *convolutional neural network* (CNN) algorithm to classify rice leaf diseases. This process involves five stages, namely *convolutional layer*, *activation function (ReLU)*, *fully connected*, and *dropout* using a dataset of 120 images divided into three classes of 40 images. The model training process uses training data with 50 epochs, where the training results show an accuracy of 80% with a loss value 49%, while evaluation using *confusion matrix* on test data shows an accuracy value 79%. The result of this study indicate that the CNN model is able to recognize leaf diseases well, so it is potentially important in detecting diseases early and monitoring effectively to increase the productivity of rice plants.

**Keywords:** *Rice Plant, Leaf Disease, Convolutional Neural Network, Image Classification, Confusion Matrix*

### 1. PENDAHULUAN

Tanaman padi merupakan salah satu produk unggulan petani yang menjadi sumber utama makanan pokok Sebagian besar masyarakat Indonesia. Oleh karena itu, sektor pertanian memiliki peran penting dalam mendukung kebutuhan pangan dan banyak dikembangkan diberbagai wilayah Indonesia. Namun,serangan organisme pengganggu tanaman (OPT) masih menjadi ancaman bagi keberhasilan panen padi [1].

OPT dapat menyebabkan kerusakan pada daun padi yang mempengaruhi pada produktivitas dan kesejahteraan petani, padi sangat rentan terhadap berbagai macam penyakit seperti *bacterial leaf blight*, *brown spot*, dan *leaf smut* yang dapat mengaruhi hasil panen secara besar jika tidak segera diatasi [2]. Selain itu, banyak kendala yang dihadapi petani, terutama dalam mendeteksi penyakit pada tanaman padi yang biasanya dilakukan secara manual melalui gejala-gejala seperti kerusakan daun, pertumbuhan yang tidak sempurna, hingga kegagalan dalam

menghasilkan biji padi yang memakan waktu lama atau menunggu gejala yang lebih jelas, sebelum dideteksi jenis pentakit tang menyerangan tanaman, sehingga Sebagian besar gagal panen padi yang disebabkan oleh penyakit duan padi [3].

Dengan perkembangan teknologi informasi memungkinkan dapat mengidentifikasi pentakit dengan memanfaatkan kecerdassan buatan, salah satunya melalui model pengenalan citra. Salah satu cabang dari kecerdasan buatan, yaitu *machine learning* yang telah mengalami perkembangan dengan pesat, yang pada akhirnya membawa kepada pengembangan lebih lanjut yaitu *deep learning* [4]. Metode *deep learning* juga memiliki beberapa algoritma seperti *Artificial Neural Network* (ANN), *Convolutional Neural Network* (CNN), dan *Recurrent Neural Network* (RNN) [5]. Salah satu algoritma yang paling banyak digunakan dalam *deep learning* adalah *Convolutional Neural Network* (CNN) [6].

CNN merupakan metode *deeo learning* yang elah efektif dalam bidang pertanian. CNN bekerja dengan cara yang mirip dengan mata manusia mengenali pola, sehingga mampu mendeteksi pola-pola rumit dalam gambar. Dalam hal Kesehatan tanaman padi, CNN dapat digunakan untuk membedakan antara yang sehat dan yang terjangkit penyakit. Dengan sistem ini, petani bisa lebih cepat dan tepat dalam mengambil Tindakan pencegahan atau perbaikan, menghemat pengguna sumberdaya, dan pada akhirnya meningkatkan hasil panen [7]. CNN memiliki sejumlah keunggulan dalam berbagai aplikasi, terutama dalam pengenalan citra atau gambar. Salah satu kekuatan utama CNN adalah kemampuannya untuk mendeteksi pola-pola yang kompleks dengan tingkat akurasi yang tinggi, bahkan pada citra yang sulit diinterpretasikan oleh manusia [8].

Menurut beberapa penelitian yang telah dilakukan menggunakan algoritma *Convolutional Neural Network* (CNN) dalam klasifikasi penyakit daun. Salah satunya penelitian dengan judul Klasifikasi Penyakit Pada Tanaman Buah Menggunakan *Convolutional Neural Network* (CNN), bahwa CNN dapat digunakan dalam mengidentifikasi enam jenis penyakit daun dan dua daun sehat. Kemudian mendapatkan nilai akurasi dari rumus *confusion matrix* sebesar 94,76% [9]. Selanjutnya, peneltian tentang Klasifikasi Penyakit Daun Padi Menggunakan Algortima *Convolutional Neural Network* (CNN) dalam membantu petani mengatasi penyakit daun padi. Dalam penelitian ini, menggunakan 600 dataset yang dibagi manjadi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Pada 100 *epoch* pelatihan mencapai 98% dan nilai akurasi menggunakan *confusion matrix* sebesar 98% [10]. Penelitian terdahulu dalam jurnal yang berjudul Klasifikasi Penyakit Tanaman Dari Citra Daun Padi dengan menggunakan *Convolutional Neural Network*, metode CNN berhasil diterpkan dengan baik dengan menghasilkan tingkat akurasi

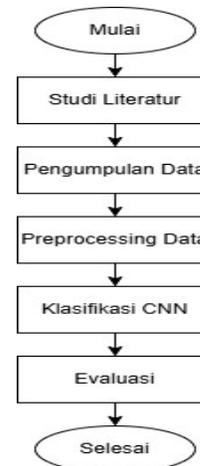
sebesar 97,1% [11]. Berdasarkan beberapa penelitian terdahulu tersebut, dapat disimpulkan bahwa algoritma CNN memiliki tingkat akurasi yang tinggi dan efektif yang baik dalam mengklasifikasi penyakit daun dalam berbagai jenis tanaman khususnya pada duan padi.

Berdasarkan pemaparan diatas, maka pada penelitian ini akan mengimplementasikan algortima *Convolutional Neural Network* (CNN) dalam mengklasifikasi penyakit daun pada tanaman padi dan diharapkan dapat memberikan solusi yang efektif dan akurat dalam membantu petani untuk mengenali jenis penyakit daun padi, sehingga mampu meningkatkan efisiensi dalam penanganan penyakit serta meningkatkan hasil panen.

## 2. METODE PENELITIAN

### 2.1 Alur Penelitian

Dalam membangun sistem klasifikasi penyakit daun pada tanaman padi dibutuhkan alur atau tahapan guna menjadi landasan dalam melakukan penelitian. Berikut alur penelitian yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Pada Gambar 1 menunjukkan alur penelitian yang diawalu dengan studi literatur untuk memahami konsep dasar mengenai algoritma *Convolutional Neural Network* (CNN) dan klasifikasi penyakit daun pada tanaman padi. Referensi berbagai jurnal buku digunakan untuk menentukan pendekatan yang sesuai. Tahap selanjutnya adalah pengumpulan data, dimana citra daun padi dari Kaggle.com yang berisi dataset 120 citra gambar daun padi yang terinfeksi penyakit yang dikelompokkan menjadi 3 kelas berdasarkan penyakit, yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut*, setiap kelas terdiri dari 40 citra gambar. Setelah tahap pengumpulan data, dilakukan *preprocessing* data untuk meningkatkan citra daun padi sebelum digunakan untuk pelatihan model. Proses ini meliputi perubahan ukuran gambar menjadi 200x200 piksel dan augmentasi data seeperti rotasi, *flipping*, dan

pergeseran. Selanjutnya, tahap klasifikasi dengan menerapkan algoritma CNN, model ini dirancanag dengan beberapa lapisan diantaranya *convolutional layer*, *ReLU*, *pooling*, dan *fully connected* untuk meningkatkan kemampuan dalam mengklasifikasi penyakit daun. Pada tahap akhir adalah evaluasi untuk menilai performa model dalam mengklasifikasi daun dengan menggunakan *confusion matrix* dan *classification report*.

### 2.1.1 Studi Literatur

Studi literatur adalah proses mengumpulkan data dan menganalisis sumber informasi dari berbagai artikel, buku, dan jurnal yang relevan untuk mendukung pemahaman penelitian terhadap topik yang sedang diteliti. Studi literatur bertujuan untuk mendapatkan pemahaman atau wawasan dari penelitian terdahulu, teori, metode, dan pendekatan yang sudah ada yang berkaitan dengan metode CNN dan klasifikasi penyakit daun padi.

### 2.1.2 Pengumpulan Data

Tahap pengumpulan data adalah tahap yang digunakan sebagai bahan proses penelitian. Pada penelitian ini dataset yang diperoleh dari link URL [Rice Leaf Diseases Dataset - UCI Repository UCI Machine Learning Repository](#) berjumlah 120 citra gambar yang dibagi menjadi 3 kelas ( 40 citra perkelas penyakit *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Smut*). Dataset ini mencakup citra yang mewakili berbagai jenis penyakit daun padi yang nantinya digunakan untuk melatih model klasifikasi penyakit tanaman.

Table 1. Dataset

No	Nama Kelas	Jumlah Dat
1.	Bacterial Leaf Blight	40
2.	Brown Spot	40
3.	Leaf Smut	40



Gambar 2.  
*Bacterial  
Leaf Blight*



Gambar 3.  
*Brown Spot*



Gambar 4.  
*Leaf Smut*

### 2.1.3 Preprocessing Data

Tahap *preprocessing* merupakan tahap awal pengelolaan citra yakni dengan melakukan *resizing* dan augmentasi citra yang selanjutnya akan digunakan pada proses klasifikasi. Pada penelitian ini akan dilakukan *resizing* citra sebesar 200x200 piksel agar proses komputasi model CNN tidak memakan waktu lama tanpa mengurugu detail dari citra. Setalh itu, tahap augmentasi yakni memodifikasi seperti *flipping*, *rescale*, *zoom*, dan rotasi acak. Augmentasi bertujuan menambah variasi data, mengatasi jumlah data latih yang terbatas, dan meningkatkan performa model CNN.

### 2.1.4 Klasifikasi CNN

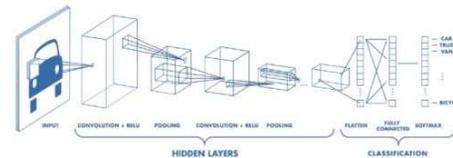
Setelah melalui *preprocessing* dan augmentasi dataset akan menjadi inputan model CNN yang dibuat. Langkah pertama yang dilakukan adalah mengupload dataset ke *google drive* kemudian dihubungkan dengan *google colab*. Setelah dataset terhubung dengan *google colab*, data diproses dengan pembagian dataset menjadi data latih, validasi, dan uji dengan ratio 80:20:20, serta dilakukan normalisasi piksel, *resizing*, dan augmentasi untuk meningkatkan kualitas data. Dataset yang telah diproses kemudian digunakan sebagai input untuk membangun model CNN yang terdiri dari lapisan *convolutional* untuk ekstraksi fitur, *pooling* untuk reduksi dimensi, *fully connected* untuk klasifikasi, dan *dropout* untuk mengurangi overfitting. Model dilatih menggunakan data latih dan validasi dengan algoritma optimasi seperti adam dan fungsi loss categorical.

### 2.1.5 Evaluasi

Evaluasi dilakukan dengan menggunakan *confusion matrix* untuk mendaptkan nilai akurasi, presisi, recall, dan f1-score, sehingga dapat mengukur seberapa baik model dalam megklasifikasi data dengan benar dan juga memahami kinerja model.

## 2.2 Convolutional Neural Network (CNN)

*Convolutional Neural Network (CNN)* merupakan salah satu metode pengembangan *deep learning* untuk digunakan pengenalan objek atau *image processing* [12]. Metode CNN memiliki beberapa *layer* untuk mengektrasi dan klasifikasi yaitu *convolutional layer*, *pooling layer*, *fully connected layer*, dan *dropout* [13]. Sebagaimana ditunjukkan pada Gambar 5.



Gambar 5. Arsitektur CNN

### 2.2.1 Convolutional Layer

*Convolutional Layer* terdiri dari filter dan *image maps*, yang berfungsi untuk mengekstrak fitur dari data masukan. *Layer* ini memperkuat korelasi spasial gambar dan data dengan masukan kecil dan berbentuk persegi [14]. Adapun rumus *convolutional layer* ditunjukkan pada persamaan (1) [15].

$$n_{out} = \binom{n_{in}-k+2p}{s} + 1 \quad (1)$$

Adapun rumus operasi konvolusi adalah persamaan (2) [16].

$$FM[i]_{j,k} = (\sum_m \sum_n N_{[j-m,k-n]} F_{[m,n]}) + 1 \quad (2)$$

Setelah proses konvolusi selsai, langkah selanjutnya adalah mengaktifkan pemicu menggunakan fungsi *Rectified Linear Unit (ReLU)*. Fungsi ini diterapkan pada setiap piksel di *feature map*, dimana piksel dengan nilai ukuran 0 akan diubah menjadi 0. Proses rumus  $f(x) = \{x, \text{jika } x > 0; 0, \text{jika } x < 0\}$  [17].

### 2.2.2 Pooling Layer

*Pooling Layer* digunakan untuk memperkecil *feature map*. Terdapat dua jenis *pooling layer*, yaitu *average pooling* dan *max pooling*, namun fitur yang paling digunakan di CNN adalah *max pooling* yakni dengan mengubah keluaran lapisan konvolusi menjadi beberapa grid kecil, kemudian nilai maksimum dari setiap grid tersebut ditentukan untuk menghasilkan matriks citra yang telah diekstraksi [18]. Proses ini akan menghasilkan *output* berupa *array* yang berisi nilai maksimum yang dipilih.

### 2.2.3 Fully Connected Layer

*Fully Connected Layer* adalah lapisan terakhir dari arsitektur *perception multilayer* yang mana setiap neuron dari lapisan aktivitas sebelumnya akan terhubung ke lapisan ini dan sebelum itu setiap neuron di lapisan masukan perlu diubah menjadi data 1-Dimensi [19]. *Fully Connected Layer* terdiri atas *input layer*, *hidden layer*, dan *output layer* yang masing-masing mempunyai fungsi berbeda.

#### i. Input Layer

*Input Layer* merupakan proses penggaungan seluruh matriks *feature map* yang diperoleh dari *pooling layer* lalu semua piksel tersebut diubah menjadi vector yang mencakup jumlah piksel dari matriks *feature map* hasil *pooling layer*, dan selanjutnya semua nilai *input layer* digunakan untuk perhitungan pada *hidden layer* [20].

#### ii. Hidden Layer

*Hidden Layer* yaitu proses perkalian antara nilai-nilai dari *input layer* dengan bobot yang sudah dinormalisasikan lalu ditambah dengan nilai bias. Adapun rumus perhitungannya ditunjukkan pada persamaan (3)[20].

$$z_{in_i} = \sum_{j=1}^n X_j * V_{j,i} + V_{0,i} \quad (3)$$

Setelah melakukan perhitungan maka langkah selanjutnya adalah mengaktifkan fungsi ReLU untuk seluruh hasil perhitungan yang akan menghasilkan nilai Z. hasil dari proses perhitungan ini akan diterapkan pada proses perhitungan *output layer*.

#### iii. Output Layer

*Output Layer* yaitu proses pengkalian antara nilai-nilai hasil dari perhitungan pada *hidden layer* dengan bobot yang sudah dinormalisasikan, lalu ditambah dengan nilai bias. Rumus perhitungan ditunjukkan pada persamaan (4) [21].

$$z_r = \sum_c X_c W_{cr} + b_r \quad (4)$$

Setelah perhitungan selesai, masukkan fungsi *softmax* untuk semua hasil perhitungan untuk mendapatkan nilai keluaran Y. Rumus perhitungan ditunjukkan pada persamaan (5) [19].

$$p(x) = \frac{e^x}{\sum_{k=1}^k e^x} \quad (5)$$

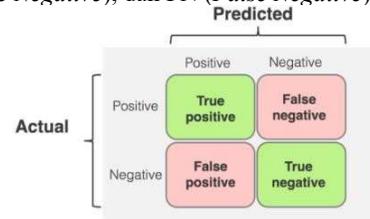
### 2.2.4 Dropout

*Dropout* adalah teknik untuk mencegah overfitting dengan cara menonaktifkan beberapa neuron yang tidak diperlukan secara acak [22]. Hal ini menunjukkan bahwa meskipun neuron yang baru terbentuk akan dihilangkan sementara, jaringan dan

bobot baru tidak akan dimasukkan kedalam neuron yang ada selama proses *backpropagation* [23].

### 2.3 Confusion Matrix

*Confusion Matrix* merupakan hasil evaluasi yang digunakan dari kalsifikasi data mining yang menampilkan dalam bentuk table untuk menghitung nilai akurasi, presisi, recall, dan f1-score dari sistem yang dirancang. Dalam *confusion matrix* memiliki 4 istilah TP (*True Positive*), FP (*False Positive*), TN (*True Negative*), dan FN (*False Negative*) [24].



Gambar 6. Confusion Matrix

Rumus untuk menentukan nilai akurasi, presisi, recall, dan f1-score ditunjukkan pada persamaan (6), persamaan (7), persamaan (8), dan persamaan (9) [25].

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

$$Presisi = \frac{TP}{TP+FP} \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$F1-score = \frac{2 \times presisi \times recall}{presisi+recall} \quad (9)$$

## 3. HASIL DAN ANALISIS

Hasil penelitian ini, peneliti dilakukan dengan mengimplementasikan algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasi penyakit daun pada tanaman padi. Hasil ini diperoleh dari penelitian yang sudah dilakukan.

### 3.1 Pembagian Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari 3 jenis penyakit daun padi, yaitu *bacterial leaf blight*, *brown spot*, dan *leaf smut*. Masing-masing jenis penyakit memiliki total 40 citra yang dibagi menjadi 2 kelompok, yaitu data training dan data testing. Sebanyak 32 citra untuk model training, sementara 8 citra digunakan untuk menguji model testing. Pembagian dataset ini untuk ketiga jenis penyakit dengan rasio 80:20, dimana 80% dari total data digunakan untuk pelatihan dan 20% digunakan untuk pengujian. Berikut pembagian dataset ditunjukkan pada Gambar 7.

Jumlah All Bacterial leaf blight : 40  
 Jumlah Train Bacterial leaf blight: 32  
 Jumlah Test Bacterial leaf blight : 8  
 Jumlah All Brown spot : 40  
 Jumlah Train Brown spot : 32  
 Jumlah Test Brown spot : 8  
 Jumlah All Leaf smut : 40  
 Jumlah Train Leaf smut : 32  
 Jumlah Test Leaf smut : 8

Gambar 7. Pembagian Dataset

### 3.2 Preprocessing

Setelah membagi dataset, langkah berikutnya adalah *preprocessing* data agar siap digunakan. Proses ini mencakup berbagai tahap, seperti *resize* citra agar ukuran gambar menjadi 200x200 piksel, kemudian melakukan normalisasi nilai piksel untuk mempercepat proses pelatihan dan menerapkan augmentasi data, seperti *rotasi*, *flipping*, dan pengeseran. Berikut beberapa fitur pada citra daun padi berpenyakit ditunjukkan pada Gambar 8.



Gambar 8. Fitur Daun Padi Berpenyakit

### 3.3 Klasifikasi CNN

Setelah melakukan *preprocessing* pada dataset, langkah selanjutnya adalah membangun model menggunakan metode CNN untuk mengenali pola visual pada citra. Arsitektur CNN ini terdiri dari 4 lapisan konvolusi yang berfungsi mengekstraksi fitur penting seperti tekstur dan tepi pada citra, dengan bantuan aktivasi ReLU untuk mengubah nilai matrik negative menjadi nol yang bertujuan untuk mengurangi *overfitting*. Setiap lapisan konvolusi diikuti oleh lapisan *max pooling* untuk mengurangi ukuran data dan mempercepat proses kinerja sistem. Setelah fitur diekstraksi, data diratakan menggunakan lapisan *flatten* dan diteruskan *fully connected* masing-masing memiliki 200 dan 500 neuron. Untuk menghindari *overfitting*, dua lapisan *dropout* ditambahkan di antara lapisan *fully connected*. Model ini diakhiri dengan lapisan *output* yang memiliki neuron, sesuai dengan tiga kategori penyakit daun padi. Berikut hasil arsitektur dari model yang telah dikembangkan menggunakan algoritma CNN dalam bentuk *sequential* ditunjukkan pada Gambar 9.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 16)	448
max_pooling2d (MaxPooling2D)	(None, 99, 99, 16)	0
conv2d_1 (Conv2D)	(None, 97, 97, 32)	4,648
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 21, 21, 128)	73,856
max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 200)	2,560,200
dropout (Dropout)	(None, 200)	0
dense_1 (Dense)	(None, 500)	100,500
dropout_1 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 3)	1,503

Total params: 2,750,643 (10.53 MB)  
 Trainable params: 2,750,643 (10.53 MB)  
 Non-trainable params: 0 (0.00 B)

Gambar 9. Hasil Arsitektur Model CNN

Setelah merancang dan merangkum arsitektur model CNN, tahap berikutnya adalah pelatihan model. Pada tahap ini, data pelatihan digunakan untuk melatih model dalam mengenali pola-pola pada citra daun padi. Proses pelatihan dilakukan dengan menggunakan 50 *epoch*, dengan hasil model mencapai nilai akurasi sebesar 0,81 atau 81% dan nilai loss model sebesar 0,49 atau 49%. Proses ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali pola-pola pada data pelatihan. Dimana bobot dan bias model terus disesuaikan menggunakan algoritma *backpropagation* dan optimasi untuk mengurangi kesalahan prediksi. Performa model selama proses pelatihan dan validasi ditunjukkan pada Gambar 10.

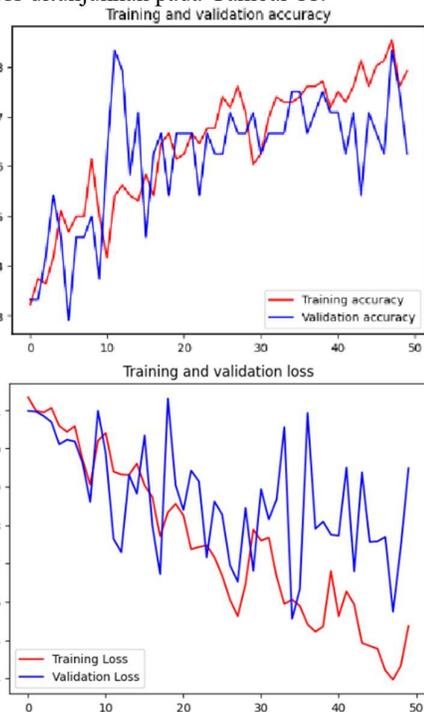
```

Epoch 1/50
/home/ismawati/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your 'set_
sel_
warn_if_super_not_called()
Epoch 2/50
10/10 ----- 12s 782ms/step - accuracy: 0.3367 - loss: 1.1203 - val_accuracy: 0.3333 - val_loss: 1.0079
Epoch 3/50
10/10 ----- 7s 478ms/step - accuracy: 0.3759 - loss: 1.0977 - val_accuracy: 0.3333 - val_loss: 1.0962
Epoch 4/50
10/10 ----- 10s 480ms/step - accuracy: 0.3534 - loss: 1.0934 - val_accuracy: 0.4167 - val_loss: 1.0847
Epoch 5/50
10/10 ----- 12s 494ms/step - accuracy: 0.4379 - loss: 1.0819 - val_accuracy: 0.5417 - val_loss: 1.0686
Epoch 6/50
10/10 ----- 11s 708ms/step - accuracy: 0.5714 - loss: 1.0505 - val_accuracy: 0.4583 - val_loss: 1.0108
Epoch 7/50
10/10 ----- 8s 482ms/step - accuracy: 0.4899 - loss: 1.0258 - val_accuracy: 0.2917 - val_loss: 1.0226
Epoch 8/50
10/10 ----- 11s 488ms/step - accuracy: 0.4839 - loss: 1.0414 - val_accuracy: 0.4583 - val_loss: 1.0185
Epoch 9/50
10/10 ----- 12s 625ms/step - accuracy: 0.4757 - loss: 0.9883 - val_accuracy: 0.4583 - val_loss: 0.9633
Epoch 10/50
10/10 ----- 7s 470ms/step - accuracy: 0.5862 - loss: 0.9097 - val_accuracy: 0.5000 - val_loss: 0.8605
Epoch 11/50
10/10 ----- 10s 462ms/step - accuracy: 0.5289 - loss: 0.9895 - val_accuracy: 0.3750 - val_loss: 1.0982
Epoch 12/50
10/10 ----- 9s 697ms/step - accuracy: 0.3768 - loss: 1.0921 - val_accuracy: 0.6250 - val_loss: 0.9864
Epoch 13/50
10/10 ----- 7s 458ms/step - accuracy: 0.6272 - loss: 0.9148 - val_accuracy: 0.8333 - val_loss: 0.7647
Epoch 14/50
10/10 ----- 9s 702ms/step - accuracy: 0.5355 - loss: 0.9306 - val_accuracy: 0.7917 - val_loss: 0.7290
Epoch 15/50
10/10 ----- 8s 474ms/step - accuracy: 0.5448 - loss: 0.8872 - val_accuracy: 0.5833 - val_loss: 0.9315
Epoch 16/50
10/10 ----- 11s 475ms/step - accuracy: 0.5956 - loss: 0.9290 - val_accuracy: 0.7083 - val_loss: 0.8818
Epoch 17/50
10/10 ----- 9s 683ms/step - accuracy: 0.5947 - loss: 0.8754 - val_accuracy: 0.4583 - val_loss: 1.0340
Epoch 18/50
10/10 ----- 8s 466ms/step - accuracy: 0.6287 - loss: 0.8074 - val_accuracy: 0.6250 - val_loss: 0.7085
Epoch 19/50
10/10 ----- 9s 674ms/step - accuracy: 0.6802 - loss: 0.7759 - val_accuracy: 0.6667 - val_loss: 0.6724
Epoch 20/50
10/10 ----- 7s 482ms/step - accuracy: 0.6711 - loss: 0.7489 - val_accuracy: 0.5417 - val_loss: 1.1301
Epoch 21/50
10/10 ----- 9s 670ms/step - accuracy: 0.5877 - loss: 0.8726 - val_accuracy: 0.6667 - val_loss: 0.9036
Epoch 22/50
10/10 ----- 8s 473ms/step - accuracy: 0.7036 - loss: 0.7955 - val_accuracy: 0.6667 - val_loss: 0.8396
Epoch 23/50
10/10 ----- 8s 538ms/step - accuracy: 0.7175 - loss: 0.7124 - val_accuracy: 0.6667 - val_loss: 0.9421
Epoch 24/50
10/10 ----- 11s 703ms/step - accuracy: 0.6488 - loss: 0.7403 - val_accuracy: 0.5417 - val_loss: 0.9143
Epoch 25/50
10/10 ----- 7s 459ms/step - accuracy: 0.6442 - loss: 0.7806 - val_accuracy: 0.6667 - val_loss: 0.7258
Epoch 26/50
10/10 ----- 9s 670ms/step - accuracy: 0.6276 - loss: 0.7956 - val_accuracy: 0.6250 - val_loss: 0.8618
Epoch 27/50
10/10 ----- 8s 474ms/step - accuracy: 0.7647 - loss: 0.6793 - val_accuracy: 0.6250 - val_loss: 0.8273
Epoch 28/50
10/10 ----- 9s 600ms/step - accuracy: 0.6899 - loss: 0.6401 - val_accuracy: 0.7083 - val_loss: 0.6956
Epoch 29/50
10/10 ----- 7s 501ms/step - accuracy: 0.7494 - loss: 0.6033 - val_accuracy: 0.6667 - val_loss: 0.6523
Epoch 30/50
10/10 ----- 10s 482ms/step - accuracy: 0.6629 - loss: 0.7265 - val_accuracy: 0.6667 - val_loss: 0.8451
Epoch 31/50
10/10 ----- 9s 705ms/step - accuracy: 0.5522 - loss: 0.8185 - val_accuracy: 0.7083 - val_loss: 0.6804
Epoch 32/50
10/10 ----- 8s 457ms/step - accuracy: 0.5974 - loss: 0.7239 - val_accuracy: 0.6250 - val_loss: 0.8940
    
```

Epoch 32/50	96	643ms/step	- accuracy: 0.6885 - loss: 0.7988 - val_accuracy: 0.6667 - val_loss: 0.8158
Epoch 33/50	88	466ms/step	- accuracy: 0.7426 - loss: 0.7199 - val_accuracy: 0.6667 - val_loss: 0.8666
Epoch 34/50	106	468ms/step	- accuracy: 0.6745 - loss: 0.6585 - val_accuracy: 0.6667 - val_loss: 1.0552
Epoch 35/50	86	469ms/step	- accuracy: 0.7388 - loss: 0.6242 - val_accuracy: 0.7500 - val_loss: 0.5558
Epoch 36/50	86	590ms/step	- accuracy: 0.7581 - loss: 0.5521 - val_accuracy: 0.7500 - val_loss: 0.6326
Epoch 37/50	86	473ms/step	- accuracy: 0.7295 - loss: 0.5838 - val_accuracy: 0.6667 - val_loss: 1.0926
Epoch 38/50	124	673ms/step	- accuracy: 0.7506 - loss: 0.5361 - val_accuracy: 0.7083 - val_loss: 0.7905
Epoch 39/50	86	467ms/step	- accuracy: 0.7983 - loss: 0.5456 - val_accuracy: 0.7500 - val_loss: 0.8092
Epoch 40/50	96	552ms/step	- accuracy: 0.7927 - loss: 0.5478 - val_accuracy: 0.7083 - val_loss: 0.7752
Epoch 41/50	96	550ms/step	- accuracy: 0.7882 - loss: 0.5481 - val_accuracy: 0.7083 - val_loss: 0.7724
Epoch 42/50	96	455ms/step	- accuracy: 0.7145 - loss: 0.6999 - val_accuracy: 0.6250 - val_loss: 0.9584
Epoch 43/50	96	662ms/step	- accuracy: 0.7982 - loss: 0.4796 - val_accuracy: 0.7083 - val_loss: 0.6791
Epoch 44/50	76	463ms/step	- accuracy: 0.8017 - loss: 0.5421 - val_accuracy: 0.5417 - val_loss: 0.9376
Epoch 45/50	96	595ms/step	- accuracy: 0.7882 - loss: 0.4458 - val_accuracy: 0.7083 - val_loss: 0.7562
Epoch 46/50	86	472ms/step	- accuracy: 0.7847 - loss: 0.4712 - val_accuracy: 0.6667 - val_loss: 0.7578
Epoch 47/50	106	468ms/step	- accuracy: 0.7969 - loss: 0.4242 - val_accuracy: 0.6250 - val_loss: 0.7088
Epoch 48/50	114	468ms/step	- accuracy: 0.8128 - loss: 0.4998 - val_accuracy: 0.8333 - val_loss: 0.5743
Epoch 49/50	124	708ms/step	- accuracy: 0.7134 - loss: 0.4741 - val_accuracy: 0.7500 - val_loss: 0.7428
Epoch 50/50	76	450ms/step	- accuracy: 0.8168 - loss: 0.4994 - val_accuracy: 0.6250 - val_loss: 0.9486

Gambar 10. Hasil Pelatihan Model CNN

Setelah itu, hasil proses pelatihan ditampilkan dalam bentuk grafik untuk memperlihatkan perkembangan model dalam mengklasifikasi data dengan benar pada setiap *epoch* selama proses pelatihan, sedangkan grafik loss menunjukkan penurunan kesalahan prediksi secara bertahap. Melalui analisis grafik ini, dapat dipastikan bahwa model telah mencapai performa yang optimal tanpa mengalami *overfitting*. Berikut grafik validasi akurasi dan loss ditunjukkan pada Gambar 11.

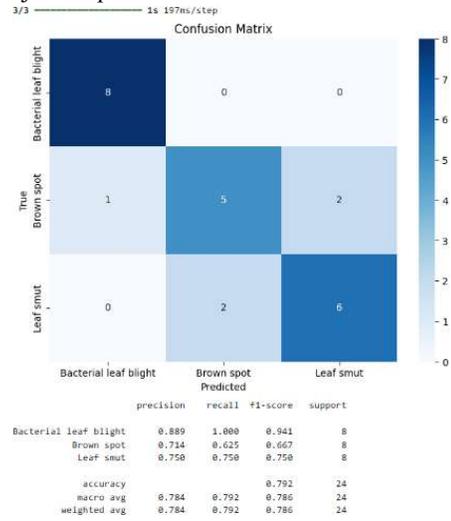


Gambar 11. Grafik Validasi Akurasi dan Loss

### 3.4 Evaluasi Model

Evaluasi merupakan tahap penting yang selalu dilakukan setelah penyelesaian pembuatan model sistem, evaluasi ini bertujuan untuk menguji keberhasilan yang diperoleh dari suatu penelitian. Dalam penelitian ini, pengujian dilakukan dengan menggunakan *confusion matrix* di *google colab* untuk mengukur nilai akurasi, presisi, recall dan f1-score

dari model yang dikembangkan. Model diuji menggunakan dataset pengujian yang terdiri dari 24 citra, yaitu 8 citra *bacterial leaf blight*, 8 citra *brown smut*, dan 8 citra *leaf smut*. Hasil *confusion matrix* ditunjukkan pada Gambar 12.



Gambar 12. Hasil Confusion Matrix

Pada Gambar 12 menunjukkan *confusion matrix* dengan model CNN yang digunakan untuk mengklasifikasi 3 jenis penyakit daun padi. Dengan menggunakan persamaan (6-9), dapat dihitung metrik evaluasi seperti presisi, recall, dan f1-score. Diketahui bahwa dengan menggunakan 50 *epoch* dalam *classification report*, model memiliki performa yang bervariasi untuk setiap kelas. Untuk kelas *bacterial leaf blight*, model mencapai nilai presisi sebesar 88% dan f1-score sebesar 94%, menunjukkan hasil yang sangat baik. namun, untuk kelas *brown spot*, model memiliki nilai presisi sebesar 71%, recall sebesar 62%, dan f1-score sebesar 66%, yang menunjukkan bahwa model ini sering melakukan kesalahan dalam mengklasifikasi. Sementara itu, untuk kelas *leaf smut*, model memperoleh nilai presisi sebesar 75%, recall sebesar 75%, dan f1-score sebesar 75%, yang menunjukkan hasil yang lebih stabil dibandingkan *brown spot*. Secara keseluruhan, model ini dengan menggunakan algoritma CNN memiliki nilai akurasi sebesar 79%, menunjukkan kemampuan yang baik dalam mengklasifikasi penyakit berdasarkan citra daun padi.

### 4. KESIMPULAN

Pada penelitian ini mengimplementasikan algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasi penyakit daun padi yang diambil dari situs *Kaggle.com* sebanyak 120 citra gambar yang terbagi dalam 3 kelas yaitu, *bacterial leaf blight* sebanyak 40, *brown spot* sebanyak 40, dan *leaf smut* sebanyak 40 dengan model CNN berupa 4 *layer* beserta *activation* ReLU pada setiap layernya, 4 *layer pooling* (*Max Pooling*), 1 *layer flatten*, 2 *layer*

*fully connedted*, dan 1 *layer output* yang menggunakan aktivasi *softmax* dengan 3 neuron sesuai jumlah kelas. Berdasarkan history model CNN ysng telah dibuat, memiliki nilai akurasi sebesar 0,81% atau 81% dengan loss sebesar 0,49 atau 49%, serta berdasarkan evaluasi model CNN ysng telah dibuat menggunakan perhitungan *confusion matrix* diperoleh nilai akurasi sebesar 79%. Dengan demikian dapat disimpulkan bahwa pengimplementasian algoritma CNN untuk klasifikasi penyakit daun padi memberikan akurasi yang cukup baik.

## 5. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Abd. Chariz Fauzan, S.Kom., M.Kom. selaku dosen mata kuliah atas saran dan dukungannya selama proses penelitian ini. Penulis juga mengucapkan terima kasih kepada teman-teman di peminatan deep learning atas kontribusi dan saran berharga yang telah diberikan untuk mendukung penelitian ini. Sebagai penutup, penulis menyampaikan rasa terima kasih yang mendalam kepada keluarga dan teman-teman atas dukungan, pengertian, dan motivasi yang mereka berikan selama proses penelitian ini.

## 6. REFERENSI

- [1] S. Suhendar, A. Purnama, and E. Fauzi, "Deteksi Penyakit Pada Daun Tanaman Padi Menggunakan Metode Convolutional Neural Network," *J. Ilm. Inform. Glob.*, vol. 14, no. 3, pp. 62–67, 2023, doi: 10.36982/jiig.v14i3.3478.
- [2] S. Azizah, A. I. Pradana, and D. Hartanti, "Identifikasi Kesehatan Daun Tanaman Padi Menggunakan Klasifikasi Biner Sehat dan Tidak Sehat dengan Algoritma Convolutional Neural Network ( CNN ) Di Kabupaten Klaten," vol. 13, 2024, doi: 10.34010/komputika.v13i2.12771.
- [3] S. Sheila, I. Permata Sari, A. Bagas Saputra, M. Kharil Anwar, and F. Restu Pujianto, "Deteksi Penyakit Pada Daun Padi Berbasis Pengolahan Citra Menggunakan Metode Convolutional Neural Network (CNN)," *Multinetics*, vol. 9, no. 1, pp. 27–34, 2023, doi: 10.32722/multinetics.v9i1.5255.
- [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] G. Y. Christiawan, R. A. Putra, A. Sulaiman, E. Poerbaningtyas, and S. W. Putri Listio, "Penerapan Metode Convolutional Neural Network (CNN) Dalam Mengklasifikasikan Penyakit Daun Tanaman Padi," *J-Intech*, vol. 11, no. 2, pp. 294–306, 2023, doi: 10.32664/j-intech.v11i2.1006.
- [6] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, and Galih Wasis Wicaksono, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, 2021, doi: 10.29207/resti.v5i6.3607.
- [7] M. R. Alamadani and D. Indriyana, "KLASIFIKASI KESEHATAN TANAMAN PADI MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK," vol. 8, no. 5, pp. 10177–10182, 2024.
- [8] B. Irawan *et al.*, "Optimasi Algoritma Adam untuk Peningkatan Akurasi Klasifikasi Penyakit Daun Padi Berbasis CNN," vol. 13, no. 105, pp. 738–743, 2024.
- [9] J. V. P. Putra, F. Ayu, and B. Julianto, "Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 2, no. 1, pp. 155–162, 2023, [Online]. Available: <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/2888>
- [10] A. A. SHELEMO, "KLASIFIKASI HAMA TANAMAN PADI BERDASARKAN CITRA DAUN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK," *Nucl. Phys.*, vol. 13, no. 1, pp. 104–116, 2023.
- [11] S. Yuliany, Aradea, and Andi Nur Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022, doi: 10.24002/jbi.v13i1.5022.
- [12] M. Kholilurrahman, W. A. Syafei, and O. D. Nurhayati, "Klasifikasi Image Processing Pada Citra Warna Daun Padi Menggunakan Metode Convolutional Neural Network," *J. Ilm. Sains*, vol. 23, no. 2, pp. 175–186, 2023, doi: 10.35799/jis.v23i2.50415.
- [13] Anissa Ollivia Cahya Pratiwi, "Klasifikasi Jenis Anggur Berdasarkan Bentuk Daun Menggunakan Convolutional Neural Network Dan K-Nearest Neighbor," *J. Ilm. Tek. Inform. dan Komun.*, vol. 3, no. 2, pp. 201–224, 2023, doi: 10.55606/juitik.v3i2.535.
- [14] N. P. Ekananda and D. Riminarsih, "Identifikasi Penyakit Pneumonia

- Berdasarkan Citra Chest X-Ray Menggunakan Convolutional Neural Network,” *J. Ilm. Inform. Komput.*, vol. 27, no. 1, pp. 79–94, 2022, doi: 10.35760/ik.2022.v27i1.6487.
- [15] P. Dabbo and F. Y. Bisilisin, “Klasifikasi Motif Kain Tenun Sabu Rajua Menggunakan Convolutional Neural Network (CNN) Berbasis Citra,” *KETIK J. Inform.*, vol. 1, no. 06, pp. 11–18, 2024.
- [16] H. A. Pratama, “Implementasi convolutional neural network pada pengenalan isyarat tangan untuk komunikasi tuna wicara,” *J. Komput. Dan Teknol. Inf.*, vol. 2, no. 2, pp. 65–71, 2024, doi: 10.26714/jkti.v2i2.13931.
- [17] Bagus Untung Saputra, Gunawan, and Wresti Andriani, “Pengenalan Motif Batik Pesisir Pulau Jawa Menggunakan Convolutional Neural Network,” *Nuansa Inform.*, vol. 17, no. 2, pp. 119–125, 2023, doi: 10.25134/ilkom.v17i2.32.
- [18] M. I. Wahid, S. A. Mustamin, and D. A. Lawi, “Identifikasi Dan Klasifikasi Citra Penyakit Daun Tomat Menggunakan Arsitektur Inception V4,” *onferensi Nas. Ilmu Komput.*, vol. 5, no. 2019, pp. 257–264, 2021.
- [19] D. N. Fauzi, R. Y. N. Fuadah, and I. Safitri, “Klasifikasi Supraventrikular Arrhythmia Berdasarkan Sinyal EKG Menggunakan Convolutional Neural Network Supraventrikular Arrhythmia Classification Based On ECG Signal With Convolutional Neural Network,” *e-Proceeding Eng.*, vol. 8, no. 6, pp. 3255–3262, 2022.
- [20] F. M. Qotrunnada and P. H. Utomo, “Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker,” *Prisma*, vol. 5, pp. 799–807, 2022.
- [21] Rexion Alondeo Boimau and Yampi R. Kaesmetan, “Klasifikasi Citra Digital Bumbu dan Rempah Dengan Algoritma Convolutional Neural Network (CNN),” *Repeater Publ. Tek. Inform. dan Jar.*, vol. 2, no. 3, pp. 26–34, 2024, doi: 10.62951/repeater.v2i3.81.
- [22] “Klasifikasi Kanker Kulit Berdasarkan Data Citra Benign Dan Malignant Menggunakan Convolutional Neural Network,” vol. 3, no. April, 2024.
- [23] R. A. Hasyani, S. M. Simbolon, Y. Mufida, Y. Ester, and B. Ritonga, “Klasifikasi Malaria melalui Penggunaan Convolutional Neural Network pada Citra Sel Darah,” no. 6, 2023.
- [24] M. Farij Amrulloh *et al.*, “Klasifikasi Penyakit Daun Bawang Menggunakan Algoritma CNN Xception Penulis Korespondensi,” *Agustus*, vol. 8, pp. 2549–7952, 2024.
- [25] A. T. Akbar *et al.*, “KLASIFIKASI EKSPRESI WAJAH MENGGUNAKAN COVOLUTIONAL NEURAL FACIAL EXPRESSION CLASSIFICATION USING CONVOLUTIONAL NEURAL,” vol. 11, no. 6, 2024, doi: 10.25126/jtiik.202411888.