

## ARTICLE

# Klasifikasi Jenis Penyakit Pada Tanaman Padi Menggunakan Algoritma Convolutional Neural Network

## *Classification of Paddy Rice Plant Diseases Using Convolutional Neural Network Algorithms*

Bagus Mauludi Kusuma,\* Teguh Iman Hermanto, dan Candra Dewi Lestari

Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana, Purwakarta, Indonesia

\*Penulis Korespondensi: bagusmauludi11@wastukencana.ac.id

(Disubmit 20-07-24; Diterima 09-08-24; Dipublikasikan online pada 05-02-25)

### Abstrak

Padi adalah tanaman pangan utama di Indonesia dan memiliki peran vital dalam perekonomian serta kehidupan sehari-hari masyarakat. Namun, produksi padi saat ini mengalami penurunan akibat serangan hama dan penyakit. Deteksi dini dan klasifikasi penyakit padi yang akurat sangat penting untuk mengurangi dampak negatif ini. Pada penelitian ini dilakukan pembangunan dan pelatihan model Convolutional Neural Network (CNN) untuk mengenali kondisi kesehatan tanaman padi. Model dilatih dengan dataset citra daun padi yang berasal dari situs Kaggle.com dengan jumlah data sebanyak 53.300 data citra, dan terdiri dari 4 kategori yaitu Brownspot, Leafblast, Hispa, dan Healthy. Proses persiapan data meliputi resize data, augmentasi data, dan rescaling pixel. Model dioptimasi dengan parameter terbaik yaitu 30 epoch, batch size 45, dan optimizer Lion. Hasil pengujian menunjukkan akurasi 75% untuk data uji dengan loss 59%, dan akurasi 76% untuk data latih dengan loss 61%. Model ini juga berhasil diimplementasikan dalam aplikasi mobile berbasis Android. Penelitian ini diharapkan dapat berkontribusi pada sektor pertanian Indonesia dengan menyediakan alat deteksi penyakit padi yang lebih efisien dan efektif.

**Kata kunci:** Convolutional Neural Network; Klasifikasi; Penyakit Padi

### Abstract

Paddy Rice is the main staple crop in Indonesia and plays a vital role in the country's economy and daily life. However, rice production is currently declining due to pest and disease attacks. Early detection and accurate classification of rice diseases are crucial to mitigate these negative impacts. This study involves the development and training of a Convolutional Neural Network (CNN) model to recognize the health condition of rice plants. The model was trained using a dataset of rice leaf images obtained from Kaggle.com, consisting of 53,300 images categorized into four classes: Brownspot, Leafblast, Hispa, and Healthy. Data preparation included resizing, data augmentation, and pixel rescaling. The model was optimized with the best parameters, including 30 epochs, a batch size of 45, and the Lion optimizer. Testing results showed an accuracy of 75% for the test data with a loss of 59%, and an accuracy of 76% for the training data with a loss of 61%. The model was successfully implemented in an Android-based mobile application. This research is expected to contribute to the Indonesian agricultural sector by providing a more efficient and effective tool for rice disease detection.

**KeyWords:** Classification; Convolutional Neural Network; Paddy Rice Plant Diseases

## 1. Pendahuluan

Beras, yang dihasilkan dari tanaman padi (*Oryza sativa*), adalah makanan pokok utama bagi masyarakat Indonesia. Dengan konsumsi per kapita mencapai sekitar 111,58 kg per tahun, sehingga tanaman padi dapat dikatakan memainkan peran penting dalam kehidupan sehari-hari masyarakat dan menjadi tulang punggung ketahanan pangan nasional[1]. Melihat pernyataan tersebut, dapat dikatakan bahwa tanaman padi adalah salah satu tumbuhan yang harus sangat diperhatikan. Namun, menurut data yang dikeluarkan oleh Badan Pusat Statistik, jumlah produksi padi pada tahun 2023 mengalami penurunan dibandingkan dengan jumlah produksi padi pada tahun 2022. Data tersebut menunjukkan penurunan sebanyak 2,05% untuk jumlah produksi padi atau senilai dengan 1,12 juta Ton GKG (Gabah Kering Giling) dan penurunan sebanyak 2,45% untuk luas panen atau senilai dengan 0,26 juta hektar[2]. Berkurangnya angka produksi dan jumlah luas panen padi ini dapat dipengaruhi oleh berbagai macam faktor, salah satu faktor yang mempengaruhi menurunnya angka tersebut adalah karena serangan hama dan penyakit pada tumbuhan padi.

Bagian daun sering menjadi tempat yang paling mudah untuk mendeteksi gejala penyakit dan serangan hama pada tumbuhan padi. Meskipun begitu, mengenali jenis penyakit memerlukan keahlian tertentu dan biasanya hanya dilakukan oleh para pakar dan para ahli[3]. Petani biasa yang tidak memiliki kompetensi yang mumpuni dalam bidang ilmu pengetahuan anatomi tumbuhan padi kerap mengalami kesulitan dalam mendeteksi penyakit dan jenis penyakit pada tumbuhan padi. Salah satu teknologi yang dapat dikembangkan untuk proses identifikasi atau klasifikasi kondisi kesehatan pada tumbuhan padi adalah dengan pengenalan citra daun padi menggunakan metode Convolutional Neural Network (CNN). CNN merupakan salah satu model algoritma pada bidang deep learning yang dapat digunakan untuk meng-klasifikasi suatu citra gambar[4].

1. Pendahuluan Beras, yang dihasilkan dari tanaman padi (*Oryza sativa*), adalah makanan pokok utama bagi masyarakat Indonesia. Dengan konsumsi per kapita mencapai sekitar 111,58 kg per tahun, sehingga tanaman padi dapat dikatakan memainkan peran penting dalam kehidupan sehari-hari masyarakat dan menjadi tulang punggung ketahanan pangan nasional[1]. Melihat pernyataan tersebut, dapat dikatakan bahwa tanaman padi adalah salah satu tumbuhan yang harus sangat diperhatikan. Namun, menurut data yang dikeluarkan oleh Badan Pusat Statistik, jumlah produksi padi pada tahun 2023 mengalami penurunan dibandingkan dengan jumlah produksi padi pada tahun 2022. Data tersebut menunjukkan penurunan sebanyak 2,05% untuk jumlah produksi padi atau senilai dengan 1,12 juta Ton GKG (Gabah Kering Giling) dan penurunan sebanyak 2,45% untuk luas panen atau senilai dengan 0,26 juta hektar[2]. Berkurangnya angka produksi dan jumlah luas panen padi ini dapat dipengaruhi oleh berbagai macam faktor, salah satu faktor yang mempengaruhi menurunnya angka tersebut adalah karena serangan hama dan penyakit pada tumbuhan padi.

Bagian daun sering menjadi tempat yang paling mudah untuk mendeteksi gejala penyakit dan serangan hama pada tumbuhan padi. Meskipun begitu, mengenali jenis penyakit memerlukan keahlian tertentu dan biasanya hanya dilakukan oleh para pakar dan para ahli[3]. Petani biasa yang tidak memiliki kompetensi yang mumpuni dalam bidang ilmu pengetahuan anatomi tumbuhan padi kerap mengalami kesulitan dalam mendeteksi penyakit dan jenis penyakit pada tumbuhan padi. Salah satu teknologi yang dapat dikembangkan untuk proses identifikasi atau klasifikasi kondisi kesehatan pada tumbuhan padi adalah dengan pengenalan citra daun padi menggunakan metode Convolutional Neural Network (CNN). CNN merupakan salah satu model algoritma pada bidang deep learning yang dapat digunakan untuk meng-klasifikasi suatu citra gambar[4].

Meskipun berbagai penelitian sebelumnya, seperti yang dilakukan oleh Khoiruddin (2022)[4] pada penelitiannya yang menghasilkan model dengan akurasi 98% dengan pelatihannya sebanyak 100 epochs. Penelitian lainnya oleh D. M. Saputra (2023)[3] telah berhasil mengembangkan model CNN untuk klasifikasi jenis penyakit pada daun padi dengan nilai akurasi 96%. Penelitian lainnya juga dilakukan oleh Hawari (2022)[5] yang berjudul Klasifikasi Penyakit Padi Menggunakan Algoritma CNN yang menghasilkan model dengan nilai akurasi sebesar 95%. Penelitian lainnya yang telah dilakukan sebelumnya oleh Sheila (2023)[6] yang menggunakan CNN dengan arsitektur Inception V3 dengan nilai akurasi sebesar 93,75%. Lalu terdapat pula penelitian lainnya yang telah dilakukan sebelumnya oleh Jinan (2022)[7] yang menggunakan CNN melalui

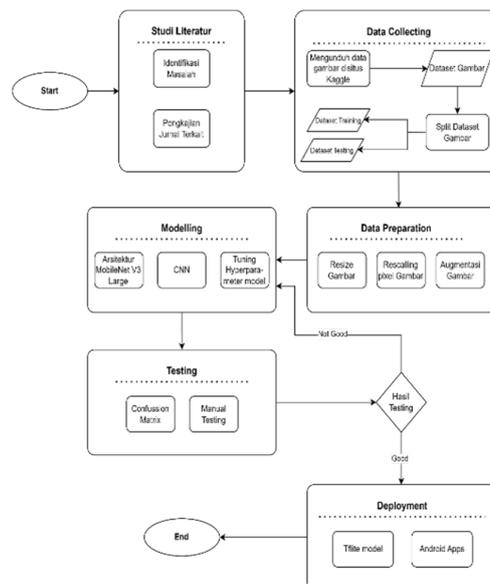
citra daun (*Multilayer Perceptron*) dengan nilai akurasi sebesar 91,75%. Namun sebagian besar penelitian-penelitian yang telah disebutkan tidak melanjutkan hasilnya ke tahap integrasi pada platform mobile berbasis Android. Keterbatasan ini menimbulkan kebutuhan akan penelitian lebih lanjut yang tidak hanya fokus pada akurasi model, tetapi juga integrasi model ke dalam aplikasi mobile, sehingga teknologi ini dapat lebih mudah diakses dan dimanfaatkan oleh petani di lapangan.

Dengan adanya pengembangan penelitian ini, harapannya dapat meningkatkan angka produksi dan luas panen padi pada masa yang akan datang dan dapat membantu para petani padi untuk mencegah juga mendeteksi terjadinya serangan hama dan penyakit pada tumbuhan padi dengan lebih mudah dan efisien.

## 2. Metode

### 2.1 Kerangka Penelitian

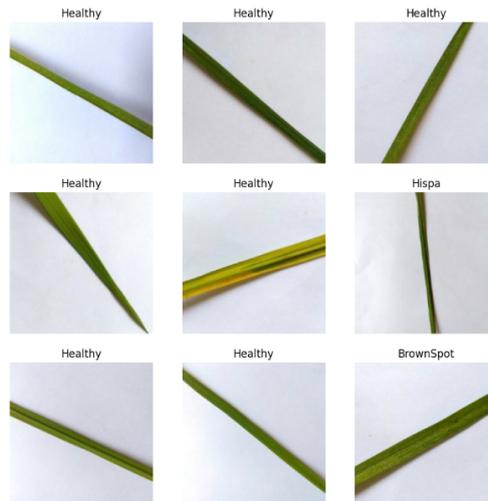
Penelitian ini dilakukan untuk membuat model yang dapat mengenali jenis penyakit pada tumbuhan padi menggunakan metode Convolutional Neural Network (CNN) serta melakukan integrasi model yang sudah dibuat kedalam aplikasi mobile berbasis Android. Aktivitas yang dilakukan pada penelitian mengacu pada kerangka penelitian yang telah penulis susun berdasarkan metode CRISP-DM. Adapun kerangka penelitian tersebut dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Penelitian

### 2.2 Data Collecting

Data yang digunakan berupa citra daun tumbuhan padi yang dimana citra gambar tersebut berasal dari dataset yang telah diunduh melalui situs Kaggle.com. Format data yang terdapat pada dataset yang telah diunduh adalah file data citra ber-ekstensi .jpg dengan ukuran keseluruhan dari dataset ini adalah sebesar 798MB, berjumlah total data citra sebanyak 53.300 data citra. Kategori penyakit yang terdapat pada dataset ini adalah brownspot, healthy, hispa, dan leafblast. Masing-masing data citra pada dataset memiliki 3 channel yaitu red, green, dan blue. Kualitas data citra pada dataset tergolong cukup baik. Sampel dari data citra pada dataset dapat dilihat pada Gambar 2.



Gambar 2. Sample Dataset

Kategori penyakit yang terdapat pada dataset dinilai cukup relevan dengan permasalahan penurunan angka produksi padi di Indonesia. Seperti yang dikutip dari Sitompul (2022)[8] yang menyebutkan bahwa diperkirakan sekitar 200.000-300.000 ton padi terserang hama dan berbagai macam penyakit seperti leaflblast, brownspot dan hispa disetiap tahunnya di Indonesia yang berpotensi menyebabkan penurunan produktivitas tanaman padi apabila serangan hama dan penyakit tersebut tidak dikendalikan.

Dataset yang sudah didapatkan selanjutnya akan dipisah menjadi dua kelompok yaitu data training yang akan digunakan untuk melatih model dalam proses mengklasifikasi citra sebanyak 2419 data atau 80% dari keseluruhan dataset, dan data testing yang akan digunakan untuk menguji model dalam mengklasifikasikan citra sebanyak 603 atau 20% dari keseluruhan dataset.

### 2.3 Data Preparation

Proses pertama pada tahapan data preparation adalah *resize data*, yakni proses penyesuaian ukuran seluruh data pada dataset. Dataset yang didapatkan memiliki data-data citra dengan ukuran yang berbeda-beda. Jika tidak ada penyesuaian ukuran data-data tersebut, maka model nantinya akan melakukan proses pengenalan fitur citra dengan tidak konsisten yang menyebabkan menurunnya tingkat akurasi model. Oleh karena itu proses *resize data* wajib dilakukan. Pada penelitian ini, penulis menyesuaikan ukuran seluruh data pada dataset menjadi 224x224 pixel.

Proses data preparation selanjutnya adalah augmentasi data, yaitu proses memperbanyak jumlah dataset dengan cara menduplikasi setiap data pada dataset namun dengan sudut pandang atau angle yang berbeda-beda. Tujuan dari proses ini adalah untuk menghindari overfitting pada model juga agar model dapat mengenali citra dengan berbagai sudut pandang[9]. Proses augmentasi dilakukan dengan menggunakan library keras dengan beberapa parameter, parameter-parameter tersebut adalah sebagai berikut:

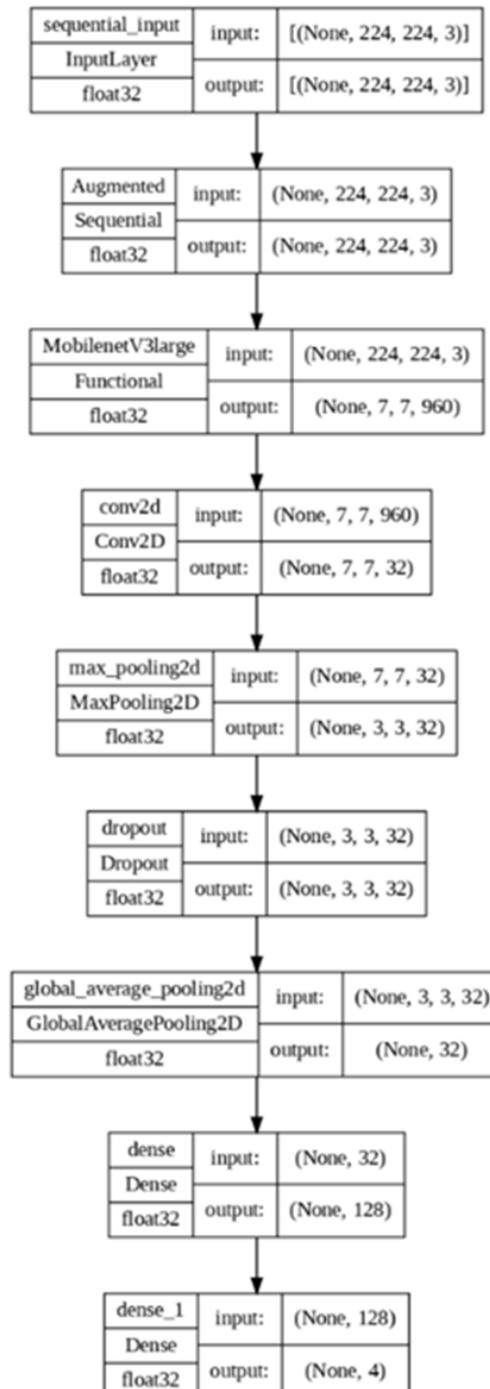
1. *RandomFlip*, digunakan sebagai lapisan augmentasi yang akan membalik gambar secara acak dalam arah horizontal.
2. *RandomRotation*, digunakan sebagai lapisan augmentasi yang akan memutar gambar secara acak sebesar  $\pm 10\%$  dari sudut aslinya.
3. *RandomZoom*, lapisan augmentasi yang akan memperbesar atau memperkecil gambar secara acak hingga 50% dari ukuran aslinya.

Proses *data preparation* yang terakhir adalah *rescaling pixel*. Proses *rescaling pixel* pada dataset dilakukan untuk memastikan bahwa semua nilai pixel berada dalam rentang 0 hingga 1. Hal ini penting karena nilai piksel asli berada dalam rentang 0 hingga 255. Tujuan dari proses ini adalah untuk meringankan beban komputasi pada model serta agar model dapat bekerja dengan lebih efisien dan lebih stabil. Dalam melakukan proses *rescaling pixel*, penulis menggunakan library keras untuk membagi setiap nilai pixel

dengan nilai 255. Dengan demikian, nilai pixel yang semula berada dalam rentang 0 hingga 255 akan diubah menjadi rentang 0 hingga 1.

## 2.4 Modelling

Model yang dibangun akan menggunakan arsitektur *MobileNetV3 Large* untuk membantu meningkatkan efisiensi model dan untuk menghindari overfitting pada model. *MobileNetV3 Large* adalah versi terbaru dari arsitektur *MobileNet* yang dirancang khusus untuk meningkatkan efisiensi dan kinerja model CNN pada perangkat seluler dan aplikasi *embedded*[10]. Dalam tahapan modelling juga, penulis akan melakukan optimasi model dengan melakukan eksperimen dengan mengubah beberapa parameter pada model. Tujuan dari optimasi model adalah untuk meningkatkan akurasi dan efisiensi model. Diagram alur model yang akan dibangun dapat dilihat pada Gambar 3 dibawah.



Gambar 3. Diagram Alur Model

Diagram alur model yang terdapat pada Gambar 3 terdiri dari beberapa lapisan yang dimulai dengan lapisan input. Lapisan input (*sequential\_input*) menerima citra dengan ukuran 224x224 piksel dan 3 saluran warna (RGB). Setelah itu, data citra akan diaugmentasi pada lapisan selanjutnya menggunakan lapisan augmentasi yang telah disiapkan pada tahapan *data preparation*.

Selanjutnya adalah lapisan ekstraksi fitur yang dimulai dengan lapisan yang memuat arsitektur *MobileNetV3large*. Output dari lapisan *MobileNetV3large* adalah tensor dengan dimensi (7, 7, 960), yang menunjukkan bahwa ada 960 fitur yang diekstraksi dari data yang masuk. Setelah itu, lapisan konvolusi (*conv2d*) mengubah dimensi output menjadi (7, 7, 32), mengurangi jumlah fitur dari 960 menjadi 32 guna mencegah terjadinya *overfitting*[11]. Kemudian, lapisan *max pooling* (*max\_pooling2d*) digunakan untuk mengurangi dimensi spasial dari output konvolusi menjadi (3, 3, 32). Untuk mengurangi *overfitting*, lapisan *dropout* diterapkan guna mematikan 10% dari keseluruhan neuron secara acak selama pelatihan[12]. Langkah selanjutnya adalah lapisan *global average pooling* (*global\_average\_pooling2d*) yang meratakan dimensi spasial dari tensor menjadi (32), sehingga hanya fitur-fitur penting yang tersisa.

Terakhir, lapisan *fully connected* pertama (*dense*) dengan 128 neuron yang menerima input dari lapisan ekstraksi fitur dan diikuti dengan lapisan *fully connected* kedua (*dense\_1*) dengan 4 neuron, yang merupakan jumlah kelas dalam klasifikasi. Output akhir dari model ini adalah probabilitas kelas untuk setiap input citra.

### 3. Hasil

Setelah melakukan pembangunan model sesuai dengan diagram alur model pada Gambar 3 diatas, proses selanjutnya adalah optimasi parameter model. Proses optimasi parameter model dilakukan untuk meningkatkan performa model yang telah dibangun sebelumnya berdasarkan nilai akurasi dan nilai loss. Nilai akurasi adalah nilai ketepatan prediksi model dengan label sebenarnya sedangkan nilai loss adalah suatu metrik yang menunjukkan seberapa besar perbedaan antara hasil prediksi model dengan nilai sebenarnya, sehingga semakin kecil nilai loss maka semakin baik model tersebut[13]. Parameter-parameter yang akan dioptimasi meliputi jumlah epoch, jumlah batch size, dan jenis optimizer. Selanjutnya model yang telah dioptimasi akan diintegrasikan ke dalam suatu aplikasi mobile berbasis Android untuk memudahkan para petani dalam menggunakan model tersebut.

#### 3.1 Optimasi Model

Eksperimen pertama yang akan dilakukan adalah dengan melatih model dengan tiga macam jumlah epoch dan batch size. Epoch sendiri merupakan suatu istilah yang merujuk pada proses dimana model mengolah seluruh dataset dalam satu siklus. Sedangkan batch size adalah jumlah kelompok data yang akan diproses oleh model[14]. Hasil dari eksperimen ini dapat dilihat pada Tabel 1 dibawah ini.

Tabel 1. Hasil Eksperimen Jumlah Epoch dan Batch Size

Batch	15		30		45	
Epoch	Training	Validasi	Training	Validasi	Training	Validasi
10	Acc:70%	Acc:70%	Acc:69%	Acc:71%	Acc:67%	Acc:69%
	Los:70%	Los:70%	Los:77%	Los:71%	Los:81%	Los:73%
20	Acc:71%	Acc:76%	Acc:71%	Acc:73%	Acc:72%	Acc:71%
	Los:73%	Los:62%	Los:70%	Los:67%	Los:69%	Los:69%
30	Acc:73%	Acc:71%	Acc:73%	Acc:74%	<b>Acc:74%</b>	<b>Acc:76%</b>
	Los:68%	Los:67%	Los:68%	Los:63%	<b>Los:66%</b>	<b>Los:59%</b>

Setelah melakukan eksperimen pertama, penulis mendapatkan nilai akurasi dan nilai *loss* terbaik yaitu pada jumlah *batch size* sebesar 45 dan jumlah epoch sebesar 30 baik untuk data training maupun data testing. Dengan begitu dapat disimpulkan bahwa semakin besar jumlah epoch dan batch size maka semakin baik pula nilai akurasi dan nilai loss yang dihasilkan oleh model untuk studi kasus ini.

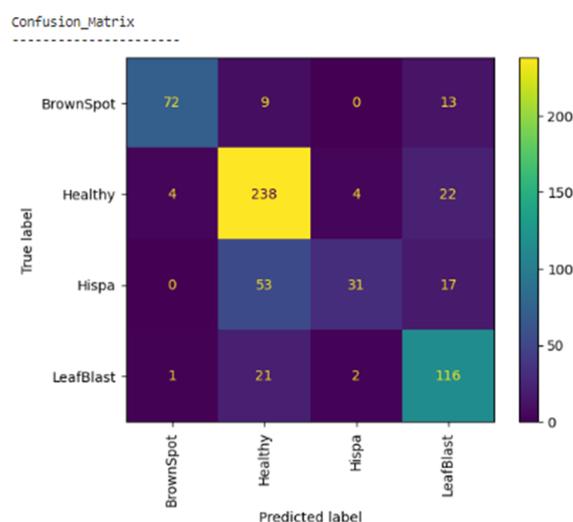
Eksperimen yang akan dilakukan selanjutnya adalah melakukan *compile* model dengan berbagai macam *optimizer*. *Optimizer* adalah algoritma atau metode yang digunakan untuk mengubah atribut *neural networks*, seperti *weights* dan *learning rate*, untuk mengurangi nilai *loss*. *Optimizer* bertujuan untuk meminimalkan (atau, dalam beberapa kasus, memaksimalkan) fungsi *loss*, yang merupakan ukuran seberapa jauh perbedaan prediksi dari hasil sebenarnya[15]. Pada eksperimen ini, penulis akan menggunakan *optimizer* variant Adam, RMSprop, dan Lion untuk mengetahui *optimizer* terbaik untuk model yang telah dibangun. Eksperimen ini dilakukan dengan menggunakan jumlah epoch dan jumlah batch size terbaik yang sudah didapatkan pada eksperimen sebelumnya. Hasil dari eksperimen ini dapat dilihat pada Tabel 2 dibawah.

Tabel 2. Hasil Eksperimen Jenis Optimizer

Optimizer Epoch, Batch Size	Adam		RMSprop		Lion	
	Training	Validasi	Training	Validasi	Training	Validasi
(30,45)	Acc:74%	Acc:76%	Acc:73%	Acc:74%	<b>Acc:76%</b>	<b>Acc:75%</b>
	Los:66%	Los:59%	Los:69%	Los:62%	<b>Los:61%</b>	<b>Los:59%</b>

Pada Tabel 2 dapat dilihat bahwa *optimizer* Lion memiliki kinerja yang lebih baik dibandingkan *optimizer* Adam dan *optimizer* RMSprop. Sehingga dapat disimpulkan bahwa algoritma milik *optimizer* Lion yang akan mengalami peningkatan kinerja seiring dengan jumlah *batch size* yang sudah ditentukan sebelumnya[16].

Tahap ini dilakukan untuk mengevaluasi performa dari model yang telah berhasil dibangun dan dioptimasi. Dalam melakukan evaluasi model, penulis telah menyiapkan *data testing* sebanyak 608 gambar yang terdiri dari 94 gambar untuk kategori *brownspot*, 268 gambar untuk kategori *healthy*, 101 gambar untuk kategori *hispa*, dan 140 gambar dari kategori *leafblast*. Proses evaluasi menggunakan metode *Confussion Matrix* untuk mengetahui bagaimana model bekerja pada seluruh dataset. *Confussion Matrix* adalah alat evaluasi yang digunakan untuk mengukur kinerja model machine learning[17]. Serta dilakukan perhitungan nilai Recall, Precissions, dan F1-Score dari hasil *Confussion Matrix* untuk memahami kualitas model dengan lebih mendalam dan objektif. Nilai F1-Score menggabungkan Precision dan Recall untuk menghasilkan skor tunggal yang mewakili kinerja keseluruhan model maupun kelas tertentu[18]. Sementara itu, Recall mengukur seberapa banyak data positif yang berhasil diklasifikasikan dengan benar, dan Precision mengukur seberapa akurat model dalam melakukan prediksi positive[19]. Hasil dari pengujian menggunakan metode *Confussion Matrix* pada model dapat dilihat pada Gambar 4 di bawah ini.



Gambar 4. Confusion Matrix

Pada Gambar 4, terlihat bahwa model secara mayoritas dapat memprediksi jenis penyakit pada daun padi dengan benar. Untuk membuktikannya dengan lebih kuat dan mendalam, berikut adalah perhitungan nilai Recall, F1-Score, dan Support pada masing-masing kategori dan pada keseluruhan hasil.

1. Kategori *Brownsport*:

- True Positives (TP) = 72
- False Negatives (FN) = 9 + 0 + 13 = 22
- False Positives (FP) = 4 + 0 + 1 = 5
- Support = TP + FN = 72 + 22 = 94

$$Recall = \frac{TP}{TP + FN} = \frac{72}{72 + 22} = 0,766 \tag{1}$$

Nilai Recall: 0.766 yang didapatkan dari perhitungan 1 menunjukkan bahwa model ini cukup baik dalam mendeteksi kategori 'BrownSpot', dengan lebih dari 76% dari kasus aktual terdeteksi dengan benar.

$$Precision = \frac{TP}{TP + FP} = \frac{72}{72 + 5} = 0,935 \tag{2}$$

Nilai Precision: 0.935 yang didapatkan dari perhitungan 2 menunjukkan bahwa sebagian besar prediksi 'BrownSpot' benar-benar adalah 'BrownSpot', nilai yang sangat bagus untuk kelas ini.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0,843 \tag{3}$$

Nilai F1 Score: 0.843 yang didapatkan dari perhitungan 3 mengindikasikan keseimbangan yang baik antara *precision* dan *recall* pada kategori *Brownsport*.

2. Kategori *Healthy*:

- True Positives (TP) = 238
- False Negatives (FN) = 4 + 4 + 22 = 30
- False Positives (FP) = 9 + 53 + 21 = 83
- Support = TP + FN = 238 + 30 = 268

$$Recall = \frac{TP}{TP + FN} = \frac{238}{238 + 30} = 0,888 \tag{4}$$

Nilai Recall: 0.888 yang didapatkan dari perhitungan 4 menunjukkan performa model yang sangat baik dalam mendeteksi kategori 'Healthy'.

$$Precision = \frac{TP}{TP + FP} = \frac{238}{238 + 83} = 0,741 \tag{5}$$

Nilai Precision: 0.741 yang didapatkan dari perhitungan 5 menunjukkan bahwa sebagian besar prediksi 'Healthy' benar-benar adalah 'Healthy', nilai yang cukup bagus meskipun lebih rendah dari kategori *Brownsport*.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0,808 \tag{6}$$

Nilai F1 Score: 0.808 yang didapatkan dari perhitungan 6 mengindikasikan keseimbangan yang baik antara *precision* dan *recall* pada kategori *Healthy*.

3. Kategori *Hispa*

- True Positives (TP) = 31
- False Negatives (FN) = 0 + 53 + 17 = 70
- False Positives (FP) = 0 + 4 + 2 = 6
- Support = TP + FN = 31 + 70 = 101

$$Recall = \frac{TP}{TP + FN} = \frac{31}{31 + 70} = 0,307 \tag{7}$$

Nilai Recall: 0.307 yang didapatkan dari perhitungan 7 menunjukkan performa model yang kurang baik dalam mendeteksi kategori *Hispa*.

$$Precision = \frac{TP}{TP + FP} = \frac{31}{31 + 6} = 0,838 \quad (8)$$

Nilai Precision: 0.838 yang didapatkan dari perhitungan 8 menunjukkan bahwa prediksi yang diberikan oleh model pada kategori Hispa cukup tepat.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0,45 \quad (9)$$

Nilai F1 Score: 0.45 yang didapatkan dari perhitungan 9 menunjukkan kesulitan model dalam menyeimbangkan precision dan recall untuk kategori Hispa. Hal ini disebabkan oleh data citra untuk kategori Hispa yang tidak terlalu memiliki ciri khusus sehingga model cukup kesulitan dalam meng-ekstraksi fitur.

#### 4. Kategori *Leafblast*

- True Positives (TP) = 116
- False Negatives (FN) = 1 + 21 + 2 = 24
- False Positives (FP) = 13 + 22 + 17 = 52
- Support = TP + FN = 116 + 24 = 140

$$Recall = \frac{TP}{TP + FN} = \frac{116}{116 + 24} = 0,829 \quad (10)$$

Nilai Recall: 0.829 yang didapatkan dari perhitungan 10 menunjukkan performa model yang sangat baik dalam mendeteksi kategori *Leafblast*.

$$Precision = \frac{TP}{TP + FP} = \frac{116}{116 + 52} = 0,69 \quad (11)$$

Nilai Precision: 0.69 yang didapatkan dari perhitungan 11 menunjukkan bahwa prediksi yang diberikan oleh model pada kategori *Leafblast* cukup tepat namun masih terdapat kesalahan prediksi.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0,754 \quad (12)$$

Nilai F1 Score: 0.754 yang didapatkan dari perhitungan 12 menunjukkan mengindikasikan keseimbangan yang cukup baik antara precision dan recall pada kategori *Leafblast*.

#### 5. Macro Average F1-Score

$$\frac{1}{n} \sum_{i=1}^{n=4} F1_i = 0,704 \quad (13)$$

Nilai *Macro-Averaged* F1 Score: 0.714 yang didapatkan dari perhitungan 13 menunjukkan bahwa, secara rata-rata, model memiliki performa yang cukup baik di berbagai kategori.

#### 6. Weighted Average F1-Score

$$\frac{1}{n} \sum_{i=1}^{n=4} Support_i \times F1_i = 0,746 \quad (14)$$

Nilai *Weighted-Averaged* F1 Score: 0.746 yang didapatkan dari perhitungan 14 menunjukkan bahwa, mempertimbangkan distribusi kategori, model ini cukup baik secara keseluruhan.

#### 7. Nilai Akurasi Model Pada Data Testing

$$Akurasi = \frac{Total\ Prediksi\ Benar\ (TP)}{Total\ Dataset} = \frac{76 + 238 + 31 + 116}{603} = \frac{457}{603} = 0,75 \quad (15)$$

### 3.2 Implementasi

Setelah mendapatkan model dengan performa yang cukup baik, tahapan selanjutnya adalah meng-integrasikan model kedalam suatu aplikasi mobile berbasis Android. Untuk dapat meng-integrasikannya, model yang telah dibangun perlu dikonversi terlebih dahulu kedalam format .tflite. TFLite dirancang untuk memberikan inferensi yang cepat dan efisien, sehingga memungkinkan aplikasi berbasis AI berjalan secara efektif di perangkat dengan sumber daya terbatas[20]. Dalam membangun aplikasi mobile berbasis Android yang dapat menjalankan model CNN yang telah dibangun untuk mengklasifikasikan jenis penyakit pada daun padi, penulis menggunakan Algoritma 1 seperti dibawah ini.

---

#### Algoritma 1 Implementasi Model CNN pada Aplikasi Android

---

```

Inisialisasi model CNN.
Deklarasi variabel gambar.
Aplikasi menerima input data gambar dari kamera atau galeri
Merubah data gambar kedalam bentuk Bitmap
Ubah ukuran gambar menjadi 224x224 piksel
Deklarasi variabel buffer byte untuk input model
Iterasi setiap piksel gambar untuk mengekstraksi nilai RGB
for each pixel in gambar do:
    int val = getPixelValue(pixel)
    byteBuffer.putFloat((val » 16) & 0xFF) // R
    byteBuffer.putFloat((val » 8) & 0xFF) // G
    byteBuffer.putFloat(val & 0xFF) // B
end for
Deklarasikan variable untuk input tensor model
Muat buffer byte ke dalam tensor input model
Model memproses data input tensor dan dapatkan nilai prediksi untuk setiap kelas
Menentukan kelas yang dengan nilai prediksi tertinggi
if kelas tertinggi == Hispa then:
    Menampilkan data gambar,
    Menampilkan tombol menuju detail penyakit Hispa
else if Kelas tertinggi == Leafblast then:
    Menampilkan data gambar,
    Menampilkan tombol menuju detail penyakit Leafblast
else if kelas tertinggi == Brownspot then:
    Menampilkan data gambar,
    Menampilkan tombol menuju detail penyakit Brownspot
else if kelas tertinggi == Healthy then:
    Menampilkan data gambar,
    Menampilkan pesan kondisi tanaman sehat
end if
Ulangi Langkah 2-9 untuk memuat input data gambar baru.

```

---

Ulangi Langkah 2-9 untuk memuat input data gambar baru

Hasil dari implementasi Algoritma 1 dapat dilihat melalui video pada link berikut [https://youtube.com/shorts/\\_4BFiCva8fU?feature=share](https://youtube.com/shorts/_4BFiCva8fU?feature=share) atau dapat dilihat pada Gambar.



Gambar 5. Hasil Implementasi

#### 4. Pembahasan

Hasil penelitian ini menunjukkan bahwa model CNN yang dioptimasi memiliki performa yang cukup baik dalam mendeteksi penyakit pada daun padi, dengan nilai akurasi, *Precision*, *Recall*, dan *F1-score* yang signifikan. Model yang diimplementasikan pada aplikasi Android juga berhasil mengklasifikasikan gambar daun padi ke dalam beberapa kategori yaitu BrownSpot, Healthy, Hispa, dan LeafBlast. Hasil pengujian dengan confusion matrix menunjukkan bahwa model ini sangat efektif dalam mendeteksi penyakit BrownSpot dan kondisi Healthy, meskipun terdapat beberapa kesalahan klasifikasi pada kelas Hispa dan LeafBlast. Temuan tak terduga terjadi pada penelitian ini dimana hasil akurasi deteksi yang lebih rendah dari yang diharapkan pada kelas Hispa, yang dapat disebabkan oleh *data testing* untuk kategori Hispa yang tidak terlalu memiliki ciri khusus sehingga model cukup kesulitan dalam meng-ekstraksi fitur yang dimana pernyataan ini selaras dengan yang dikatakan oleh Musthafa[21] bahwa tumbuhan padi yang terjangkit hispa biasanya hanya dapat dikenali dengan adanya bercak atau titik-titik putih akibat serangan serangga dewasa yang mengikis permukaan daun. Oleh karena itu model yang telah dikembangkan kerap memprediksi kondisi hispa sebagai healthy karena tidak adanya perubahan pada daun secara signifikan. Hasil akurasi deteksi kategori Hispa yang rendah pula mempengaruhi nilai akurasi model secara keseluruhan sehingga menghasilkan nilai akurasi sebesar 75%.

Dalam membandingkan penelitian ini dengan penelitian sebelumnya, implementasi model CNN ke dalam aplikasi Android adalah keunggulan utama yang dimana proses implementasi ini belum dilakukan pada penelitian sebelumnya. Penelitian sebelumnya lebih banyak berfokus pada pengembangan dan optimasi

model, tanpa mempertimbangkan aplikasi praktis di lapangan. Dengan implementasi, penelitian ini menawarkan solusi yang lebih *user-friendly* dan praktis bagi para petani. Aplikasi yang sudah dikembangkan akan membantu petani dalam mengetahui kondisi Kesehatan pada tanaman padi mereka berdasarkan citra daun padi yang sudah di-input-kan ke dalam Aplikasi. Selain dapat mengetahui kondisi Kesehatan tanaman padi mereka, para petani pun dapat mengetahui penjelasan, gejala, serta pengendalian atas penyakit yang menyerang tanaman padi mereka. Permasalahan mengenai serangan hama dan penyakit pada tanaman padi yang dapat menurunkan angka produksi padi seperti yang telah dijelaskan sebelumnya dapat diselesaikan dengan adanya aplikasi dari hasil penelitian ini.

Penelitian ini berhasil menguji hipotesis bahwa optimasi dan implementasi model CNN pada aplikasi mobile dapat dilakukan untuk memudahkan proses deteksi penyakit pada daun padi untuk meningkatkan angka produksi padi di Indonesia. Dimana pada penelitian ini memberikan solusi praktis bagi petani untuk mendeteksi penyakit tanaman secara cepat dan akurat. Namun, terdapat keterbatasan pada penelitian ini, seperti pembahasan mengenai bagaimana model ini dapat beradaptasi pada kondisi lingkungan yang berbeda dari data yang digunakan untuk melatih model. Untuk itu, potensi penelitian di masa depan dapat dilakukan seperti pengembangan lebih lanjut dari model agar model dapat mendeteksi penyakit dengan kondisi lingkungan yang beragam dan eksplorasi implementasi teknologi serupa pada berbagai jenis tanaman dan penyakit .

## 5. Simpulan

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa model yang telah dibangun dan dioptimasi dengan parameter terbaik, yaitu 30 epoch, batch size 45, dan optimizer Lion, menghasilkan akurasi 75% untuk data testing dengan nilai loss sebesar 59%, serta akurasi 76% untuk data training dengan nilai loss sebesar 61%. Model ini juga telah berhasil diintegrasikan ke dalam aplikasi berbasis mobile dengan sistem operasi Android dan dapat bekerja dengan baik tanpa adanya hambatan.

Sebagai tindak lanjut, penelitian ini dapat dikembangkan lebih lanjut dengan meningkatkan jumlah data yang digunakan serta kategori penyakit yang lebih beragam untuk meningkatkan akurasi model. Selain itu, pengembangan model lebih lanjut juga dapat dilakukan untuk memastikan performanya tetap stabil di berbagai kondisi lingkungan. Penelitian ini diharapkan dapat menjadi dasar bagi pengembangan alat deteksi penyakit padi yang lebih efisien dan efektif, serta dapat diintegrasikan dengan sistem manajemen pertanian berbasis IoT untuk pemantauan tanaman secara *real-time*.

## Pustaka

- [1] H. Herdiyanti and E. Sulistyono, "Pertumbuhan dan produksi beberapa varietas padi (*oryza sativa* L.) pada berbagai interval irigasi," *Indonesian Journal of Agronomy*, vol. 49, no. 2, pp. 129–135, 2021.
- [2] B. P. Statistik, "Luas panen dan produksi padi di indonesia 2023 (angka sementara)," *Berita Resmi Statistik*, Oct. 2023.
- [3] E. H. D. M. Saputra and S. Agustian, "Klasifikasi kesehatan pada tanaman padi menggunakan citra unmaned aerial vehicle (uav) dengan metode convolutional neural networks (cnn)," *Jurnal Ilmiah Teknologi Infomasi Terapan (JITTER)*, vol. 9, no. 3, 2023.
- [4] A. J. M. Khoiruddin and W. A. Saputra, "Klasifikasi penyakit daun padi menggunakan convolutional neural network," *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, vol. 2, no. 1, pp. 37–45, 2022.
- [5] M. R. A. F. H. Hawari, F. Fadillah and T. Arifin, "Klasifikasi penyakit tanaman padi menggunakan algoritma cnn (convolutional neural network)," *Jurnal Responsif: Riset Sains dan Informatika*, vol. 4, no. 2, pp. 184–189, 2022.
- [6] A. B. S. F. R. P. S. Sheila, M. K. Anwar and I. P. Sari, "Deteksi penyakit pada daun padi berbasis pengolahan citra menggunakan metode convolutional neural network (cnn)," *J. Multinetics*, vol. 9, no. 1, pp. 27–34, 2023.

- [7] 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*, pp. 37–44, 2022.
- [8] H. O. P. Sitompul and A. Prasetyo, "Identification of rice plant diseases through leaf image using densenet 201: Identifikasi penyakit tanaman padi melalui citra daun menggunakan densenet 201," *JOMLAI: Journal of Machine Learning and Artificial Intelligence*, vol. 1, no. 2, pp. 143–150, 2022.
- [9] M. Z. S. G. J. Z. S. Yang, W. Xiao and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv preprint arXiv:2204.08610*, 2022.
- [10] C. N. S. Qian and Y. Hu, "Mobilenetv3 for image classification," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE, 2021, pp. 490–497.
- [11] S. R. I. A. E. ALBAKIA and R. A. Saputra, "Identifikasi jenis daun tanaman obat menggunakan metode convolutional neural network (cnn) dengan model vgg16," *Jurnal Informatika Polinema*, vol. 9, no. 4, pp. 451–460, 2023.
- [12] A. K. I. S. N. Srivastava, G. Hinton and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [13] J. C. A. Demirkaya and S. Oymak, "Exploring the role of loss functions in multiclass classification," in *2020 54th annual conference on information sciences and systems (ciss)*. IEEE, 2020, pp. 1–5.
- [14] Y. A. Suwitono and F. J. Kaunang, "Implementasi algoritma convolutional neural network (cnn) untuk klasifikasi daun dengan metode data mining semma menggunakan keras," *Jurnal Komtika (Komputasi dan Informatika)*, vol. 6, no. 2, pp. 109–121, 2022.
- [15] P. E. Gomedé, "Exploring the landscape of optimizers in keras," Medium, diakses pada 8 Maret 2024. [Online]. Available: <https://medium.com/aimonks/exploring-the-landscape-of-optimizers-in-keras-c7f1b13ddb63>
- [16] X. C. et al., "Symbolic discovery of optimization algorithms," *Adv Neural InfProcess Syst*, vol. 36, 2024.
- [17] A. S. S. P. A. H. Pratama, R. Teguh and V. Wilentine, "Deteksi covid-19 berdasarkan hasil rontgen dada (chest xray) menggunakan python," *Journal of Information Technology and Computer Science*, vol. 1, no. 1, pp. 58–67, 2021.
- [18] P. K. S. K. D. G. M. M. Ahsan, M. A. P. Mahmud and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies (Basel)*, vol. 9, no. 3, p. 52, 2021.
- [19] G. A. Panharsi and I. G. Anugrah, "Klasifikasi waktu penyelesaian skripsi mahasiswa menggunakan metode weighted naive bayes (studi kasus: Program studi teknik informatika universitas muhammadiyah gresik)," 2022.
- [20] M. R. P. H. Pratikno and Y. Triwidyastuti, "Pengenalan gestur jari tangan sebagai media pembelajaran berhitung bagi paud berbasis visi komputer dan deep learning: Pengenalan gestur jari tangan berbasis visi komputer dan deep learning," *Journal of Computer, Electronic, and Telecommunication*, vol. 4, no. 1, 2023.
- [21] A. Musthafa and O. V. Putra, "Deteksi penyakit pada daun tanaman padi menggunakan metode convolutional neural network," in *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2021, pp. 103–109.