

# Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network*

1<sup>st</sup> Savira Anggita Sabrina  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

saviraanggita@students.telkomuniversity.ac.id

2<sup>nd</sup> Wikky Fawwaz Al Maki  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

wikkyfawwaz@telkomuniversity.ac.id

## Abstrak

Tanaman kopi merupakan salah satu komoditas perkebunan andalan di Indonesia, dengan sebagian besar hasil produksinya berasal dari tanaman kopi robusta. Salah satu permasalahan dalam produksi kopi robusta, yang dapat menyebabkan kerugian yang cukup signifikan, adalah rendahnya mutu hasil panen tanaman kopi robusta akibat serangan hama dan penyakit. Proses penanganan hama dan penyakit pada tanaman kopi robusta memerlukan sumber daya yang tidak sedikit. Terdapat beberapa cara untuk mengatasi permasalahan tersebut, salah satunya adalah dengan membangun sistem yang dapat melakukan identifikasi terhadap hama dan penyakit pada tanaman kopi robusta. Penelitian ini menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *EfficientNet-B0* untuk menghasilkan sistem klasifikasi penyakit pada citra daun tanaman kopi robusta. Teknik augmentasi data dan pendekatan *fine-tuning* diterapkan untuk membantu menghasilkan sistem dengan performansi optimal. Evaluasi performansi dilakukan pada dua algoritma optimasi berbeda, yaitu Adam dan *Root Mean Square Propagation* (RMSProp). Kedua algoritma memperoleh hasil terbaik yang sama dengan nilai akurasi sebesar 91% dan *F1-Score* secara keseluruhan sebesar 91%.

**Kata kunci :** tanaman kopi robusta, CNN, Adam, RMSProp

## Abstract

*Coffee plants are one of the mainstay plantation commodities in Indonesia, with most of its production coming from robusta coffee plants. One of the problems in robusta coffee production, which can cause significant losses, is the low quality of robusta coffee yields due to pests and diseases. The process of handling pests and diseases on robusta coffee plants requires a lot of resources. There are several ways to overcome these problems, one of which is to build a system that is able to identify pests and diseases on robusta coffee plants. This study used Convolutional Neural Network (CNN) with EfficientNet-B0 architecture to produce a disease classification system in robusta coffee leaf imagery. Data augmentation technique and fine-tuning approach were applied to help produce a system with optimal performance. Performance evaluation was carried out on two different optimization algorithms, namely Adam and Root Mean Square Propagation (RMSProp). Both algorithms obtained the same best results with an accuracy value of 91% and an overall F1-Score of 91%.*

**Keywords:** robusta coffee plant, CNN, Adam, RMSProp

## I. PENDAHULUAN

### A. Latar Belakang

Penyakit pada tanaman kopi robusta dapat disebabkan oleh beberapa organisme seperti hama, jamur, bakteri, dan nematoda [1]. Perkembangan organisme-organisme tersebut dipengaruhi oleh kondisi lingkungan seperti suhu, curah hujan, kelembapan udara, dan sinar matahari [1]. Bagian tanaman kopi yang cenderung terserang penyakit terdapat pada bagian akar, batang, daun, dan buah.

Salah satu cara termudah untuk mengidentifikasi penyakit pada tanaman kopi robusta adalah dengan mengamati kondisi permukaan daun. Jenis penyakit umum yang dapat diidentifikasi melalui kondisi permukaan daunnya adalah penyakit

karat daun. Penyakit ini dapat menyerang berbagai jenis tanaman kopi seperti robusta, arabika, dan liberika. Penyakit ini dapat menyebabkan penurunan hasil panen tanaman kopi hingga 70% [1]. Selain karat daun, keberadaan tungau laba-laba merah pada tanaman kopi robusta juga dapat diketahui dengan memperhatikan kondisi permukaan daun. Tanpa penanganan yang tepat, tungau ini dapat mengakibatkan kerusakan daun yang cukup parah hingga kematian tanaman [2].

Untuk mengatasi kerugian dan mencegah perkembangan penyakit, proses identifikasi penyakit pada tanaman kopi robusta perlu dilakukan sejak dini. Hal ini dapat dilakukan dengan membangun sebuah sistem yang menerima input berupa citra daun yang terserang penyakit. Salah satu metode yang telah terbukti menghasilkan performansi yang optimal

pada kasus serupa adalah *Convolutional Neural Network* (CNN). Metode CNN telah digunakan untuk mengklasifikasi penyakit pada tanaman kopi jenis arabika dalam penelitian [3] yang menghasilkan nilai akurasi terbaik sebesar 97,07%. Penelitian [4] juga menggunakan CNN untuk mengklasifikasi penyakit pada tanaman kopi liberika dan menghasilkan akurasi sebesar 97,20%.

Dalam penggunaan CNN, diperlukan jumlah data yang banyak dan sumber daya komputer yang besar. Data yang tersebar di dunia nyata jumlahnya terbatas dan seringkali tidak sesuai kebutuhan, terutama untuk melakukan identifikasi penyakit seperti karat daun dan hama tungau laba-laba merah. Hal ini membuat dataset yang terkumpul memiliki jumlah citra yang sedikit dan distribusi data yang tidak seimbang atau *imbalance*. Beberapa penelitian telah dilakukan untuk menangani masalah data yang *imbalance*, salah satunya adalah penelitian [5] yang menggunakan teknik augmentasi data pada dataset *imbalance* untuk mengidentifikasi penyakit pada tanaman timun dan menghasilkan nilai akurasi terbaik sebesar 94% dengan menggunakan CNN.

Dalam kurun waktu 5 tahun terakhir, sistem yang telah dibangun untuk identifikasi penyakit tanaman kopi di Indonesia masih menggunakan data berbentuk tabular. Beberapa sistem yang dibangun adalah sistem pakar berbentuk aplikasi web [6], [7], dan belum ditemukan sistem yang dapat mengidentifikasi penyakit melalui citra daun khusus untuk tanaman kopi robusta. Selain itu, dataset citra daun kopi robusta juga belum ditemukan di Indonesia sehingga dataset yang digunakan perlu diambil dari sumber lain yang sudah ada, sebagai contoh adalah dataset citra daun kopi robusta yang diambil di perkebunan kopi di Ekuador [8].

Berdasarkan penelitian yang telah dilakukan sebelumnya, metode CNN dapat menghasilkan performansi sistem yang sangat baik dan cukup menjanjikan sehingga penelitian ini memutuskan untuk menggunakan CNN dalam mengklasifikasi penyakit pada tanaman kopi robusta. Teknik augmentasi data juga diterapkan untuk membuktikan apakah peningkatan jumlah data dan penyetaraan distribusi data berpengaruh terhadap peningkatan performansi sistem.

## B. Topik dan Batasannya

Dari latar belakang permasalahan yang telah diuraikan, permasalahan yang diangkat dalam penelitian ini adalah bagaimana membangun sistem klasifikasi CNN untuk mengklasifikasi penyakit pada tanaman kopi robusta dengan bantuan augmentasi data pada dataset yang tidak seimbang. Adapun batasan permasalahan pada penelitian ini, antara lain:

- Dataset yang digunakan berasal dari negara lain dengan jumlah 1410 citra daun yang terdiri dari 6 kelas penyakit dengan distribusi data yang bersifat *imbalance*.

- Arsitektur CNN yang digunakan adalah EfficientNet-B0.
- Algoritma optimasi yang digunakan sebagai perbandingan performansi adalah Adam dan *Root Mean Square Propagation* (RMSProp).

## C. Tujuan

Penelitian ini dilakukan dengan tujuan utama untuk membangun sistem klasifikasi menggunakan CNN dengan bantuan augmentasi data pada dataset yang bersifat *imbalance* untuk mengklasifikasi penyakit pada tanaman kopi robusta.

## D. Organisasi Tulisan

Penulisan laporan Tugas Akhir ini terbagi menjadi lima bagian, yaitu bagian latar belakang dan rumusan permasalahan, bagian studi literatur dan landasan teori, bagian perancangan sistem, bagian hasil evaluasi dan analisis, serta bagian kesimpulan dan saran.

## II. KAJIAN TEORI

### A. Studi Literatur

Penelitian mengenai klasifikasi penyakit pada tanaman kopi menggunakan CNN telah dilakukan pada jenis kopi arabika dalam penelitian [3] dengan arsitektur AlexNet, GoogleNet, ResNet50, dan VGG16. Penelitian ini melakukan klasifikasi pada 3 dataset berbeda, yaitu data *Biotic Stress* dan *Severity* yang berjumlah 1685 citra daun dengan 5 kelas penyakit, serta data *Symptom* yang berjumlah 2722 data dengan 5 kelas penyakit. Klasifikasi dilakukan dengan bantuan augmentasi data dan algoritma optimasi *Stochastic Gradient Descent* (SGD). *Transfer learning* diterapkan dengan mengatur seluruh *layer* pada tiap arsitektur menjadi *trainable*. Performansi tertinggi dihasilkan oleh arsitektur ResNet50 pada data *Symptom* dengan nilai akurasi sebesar 97,07%. Performansi terendah dihasilkan oleh ResNet50 pada data *Severity* yang bersifat *imbalance* dengan nilai akurasi 78,57%.

Klasifikasi penyakit pada tanaman kopi jenis liberika juga dilakukan pada penelitian [4] menggunakan CNN dengan arsitektur ResNetV2-152, VGG16, dan Xception. Penelitian ini menggunakan 4667 data yang terdiri dari 4 kelas penyakit. Klasifikasi dilakukan dengan bantuan augmentasi data, *transfer learning*, dan algoritma optimasi SGD. Nilai performansi tertinggi diperoleh sebesar 97,20% pada arsitektur VGG16, sedangkan nilai performansi terendah diperoleh sebesar 90,83% pada arsitektur ResNetV2-152.

Klasifikasi penyakit pada daun tanaman timun dilakukan dengan bantuan augmentasi data pada penelitian [5]. Teknik augmentasi data dilakukan dengan rotasi dan *flip* untuk meningkatkan jumlah data yang semula berjumlah 1.184 citra daun menjadi 14.208 citra daun. Klasifikasi dilakukan dengan membandingkan arsitektur CNN AlexNet dengan

*Support Vector Machine (SVM)* dan *Random Forest*, dimana AlexNet memperoleh nilai akurasi terbaik sebesar 94%.

#### B. Penyakit pada Tanaman Kopi Robusta

Salah satu jenis penyakit yang umum menyerang tanaman kopi robusta adalah karat daun, yang disebabkan oleh jamur *Hemileia vastatrix B. et. Br* [1]. Penyakit ini dapat menyebabkan penurunan hasil panen tanaman kopi mulai dari 20% hingga 70% [1]. Regional International Organization for Plant Protection and Animal Health, atau biasa disebut sebagai OIRSA, menetapkan skala resmi untuk mengukur tingkat keparahan penyakit ini dengan membagi tingkat keparahan berdasarkan persentase ukuran area daun yang terinfeksi [8], seperti yang tercantum pada Tabel 1.

TABEL 1. Tingkat keparahan penyakit karat daun

Tingkat	Area yang terinfeksi
1	1 – 5%
2	6 – 20%
3	21 – 50%
4	>50%

Terdapat jenis penyakit lain yang dapat menyebabkan kerusakan daun dan mengganggu proses fotosintesis pada tanaman kopi robusta, penyakit ini disebabkan oleh hama tungau laba-laba merah *Tetranychus urticae* [2]. Dalam jumlah besar, tungau ini dapat mengakibatkan penurunan kemampuan fotosintesis tanaman kopi yang dapat berujung pada penurunan nutrisi, penghambatan perkembangan buah, hingga kematian tanaman [2].

#### C. Convolutional Neural Network

CNN merupakan salah satu algoritma *deep learning* yang umum digunakan untuk menyelesaikan masalah identifikasi penyakit tanaman dalam kurun waktu lima tahun terakhir. CNN dapat dikembangkan menjadi berbagai arsitektur berbeda, yang secara umum terdiri dari 3 lapisan atau *layer* utama, yaitu *convolution layer*, *pooling layer*, dan *fully-connected layer*.

##### a. Convolution Layer

Lapisan ini merupakan lapisan dasar yang berperan untuk melakukan ekstraksi fitur terhadap data input [9]. Ekstraksi fitur dilakukan melalui proses konvolusi yang melibatkan kumpulan *filter* dua dimensi dan fungsi aktivasi tertentu untuk menghasilkan *feature map*.

##### b. Pooling Layer

Lapisan ini digunakan untuk mengurangi dimensi *feature map* [9]. Perhitungan *pooling* terbagi menjadi dua, yaitu *max pooling* dan *average pooling*. *Max pooling* mengambil nilai maksimum dari *feature map*, sedangkan

*average pooling* mengambil nilai rata-rata dari *feature map*.

##### c. Fully-connected Layer

Lapisan ini melakukan klasifikasi secara linier dengan bantuan fungsi aktivasi *Softmax* yang menghitung probabilitas dari setiap nilai vektor input terhadap tiap kelas yang memungkinkan. Fungsi aktivasi *Softmax* dapat dilihat pada persamaan 1.

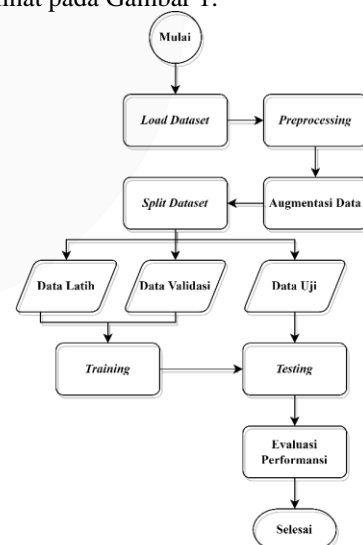
$$f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad (1)$$

Dimana  $f(x)_i$  menyatakan nilai fungsi aktivasi Softmax pada elemen vektor ke- $i$ ,  $x$  menyatakan nilai vektor input, dan  $k$  menyatakan banyak kelas.

### III. METODE

#### A. Gambaran Umum Sistem

Perancangan sistem yang dibangun diawali dengan *preprocessing* data citra daun kopi robusta, yang kemudian dilanjutkan dengan proses augmentasi data hingga total data berjumlah 4869 citra. Memasuki pembangunan sistem klasifikasi, proses terbagi menjadi tahap *training* dan *testing*. Kedua tahap tersebut melibatkan pembagian data input menjadi 3 bagian, yaitu data latih, data validasi, dan data uji. Data latih dan data uji masing-masing memiliki persentase sebesar 70% dan 15% dari total data pada dataset, sedangkan data validasi memiliki persentase sebesar 15% yang diambil dari total data pada data latih. Total data latih, data validasi, dan data uji secara berurutan berjumlah 3517, 621, dan 731 citra daun. Alur perancangan sistem secara umum dapat dilihat pada Gambar 1.



GAMBAR 1. Alur perancangan sistem

#### B. Dataset





Dataset yang digunakan pada penelitian ini merupakan dataset sekunder Bernama *RoCoLe* yang diperoleh dari situs resmi Mendeley. Data citra daun



diambil secara langsung menggunakan *smartphone* di perkebunan kopi yang terletak di Ekuador [8]. Citra daun pada dataset ini berjumlah 1560 data yang terdiri dari 6 kelas berbeda yang mewakili daun sehat, daun dengan tungau laba-laba merah, karat daun tingkat 1, karat daun tingkat 2, karat daun tingkat 3, dan karat daun tingkat 4. Detail distribusi data pada tiap kelas sebelum dan setelah preprocessing dapat dilihat pada Tabel 2.

### C. Preprocessing

*Preprocessing* dilakukan untuk memperoleh dataset dengan struktur data citra daun yang sesuai dengan kebutuhan sistem yang dibangun. Teknik yang diterapkan dalam *preprocessing* pada penelitian ini adalah *crop*, *resize*, dan *duplicate removal*. *Crop* dan *resize* dilakukan untuk menyamakan proporsi bentuk daun, sedangkan *duplicate removal* digunakan untuk menghapus data citra yang duplikat pada dataset. Detail distribusi data dan citra daun pada tiap kelas setelah preprocessing dapat dilihat pada Tabel 2.

TABEL 2. Distribusi dataset sebelum dan setelah *preprocessing*

Kelas	Jumlah Data Awal	Jumlah Data Hasil <i>Preprocessing</i>	Citra Daun Hasil <i>Preprocessing</i>
Sehat	791	780	
Tungau laba-laba merah	167	152	
Karat daun tingkat 1	344	291	
Karat daun tingkat 2	166	120	

Karat daun tingkat 3	62	51	
Karat daun tingkat 4	30	16	
Total	1560	1410	-

### D. Augmentasi Data

Augmentasi data dilakukan dengan mengubah *brightness* dan *contrast* dari citra daun secara acak, serta melakukan rotasi, translasi, dan *crop* secara acak. Distribusi data per kelas hasil augmentasi dapat dilihat pada Tabel 3.

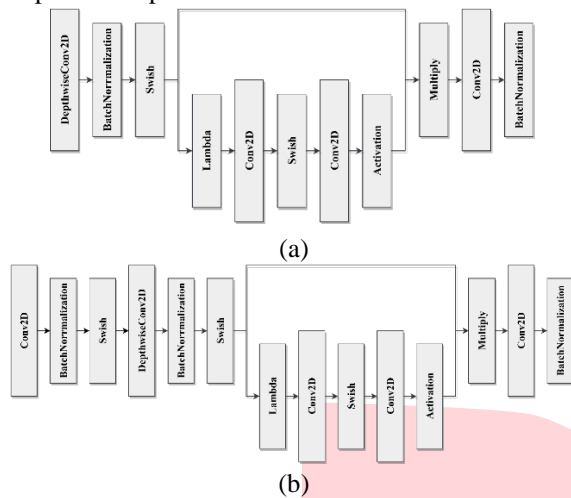
TABEL 3. Distribusi data dataset hasil augmentasi

Kelas	Jumlah Data Hasil Augmentasi
Sehat	780
Tungau laba-laba merah	760
Karat daun tingkat 1	873
Karat daun tingkat 2	840
Karat daun tingkat 3	816
Karat daun tingkat 4	800
Total	4869

### E. Model

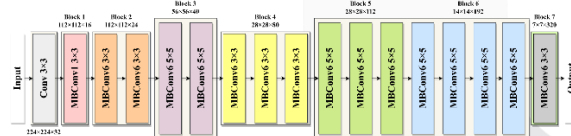
Sistem klasifikasi pada penelitian ini dibangun dengan arsitektur EfficientNet, tepatnya EfficientNet-B0. EfficientNet adalah salah satu arsitektur yang diperkenalkan pada tahun 2019 yang terdiri dari 8 bagian, mulai dari EfficientNet-B0 sampai dengan EfficientNet-B7. EfficientNet-B0 merupakan *baseline* dari EfficientNet yang dapat ditingkatkan menggunakan pendekatan *compound scaling* [10]. EfficientNet-B0 terdiri dari 7 blok yang masing-masing tersusun atas *inverted residual block* atau biasa disebut sebagai blok MBConv [10]. Terdapat 2 tipe blok MBConv yang digunakan pada arsitektur EfficientNet-B0, yaitu MBConv1 dan MBConv6. MBConv1 melakukan ekspansi dengan faktor 1 sedangkan MBConv6 melakukan ekspansi dengan faktor 6. Tiap blok MBConv tersusun atas lapisan *depthwise convolution* dan blok *squeeze-and-excitation* (SE). Lapisan *depthwise convolution* berfungsi untuk melakukan konvolusi secara mendalam pada tiap *channel*, didukung dengan blok SE yang berperan untuk memberikan bobot pada tiap *channel* [11]. Hal ini membuat arsitektur EfficientNet-B0 lebih baik karena tidak

memperlakukan tiap *channel* secara seragam [11]. Blok MBConv1 dan MBConv6 pada EfficientNet-B0 dapat dilihat pada Gambar 2.



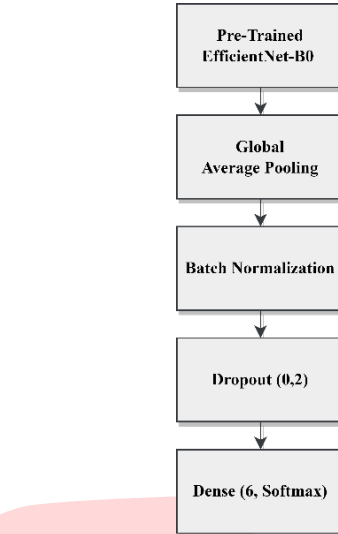
GAMBAR 2. Susunan blok MBConv dari model EfficientNet-B0 (a) MBConv1 (b) MBConv6

Arsitektur EfficientNet-B0 dipilih karena memiliki jumlah parameter yang paling sedikit sehingga memiliki waktu *training* yang lebih cepat dibandingkan arsitektur EfficientNet lainnya. Arsitektur ini juga terbukti memberikan hasil yang optimal dan lebih baik dari arsitektur CNN lain pada kasus serupa [12]. Selain itu, arsitektur ini juga belum pernah digunakan pada penelitian terkait identifikasi penyakit tanaman kopi yang sudah ada sebelumnya. Arsitektur EfficientNet-B0 secara umum dapat dilihat pada Gambar 3.



GAMBAR 3. Arsitektur EfficientNet-B0 secara umum

Pendekatan *fine-tuning* diterapkan pada model klasifikasi EfficientNet-B0 yang dibangun untuk mengurangi sumber daya komputasi yang dibutuhkan pada saat *training*. Penerapan *fine-tuning* pada penelitian ini dilakukan dengan mengatur lapisan-lapisan pada blok terakhir dari arsitektur EfficientNet-B0, yaitu blok ke-7, menjadi *trainable* dan mengubah susunan *output layer* seperti yang tercantum pada Gambar 4.



GAMBAR 4. Susunan *output layer* dari model EfficientNet-B0

IV. HASIL DAN PEMBAHASAN

Evaluasi performansi dilakukan dengan membandingkan algoritma optimasi Adam dan RMSProp pada data hasil augmentasi dan data tanpa augmentasi. Pada data tanpa augmentasi, perbandingan performansi dilakukan menggunakan sistem yang dilatih dari awal dan sistem yang telah dilatih menggunakan data hasil augmentasi.

Algoritma Adam merupakan algoritma yang bersifat adaptif terhadap *learning rate* yang dipilih karena terbukti dapat memberikan hasil optimal pada penelitian serupa [12]. Sedangkan RMSProp, salah satu algoritma adaptif yang merupakan ekstensi dari algoritma *gradient descent* [13], dipilih sebagai perbandingan terhadap algoritma Adam. Pengaturan parameter yang digunakan pada proses *training* untuk tiap algoritma optimasi dapat dilihat pada Tabel 4.

TABEL 4. Pengaturan parameter pada proses *training*

Algoritma Optimasi	Data Hasil Augmentasi			Data Tanpa Augmentasi		
	Epoch	Batch Size	Learning Rate	Epoch	Batch Size	Learning Rate
Adam	50	64	$1 \times 10^{-4}$	20	32	$1 \times 10^{-4}$
RMSProp	50	64	$1 \times 10^{-4}$	20	32	$1 \times 10^{-4}$

Pengukuran performansi dilakukan pada proses *testing* menggunakan metrik akurasi, presisi, *recall*, dan *F1-Score* yang diperoleh dari *Confusion Matrix*. Nilai akurasi, presisi, *recall*, dan *F1-Score* secara berurutan diperoleh dari persamaan 2, 3, 4, dan 5.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

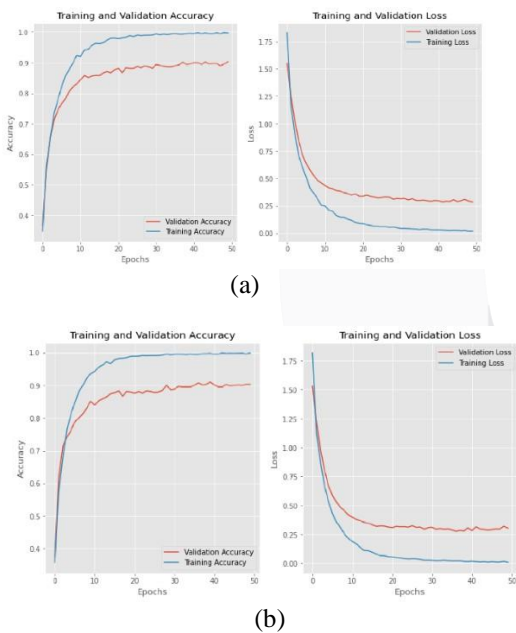
$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{(Precision + Recall)} \quad (5)$$

A. Hasil Pengujian

a. Data Hasil Augmentasi

Perbandingan performansi antara algoritma optimasi Adam dan RMSProp pada data hasil augmentasi menunjukkan bahwa algoritma Adam dan RMSProp memiliki nilai akurasi dan *F1-Score* yang sama, dengan nilai sebesar 91%. Namun, algoritma RMSProp memiliki nilai loss yang lebih rendah, yaitu sebesar 0,2862. Grafik proses *training* pada kedua algoritma dengan data hasil augmentasi terdapat pada Gambar 5.

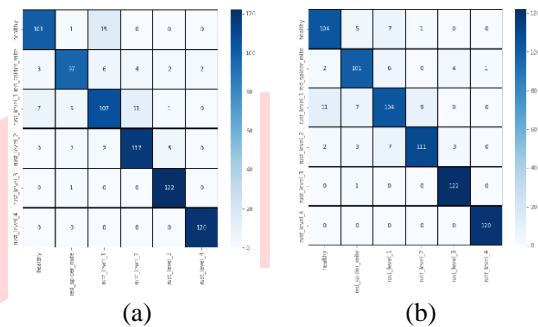


GAMBAR 5. Grafik proses *training* pada data hasil augmentasi (a) algoritma Adam (b) algoritma RMSProp

Hasil proses *testing* algoritma Adam dan RMSProp pada data hasil augmentasi diperoleh dari nilai *loss* dan akurasi secara keseluruhan, serta nilai *macro average* dari presisi, *recall*, dan *F1-Score* yang dapat dilihat pada Tabel 5. *Confusion Matrix* yang dihasilkan dari algoritma optimasi Adam dan RMSProp dengan data hasil augmentasi dapat dilihat pada Gambar 6.

TABEL 5. Hasil proses *testing* pada data hasil augmentasi

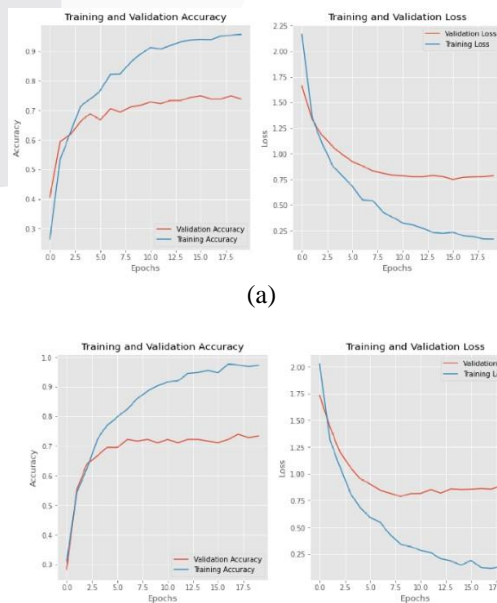
Algoritma Optimasi	Loss	Akurasi	Presi si	Recall	F1-Score
Adam	0,3014	91%	91%	91%	91%
RMSProp	0,2862	91%	91%	91%	91%



GAMBAR 6. *Confusion Matrix* pada data hasil augmentasi (a) algoritma Adam (b) algoritma RMSProp

b. Data tanpa Augmentasi

Pada data tanpa augmentasi dengan sistem yang dilatih dari awal, algoritma optimasi Adam memiliki performansi terbaik dengan nilai akurasi sebesar 75% dan *F1-Score* sebesar 47%. Sedangkan, RMSProp memiliki performansi yang lebih rendah dengan nilai akurasi sebesar 69% dan *F1-Score* sebesar 41%. Grafik proses *training* algoritma Adam dan RMSProp pada data tanpa augmentasi dengan sistem yang dilatih dari awal dapat dilihat pada Gambar 7.



(a)

(b)

GAMBAR 7. Grafik proses *training* pada data tanpa augmentasi dengan sistem yang dilatih dari awal (a) algoritma Adam (b) algoritma RMSProp

Hasil proses *testing* algoritma Adam dan RMSProp pada data tanpa augmentasi dengan sistem yang dilatih dari awal dapat dilihat pada Tabel 6. Sedangkan, hasil proses *testing* kedua algoritma pada tiap kelas dapat dilihat pada Tabel 7.

TABEL 6. Hasil proses *testing* pada data tanpa augmentasi dengan sistem yang dilatih dari awal

Algoritma Optimalisasi	Loss	Akurasi	Presi si	Recall	F1-Score
Adam	0,8106	75%	48%	46%	47%
RMSProp	1,0010	69%	42%	42%	41%

TABEL 7. Hasil proses *testing* tiap kelas pada data tanpa augmentasi dengan sistem yang dilatih dari awal

Algoritma Optimasi	Kelas	Presi si	Recall	F1-Score
Adam	Sehat	88%	95%	91%
	Tungau laba-laba merah	65%	57%	60%
	Karat daun tingkat 1	69%	55%	61%
	Karat daun tingkat 2	38%	44%	41%
	Karat daun tingkat 3	29%	25%	27%
	Karat daun tingkat 4	0%	0%	0%
RMSProp	Sehat	90%	89%	90%
	Tungau laba-laba merah	36%	39%	37%
	Karat daun tingkat 1	48%	48%	48%

Karat daun tingkat 2	46%	61%	52%
Karat daun tingkat 3	33%	12%	18%
Karat daun tingkat 4	0%	0%	0%

Hasil proses *testing* pada data tanpa augmentasi dengan sistem yang telah dilatih menggunakan data hasil augmentasi di Tabel 8 menunjukkan bahwa kedua algoritma memiliki nilai akurasi, *recall*, dan *F1-Score* yang sama, namun memiliki nilai *loss* dan presisi yang berbeda. Algoritma Adam memiliki nilai *loss* yang sedikit lebih rendah dibandingkan algoritma RMSProp, sedangkan algoritma RMSProp memiliki nilai presisi yang sedikit lebih baik dibandingkan algoritma Adam. Tabel 9 menunjukkan hasil proses *testing* terkait Tabel 8 pada tiap kelas.

TABEL 8. Hasil proses *testing* pada data tanpa augmentasi dengan sistem yang telah dilatih menggunakan data hasil augmentasi

Algoritma Optimalisasi	Loss	Akurasi	Presi si	Recall	F1-Score
Adam	0,2176	93%	92%	95%	94%
RMSProp	0,2233	93%	93%	95%	94%

TABEL 9. Hasil proses *testing* tiap kelas pada data tanpa augmentasi dengan sistem yang telah dilatih menggunakan data hasil augmentasi

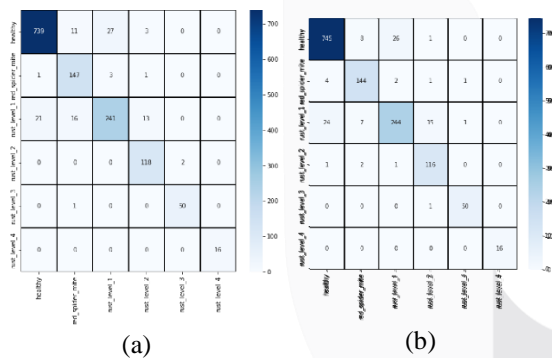
Algoritma Optimasi	Kelas	Presi si	Recall	F1-Score
Adam	Sehat	97%	95%	96%
	Tungau laba-laba merah	84%	97%	90%
	Karat daun tingkat 1	89%	83%	86%
	Karat daun tingkat 2	87%	98%	93%
	Karat daun	96%	98%	97%

RMSProp	tingkat 3			
	<b>Karat daun tingkat 4</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>
	Sehat	96%	96%	96%
	Tungau laba-laba merah	89%	95%	92%
	Karat daun tingkat 1	89%	84%	87%
	Karat daun tingkat 2	87%	97%	91%
	Karat daun tingkat 3	96%	98%	97%
<b>Karat daun tingkat 4</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	

pada *sparse data* atau data yang jarang serta data dengan *noise* yang tinggi [13], yang membuat algoritma tersebut dapat bekerja lebih baik pada data tanpa augmentasi yang bersifat *imbalance*.

Pada algoritma optimasi Adam dan RMSProp, penggunaan nilai *learning rate* yang rendah dapat menurunkan nilai *loss* dan mencapai keadaan konvergen lebih cepat dibandingkan nilai *learning rate* yang tinggi. Penggunaan *batch size* yang lebih kecil juga dapat membantu sistem mencapai keadaan konvergen yang lebih cepat dibandingkan penggunaan *batch size* yang lebih besar. Hal ini dikarenakan penggunaan *batch size* yang lebih kecil dapat mengurangi waktu yang dibutuhkan pada setiap iterasi dan mengurangi data *noise* yang mungkin ditemukan dalam penggunaan *batch size* yang lebih besar [12].

Pada proses *testing*, nilai akurasi, presisi, *recall*, dan *F1-Score* pada dua algoritma optimasi yang digunakan secara keseluruhan memiliki selisih yang tidak jauh berbeda satu sama lain. Hal ini berarti bahwa penggunaan augmentasi data untuk menyetarakan distribusi data dapat membantu proses evaluasi pada tiap kelas menjadi lebih saksama, seperti terlihat pada Gambar 6 dan Gambar 8, dimana sistem klasifikasi yang dibangun dapat mengidentifikasi penyakit karat daun tingkat 4 lebih baik dibandingkan penyakit lainnya setelah diuji menggunakan sistem yang telah dilatih dengan data hasil augmentasi.



GAMBAR 8. Confusion Matrix pada data tanpa augmentasi dengan sistem yang telah dilatih menggunakan data hasil augmentasi (a) algoritma Adam (b) algoritma RMSProp

B. Analisis Hasil Pengujian

Dari hasil yang diperoleh pada proses *training* menggunakan data hasil augmentasi, secara umum kedua algoritma memiliki performansi yang baik dan dapat mencapai kondisi konvergen yang cukup cepat. Namun, algoritma RMSProp lebih baik karena memiliki nilai *loss* yang lebih rendah, dengan selisih sebesar 0,015 dari algoritma Adam. Pada proses *training* dengan data tanpa augmentasi, algoritma Adam memiliki performansi yang lebih baik karena mampu menurunkan nilai *loss* sampai *epoch* terakhir. Sedangkan pada algoritma RMSProp, nilai *loss* terus meningkat mulai dari *epoch* ke-7 hingga *epoch* terakhir. Algoritma Adam dapat bekerja dengan baik

V. KESIMPULAN

Berdasarkan hasil penelitian yang ada, CNN dengan arsitektur EfficientNet-B0 menghasilkan performansi yang baik pada algoritma optimasi Adam dan RMSProp, yang masing-masing memiliki nilai *F1-Score* sebesar 91% pada data hasil augmentasi, dan 94% pada data tanpa augmentasi dengan sistem yang telah dilatih menggunakan data hasil augmentasi. Hal ini berarti bahwa peningkatan jumlah data dan penyetaraan distribusi data yang tidak seimbang dengan augmentasi data terbukti dapat membantu meningkatkan performansi sistem.

Dalam penggunaan CNN, pemilihan parameter yang tepat perlu diperhatikan untuk menghasilkan sistem klasifikasi dengan performansi yang optimal. Nilai *epoch*, *batch size*, *learning rate*, dan parameter lainnya memiliki faktor penting dalam menentukan lama proses *training* dan performansi sistem yang dihasilkan.

Untuk penelitian lebih lanjut, dapat dilakukan dengan meningkatkan jumlah data pada penyakit yang sama atau menambah jenis penyakit lain. Penggunaan algoritma optimasi dan arsitektur CNN lain juga dapat dilakukan pada dataset yang sama dengan *task* berbeda. Selain itu, penggunaan teknik lain seperti *oversampling* juga dapat dilakukan untuk menangani permasalahan distribusi data yang bersifat *imbalance*.



## REFERENSI

- [1] R. Harni *et al.*, “Teknologi Pengendalian Hama dan Penyakit Kopi,” *IAARD*, 2015.
- [2] S. W. Indiati and N. Saleh, “HAMA TUNGAU MERAH *Tetranychus urticae* PADA TANAMAN UBIKAYU DAN UPAYA PENGENDALIANNYA,” *Buletin Palawija*, pp. 73–79, 2010.
- [3] J. G. M. Esgario, R. A. Krohling, and J. A. Ventura, “Deep Learning for Classification and Severity Estimation of Coffee Leaf Biotic Stress,” Jul. 2019, [Online]. Available: <http://arxiv.org/abs/1907.11561>
- [4] F. J. P. Montalbo and A. A. Hernandez, “Classifying barako coffee leaf diseases using deep convolutional models,” *International Journal of Advances in Intelligent Informatics*, vol. 6, no. 2, pp. 197–209, Jul. 2020, doi: 10.26555/ijain.v6i2.495.
- [5] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, “A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network,” *Computers and Electronics in Agriculture*, vol. 154, pp. 18–24, Nov. 2018, doi: 10.1016/j.compag.2018.08.048.
- [6] F. R. Lumbanraja *et al.*, “Sistem Pakar Diagnosis Hama Dan Penyakit Tanaman Kopi Menggunakan Metode Breadth First Search (BFS) Berbasis Web,” *Jurnal Sistem Informasi dan Telematika*, vol. 11, Jun. 2020.
- [7] Y. D. Kartikasari and D. Arifianto, “SISTEM PAKAR DIAGNOSA HAMA DAN PENYAKIT TANAMAN KOPI MENGGUNAKAN METODE
- BACKWARD CHAINING BERBASIS WEB.”
- [8] J. Parraga-Alava, K. Cusme, A. Elica Loor, and E. Santander, “RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition,” 2019, doi: 10.17632/c5yvn32dzc.2.
- [9] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Computation*, vol. 29, no. 9. MIT Press Journals, pp. 2352–2449, Sep. 01, 2017. doi: 10.1162/NECO\_a\_00990.
- [10] M. Tan and Q. v Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proceedings of the 36th International Conference on Machine Learning (PMLR)*, 2019, pp. 6105–6114.
- [11] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, “Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention,” *IEEE Access*, vol. 9, pp. 14078–14094, 2021, doi: 10.1109/ACCESS.2021.3051085.
- [12] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model,” *Ecological Informatics*, vol. 61, Mar. 2021, doi: 10.1016/j.ecoinf.2020.101182.
- [13] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016, [Online]. Available: <http://arxiv.org/abs/1609.04747>