

OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI

(OPTIMIZATION OF CONVOLUTIONAL NEURAL NETWORK HYPERPARAMETERS FOR CLASSIFICATION OF RICE PLANT DISEASES)

Afis Julianto¹⁾, Andi Sunyoto²⁾, dan Ferry Wahyu Wibowo³⁾

^{1, 2, 3)}Magister Teknik Informatika, Universitas Amikom Yogyakarta

Yogyakarta, Indonesia

e-mail: afis.julianto@students.amikom.ac.id¹⁾, andi@amikom.ac.id²⁾, ferry.w@amikom.ac.id³⁾

ABSTRAK

Penyakit tanaman merupakan sebuah tantangan pada sektor pertanian terutama bagi petani padi. Mengidentifikasi penyakit pada daun padi merupakan langkah awal untuk memberantas dan mengobati penyakit, sehingga dapat meminimalisir terjadinya gagal panen. Dengan perkembangan cepat dari *convolutional neural network* (CNN), penyakit daun padi dapat dikenali dengan baik tanpa bantuan seorang ahli. Arsitektur *MobileNet-V2* digunakan untuk mengklasifikasi penyakit daun padi karena yang memiliki ukuran yang kecil namun dengan performa yang baik. Untuk meningkatkan performa dari model CNN, akan dilakukan optimasi *hyperparameter* yang terdiri *epoch*, *batch size*, *learning rate* dan *optimizer*. Penelitian ini bertujuan untuk untuk mendapatkan *hyperparameter* yang optimal sehingga memberikan performa yang baik pada model CNN. Dataset yang digunakan terdiri dari 3 kelas penyakit yang menyerang daun tanaman padi antara lain *blast*, *blight* dan *tungro*. Berdasarkan percobaan yang sudah dilakukan, penentuan *hyperparameter* sangat berpengaruh terhadap performa model. *Hyperparameter* dengan jumlah *epoch* 100, *batch size* 32, *learning rate* 0,001 dan *optimizer* *RMSProp* memberikan hasil yang paling optimal dengan nilai *accuracy* 97,56%, *precision* 97,64%, *recall* 97,57% dan *f1-score* 97,57%.

Kata Kunci: Optimasi Hyperparameter, Convolutional Neural Network, MobileNet-V2, Penyakit Tanaman Padi

ABSTRACT

Plant disease is a challenge in the agricultural sector, especially for rice farmers. Identification of diseases on rice leaves is the first step to eradicating and treating diseases, to minimize crop failure. With the rapid development of the *convolutional neural network* (CNN), rice leaf disease can be recognized well without the help of an expert. The *MobileNet-V2* architecture is used to classify rice leaf diseases due to its small size but good performance. To improve the performance of the CNN model, a *hyperparameter* consisting of an *epoch*, *batch size*, *learning rate*, and *optimizer*. This study purpose to have *hyperparameters* optimal The dataset used consists of 3 classes of diseases that attack the leaves of rice plants, including *blast*, *blight*, and *tungro*. Based on the experiments that have been carried out, the determination of *hyperparameters* greatly influences the model performance. *Hyperparameter* with *epochs*, *batch sizes* 32 *learning rate* and *optimizer* gives the most optimal results with *accuracy* 97.56%, *precision* 97.64%, *recall* 97.57%, and *f1-score* 97.57%.

Keywords: Hyperparameter Optimization, Convolutional Neural Network, MobileNet-V2, Diseases of Rice Plants

I. PENDAHULUAN

Tanaman padi menjadi salah satu tanaman yang paling banyak di tanam di Indonesia, hal ini dikarenakan beras merupakan salah satu makanan pokok masyarakat Indonesia. Menurut Badan Pusat Statistik Indonesia produksi padi tahun 2021 mengalami penurunan 0,23 juta Ton dibandingkan tahun 2020 [1]. Salah satu penyebab penurunan produksi padi biasanya dikarenakan rusaknya tanaman padi akibat serangan penyakit. Tanaman padi yang terserang penyakit dapat mengakibatkan berkurangnya kualitas dan kuantitas padi yang dihasilkan. Dampak terburuk yang dapat dialami adalah terjadinya gagal panen yang menyebabkan kerugian besar bagi para

petani. Oleh sebab itu penting bagi para petani untuk mengidentifikasi jenis penyakit yang menyerang tanaman padi, sehingga dapat dengan tepat memilih jenis penanganan dan pengobatan yang akan digunakan.

Secara umum identifikasi penyakit tanaman padi dapat dilakukan dengan uji coba di laboratorium atau dilakukan pengamatan dan penilaian secara visual terhadap gejala yang muncul [2]. Proses identifikasi penyakit di laboratorium masih kurang efektif karena membutuhkan berbagai bahan kimia [3] dan waktu lama [4]. Identifikasi secara visual merupakan pendekatan yang dilakukan secara subjektif dan masih dianggap sering terjadi kesalahan [5] hal ini disebabkan setiap orang memiliki penilaian yang berbeda-beda terhadap objek visual [6]. Kedua pendekatan

konvensional tersebut membutuhkan peran pakar untuk dapat melakukan identifikasi penyakit.

Dengan berkembangnya teknologi saat ini, pengenalan suatu objek dapat dilakukan dengan menggunakan *computer vision* dan *deep learning* seperti melakukan deteksi covid-19 [7], deteksi ikan [8], identifikasi hama dan penyakit tanaman [9]. Metode *deep learning* yang paling umum digunakan saat ini adalah *convolutional neural network* (CNN) [10]. CNN memiliki kemampuan utama dapat mengenali informasi prediktif dari suatu objek seperti gambar, teks, suara dan video meskipun objek tersebut diposisikan dimana saja pada saat diinputkan [11].

Penelitian mengenai identifikasi penyakit padi pernah dilakukan dengan menggunakan metode CNN dan menerapkan *10 fold cross validation* dapat menghasilkan akurasi 95,48% [12]. Penelitian lainnya juga dilakukan dengan menggunakan model arsitektur CNN seperti VGG 16 didapat akurasi 92,46% [13], AlexNet didapat akurasi 91,23 % [14] dan VGG 19 didapat akurasi 92% [15]. Meskipun arsitektur diatas memiliki performa yang baik, akan tetapi jumlah paramater yang dihasilkan model cukup besar. Semakin besar parameter yang dihasilkan maka ukuran model yang dihasilkan juga semakin besar dan proses komputasi yang dibutuhkan juga semakin lama [16].

Saat ini terdapat arsitektur CNN yang memiliki jumlah parameter yang kecil dengan peforma yang baik yaitu arsitektur CNN MobileNet-V2. Pada *framework* keras arsitektur MobileNet-V2 memiliki jumlah parameter 3,5 Juta. Penelitian dengan menggunakan MobileNet-V2 pernah dilakukan untuk mengklasifikasi penyakit daun jeruk [17] mendapat akurasi 98% dan klasifikasi daun kentang [18] mendapat akurasi 97,90%. Meskipun model CNN menghasilkan kinerja yang cukup baik, namun kinerjanya masih dapat ditingkatkan untuk mendapatkan model yang lebih optimal dengan melakukan hyperparameter tuning [19]. Hyperparameter tuning memiliki peran penting dalam mengoptimalkan dan meningkatkan kinerja CNN [20], [21]. Beberapa *hyperparameter tuning* seperti *epoch* [22], *batch size* [23], *learning rate* [24] dan *optimizer* [25] sering dilakukan penyesuaian untuk mengoptimalkan kinerja model CNN.

Berdasarkan uraian diatas, peneliti mencoba melakukan sebuah penelitian yang berfokus pada klasifikasi 3 jenis penyakit yang menyerang daun padi antara lain *blast*, *blight* dan *tungro* dengan menggunakan arsitektur CNN MobileNet-V2. Untuk mendapatkan hasil akurasi yang optimal

pada model CNN, peneliti mencoba melakukan *hyperparameter tuning* pada *epoch*, *batch size*, *learning rate* dan *optimizer* selama proses training model. Model akan di lakukan evaluasi dengan menggunakan *confusion matrix* untuk melihat tingkat akurasi, *precision*, *recall* dan *f1-score* yang dihasilkan model. Model yang dihasilkan dari penelitian ini diharapkan dapat membantu para petani dalam melakukan identifikasi penyakit tanaman padi sehingga petani dapat melakukan perawatan tanaman padi dengan tepat.

II. STUDI PUSTAKA

Beberapa penelitian terdahulu yang dijadikan acuan dan tinjauan pustaka pada penelitian ini diantaranya adalah :

Penelitian yang dilakukan oleh (Bhattacharya et al., 2020) dengan judul “*A Deep Learning Approach for the Classification of Rice Leaf Diseases*” mengenai klasifikasi 3 kelas penyakit daun padi dengan menggunakan metode CNN. Pada penelitian ini mengusulkan model CNN dengan hanya menerapkan dua hidden layers. Dari hasil percobaan yang dilakukan didapatkan hasil akurasi sebesar 94% untuk membedakan daun padi yang sehat dan sakit sedangkan dari hasil mengkategorikan berbagai jenis penyakit daun padi yang sakit didapatkan akurasi sebesar 78,44%. Dataset yang digunakan sebanyak 2000 gambar dibagi kedalam 4 kelas antara lain daun padi sehat, *bacterial blight*, *blast*, dan *brown spot* [26].

Penelitian lainnya yang dilakukan oleh (Priyanka & Kumara, 2021) melakukan klasifikasi 7 kelas penyakit padi menggunakan metode CNN dengan arsitektur VGG19. Proses augmentasi data diterapkan dengan menggunakan *sheer range*, *zoom range*, *horizontal flip* dan *rotation range*. Dari beberapa sekenario percobaan yang dilakukan, didapatkan hasil akurasi sebesar 95,25% dengan menerapkan augmentasi data *zoom range* dan *shear range* [27].

Penelitian mengenai arsitektur CNN MobileNet pernah dilakukan oleh (Riswandi dkk, 2021) untuk mengklasifikasi penyakit pada citra daun jeruk. Terdapat 3 kelas penyakit daun jeruk yang digunakan pada penelitian ini antara lain, daun jeruk normal, CTV dan HLB dengan ukuran citra 150 x 15 px. *Optimizer* RMSprop dan *learning rate* 0,001 digunakan selama proses pengujian model CNN. Dari percobaan yang dilakukan, didapatkan nilai akurasi hasil pada proses *training* mencapai 98% [17].

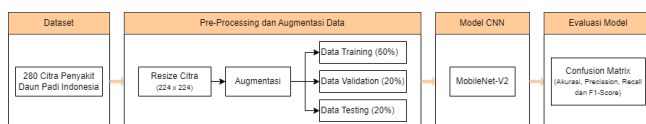
Penelitian sebelumnya mengenai *hyperparameter tuning* pernah dilakukan oleh (Minarno

dkk, 2021) terhadap model CNN untuk mengklasifikasi penyakit tumor otak pada citra MRI. Dataset yang digunakan sebanyak 3264 citra yang terdiri dari 4 kelas antara lain, *glioma*, *meningioma*, *hipofisis* dan tanpa tumor. Penggunaan CNN yang dikombinasikan dengan *hyperparameter tuning* dapat memberikan hasil yang optimal dalam melakukan klasifikasi tumor otak. Dari percobaan yang dilakukan, didapatkan akurasi sebesar 96%.

Penelitian lainnya yang dilakukan oleh (Putri, 2022) menggunakan arsitektur CNN MobileNet untuk melakukan klasifikasi penyakit tanaman daun kentang. MobileNet digunakan karena memiliki kelebihan antara lain *low cost*, *stable* dan *high precision*. Pada penelitian ini melakukan empat skenario perbandingan terhadap *optimizer*, *batch size*, *learning rate* dan *epoch*. Hasil terbaik yang didapat adalah dengan menggunakan *optimizer* RMSprop, *learning rate* 0,0001, *epoch* 50 dan *batch size* 32 dengan memperoleh akurasi 97,90% dan *loss* 0,0390 [18].

III. METODE PENELITIAN

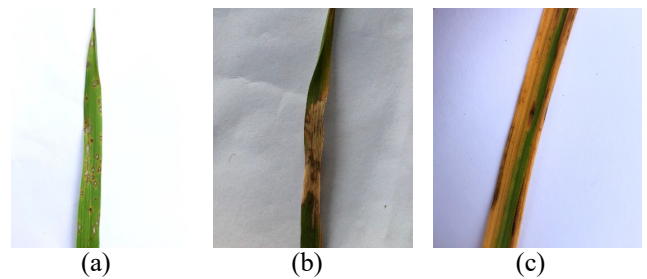
Metodologi penelitian yang digunakan pada penelitian ini dimulai dari menggunakan dataset publik penyakit daun padi indonesia yang didapat melalui situs kaggle. Selanjutnya melakukan *pre-processing* dan augmentasi pada dataset yang tersedia. Kemudian dilanjutkan dengan melakukan pelatihan data menggunakan arsitektur CNN MobileNet-V2. Tahap terakhir melakukan evaluasi model CNN dengan menggunakan *confusion matrix* untuk mengetahui tingkat akurasi, *precision*, *recall* dan *f1-score*. Alur metodologi penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian

A. Dataset

Pada penelitian ini menggunakan dataset citra penyakit tanaman padi yang ada di indonesia. Dataset ini memiliki 240 gambar penyakit daun padi yang terdiri dari 3 kelas penyakit, diantaranya blast, blight dan tungro dengan masing kelas terdiri dari 80 gambar seperti yang ditunjukkan pada Gambar 2. Semua data gambar memiliki ukuran 1440 x 1920 px dengan format JPG. Dataset ini dapat di akses melalui situs <https://www.kaggle.com/tedisetiady/leaf-rice-disease-indonesia>.



Gambar 2. Citra Penyakit Daun Padi (a) *Blast* (b) *Blight* (c) *Tungro*

B. Preprocessing dan Augmentasi Data

Pada tahap pre-processing data dilakukan dengan melakukan *resize* citra pada dataset yang didapatkan. Proses ini bertujuan untuk mengurangi dimensi citra yang diharapkan dapat mengurangi waktu komputasi pada saat proses pelatihan model CNN. Citra akan di *resize* menjadi 224 x 224 Px dari ukuran sebelumnya 1440 x 1920 Px. Jumlah citra yang tersedia pada dataset ini hanya sebanyak 240 citra. Hal ini dapat mempengaruhi akurasi yang dihasilkan oleh model CNN pada saat pelatihan dikarenakan dataset yang dimiliki sangat sedikit. Semakin banyak dataset yang dimiliki maka jaringan akan semakin baik dalam mengklasifikasi [28].

Augmentasi data akan dilakukan untuk mengatasi masalah kekurangan dataset dengan melakukan *horizontal flip*, *vertical flip*, *zoom range* 0,2, *shear range* 0,2 dan *rotation range* 25. Setiap citra akan diperbanyak sebanyak 5 kali sehingga setiap kelas penyakit akan menghasilkan 400 citra baru, maka didapatkan total citra baru sebanyak 1200 dari ketiga kelas penyakit. Selanjutnya citra baru hasil augmentasi digabungkan dengan citra asli, maka didapatkan total citra sebanyak 1440 untuk digunakan sebagai data penelitian seperti yang ditunjukkan pada Tabel 1. Hasil augmentasi citra dapat dilihat pada Gambar 3.

Tabel 1. Rincian Dataset Penyakit Daun Padi

Nama Penyakit	Citra Asli	Citra Hasil Augmentasi	Total Citra	Training (60%)	Validation (20%)	Testing (20%)
Blast	80	400	480	288	96	96
Blight	80	400	480	288	96	96
Tungro	80	400	480	288	96	96
Total	240	1200	1440	864	288	288



Gambar 3. Hasil Augmentasi Citra Penyakit Blast

C. Arsitektur MobileNet-V2

Arsitektur MobileNet-V2 didesain untuk mengurangi *complexity cost* dan ukuran model dalam *network* [29]. MobileNet sangat cocok untuk media yang memiliki *computational power* rendah seperti *smartphone* atau *rasberrypi*. Dalam MobileNet-V2 memiliki 2 tipe blok, yang pertama digunakan untuk *residual block* dan yang kedua digunakan untuk *down sizing*.

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 4. Arsitektur MobileNet-V2

Struktur dari block MobileNet-V2 dapat dilihat pada Gambar 4. Disetiap baris mendeskripsikan *sequence* dari 1 atau lebih modul *stride layers*, beserta jumlah berapa kali pengulangan *n*. Setiap *layer* di *sequence* yang sama memiliki nilai *c* sebagai *output channel*. Di layer pertama setiap *sequence* memiliki nilai *stride s* dan *stride* lainnya menggunakan *stride 1*. Seluruh *spatial convolution* menggunakan 3×3 *kernels* [29].

D. Evaluasi Model

Pada tahap ini bertujuan untuk mengevaluasi kinerja model CNN arsitektur MobileNet-V2. Evaluasi model akan dilakukan dengan menggunakan *confusion matrix* untuk mengetahui nilai *accuracy*, *precision*, *recall* dan *f1-score* dari model yang digunakan. Berikut ini *confusion matrix multi class* yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. *Confusion Matrix Multi Class*

		Prediction Class			
		Class 1	Class 2	...	Class n
Actual Class	Class 1	X_{11}	X_{12}	...	X_{1n}
	Class 2	X_{21}	X_{22}	...	X_{2n}

	Class n	X_{n1}	X_{n2}	...	X_{nn}

Dari tabel diatas, akan didapatkan jumlah *true positive* (TTP) untuk semua kelas, *true negative* (TTN), *false positive* (TFP) dan *false negative* (TFN) untuk setiap kelas *i* yang di hitung dengan menggunakan persamaan (1), (2), (3) dan (4) [30].

$$TTP_{all} = \sum_{j=1}^n x_{jj} \quad (1)$$

$$TTN_i = \sum_{j=1, j \neq i}^n \sum_{k=1, k \neq i}^n x_{jk} \quad (2)$$

$$TFP_i = \sum_{j=1, j \neq i}^n x_{ji} \quad (3)$$

$$TFN_i = \sum_{j=1, j \neq i}^n x_{ij} \quad (4)$$

Dari hasil TPP, TTN, TFP dan TFN, selanjutnya akan digunakan untuk menghitung akurasi (A), *precision* (P), *recall* (R), *f-measure score* (F) dari setiap kelas *i* dihitung dengan menggunakan persamaan (5), (6), (7) dan (9) [30][31].

$$A = \frac{TTP_{all}}{(Total\ Number\ of\ Entries)} \quad (5)$$

$$P_i = \frac{(TTP_{all})}{(TTP_{all} + TFP_i)} \quad (6)$$

$$R_i = \frac{(TTP_{all})}{(TTP_{all} + TFN_i)} \quad (7)$$

$$F_i = \frac{2(P_i)(R_i)}{P_i + R_i} \quad (8)$$

IV. HASIL DAN PEMBAHASAN

Bagian ini akan menjelaskan hasil dari percobaan yang telah dilakukan. Percobaan dilakukan dengan melatih data training dan data validation menggunakan arsitektur CNN MobileNet-V2. Pada tahap pelatiba model akan dilakukan dengan beberapa skenario yang bertujuan untuk mengoptimalkan model dalam mengklasifikasi penyakit tanaman padi. Terdapat 4 skenario optimisasi model CNN yang digunakan pada penelitian ini. Skenario pertama dilakukan pengujian terhadap jumlah *epoch*, skenario kedua dilakukan pengujian pengaruh *batch size*, skenario ketiga dilakukan pengujian pengaruh *learning rate* dan skenario keempat dilakukan pengujian pengaruh jenis *optimizer* yang optimal bagi model dalam melakukan pelatihan model sehingga mendapatkan peforma terbaik. Selanjutnya model akan dilakukan pengujian dengan menggunakan data testing yang terdiri dari 288 citra untuk mengetahui peforma dari model yang digunakan.

A. Pengaruh Jumlah Epoch

Skenario pertama dilakukan pengujian terhadap penggunaan jumlah *epoch* pada saat pelatihan model yang bertujuan untuk mendapatkan jumlah *epoch* yang optimal sehingga dapat

memberikan hasil peforma terbaik. *Epoch* merupakan suatu keadaan dimana seluruh dataset yang menjadi masukan pada model training telah selesai melewati seluruh *neural network* dalam satu kali putaran [32]. Pada penelitian ini melakukan pengujian terhadap beberapa nilai *epoch*, antara lain 25, 50, 75 dan 100. Saat melakukan pengujian *epoch* beberapa parameter seperti *batch size* diatur menggunakan nilai 32, *learning rate* menggunakan nilai 0,001 dan *optimizer* menggunakan RMSProp. Hasil pengujian *epoch* dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Akurasi dan Loss Terhadap Penggunaan Jumlah *Epoch*

Epoch	Akurasi Training	Akurasi Validation	Akurasi Testing	Loss
25	99,3	88,97	89,09	0,4414
50	99,76	90,62	91,31	0,3359
75	100	95,81	96,18	0,0903
100	100	98,26	97,56	0,0525

Berdasarkan Tabel 3 dapat dilihat bahwa semakin bertambahnya nilai *epoch* yang digunakan menghasilkan nilai akurasi yang semakin baik dan nilai *loss* yang semakin menurun. Jumlah *epoch* yang terlalu kecil mengakibatkan kegagalan pada proses update bobot untuk mencapai titik optimalnya. *Epoch* dengan jumlah 100 memberikan performa terbaik pada saat pelatihan model dengan tidak terjadinya *overfitting* dan *underfitting* hal ini dibuktikan dengan nilai akurasi data *training* dan data *validation* yang tidak berbeda jauh. Selain itu nilai akurasi yang didapatkan model menggunakan data *testing* sebesar 97,56%, lebih tinggi dibandingkan dengan menggunakan *epoch* lainnya. Berdasarkan hasil tersebut, dapat diambil kesimpulan bahwa *epoch* 100 merupakan jumlah *epoch* yang optimal digunakan dalam melakukan klasifikasi penyakit daun padi dengan menggunakan arsitektur CNN MobileNet-V2.

B. Pengaruh Nilai Batch Size

Skenario kedua dilakukan pengujian terhadap penggunaan nilai *batch size* yang bertujuan untuk mendapatkan nilai *batch size* yang mampu memberikan peforma terbaik pada saat pelatihan model. *Batch size* merupakan jumlah sampel data pelatihan yang digunakan pada setiap iterasi [24]. Pengujian dilakukan terhadap beberapa nilai *batch size* kecil seperti 8, 16 dan 32. Pada saat melakukan pengujian *batch size* beberapa parameter seperti *epoch* diatur sebanyak 100, *learning rate* menggunakan nilai 0,001 dan *optimizer* menggunakan RMSProp. Hasil pengujian *batch size* dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan Akurasi dan Loss Terhadap Penggunaan Nilai Batch Size

Batch Size	Akurasi Training	Akurasi Validation	Akurasi Testing	Loss
8	100	95,13	96,52	0,2080
16	100	97,13	96,52	0,1624
32	100	98,26	97,56	0,0525

Berdasarkan Tabel 4 dapat dilihat bahwa semakin besar nilai *batch size* menghasilkan akurasi yang semakin baik dan nilai *loss* yang semakin kecil. Pada saat pelatihan model menggunakan *batch size* 32 mendapatkan nilai akurasi 100% pada data *training* dan 98,26% pada data *validation*. Hasil pengujian dengan menggunakan data *testing* mendapatkan nilai akurasi 97,56%, sedikit lebih baik dibandingkan dengan *batch size* lainnya yang diuji.

C. Pengaruh Nilai Learning Rate

Skenario ketiga melakukan pengujian terhadap penggunaan nilai *learning rate* yang paling optimal pada model CNN yang digunakan. *Learning rate* merupakan salah satu parameter yang berpengaruh terhadap peforma model CNN [24], hal ini dikarenakan *learning rate* menghitung nilai koreksi bobot pada saat proses *training* berlangsung [7]. *Learning rate* yang dilakukan pengujian antara lain 0.1, 0.01, 0.001, 0.0001 dan 0.00001. Pada saat melakukan pengujian *learning rate* beberapa parameter seperti *epoch* diatur sebanyak 100, *batch size* menggunakan nilai 32 dan *optimizer* menggunakan RMSProp. Hasil pengujian *learning rate* dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan Akurasi dan Loss Terhadap Penggunaan Nilai Learning Rate

Learning Rate	Akurasi Training	Akurasi Validation	Akurasi Testing	Loss
0,1	97,56	89,23	89,23	0,6060
0,01	98,95	89,93	91,66	0,5799
0,001	100	98,26	97,56	0,0525
0,0001	97,10	83,68	87,56	0,4051
0,00001	88,77	78,42	80,90	0,5806

Berdasarkan Tabel 5 dapat dilihat bahwa disaat menggunakan nilai *learning rate* besar, nilai akurasi yang didapatkan dari data *training*, *validation* dan *testing* menghasilkan nilai yang rendah. Namun penggunaan *learning rate* yang terlalu kecil juga menghasilkan nilai akurasi yang lebih rendah. Hal ini dikarenakan nilai *learning rate* yang semakin besar berdampak pada proses *training* model yang semakin cepat tetapi tingkat ketelitian *neural network* akan berkurang. Sedangkan nilai *learning rate* yang semakin kecil menyebabkan tingkat ketelitian *neural network* yang semakin besar tetapi proses *training* akan memerlukan waktu yang lebih lama. *Learning rate* yang memiliki peforma terbaik pada percobaan ini adalah 0.001.

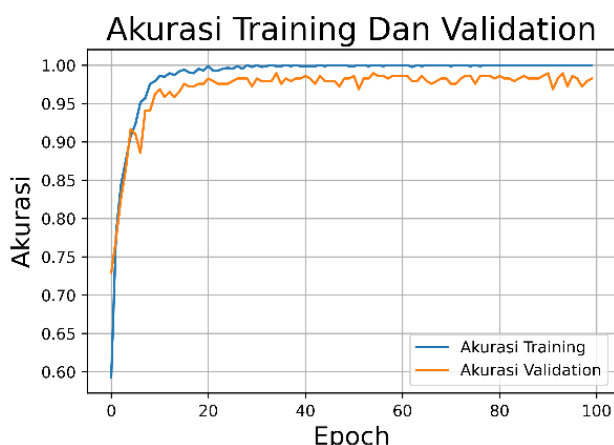
D. Pengaruh Optimizer

Skenario keempat melakukan pengujian pada beberapa jenis *optimizer* seperti Adam, RMSProp, SGD dan Nadam. *Optimizer* merupakan salah satu parameter yang digunakan untuk meminimalisir kesalahan, menemukan bobot optimal dan menghasilkan prediksi yang akurasi [33]. Pada saat melakukan pengujian *optimizer* beberapa parameter seperti *epoch* diatur sebanyak 100, *batch size* menggunakan nilai 32 dan *learning rate* 0.001. Hasil pengujian penggunaan *optimizer* dapat dilihat pada Tabel 6.

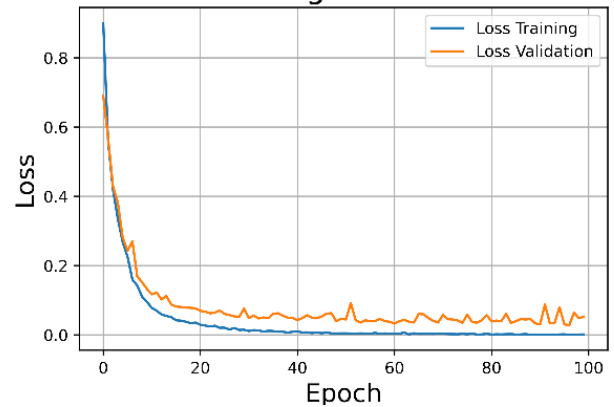
Tabel 6. Perbandingan Akurasi dan Loss Terhadap Penggunaan Nilai *Learning Rate*

Opti-mizer	Akurasi Training	Akurasi Vali-dation	Akurasi Testing	Loss
Adam	100	95,44	96,87	0,1447
RMSProp	100	98,26	97,56	0,0525
SGD	97,22	92,70	90,62	0,3309
Nadam	100	94,79	96,97	0,1456

Berdasarkan Tabel 6 dapat dilihat bahwa dari seluruh jenis *optimizer* yang di ujicoba, RMSProp menghasilkan nilai akurasi terbaik dengan nilai *loss* terendah ketika diujicoba menggunakan data *testing*. Sedangkan untuk *optimizer* Adam dan Nadam menghasilkan nilai akurasi dan *loss* tidak terlalu berbeda jauh, akan tetapi untuk *optimizer* SGD meskipun mendapatkan akurasi diatas 90% namun peformanya masih dibawah *optimizer* lainnya yang diujicoba. Grafik akurasi dan *loss* pada saat *training* model CNN dengan menggunakan *optimizer* RMSProp dapat dilihat pada Gambar 5.



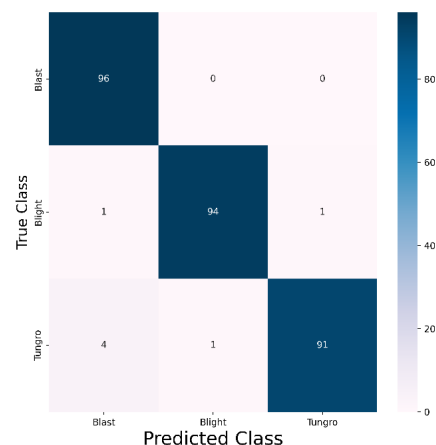
Loss Training Dan Validation



Gambar 5. Grafik Akurasi dan *Loss* MobileNet-V2 menggunakan *Optimizer* RMSProp

E. Evaluasi Model

Untuk mengetahui lebih lanjut performa model CNN yang digunakan, maka akan dilakukan evaluasi dengan menggunakan *confusion matrix* pada data *testing*. Model CNN dievaluasi untuk mendapatkan nilai *accuracy*, *precision*, *recall* dan *f1-score*. Hasil *confusion matrix* dapat dilihat pada Gambar 6.



Gambar 6. Hasil *Confusion Matrix*

Berdasarkan Gambar 6, ditemukan bahwa dari 96 citra pada kelas penyakit *blast* seluruh citra berhasil diprediksi dengan benar tanpa adanya kesalahan. Pada kelas penyakit *blight* yang terdiri dari 96 citra, 94 citra berhasil diprediksi benar, akan tetapi masih terdapat 2 citra yang diprediksi salah yaitu 1 citra diprediksi sebagai *blight* dan 1 citra diprediksi sebagai *tungro*. Terakhir pada kelas penyakit *tungro* yang terdiri dari 96 citra, 91 citra berhasil diprediksi benar, akan tetapi masih terdapat 5 citra yang diprediksi salah yaitu 4 citra diprediksi *blast* dan 1 citra diprediksi *blight*. Dari hasil *confusion matrix* didapatkan nilai *accuracy*, *precision*, *recall* dan *f1-score* seperti yang dilihat pada Tabel 7.

Tabel 7. Hasil Evaluasi Model

Akurasi Pengujian	Precision	Recall	F1-Score
97,56	97,64	97,57	97,57

Berdasarkan Tabel 7, dapat dilihat bahwa arsitektur CNN MobileNet-V2 yang digunakan pada penelitian ini dengan melakukan beberapa optimasi *hyperparameter* seperti *epoch*, *batch size*, *learning rate* dan *optimizer* dapat memberikan hasil yang sangat baik. Hal ini dibuktikan dari serangkaian percobaan yang dilakukan sehingga mendapatkan nilai akurasi sebesar 97,56%, *precision* 97,64%, *recall* 97,57% dan *f1-score* 97,57%.

V. KESIMPULAN

Penelitian ini melakukan optimalisasi model CNN dengan menggunakan beberapa *hyperparameter* seperti *epoch*, *batch size*, *learning rate* dan *optimizer* untuk klasifikasi penyakit padi indonesia. Penelitian ini bertujuan untuk mendapatkan *hyperparameter* yang optimal sehingga memberikan performa yang baik pada model CNN. Penelitian ini menggunakan CNN arsitektur MobileNet-V2 sebagai model pelatihan. Berdasarkan percobaan yang sudah dilakukan, penentuan *hyperparameter* sangat berpengaruh terhadap performa model. *Hyperparameter* dengan jumlah *epoch* 100, *batch size* 32, *learning rate* 0,001 dan *optimizer* RMSProp memberikan hasil yang paling optimal dengan nilai *accuracy* 97,56%, *precision* 97,64%, *recall* 97,57% dan *f1-score* 97,57% pada data *testing* yang digunakan. Hal ini menunjukkan bahwa model mampu dengan baik mengklasifikasi data *testing* sesuai dengan kelasnya. Pada penelitian ini hanya berfokus pada klasifikasi penyakit yang menyerang tanaman padi saja, diharapkan pada penelitian selanjutnya dapat melakukan klasifikasi pada hama yang menyerang tanaman padi. Perlu melakukan perbandingan dengan arsitektur CNN lainnya seperti DenseNet, Resnet dan Alexnet untuk mendapatkan akurasi yang lebih baik.

DAFTAR PUSTAKA

- [1] B. P. Statistik, *Luas Panen dan Produksi Padi Di Indonesia 2021*. Indonesia: Badan Pusat Statistik, 2021.
- [2] F. Martinelli *et al.*, “Advanced methods of plant disease detection. A review,” *Agron. Sustain. Dev.*, vol. 35, no. 1, pp. 1–25, 2015, doi: 10.1007/s13593-014-0246-1.
- [3] M. Turkoglu and D. Hanbay, “Apricot Disease Identification based on Attributes Obtained from Deep Learning Algorithms,” *2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018*, no. September 2018, 2019, doi: 10.1109/IDAP.2018.8620831.
- [4] J. G. A. Barbedo, “Digital Image Processing for Detecting and Classifying Plant Diseases,” *Springerplus*, vol. 2, 2013, doi: 10.22632/ccs-2017-252-66.
- [5] F. Jiang, Y. Lu, Y. Chen, D. Cai, and G. Li, “Image recognition of four rice leaf diseases based on deep learning and support vector machine,” *Comput. Electron. Agric.*, vol. 179, no. October, p. 105824, 2020, doi: 10.1016/j.compag.2020.105824.
- [6] R. P. Ramadhan and N. L. Marpaung, “Identifikasi jenis penyakit daun tanaman jagung menggunakan jaringan saraf tiruan berbasis backpropagation,” *Jom FTEKNIK*, vol. 6, no. 1, pp. 1–5, 2019.
- [7] S. Kumar *et al.*, “Chest X ray and cough sample based deep learning framework for accurate diagnosis of COVID-19,” *Comput. Electr. Eng.*, vol. 103, p. 108391, 2022, doi: 10.1016/j.compeleceng.2022.108391.
- [8] N. F. F. Alshdaifat, A. Z. Talib, and M. A. Osman, “Improved deep learning framework for fish segmentation in underwater videos,” *Ecol. Inform.*, vol. 59, 2020, doi: 10.1016/j.ecoinf.2020.101121.
- [9] Y. Yuan, L. Chen, H. Wu, and L. Li, “Advanced agricultural disease image recognition technologies: A review,” *Inf. Process. Agric.*, pp. 1–12, 2021, doi: 10.1016/j.inpa.2021.01.003.
- [10] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.
- [11] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, vol. 4. 2020.
- [12] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/j.neucom.2017.06.023.
- [13] S. Ghosal and K. Sarkar, “Rice Leaf Diseases Classification Using CNN With Transfer Learning,” in *2020 IEEE Calcutta Conference*, Feb. 2020, pp. 230–236, doi: 10.1109/CALCON49167.2020.9106423.
- [14] R. R. and D. Park, “A Multiclass Deep Convolutional Neural Network Classifier for Detection of Common Rice Plant Anomalies,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no.1, pp. 67–70, 2018, doi: 10.14569/IJACSA.2018.090109.
- [15] J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkan, “Using deep transfer learning for image-based plant disease identification,” *Comput. Electron. Agric.*, vol. 173, no. November 2019, p. 105393, Jun. 2020, doi: 10.1016/j.compag.2020.105393.
- [16] A. Julianto and A. Sunyoto, “A performance evaluation of convolutional neural network architecture for classification of rice leaf disease,” *IAES Int. J. Artif. Intell.*, vol. 10, no. 4, pp. 1069–1078, 2021, doi: 10.11591/IJAI.V10.I4.PP1069-1078.
- [17] R. Riswandi, R. Jamiah, N. Mardhatillah, and H. P. Hamid, “Klasifikasi Penyakit Pada Citra Daun Jeruk Menggunakan Arsitektur MobileNet berbasis Mobile Platform,” *J. Fokus Elektroda Energi List. Telekomun. Komputer, Elektron. dan Kendali*, vol. 6, no. 4, p. 212, 2021, doi: 10.33772/jfe.v6i4.19113.
- [18] Putri Teresia Ompusunggu, “Klasifikasi Penyakit Tanaman Pada Daun Kentang Dengan Metode Convolutional Neural Network Arsitektur Mobilenet,” *J. Syntax Fusion*, vol. 2, no. 9, pp. 740–751, 2022, doi: 10.54543/fusion.v2i09.217.
- [19] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.
- [20] P. Lacerda, B. Barros, C. Albuquerque, and A. Conci,

- “Hyperparameter optimization for COVID-19 pneumonia diagnosis based on chest CT,” *Sensors*, vol. 21, no. 6, pp. 1–11, 2021, doi: 10.3390/s21062174.
- [21] A. E. Minarno, M. Hazmi Cokro Mandiri, Y. Munarko, and H. Hariyady, “Convolutional Neural Network with Hyperparameter Tuning for Brain Tumor Classification,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, 2021, doi: 10.22219/kinetik.v6i2.1219.
- [22] I. G. T. Isa and B. Junedi, “Hyperparameter Tuning Epoch Dalam Meningkatkan Akurasi Data Latih Dan Data Validasi Pada Citra Pengendara,” *Pros. Semin. Nas. Sains dan Teknol.*, vol. 12, no. 1, pp. 231–237, 2022.
- [23] Y. N. Fuadah, I. D. Ubaidullah, N. Ibrahim, F. F. Taliningsing, N. K. Sy, and M. A. Pramuditho, “Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, p. 728, 2022, doi: 10.26760/elkomika.v10i3.728.
- [24] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtjas, W. Yustanti, and A. Prihanto, “Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam,” *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 2, pp. 44–48, 2021, doi: 10.26740/jieet.v5n2.p44-48.
- [25] D. Motta *et al.*, “Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes,” *PLoS One*, vol. 15, no. 7, pp. 1–30, 2020, doi: 10.1371/journal.pone.0234959.
- [26] S. Bhattacharya, A. Mukherjee, and S. Phadikar, “A Deep Learning Approach for the Classification of Rice Leaf Diseases,” *Adv. Intell. Syst. Comput.*, vol. 1109, no. May 2020, pp. 61–69, 2020, doi: 10.1007/978-981-15-2021-1_8.
- [27] A. A. J. V. Priyanka and I. M. S. Kumara, “Classification Of Rice Plant Diseases Using the Convolutional Neural Network Method,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 12, no. 2, p. 123, 2021, doi: 10.24843/lkjiti.2021.v12.i02.p06.
- [28] W. Widystuti and J. B. B. Darmawan, “Pengaruh jumlah data set terhadap akurasi pengenalan dalam deep convolutional network,” *Konf. Nas. Sist. Inf.*, pp. 8–9, 2018.
- [29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [30] C. Manliguez, “Generalized Confusion Matrix for Multiple Classes,” no. November, pp. 5–7, 2016, doi: 10.13140/RG.2.2.31150.51523.
- [31] D. Zhang, J. Wang, and X. Zhao, “Estimating the Uncertainty of Average F1 Scores,” in *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, Sep. 2015, pp. 317–320, doi: 10.1145/2808194.2809488.
- [32] Faiz Nashrullah, Suryo Adhi Wibowo, and Gelar Budiman, “The Investigation of Epoch Parameters in ResNet-50 Architecture for Pornographic Classification,” *J. Comput. Electron. Telecommun.*, vol. 1, no. 1, pp. 1–8, 2020, doi: 10.52435/complete.v1i1.51.
- [33] S. N. Rahmawati, E. W. Hidayat, and H. Mubarak, “Implementasi Deep Learning Pada Pengenalan Aksara Sunda Menggunakan Metode Convolutional Neural Network,” *Inser. Inf. Syst. Emerg. Technol. J.*, vol. 2, no. 1, p. 46, 2021, doi: 10.23887/insert.v2i1.37405.