

## Implementasi *Deep Learning* pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode *Convolutional Neural Network* (CNN)

Susi Yuliany<sup>1</sup>, Aradea<sup>2</sup>, Andi Nur Rachman<sup>3</sup>

<sup>123</sup>Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi

Jl. Siliwangi No. 24, Tasikmalaya, 46115, Jawa Barat, Indonesia

Email: <sup>1</sup>177006071@student.unsil.ac.id, <sup>2</sup>aradea@unsil.ac.id, <sup>3</sup>andy.rachman@unsil.ac.id

**Abstract.** *Implementing Deep Learning in Rice Pests Classification System Using Convolutional Neural Network (CNN).* Pest control in rice plants often results in the killing of non-pest organisms. Some researchers attempted to classify damage on the plants instead of the pests. Therefore, classifying the types of pests on rice plants will help farmers deal with pests on rice plants according to the type of pest immediately. This study uses the Convolutional Neural Network (CNN) method. The CNN method in several studies has overfitting problems. Therefore, this study proposes the application of three types of distribution of training data and testing data and using several parameters that function to reduce the problem of overfitting. Based on the evaluation, the 90%:10% data share is the most suitable data share for the dataset. The architecture used has training accuracy of 83.02%, 78.30%, and 81.13%. The test accuracy values of the three models are 69.33%, 77.33%, and 76%.

**Keywords:** *Convolutional Neural Network, training, test, overfitting, pest*

Abstrak. Pengendalian hama pada tanaman padi seringkali mengakibatkan terbunuhnya organisme bukan hama. Beberapa peneliti telah mencoba mengklasifikasi kerusakan pada tanaman alih-alih mengklasifikasikan hama itu sendiri. Oleh karena itu, proses klasifikasi jenis hama pada tanaman padi akan membantu petani melakukan penanganan hama pada tanaman padi sesuai jenis hama dengan segera. Penelitian ini menggunakan metode Convolutional Neural Network (CNN). Metode CNN pada beberapa penelitian mengalami masalah overfitting. Oleh sebab itu, penelitian ini mengusulkan penerapan tiga jenis pembagian data training dan data testing serta penggunaan beberapa parameter yang berfungsi mengurangi masalah overfitting. Berdasarkan evaluasi, pembagian data 90%:10% adalah pembagian data yang paling cocok untuk *dataset*. Arsitektur yang digunakan ialah dengan akurasi training 83,02%, 78,30%, dan 81,13%. Nilai akurasi pengujian dari ketiga model tersebut yaitu 69,33%, 77,33%, dan 76%.

**Kata kunci:** *Convolutional Neural Network, training, testing, overfitting, hama*

### 1. Pendahuluan

Sektor pertanian merupakan faktor yang sangat penting terkait ketahanan pangan dan teknologi AI banyak dimanfaatkan pada sektor tersebut. Beberapa contoh teknologi AI pada sektor pertanian ialah irigasi tetes, *drone* pertanian, sistem prediksi panen, dan *transplanter*. Salah satu bidang yang banyak diteliti dan dikembangkan adalah kemampuan komputer mengenali objek melalui citra atau gambar. Salah satu teknik dalam masalah pengenalan gambar atau objek digital (*image processing*) adalah teknik klasifikasi jaringan saraf tiruan (JST). Baru-baru ini, *deep learning* telah menjadi pusat pengembangan *Machine Learning*. Penggunaan *deep learning* dapat diterapkan pada berbagai jenis pekerjaan seperti memprediksi peluang dan kejadian, pengenalan objek, diagnosis penyakit [1]. Sistem *image processing* bertujuan membantu manusia mengenali atau mengklasifikasi objek secara efisien, cepat, tepat, serta dapat memproses data dalam jumlah banyak sekaligus [1].

Beberapa peneliti telah berusaha untuk mengklasifikasi kerusakan tanaman disbanding mengklasifikasikan hama tanaman. Dalam melakukan pengendalian hama ini, sebagian besar petani melakukan penyemprotan pestisida tanpa mempertimbangkan dosis, waktu, metode, dan

sasaran yang tepat. Akibatnya, penyemprotan tersebut justru akan membunuh organisme bukan hama sasaran. Banyak musuh alami hama yang terdapat di lahan pertanaman padi sawah menjadi terbunuh. Oleh karena itu, proses klasifikasi jenis hama pada tanaman padi memungkinkan petani melakukan penanganan hama tanaman padi sesuai jenis hama yang menyerangnya.

Metode *Convolutional Neural Network* (CNN) banyak digunakan dalam *image processing* karena tingkat akurasi yang tinggi dan lebih baik dalam pengenalan gambar visual. Terdapat beberapa penelitian yang melakukan klasifikasi citra menggunakan metode CNN. Mardiyah, 2020 [2] dalam penelitiannya yang berjudul “Implementasi *Deep Learning* Untuk *Image Classification* menggunakan Algoritma *Convolutional Neural Network* (CNN) pada citra Kebun dan Sawah” menggunakan *dataset* berjumlah 100 citra. Pembagian data yang digunakan yaitu 80%:20% karena dianggap memiliki nilai akurasi validasi paling tinggi yaitu 75% dengan akurasi *training* sebesar 96,25%. Hasil pengujian yang didapatkan menggunakan 20 data uji untuk dua kelas yaitu 75%.

Pada penelitian yang lain yaitu Alamsyah&Pratama, 2020 [3] dengan judul “Implementasi *Convolutional Neural Networks* (CNN) untuk Klasifikasi Ekspresi Citra Wajah Pada Fer-2013 *Dataset*” menggunakan beberapa *optimizer* dan tanpa *dropout*. Akurasi validasi tertinggi diraih saat menggunakan *optimizer* Adam dengan nilai 66% dan akurasi *training* sebesar 84%. Pengujian menggunakan model tersebut menghasilkan akurasi sebesar 64%. Kedua penelitian tersebut mengalami *overfitting* terbukti dari nilai akurasi validasi dan akurasi *training* yang cukup jauh.

Akurasi tinggi pada metode CNN tidak lepas dari kualitas komponen yang mendukung metode CNN itu sendiri. Selain itu, akurasi tinggi juga dipengaruhi oleh arsitektur dan parameter yang membangun metode CNN. Pada penelitian ini akan dilakukan pelatihan dengan beberapa pembagian data *training* dan data *testing* juga penggunaan beberapa parameter yang berfungsi mengurangi masalah *overfitting*.

## 2. Tinjauan Pustaka

### 2.1. Penelitian Terdahulu

Beberapa peneliti telah melakukan penelitian terkait hama maupun CNN, diantaranya pada penelitian yang dilakukan oleh Ririd dkk., 2018 [4] dengan judul “Implementasi Metode *Support Vector Machine* untuk identifikasi Penyakit Daun Tanaman Kubis” dengan hasil dari klasifikasi dipengaruhi oleh proses segmentasi yang dilakukan serta input parameter yang digunakan saat proses *training*. Pengujian menunjukkan rata-rata akurasi hasil klasifikasi mencapai 80,55%.

Berdasarkan penelitian yang dilakukan oleh Paliwang dkk., 2020 [5] dalam jurnalnya yang berjudul “Klasifikasi Penyakit Tanaman Apel dari Citra Daun dengan *Convolutional Neural Network*” metode CNN berhasil digunakan dan aplikasi berhasil diimplementasikan dengan baik dengan hasil final *test accuracy* yang diperoleh yaitu didapat keakuratan akurasi sebesar 97,1%. Metode *Convolutional Neural Network* juga digunakan pada penelitian yang berjudul “Implementasi *Convolutional Neural Network* untuk Identifikasi Penyakit Daun Gambas” oleh Sari&Swanjaya, 2020 [6] dengan menerapkan arsitektur MobileNet untuk identifikasi penyakit pada tanaman gambas dan memiliki akurasi 90%.

Mardiyah, 2020 [2] dalam penelitiannya yang berjudul “Implementasi *Deep Learning* untuk *Image Classification* Menggunakan Algoritma *Convolutional Neural Network* (CNN) pada Citra Kebun dan Sawah” menggunakan *dataset* berjumlah 100 citra mengalami *overfitting*. Pembagian data yang digunakan yaitu 80%:20% karena dianggap memiliki nilai akurasi validasi paling tinggi yaitu 75% dengan akurasi *training* sebesar 96,25%. Hasil pengujian yang didapatkan dengan menggunakan 20 data uji untuk dua kelas yaitu 75%.

Pada penelitian yang lain yaitu Alamsyah & Pratama (2020) [3] dengan judul “Implementasi *Convolutional Neural Networks* (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada Fer-2013 *Dataset*” menggunakan beberapa *optimizer* dan tanpa *dropout* mengalami *overfitting* juga. Akurasi validasi tertinggi diraih saat menggunakan *optimizer* Adam dengan

nilai 66% dan akurasi *training* sebesar 84%. Pengujian menggunakan model tersebut menghasilkan akurasi sebesar 64%.

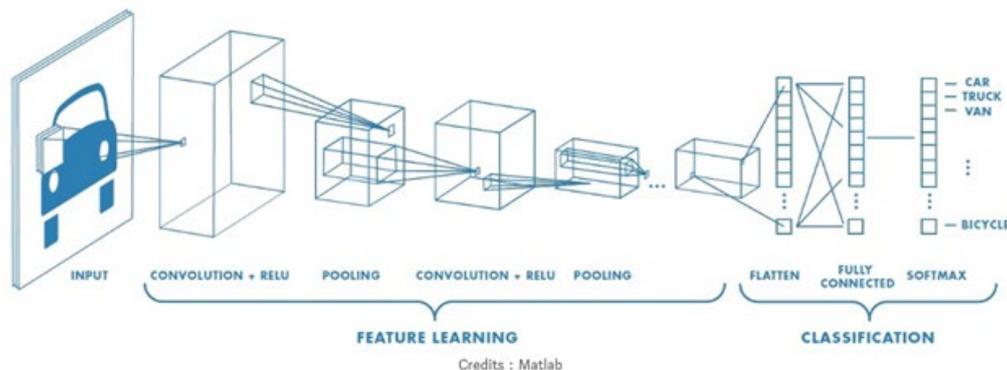
Penelitian [4], [5], dan [6] berfokus pada penyakit tanaman daripada hama yang menyerangnya maka penelitian ini mengusulkan untuk membentuk sistem klasifikasi hama khususnya pada tanaman padi. Kemudian pada penelitian [5] mendapatkan nilai akurasi testing yang tinggi yaitu sebesar 97,1% karena *dataset* yang digunakan berasal dari Kaggle dan berjumlah 7000 data *train*. Sementara itu, penelitian [2] dan [3] mengalami *overfitting* pada model sehingga ketika digunakan untuk pengujian hasil akurasinya tidak terlalu tinggi. Penelitian ini mengusulkan percobaan dengan tiga macam pembagian *dataset* juga penggunaan parameter lain untuk mengurangi *overfitting*.

## 2.2. Convolutional Neural Network (CNN)

Beberapa algoritma yang digunakan pada bidang *image processing* antara lain, *Naive Bayes*, *Support Vector Machine* dan *Neural Network*. *Convolutional Neural Network* (CNN) merupakan algoritma pengembangan *Neural Network*. Algoritma CNN memiliki hasil yang paling signifikan dalam pengenalan citra digital karena CNN diimplementasikan berdasarkan sistem pengenalan citra pada visual *cortex* manusia [1]. Hal ini juga diperkuat oleh penelitian yang mengklaim CNN sebagai model terbaik untuk memecahkan permasalahan *object detection* dan *object recognition* [7].

Secara teknis, *Convolutional Neural Network* (CNN) adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap terdiri dari beberapa *array* yang disebut *feature map*. CNN adalah perpaduan antara konvolusi citra yang berfungsi untuk proses ekstraksi fitur, dan *neural network* yang berfungsi untuk klasifikasi. Berikut adalah arsitektur *Convolutional Neural Network* [8].

Berdasarkan arsitektur LeNet5, terdapat 4 macam *layer* utama pada sebuah CNN yaitu *convolutional layer*, *relu layer*, *subsampling layer/pooling layer*, dan *fully connected layer*. Arsitektur CNN dapat dilihat pada Gambar 1.



**Gambar 1. Arsitektur CNN**

Berikut ini adalah penjelasan mengenai masing-masing *layer* [8].

### a. Convolution Layer

Konvolusi pada data citra bertujuan untuk mengekstraksi fitur dari citra input menggunakan filter. Filter ini berisi bobot yang berfungsi untuk mendeteksi karakter dari objek seperti tepi, kurva, atau warna.

### b. ReLU Layer (Fungsi Aktivasi ReLU)

ReLU (*Rectification Linear Unit*) merupakan operasi untuk pengenalan nonlinieritas dan peningkatan representasi dari model. Fungsi aktivasi ReLU adalah  $f(x) = \max(0, x)$  [9]. Nilai output dari neuron dapat dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri [9].

c. *Pooling (Subsampling) Layer*

*Pooling* atau *subsampling* adalah proses reduksi atau pengurangan ukuran sebuah data citra atau dapat disebut pengurangan ukuran matriks. Menurut Bejiga et al. (2017) terdapat dua macam *pooling* yang umum digunakan yaitu *average pooling* dan *max pooling*. Zhi et al. (2016) mengatakan nilai yang diambil pada *average pooling* adalah nilai rata-rata sedangkan pada *max pooling* adalah nilai maksimum [9].

d. *Fully Connected Layer*

Input pada lapisan ini berasal dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari lapisan ini adalah untuk menyatukan semua node menjadi satu dimensi (Albelwi & Mahmood, 2017 dalam [9]).

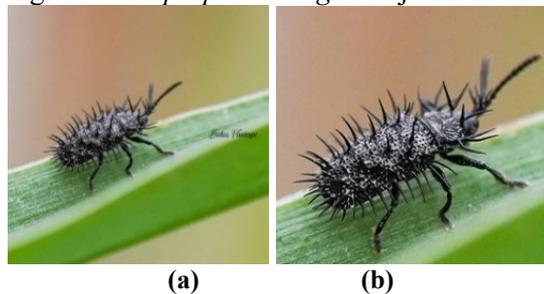
### 3. Metodologi Penelitian

Perangkat keras yang digunakan adalah 1 unit laptop Lenovo Ideapad 320 dengan spesifikasi: (1) *Operating System*: Windows 10 Pro; (2) *Processor*: AMD A9-9420; dan (3) *Memory*: RAM 4,00 GB. Perangkat lunak yang digunakan antara lain: (1) Bahasa Pemrograman *Python*; (2) *Google Colaboratory*; dan (3) *Library Tensorflow*. Langkah selanjutnya adalah representasi sistem yakni melakukan pengumpulan data dan perancangan dan desain dari sistem yang akan dibuat.

#### 3.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data sekunder, data diambil dari *Google* dan *hashtag* di *Instagram*. Gambar tersebut merupakan gambar berwarna/ RGB dengan format jpg dan merupakan gambar hama yang terdiri dari lima kategori hama dan 1.065 total seluruh gambar. Jenis hama yang diidentifikasi pada penelitian ini yaitu: (1) Belalang (258); (2) Hispa Padi (140); (3) Penggerek Batang Padi (284); (4) Ulat Grayak (278); dan (5) Wereng Hijau (105).

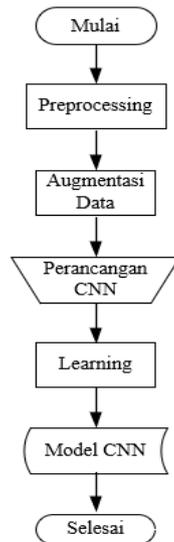
*Preprocessing* sebelum digunakan sebagai *input* pada model dilakukan pada *dataset* tersebut dengan memotong gambar agar objek hama terlihat jelas sehingga objek tersebut hampir menyentuh tepi bingkai. Hasil *preprocessing* ditunjukkan oleh Gambar 2.



Gambar 2. (a) Gambar Sebelum di-*crop*; (b) Gambar Sesudah di-*crop*

#### 3.2. Perancangan Sistem

Terdapat beberapa tahapan dalam proses pembuatan sistem klasifikasi CNN seperti yang ditunjukkan pada Gambar 3.



**Gambar 3. Perancangan Sistem**

Penelitian ini melakukan *preprocessing* dengan mengubah ukuran semua gambar (*resize*) menjadi persegi dengan ukuran 100x100 pixel. Kemudian dilakukan pembagian data *training* dan data validasi dari seluruh jumlah gambar yaitu 1065 dimana perbandingan datanya yakni 70%:30%, 80%:20%, dan 90%:10%. Tabel 1. berikut merupakan skenario pembagian data.

**Tabel 1. Skenario Pembagian Data**

Skenario 70%:30%		Skenario 80%:20%		Skenario 90%:10%	
Data <i>Training</i>	Data Validasi	Data <i>Training</i>	Data Validasi	Data <i>Training</i>	Data Validasi
746	319	852	213	959	106

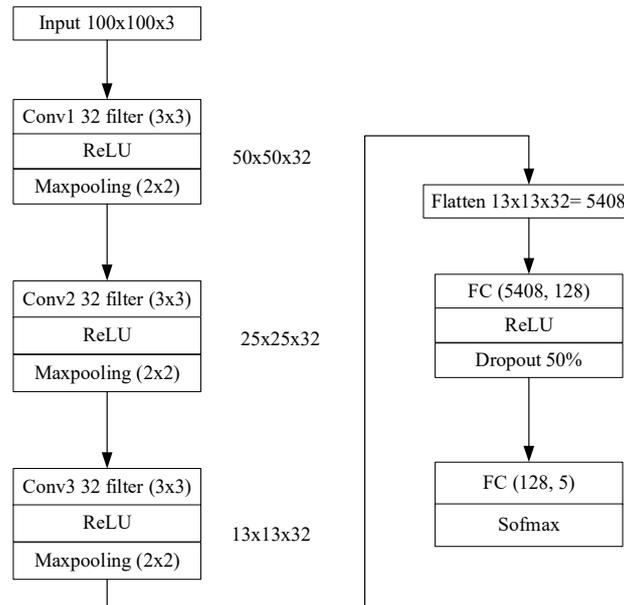
Augmentasi yang dilakukan dalam penelitian ini adalah membalikkan gambar secara horizontal (*horizontal flip*), rotasi gambar secara acak (*random rotation*), dan memperbesar gambar secara acak (*random zoom*) sebesar 20%. Melakukan augmentasi citra dapat meningkatkan akurasi model CNN yang dilatih karena dengan proses augmentasi model mendapatkan data-data tambahan yang berguna untuk membuat model yang dapat melakukan generalisasi secara lebih baik [10]. Augmentasi citra ditunjukkan oleh Gambar 4.



**Gambar 4. Augmentasi Citra**

Arsitektur CNN dalam penelitian ini memiliki 3 *convolutional layer* yang menggunakan fungsi aktivasi ReLU dipadukan dengan *Max Pooling*. Selain itu, dilakukan penyisipan *padding* agar rasio gambar tersebut tidak berubah. Tahap klasifikasi memiliki 2 *fully connected layer*. Fungsi aktivasi yang digunakan pada *hidden layer* adalah ReLU, sedangkan fungsi aktivasi pada

output adalah *Softmax*. Selain itu, penelitian menggunakan *dropout* sedangkan pada penelitian [3] tidak menggunakan *dropout*. Model CNN dapat dilihat pada Gambar 5.



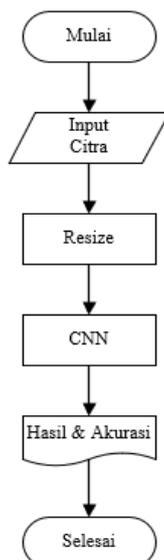
**Gambar 5. Rancangan Model CNN**

Pada proses *learning*, parameter yang diinisiasi antara lain jumlah *epoch*, optimasi, dan *learning rate*. Pada penelitian ini, optimasi yang digunakan adalah Adam mengacu pada penelitian sebelumnya [8] yang menemukan optimasi Adam sebagai yang terbaik. Kemudian penggunaan beberapa jumlah *epoch* dan beberapa nilai *learning rate* pada setiap pembagian *dataset* sehingga diperoleh jumlah *epoch* dan *learning rate* yang memberikan akurasi yang optimum.

Jumlah *epoch* yang digunakan yaitu 50, 100, 200 dan 500. *Epoch* merupakan proses ketika seluruh *dataset* sudah melalui proses *training* atau pelatihan dalam *neural network* sampai proses itu kembali ke awal [11]. Sedangkan nilai *learning rate* yang digunakan yaitu 0,1; 0,01; 0,001; dan 0,0001. *Learning rate* merupakan salah satu parameter *training* yang berfungsi untuk menghitung nilai koreksi bobot pada waktu proses *training*. Semakin besar nilai *learning rate* maka proses *training* berjalan semakin cepat. Semakin besar nilai *learning rate*, maka ketelitian jaringan akan semakin berkurang dan berlaku sebaliknya[12]. Model CNN terbaik didapatkan apabila telah melakukan pelatihan atau *training* dengan percobaan menggunakan beberapa parameter yang telah ditentukan.

### 3.3. Testing (Pengujian)

Proses *testing* dimulai dengan melakukan *resize* citra input menjadi 100x100. Kemudian citra input diklasifikasi oleh CNN. *Output* dari proses klasifikasi CNN merupakan klasifikasi kelas hama dan probabilitas citra uji. Proses *testing* pada penelitian ini menggunakan data citra uji sebanyak 75 dengan rincian 15 citra dari setiap 5 kelas hama. Gambar 6 berikut merupakan alur proses *testing*.



**Gambar 6. Alur Proses *Testing***

Persamaan yang digunakan untuk menghitung akurasi ditunjukkan pada persamaan (1).

$$\text{Akurasi} = \frac{\text{Jumlah data benar}}{\text{Jumlah data uji}} \times 100\% \quad (1)$$

Penarikan kesimpulan dilakukan untuk memberikan gambaran umum terhadap analisis data dan hasil evaluasi percobaan yang mencakup keseluruhan penelitian.

## 4. Hasil dan Diskusi

### 4.1. *Learning*

Pelatihan dilakukan dengan menggunakan *epoch* 50, 100, 200, dan 500 serta *learning rate* 0,001 dan 0,0001 pada masing-masing *epoch*. Pelatihan yang menggunakan *learning rate* 0,1 dan 0,01 menghasilkan grafik yang tidak konvergen. Maka pelatihan tersebut tidak akan dicantumkan. Hasil pelatihan dapat dilihat pada Tabel 2. Tabel warna kuning menunjukkan akurasi validasi dan *loss* validasi tertinggi pada setiap pembagian data. Kemudian tabel berwarna hijau menunjukkan hasil akurasi validasi tertinggi dari semua pelatihan.

**Tabel 2. Hasil Pelatihan dengan *Epoch* 50, 100, 200 dan, 500**

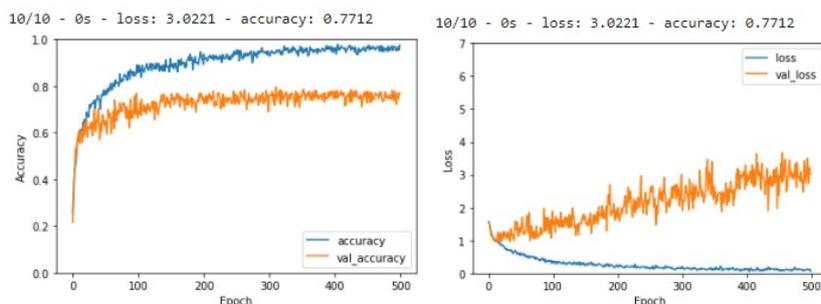
<i>Epoch</i>	<i>Learning Rate</i>	Pembagian Data			
		70%:30%	80%:20%	90%:10%	
50	0.001	<i>Training Acc</i>	0.8016	0.7852	0.7925
		<i>Training Loss</i>	0.5607	0.5904	0.5835
		<i>Validation Acc</i>	0.6803	0.7089	0.6887
		<i>Validation Loss</i>	1.2996	1.0895	1.0328
		<i>Time (s)</i>	30	33	50
	0.0001	<i>Training Acc</i>	0.6783	0.7277	0.7028
		<i>Training Loss</i>	0.8002	0.7461	0.7697
		<i>Validation Acc</i>	0.6520	0.6150	0.6604
		<i>Validation Loss</i>	0.9231	0.9555	0.9282
		<i>Time (s)</i>	31	33	51
100	0,001	<i>Training Acc</i>	0.8700	0.9026	0.8759
		<i>Training Loss</i>	0.3506	0.2630	0.3302
		<i>Validation Acc</i>	0.7053	0.6995	0.8302
	<i>Validation Loss</i>	1.4793	1.7013	0.7708	
	<i>Time (s)</i>	61	63	92	
	0,0001	<i>Training Acc</i>	0.7601	0.7347	0.7602
<i>Training Loss</i>		0.5985	0.6279	0.5922	

		<i>Validation Acc</i>	0.6426	0.6150	0.6792
		<i>Validation Loss</i>	1.0248	1.1748	1.1033
		<i>Time (s)</i>	61	62	91
200	0,001	<i>Training Acc</i>	0.8834	0.9331	0.9270
		<i>Training Loss</i>	0.3145	0.1738	0.1839
		<i>Validation Acc</i>	0.7367	0.7512	0.7830
		<i>Validation Loss</i>	1.3710	1.6345	1.3225
		<i>Time (s)</i>	120	127	191
	0,0001	<i>Training Acc</i>	0.8445	0.8380	0.8394
		<i>Training Loss</i>	0.4222	0.4443	0.4450
		<i>Validation Acc</i>	0.6646	0.6291	0.6415
		<i>Validation Loss</i>	1.0534	1.3148	1.3268
		<i>Time (s)</i>	120	126	196
500	0,001	<i>Training Acc</i>	0.9759	0.9624	0.9562
		<i>Training Loss</i>	0.0630	0.1066	0.1098
		<i>Validation Acc</i>	0.7712	0.7606	0.8113
		<i>Validation Loss</i>	3.0221	2.5829	2.1615
		<i>Time (s)</i>	305	359	457
	0,0001	<i>Training Acc</i>	0.9276	0.9178	0.9134
		<i>Training Loss</i>	0.1920	0.2113	0.2172
		<i>Validation Acc</i>	0.6803	0.6948	0.6981
		<i>Validation Loss</i>	1.5669	1.6551	1.2493
		<i>Time (s)</i>	308	370	458

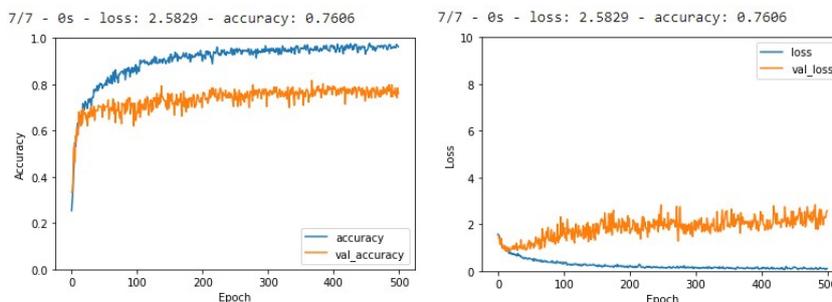
Hasil pelatihan terbaik akan dituangkan dalam Tabel 3 dan grafik pelatihan tersedia pada Gambar 7 sampai dengan Gambar 9.

**Tabel 3. Model Training Terbaik pada Setiap Pembagian Data**

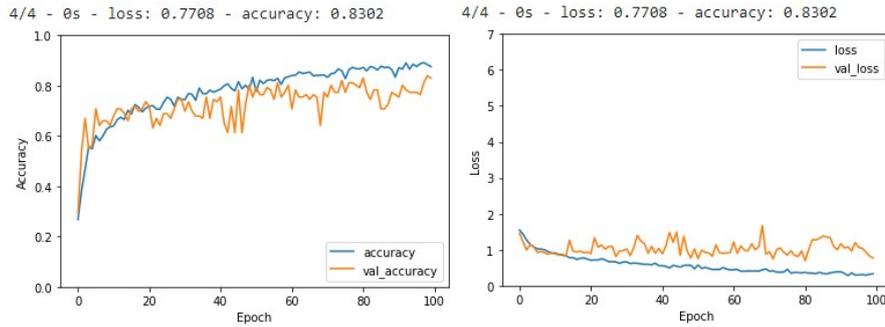
Pembagian Data	Epoch	Learning Rate	Validation Acc	Validation Loss
70%:30%	500	0,001	77,12%	3,0221
80%:20%	500	0,001	76,06%	2,5829
90%10%	100	0,001	83,02%	0,7708



**Gambar 7. Grafik Pelatihan 70%:30%**



**Gambar 8. Grafik Pelatihan 80%:20%**



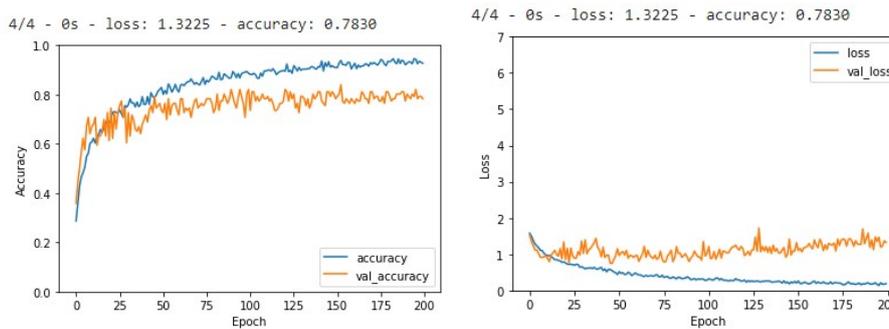
**Gambar 9. Grafik Pelatihan 90%:10%**

Tiga pelatihan terbaik dari semua pelatihan diraih pada saat pembagian data *training* dan data validasi 90%:10%. Tabel 4 merupakan tiga pelatihan terbaik dari semua pelatihan.

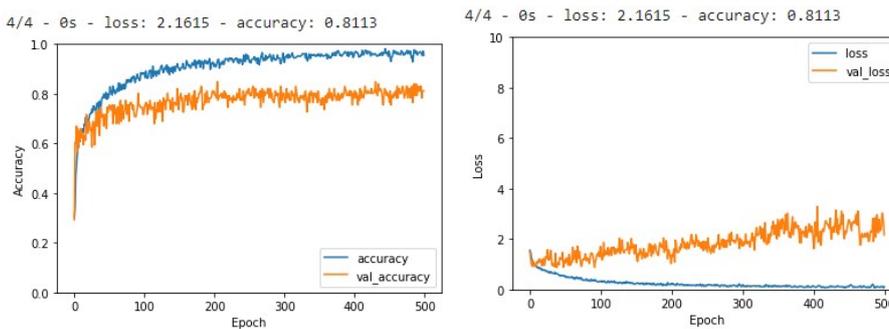
**Tabel 4. Model Pelatihan Terbaik dari Seluruh Pelatihan**

Pembagian Data	Epoch	Learning Rate	Validation Acc	Validation Loss
90%:10%	100	0,001	83,02%	0,7708
90%:10%	200	0,001	78,30%	1,3225
90%:10%	500	0,001	81,13%	2,1615

Grafik pelatihan *epoch* 100 disajikan pada Gambar 8. Sementara grafik pelatihan *epoch* 200 dan 500 disajikan pada Gambar 10 dan Gambar 11.



**Gambar 10. Grafik dengan Epoch 200**



**Gambar 11. Grafik dengan Epoch 500**

#### 4.2. Testing (Pengujian)

Data uji yang digunakan sebanyak 75 citra dengan jumlah setiap kategori sebanyak 15 citra. Data tersebut tidak termasuk *dataset* yang terintegrasi dengan sistem. Tabel 5 dan Tabel 6 merupakan akurasi pengujian model CNN yang telah dipilih. Akurasi dalam tahap ini merupakan variabel yang merepresentasikan kinerja yang digunakan untuk menilai tolok ukur keberhasilan model CNN untuk mengklasifikasi citra hama.

**Tabel 5. Pengujian Model CNN (i)**

Pembagian Data	Epoch	Learning Rate	Kelas	Hasil Klasifikasi		Akurasi
				Benar	Salah	
70%:30%	500	0,001	Belalang	13	2	69,33%
			Hispa	13	2	
			Penggerek Batang	12	3	
			Ulat Grayak	5	10	
			Wereng Hijau	9	6	
<b>Total</b>			<b>52</b>	<b>23</b>		
80%:20%	500	0,001	Belalang	14	1	72%
			Hispa	11	4	
			Penggerek Batang	11	4	
			Ulat Grayak	7	8	
			Wereng Hijau	11	4	
<b>Total</b>			<b>54</b>	<b>22</b>		
90%:10%	100	0,001	Belalang	7	8	69,33%
			Hispa	15	0	
			Penggerek Batang	14	1	
			Ulat Grayak	5	10	
			Wereng Hijau	11	4	
<b>Total</b>			<b>52</b>	<b>23</b>		

**Tabel 6. Pengujian Model CNN (ii)**

Pembagian Data	Epoch	Learning Rate	Kelas	Hasil Klasifikasi		Akurasi
				Benar	Salah	
90%:10%	100	0,001	Belalang	7	8	69,33%
			Hispa	15	0	
			Penggerek Batang	14	1	
			Ulat Grayak	5	10	
			Wereng Hijau	11	4	
<b>Total</b>			<b>52</b>	<b>23</b>		
90%:10%	200	0,001	Belalang	13	2	77,33%
			Hispa	13	2	
			Penggerek Batang	15	0	
			Ulat Grayak	6	9	
			Wereng Hijau	11	4	
<b>Total</b>			<b>58</b>	<b>17</b>		
90%:10%	500	0,001	Belalang	13	2	76%
			Hispa	14	1	
			Penggerek Batang	13	2	
			Ulat Grayak	5	10	
			Wereng Hijau	12	3	
<b>Total</b>			<b>57</b>	<b>18</b>		

Berdasarkan proses pengujian yang telah dilakukan, didapatkan akurasi paling tinggi pada model pelatihan dengan pembagian data *training* dan data validasi 90%:10% dengan *epoch* 200 dengan nilai 77,33% dengan prediksi 58 benar dan 17 salah. Hal ini menunjukkan bahwa model pelatihan terbaik belum tentu menghasilkan klasifikasi terbaik pula. Karena pada penelitian ini, model pelatihan terbaik memiliki nilai akurasi validasi 83,02% sedangkan pada pengujianya, hasil akurasi yang didapat yaitu 69,33%.

Sementara itu, pengujian yang dilakukan pada kelas Ulat Grayak memiliki jumlah benar yang selalu lebih sedikit daripada jumlah salah. Maka dapat dikatakan bahwa sistem klasifikasi hama tanaman padi yang dibuat ini tidak cukup layak digunakan pada kelas Ulat Grayak.

### 4.3. Ancaman terhadap Validitas

#### 4.3.1. Validitas Internal

Dataset dalam penelitian ini bukan dataset standar komunitas peneliti dikarenakan data yang dibutuhkan tidak ada pada website atau tempat dataset yang terpercaya (seperti Kaggle, UCI, dan Google's Dataset), sehingga dataset yang digunakan berasal dari pencarian manual. Jumlah totalnya hanya 1065 dan data pada setiap kelasnya tidak seimbang. Pada gambar yang digunakan, terdapat benda-benda lain disekitar hama yang mungkin akan terdeteksi oleh mesin sehingga dapat mempengaruhi proses learning. Jumlah data yang digunakan cukup memadai tetapi alangkah baiknya untuk penelitian selanjutnya bisa menggunakan data yang lebih banyak dan lebih fokus pada objek yang akan diklasifikasi.

#### 4.3.2. Validitas Konstruksi

*Preprocessing* yang dilakukan terhadap data hanya *me-resize* ukuran citra menjadi 100x100 pixel. Ukuran tersebut akan mengalami tiga kali konvolusi sehingga akhirnya menghasilkan ukuran citra 13x13 pixel. Ukuran citra *input*-an tersebut bisa dinaikkan lagi agar ketika mengalami konvolusi hasil akhirnya tidak terlalu kecil sehingga mesin dapat mengenali pola lebih baik.

#### 4.3.3. Validitas Kesimpulan

Tiga model terbaik yang terpilih memiliki nilai akurasi validasi 83,02%; 78,30%; dan 81,13%. Ketiga model tersebut mengalami *overfitting* meskipun sudah diterapkan augmentasi data, penyisipan *padding*, dan penggunaan *dropout*. Hal ini disebabkan oleh kualitas data yang digunakan.

#### 4.3.4. Validitas Eksternal

Hasil pengujian menunjukkan akurasi tertinggi pada model dengan pembagian data 90%:10% dengan *learning rate* 0,001 dan *epoch* 200. Model tersebut memiliki proses *learning* yang lebih baik daripada dua pelatihan yang lain jika dilihat dari grafik. Meskipun ketiganya mengalami *overfitting*, model kedua tersebut memiliki grafik yang jarak akurasi training dan validasinya tidak terlalu jauh. Naik turunnya nilai akurasi validasi tidak terlalu drastis dibanding model yang pertama. Model pertama memiliki nilai akurasi validasi yang naik dan penurunan drastis pada proses *learning*. Selain itu, nilai akurasi pengujian dipengaruhi oleh jumlah data uji yang digunakan.

## 5. Kesimpulan dan Saran

Arsitektur CNN yang digunakan dalam penelitian ini yaitu tiga lapisan konvolusi dan dua lapisan *fully connected* dengan menambahkan *dropout* dan augmentasi citra untuk mengurangi masalah *overfitting*. Pembagian data yang digunakan yaitu 70%:30%, 80%:20%, dan 90%:10%. Berdasarkan percobaan beberapa pelatihan menggunakan *optimizer* Adam dan *learning rate* 0,001, nilai akurasi validasi dari masing-masing pembagian data yaitu 77,12%; 76,06%; dan 83,02% dengan *epoch* yang berbeda, yaitu 500, 500, dan 100. Sementara itu, nilai akurasi validasi untuk tiga model pelatihan tertinggi ada pada pembagian data 90%:10% dengan nilai 83,02%; 78,30%; dan 81,13% dengan *epoch* 100, 200, dan 500. Total model CNN yang dipilih untuk dijadikan pengujian berjumlah lima. Nilai akurasi pengujian masing-masing model yaitu 69,33%; 72%; 69,33%; 77,33%; dan 76%. Nilai akurasi tertinggi yaitu 77,33% diraih oleh model dengan pembagian data 90%:10% dengan akurasi *training* sebesar 78,30%.

Semua pelatihan yang dilakukan mengalami *overfitting*. Menyisipkan *padding*, melakukan augmentasi data dan menggunakan *dropout* juga tidak sepenuhnya menghilangkan *overfitting* tetapi proses tersebut cukup membantu mengurangi *overfitting* pada model. Hal tersebut dipengaruhi oleh *dataset* yang tidak seimbang dan kualitas *dataset* yang digunakan. Data gambar pada penelitian ini memiliki *background* yang terlalu luas dan cenderung tidak fokus kepada objek yang akan di klasifikasi sehingga mempengaruhi nilai akurasi yang dihasilkan. Berdasarkan hasil penelitian, sistem klasifikasi hama tanaman padi ini memiliki nilai akurasi lebih baik daripada dua penelitian sebelumnya yang telah dijelaskan pada bagian

Pendahuluan karena melakukan beberapa percobaan pembagian data dan penggunaan beberapa parameter untuk mengurangi masalah *overfitting*.

Penelitian ini masih terdapat banyak kekurangan dan memerlukan upaya perbaikan diantaranya menghilangkan *background* atau menghilangkan benda-benda lain pada citra yang mungkin mempengaruhi pelatihan sehingga citra inputan fokus terhadap objek yang akan di klasifikasi. Kemudian menggunakan *dataset* yang seimbang setiap kelasnya, selain itu resolusi citra pada tahap *preprocessing* dapat ditingkatkan. Proses tersebut diharapkan dapat menaikkan akurasi validasi dan mengurangi *overfitting*.

## Referensi

- [1] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 01, no. 02, pp. 104–108, 2019.
- [2] M. I. Mardiyah, "Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Kebun dan Sawah," Universitas Islam Indonesia, 2020.
- [3] D. Alamsyah and D. Pratama, "Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada Fer-2013," vol. 4, no. 2, pp. 350–355, 2020.
- [4] A. R. T. H. Ririd, A. W. Kurniawati, and Y. Yunhasnawa, "Implementasi Metode Support Vector Machine Untuk Identifikasi Penyakit Daun Tanaman Kubis," *J. Inform. Polinema*, vol. 4, no. 3, p. 181, 2018, doi: 10.33795/jip.v4i3.204.
- [5] A. A. A. Paliwang, M. R. D. Septian, M. Cahyanti, and E. R. Swedia, "Klasifikasi Penyakit Tanaman Apel Dari Citra Daun Dengan," *Sebatik*, pp. 207–212, 2020.
- [6] D. F. Sari and D. Swanjaya, "Implementasi Convolutional Neural Network Untuk Identifikasi Penyakit Daun Gambas," 2020.
- [7] K. O. Lauw, L. W. Santoso, and R. Intan, "Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan Convolutional Neural Network Berbasis Android," 2019.
- [8] M. I. Arifin, "Pengolahan Citra Dengan Metode Convolutional Neural Network ( Cnn )," Politeknik Perkapalan Negeri Surabaya, 2019.
- [9] E. N. Arrofiqoh and Harintaka, "Implementasi Metode Convolutional Neural Network untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi (The Implementation of Convolutional Neural Network Method for Agricultural Plant Classification in High Resolution Imagery)," *Geomatika*, vol. 24, no. 2, pp. 61–68, 2018, doi: <http://dx.doi.org/10.24895/JIG.2018.24-2.810>.
- [10] K. H. Mahmud, Adiwijaya, and S. Al Faraby, "Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network," in *e-Proceeding of Engineering*, 2019, vol. 6, no. 1, pp. 2127–2136.
- [11] W. Anggraini, "Deep Learning untuk Deteksi Wajah yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) dengan Tensorflow," Universitas Islam Negeri Ar-Raniry, 2020.
- [12] T. Retnowardhani, Astari; Ramdani, "Apakah Deep Learning?," 2019. <https://mmsi.binus.ac.id/2019/11/26/apakah-deep-learning/> (accessed Sep. 06, 2021).