



## IMPLEMENTASI *DEEP LEARNING* UNTUK IDENTIFIKASI UMUR TANAMAN BERDASARKAN CITRA DAUN PADA *SMART FARMING*

Budi Prayitno<sup>1)</sup>, Pritasari Palupiningsih<sup>\*2)</sup>, Farhan Muhamad Ikhsan<sup>3)</sup>

<sup>1, 2, 3</sup> Teknik Informatika, Institut Teknologi PLN

<sup>1, 2, 3</sup> Menara PLN, Jl. Lkr. Luar Barat, Duri Kosambi, Kecamatan Cengkareng, Jakarta Barat

Email: <sup>1</sup>budiprayitno@itpln.ac.id, <sup>2\*</sup>pritasari@itpln.ac.id, <sup>3</sup>farhan1931129@itpln.ac.id

### Abstract

*Technology plays an important role in optimizing agricultural production, one of which is through the application of smart farming. Smart Farming is a paradigm in agriculture that utilizes information and communication technology (ICT). The case study raised in this study is the use of smart farming in determining plant age. Plant age is an important factor in determining the harvest. Plants that are harvested at the right time can produce quality products in optimal quantities. Traditional farmers determine plant age manually. This has challenges, namely the process takes a long time and a lot of energy, especially for large agricultural areas. Plant age must be identified quickly and easily, the results of plant age identification are accurate and consistent and can be applied to large agricultural areas. The urgency of this research is the creation of a deep learning model that is used to detect the optimum plant age with a high accuracy value. The importance of this research lies not only in the development of technology but also in its contribution to the farmer's economy and the progress of the agricultural sector. This study aims to implement deep learning to form a classification model for identifying plant age based on leaf images and to evaluate the classification model to produce high accuracy. The research method used follows a flow consisting of problem understanding, data understanding, data preparation, modeling, and evaluation. The deep learning method used is classification with the application of the Convolutional Neural Network (CNN) VGG architecture algorithm, which has been proven effective in image analysis. The results of this study are Research on age classification models on plant leaf images using the classification method with the CNN algorithm is carried out with the stages of data collection and class division, image resizing, data augmentation, adding keras models, convolution, max pooling, flatten, relu, and with the training of 20 epochs. The results of model formation with the CNN algorithm using VGG16 get higher accuracy than VGG19. The best accuracy value is 78% from the confusion matrix results using VGG19 with a data ratio of 60% training data, 20% validation data, and 20% testing data.*

**Keywords:** *deep learning; smart farming; convolutional neural network; vgg architecture; plant life*

### Abstrak

Teknologi memiliki peran penting dalam mengoptimalkan hasil produksi pertanian, salah satunya melalui penerapan *smart farming*. *Smart Farming* merupakan sebuah paradigma dalam bidang pertanian yang memanfaatkan teknologi informasi dan komunikasi (TIK). Studi kasus yang diangkat dalam penelitian ini adalah pemanfaatan *smart farming* pada penentuan umur tanaman. Umur tanaman merupakan faktor penting dalam menentukan hasil panen. Tanaman yang dipanen pada waktu yang tepat dapat menghasilkan produk berkualitas dan kuantitas optimal. Petani tradisional menentukan umur tanaman secara manual. Hal ini memiliki tantangan, yaitu prosesnya memakan waktu lama dan tenaga besar, terutama untuk area pertanian luas. Umur tanaman harus dapat diidentifikasi dengan cepat dan mudah, hasil identifikasi umur tanaman akurat dan konsisten serta mampu diterapkan pada area pertanian luas. Urgensi dari penelitian ini adalah terciptanya model *deep learning* yang diterapkan untuk mendeteksi umur tanaman yang optimum dengan nilai akurasi yang tinggi. Pentingnya penelitian ini tidak hanya terletak pada pengembangan teknologi, tetapi juga pada kontribusinya terhadap perekonomian petani dan kemajuan sektor pertanian secara keseluruhan. Penelitian ini memiliki tujuan untuk mengimplementasikan *deep learning* untuk membentuk model klasifikasi dalam mengidentifikasi umur tanaman berdasarkan citra daun dan melakukan evaluasi model klasifikasi untuk menghasilkan akurasi yang tinggi. Metode penelitian yang digunakan mengikuti alur yang terdiri atas *problem understanding*, *data understanding*, *data preparation*, *modelling*, dan *evaluation*. Metode *deep learning* yang digunakan adalah klasifikasi dengan penerapan algoritma *Convolutional Neural Network* (CNN) arsitektur VGG, yang telah terbukti efektif dalam analisis citra. Hasil dari penelitian ini adalah Penelitian model klasifikasi umur pada citra daun tanaman menggunakan metode klasifikasi dengan algoritma CNN dilakukan dengan tahapan pengumpulan data dan pembagian kelas, *image resizing*, *augmentation* data, menambahkan model keras, *convolution*, *maxpooling*, *flatten*, *relu*, dan dengan *training* sebanyak 20 *epoch*. Hasil pembentukan model dengan algoritma CNN menggunakan arsitektur VGG16 mendapatkan akurasi lebih tinggi dibanding arsitektur VGG19. Sehingga didapatkan nilai akurasi terbaik adalah sebesar 78% dari hasil *confusion matrix* dengan menggunakan VGG19 dengan rasio data 60% data *training*, 20% data *validation* dan 20% data *testing*.



**Kata Kunci:** *deep learning; smart farming; convolutional neural network; arsitektur vgg; umur tanaman.*

## 1. PENDAHULUAN

*Smart Farming* merupakan sebuah paradigma dalam bidang pertanian yang memanfaatkan teknologi informasi dan komunikasi (TIK) untuk mengumpulkan data secara *real-time* tentang kondisi tanaman dan lingkungannya sehingga mendapatkan informasi yang dapat digunakan dalam pengambilan keputusan yang presisi dan terukur guna meningkatkan efisiensi, produktivitas, dan keberlanjutan pertanian.

Umur tanaman merupakan faktor penting dalam menentukan hasil panen. Tanaman yang dipanen pada waktu yang tepat dapat menghasilkan produk berkualitas dan kuantitas optimal [1]. Panen prematur dapat berakibat pada penurunan kualitas dan kuantitas produk. Secara tradisional, identifikasi umur tanaman dilakukan manual oleh petani [2]. Cara ini menjadi tantangan tersendiri, diantaranya yaitu memakan waktu dan tenaga, prosesnya memakan waktu lama dan tenaga besar, terutama untuk area pertanian luas [3]. Oleh karenanya, diperlukan solusi yang lebih baik untuk identifikasi umur tanaman yang mengatasi kelemahan metode manual. Mengidentifikasi umur tanaman harus dapat dengan cepat dan mudah, akurat, hasil konsisten dan mampu diterapkan pada area pertanian luas.

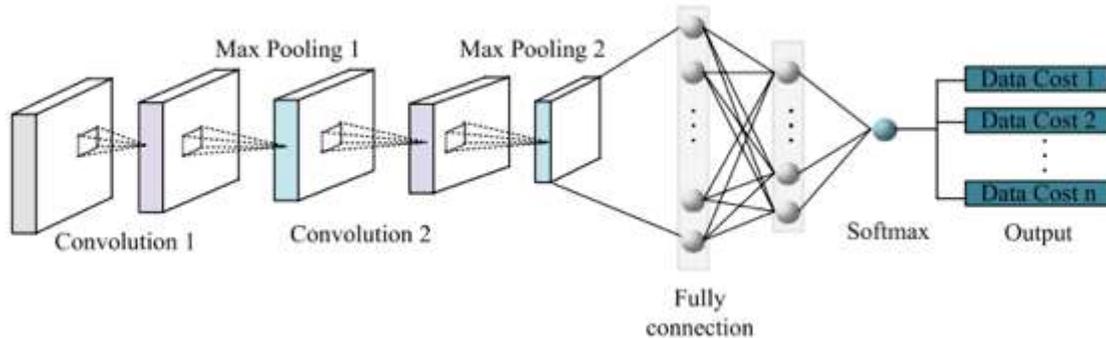
*Deep Learning* merupakan teknologi kecerdasan buatan yang memiliki kemampuan belajar dari data dan mampu membuat prediksi, memiliki potensi besar untuk mengatasi kelemahan metode manual dan menjadi solusi yang lebih baik. *Deep Learning* telah terbukti efektif dalam pengolahan citra, termasuk klasifikasi objek, segmentasi gambar, dan deteksi objek [4], [5]. *Deep learning* memiliki potensi besar untuk diaplikasikan bersama dalam *Smart Farming*. *Deep learning* dapat digunakan untuk memprediksi hasil panen berdasarkan data historis, kondisi cuaca, dan hama penyakit. *Deep learning* dapat digunakan untuk mendeteksi hama dan penyakit tanaman secara dini dan akurat [6]. *Deep learning* membantu petani menentukan jenis dan jumlah pupuk serta pestisida yang tepat dengan menganalisis citra daun dan memprediksi umur tanaman berdasarkan ciri-ciri yang teridentifikasi.

*Smart farming* merupakan sebuah paradigma dalam bidang pertanian dengan memanfaatkan teknologi kecerdasan buatan, komputasi awan, *internet of things* (IoT), *drone*, dan robotika. Contoh pengaplikasian *smart farming* adalah dibuatnya *automatic watering systems* pada *vertical farming* [7]. Metodologi pertanian modern ini sangat penting dalam mengatasi tantangan meningkatnya permintaan akan produksi pangan [8], [9]. Pemanfaatan *deep learning*, sebagai contoh adalah pemanfaatan algoritma CNN, mengambil peran penting dalam bidang *smart farming*, yaitu dengan memberikan hasil kinerja unggul dalam hal kemampuan klasifikasi gambar dan visi komputer, terutama di bidang identifikasi penyakit tanaman [10], [11]. Penerapan CNN dalam mendeteksi dan mengidentifikasi penyakit secara akurat sejak dini, petani dapat mengambil tindakan tepat waktu untuk mengurangi kerugian [11]. Arsitektur VGG-16, mampu digunakan untuk mendiagnosis penyakit tanaman dengan akurasi tinggi. Arsitektur VGG-16 menggunakan fungsi aktivasi *Rectified Linear Units* (ReLU), menunjukkan kemampuan beradaptasi dan efisiensi model *deep learning* dengan nilai akurasi yang tinggi dibandingkan dengan MobileNet [12], [13]. Metode *deep learning* tidak hanya digunakan untuk identifikasi penyakit saja, akan tetapi juga dapat dimanfaatkan untuk identifikasi hama tanaman [6]. Aplikasi pemanfaatan *deep learning* dapat menjadi solusi untuk proses pengambilan keputusan, efisiensi operasional, dan keberlanjutan praktik pertanian secara keseluruhan [14]. *Smart farming* yang memanfaatkan teknik *deep learning* seperti CNN dan *Recurrent Neural Networks* (RNN), dapat menjadi solusi untuk berbagai aspek pertanian dalam mengembangkan sektor pertanian [15]. Sinergi antara *smart farming* dan *deep learning*, dalam penggunaan CNN dengan arsitektur VGG, menghadirkan solusi menguntungkan untuk meningkatkan produktivitas pertanian, identifikasi penyakit, dan optimasi umur dan masa panen tanaman, yang berdampak dapat memberikan kontribusi besar untuk ketahanan pangan, peningkatan ekonomi petani dan keberlanjutan pertanian menjadi lebih maju dan modern [10]. Berdasarkan uraian penelitian sebelumnya tersebut di atas, *deep learning* memiliki kelebihan dan keunggulan untuk diterapkan pada bidang pertanian. Namun demikian, belum ada penelitian yang mengangkat implementasi *deep learning* untuk mengidentifikasi umur tanaman. Oleh karena itu, maka penelitian ini akan mengangkat implementasi *deep learning* untuk identifikasi umur tanaman pada *smart farming*. Penelitian ini memiliki tujuan utama yaitu untuk mengevaluasi tingkat akurasi model yang dihasilkan oleh algoritma CNN dalam mengklasifikasi fase pertumbuhan sawi berdasarkan citra dengan akurat. Metode yang digunakan melibatkan penggunaan algoritma CNN dengan mengaplikasikan arsitektur VGG, yang telah terbukti efektif dalam analisis citra.

## 2. METODE PENELITIAN

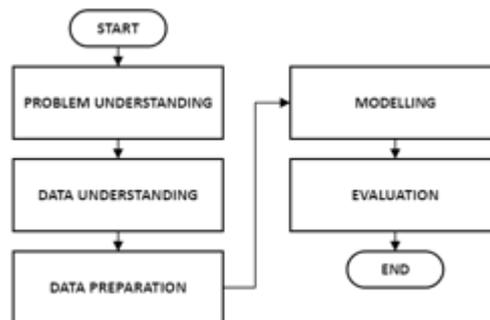
Untuk menyelesaikan permasalahan penelitian, peneliti merumuskan ke dalam sebuah kerangka pemikiran yang terdiri atas indikator, proposed method, objective dan measurement. Indikator merupakan data masukan yang digunakan, yaitu berupa data citra daun. Proposed method merupakan metode dan algoritma yang akan digunakan untuk memproses data. Metode yang digunakan mengikuti alur yang terdiri atas *problem understanding, data understanding, data preparation, modelling, dan evaluation*. Algoritma yang digunakan adalah algoritma CNN dengan arsitektur VGG. Objective yang akan dicapai dari penerapan algoritma tersebut adalah model klasifikasi umur tanaman. *Measurement*

adalah pengukuran yang digunakan untuk mengevaluasi model yang telah dihasilkan, yaitu menggunakan confusion matrix.



**Gambar 1. Struktur CNN[16]**

Penelitian ini menggunakan pendekatan metodologi yang terstruktur untuk menyelesaikan permasalahan penelitian yang ada. Penerapan *deep learning* yang dimaksud adalah menerapkan algoritma *Convolutional Neural Network* (CNN) (gambar 1) dengan arsitektur VGG. Dengan kombinasi ini, diharapkan mampu mencapai tujuan penelitian, yaitu untuk mengembangkan sebuah sistem yang dapat mengidentifikasi umur tanaman dengan akurasi yang tinggi. Objek tanaman yang digunakan pada penelitian ini adalah sawi hijau. Gambar 2 berikut merupakan diagram alir penelitian yang disusun.



**Gambar 2. Digram Alir Penelitian**

Diagram alir tersebut di atas terdiri atas 5 tahap, diantaranya yaitu *problem understanding*, *data understanding*, *data preparation*, *modelling*, dan *evaluation*. Berikut adalah penjelasan dari setiap tahapnya:

**2.1. Problem Understanding**

Pada tahap ini dilakukan identifikasi konteks dan masalah yang berkaitan dengan deteksi umur tanaman, diantaranya adalah pemahaman tentang pentingnya identifikasi umur tanaman dalam pertanian, tantangan yang dihadapi dalam proses ini, dan potensi solusi yang dapat diberikan oleh implementasi algoritma CNN. Selain itu juga dilakukan analisis literatur yang berkaitan dengan penerapan *deep learning* untuk deteksi umur tanaman, salah satunya adalah menggunakan CNN. Hal ini mencakup penelusuran literatur terhadap penelitian-penelitian sebelumnya yang telah menggunakan CNN untuk tujuan serupa, termasuk teknik-teknik yang telah digunakan, *dataset* yang telah digunakan, dan hasil yang telah dicapai. Berdasarkan pemahaman tentang konteks, masalah, dan literatur terkait, peneliti merumuskan pertanyaan penelitian yaitu, bagaimana algoritma CNN dapat diterapkan untuk mengidentifikasi umur tanaman berdasarkan citra daun secara akurat.

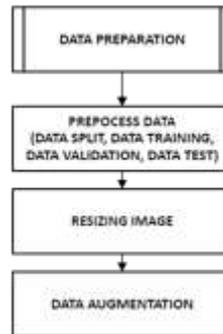
**2.2. Data Understanding**

Tahap selanjutnya yaitu memahami jenis data yang akan digunakan dalam penelitian, seperti kebutuhan citra tanaman dengan berbagai tahapan pertumbuhan. Adapun karakteristik data yang perlu diperhatikan yaitu terkait dengan resolusi, kejelasan, dan variasi yang mungkin terjadi antar citra daun tanaman. Jenis data yang akan digunakan, yaitu citra daun tanaman sawi. Pemahaman akan karakteristik citra tersebut sangatlah penting, yaitu terkait dengan resolusi, format, skala warna, dan kejelasan gambar. Selain itu, ada hal lain yang harus diperhatikan, yaitu variasi yang mungkin terjadi dalam citra daun, seperti perbedaan dalam tekstur, bentuk, dan ukuran daun tanaman sawi. Pada tahap ini peneliti akan mengumpulkan data citra daun tanaman sawi menjadi sebuah *dataset*.

Sebagai studi kasus pengambilan sampel citra daun, dipilih citra daun tanaman sawi hijau. Data diperoleh dari Jirifarm yang berada di Jl. Alun Alun No.I No.58, Curug Kulon, Kec. Curug, Kabupaten Tangerang, Banten. Data pada penelitian ini menggunakan data primer yang diekstraksi.

### 2.3. Data Preparation

Tahap ini merupakan tahapan untuk mempersiapkan *dataset* citra sebelum dilakukan pelatihan model CNN. Tahapan *preprocessing* data diantaranya adalah normalisasi, *resizing* citra, dan *augmentation* data yang dilakukan untuk meningkatkan variasi dalam *dataset*.



**Gambar 3.** Proses data *preparation*

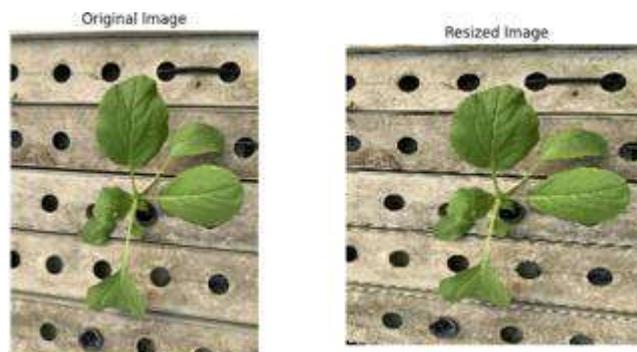
Diagram alir di dalam tahap data preparation seperti pada gambar diatas, diantaranya yaitu:

a. *Split Data*

Pada tahap ini dilakukan pembagian *dataset* yang ada ke dalam kategori yaitu *training*, *validation* dan *test*.

b. *Resizing Image*

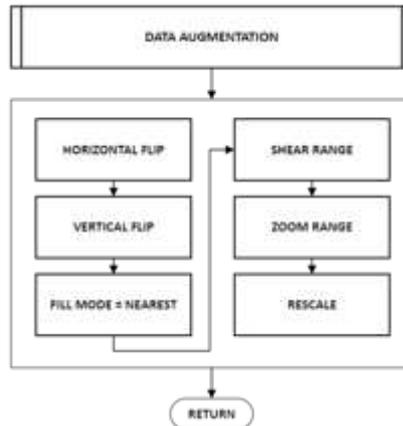
Pada tahap ini semua ukuran citra dilakukan penyamaan ukuran citra pada data agar mempersingkat waktu proses latihan. Ilustrasi proses ini sebagaimana hasilnya pada gambar 4 berikut ini.



**Gambar 4.** Perbandingan hasil proses *resize image*

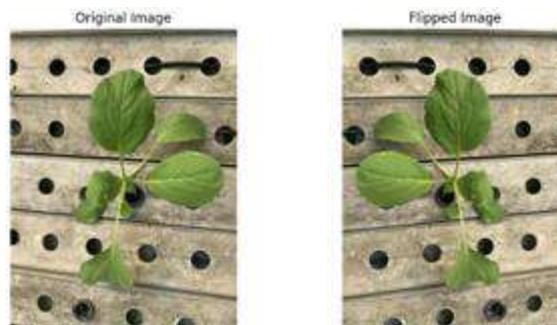
c. *Data Augmentation*

Pemberlakuan augmentasi merupakan cara dalam mengatasi *overfitting* dengan cara melakukan modifikasi pada data menjadi berbagai bentuk dengan cara melakukan berbagai teknik pada citra seperti, *horizontal flip*, *vertical flip*, *fill mode nearest*, *shear range*, *zoom range*, *rescale* dan teknik augmentasi lainnya. Berikut merupakan yang dilakukan pada augmentasi data sebagai berikut :



**Gambar 5. Proses data augmentation**

Ilustrasi proses Data Augmentation dapat dilihat pada gambar berikut.



**Gambar 6. Perbandingan hasil proses flip image**

Data yang bervariasi akan berpengaruh pada saat tahap *modelling*. Banyaknya data yang bervariasi akan meningkatkan hasil akurasi pada proses *training*. Apabila jumlah data terlalu sedikit dapat mengakibatkan terjadinya *overfitting* pada model.

## 2.4. Modelling

Pada tahap ini peneliti menerapkan deep learning untuk membuat model klasifikasi umur tanaman. Pada tahap ini dilakukan perancangan arsitektur model CNN yang akan digunakan untuk deteksi umur tanaman. Arsitektur model CNN harus dirancang sedemikian rupa agar mampu menangkap fitur-fitur penting dari citra daun yang berkaitan dengan umur tanaman. Arsitektur CNN yang digunakan adalah arsitektur VGG. Selanjutnya, dengan arsitektur tersebut akan digunakan untuk melatih model menggunakan *dataset* citra daun tanaman sawi yang telah dipersiapkan sebelumnya. Proses pelatihan dilakukan dengan memberikan citra-citra daun tanaman beserta label umur tanaman ke dalam model CNN untuk mengoptimalkan bobot-bobot neuron sesuai dengan pola-pola yang terdapat dalam data. *Hyperparameter* untuk algoritma CNN yang akan digunakan sebagaimana disajikan pada tabel 1 berikut ini.

**Tabel 1. Hyperparameter**

<i>Hyperparameter</i>	
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

Percobaan dilakukan sebanyak 8 kali dengan 2 model arsitektur yang berbeda yaitu arsitektur VGG16 dan VGG19, dengan rasio data *training*, data *validation* dan testing yang berbeda untuk mendapatkan model klasifikasi dengan nilai akurasi yang optimum.



**2.5. Evaluation**

Tahap evaluasi ditujukan untuk mengetahui kemampuan model dalam mengklasifikasikan setiap kelas dengan benar. Evaluasi dilakukan menggunakan *confusion matrix*, yang mampu mengevaluasi performa model secara lebih rinci. Dari *confusion matrix*, dapat dihitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *f1-score* untuk setiap kelas.

**3. HASIL DAN PEMBAHASAN**

**3.1. Pengumpulan Data**

Telah dikumpulkan data berupa citra sawi sebanyak 574. Data citra yang berhasil dikumpulkan berformat \*.jpg. Data citra yang dikumpulkan berasal dari foto yang diambil langsung menggunakan kamera digital dari petani sawi yaitu Jirifarm, yang ada di kabupaten Tangerang.



**Gambar 7. Contoh data citra daun sawi**

Total dari dataset dibagi kedalam 3 kelas yaitu minggu pertama, minggu ke-2 dan minggu ke-3 sebagaimana tabel 2 berikut.

**Tabel 2. Jumlah citra dan kelas**

No	Kelas citra daun sawi	Jumlah Citra
1	Minggu pertama	195
2	Minggu kedua	183
3	Minggu ketiga	196

**3.2. Analisis Data**

Data citra daun sawi dianalisis menggunakan teknik *preprocessing* seperti *resize*, *augmentasi*, dan *segmentasi*. Model *deep learning* yang digunakan adalah *Convolutional Neural Network* (CNN) dengan arsitektur VGG16 dan VGG19. Data yang telah dikumpulkan dikelompokkan sebagaimana tabel 3 berikut.

**Tabel 3. Sampel citra daun sawi berdasarkan kelas**

No	Citra	Label/Class
1		Minggu pertama
2		Minggu ke-2
3		Minggu ke-3



Dari tabel 3 terdapat total 574 data pada 3 kelas yang selanjutnya dikelompokkan menjadi data *train*, data *validation*, dan data *testing*, dengan komposisi di setiap percobaan dengan komposisi sebagaimana di tabel 4.

**Tabel 4. Komposisi pembagian dataset kedalam 3 subset di setiap percobaan**

No	Nama	Training	Validation	Testing
1	Percobaan 1 - Pembentukan Model CNN menggunakan arsitektur VGG-16	60%	20%	20%
2	Percobaan 2 - Pembentukan Model CNN menggunakan arsitektur VGG-16	70%	15%	15%
3	Percobaan 3 - Pembentukan Model CNN menggunakan arsitektur VGG-16	80%	10%	10%
4	Percobaan 4 - Pembentukan Model CNN menggunakan arsitektur VGG-19	60%	20%	20%
5	Percobaan 5 - Pembentukan Model CNN menggunakan arsitektur VGG-19	70%	15%	15%
6	Percobaan 6 - Pembentukan Model CNN menggunakan arsitektur VGG-19	80%	10%	10%
7	Percobaan 7 - Pembentukan Model CNN menggunakan arsitektur VGG-16 dengan augmentasi data	80%	10%	10%

Pada tabel 4 tersebut ditentukan jumlah data yang akan digunakan untuk *training* dan validasi dimana penentuan data ini dapat menentukan hasil dari akurasi model yang telah dibuat. Dimana semakin banyak dan bervariasi data yang dilatih maka kemungkinan model mendapatkan akurasi dengan nilai terbaik semakin tinggi.

### 3.3. Pembentukan Model

a. Percobaan 1

Hasil dari percobaan 1 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 6 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-16 terlebih dahulu dan dilakukan 1 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 5. Hyperparameter Percobaan 1**

Hyperparameter	
<i>Split Data</i>	60% Data Training, 20% Data Validation, 20% Data Testing
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

b. Percobaan 2

Hasil dari percobaan 2 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 6 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-16 terlebih dahulu dan dilakukan 1 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 6. Hyperparameter Percobaan 2**

Hyperparameter	
<i>Split Data</i>	70% Data Training, 15% Data Validation, 15% Data Testing
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

c. Percobaan 3

Hasil dari percobaan 3 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 6 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-16 terlebih dahulu dan dilakukan 1 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 7. Hyperparameter Percobaan 3**

Hyperparameter	
<i>Split Data</i>	80% Data Training, 10% Data Validation, 10% Data Testing
<i>Batch size</i>	32



<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

d. Percobaan 4

Hasil dari percobaan 4 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 19 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-19 terlebih dahulu dan dilakukan 4 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 8. Hyperparameter Percobaan 4**

<i>Hyperparameter</i>	
<i>Split Data</i>	60% Data Training, 20% Data Validation, 20% Data Testing
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

e. Percobaan 5

Hasil dari percobaan 5 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 19 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-19 terlebih dahulu dan dilakukan 4 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 9. Hyperparameter Percobaan 5**

<i>Hyperparameter</i>	
<i>Split Data</i>	70% Data Training, 15% Data Validation, 15% Data Testing
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

f. Percobaan 6

Hasil dari percobaan 6 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 19 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-19 terlebih dahulu dan dilakukan 4 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 10. Hyperparameter Percobaan 6**

<i>Hyperparameter</i>	
<i>Split Data</i>	80% Data Training, 10% Data Validation, 10% Data Testing
<i>Batch size</i>	32
<i>Layers</i>	6
<i>Epochs</i>	20
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	Default by Adam

g. Percobaan 7

Hasil dari percobaan 7 dengan input 224x224x3, maka dibentuklah arsitektur dengan total 6 *layer* dimana dilakukan *modelling* menggunakan arsitektur VGG-19 terlebih dahulu dan dilakukan 1 kali tahapan *convolution* dan *pooling*. Selanjutnya akan menjadi input untuk *flatten layer* yang mengkonversi menjadi *matrix* berukuran 1 dimensi sehingga dapat menjadi input pada *fully connected layer*.

**Tabel 11. Hyperparameter Percobaan 7**

<i>Hyperparameter</i>	
<i>Split Data</i>	60% Data Training, 20% Data Validation, 20% Data Testing



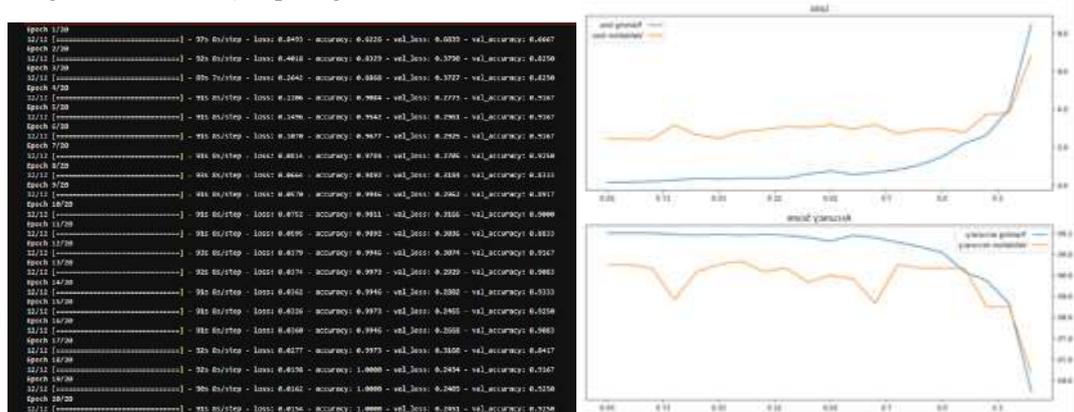
Batch size	32
Layers	6
Epochs	20
Optimizer	Adam
Learning Rate	Default by Adam

**3.4. Training Model**

Proses ini dilakukan pembentukan model atau model *fitting* yang akan digunakan untuk proses pengujian. *Hyperparameter* yang sudah ditentukan pada tahapan sebelumnya sangat menentukan nilai akurasi yang akan diperoleh.

a. Percobaan 1

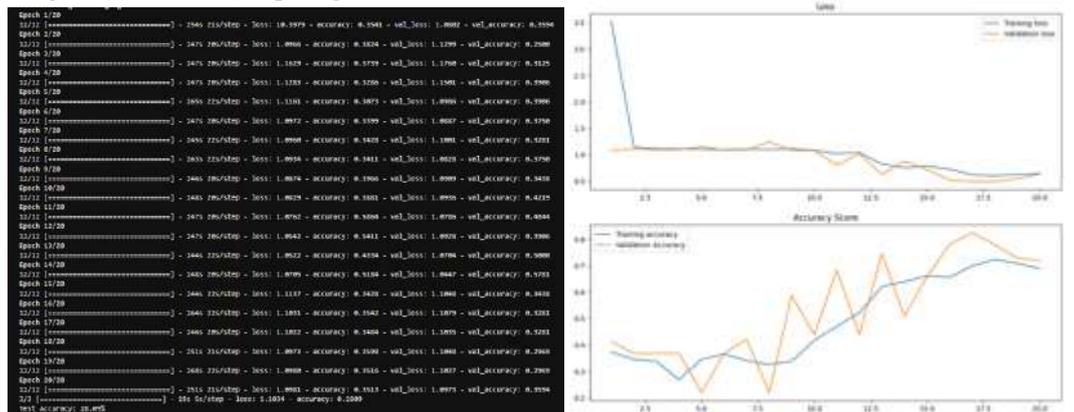
Percobaan 1 dilakukan dengan 20 *epoch* atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 78% dengan *loss*, *accuracy* seperti gambar dibawah ini.



**Gambar 8. Hasil training model percobaan 1**

b. Percobaan 2

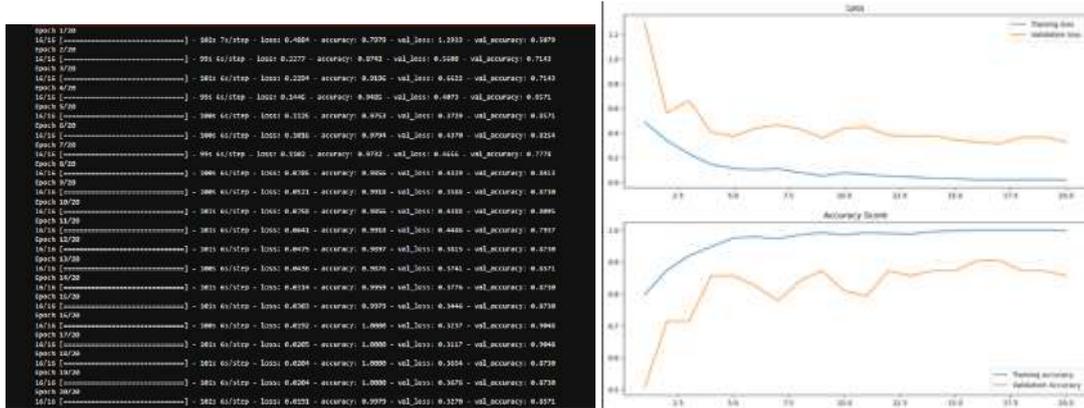
Percobaan 2 dilakukan dengan 20 *epoch* atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 43% dengan *loss*, *accuracy* seperti gambar dibawah ini.



**Gambar 9. Hasil training model percobaan 2**

c. Percobaan 3

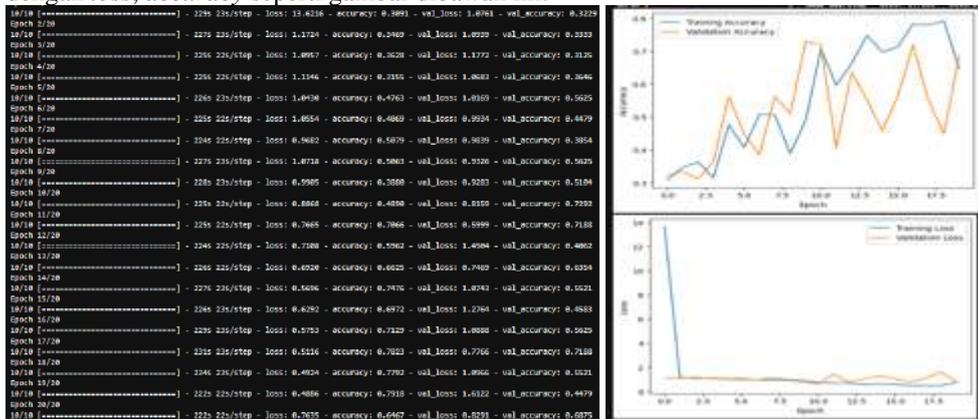
Percobaan 3 dilakukan dengan 20 *epoch* atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 76% dengan *loss*, *accuracy* seperti gambar dibawah ini.



Gambar 10. Hasil training model percobaan 3

d. Percobaan 4

Percobaan 4 dilakukan dengan 20 epoch atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 71,67% dengan loss, accuracy seperti gambar dibawah ini.



Gambar 11. Hasil training model percobaan 4

e. Percobaan 5

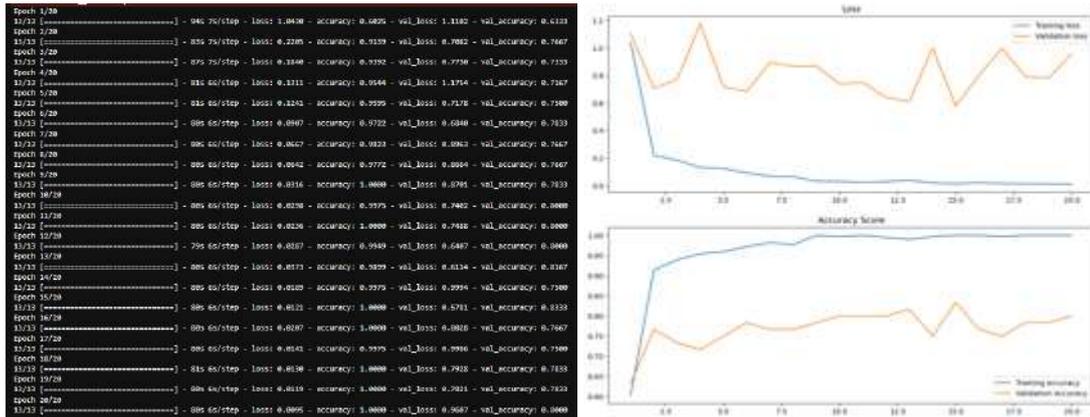
Percobaan 5 dilakukan dengan 20 epoch atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 75,86% dengan loss, accuracy seperti gambar dibawah ini.



Gambar 12. Hasil training model percobaan 5

f. Percobaan 6

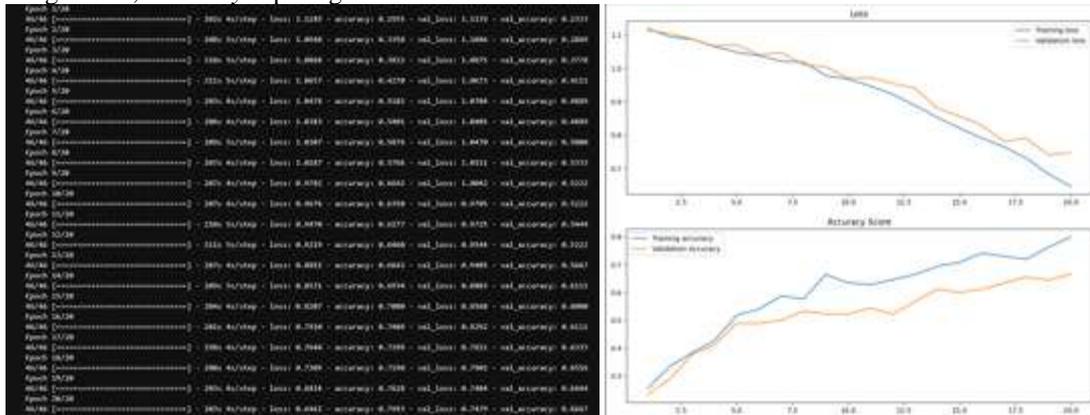
Percobaan 6 dilakukan dengan 20 epoch atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 75,86% dengan loss, accuracy seperti gambar dibawah ini.



**Gambar 13. Hasil training model percobaan 6**

g. Percobaan 7

Percobaan 6 dilakukan dengan 20 epoch atau 20 kali iterasi, pelatihan model ini menghasilkan akurasi 35% dengan loss, accuracy seperti gambar dibawah ini.



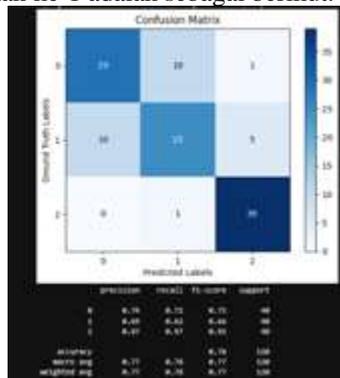
**Gambar 14. Hasil training model percobaan 7**

**3.5. Pengujian Model**

Pengujian model menggunakan *confussion matrix* di setiap percobaan.

a. Percobaan ke-1

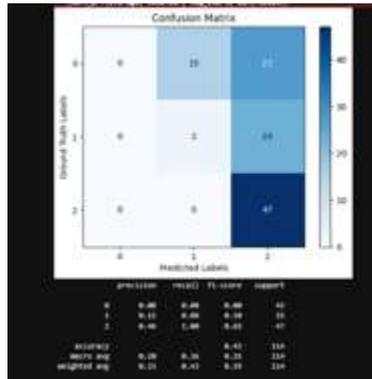
Hasil pengujian model untuk percobaan ke-1 adalah sebagai berikut.



**Gambar 15. Hasil pengujian model untuk percobaan ke-1**

b. Percobaan ke-2

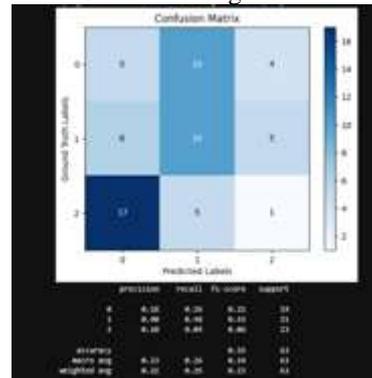
Hasil pengujian model untuk percobaan ke-2 adalah sebagai berikut.



**Gambar 16. Hasil pengujian model untuk percobaan ke-2**

c. Percobaan ke-3

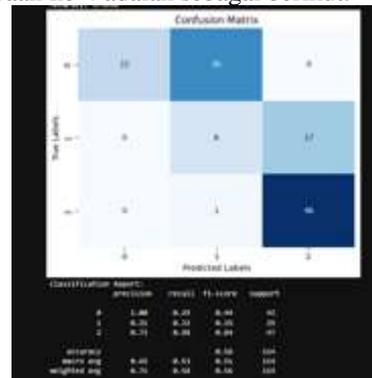
Hasil pengujian model untuk percobaan ke-3 adalah sebagai berikut.



**Gambar 17. Hasil pengujian model untuk percobaan ke-3**

d. Percobaan ke-4

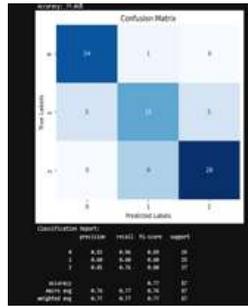
Hasil pengujian model untuk percobaan ke-4 adalah sebagai berikut.



**Gambar 18. Hasil pengujian model untuk percobaan ke-4**

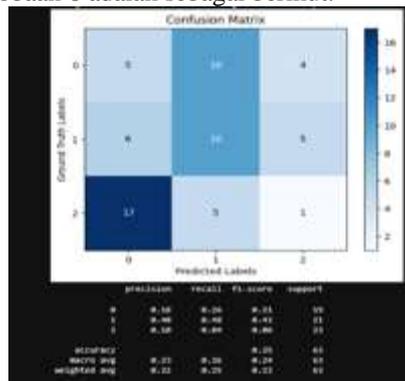
e. Percobaan ke-5

Hasil pengujian model untuk percobaan ke-5 adalah sebagai berikut.



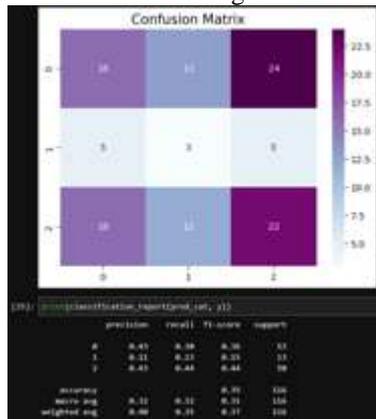
Gambar 19. Hasil pengujian model untuk percobaan ke-5

- f. Percobaan ke-6  
 Hasil pengujian model untuk percobaan 6 adalah sebagai berikut.



Gambar 20. Hasil pengujian model untuk percobaan ke-6

- g. Percobaan ke-7  
 Hasil pengujian model untuk percobaan ke-7 adalah sebagai berikut.



Gambar 21. Hasil pengujian model untuk percobaan ke-7

**3.6. Hasil analisis dan pembahasan**

Percobaan ini dilakukan dengan beragam arsitektur dengan kedalaman layer serta hyperparameter yang berbeda-beda. Adapun pembahasan hasil pembuatan arsitektur CNN seperti ditampilkan pada tabel 12 berikut.

**Tabel 12. Komparasi hasil percobaan**

No	Percobaan	Data ratio	Accuracy	Precision	Recall	F1-Score	Hasil
1	Percobaan 1	Data Training: 60% Data Validation: 20% Data Testing: 20%	78%	77%	80%	80%	Good
2	Percobaan 2	Data Training: 70% Data Validation: 15%	43%	20%	36%	25%	Underfitting



3	Percobaan 3	Data <i>Testing</i> : 15% Data <i>Training</i> : 80% Data <i>Validation</i> : 10%	25%	23%	26%	23%	<i>Underfitting</i>
4	Percobaan 4	Data <i>Testing</i> : 10% Data <i>Training</i> : 60% Data <i>Validation</i> : 20%	58%	65%	53%	51%	<i>Underfitting</i>
5	Percobaan 5	Data <i>Testing</i> : 20% Data <i>Training</i> : 70% Data <i>Validation</i> : 15%	77%	76%	77%	76%	<i>Good</i>
6	Percobaan 6	Data <i>Testing</i> : 15% Data <i>Training</i> : 80% Data <i>Validation</i> : 10%	25%	64%	53%	51%	<i>Underfitting</i>
7	Percobaan 7	Data <i>Testing</i> : 10% Data <i>Training</i> : 70% Data <i>Validation</i> : 15%	35%	32%	32%	32%	<i>Underfitting</i>

Pada tabel 11 tersebut disajikan rincian percobaan dengan mencoba berbagai eksperimen dengan arsitektur dan *hyperparameter* yang berbeda, pada percobaan 1 diperoleh bahwa dengan algoritma CNN menggunakan arsitektur VGG16 dengan data rasio 60% berupa data *training*, 20% berupa data *validation* dan 20% berupa data *testing*, dapat mendapatkan hasil dengan nilai akurasi terbaik yaitu sebesar 78%, dan pada percobaan 3 dengan algoritma CNN menggunakan arsitektur VGG16 dengan data rasio 80% merupakan data *training*, 10% merupakan data *validation* dan 10% merupakan data *testing* mendapatkan nilai akurasi terendah dengan nilai 25%. Model *deep learning* dengan algoritma CNN yang dilatih menunjukkan akurasi terbaik yaitu sebesar 78% dalam mengidentifikasi umur tanaman berdasarkan citra daun. Hasil validasi menunjukkan bahwa model ini dapat diandalkan untuk digunakan dalam sistem *smart farming*.

#### 4. KESIMPULAN

Sejumlah eksperimen dengan arsitektur dan *hyperparameter* yang berbeda telah dilakukan. Pada percobaan 1 diperoleh bahwa dengan algoritma CNN menggunakan arsitektur VGG16 dengan data rasio 60% berupa data *training*, 20% berupa data *validation* dan 20% berupa data *testing*, dapat mendapatkan hasil dengan nilai akurasi terbaik yaitu sebesar 78%, dan pada percobaan 3 dengan algoritma CNN menggunakan arsitektur VGG16 dengan data rasio 80% merupakan data *training*, 10% merupakan data *validation* dan 10% merupakan data *testing* mendapatkan nilai akurasi terendah dengan nilai 25%. Model *deep learning* dengan algoritma CNN yang dilatih menunjukkan akurasi terbaik yaitu sebesar 78% dalam mengidentifikasi umur tanaman berdasarkan citra daun. Hasil validasi menunjukkan bahwa model ini dapat diandalkan untuk digunakan dalam sistem *smart farming*.

#### UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih yang mendalam kepada Kemdikbud Republik Indonesia atas dukungan pendanaan melalui skema hibah Penelitian Dosen Pemula (PDP) tahun 2024, yang telah memberikan peran penting dalam memfasilitasi pelaksanaan penelitian ini. Selain itu, penulis menyampaikan apresiasi kepada LPPM Institut Teknologi PLN atas dukungan yang diberikan selama melaksanakan penelitian. Selanjutnya, terima kasih disampaikan kepada Jirifarm yang telah mengizinkan untuk pengambilan data penelitian berupa citra daun tanaman sawi.

#### DAFTAR PUSTAKA

- [1] J. R. Ubbens dan I. Stavness, "Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks," *Front Plant Sci*, vol. 8, Jul 2017, doi: 10.3389/fpls.2017.01190.
- [2] A. Ahmad, D. Saraswat, dan A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, hlm. 100083, 2023, doi: <https://doi.org/10.1016/j.atech.2022.100083>.
- [3] A. K. Singh, B. Ganapathysubramanian, S. Sarkar, dan A. Singh, "Deep Learning for Plant Stress Phenotyping: Trends and Future Perspectives," *Trends Plant Sci*, vol. 23, no. 10, hlm. 883–898, 2018, doi: <https://doi.org/10.1016/j.tplants.2018.07.004>.
- [4] X. Zhu, M. Zhu, dan H. Ren, "Method of plant leaf recognition based on improved deep convolutional neural network," *Cogn Syst Res*, vol. 52, hlm. 223–233, 2018, doi: <https://doi.org/10.1016/j.cogsys.2018.06.008>.
- [5] X. Guan, "A Novel Method of Plant Leaf Disease Detection Based on Deep Learning and Convolutional Neural



- Network,” dalam *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, Apr 2021, hlm. 816–819. doi: 10.1109/ICSP51882.2021.9408806.
- [6] C. L. Nazalia, P. Palupiningsih, B. Prayitno, dan Y. S. Purwanto, “Implementation of Convolutional Neural Network Algorithm to Pest Detection in Caisim,” dalam *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, 2023. doi: 10.1109/ICCoSITE57641.2023.10127792.
- [7] R. R. A. Siregar, P. Palupiningsih, I. S. Lailah, I. B. M. Sangadji, S. Sukmajati, dan N. G. Pahiyanti, “Automatic Watering Systems in Vertical Farming Using the Adaline Algorithm,” dalam *Proceedings of the International Seminar of Science and Applied Technology (ISSAT 2020)*, Atlantis Press, 2020, hlm. 429–435. doi: 10.2991/aer.k.201221.070.
- [8] P. Pravin, T. Parvati, T. Rutuja, Y. Pruthviraj, dan Z. Sudarshan, “Smart Farming Using Deep Learning,” *Int J Sci Res Sci Technol*, hlm. 371–378, Mei 2023, doi: 10.32628/IJSRST52310379.
- [9] A. G. Mote, R. Suryawanshi, P. Yalameli, dan N. Hajariwale, “SMART FARMING USING DEEP LEARNING,” *International Research Journal of Modernization in Engineering Technology and Science*, Jun 2023, doi: 10.56726/IRJMETS39923.
- [10] S. Balaji, B. N. Shivacharan, A. Nissar, dan S. V Bhaskar, “Plant Infirmity Detection Using Vgg -16 Convolutional Neural Network,” dalam *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, IEEE, Apr 2023, hlm. 485–491. doi: 10.1109/CISES58720.2023.10183541.
- [11] Hemavathi dan S. Akhila, “Deep Learning Based Approach for Plant Leaf Disease Detection for Smart Farming,” dalam *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, IEEE, Apr 2023, hlm. 496–500. doi: 10.1109/ICAECIS58353.2023.10170703.
- [12] S. D. Deb, R. K. Jha, dan S. Kumar, “ConvPlant-Net: A Convolutional Neural Network based Architecture for Leaf Disease Detection in Smart Agriculture,” dalam *2023 National Conference on Communications (NCC)*, 2023, hlm. 1–6. doi: 10.1109/NCC56989.2023.10067920.
- [13] A. Rifa, I. Sujiwanto, R. Ronggo Bintang Pratomo Prawirodirjo, dan P. Palupingsih, “Analisis Perbandingan Performa Model Klasifikasi Kesehatan Daun Tomat menggunakan Arsitektur VGG, MobileNet, dan Inception V3 Analysis Tomato Leaf Health Classification Model Performance Comparison Using VGG, MobileNet, and Inception V3,” *Jurnal Ilmu Komputer Agri-Informatika*, vol. 10, no. 1, hlm. 98–110, 2023, [Daring]. Tersedia pada: <https://jurnal.ipb.ac.id/index.php/jika>
- [14] X. Lu, “Deep Learning-Based Plant Phenotyping Framework: Analysis of Crop Life Cycle Data for Indian Farmers to Develop a Smart Agri-Field Management System,” hlm. 163–181, 2023, doi: 10.1007/978-981-99-1699-3\_11.
- [15] J. Hu, “Application of deep learning in smart agriculture research,” *Applied and Computational Engineering*, vol. 5, no. 1, hlm. 508–512, 2023, doi: 10.54254/2755-2721/5/20230630.
- [16] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, dan M. Parmar, “A review of convolutional neural networks in computer vision,” *Artif Intell Rev*, vol. 57, no. 4, Apr 2024, doi: 10.1007/s10462-024-10721-6.