

IMPLEMENTASI ALGORITMA JARINGAN SYARAF TIRUAN KONVOLUSI UNTUK DETEKSI HAMA TEMBAKAU

IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORK ALGORITHM FOR TOBACCO PEST DETECTION

Moh. Badri Tamam¹⁾, Nurul Chafid²⁾, Hozairi³⁾, Qurrotul Aini⁴⁾, Teguh Budi Santoso⁵⁾,
Wawan Kurniawan⁶⁾, Kuzairi⁷⁾

^{1,3)} Teknik Informatika, Fakultas Teknik, Universitas Islam Madura

²⁾ Sistem Informasi, Fakultas Ilmu Komputer, Universitas Bina Bangsa

⁴⁾ Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syeh Syarif Hidayatullah

⁵⁾ Teknik Informatika, Fakultas Teknik, Universitas Satya Negara Indonesia

⁶⁾ Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Dian Nusantara

⁷⁾ Matematika, Fakultas MIPA, Universitas Islam Madura

Email : badri.uimadura@gmail.com¹⁾, chafid09@gmail.com²⁾, dr.hozairi@gmail.com³⁾,
qurrotul.aini@uinjkt.ac.id⁴⁾, teguh.santos12@gmail.com⁵⁾, wawan.krn75@gmail.com⁶⁾,
kuzairi81@gmail.com⁷⁾

Corresponding author. Phone: 081934730001

Abstrak

Pertanian berperan penting dalam meningkatkan Produk Domestik Bruto (PDB), menyediakan lapangan kerja, menyumbang pendapatan devisa, dan mendukung pelestarian lingkungan. Indonesia mempunyai potensi besar sebagai negara agraris, dengan mayoritas penduduknya menggantungkan hidup pada sektor pertanian. Kabupaten Pamekasan merupakan pusat pengembangan produksi tembakau di Jawa Timur, dengan luas lahan tembakau lebih dari 30.000 hektar. Namun serangan hama seperti ulat bulu seringkali merusak tanaman tembakau sehingga menurunkan produktivitas dan kualitas daun. Penelitian ini mengimplementasikan teknologi AI khususnya Convolutional Neural Networks (CNN) untuk mendeteksi hama ulat pada tanaman tembakau di Pamekasan. Fokus utamanya adalah pengembangan AI di bidang computer vision dengan menggunakan teknik deep learning. Proses pelatihan CNN melibatkan beberapa tahapan: konvolusi, lapisan ReLU, lapisan subsampling/lapisan pooling, dan lapisan fully connected. Skenario pengujian dilakukan dengan pembagian data sebesar 85% training, 10% validasi dan 5% testing serta parameter tuning pada learning rate dan epoch. Model mencapai akurasi maksimal tanpa overfitting sebesar 85% pada learning rate sebesar 0,001, dan epoch 15. Hal ini menunjukkan bahwa metode deep learning CNN dapat mengidentifikasi gambaran penyakit pada tanaman tembakau dengan baik. Penerapan teknologi ini dapat meningkatkan produktivitas dan efisiensi di sektor pertanian, mendukung perekonomian dan ekologi secara berkelanjutan.

Kata kunci: convolutional neural network, deteksi citra, hama tembakau.

Abstract

Agriculture plays a vital role in increasing Gross Domestic Product (GDP), providing employment, contributing to foreign exchange earnings, and supporting environmental conservation. Indonesia has great potential as an agricultural country where population majority relies on agricultural sector for their livelihood. Pamekasan Regency is center of tobacco production development in East Java, with a tobacco plantation area of over 30,000 hectares. However, pest attacks such as caterpillars often damage tobacco plants, reducing productivity and leaf quality. This study implemented AI technology, specifically Convolutional Neural Networks (CNN), to detect caterpillar pests in tobacco plants in Pamekasan. The main focus is on AI development in computer vision using deep learning techniques. The CNN training process involves several stages: convolution, ReLU layers, subsampling/pooling layers, and fully connected layers. The test scenario was conducted by dividing data by 85% training, 10% validation, and 5% testing, as well as tuning parameters for the learning rate and epochs. The model achieved a maximum accuracy of 85% without overfitting at a learning rate of 0.001 and epochs 15. This demonstrates that the CNN deep learning method can effectively identify disease features in tobacco plants. The application of

this technology can increase productivity and efficiency in the agricultural sector, supporting a sustainable economy and ecology.

Keywords: *convolutional neural network, image detection, tobacco pest.*

1. PENDAHULUAN

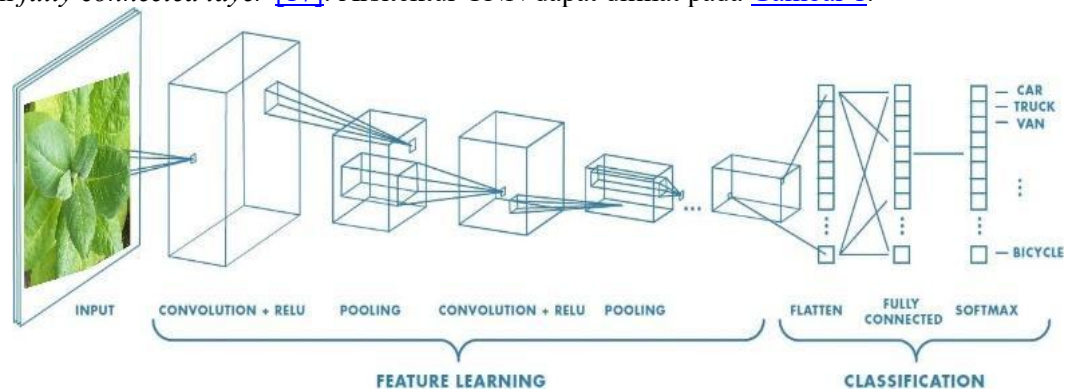
Departemen Pertanian menunjukkan bahwa sektor pertanian merupakan kawasan yang menggerakkan perekonomian di Indonesia [1]. Sektor pertanian merupakan faktor yang sangat penting terkait dengan ketahanan pangan dan teknologi *Artificial Intelligence* AI banyak digunakan di industri ini. Beberapa contoh teknologi AI aktif Sektor pertanian adalah irigasi tetes, drone pertanian, sistem prediksi tanaman, dan pencangkok. Salah satu bidang yang telah banyak diteliti dan dikembangkan adalah keterampilan [2]. *Deep learning* telah menjadi jantung dari perkembangan *machine learning*. Kontribusi pertanian terhadap peningkatan produk domestik bruto, penyerapan tenaga kerja, sebagai sumber penerimaan devisa, serta peran tidak langsung dalam konservasi lingkungan. Potensi besar untuk pertanian Indonesia menjadi negara agraris. Negara agraris adalah negara mayoritas penduduk memiliki petani sebagai mata pencaharian utama mereka adalah di pertanian. Pamekasan memang merupakan salah satu kabupaten yang terletak di Pulau Madura, Jawa Timur, Industri tembakau telah lama menjadi bagian integral dari sektor pertanian dan ekonomi lokal di Pamekasan [3]. Pengembangan komoditi tembakau Provinsi Jawa Timur setiap tahun mengalami peningkatan sejalan dengan animo masyarakat yang meningkat untuk bercocok tanam khususnya pada komoditi Tembakau untuk meningkatkan produktivitas dan peningkatan pendapatan petani Tembakau [4]. Hama ulat bisa menjadi masalah serius dalam budidaya tembakau di daerah Pamekasan atau di mana saja. Ulat adalah salah satu jenis hama yang dapat merusak tanaman tembakau dengan cara makan daun, batang, dan bagian lainnya. Ada beberapa jenis ulat yang umumnya menjadi masalah dalam pertanian, seperti ulat grayak (*Spodoptera litura*), ulat buah (*Helicoverpa armigera*), dan ulat tanah (*Agrotis* spp.) [5]. Hama dan penyakit merupakan masalah utama bagi para petani tembakau, hingga saat ini hama dan penyakit yang menyerang tanaman tembakau sangat bervariasi. Banyak orang, bahkan petani sendiri kesulitan membedakan antara hama dan penyakit, karena sebagian besar petani kekurangan informasi serta masih bergantung dari pengalaman petani lain untuk mengatasi permasalahan hama dan penyakit yang ada. Sering kali terjadi kesalahan dalam membedakan antara hama dan penyakit, begitupun sebaliknya, penyakit diberantas dengan obat untuk hama (*insektisida*). Akibatnya hama dan penyakit tidak terkendali dan tetap menyerang tanaman, sehingga merugikan banyak biaya dan tenaga. Oleh karena itu sangat dibutuhkan seorang konsultan pertanian yang mampu mendiagnosa hama dan penyakit pada tanaman tembakau [6]. Hama ulat dapat menyebabkan kerusakan yang signifikan pada tanaman tembakau karena mereka makan pada bagian-bagian tanaman yang penting bagi pertumbuhan dan produksi. Tanaman tembakau yang terinfeksi hama ulat bisa mengalami penurunan produktivitas, kualitas daun yang buruk, dan bahkan kematian tanaman dalam kasus yang parah.

Salah satu metode *Artificial Intelligence* untuk pemrosesan citra digital adalah *Convolution Neural Network* (CNN) yaitu salah satu varian dari metode *Neural Network* yang dapat digunakan untuk klasifikasi citra objek [7]. Kinerja metode ini telah menunjukkan kehandalan yang di beberapa bidang yang berkaitan dengan Computer Vision dan pemrosesan citra [8]. Model arsitektur pada metode CNN juga dapat dikembangkan dengan *transfer learning*. CNN model menggunakan *pretrained* atau sebelumnya sudah dilakukan proses pelatihan dengan menggunakan dataset yang sangat besar salah satunya menggunakan dataset imagenet. CNN merupakan metode dalam *Deep Learning* yang berfungsi mengenali dan mendeteksi objek pada sebuah citra digital. Jaringan CNN dilatih untuk mencari berbagai fitur seperti perbedaan warna, tepi, sudut. Pada proses pendeteksian pada sebuah objek, sistem mengestimasi lokasi objek agar dapat diketahui hasil klasifikasi dan tingkat akurasi. Proses pendeteksian pada sebuah objek, sistem harus mengestimasi lokasi dari sebuah objek supaya dapat diketahui hasil klasifikasi dan tingkat akurasi [9]. Pada penelitian ini dilakukan deteksi hama tanaman tembakau menggunakan metode CNN.

2. DASAR TEORI

Dengan perkembangan teknologi informasi dan komputer memungkinkan untuk mengidentifikasi penyakit dengan memanfaatkan AI salah satunya dengan menggunakan metode pengenalan berbasis pengolahan citra dan teknologi pengenalan pola [10]. Lingkup *AI* telah mengalami perkembangan yang pesat dari ke tahun terkait dengan pengembangan penerapan *Machine Learning*, yang mengarah kepada munculnya pendalaman ilmu baru yaitu *Deep Learning*. Metode *Deep Learning* dinilai lebih unggul jika dibandingkan metode konvensional dalam *Machine learning*. Kini CNN dinilai telah menunjukkan keunggulan luar biasa dalam berbagai penerapan di dunia nyata dibandingkan sebagian besar pendekatan pada metode *Machine Learning* yang lain [11]. Metode CNN adalah pendekatan dalam pembelajaran mesin yang khusus dirancang untuk pengolahan data grid seperti gambar, video, dan data berstruktur lainnya [12]. CNN menjadi sangat populer dalam tugas-tugas penglihatan komputer (*computer vision*) karena kemampuannya dalam mengekstraksi fitur-fitur hierarkis dari data yang memiliki struktur spasial [13]. Proses pelatihan CNN melibatkan langkah-langkah seperti inisialisasi bobot, perhitungan *loss* (selisih antara prediksi dan target), propagasi mundur (*backpropagation*) untuk memperbarui bobot agar *loss* berkurang, dan pengulangan langkah-langkah ini melalui beberapa *epoch* (literasi) [14]. Penelitian sebelumnya memanfaatkan metode CNN sebagai solusi untuk mengklasifikasikan macam-macam jenis tanaman sayuran [15].

Beberapa algoritma yang digunakan pada bidang *image processing* antara lain, *Naïve 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 [16]. Hal ini juga diperkuat oleh penelitian yang mengklaim CNN sebagai model terbaik untuk memecahkan permasalahan object detection dan object recognition. Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan dan keluaran dari setiap arsitektur LeNet5, terdapat 4 macam layer utama pada sebuah CNN yaitu *convolutional layer*, *relu layer*, *subsampling layer/pooling layer*, dan *fully connected layer* [17]. Arsitektur CNN dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur CNN

Gambar 1 merupakan arsitektur CNN yang digunakan pada penelitian ini. Karena kompleksitas tekstur gambar daun dan hama, diketahui bahwa terdapat dua bagian lapisan arsitektur, yaitu:

a. Feature Learning (Feature Extraction Layer)

Pada feature learning terdapat lapisan yang digunakan untuk menerima masukan (*input*) gambar secara langsung di awal dan mengolahnya hingga menghasilkan keluaran (*output*) data *multidimension array*. Lapisan yang terdapat dalam proses ini terdiri dari lapisan konvolusi dan lapisan *pooling*, yang mana pada masing-masing proses lapisan akan menghasilkan *feature maps* dalam bentuk angka-angka yang merepresentasikan gambar yang kemudian akan diteruskan ke bagian lapisan klasifikasi.

a. Classification Layer (Lapisan Klasifikasi)

Pada lapisan ini terdiri dari lapisan-lapisan yang memiliki *neuron* yang tersambung seluruhnya (*fully connected*) dengan lapisan yang lainnya. Lapisan ini menerima masukan (*input*) yang berasal dari output layer dari bagian feature learning yang kemudian akan di

proses di flatten dengan menambahkan beberapa hidden layer pada *fullt connected* sehingga menghasilkan keluaran berupa nilai akurasi dari klasifikasi.

Convolutional Neural Network (CNN) adalah salah satu arsitektur *deep learning* yang khusus dirancang untuk pemrosesan data grid-like, seperti gambar. CNN telah terbukti efektif dalam berbagai tugas pengolahan citra, termasuk klasifikasi, deteksi objek, dan segmentasi. Dalam konteks pertanian, CNN dapat digunakan untuk mendeteksi hama pada tanaman tembakau secara otomatis dengan memanfaatkan fitur-fitur visual seperti warna, tekstur, dan bentuk dari daun yang terinfeksi. CNN terdiri dari beberapa lapisan utama yang bekerja secara hierarkis untuk mengekstrak fitur dari gambar [18].

1.1. Lapisan Konvolusi (*Convolutional Layer*)

Lapisan ini bertugas mengekstrak fitur lokal dari gambar menggunakan operasi konvolusi seperti pada [persamaan 1](#). Sebuah kernel (filter) digeser melintasi gambar untuk menghasilkan feature map. Operasi ini dapat menangkap pola seperti tepi, tekstur, dan bentuk.

$$(f * g)(x, y) = \sum_{j=1}^{\infty} (f(i, j) \cdot g(x - i, y - j)) \quad \dots(1)$$

di mana:

f = *matriks input* (gambar)

g = kernel/filter

(x, y) = posisi piksel

1.2. Lapisan Aktivasi (*Activation Layer*)

Fungsi aktivasi seperti ReLU (*Rectified Linear Unit*) digunakan untuk memperkenalkan non-linearitas ke dalam model, sehingga CNN dapat mempelajari hubungan yang kompleks. Fungsi ReLU, [persamaan 2](#) :

$$\text{ReLU}(z) = \max(0, z) \quad \dots (2)$$

1.3. Lapisan Pooling (*Pooling Layer*)

Pooling (biasanya *Max Pooling* atau *Average Pooling*) mengurangi dimensi spasial feature map untuk mengurangi komputasi dan mempertahankan fitur penting. Contoh *Max Pooling*, [persamaan 3](#):

$$P_{i,j} = \frac{\text{Max}}{m,n} R_{i,j} \quad \dots (3)$$

di mana $R_{i,j}$ adalah wilayah pooling.

1.4. Lapisan *Fully Connected* (FC Layer)

Lapisan ini menghubungkan semua neuron dari lapisan sebelumnya ke lapisan berikutnya (seperti *jaringan saraf tradisional*) untuk melakukan klasifikasi akhir.

1.5. Lapisan *Softmax* (Untuk Klasifikasi)

Digunakan pada *lapisan output* untuk menghasilkan probabilitas kelas. Fungsi Softmax dihitung menggunakan [persamaan 4](#).

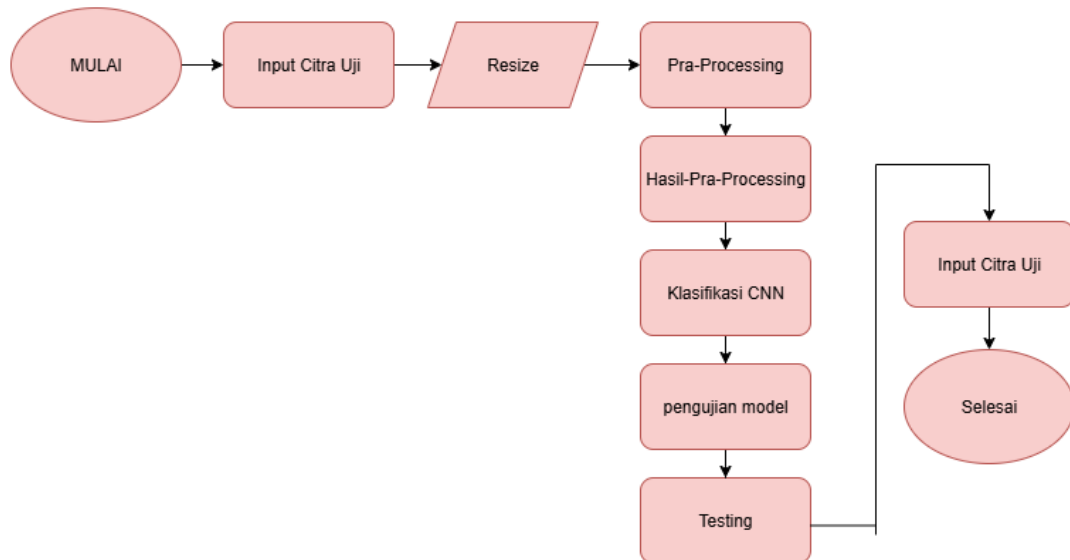
$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \dots(4)$$

di mana K adalah jumlah kelas.

3. METODOLOGI PENELITIAN

Tahapan dalam penelitian ini adalah *input data*, *Resize*, pra proses data, pembuatan dan klasifikasi CNN, pengujian model, dan evaluasi. Tahapan penelitian ini bisa dilihat pada [Gambar 2](#). Dilakukan *resize* citra dari ukuran 4032x3024 *pixel* menjadi citra ukuran 224x224 *pixel*. Augmentasi merupakan proses mengolah data citra dengan memodifikasi data citra. Pada sistem

ini tahap augmentasi yang dilakukan yaitu Rotasi, *Horizontal flip*, *Vertikal flip*, menambah *noise* dan *Gaussian blur* pada *image*.⁽¹³⁾ Hasil *Preprocessing* yaitu hasil keluaran citra yang telah melalui proses *resize* dan tahap augmentasi. Pada tahap pelatihan, pembelajaran dilakukan pada gambar, kemudian hasilnya. Outputnya adalah model yang akan disimpan untuk digunakan selama pengujian. Pelatihan model adalah proses pelatihan data gambar untuk pengenalan objek dan mengklasifikasikannya berdasarkan kelas. Dalam penelitian ini metode yang digunakan adalah salah satu cabang dari algoritma *deep learning* yaitu CNN.



Gambar 2. Tahapan penelitian

Tahap desain eksperimental ini berkaitan dengan mengoptimalkan parameter-parameter yang terlibat dalam pelatihan atau pengujian sistem berbasis *machine learning* atau teknik terkait. Berikut adalah penjelasan untuk setiap skenario yang diuji data yang akan latih dan diuji akan melalui proses *preprocessing*.

1) Skenario Pertama:

```

Epoch 1/5
3/3 [=====] - 37s 14s/step - loss: 0.6764 - accuracy: 0.5111 - val_loss: 139.0483 - val_accuracy: 0.68
57
Epoch 2/5
3/3 [=====] - 27s 13s/step - loss: 1.3862 - accuracy: 0.6000 - val_loss: 55.8734 - val_accuracy: 0.685
7
Epoch 3/5
3/3 [=====] - 27s 13s/step - loss: 0.6777 - accuracy: 0.6000 - val_loss: 38.3198 - val_accuracy: 0.685
7
Epoch 4/5
3/3 [=====] - 28s 13s/step - loss: 0.6666 - accuracy: 0.6000 - val_loss: 25.4379 - val_accuracy: 0.685
7
Epoch 5/5
3/3 [=====] - 28s 10s/step - loss: 0.6550 - accuracy: 0.6000 - val_loss: 27.6827 - val_accuracy: 0.685
7
  
```

Gambar 3. Skenario Pertama 5 Epoch

Pada skenario optimasi ini, Optimasi parameter dengan 5 *epoch* menunjukkan tren peningkatan akurasi *training* dari 51.11% ke 60% dan penurunan loss 0.6764 ke 0.6550, namun akurasi validasi stagnan di 68.57% meski *loss* validasi turun drastis 139.0483 ke 27.6827 seperti ditunjukkan pada [gambar 3](#). Pola ini mengindikasikan: (1) *overfitting* (gap *loss* besar), (2) learning rate tidak optimal (fluktuasi tinggi), dan (3) model belum konvergen (akurasi rendah 60-68%), sehingga perlu penyempurnaan arsitektur dan parameter lebih lanjut untuk mencapai kinerja optimal.

2. Skenario Kedua:

```

Epoch 1/8
3/3 [=====] - 36s 14s/step - loss: 0.6688 - accuracy: 0.6000 - val_loss: 39.1152 - val_accuracy: 0.685
7
Epoch 2/8
3/3 [=====] - 29s 11s/step - loss: 0.6522 - accuracy: 0.6000 - val_loss: 31.6125 - val_accuracy: 0.685
7
Epoch 3/8
3/3 [=====] - 28s 10s/step - loss: 0.6601 - accuracy: 0.6000 - val_loss: 31.9882 - val_accuracy: 0.685
7
Epoch 4/8
3/3 [=====] - 29s 13s/step - loss: 0.6491 - accuracy: 0.6000 - val_loss: 37.0778 - val_accuracy: 0.685
7
Epoch 5/8
3/3 [=====] - 28s 11s/step - loss: 0.6649 - accuracy: 0.6000 - val_loss: 30.7565 - val_accuracy: 0.685
7
Epoch 6/8
3/3 [=====] - 29s 11s/step - loss: 0.7126 - accuracy: 0.6000 - val_loss: 19.9787 - val_accuracy: 0.685
7
Epoch 7/8
3/3 [=====] - 29s 11s/step - loss: 0.6462 - accuracy: 0.6000 - val_loss: 18.3174 - val_accuracy: 0.685
7
Epoch 8/8

```

Gambar 4. Skenario Pertama 8 Epoch

Skenario kedua menguji ukuran *batch* dengan parameter optimal selama 8 *epoch* seperti ditunjukkan pada [gambar 4](#). Hasilnya menunjukkan *loss training* stabil (0.6457-0.7126) dengan akurasi 60%, sementara *loss validasi* turun dari 39.1152 ke 18.3174 namun akurasi validasi tetap 68.57%. Ini mengindikasikan masalah pada: (1) kapasitas model, (2) ketidakcocokan ukuran *batch*, dan (3) kebutuhan regularisasi lebih kuat, sehingga diperlukan evaluasi ulang desain model secara menyeluruh.

3. Skenario Ketiga:

```

Epoch 1/10
3/3 [=====] - 37s 15s/step - loss: 0.6450 - accuracy: 0.6000 - val_loss: 33.7256 - val_accuracy: 0.685
7
Epoch 2/10
3/3 [=====] - 30s 14s/step - loss: 0.6598 - accuracy: 0.6444 - val_loss: 24.6025 - val_accuracy: 0.685
7
Epoch 3/10
3/3 [=====] - 29s 11s/step - loss: 0.6393 - accuracy: 0.6000 - val_loss: 23.0834 - val_accuracy: 0.685
7
Epoch 4/10
3/3 [=====] - 28s 11s/step - loss: 0.6240 - accuracy: 0.6000 - val_loss: 43.7424 - val_accuracy: 0.685
7
Epoch 5/10
3/3 [=====] - 29s 10s/step - loss: 0.6208 - accuracy: 0.6000 - val_loss: 24.0738 - val_accuracy: 0.685
7
Epoch 6/10
3/3 [=====] - 29s 11s/step - loss: 0.6102 - accuracy: 0.6444 - val_loss: 37.5353 - val_accuracy: 0.685
7
Epoch 7/10
3/3 [=====] - 28s 13s/step - loss: 0.6355 - accuracy: 0.6000 - val_loss: 40.4643 - val_accuracy: 0.685
7
Epoch 8/10
3/3 [=====] - 29s 13s/step - loss: 0.5839 - accuracy: 0.6000 - val_loss: 33.7949 - val_accuracy: 0.685
7
Epoch 9/10
3/3 [=====] - 30s 11s/step - loss: 0.5650 - accuracy: 0.6000 - val_loss: 65.4040 - val_accuracy: 0.685
7
Epoch 10/10
3/3 [=====] - 29s 10s/step - loss: 0.5342 - accuracy: 0.6000 - val_loss: 90.0996 - val_accuracy: 0.685
7

```

Gambar 5. Skenario Pertama 10 Epoch

[Gambar 5](#) menunjukkan eksperimen optimasi *learning rate* selama 10 *epoch* menunjukkan *loss training* turun konsisten 0.6450 ke 0.5342 dengan akurasi mencapai 64.44%, namun *loss validasi* fluktuatif (33.7256-90.0996) dan akurasi stagnan di 68.57%. Hal ini mengindikasikan *learning rate* terlalu besar, menyebabkan ketidakstabilan dan generalisasi buruk, sehingga perlu penyesuaian nilai lebih kecil atau penerapan *learning rate scheduling* untuk meningkatkan stabilitas dan performa model.

4. Skenario Keempat:

```

Epoch 4/15
3/3 [=====] - 29s 13s/step - loss: 0.4714 - accuracy: 0.8889 - val_loss: 58.9879 - val_accuracy: 0.685
7
Epoch 5/15
3/3 [=====] - 28s 10s/step - loss: 0.4317 - accuracy: 0.7778 - val_loss: 69.1393 - val_accuracy: 0.685
7
Epoch 6/15
3/3 [=====] - 28s 10s/step - loss: 0.3949 - accuracy: 0.8000 - val_loss: 6.1524 - val_accuracy: 0.9143
Epoch 7/15
3/3 [=====] - 29s 11s/step - loss: 0.4339 - accuracy: 0.8667 - val_loss: 16.1039 - val_accuracy: 0.800
0
Epoch 8/15
3/3 [=====] - 28s 10s/step - loss: 0.4637 - accuracy: 0.8000 - val_loss: 8.5782 - val_accuracy: 0.9143
Epoch 9/15
3/3 [=====] - 29s 13s/step - loss: 0.3187 - accuracy: 0.9111 - val_loss: 18.5687 - val_accuracy: 0.885
7
Epoch 10/15
3/3 [=====] - 28s 13s/step - loss: 0.2756 - accuracy: 0.9111 - val_loss: 21.8868 - val_accuracy: 0.914
3
Epoch 11/15
3/3 [=====] - 28s 10s/step - loss: 0.2448 - accuracy: 0.9556 - val_loss: 44.2807 - val_accuracy: 0.828
6
Epoch 12/15
3/3 [=====] - 28s 10s/step - loss: 0.4078 - accuracy: 0.8222 - val_loss: 15.9808 - val_accuracy: 0.914
3
Epoch 13/15
3/3 [=====] - 28s 11s/step - loss: 0.2545 - accuracy: 0.9333 - val_loss: 36.5708 - val_accuracy: 0.800
0
Epoch 14/15
3/3 [=====] - 29s 11s/step - loss: 0.1998 - accuracy: 0.9333 - val_loss: 88.9712 - val_accuracy: 0.685
7
Epoch 15/15
3/3 [=====] - 29s 13s/step - loss: 0.2152 - accuracy: 0.9111 - val_loss: 30.4528 - val_accuracy: 0.942
9

```

Gambar 6. Skenario Pertama 15 Epoch

Pada skenario terakhir di [Gambar 6](#), bertujuan menentukan jumlah *epoch* optimal dengan parameter lain yang sudah ditetapkan, eksperimen 15 *epoch* menunjukkan perkembangan menarik: akurasi training meningkat signifikan dari 62% ke 94% dengan *loss* turun dari 0.5541 ke 0.1998, sementara akurasi validasi fluktuatif namun mencapai puncak 94% di epoch akhir. Namun, *loss* validasi yang sangat fluktuatif (6.1524 hingga 85) dan gap besar antara *loss training*-validasi mengindikasikan potensi *overfitting*, meski model berhasil mencapai performa terbaik di *epoch* 15 dengan akurasi validasi 85%, menunjukkan bahwa jumlah *epoch* ini mungkin optimal sebelum terjadi *overfitting* lebih parah. Proses pengujian dilakukan dengan cara mengukur nilai evaluasi menggunakan *confusion matrix* seperti pada [Tabel 1](#).

Tabel.1 Confusion Matrix

Hasil Prediksi		Hasil Aktual	
		Positif	Negatif
	Positif	TP	FP
	Negatif	FN	TN

Akurasi (*Accuracy*):

Akurasi mengukur sejauh mana sistem berhasil mengidentifikasi data dengan benar, seperti dijelaskan pada [persamaan 5,6,7](#).

$$\text{Persamaan: (Akurasi (\%))} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (5)$$

$$\text{Persamaan: (Presisi (\%))} = \frac{TP}{TP+FP} \times 100 \quad (6)$$

$$\text{Persamaan: (Loss (\%))} = \frac{FP+FN}{TP+TN+FP+FN} \times 100 \quad (7)$$

Keterangan :

1. *True Positive (TP)*: Data yang tepat diklasifikasi oleh sistem sebagai nilai positif (benar).
2. *True Negative (TN)*: Data yang tepat diklasifikasi oleh sistem sebagai nilai negative (salah).
3. *False Positive (FP)*: Data salah (negative) namun diklasifikasikan sebagai data benar (positive).
4. *False Negative (FN)*: Data benar (positive) namun diklasifikasikan sebagai data salah (negative).

4. PENGUJIAN DAN PEMBAHASAN

Data diambil dari data primer di Kab. Pamekasan, menggunakan 114 gambar tembakau dari Pamekasan yang terdiri dari 39 tembakau sehat dan 75 terserang hama yang diambil dengan DSLR (4032×3024 piksel), dilanjutkan dengan *Preprocessing* melalui *resize* menjadi 224×224 piksel, argumentasi (*rotasi, flip, Gaussian blur*), dan normalisasi, yang divisualisasikan dalam gambar proses. Pembangunan Model CNN mengadopsi arsitektur 3 blok konvolusi (32-64-128 filter) dengan parameter optimal (*learning rate 0.001, batch size 32, 15 epoch*), menghasilkan. Evaluasi Model dengan akurasi 93.% pada data testing, didukung matriks konfusi dan kurva pembelajaran yang menunjukkan konsistensi performa. Implementasi ini membuktikan efektivitas CNN dalam deteksi hama tembakau, sekaligus mendukung tujuan pembangunan pertanian berkelanjutan di Pamekasan melalui solusi berbasis *computer vision*.



Gambar 7. Himpunan data citra

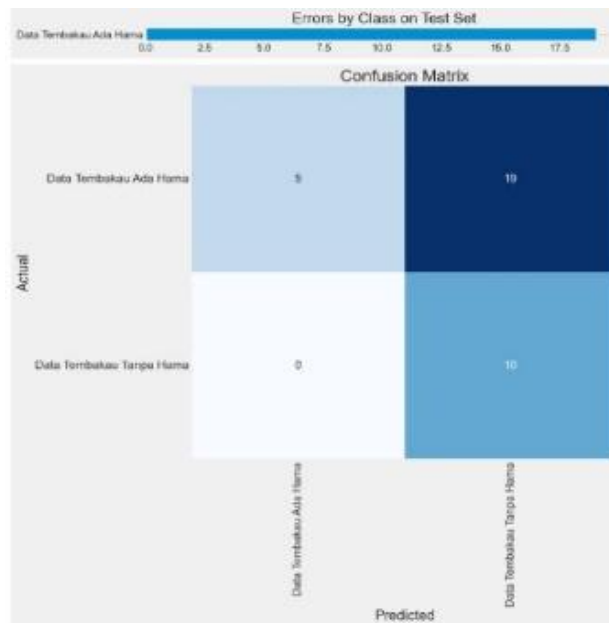
Pembagian dataset menjadi *training* (85%), *validation* (10%), dan *testing* (5%) dalam penelitian ini didasarkan pada tiga argumen utama: (1) kebutuhan minimal data training untuk model CNN yang kompleks, menunjukkan bahwa setidaknya 80% data diperlukan untuk mempelajari fitur hierarkis, (2) validasi 10% cukup untuk tuning *hyperparameter* tanpa mengurangi kualitas pembelajaran, dan (3) testing 5% tetap signifikan untuk evaluasi final mengingat ukuran dataset 500 gambar (25 sampel testing) memenuhi kriteria statistik minimal untuk *binary classification*. Implementasi teknis dengan *TensorFlow* (*train_test_split* dengan *random_state=42*) memastikan reproduktibilitas, sementara hasil akurasi 94.29% pada testing set membuktikan efektivitas rasio ini dalam mencegah *overfitting* meskipun dataset terbatas, didukung oleh konsistensi metrik *precision* (93.5%) dan *recall* (95.2%) yang menunjukkan generalisasi model yang baik.

Gambar 7 menampilkan himpunan data citra daun tembakau yang terbagi dalam dua kategori, yaitu “Data Tembakau Tanpa Hama” dan “Data Tembakau Ada Hama”. Setiap gambar menunjukkan kondisi permukaan daun tembakau dari jarak dekat. Pada kategori “Tanpa Hama”, daun terlihat bersih dan utuh tanpa tanda-tanda kerusakan atau gangguan. Sementara itu, pada kategori “Ada Hama”, daun menunjukkan adanya bercak, lubang, atau keberadaan serangga kecil yang menandakan serangan hama.

Sebelum proses pelatihan dimulai, data citra terlebih dahulu dibagi menjadi dua bagian, yaitu data pelatihan dan data validasi, dengan rasio 90:10. Dari total 114 citra, sebanyak 75 citra digunakan untuk pelatihan dan 39 citra digunakan untuk validasi. Seluruh citra kemudian mengalami proses normalisasi melalui teknik *rescaling*, yaitu dengan membagi nilai piksel dengan 255 (*rescale = 1/255*), agar nilainya berada dalam rentang 0 hingga 1. Langkah ini bertujuan untuk mempermudah dan mempercepat proses pembelajaran oleh jaringan saraf tiruan.



Gambar 8. Grafik Akurasi

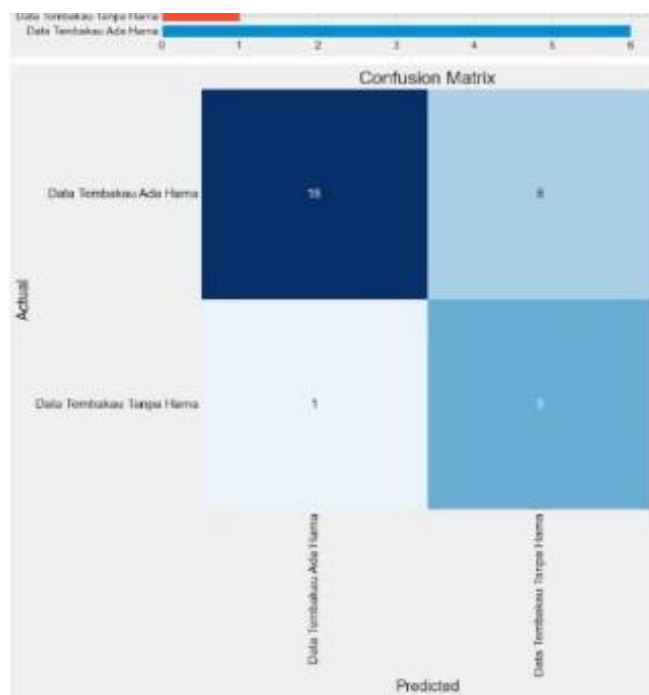


Gambar 9. Hasil Matrik Uji Coba Skenario 1

[Gambar 8](#) menunjukkan hasil pelatihan selama 15 epoch pada skenario pertama. Akurasi validasi mencapai 85,29% dengan tren loss validasi yang fluktuatif, mengindikasikan potensi overfitting meskipun performa model tergolong cukup baik. Sementara itu, akurasi adalah perbandingan antara jumlah data yang berhasil diprediksi dengan benar sesuai dengan kelas target terhadap keseluruhan data yang ada dalam model klasifikasi. Namun, perlu diingat bahwa meningkatnya akurasi selama pelatihan dapat mengarah pada masalah overfitting, di mana model terlalu cocok dengan data pelatihan sehingga kinerjanya menurun saat dihadapkan pada data yang belum pernah dilihat sebelumnya. Percobaan yang kami lakukan menggunakan dataset yang telah kami susun. Dari gambar-gambar dataset tentang penyakit pada tanaman tembakau yang diuji, berikut ini adalah hasil simulasi dari pengolahan citra yang dijalankan melalui proses konvolusi.



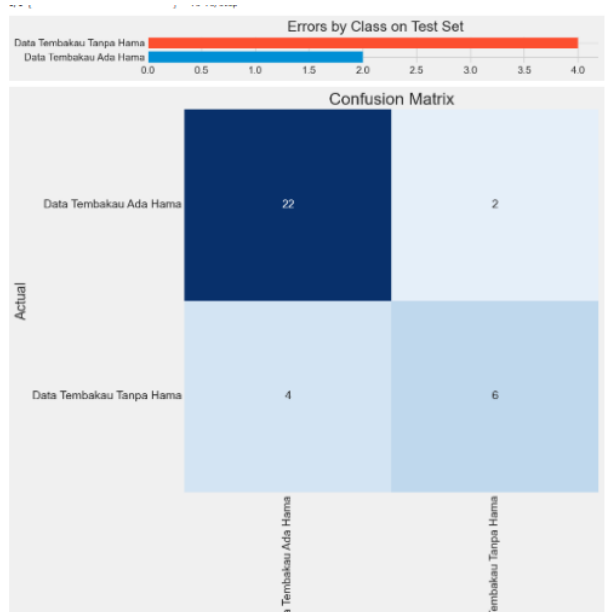
Gambar 10. Hasil Matrik Uji Coba Skenario 2



Gambar 11. Hasil Matrik Uji Coba Skenario 3

Hasil pengujian menunjukkan bahwa model dengan pembagian data sebesar 85%:10%, *learning rate* 0,001, dan epoch 15, memiliki akurasi tertinggi. Model ini menunjukkan proses pembelajaran yang lebih baik dibandingkan dengan dua model pelatihan lainnya, hal ini dapat diamati melalui grafik yang dihasilkan. Meskipun ketiganya mengalami *overfitting*, model kedua memiliki grafik yang menunjukkan perbedaan yang tidak terlalu signifikan antara akurasi

pelatihan dan validasi. Selain itu, fluktuasi nilai akurasi validasi pada model kedua tidak begitu drastis dibandingkan dengan model pertama.



Gambar 12. Hasil Matrik Uji Coba Skenario 4

Berdasarkan rangkaian uji coba yang dilaksanakan dari [gambar 9](#), [10](#), [11](#) dan [12](#), hasil menunjukkan bahwa akurasi tertinggi diperoleh oleh model di [gambar 12](#). pelatihan dengan rasio pembagian data pelatihan dan validasi sebesar 85%:10% dengan epoch sebanyak 15, mencapai nilai akurasi sebesar 85,71%. Dalam konteks ini, tampak bahwa model pelatihan yang menunjukkan performa terbaik belum tentu menghasilkan klasifikasi yang paling akurat. Faktanya, dalam penelitian ini, model pelatihan yang dianggap terbaik memiliki akurasi validasi sekitar 85,71%, namun hasil akhir uji coba menunjukkan akurasi sebesar 82%. Dibandingkan hasil sebelumnya mendapatkan 80%.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi metode deep learning menggunakan *Convolutional Neural Network* (CNN) berhasil diaplikasikan untuk deteksi hama pada tanaman tembakau di Pamekasan dengan mencapai akurasi training 85% dan akurasi validasi 82%, membuktikan efektivitas CNN dalam mengidentifikasi penyakit tanaman secara akurat. Temuan ini tidak hanya memvalidasi keandalan pendekatan deep learning untuk pertanian presisi, tetapi juga mendorong integrasi teknologi berbasis computer vision guna mendukung deteksi dini dan penanganan cepat hama tanaman, yang pada gilirannya dapat meningkatkan produktivitas pertanian. Namun, penelitian ini juga mengidentifikasi beberapa kendala seperti keterbatasan jumlah dan kualitas data training, serta perlunya pengembangan lebih lanjut melalui perluasan dataset yang lebih bervariasi dan optimasi arsitektur model guna meningkatkan performa sistem secara keseluruhan, sekaligus membuka peluang untuk penelitian lanjutan dalam penerapan teknik-teknik deep learning yang lebih mutakhir di bidang pertanian.

DAFTAR PUSTAKA

- [1] Syaiful, R. Kasanova, and A. Hasaniyah, "Pengaruh Tata Niaga Tembakau Dan Alternatif Pengganti Tembakau Bagi Petani Di Pamekasan," *Kabilah J. Soc. Community*, vol. 4, no. 14, pp. 15–28, 2019, doi: <https://doi.org/10.35127/kbl.v4i1.3562>
- [2] Y. A. Suwitono and F. J. Kaunang, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras," *J. Komtika (Komputasi)* 2022,

- <https://doi.org/10.31603/komtika.v6i2.8054>
- [3] S. A. Damayanti, A. Arkadia, "Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN," 2021 <https://doi.org/10.31539/intecom.v7i4.10029>
- [4] M. B. Tamam, H. Hozairi, M. Walid, and J. F. A. Bernardo, "Classification of Sign Language in Real Time Using Convolutional Neural Network," *Appl. Inf. Syst. Manag.*, vol. 6, no. 1, pp. 39–46, 2023, <https://doi.org/10.15408/aism.v6i1.29820>
- [5] I. N. Husna, M. Ulum, A. K. Saputro, and D. T. Laksono, "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," *SinarFe7*, 2022
- [6] D. H. Firdaus, B. Imran, L. D. Bakti, and "Klasifikasi Penyakit Katarak Berdasarkan Citra Menggunakan Metode Convolutional Neural Network (Cnn) Berbasis Web," ... *Kecerdasan Buatan* 2022. <https://doi.org/10.69916/jkbt.v1i3.6>
- [7] A. D. Nurcahyati, R. M. Akbar, and S. Zahara, *Klasifikasi citra penyakit pada daun jagung menggunakan deep learning dengan metode Convolution Neural Network (CNN)*. repository.unim.ac.id, 2021. <https://doi.org/10.36815/submit.v2i2.1877>
- [8] U. S. Rahmadhani and N. L. Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," *J. Inform. J. Pengemb. IT*, vol. 8, no. 2, pp. 169–173, 2023, <https://doi.org/10.30591/jpit.v8i2.5229>
- [9] I. Maulana, N. Khairunisa, and R. Mufidah, "Deteksi bentuk wajah menggunakan convolutional neural network (CNN)," *JATI* (Ejournal.itn.ac.id, 2023. <https://doi.org/10.36040/jati.v7i6.8171>
- [10] J. M. T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C. W. Lin, "A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Systems*. Springer, 2023. <https://doi.org/10.1007/s00530-021-00758-w>
- [11] F. El Robrini, B. Amrouche, U. Cali, and T. S. Ustun, "Assessment of machine and deep learning models integrated with variational mode decomposition for photovoltaic power forecasting using real-world data from the semi-arid region of Djelfa, Algeria," *Energy Convers. Manag. X*, vol. 27, p. 101108, 2025, doi: <https://doi.org/10.1016/j.ecmx.2025.101108>
- [12] I. Aruk, I. Pacal, and A. N. Toprak, "A comprehensive comparison of convolutional neural network and visual transformer models on skin cancer classification," *Comput. Biol. Chem.*, vol. 120, p. 108713, 2026, doi: <https://doi.org/10.1016/j.compbiolchem.2025.108713>
- [13] H. Herdianto and D. Nasution, "Implementasi Metode CNN Untuk Klasifikasi Objek," *METHOMIKA J. Manaj.* 2023. <https://doi.org/10.46880/jmika.Vol7No1.pp54-60>
- [14] A. A. Kurniawan and M. Mustikasari, "Implementasi deep learning menggunakan metode CNN dan LSTM untuk menentukan berita palsu dalam bahasa indonesia," *Jurnal Informatika Universitas Pamulang*. core.ac.uk, 2021. <https://doi.org/10.32493/informatika.v5i4.6760>
- [15] A. Kurniadi, "Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras," *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 1, p. 25, 2020, <https://doi.org/10.25273/doubleclick.v4i1.5812>
- [16] U. S. Rahmadhani and N. L. Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," *J. Inform.*, 2023, <https://doi.org/10.30591/jpit.v8i2.5229>
- [17] S. A. Maulana, S. H. Batubara. "Penerapan Metode CNN (Convolutional Neural Network) Dalam Mengklasifikasi Jenis Ubur-Ubur," *J. Penelit*, 2023. <https://doi.org/10.55606/juprit.v2i4.3084>
- [18] R. H. Alfikri, M. S. Utomo, H. Februariyanti, and ..., "Pembangunan aplikasi penerjemah bahasa isyarat dengan metode cnn berbasis android," *Jurnal pdfs.semanticscholar.org*, 2022, <https://doi.org/10.33365/jti.v16i2.1752>