

# Identifikasi Penyakit Bercak Daun Kelapa Sawit Menggunakan Algoritma CNN dengan Arsitektur VGG19 Berbasis Citra Digital

Arif Fadllullah<sup>1\*</sup>, Sri Erdina<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Komputer, Fakultas Teknik Komputer, Universitas Borneo Tarakan, Indonesia

## Info Artikel

### Riwayat Artikel:

Diterima : **6 November 2025**

Direvisi : **18 Desember 2025**

Diterbitkan : **31 Desember 2025**

### Kata Kunci:

Convolutional Neural Network (CNN)

VGG19

Penyakit Daun Kelapa Sawit

Klasifikasi Citra

Sistem Deteksi Dini

## ABSTRAK

Penelitian ini bertujuan untuk mengembangkan sistem identifikasi penyakit bercak daun pada tanaman kelapa sawit berbasis citra digital dengan memanfaatkan algoritma CNN (*Convolutional Neural Network*) menggunakan arsitektur VGG19. Dataset yang digunakan terdiri atas 330 citra daun kelapa sawit primer yang dikategorikan ke dalam tiga kelas, yaitu daun terinfeksi karat daun, daun sehat, dan daun terinfeksi *curvularia* dengan proporsi pembagian dataset menjadi 64% data latih, 16% data validasi, dan 20% data uji. Proses pengembangan sistem meliputi tahap *preprocessing*, augmentasi data, *split* data, pelatihan model dengan arsitektur VGG19 dan evaluasi model. Hasil pelatihan model sebanyak 200 epoch menunjukkan akurasi sebesar 0,93 pada data latih dan 0,98 pada data validasi. Kemudian evaluasi model terhadap data uji menghasilkan nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 0,94, 0,81, dan 0,87 untuk kelas "Karat Daun", 0,84, 0,95, dan 0,89 untuk kelas "Daun Sehat", serta 1,00 untuk kelas "Curvularia". Hasil pengujian menunjukkan performa yang konsisten, sehingga sistem usulan dinilai efektif dalam melakukan klasifikasi penyakit bercak daun kelapa sawit. Sistem usulan yang dikembangkan berpotensi digunakan sebagai alat bantu deteksi dini penyakit bercak daun guna mendukung peningkatan produktivitas tanaman kelapa sawit.

## Keywords:

Convolutional Neural Network (CNN)

VGG19

Oil Palm Leaf Disease

Image Classification

Early Detection System

## ABSTRACT

*This study aims to develop a digital image-based system for identifying leaf spot diseases in oil palm plants using the CNN (Convolutional Neural Network) algorithm with the VGG19 architecture. The dataset consists of 330 primary oil palm leaf images categorized into three classes: leaves infected with leaf rust, healthy leaves, and leaves infected with curvularia. The dataset was divided into 64% training data, 16% validation data, and 20% testing data. The system development process includes preprocessing, data augmentation, data splitting, model training using the VGG19 architecture, and model evaluation. The training results over 200 epochs achieved an accuracy of 0.93 on the training data and 0.98 on the validation data. Model evaluation on the test data produced precision, recall, and F1-score values of 0.94, 0.81, and 0.87 for the "Leaf Rust" class; 0.84, 0.95, and 0.89 for the "Healthy Leaf" class; and 1.00 for the "Curvularia" class. The testing results indicate consistent performance, suggesting that the proposed system is effective in classifying oil palm leaf spot diseases. The developed system has the potential to be used as an early detection tool for leaf spot diseases to support the improvement of oil palm productivity.*

## Penulis Korespondensi:

Arif Fadllullah,

Jurusan Teknik Komputer

Universitas Borneo Tarakan

Email: [arif.fadl@borneo.ac.id](mailto:arif.fadl@borneo.ac.id)

This is an open access article under the [CC BY-SA](#) license



## 1. PENDAHULUAN

Kelapa sawit (*Elaeis guineensis*), yang dikenal sebagai “emas hijau tropis”, merupakan komoditas penting bagi perekonomian Indonesia dan negara tropis lainnya. Hal ini dikarenakan kelapa sawit merupakan tanaman industri yang digunakan sebagai bahan baku minyak nabati, minyak industri, dan bahan bakar yang dapat berkontribusi signifikan terhadap pendapatan nasional dan kesejahteraan masyarakat [1], sepanjang pengelolaannya dilakukan secara berkelanjutan sehingga manfaat ekonomi yang diperoleh lebih besar dibandingkan dampak dan biaya kerusakan lingkungan yang ditimbulkan [2][3]. Tanaman yang berasal dari Afrika Barat ini telah berhasil beradaptasi di Indonesia, menjadikan negara ini salah satu produsen minyak sawit terbesar dunia dan berkontribusi signifikan terhadap perekonomian melalui ekspor produk turunannya. Keberhasilan pengembangan kelapa sawit di Indonesia tidak terlepas dari kesesuaian kondisi agroklimat wilayah tropis yang mendukung pertumbuhan tanaman secara optimal. Faktor lingkungan menjadi aspek penting yang menentukan produktivitas, kesehatan tanaman, serta kualitas hasil panen kelapa sawit.

Pertumbuhan tanaman ini sangat dipengaruhi oleh kondisi lingkungan, seperti iklim tropis, curah hujan 2.000–2.500 mm/tahun [4], suhu rata-rata 24–28°C dengan kelembapan udara optimal yang ideal adalah 80–90% [5], pH tanah 4–6, serta nutrisi dan drainase yang memadai. Kondisi tersebut tidak hanya mendukung produksi minyak sawit tetapi juga berperan dalam menjaga keseimbangan ekosistem sawit [6]. Namun demikian, kondisi lingkungan yang optimal tidak sepenuhnya menjamin keberhasilan produksi apabila tanaman mengalami gangguan biotik, khususnya serangan penyakit. Oleh karena itu, pengelolaan kesehatan tanaman menjadi faktor penting dalam menjaga produktivitas dan keberlanjutan perkebunan kelapa sawit.

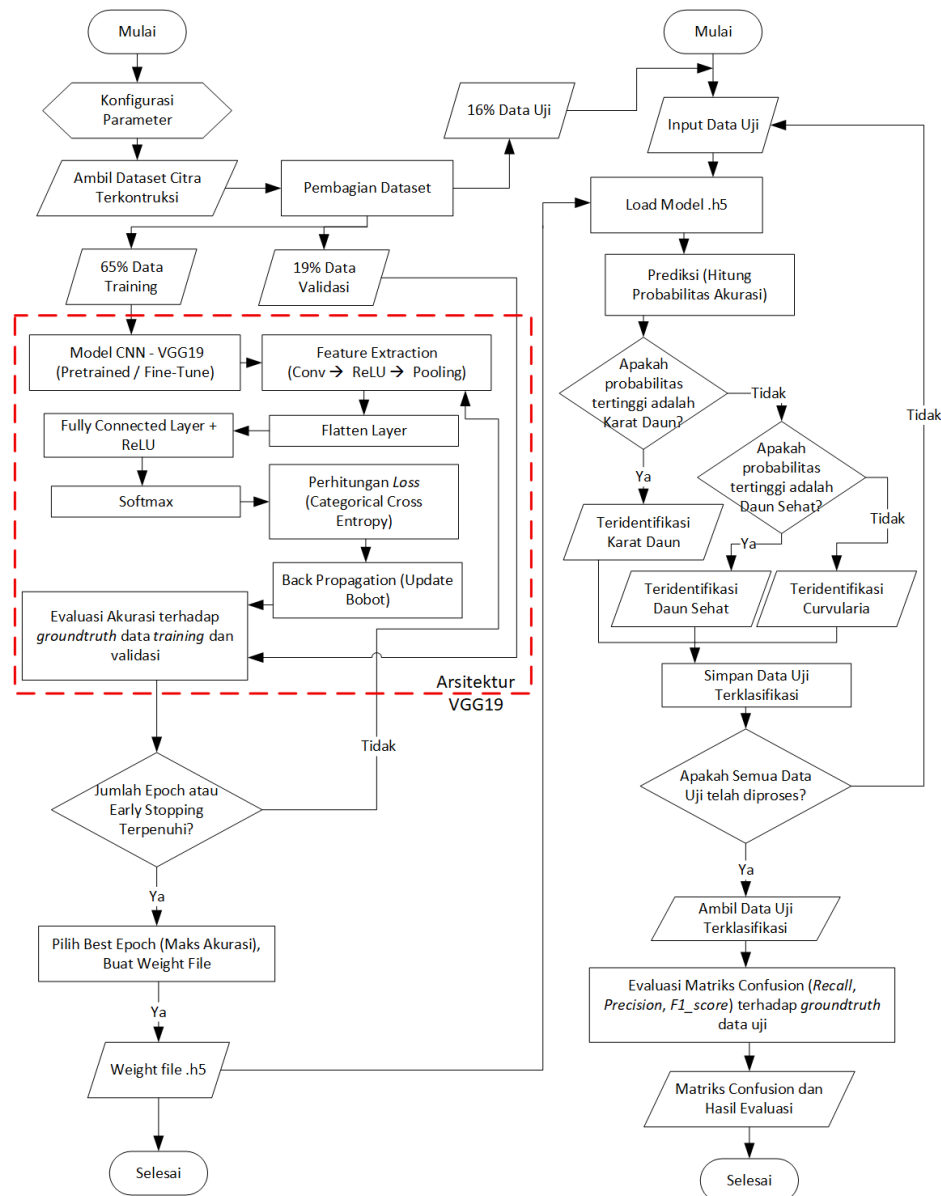
Penyakit tanaman merupakan faktor utama yang dapat menurunkan kualitas dan hasil produksi kelapa sawit. Penyakit dapat disebabkan oleh jamur, bakteri, virus, maupun faktor lingkungan, dan sering menyerang daun, batang, akar, atau buah. Gejala umum terlihat pada bercak daun, daun keriting, layu, atau perubahan warna menjadi cokelat [7]. Pengamatan manual terhadap gejala penyakit umumnya dilakukan secara langsung di lapangan. Namun, metode ini memerlukan waktu yang cukup lama, sangat bergantung pada pengalaman dan subjektivitas pengamat, serta menjadi kurang efisien apabila diterapkan pada areal perkebunan yang luas dengan jumlah tanaman yang banyak, sehingga berisiko terjadi kesalahan identifikasi. Penelitian sebelumnya oleh Rasywir, dkk [8] menunjukkan potensi penggunaan *Convolutional Neural Network* (CNN) dalam mendiagnosis penyakit kelapa sawit. Penelitian sebelumnya oleh Rasywir, dkk [8] menunjukkan potensi penggunaan *Convolutional Neural Network* (CNN) dalam mendiagnosis penyakit kelapa sawit. Dengan menganalisis 2.490 citra yang dikategorikan ke dalam 11 jenis penyakit, penelitian tersebut berhasil mencapai akurasi rata-rata 0,87. Namun, dalam penelitian tersebut, fokus penyakit yang diteliti masih bersifat umum, tanpa menitikberatkan pada indikator awal serangan penyakit pada daun kelapa sawit yang paling utama berupa bercak daun.

Oleh karena itu, penelitian ini mengusulkan pengembangan sistem identifikasi penyakit bercak daun kelapa sawit berbasis citra daun menggunakan algoritma CNN (*Convolutional Neural Network*) di Desa Binai. Sistem yang diusulkan memanfaatkan kemampuan CNN dengan arsitektur VGG19 untuk mengekstraksi fitur warna, tekstur, dan pola visual pada daun secara otomatis. Arsitektur VGG19 merupakan salah satu varian yang memiliki total 19 lapisan yang memiliki bobot (*trainable layers*), terdiri atas 16 lapisan konvolusi, dan 3 lapisan FC (*fully connected*). Arsitektur ini dirancang untuk mendeteksi pola visual dengan kedalaman yang tinggi menggunakan filter kecil berukuran 3×3, sehingga dapat mendeteksi jenis penyakit bercak daun kelapa sawit secara otomatis sehingga mampu mendeteksi gejala penyakit bercak daun secara lebih dini dan akurat sebagai dasar pengambilan keputusan dalam pengelolaan tanaman sawit. Dengan penerapan teknologi *deep learning* berbasis CNN dengan arsitektur VGG19, diharapkan sistem ini dapat meningkatkan efisiensi identifikasi penyakit, mendukung langkah penanganan tepat sasaran, mengurangi risiko penyebaran penyakit bercak daun kelapa sawit, serta meningkatkan produktivitas dan keberlanjutan industri kelapa sawit, khususnya di Desa Binai [1][7][8].

## 2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan memanfaatkan citra daun kelapa sawit sebagai data utama. Tahapan penelitian meliputi data *collection*, *preprocessing*, augmentasi citra, dan *split* data,

serta pelatihan model menggunakan CNN (*Convolutional Neural Network*), dan mengevaluasi kinerja model dalam mengidentifikasi penyakit bercak daun kelapa sawit.



Gambar 1. Flowchart Perancangan Sistem

## 2.1 Data Collection

Dataset terdiri dari 330 citra daun kelapa sawit yang dikumpulkan secara primer di Desa Binai, Kalimantan Utara dan terbagi ke dalam tiga kelas: "Daun Sehat", "Karat Daun", dan "Terinfeksi Curvularia". Kelas Daun Sehat merupakan jenis kelas penyakit daun tanpa gejala penyakit, hijau normal, dan tidak terdapat bercak daun. Kelas Karat Daun merupakan jenis kelas penyakit daun yang disebabkan oleh alga parasit *Cephaleuros virescens* yang ditandai dengan adanya bercak kecil hingga sedang, berwarna coklat-jingga seperti karat [9]. Sedangkan kelas Daun Terinfeksi Curvularia adalah jenis kelas penyakit daun yang disebabkan oleh jamur *Curvularia* sp., ditandai dengan bercak gelap tidak beraturan yang dapat menyebar dan menyebabkan kerusakan jaringan daun [10].

## 2.2 Konstruksi Data

Setelah dataset berhasil dikoleksi, maka selanjutnya dilakukan proses konstruksi data yang bertujuan untuk membersihkan data agar siap di-*training*. Beberapa proses konstruksi data yang dilakukan diantaranya: *preprocessing* mencakup *resize* seluruh citra primer ke ukuran  $224 \times 224$  piksel sebagai ukuran input standar arsitektur VGG19 [11], pelabelan sesuai kategori, *cropping* untuk menghilangkan latar belakang yang tidak

relevan, serta augmentasi data (*rotasi, flipping, zoom*) untuk menambah variasi dataset dan menghindari *overfitting* [12]. Hasil dari *preprocessing* dan augmentasi menghasilkan dataset terkonstruksi yang siap digunakan untuk pemodelan dan pengujian yang dibagi menjadi 64% untuk data latih, 16% data validasi, dan 20% data uji guna memastikan proses pembelajaran model berlangsung optimal untuk jumlah dataset sedikit (kurang dari 1000 citra) yang meminimalkan *overfitting*, serta menghasilkan evaluasi yang objektif terhadap performa model.

### 2.3 Training Model dengan CNN-VGG19

Proses pelatihan dilakukan menggunakan algoritma CNN (*Convolutional Neural Network*) dengan arsitektur VGG19 melalui metode *transfer learning* terhadap data latih yang mencakup tiga kelas yaitu daun sehat, karat daun, dan curvularia [8]. Gambar 1 menunjukkan tahapan perancangan sistem yang dimulai dari konfigurasi parameter pelatihan, seperti nilai *learning rate* ( $\eta$ ), jumlah *epoch*, ukuran *batch*, serta pemilihan fungsi aktivasi dan optimizer. Selanjutnya mengambil dataset hasil konstruksi untuk kemudian dibagi ke dalam 64% data *training*, 16% data validasi, dan 20% data uji. Setelah pembagian dataset selesai, berikutnya membuat *training* model menggunakan arsitektur VGG19 terhadap data *training* untuk memperbaharui bobot (*weight*) dengan data validasi sebagai evaluasi selama proses *training*. Hasil dari *training* model inilah yang kemudian berupa file pembobotan (.h5) yang dapat di-*loading* selanjutnya untuk diuji terhadap data uji. Lebih lanjut penjabarannya dijelaskan dengan alur sebagai berikut:

#### 2.3.1 Input Layer

Citra masukan hasil konstruksi diubah menjadi bentuk tensor berdimensi seperti pada Persamaan 1.

$$X_{raw} \in \mathbb{R}^{H \times W \times C} \quad (1)$$

Keterangan:

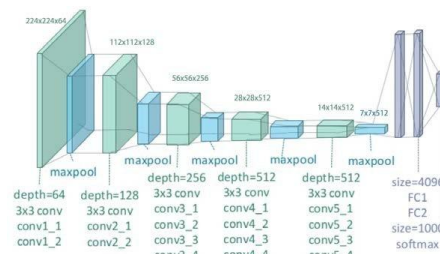
- $X_{raw}$  : tensor input citra hasil konstruksi
- $H$  : tinggi citra (224 piksel),
- $W$  : lebar citra (224 piksel),
- $C$  : jumlah kanal warna (3 untuk RGB).
- $\mathbb{R}$  : setiap elemen bernilai bilangan riil.

Sebelum masuk ke lapisan pertama, untuk versi Keras/TensorFlow setiap piksel dinormalisasi ke rentang  $[-123, +132]$  untuk mempercepat konvergensi pelatihan seperti yang ditunjukkan pada Persamaan 2, yang mana nilai *mean* diperoleh dari dataset ImageNet dengan tujuan menyesuaikan distribusi data agar sama dengan data pelatihan model VGG19 asli (*zero-centered input*) [13] [14].

$$X = X_{raw} - [123.68, 116.779, 103.939] \quad (2)$$

Angka  $[123.68, 116.779, 103.939]$  pada VGG19 bukan angka acak, melainkan nilai rata-rata (*mean*) kanal warna RGB dari dataset ImageNet, yaitu dataset yang digunakan untuk melatih VGG19 pertama kali [15].

#### 2.3.2 Convolution Layer



Gambar 2. Arsitektur VGG19 [16]

Lapisan konvolusi berfungsi mengekstraksi fitur lokal dari citra ternormalisasi ( $X$ ) menggunakan *filter kernel* berukuran  $3 \times 3$ . Operasi konvolusi dinyatakan dengan Persamaan 3.

$$Z_{i,j,k} = \sum_{m=1}^M \sum_{n=1}^N \sum_{c=1}^C X_{(i+m),(j+n),c} \cdot W_{m,n,c,k} + b_k \quad (3)$$

Keterangan:

- $X$  : citra input hasil normalisasi
- $W$  : bobot kernel
- $k$  : jumlah kanal (*depth*/jumlah filter)
- $b_k$  : bias untuk filter ke- $k$ ,
- $Z_{i,j,k}$  : nilai hasil konvolusi pada posisi  $(i, j)$  untuk kanal ke- $k$ .
- $M, N$  : tinggi dan lebar kernel

VGG19 menggunakan pola konvolusi berulang dengan jumlah *feature map* bertambah secara bertahap  $[64, 64] \rightarrow [128, 128] \rightarrow [256, 256, 256, 256] \rightarrow [512, 512, 512, 512] \rightarrow [512, 512, 512, 512]$ , seperti yang ditunjukkan pada Gambar 2 [11]. Hal ini memungkinkan model menangkap pola sederhana di lapisan awal dan pola kompleks di lapisan lebih dalam.

### 2.3.3 Fungsi Aktivasi RELU (*Rectified Linear Unit*) dan *Max Pooling Layer*

Setiap hasil konvolusi kemudian dilewatkan ke fungsi aktivasi ReLU (*Rectified Linear Unit*) untuk menambahkan non-linearitas dan menghindari *vanishing gradient*. Setelah beberapa lapisan konvolusi hasil aktivasi RELU, dilakukan *max pooling* untuk mereduksi ukuran fitur spasial dari *feature map* tanpa kehilangan informasi penting [17]. Fungsi aktivasi RELU dan *max pooling* Layer dapat dilihat pada Persamaan 4 dan 5

$$A_{i,j,k} = f(Z_{i,j,k}) = \max(0, Z_{i,j,k}) \quad (4)$$

$$P_{i,j,k} = \max_{(m,n) \in R} A_{s \cdot (i+m), s \cdot (j+n), k} \quad (5)$$

keterangan:

- $A_{i,j,k}$  : nilai aktivasi RELU pada posisi  $(i, j)$  untuk kanal ke- $k$ .
- $R$  : area *pooling* (biasanya  $2 \times 2$ )
- $s$  : *stride* (biasanya 2)
- $P_{i,j,k}$  : nilai *max pooling* pada posisi  $(i, j)$  untuk kanal ke- $k$ .

Persamaan 4 menunjukkan bahwa fungsi ReLU hanya akan mengaktifkan nilai hasil konvolusi yang bernilai positif, sedangkan nilai negatif akan diubah menjadi nol. Mekanisme ini memungkinkan jaringan saraf untuk belajar representasi fitur yang lebih kompleks dan efisien, serta mempercepat proses konvergensi selama pelatihan. Sedangkan Persamaan 5 menggambarkan proses *max pooling*, yaitu pengambilan nilai maksimum dari suatu area lokal  $R$  pada *feature map*. Operasi ini bertujuan untuk memperkecil ukuran dimensi spasial fitur sehingga dapat mengurangi kompleksitas komputasi dan jumlah parameter pada jaringan. Selain itu, *max pooling* juga membantu meningkatkan ketahanan model terhadap pergeseran kecil (*translation invariance*) pada citra input, sehingga fitur penting seperti pola tekstur atau bentuk penyakit pada daun kelapa sawit tetap dapat dikenali meskipun terjadi variasi posisi atau skala objek.

### 2.3.4 Flatten dan Fully Connected Layer

Matriks  $P_{i,j,k}$  ciri hasil ekstraksi dari lapisan konvolusi dan *max pooling* diratakan menjadi vektor satu dimensi (*flattened*). Untuk VGG19, *max pooling* terakhir memiliki ukuran  $7 \times 7 \times 512$  sehingga persamaan *flatten* menjadi vektor dengan panjang  $N = 25.088$  neuron dapat dilihat pada Persamaan 6.

$$x_{flat} = [P_{0,0,0}, P_{0,1,0}, \dots, P_{6,6,511}]^T \in \mathbb{R}^{25088 \times 1} \quad (6)$$

Vektor  $x_{flat}$  inilah yang kemudian menjadi input untuk 2 lapisan FC (*fully connected*) yang dikenakan fungsi aktivasi RELU Kembali seperti yang ditunjukkan pada Persamaan 7.

$$x_{FC} = f(W_2 f(W_1 x_{flat} + b_1) + b_2) \quad (7)$$

Keterangan:

- $x_{flat}$  : hasil *flatten* dari blok konvolusi terakhir
- $W_1, W_2$  : bobot lapisan FC1 dan FC2
- $b_1, b_2$  : bias masing-masing lapisan
- $x_{FC}$  : vektor fitur hasil 2 lapisan FC

Lapisan *fully connected* berfungsi untuk mempelajari hubungan non-linear antar fitur hasil ekstraksi sebelumnya serta melakukan pemetaan fitur menuju ruang klasifikasi. Penggunaan fungsi aktivasi ReLU pada kedua lapisan FC membantu meningkatkan kemampuan representasi model dan mempercepat proses konvergensi selama pelatihan. Bobot  $W_1$  dan  $W_2$  akan diperbarui melalui proses pembelajaran untuk memaksimalkan akurasi klasifikasi, sedangkan bias  $b_1$  dan  $b_2$  berfungsi untuk meningkatkan fleksibilitas model dalam menyesuaikan pola data.

### 2.3.5 Lapisan *Softmax* (Output Layer), Prediksi Kelas Akhir, dan Fungsi Loss

Lapisan terakhir dari VGG19 menggunakan fungsi aktivasi *softmax* untuk mengubah output linear dari lapisan *fully connected* terakhir menjadi probabilitas untuk setiap kelas (3 neuron/kelas pada penelitian ini: daun sehat, karat daun, *Curvularia*). Perhitungan lapisan *fully connected* terakhir, probabilitas *softmax*, dan prediksi kelas akhir dapat dilihat pada Persamaan 8, 9, dan 10.

$$Z = W_3 x_{FC} + b_3 \quad (8)$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^3 e^{z_j}} \quad (9)$$

$$\text{Prediksi} = \arg \max_i (\hat{y}_i) \quad (10)$$

Keterangan:

- $W_3 \in \mathbb{R}^{3 \times 4096}$  : bobot yang menghubungkan 4096 neuron FC2 ke 3 neuron output
- $b_3 \in \mathbb{R}^3$  : bias untuk tiap kelas
- $x_{FC} \in \mathbb{R}^{4096}$  : vektor hasil dari FC2
- $Z = [z_1, z_2, z_3]^T \in \mathbb{R}^3$  : nilai aktivasi linear (logit) sebelum fungsi softmax
- $\hat{y}_i$  : nilai probabilitas prediksi citra masuk kelas ke-i
- $\sum_{i=1}^3 \hat{y}_i$  : 1
- *Prediksi* : label kelas akhir yang terprediksi

Proses ini diawali dengan perhitungan nilai linear (*logit*) menggunakan bobot dan bias pada lapisan output sebagaimana ditunjukkan pada Persamaan 8, kemudian nilai tersebut dinormalisasi menggunakan fungsi *softmax* pada Persamaan 9 sehingga menghasilkan probabilitas untuk masing-masing kelas. Selanjutnya, kelas dengan nilai probabilitas tertinggi dipilih sebagai hasil prediksi akhir sesuai Persamaan 10. Untuk mengukur seberapa jauh hasil probabilitas prediksi ( $\hat{y}$ ) dari target sebenarnya ( $y$ ), digunakan fungsi *loss* yaitu dengan melihat selisih antara hasil prediksi dan label sebenarnya menggunakan *categorical cross-entropy loss* [18] tiap citra data latih seperti yang ditunjukkan pada Persamaan 11.

$$L_l = - \sum_{i=1}^3 y_i \log (\hat{y}_i) \quad (11)$$

Semakin kecil nilai *loss* yang dihasilkan, maka semakin baik kinerja model dalam melakukan klasifikasi. *Categorical cross-entropy* sangat sesuai digunakan pada kasus klasifikasi multikelas karena mampu memberikan penalti yang besar ketika model memberikan probabilitas rendah pada kelas yang seharusnya benar, sehingga mendorong proses pembelajaran menuju konvergensi yang optimal.

### 2.3.6 Backpropagation dan Update Bobot

Proses pelatihan dilakukan secara bertahap dengan ukuran *batch* (*batch size*) sebesar 32 yang artinya tahapan subbab 2.3.1 sampai dengan 2.3.5 dilakukan sekaligus terhadap  $\leq 32$  citra. Pada setiap *batch*, fungsi *loss* dihitung menggunakan Persamaan 12.

$$L_{\text{batch}} = \frac{1}{B} \sum_{l=1}^B L_l \quad (12)$$

Keterangan:

- $L_l$  = nilai *loss* citra ke-l
- $B$  = jumlah citra dalam satu *batch* (*batch size*)
- $L_{\text{batch}}$  = rata-rata *loss* dari seluruh citra dalam *batch*

Setelah diperoleh rata-rata loss per batch  $L_{batch}$  seperti pada Persamaan 12, proses pelatihan dilanjutkan dengan *backpropagation* untuk menyesuaikan bobot jaringan berdasarkan arah penurunan nilai *loss* yang dimulai dengan menghitung gradien pada lapisan output (*softmax*) berdasarkan Persamaan 13.

$$\frac{\partial L_{i,l}}{\partial z_{i,l}} = \hat{y}_{i,l} - y_{i,l} \quad (13)$$

Kemudian dilakukan propagasi mundur ke lapisan sebelumnya menggunakan aturan rantai (*chain rule*) untuk menghitung gradien *loss* terhadap bobot  $W$  dan bias  $b$  untuk citra ke- $l$  seperti pada Persamaan 14 dan 15.

$$\frac{\partial L_l}{\partial W} = (\hat{y}_l - y_l) x_l^T \quad (14)$$

$$\frac{\partial L_l}{\partial b} = \hat{y}_l - y_l \quad (15)$$

Perhitungan diatas dilakukan terhadap seluruh citra dalam setiap *batch* untuk kemudian diperoleh rata-rata gradien *loss*  $W$  dan  $b$  untuk setiap *batch* berdasarkan Persamaan 16 dan 17

$$\nabla W = \frac{1}{B} \sum_{l=1}^B \frac{\partial L_l}{\partial W} \quad (16)$$

$$\nabla b = \frac{1}{B} \sum_{l=1}^B \frac{\partial L_l}{\partial b} \quad (17)$$

Rata-rata gradien  $\nabla W$  dan  $\nabla b$  digunakan untuk memperbaharui bobot  $W$  dan  $b$  perhitungan *batch* selanjutnya menggunakan algoritma *gradient descent* atau *Adam optimizer* seperti pada Persamaan 18 dan 19.

$$W_{(t+1)} = W_{(t)} - \eta \nabla W \quad (18)$$

$$b_{(t+1)} = b_{(t)} - \eta \nabla b \quad (19)$$

dengan  $\eta$  sebagai *learning rate* dan  $t$  indeks iterasi (*batch* ke- $t$ ).

Proses pembaruan bobot dan bias pada setiap iterasi tersebut berlangsung secara berulang hingga seluruh *batch* dalam satu *epoch* selesai diproses. Setelah satu *epoch*, nilai *loss* rata-rata ( $L_{batch}$ ) dan akurasi model pada data latih serta data validasi dievaluasi untuk menilai sejauh mana model mengalami konvergensi. Jika belum konvergen, maka akan diproses ke *epoch* selanjutnya hingga nilai *loss* (saat perubahan nilai bobot  $W$  dan bias  $b$  sudah sangat kecil) terus menurun dan akurasi meningkat secara stabil. Model optimal kemudian disimpan dalam format .h5 untuk digunakan pada tahap pengujian dan evaluasi berikutnya [19].

## 2.4 Deteksi Penyakit Bercak Daun pada Daun Kelapa Sawit dan Evaluasi Hasil

Setelah proses pelatihan selesai, selanjutnya dilakukan proses implementasi deteksi seperti yang ditunjukkan pada Gambar 1 bagian deteksi terhadap data uji. Pada tahapan ini diawali dengan menginputkan data citra uji daun kelapa sawit. Setelah itu, citra uji diproses menggunakan model CNN yang telah dilatih sebelumnya untuk mengklasifikasikan kondisi daun berdasarkan kriteria visual [20][21] dengan 3 kelas, yaitu: 1) daun sehat dengan pola terstruktur tanpa bercak; 2) karat daun yang terdapat bercak coklat kehitaman; dan 3) *culvaria* yang terdapat bercak berwarna oranye kekuningan. Proses ini menunjukkan kemampuan sistem dalam mendeteksi dan mengklasifikasikan kondisi daun kelapa sawit secara otomatis berdasarkan ciri visual yang terdeteksi oleh model CNN. Namun, hasil deteksi dari pemodelan CNN tersebut perlu diuji lebih lanjut apakah sudah sesuai dengan data sebenarnya melalui evaluasi performa menggunakan *accuracy* dan *confusion matrix* (*precision*, *recall*, dan *f1-score*) seperti yang ditunjukkan pada Persamaan 20 sampai dengan 23.

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Sampel}} = \frac{TP}{TP+TN+FP+FN} \quad (20)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (21)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (22)$$

$$\text{F1\_Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (23)$$

Dimana: 1) TP (*True Positive*): Citra daun dari suatu kelas yang diklasifikasikan benar sebagai kelas tersebut; 2) TN (*True Negative*): Citra daun dari kelas lain yang tidak diklasifikasikan sebagai kelas tersebut; 3) FP (*False*

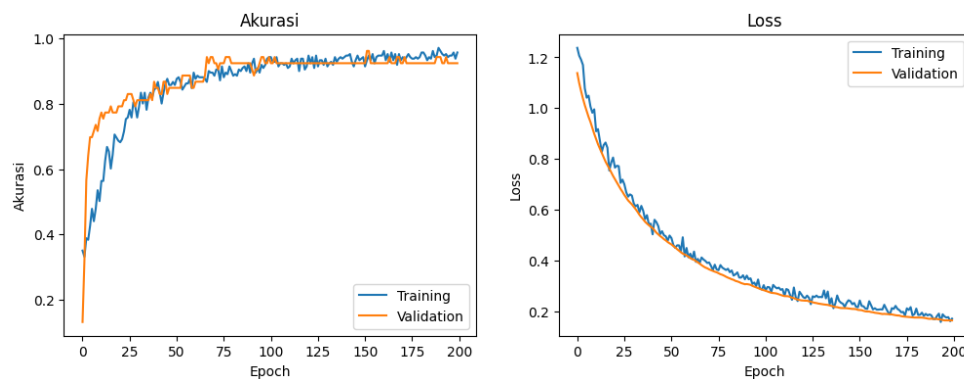
*Positive*): Citra daun dari kelas lain yang salah diklasifikasikan sebagai kelas tersebut; dan 4) FN (*False Negative*): Citra daun dari kelas tersebut yang salah diklasifikasikan ke kelas lain.

### 3. HASIL DAN ANALISIS

Penelitian ini telah berhasil melakukan pelatihan model CNN dengan arsitektur VGG19 terhadap 64% data latih (221 citra) menggunakan GPU [22], serta parameter utama yaitu *optimizer* Adam, *learning rate* = 0.001, *epoch* = 200, *batch size* = 32, dan fungsi *loss* menggunakan *categorical cross-entropy* [18].

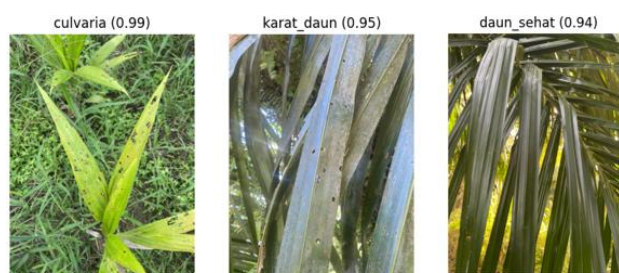
```
Epoch 197/200
7/7 — 0s 349ms/step - accuracy: 0.9424 - loss: 0.1989
Epoch 197: val_loss improved from 0.15456 to 0.15369, saving model to /content/drive/MyDrive/best_model_daun_sawit.h5
WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save_model(model)'. This file
7/7 — 3s 469ms/step - accuracy: 0.9437 - loss: 0.1897 - val_accuracy: 0.9811 - val_loss: 0.1537
Epoch 198/200
7/7 — 0s 343ms/step - accuracy: 0.9769 - loss: 0.1441
Epoch 198: val_loss improved from 0.15369 to 0.15340, saving model to /content/drive/MyDrive/best_model_daun_sawit.h5
WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save_model(model)'. This file
7/7 — 3s 472ms/step - accuracy: 0.9745 - loss: 0.1487 - val_accuracy: 0.9811 - val_loss: 0.1534
Epoch 199/200
7/7 — 0s 484ms/step - accuracy: 0.9511 - loss: 0.1751
Epoch 199: val_loss did not improve from 0.15340
7/7 — 4s 551ms/step - accuracy: 0.9501 - loss: 0.1767 - val_accuracy: 0.9811 - val_loss: 0.1538
Epoch 200/200
7/7 — 0s 343ms/step - accuracy: 0.9301 - loss: 0.2133
Epoch 200: val_loss improved from 0.15340 to 0.15291, saving model to /content/drive/MyDrive/best_model_daun_sawit.h5
WARNING:absl:You are saving your model as an HDF5 file via 'model.save()' or 'keras.saving.save_model(model)'. This file
7/7 — 4s 505ms/step - accuracy: 0.9330 - loss: 0.2097 - val_accuracy: 0.9811 - val_loss: 0.1529
```

**Gambar 3. Hasil Pelatihan dalam 4 Epoch Terakhir**



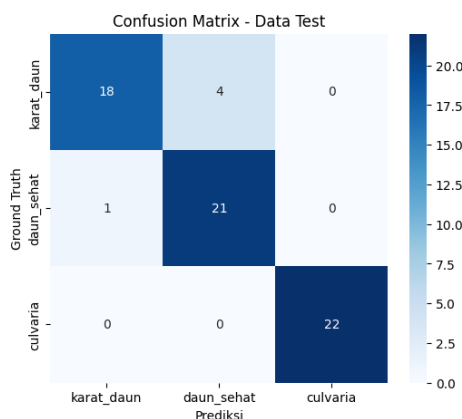
**Gambar 4. Hasil Akurasi dan Loss Data Latih dan Validasi Tiap Epoch**

Berdasarkan Gambar 3, saat *epoch* ke-200, nilai akurasi dan nilai *loss* terhadap data latih masing-masing sebesar 0,93 dan 0,21, sedangkan nilai akurasi dan nilai *loss* model pelatihan yang dievaluasi terhadap data validasi masing-masing sebesar 0,98 dan 0,15. Selain itu, pada Gambar 4 menunjukkan progres kenaikan akurasi, yang mana pada tahap awal pelatihan nilai akurasi model masih rendah, yakni berkisar antara 20–30%. Namun, seiring bertambahnya *epoch*, akurasi meningkat secara signifikan dan melampaui 80% setelah sekitar 40 *epoch*. Proses peningkatan terus berlanjut hingga mencapai akurasi sebesar pada *epoch* ke-200. Sementara itu, grafik *loss* memperlihatkan bahwa pada awal pelatihan nilai *loss* relatif tinggi (lebih dari 1,0), namun mengalami penurunan signifikan hingga mendekati 0,1 pada akhir pelatihan. Kedekatan antara kurva *loss* data latih dan validasi menunjukkan bahwa proses pelatihan berlangsung stabil tanpa adanya perbedaan yang berarti, sehingga dapat disimpulkan bahwa model tidak mengalami *overfitting*. Lebih lanjut, pola akurasi yang serupa antara data latih dan data validasi menunjukkan bahwa model tidak hanya mampu mempelajari karakteristik pada data pelatihan, tetapi juga berhasil mempertahankan performanya pada data validasi, yang mengindikasikan bahwa model yang dibangun memiliki kemampuan klasifikasi citra daun yang konsisten, akurat, dan andal.



**Gambar 5. Model CNN berbasis Arsitektur VGG19 Berhasil Diterapkan pada Data Uji**

Selanjutnya, model yang telah dibangun coba diterapkan dan dievaluasi kepada data uji. Gambar 5 menunjukkan bagaimana model usulan berhasil diterapkan pada salah satu sampel data uji. Tampak dalam sampel data uji gambar tersebut, model mendeteksi dengan benar citra daun kelapa sawit sesuai dengan kelas penyakit daun masing-masing dengan akurasi di atas 90%. Namun hasil ini belum mampu menggeneralisir kemampuan model dalam mengklasifikasikan penyakit daun kelapa sawit. Untuk itu, model perlu dievaluasi terhadap seluruh data uji menggunakan *confusion matrix* seperti yang ditunjukkan pada Gambar 6.



**Gambar 6. Confusion Matrix Data Uji**

Berdasarkan data dari *confusion matrix*, sebagian besar dari total 66 citra data uji berhasil diklasifikasikan dengan benar pada masing-masing kelas. Pada kelas karat daun, terdapat 18 sampel yang terprediksi dengan benar, sementara 4 sampel lainnya keliru terdeteksi sebagai daun sehat. Untuk kelas daun sehat, sebanyak 21 sampel berhasil dikenali dengan benar, dan hanya 1 sampel yang salah diklasifikasikan sebagai karat daun. Adapun pada kelas *culvaria*, seluruh 22 sampel berhasil diklasifikasikan dengan benar tanpa kesalahan prediksi. Jika data ini diolah menggunakan perhitungan *precision*, *recall*, *F1-score*, maka diperoleh hasil seperti yang ditunjukkan pada Tabel 1.

**Tabel 1. Hasil Perhitungan Matriks Data Testing/uji**

| Kategori     | TP | TN | FP | FN | Precision | Recall | F1-Score |
|--------------|----|----|----|----|-----------|--------|----------|
| Karat Daun   | 18 | 43 | 1  | 4  | 0,94      | 0,81   | 0,87     |
| Daun Sehat   | 21 | 40 | 4  | 1  | 0,84      | 0,95   | 0,89     |
| Culvaria     | 22 | 44 | 0  | 0  | 1,00      | 1,00   | 1,00     |
| Macro Avg    | -  | -  | -  | -  | 0,93      | 0,92   | 0,92     |
| Weighted Avg | -  | -  | -  | -  | 0,93      | 0,92   | 0,92     |

Tabel 1 menunjukkan bahwa nilai rata-rata (*macro = weighted*, karena data tiap kelas seimbang) *precision*, *recall*, dan *F1-score* seluruh kelas masing-masing sebesar 0,93, 0,92, dan 0,92. Terlihat dari data bahwa klasifikasi tertinggi terdapat pada kelas "Culvaria" dengan rincian *precision*, *recall*, dan *F1-score* seluruhnya sebesar 1. Hal ini menandakan bahwa pola citra penyakit tersebut relatif sangat mudah dikenali oleh model. Sedangkan dua kelas lainnya seperti kelas "Daun Sehat" masih lebih unggul dalam hal keseimbangan klasifikasi bila dibandingkan dengan kelas "Culvaria" dengan nilai *F1-score* sebesar 0,89. Akan tetapi, dalam konteks deteksi penyakit tanaman, nilai *recall* tentu memiliki peran yang sangat penting karena kesalahan dalam mendeteksi daun yang terinfeksi (*false negative*) dapat menyebabkan keterlambatan penanganan dan berpotensi mempercepat penyebaran penyakit. Jika melihat dari nilai *recall*, maka sistem masih dianggap andal sebagai sistem pendukung keputusan lapangan, karena *recall* semua jenis kelas mencapai diatas 0,80. Meskipun masih terdapat nilai *recall* terendah yaitu saat mendeteksi "Karat Daun" dengan nilai *recall* sebesar 0,81. Hal ini terjadi karena adanya kemiripan visual dan tekstur antara kelas karat daun dengan daun sehat. Fenomena ini sejalan dengan temuan penelitian sebelumnya yang menyebutkan bahwa kesamaan fitur visual antar jenis daun dapat memengaruhi tingkat akurasi klasifikasi [23]. Namun secara umum, hasil penelitian ini menegaskan bahwa penerapan CNN berbasis arsitektur VGG19 mampu menghasilkan sistem pendeteksian

penyakit bercak daun kelapa sawit yang akurat dan konsisten. Temuan ini mendukung hasil studi sebelumnya oleh [18] dan [19], yang juga melaporkan efektivitas penggunaan CNN dalam pengenalan citra tanaman.

#### 4. KESIMPULAN

Model CNN berbasis arsitektur VGG19 berhasil dibuat untuk mengklasifikasikan penyakit daun kelapa sawit ke dalam 3 kelas, yaitu "Daun Sehat", "Karat Daun", "Culvaria". Dengan menggunakan total 221 citra data latih (64% dari 330 total dataset), model mampu mencapai akurasi data latih dan data validasi (16% dari 330 total dataset) masing-masing sebesar 0,93 dan 0,98 pada *epoch* ke-200. Kemudian evaluasi model terhadap 66 citra data uji (20% dari 330 total dataset) menghasilkan nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 0,94, 0,81, dan 0,87 untuk kelas "Karat Daun", 0,84, 0,95, dan 0,89 untuk kelas "Daun Sehat", serta 1,00 untuk kelas "Curvularia". Hasil pengujian menunjukkan performa yang konsisten, sehingga sistem usulan dinilai efektif dalam melakukan klasifikasi penyakit bercak daun kelapa sawit.

Namun demikian, penelitian ini masih memiliki beberapa keterbatasan, diantaranya jumlah dataset yang relatif terbatas serta variasi kondisi citra yang belum sepenuhnya merepresentasikan kondisi lapangan secara nyata. Selain itu, model masih mengalami kesulitan dalam membedakan gejala awal penyakit karat daun yang memiliki kemiripan visual dengan daun sehat. Keterbatasan lain terletak pada ketergantungan terhadap kualitas citra input, seperti pencahayaan dan sudut pengambilan gambar, yang dapat memengaruhi akurasi prediksi. Oleh karena itu, diperlukan pengembangan lebih lanjut melalui penambahan jumlah dan variasi data, penerapan teknik augmentasi yang lebih beragam, serta eksplorasi arsitektur CNN yang lebih adaptif agar sistem ke depan berpotensi digunakan sebagai alat bantu deteksi dini penyakit bercak daun guna mendukung pemantauan kesehatan tanaman secara lebih cepat, akurat, dan efisien.

#### UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan kontribusi dalam pelaksanaan penelitian ini, khususnya Program Studi Teknik Komputer Universitas Borneo Tarakan.

#### REFERENSI

- [1] M. H. As-Siddiqi, K. Auliasari, and R. Primaswara Prasetya, "PEMBANGUNAN SISTEM REKOMENDASI JENIS PUPUK PADA TANAMAN SAWIT MENGGUNAKAN METODE AHP (Analytical Hierarchy Process)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 957–964, 2023, doi: 10.36040/jati.v6i2.5407.
- [2] R. Utami, E. I. K. Putri, and M. Ekayani, "Dampak Ekonomi dan Lingkungan Ekspansi Perkebunan Kelapa Sawit (Studi Kasus: Desa Penyabungan, Kecamatan Merlung, Kabupaten Tanjung Jabung Barat, Jambi)," *J. Ilmu Pertan. Indones.*, vol. 22, no. 2 SE-Articles, pp. 115–126, doi: 10.18343/jipi.22.2.115.
- [3] Muhammad Riyan Hidayah, "Dampak Perkebunan Kelapa Sawit terhadap Lingkungan:Menyeimbangkan Risiko Ekologis dengan Keuntungan Ekonomi," *Globe Publ. Ilmu Tek. Teknol. Kebumihan, Ilmu Perkapalan*, vol. 3, no. 1 SE-Articles, pp. 90–94, Feb. 2025, doi: 10.61132/globe.v3i1.763.
- [4] N. T. M. A. Mardia, H. Wirianata, and B. Yuniasih, "Pengaruh Curah Hujan terhadap Produktivitas Kelapa Sawit," *AGROFORETECH*, vol. 3, no. 3 SE-Agroteknologi, pp. 1711–1724, Sep. 2025, [Online]. Available: <https://jurnal.instiperjogja.ac.id/index.php/JOM/article/view/2261>
- [5] D. Saputra, B. Yuniasih, and N. M. Titiaryanti, "Pengaruh Kerapatan Nephrolepis biserrata terhadap Kondisi Iklim Mikro di Kebun Kelapa Sawit," *AGROFORETECH*, vol. 1, no. 2 SE-Agroteknologi, pp. 940–945, Jun. 2023, [Online]. Available: <https://jurnal.instiperjogja.ac.id/index.php/JOM/article/view/642>
- [6] I. I. Sihombing and M. M. SP, *SUMBER DAYA LAHAN DI PERKEBUNAN KELAPA SAWIT*. PENERBIT KBM INDONESIA, 2025.
- [7] F. B. Laksono, "Deteksi penyakit tanaman dengan convolution neural network: Kombinasi arsitektur VGG16 dan ResNet34 untuk klasifikasi daun," *J. Komput. Dan Teknol. Inf.*, vol. 2, no. 2, pp. 72–79, 2024, doi: 10.26714/jkti.v2i2.13932.
- [8] E. Rasywir, R. Sinaga, and Y. Pratama, "Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)," *Paradig. - J. Komput. dan Inform.*, vol. 22, no. 2, pp. 117–123, 2020, doi: 10.31294/p.v22i2.8907.

- [9] A. Susanto, A. E. Prasetyo, H. Priwiratama, and M. Syarovi, "Laju Fotosintesis pada Tanaman Kelapa Sawit Terinfeksi Karat daun *Cephaleuros virescens*," *J. Fitopatol. Indones.*, vol. 16, no. 1 SE-Articles, pp. 21–29, doi: 10.14692/jfi.16.1.21-29.
- [10] A. V. Febriani and R. S. Kasiamdari, "Identification of *Curvularia eragrostidis* (Henn.) J.A.Mey. the Leaf Spot Pathogen of Oil Palm (*Elaeis guineensis* Jacq.) and It's Control by False Elder (*Peronema canescens* Jack) Leaf Extract," *Int. J. Oil Palm*, vol. 6, no. 2 SE-Articles, pp. 25–34, May 2024, doi: 10.35876/ijop.v6i2.111.
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014, doi: 10.48550/arXiv.1409.1556.
- [12] W. Styorini, W. E. Putra, W. Khabzli, and Y. Triyani, "Penerapan Deep Learning Pada Jenis Penyakit Tanaman Kelapa Sawit Menggunakan Algoritma Convolutional Neural Network," *J. Komput. Terap.*, vol. 8, no. 2, pp. 359–367, 2022, doi: 10.35143/jkt.v8i2.5522.
- [13] M. U. Hossain, M. A. Rahman, M. M. Islam, A. Akhter, M. A. Uddin, and B. K. Paul, "Automatic driver distraction detection using deep convolutional neural networks," *Intell. Syst. with Appl.*, vol. 14, p. 200075, 2022, doi: 10.1016/j.iswa.2022.200075.
- [14] Y. A. Nizamli and A. Y. Filatov, "Improving transfer learning performance for abnormality detection in brain MRI images using feature optimization techniques," in *2024 XXVII International Conference on Soft Computing and Measurements (SCM)*, IEEE, 2024, pp. 432–435. doi: 10.1109/SCM62608.2024.10554161.
- [15] Y. Ren, C. Zhu, and S. Xiao, "Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN," 2018. doi: 10.3390/app8050813.
- [16] I. Ahmad, J. Rashid, M. Faheem, A. Akram, N. A. Khan, and R. ul Amin, "Autism spectrum disorder detection using facial images: A performance comparison of pretrained convolutional neural networks," *Healthc. Technol. Lett.*, vol. 11, no. 4, pp. 227–239, 2024, doi: 10.1049/htl2.12073.
- [17] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022, doi: 10.1016/j.neucom.2022.06.111.
- [18] M. F. Gunardi, "Implementasi Augmentasi Citra pada Suatu Dataset," *J. Inform.*, vol. 9, no. 1, pp. 1–5, 2023.
- [19] J. R. A. Purba, I. Subandar, and U. AR, "Analisis Infeksi Karat Daun (*Cephaleuros vircens*) pada Tanaman Kelapa Sawit (*Elaeis guineensis* jacq) di Perkebunan Jaya Seujahtera Afdeling III PT. ASN," *Biofarm J. Ilm. Pertan.*, vol. 18, no. 2, p. 96, 2022, doi: 10.31941/biofarm.v18i2.2299.
- [20] F. N. Cahya, "Perbandingan Identifikasi Wajah Dengan Ekstraksi Fitur Haralick Dan CNN," *Informatics Digit. Expert*, vol. 2, no. 2, pp. 60–66, 2021, doi: 10.36423/index.v2i02.594.
- [21] F. Felix, S. Faisal, T. F. M. Butarbutar, and P. Sirait, "Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun," *J. SIFO Mikroskil*, vol. 20, no. 2, pp. 117–134, 2019, doi: 10.55601/jsm.v20i2.670.
- [22] M. I. Syahputra and A. T. Wibowo, "Klasifikasi Genus Tanaman Anggrek berdasarkan Citra Kuntum Bunga Menggunakan Metode Convolutional Neural Network (CNN)," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8015–8023, 2020.
- [23] N. T. Yildiz, H. Kocaman, H. Yildirim, and M. Canli, "An investigation of machine learning algorithms for prediction of temporomandibular disorders by using clinical parameters," *Med. (United States)*, vol. 103, no. 41, p. e39912, 2024, doi: 10.1097/MD.00000000000039912.