

KLASIFIKASI PENYAKIT DAUN PADI MENGGUNAKAN SUPPORT VECTOR MACHINE BERDASARKAN FITUR MENDALAM (*DEEP FEATURE*)

Devina Margarita ^{*1)}, Hendra Maulana²⁾, Eka Prakarsa Mandyartha³⁾

1. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia
2. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia
3. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia

Article Info

Kata Kunci: Penyakit daun padi, CNN, SVM, SMO, arsitektur VGG-19.

Keywords: Rice leaf disease, CNN, SVM, SMO, VGG-19 architecture.

Article history:

Received 18 September 2024

Revised 17 Oktober 2024

Accepted 11 November 2024

Available online 4 December 2024

DOI :

<https://doi.org/10.29100/jupi.v9i4.5634>

* Corresponding author.

Hendra Maulana

E-mail address:

hendra.maulana.if@upnjatim.ac.id

ABSTRAK

Tanaman padi memiliki peran yang sangat penting dalam menyediakan pangan bagi populasi global. Namun, serangan hama dan bakteri dapat menghambat produksi padi dengan mengganggu proses fotosintesis dan fase generatifnya, yang berujung pada penurunan kualitas dan kuantitas panen. Untuk mengatasi tantangan ini, penelitian ini mengusulkan pemanfaatan teknologi pemrosesan citra dan *machine learning*. Metode yang digunakan mencakup *Convolutional Neural Network* (CNN) dengan arsitektur VGG-19 untuk mengekstraksi fitur citra, serta *Support Vector Machine* (SVM) dengan pendekatan *training* menggunakan *Sequential Minimal Optimization* (SMO) untuk klasifikasi. Penelitian ini terdiri dari lima tahap utama: *preprocessing*, pembagian data, ekstraksi fitur CNN, pelatihan SVM, dan evaluasi hasil. Berbagai skenario dengan kernel SVM yang berbeda dievaluasi, di mana hasilnya menunjukkan bahwa kernel RBF dan linear mampu mencapai akurasi tertinggi, yaitu 93,94%. Penelitian ini menunjukkan bahwa penggunaan CNN dan SVM dalam mengatasi tantangan klasifikasi citra penyakit daun pada tanaman padi, dapat memberikan hasil yang signifikan.

ABSTRACT

Rice gardens have a very important role in providing food for the global population. However, pest and bacterial attacks can hamper rice production by disrupting the photosynthesis process and the generative phase, which ends in a decrease in the quality and quantity of the harvest. utilization of image processing and machine learning technology. The methods used include Convolutional Neural Network (CNN) with VGG-19 architecture to extract image features, as well as Support Vector Machine (SVM) with a training approach using Sequential Minimal Optimization (SMO) for classification consisting of five main stages: preprocessing, data division, CNN feature extraction, SVM training, and results evaluation. Various scenarios with different SVM kernels were evaluated, where the results showed that the RBF and linear kernels were able to achieve the highest accuracy, namely 93.94%. This research shows that the use of CNN and SVM in overcoming barriers to image classification of leaf diseases in rice plants, can provide significant results.

I. PENDAHULUAN

TANAMAN padi memiliki peranan penting dalam pangan Indonesia, menjadi sumber utama bahan pangan bagi lebih dari setengah populasi global [1]. Menurut Badan Pusat Statistik, pada tahun 2023, produksi padi mencapai 53,63 juta ton GKG, mengalami penurunan sekitar 1,12 juta ton GKG atau sekitar 2,05% dibandingkan dengan tahun sebelumnya. Sementara itu, produksi beras pada tahun 2023 diperkirakan turun sebesar 645,09 ribu ton atau sekitar 2,05% dari produksi tahun 2022 [2]. Menurut data dari *Food and Agriculture Organization* (FAO), sekitar 20-40% dari kegagalan produksi pangan global disebabkan oleh serangan hama dan penyakit. Sementara itu, *International Rice Research Institute* (IRRI) mencatat bahwa petani mengalami kegagalan panen sekitar 37% dari tanaman padi setiap tahunnya karena serangan tersebut [3].

Dilakukan penelusuran langsung terhadap kondisi tanaman padi yang terinfeksi hama, yaitu di Kota Surabaya, tepatnya di Jalan Kuwukan, Kecamatan Sambikerep. Hasil penelusuran dan wawancara dengan petani setempat menunjukkan bahwa penyakit pada tanaman padi dapat terdeteksi awalnya dari kondisi daunnya. Penyakit yang

sering ditemui di sana adalah hama ulat dan bacterial leaf blight. *Bacterial leaf blight* ditandai dengan daun yang menguning di bagian tepiannya, kemudian mengering hingga tulang daun dan keseluruhan daun. Sementara, hama ulat ditandai dengan bercak putih pada daun karena ulat memakan zat hijau daun.

Penyakit hama dan bakteri pada padi mengganggu fotosintesis dan fase generatif, yang menyebabkan pengurangan berat dan kualitas panen [4]. Petani di sana kesulitan dengan pendeteksian penyakit daun padi yang hanya mengandalkan pengamatan mata manusia. Masalahnya, mata manusia rentan kesalahan terutama pada petani tua yang penglihatannya menurun. Untuk mengatasi masalah ini, teknologi pemrosesan citra dan pembelajaran mesin dapat digunakan untuk klasifikasi penyakit pada daun padi secara akurat. Sehingga dapat membantu dalam mengidentifikasi jenis penyakit pada daun padi, dan dapat menanggulangi penyakit tanaman padi secara dini.

Salah satu pendekatan yang efektif dan umum digunakan dalam klasifikasi jenis penyakit daun pada tanaman padi, yaitu menggunakan *deep learning* dan *machine learning*. *Deep Learning*, yaitu subbidang *machine Learning* yang menggunakan algoritma matematika dan beroperasi seperti pemikiran manusia [5]. Salah satu teknik yang sering digunakan dalam Deep Learning untuk mengolah citra adalah *Convolutional Neural Network* (CNN). CNN dikembangkan dari metode *MultiLayer Perceptron* (MLP) dalam jaringan saraf tiruan [6]. CNN memiliki kemampuan untuk mengekstraksi fitur-fitur mendalam (*deep features*) dari data citra melalui lapisan-lapisan dalam arsitektur *neural network*, seperti konvolusi, pooling, dan aktivasi [7]. Dengan menggunakan fitur-fitur ini, CNN dapat mempelajari pola-pola yang kompleks dari gambar.

Machine learning adalah cabang ilmu komputer yang memberikan kemampuan belajar pada komputer tanpa diprogram secara eksplisit [8]. Salah satu tugas dalam machine learning adalah klasifikasi, yakni pengelompokan data yang memiliki label [8]. Algoritma *machine learning* dapat dikategorikan menjadi empat jenis: *supervised learning*, *semi-supervised learning*, *unsupervised learning*, dan *reinforcement learning*. *Supervised learning* merupakan pembelajaran terawasi, dimana algoritma belajar dari data berlabel [9]. Tujuannya untuk membuat prediksi atau melakukan klasifikasi. Salah satu contoh dari supervised learning adalah algoritma dari metode *Support Vector Machine* (SVM). SVM adalah klasifikasi linear dengan fungsi kernel dan teori optimisasi. SVM mencari hyperplane terbaik untuk memisahkan data dari kelas yang berbeda. SVM memaksimalkan margin antara hyperplane dan support vector [10].

Beberapa penelitian terdahulu telah menerapkan metode SVM sebagai alat klasifikasi dan CNN sebagai alat ekstraksi fitur. Contohnya, dalam penelitian "Plant Disease Identification and Classification Using Convolutional Neural Network and SVM" [11] dan "Identification of Potato Leaf Diseases Using Hybrid Convolution Neural Network with Support Vector Machine" [12]. Dalam penelitian pertama, CNN digunakan untuk ekstraksi fitur dan SVM untuk klasifikasi, mencapai akurasi tinggi 98,9% untuk daun anggur dengan arsitektur CNN VGG19. Penelitian kedua juga mengadopsi pendekatan yang serupa, dengan Hybrid CNN (ResNet50) dan SVM memberikan akurasi tinggi sebesar 97,3%.

Dalam konteks penelitian ini, CNN dan SVM bekerja bersama untuk mengklasifikasikan penyakit daun padi secara efektif. CNN bertugas untuk mengekstraksi fitur-fitur mendalam dari citra daun padi melalui proses konvolusi dan pooling. Fitur-fitur yang diekstraksi dari lapisan-lapisan ini kemudian dilewatkan ke lapisan *flatten*. Lapisan *flatten* berfungsi untuk mengubah matriks fitur menjadi vektor satu dimensi yang dapat diinput ke dalam SVM. Penggunaan lapisan *flatten* penting karena SVM hanya bisa menerima input berupa vektor satu dimensi, bukan matriks. Kombinasi ini memungkinkan sistem untuk melakukan klasifikasi dengan akurasi tinggi karena memanfaatkan CNN dalam ekstraksi fitur dan SVM, dengan menggunakan metode *Sequential Minimal Optimization* (SMO) dalam klasifikasi. CNN mampu menangkap pola-pola kompleks dari citra daun padi, sedangkan SVM efisien dalam mencari hyperplane terbaik yang memisahkan data dari kelas yang berbeda berdasarkan fitur yang telah diekstraksi oleh CNN. Oleh karena itu, penggunaan kedua metode ini memungkinkan dapat meningkatkan keakuratan dan efisiensi dalam klasifikasi penyakit daun padi.

Penelitian ini memberikan kontribusi signifikan dengan memanfaatkan arsitektur CNN VGG19 untuk ekstraksi fitur dan algoritma SVM dengan SMO untuk klasifikasi penyakit daun padi. Penelitian ini mengumpulkan dataset secara langsung dari lapangan, memastikan bahwa data yang digunakan representatif dan relevan dengan kondisi nyata di lapangan. Meskipun kombinasi VGG19 dan SVM telah banyak diterapkan dalam berbagai penelitian, penelitian ini fokus pada pengujian parameter kernel yang berbeda untuk menemukan konfigurasi optimal dalam klasifikasi penyakit daun padi. Parameter-parameter yang diuji meliputi, *C* (*Cost*), *gamma* (γ), *degree*, dan *coef* (koefisien).

Hasil penelitian ini diimplementasikan dalam bentuk klasifikasi penyakit daun padi menggunakan teknik *deep learning* dan *machine learning*. Hasil klasifikasi ini diharapkan dapat digunakan oleh peneliti selanjutnya atau pengembang teknologi pertanian untuk membantu dalam pengembangan alat diagnostik yang lebih akurat dan

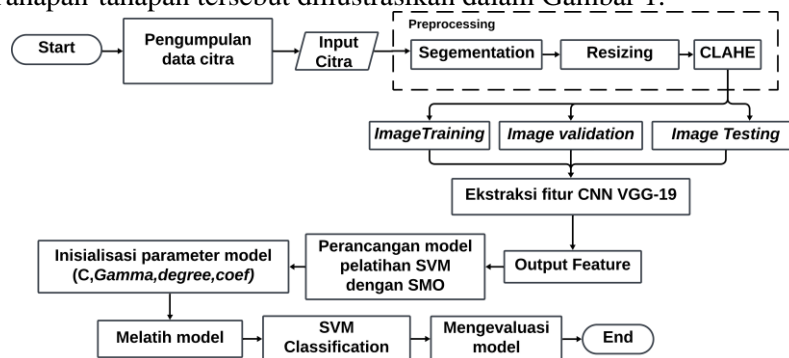
efisien. Dengan demikian, meskipun penelitian ini tidak sampai pada tahap pembuatan aplikasi praktis, hasilnya dapat memberikan dasar untuk pengembangan sistem deteksi penyakit tanaman yang lebih lanjut.

Berdasarkan uraian di atas, penggunaan metode SVM dengan fitur mendalam dari CNN telah terbukti efektif dalam meningkatkan akurasi klasifikasi citra. Oleh karena itu, penelitian ini bertujuan untuk mendalami penerapan metode tersebut dalam klasifikasi citra penyakit daun pada tanaman padi. Pada penelitian ini metode CNN, akan menggunakan arsitektur VGG19 untuk ekstraksi fitur yang akan dijadikan input data dalam metode SVM untuk proses klasifikasinya. Selanjutnya, pelatihan SVM akan menggunakan algoritma *Sequential Minimal Optimization* (SMO). Selain itu, akan dilakukan pengujian terhadap parameter kernel, nilai *cost*, nilai *gamma*, nilai *degree*, dan nilai *coef* untuk memperoleh hasil akurasi yang optimal.

Dilakukan klasifikasi untuk tiga jenis kelas, yaitu daun yang terinfeksi oleh hama ulat, daun yang terkena *Bacterial leaf blight*, dan daun yang dalam kondisi normal atau tidak terinfeksi. Data yang digunakan dalam penelitian ini dikumpulkan langsung dari lapangan, di mana citra-citra daun padi diambil menggunakan kamera *handphone* Poco F4. Penelitian ini terdiri dari lima tahapan utama: preprocessing, pembagian data (pengujian, pelatihan, dan validasi), ekstraksi fitur CNN, pelatihan SVM, dan hasil akurasi terhadap data uji. Tahap preprocessing mencakup segmentasi, *resizing*, dan penerapan metode CLAHE (*Contrast Limited Adaptive Histogram Equalization*).

II. METODE PENELITIAN

Dalam penelitian ini, terdapat beberapa tahapan yang dilakukan dalam proses klasifikasi untuk memastikan kelancaran penelitian. Tahapan-tahapan tersebut diilustrasikan dalam Gambar 1.

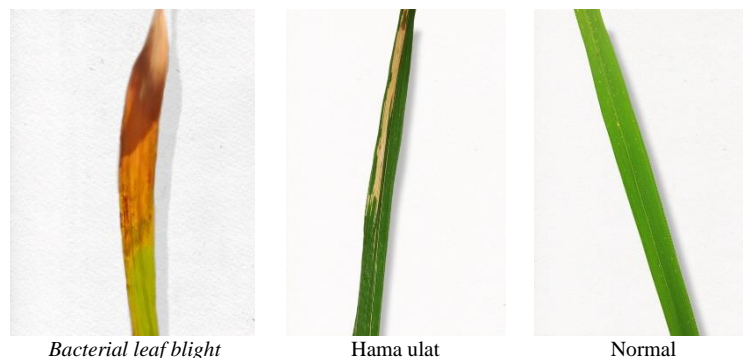


Gambar. 1. Alur Tahapan Penelitian

A. Pengumpulan data

Pada tahap pengumpulan data dalam penelitian ini, data dikumpulkan langsung dari lapangan dalam kondisi nyata. Citra penyakit pada daun padi diambil menggunakan kamera *handphone* Poco F4, dan hasil foto tersebut digunakan untuk dataset dalam penelitian ini. Terdapat dua jenis penyakit utama yang diamati dan didokumentasikan, yaitu *bacterial leaf blight* dan serangan hama ulat.

Penyakit *bacterial leaf blight* ditandai dengan daun yang menguning di bagian tepinya dan kemudian mengering hingga tulang daun dan keseluruhan daun. Sementara itu, serangan hama ulat ditandai dengan adanya bercak putih pada daun akibat ulat yang memakan zat hijau daun. Identifikasi penyakit dilakukan berdasarkan observasi langsung di lapangan serta wawancara dengan petani. Selain itu, terdapat juga jenis citra daun padi yang sehat atau normal, tidak terkena penyakit. Bentuk hasil citra dari pengumpulan data dapat dilihat pada Gambar 2.



Gambar. 2. Sampel Citra Dataset

Total dataset dalam tiap kelas bertotalkan 330 sampel data. Dimana total dari kelas Bacterial leaf blight adalah 110 data, total dari kelas hama ulat adalah 110 data, dan kelas untuk normal atau tidak berpenyakit adalah 110 data. Setelah dataset dikumpulkan, langkah selanjutnya adalah menyimpannya dalam sistem Windows yang telah ditentukan. Kemudian, dataset perlu dihubungkan dengan kode program yang akan digunakan. Tahapan ini bertujuan untuk mengorganisir dataset ke dalam struktur data yang sesuai agar dapat diproses oleh kode program. Berikut adalah rancangan model algoritma untuk menghubungkan dataset dengan kode program.

```

sdir = r'C:\Users\asus\Downloads\Dataset skripsi 3'
# Jenis penyakit
jenis_penyakit = ['Bacterial leaf blight', 'Hama ulat', 'Normal']

file_data = []

for root, dirs, files in os.walk(sdir):
    for file in files:
        filepath = os.path.join(root, file)
        label = os.path.basename(root)
        file_data.append((filepath, label))

df = pd.DataFrame(file_data, columns=['file_path', 'label'])
    
```

Gambar. 3. Pengambilan dan pengolahan data

B. Segmentasi citra

Segmentasi citra adalah proses untuk membagi atau memisahkan citra digital menjadi beberapa bagian yang memiliki sifat atau ciri yang sama [13]. Proses segmentasi ini bertujuan untuk memisahkan bagian daun padi yang terinfeksi penyakit dari latar belakangnya, yang memiliki warna hijau untuk daun sehat dan putih untuk latar belakang. Sehingga menyisakan objek utamanya, yaitu bagian daun yang terkena penyakit, sementara latar belakangnya terdiri dari warna hijau (daun sehat) dan putih (latar belakang).

Dalam penelitian ini proses segmentasi citra dilakukan berdasarkan warna, menggunakan ruang warna HSV. Berikut ini adalah tahapan untuk proses segmentasi yang dilakukan.

1. Konversi ke ruang warna HSV

Citra awal dikonversi dari ruang warna RGB ke HSV untuk memisahkan informasi warna dalam komponen *Hue* (H), *Saturation* (S), dan *Value* (V). Proses konversi ini dilakukan dengan menggunakan persamaan rumus berikut [14]:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)} \quad (1)$$

$$V = \max(r, g, b) \quad (2)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & V > 0 \end{cases} \quad (3)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 \times (g-b)}{S \times V}, & \text{jika } V = r \\ 60 \times \left[2 + \frac{b-r}{S \times V} \right], & \text{jika } V = g \\ 60 \times \left[4 + \frac{r-g}{S \times V} \right], & \text{jika } V = b \end{cases} \quad (4)$$

$$H = H + 360, \text{ jika } H < 0$$

2. Definisi nilai H,S,V

Setelah mengkonversi ke HSV, selanjutnya menentukan batas nilai Hue, Saturation, dan Value untuk menentukan warna piksel yang termasuk latar belakang (hijau dan putih) dan objek penyakit daunnya. Batas ini digunakan untuk memisahkan objek dan latar belakang dalam citra.

3. Pembuatan mask untuk latar belakang dan objek

Masking citra adalah proses untuk menghilangkan atau menyorot bagian tertentu dari citra. Mask dibuat dengan mengidentifikasi piksel-piksel pada citra yang sesuai dengan batas warna latar belakang hijau dan putih. Setiap piksel dievaluasi, yang sesuai dengan batas diberi nilai 1 (putih) dalam mask, sedangkan yang tidak sesuai diberi nilai 0 (hitam). Ini memisahkan objek dan latar belakang dalam citra, di mana nilai 1

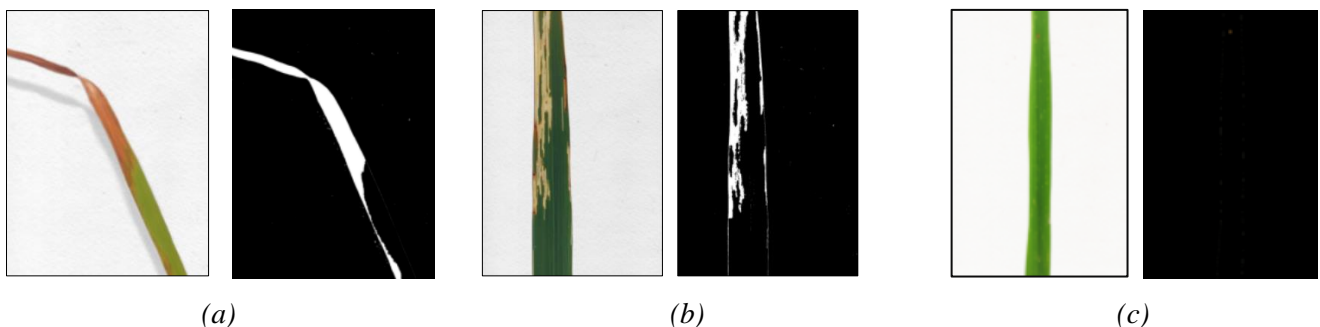
mewakili objek dan nilai 0 mewakili latar belakang. Mask juga dibuat untuk objek daun yang terinfeksi penyakit dengan menggunakan batas nilai HSV yang telah ditentukan sebelumnya.

4. Inversi *mask* latar belakang

Mask latar belakang yang telah diidentifikasi sebelumnya diubah menjadi *mask invers*, dimana piksel yang semula memiliki nilai 1 (putih) menjadi 0 (hitam), dan sebaliknya. Tujuan dari inversi *mask* adalah untuk mempersiapkan *mask* yang digunakan untuk menghilangkan latar belakang dari citra asli.

5. Penghapusan latar belakang

Selanjutnya dilakukan penghapusan piksel latar belakang. Penghapusan ini menerapkan antara piksel gambar asli dan piksel dari *mask invers* latar belakang. Piksel yang memiliki nilai 1 (putih) dalam *mask* akan dipertahankan (yang mewakili objek) sedangkan piksel yang memiliki nilai 0 (hitam) dalam *mask* akan dihapus. Sehingga hanya objek yang tersisa dalam gambar hasil segmentasi. Berikut ini adalah hasil citra dari proses segmentasi.



Gambar. 4. Hasil Proses Segemntasi Citra, (a) Segmentasi kelas Bacterial Leaf Blight, (b) Segmentasi Kelas Hama Ulat, (c) Segementasi Kelas Normal

C. *Resizing citra*

Pada tahap ini, citra yang telah dipisahkan antara objek dan latar belakang akan diubah ukurannya agar sesuai semua. Proses *resizing* ini memerlukan ukuran target citra yang diinginkan. Dalam penelitian ini citra hasil segmentasi memiliki ukuran yang berbeda-beda, yaitu ada 1080×1440 piksel dan 1080×1608 piksel, lalu diubah menjadi gambar berukuran 224×224 piksel, agar memiliki ukuran yang sesuai semua dengan ukuran target.

D. CLAHE (*Contrast Limited Adaptive Histogram Equalization*)

CLAHE), merupakan metode untuk meningkatkan kontras citra secara lokal. Metode ini mengatasi batasan pendekatan global dengan melakukan peningkatan kontras lokal pada citra [15]. CLAHE bergantung pada dua hiperparameter, yaitu jumlah *tiles* atau *grid* dan *clip limit*. Jumlah *tiles* mengacu pada jumlah bagian kecil dari citra yang diperlukan secara independen, sedangkan *clip limit* mengatur tingkat maksimum kontras yang dapat diterapkan pada setiap tile. Clip limit dapat dihitung menggunakan persamaan (5) berikut:

$$\beta = \frac{M}{N} \left(1 + \frac{\alpha}{100} (S_{max} - 1) \right) \quad (5)$$

M adalah luas permukaan, N mewakili 256 nilai *grayscale*, dan α adalah faktor klip yang menentukan penambahan ambang histogram dari 0 hingga 100. Piksel yang melebihi *clip limit* dianggap piksel berlebih. Hal ini akan membuat histogram lebih seragam karena menyebar ke area di bawah *clip limit* [16]. Pada penelitian ini, CLAHE bekerja dengan cara membagi citra ke dalam komponen warna LAB, kemudian mengaplikasikan metode *histogram equalization* yang disesuaikan secara adaptif pada saluran luminansi (L). Ini bertujuan untuk meningkatkan kontras citra, khususnya di daerah yang gelap atau terlalu terang. Proses ini diikuti dengan menggabungkan kembali saluran warna untuk menghasilkan citra yang telah diperbaiki kontrasnya.



Gambar.5. Hasil Proses CLAHE (grid=[6,6] dan CL=3,0)

E. Pembagian data

Penelitian ini membagi data menjadi tiga bagian utama, yaitu data latih, data validasi, dan data uji. Untuk menguji performa model SVM, dilakukan dua skenario pembagian data yang berbeda. Pertama, 70% data latih, 15% data validasi, dan 15% data uji. Kedua, 80% data latih, 10% data validasi, dan 10% data uji. Pembagian data ini dilakukan dengan menggunakan library dari scikit-learn dengan fungsi “train_test_split”.

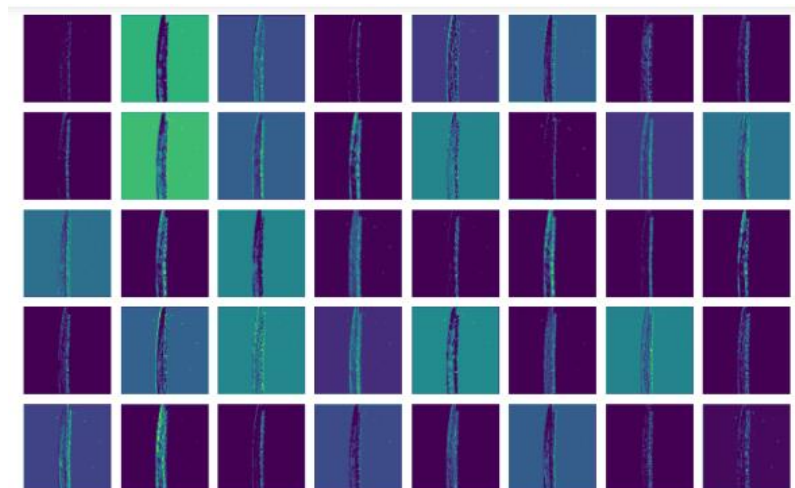
F. Ekstraksi fitur CNN VGG-19

Metode ekstraksi fitur pada penelitian ini melibatkan penerapan *Convolutional Neural Network* (CNN) sebagai *deep feature*. *Deep feature* merupakan representasi fitur yang dihasilkan oleh lapisan-lapisan dalam arsitektur *neural network*. Pemilihan arsitektur CNN yang digunakan dalam penelitian ini, yaitu VGG-19. VGG-19 adalah varian dari jaringan VGG (*Visual Geometry Group*) yang memiliki 19 lapisan yang mendalam. Arsitektur ini terdiri dari lapisan konvolusi yang terhubung dengan menggunakan *Maxpooling* untuk pengurangan dimensi dan *SoftMax* untuk klasifikasi [17]. Melalui proses pelatihan yang mendalam, CNN mampu mengekstrak fitur-fitur dari citra. Fitur-fitur tersebut dapat berupa tepi, sudut, tekstur, pola, dan objek yang sederhana maupun kompleks.



Gambar.6. Arsitektur VGG-19

Pada Gambar 6, dapat dilihat bahwa arsitektur VGG19 terdiri dari total 27 lapisan, yang terdiri dari 1 lapisan input, 16 lapisan konvolusi, 5 lapisan MaxPooling, 1 lapisan Flatten, 3 lapisan Dense, dan 1 lapisan output. Di lapisan konvolusi, filter atau kernel dilakukan operasi geser (*sliding*) melintasi citra untuk mendeteksi fitur tertentu, seperti garis, sudut, atau tekstur, hingga fitur yang lebih kompleks dan abstrak di lapisan yang lebih dalam. Setiap filter menghasilkan peta ciri (*feature map*) yang menggambarkan sejauh mana fitur-fitur tersebut muncul dalam citra.



Gambar.7. Hasil *Feature Map* Dalam Lapisan Konvolusi

Selain pemilihan arsitektur CNN, parameter-parameter pelatihan model juga diatur. Pada penelitian ini, model dikompilasi menggunakan optimizer Adam dengan learning rate sebesar 0.0001, dan *loss function* yang digunakan adalah *categorical crossentropy*. Proses pelatihan model dilakukan dengan menggunakan ”metode fit()” pada TensorFlow, dengan jumlah epochs sebanyak 10.

G. Model pelatihan SVM

Support Vector Machine (SVM) adalah salah satu model pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. SVM merupakan pengembangan dari *Support Vector Classifier* (SVC) yang memungkinkan pemodelan *decision boundary* sebagai fungsi non-linear melalui penggunaan teknik kernel. *Decision boundary* sendiri adalah *hyperplane* yang membagi ruang fitur menjadi dua bagian, di mana setiap bagian mewakili satu kelas

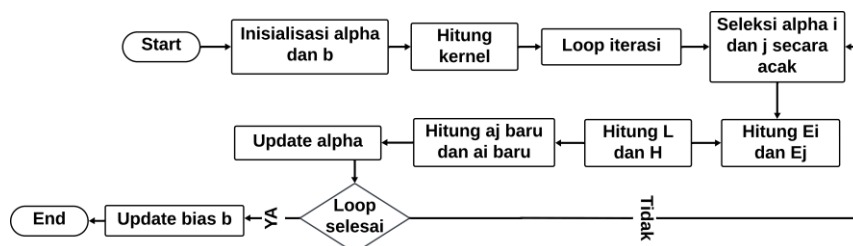
[9]. SVM bertujuan untuk mencari *decision boundary* yang memaksimalkan margin antara kelas-kelas yang berbeda dalam dataset. Margin ini adalah jarak antara *decision boundary* dan *support vectors* [9]. Dengan menggunakan berbagai macam fungsi kernel, SVM dapat menangani data yang tidak linier dengan baik. Berikut adalah berbagai macam fungsi kernel yang umum digunakan:

Tabel I

PERBEDAAN FUNGSI KERNEL SVM

Fungsi kernel	Rumus	Parameter
Linear	$K(x, x_i) = x * x_i$	C
Polynomial	$K(x, x_i) = [\gamma * (x, x_i) + coef]^d$	$C, \gamma, coef, \text{ dan } d$
Radial Basis Function (RBF)	$K(x, x_i) = \exp(-\gamma * \ x, x_i\ ^2)$	$C, \text{ dan } \gamma$
Sigmoid	$K(x, x_i) = \tanh(\gamma(x, x_i) + coef)$	$C, \gamma, \text{ dan } coef$

Selanjutnya dalam melatih model SVM, ada beberapa model algoritma yang dapat digunakan, salah satu yang umum digunakan adalah *Sequential Minimal Optimization* (SMO). SMO adalah algoritma yang digunakan untuk memecahkan masalah *quadratic programming* (QP) yang muncul selama pelatihan SVM [18]. SMO digunakan untuk menemukan nilai *lagrange* atau α yang optimal. *Lagrange* ini digunakan dalam formulasi masalah *optimasi quadratic* SVM, yang berperan dalam menentukan posisi dan orientasi hyperplane pemisah [18].



Gambar.8. Tahapan Pelatihan SMO

Dalam penelitian ini pelatihan SVM dilatih dengan menggunakan algoritma dari SMO. Proses pelatihan model SVM menggunakan SMO melibatkan beberapa parameter yang diuji, yaitu *cost* (C), *gamma* (γ), *degree* (d), *coef*, dan kernel. C (*Cost*) adalah parameter yang mengontrol trade-off antara kompleksitas dan kesalahan margin. Alur dari tahapan pelatihan menggunakan optimasi SMO dapat dilihat pada Gambar 8. Dalam tahap SMO, pertama dilakukan inisialisasi nilai *Lagrange/alpha* dan parameter bias (b) ke 0. Setelah inisialisasi, dilakukan perhitungan kernel. Fungsi kernel yang digunakan dalam penelitian ini adalah *polynomial*, RBF, sigmoid, dan linear. Selanjutnya memasuki iterasi, Dalam setiap iterasi, dipilih dua *alpha* secara acak, yaitu, a_i dan a_j , dari sampel data pada set pelatihan SVM, dimana $i \neq j$.

Langkah awal dalam proses optimisasi adalah menghitung kesalahan (E_i , dan E_j) untuk masing-masing *alpha* yang terpilih. Rumus yang diterapkan untuk perhitungan kesalahan E_i dan E_j adalah sebagai berikut [18]:

$$E_i = f(x_i) - y_i \tag{10}$$

$$E_j = f(x_j) - y_j \tag{11}$$

$$f(x) = \left(\sum_{j=1}^n a_j y_j K(x_j, x_i) + b \right) \tag{11}$$

$f(x)$ merupakan fungsi yang diprediksi, yang dihitung menggunakan kernel antara titik data x_i dan x_j , serta nilai *alpha* yang sesuai. Selain itu, n adalah jumlah sampel dalam dataset, $K(x_j, x_i)$ adalah nilai kernel antara x_i dan x_j , dan b adalah bias. Setelah mendapatkan nilai E_i , dan E_j , Selanjutnya, adalah menentukan batas atas (H) dan batas bawah (L). Perhitungan L dan H bergantung pada label kelas y_i , dan y_j . Label ini dapat memiliki nilai yang sama atau berbeda [18].

Ketika, $y_i = y_j$. Maka persamaan L dan H :

$$L = \max(C, a_j - a_i - C), H = \min(C, a_j + a_i) \tag{12}$$

Ketika, $y_i \neq y_j$. Maka persamaan L dan H :

$$L = \max(0, a_j - a_i), H = \min(C, C + a_j - a_i) \tag{13}$$

Kemudian, langkah selanjutnya adalah perhitungan pembaruan nilai a_j dan a_i . Salah satu tahap penting dalam perhitungan ini adalah menghitung nilai *eta* (η), yang digunakan untuk memperbarui nilai a_j dalam proses optimisasi [18]. Persamaan perhitungan η , sebagai berikut:

$$\eta = 2K(x_i, x_j) - K(x_i, x_i) - K(x_j, x_j) \quad (14)$$

Selanjutnya, didapatkan perbaruan nilai a_j sebagai berikut:

$$a_j^{new} = a_j - \frac{y_i(E_i - E_j)}{\eta} \quad (15)$$

Karena dibatasi oleh L dan H, maka pembaruan nilai a_j^{new} adalah :

$$a_j^{new} = \begin{cases} H, & \text{Jika } a_j^{new} \geq H; \\ a_j^{new}, & \text{Jika } L < a_j^{new} < H; \\ L, & \text{Jika } a_j^{new} \leq L; \end{cases} \quad (16)$$

$$\text{Atau, } a_j^{new} = \min(\max(a_j^{new}, L), H)$$

Sementara untuk mendapatkan pembaruan nilai a_i , sebagai berikut:

$$a_i^{new} = a_i + y_i y_j (a_j^{old} - a_j^{new}) \quad (17)$$

Setelah perubahan pada nilai a , selanjutnya adalah memperbarui bias. Bias diperbarui berdasarkan perubahan a dan $error$. Persamaan perhitungan pembaruan nilai bias, sebagai berikut:

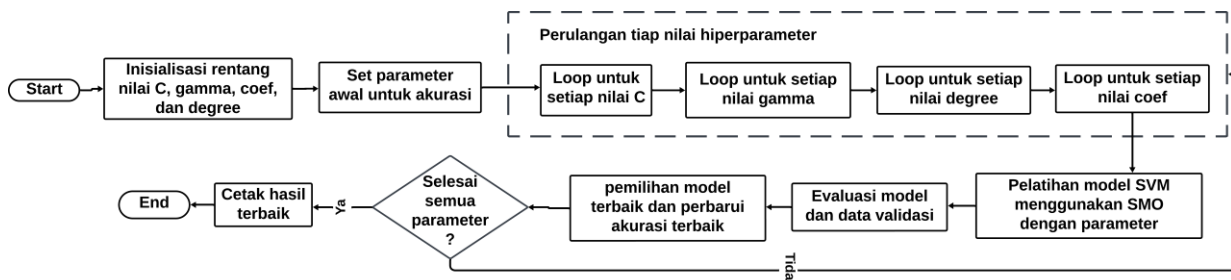
$$b_i = b - E_i - y_i(a_i^{new} - a_i)K(x_i, x_i) - y_j(a_j^{new} - a_j)K(x_i, x_j) \quad (18)$$

$$b_j = b - E_j - y_j(a_i^{new} - a_i)K(x_i, x_j) - y_j(a_j^{new} - a_j)K(x_j, x_j) \quad (19)$$

Setelah pembaruan nilai bb , terdapat kondisi yang harus dipenuhi untuk offset b , yaitu:

$$b = \begin{cases} b_i, & \text{Jika } 0 < a_i^{new} < C \\ b_j, & \text{Jika } 0 \leq a_j^{new} \leq C \\ b_i + b_j, & \text{lainnya} \end{cases} \quad (20)$$

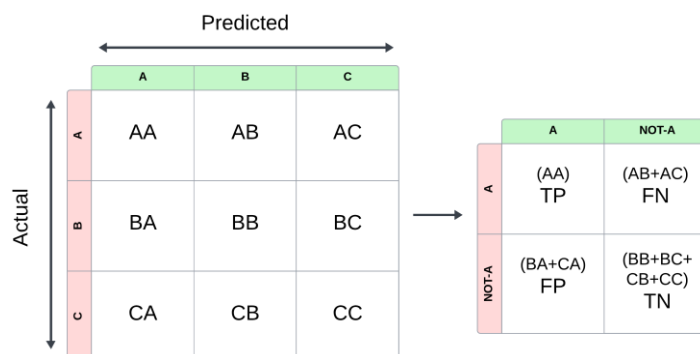
Setelah pelatihan SVM selesai dengan metode SMO, langkah berikutnya adalah menentukan parameter terbaik dengan menetapkan rentang nilai untuk C , γ , degree, dan coef. Proses ini melibatkan pelatihan model dengan berbagai kombinasi nilai parameter untuk memilih yang paling optimal. Setelah parameter terbaik ditemukan, selanjutnya diuji performa model SVM menggunakan parameter tersebut dengan data uji. Langkah-langkah dalam menentukan parameter terbaik dengan menggunakan data uji dapat dilihat dalam Gambar 9.



Gambar.9. Tahapan Penentuan Parameter Terbaik

H. Evaluasi performa

Penelitian ini menggunakan *confusion matrix* dari *library sklearn* untuk mengevaluasi performa model klasifikasi dalam berbagai skenario. *Confusion matrix* adalah sebuah metode yang digunakan dalam evaluasi kinerja suatu algoritma machine learning dengan memvisualisasikan hasil prediksi terhadap data aktual [19]. Matriks ini terdiri dari empat bagian: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP adalah jumlah data positif yang diprediksi dengan benar, TN adalah jumlah data negatif yang diprediksi dengan benar, FP adalah jumlah data negatif yang salah diprediksi sebagai positif, dan FN adalah jumlah data positif yang salah diprediksi sebagai negatif [19].



Gambar.10. Confusion Matrix

Dari confusion matrix, dapat menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F-1 Score*, yang memberikan gambaran seberapa baik model mampu mengklasifikasikan ke dalam kelas yang benar.

1. Akurasi

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

2. Presisi

$$Presisi = \frac{TP}{TP+FP} \quad (22)$$

3. *Recall*

$$Recall = \frac{TP}{TP+FN} \quad (23)$$

4. *F-1 Score*

$$F - 1 Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (24)$$

III. HASIL PEMBAHASAN

Pada penelitian ini, terdapat 8 skenario model klasifikasi SVM. Skenario-skenario ini didasarkan pada hasil ekstraksi fitur dari VGG19 menggunakan empat jenis kernel: *polynomial*, RBF (*Radial Basis Function*), sigmoid, dan linear. Selain itu, data dibagi menjadi empat variasi: 70:15:15 (70% digunakan sebagai data pelatihan, 15% digunakan sebagai data validasi, dan 15% digunakan sebagai data uji), dan 80:10:10 (80% digunakan sebagai data pelatihan, 10% digunakan sebagai data validasi, dan 10% digunakan sebagai data uji). Setiap skenario kemudian diuji dan performanya dievaluasi berdasarkan data uji. Hasil terbaik dari setiap skenario akan dipilih.

A. Klasifikasi SVM dengan kernel *polynomial*

Pada penggunaan fungsi kernel *polynomial*, diujikan beberapa nilai *hyperparameter*. Rentang nilai C yang diuji meliputi 0.1, 1, 10, dan 100; nilai gamma dengan variasi 0.001, 0.01, 0.1, 1, dan 0.031; nilai coef yang digunakan adalah 0 dan 1. Selanjutnya, nilai degree yang diuji adalah 2, 3, 4, dan 5. Nilai-nilai ini dipilih untuk menguji performa model dalam berbagai konfigurasi parameter. Hasil uji dilakukan pada dua proporsi data, yaitu 70:15:15 dengan 231 data latih, 50 data validasi, dan 49 data uji; serta 80:10:10 dengan 264 data latih, 33 data validasi, dan 33 data uji. Selanjutnya, dilakukan analisis performa terhadap masing-masing model dan dipilih model terbaik berdasarkan hasil akurasi. Hasil akurasi untuk setiap jenis pembagian data tercantum dalam Tabel (2), Tabel (3). Sebagai kriteria seleksi, hanya nilai akurasi di atas 50% yang diambil.

TABEL II
 NILAI AKURASI DENGAN PEMBAGIAN 70:15:15 MENGGUNAKAN FUNGSI POLYNOMIAL

	C	γ	coef	d	Akurasi
70:15:15	0,1	0,001	0	2	0,8367
	0,1	0,001	0	3	0,5918
	0,1	0,001	1	4	0,61220
	0,1	0,01	1	3	0,5714
	0,1	0,001	0	5	0,5306

TABEL III
 NILAI AKURASI DENGAN PEMBAGIAN 80:10:10 MENGGUNAKAN FUNGSI POLYNOMIAL

	C	γ	coef	d	Akurasi
80:10:10	0,1	0,001	0	2	0,8182
	0,1	0,001	0	3	0,5152
	0,1	0,001	1	2	0,5758
	0,1	0,001	1	3	0,9091
	0,1	1	0	3	0,8788
	10	0,1	1	5	0,8485

Berdasarkan Tabel (2) dan Tabel (3), analisis kinerja model klasifikasi dengan kernel *polynomial* pada dua skema pembagian data yang berbeda menunjukkan pengaruh signifikan dari pemilihan parameter terhadap efektivitas model. Dalam skema pembagian data 80:10:10, dengan parameter optimal, yaitu C = 0.1, gamma = 0.001, coef = 1, dan degree = 3, model ini berhasil mencapai akurasi, yaitu 90.91%. Sebaliknya, pada skema pembagian data 70:15:15, dengan parameter C = 0.1, gamma = 0.001, degree = 2, dan coef = 0, model berhasil mencapai akurasi

83.67%. Dengan demikian, dalam kernel *polynomial* model dengan parameter $C=0.1$, $\gamma=0.001$, $\text{coef}=1$, dan $\text{degree}=3$ pada pembagian data 80:10:10 merupakan model terbaik berdasarkan akurasi. Untuk Informasi lebih rinci tentang distribusi dan kesalahan klasifikasi pada masing-masing skema, merujuk ke Tabel (4) dan Tabel (5) yang menunjukkan hasil untuk pembagian data 70:15:15. Untuk pembagian data 80:10:10, detailnya dapat dilihat pada Tabel (6) dan Tabel (7), dimana keduanya menyajikan hasil *confusion matrix*.

TABEL IV
 CONFUSION MATRIX PADA PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL *POLYNOMIAL*

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	17	1	0
Hama ulat (1)	7	10	0
Normal (2)	0	0	14

TABEL V
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL *POLYNOMIAL*

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	17	7	24	1	0,8367	0,7083	0,9444	0,8095
Hama ulat (1)	10	1	31	7	0,8367	0,9091	0,5882	0,7143
Normal (2)	14	0	35	0	1,00	1,00	1,00	1,00

TABEL VI
 CONFUSION MATRIX PADA PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL *POLYNOMIAL*

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	10	1	0
Hama ulat (1)	2	9	0
Normal (2)	0	0	11

TABEL VII
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL *POLYNOMIAL*

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	10	2	20	1	0,9091	0,8333	0,9091	0,8696
Hama ulat (1)	9	1	21	2	0,9091	0,9000	0,8182	0,8571
Normal (2)	11	0	22	0	1,0	1,0	1,0	1,0

B. Klasifikasi dengan kernel *sigmoid*

Pada penggunaan fungsi kernel *sigmoid*, diujikan beberapa nilai *hyperparameter*. Rentang nilai C yang diuji meliputi 0.1, 1, 10, dan 100; nilai γ dengan variasi 0.001, 0.01, 0.1, 1, dan 0.031; nilai coef adalah 0 dan. Uji dilakukan dengan membagi skenario data menjadi 80:10:10 dan 70:15:15. Selanjutnya, dilakukan analisis performa terhadap masing-masing model dan dipilih model terbaik berdasarkan hasil akurasi. Hasil akurasi untuk setiap jenis pembagian data tercantum dalam Tabel (8), Tabel (9). Sebagai kriteria seleksi, hanya nilai akurasi di atas 50% yang diambil.

TABEL VIII
 NILAI AKURASI DENGAN PEMBAGIAN 70:15:15 MENGGUNAKAN FUNGSI SIGMOID

	C	γ	coef	Akurasi
70:15:15	0,1	0,001	0	0,5918

TABEL IX
 NILAI AKURASI DENGAN PEMBAGIAN 80:10:10 MENGGUNAKAN FUNGSI SIGMOID

	C	γ	coef	Akurasi
80:10:10	0,1	0,1	0	0,6667

Berdasarkan Tabel (6) dan Tabel (7), analisis kinerja model klasifikasi dengan kernel *sigmoid* pada dua skema pembagian data yang berbeda menunjukkan pengaruh signifikan dari pemilihan parameter terhadap efektivitas model. Dalam skema pembagian data 80:10:10, dengan parameter optimal, yaitu $C = 0.1$, $\gamma = 0.1$, dan $\text{coef} = 0$, model ini berhasil mencapai akurasi, yaitu 66,67%. Sebaliknya, pada skema pembagian data 70:15:15, dengan parameter $C = 0.1$, $\gamma = 0.001$, dan $\text{coef} = 0$, model berhasil mencapai akurasi 59,18%. Dengan demikian, model kernel *sigmoid* dengan parameter $C=0.1$, $\gamma=0.1$, dan $\text{coef}=0$ pada pembagian data 80:10:10 merupakan model terbaik berdasarkan akurasi. Untuk Informasi lebih rinci tentang distribusi dan kesalahan klasifikasi pada masing-masing skema, merujuk ke Tabel (10) dan Tabel (11) yang menunjukkan hasil untuk pembagian data 70:15:15. Untuk pembagian data 80:10:10, detailnya dapat dilihat pada Tabel (12) dan Tabel (13), dimana keduanya menyajikan hasil *confusion matrix*.

TABEL X
 CONFUSION MATRIX PADA PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL *SIGMOID*

	BLB (0)	Hama ulat (1)	Normal (2)
--	---------	---------------	------------

BLB (0)	15	3	0
Hama ulat (1)	15	0	2
Normal (2)	0	0	14

TABEL XI

NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL *SIGMOID*

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	15	15	16	3	0,6327	0,5000	0,8333	0,6250
Hama ulat (1)	0	3	29	17	0,6923	0	0	0
Normal (2)	14	2	33	0	0,9592	0,8750	1,0	0,9333

TABEL XII

CONFUSION MATRIX PADA PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL *SIGMOID*

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	11	0	0
Hama ulat (1)	11	0	2
Normal (2)	0	0	11

TABEL XIII

NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL *SIGMOID*

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	11	11	11	0	0,6667	0,5000	1,0	0,6667
Hama ulat (1)	0	0	22	11	0,7500	0	0	0
Normal (2)	11	0	22	0	1,0	1,0	1,0	1,0

C. Klasifikasi dengan kernel RBF

Pada penggunaan fungsi kernel RBF, diujikan beberapa nilai *hyperparameter*. Rentang nilai C yang diuji meliputi 0.1, 1, 10, dan 100; dan nilai gamma dengan variasi 0.001, 0.01, 0.1, 1, dan 0.031. Selanjutnya, dilakukan analisis performa terhadap masing-masing model dan dipilih model terbaik berdasarkan hasil akurasi. Hasil akurasi untuk setiap jenis pembagian data tercantum dalam Tabel (14), Tabel (15). Sebagai kriteria seleksi, hanya nilai akurasi di atas 50% yang diambil.

TABEL XIV
 NILAI AKURASI DENGAN PEMBAGIAN 70:15:15 MENGGUNAKAN FUNGSI RBF

	C	γ	Akurasi
70:15:15	0,1	0,001	0,8163
	0,1	0,01	0,8367
	0,1	0,1	0,6531
	1	0,001	0,8571
	1	0,1	0,7143
	10	0,01	0,8776
	100	0,1	0,7347

TABEL XV
 NILAI AKURASI DENGAN PEMBAGIAN 80:10:10 MENGGUNAKAN FUNGSI RBF

	C	γ	Akurasi
80:10:10	0,1	0,01	0,7576
	0,1	0,1	0,9091
	0,1	1	0,6061
	0,1	0,031	0,9394
	100	0,001	0,5758

Berdasarkan Tabel (10) dan Tabel (11), analisis kinerja model klasifikasi dengan kernel RBF pada dua skema pembagian data yang berbeda menunjukkan pengaruh signifikan dari pemilihan parameter terhadap efektivitas model. Dalam skema pembagian data 80:10:10, dengan parameter optimal, yaitu $C = 0.1$, dan $\gamma = 0.031$, model ini berhasil mencapai akurasi, yaitu 93,94%. Sebaliknya, pada skema pembagian data 70:15:15, dengan parameter $C = 10$, dan $\gamma = 0.01$, model berhasil mencapai akurasi 87,76%. Dengan demikian, model kernel RBF dengan parameter $C=0,1$, dan $\gamma=0.031$, pada pembagian data 80:10:10 merupakan model terbaik berdasarkan akurasi. Untuk Informasi lebih rinci tentang distribusi dan kesalahan klasifikasi pada masing-masing skema, merujuk ke Tabel (16) dan Tabel (17) yang menunjukkan hasil untuk pembagian data 70:15:15. Untuk pembagian data 80:10:10, detailnya dapat dilihat pada Tabel (18) dan Tabel (19), dimana keduanya menyajikan hasil *confusion matrix*.

TABEL XVI
 CONFUSION MATRIX PADA PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL RBF

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	17	1	0
Hama ulat (1)	5	12	0
Normal (2)	0	0	14

Tabel XVII
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL RBF

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	17	5	26	1	0,8776	0,7727	0,9444	0,8507
Hama ulat (1)	12	1	31	5	0,8776	0,9231	0,7059	0,8000
Normal (2)	14	0	35	0	1,0	1,0	1,0	1,0

TABEL XVIII
 CONFUSION MATRIX PADA PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL RBF

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	11	0	0
Hama ulat (1)	2	19	0
Normal (2)	0	0	11

TABEL XIX
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL RBF

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	11	2	20	0	0,9394	0,8462	1,0	0,9167
Hama ulat (1)	9	0	22	2	0,9623	1,0	0,8182	0,9000
Normal (2)	11	0	22	0	1,0	1,0	1,0	1,0

D. Klasifikasi dengan kernel linear

Pada penggunaan fungsi kernel linear, diujikan nilai *hyperparameter* nilai C rentang yang diuji meliputi 0.1, 1, 10, dan 100. Selanjutnya, dilakukan analisis performa terhadap masing-masing model dan dipilih model terbaik berdasarkan hasil akurasi. Hasil akurasi untuk setiap jenis pembagian data tercantum dalam Tabel (20), Tabel (21). Sebagai kriteria seleksi, hanya nilai akurasi di atas 50% yang diambil.

TABEL XX
 NILAI AKURASI DENGAN PEMBAGIAN 70:15:15 MENGGUNAKAN FUNGSI LINEAR

	C	Akurasi
70:15:15	0,1	0,8367
	1	0,9091
	10	0,9394
	100	0,7959

TABEL XXI
 NILAI AKURASI DENGAN PEMBAGIAN 80:10:10 MENGGUNAKAN FUNGSI LINEAR

	C	Akurasi
80:10:10	0,1	0,7879
	1	0,9091
	10	0,9394
	100	0,8788

Berdasarkan Tabel (14) dan Tabel (15), analisis kinerja model klasifikasi dengan kernel linear pada dua skema pembagian data yang berbeda menunjukkan pengaruh signifikan dari pemilihan parameter terhadap efektivitas model. Dalam skema pembagian data 80:10:10, dengan parameter optimal, yaitu $C = 10$, model ini berhasil mencapai akurasi, yaitu 93,94%. Sebaliknya, pada skema pembagian data 70:15:15, dengan parameter $C = 0.1$, model berhasil mencapai akurasi 83,67%. Dengan demikian, model kernel linear dengan parameter $C=10$, pada pembagian data 80:10:10 merupakan model terbaik berdasarkan akurasi. Untuk Informasi lebih rinci tentang distribusi dan kesalahan klasifikasi pada masing-masing skema, merujuk ke Tabel (22) dan Tabel (23) yang menunjukkan hasil untuk pembagian data 70:15:15. Untuk pembagian data 80:10:10, detailnya dapat dilihat pada Tabel (24) dan Tabel (25), dimana keduanya menyajikan hasil *confusion matrix*.

TABEL XXII
 CONFUSION MATRIX PADA PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL LINEAR

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	17	1	0
Hama ulat (1)	7	10	0
Normal (2)	0	0	14

TABEL XXIII
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 70:15:15 DENGAN FUNGSI KERNEL LINEAR

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	17	7	24	1	0,8367	0,7083	0,9444	0,8095
Hama ulat (1)	10	1	31	7	0,8367	0,9091	0,5882	0,7143
Normal (2)	14	0	35	0	1,0	1,0	1,0	1,0

Tabel XXIV
 CONFUSION MATRIX PADA PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL LINEAR

	BLB (0)	Hama ulat (1)	Normal (2)
BLB (0)	11	0	0
Hama ulat (1)	2	9	0
Normal (2)	0	0	11

TABEL XXV
 NILAI PERSENTASE KLASIFIKASI UNTUK PEMBAGIAN DATA 80:10:10 DENGAN FUNGSI KERNEL LINEAR

Jenis kelas	TP	FP	TN	FN	Akurasi	Precision	Recall	F1-Score
BLB (0)	11	2	20	0	0,9394	0,8462	1,0	0,9167
Hama ulat (1)	9	0	22	2	0,9623	1,0	0,8182	0,9000
Normal (2)	11	0	22	0	1,0	1,0	1,0	1,0

Berikut adalah perbandingan akurasi terbaik antara masing-masing kernel pada kedua skenario pembagian data, yang tercantum dalam Tabel (26). Dari tabel tersebut, dapat diamati bahwa terdapat variasi yang signifikan dalam kinerja model SVM dengan kernel yang berbeda tergantung pada pembagian data. Pada skenario pembagian data 80:10:10, model dengan kernel RBF dan linear mencapai akurasi tertinggi, masing-masing sebesar 93,94% dan 93,94%. Sedangkan pada skenario yang sama, model dengan kernel sigmoid dan polynomial menunjukkan akurasi yang lebih rendah, yaitu 66,67% dan 90,91%.

TABEL XXVI
 PERBANDINGAN AKURASI TERBAIK MODEL SVM DENGAN BERBAGAI KERNEL PADA BERBAGAI PROPORSI DATA

Kernel	Proporsi data	C	Gamma	Coef	Degree	Akurasi
Polynomial	70:15:15	0,1	0,001	0	2	0,8367
Polynomial	80:10:10	0,1	0,001	1	3	0,9091
Sigmoid	70:15:15	0,1	0,001	0	-	0,5918
Sigmoid	80:10:10	0,1	0,1	0	-	0,6667
RBF	70:15:15	10	0,01	-	-	0,8776
RBF	80:10:10	0,1	0,031	-	-	0,9394
Linear	70:15:15	0,1	-	-	-	0,8367
Linear	80:10:10	10	-	-	-	0,9394

Dari Tabel (26), didapatkan bahwa pengaturan optimal nilai *hyperparameter* memiliki pengaruh signifikan terhadap kinerja model pada kernel RBF dan Linear. Parameter gamma pada kernel RBF dan parameter C pada kernel Linear memiliki peran dalam meningkatkan akurasi prediksi. Parameter gamma pada kernel RBF mengontrol kehalusan keputusan yang dibuat oleh model. Pada kernel RBF, nilai gamma yang lebih rendah, seperti yang terlihat pada proporsi data 80:10:10 dengan gamma sebesar 0,031, memungkinkan model untuk membuat keputusan yang lebih halus dan kurang kompleks. Dengan demikian, model menjadi lebih fleksibel dan mampu menggeneralisasi pola dengan lebih baik. Di sisi lain, nilai gamma yang terlalu besar dapat menyebabkan model menjadi terlalu sensitif terhadap data pelatihan, yang menghasilkan overfitting.

Sementara itu, parameter C pada kernel Linear mengontrol kekuatan regularisasi dalam model. Nilai C yang lebih tinggi menunjukkan bahwa model akan lebih memperhatikan setiap data pelatihan dan mencoba untuk memberikan pemisahan yang optimal antara kelas, yang dapat meningkatkan kinerja pada data pelatihan. Namun, nilai C yang terlalu tinggi juga dapat menyebabkan model menjadi terlalu kompleks dan cenderung overfitting pada data pelatihan. Oleh karena itu, memilih nilai C yang tepat penting untuk mencapai keseimbangan. Pada kernel linear, nilai C yang lebih tinggi, seperti pada proporsi data 80:10:10 dengan nilai C sebesar 10, memberikan penalti yang lebih besar terhadap kesalahan. Ini mendorong model untuk menyesuaikan dengan data pelatihan sebaik mungkin tanpa mengorbankan generalisasi. Akibatnya, model menjadi lebih cenderung untuk menangkap pola yang sebenarnya dalam data tanpa terlalu terpengaruh oleh outlier atau gangguan dalam data pelatihan. Namun, tidak hanya nilai *hyperparameter* yang berpengaruh, tetapi juga pembagian data menjadi peran penting. Pada proporsi data 80:10:10, model memiliki lebih banyak data untuk melatih, yang dapat menghasilkan model yang lebih baik.

Untuk meningkatkan penelitian di masa mendatang, perlu dilakukan eksperimen lebih lanjut yang berbeda. Beberapa pendekatan dan teknik tambahan dapat dieksplorasi. Pertama, selain menggunakan ekstraksi fitur dari arsitektur VGG19, penelitian dapat mencoba arsitektur CNN lainnya yang mungkin lebih cocok untuk tugas klasifikasi gambar tertentu. Selain itu, penggunaan teknik *preprocessing* tambahan juga dapat meningkatkan kinerja model, seperti augmentasi data, normalisasi, atau penerapan filter citra untuk pengurangan noise. Teknik-teknik ini dapat membantu meningkatkan kualitas dari citra. Dataset juga dapat diperluas baik dari segi jumlah maupun keragaman kelasnya. Memperluas dataset dengan mencakup lebih banyak sampel atau kelas yang berbeda dapat membantu model untuk belajar pola yang lebih umum. Selain itu, variasi dalam dataset juga dapat membantu dalam mengatasi masalah overfitting.

Penelitian ini memberikan kontribusi signifikan dengan memanfaatkan arsitektur CNN VGG19 untuk ekstraksi fitur dan algoritma SVM dengan *Sequential Minimal Optimization* (SMO) untuk klasifikasi penyakit daun padi. Berbeda dengan penelitian sebelumnya oleh Sultana dan Reza (2023) [12], yang menggunakan *Hybrid Convolution Neural Network* (ResNet50) dan SVM pada dataset yang sudah tersedia di *PlantVillage*, penelitian ini mengumpulkan dataset secara langsung dari lapangan. Hal ini memastikan data yang digunakan representatif dan relevan dengan kondisi nyata. Selain itu, penggunaan SMO dalam pelatihan SVM merupakan teknik yang dapat meningkatkan akurasi klasifikasi.

Hasil penelitian menunjukkan bahwa model dengan kernel RBF dan linear mencapai akurasi tertinggi sebesar 93,94% pada proporsi data 80:10:10. Penelitian ini tidak hanya menunjukkan bahwa kombinasi CNN dan SVM efektif dalam klasifikasi penyakit tanaman, tetapi juga menyoroti pentingnya pengaturan *hyperparameter* yang tepat dan pembagian data yang optimal dalam meningkatkan kinerja model.

IV. KESIMPULAN

Berdasarkan hasil penelitian ini, analisis mengungkapkan kinerja dari berbagai skenario penggunaan model klasifikasi SVM, yang mengintegrasikan hasil ekstraksi fitur menggunakan CNN VGG-19 dengan empat jenis kernel: polynomial, RBF, sigmoid, dan linear. Ada dua skema pembagian data yang digunakan dalam penelitian ini, yaitu 70% data untuk pelatihan, 15% untuk uji, dan 15% untuk validasi, serta skema kedua dengan 80% data pelatihan, 10% data uji, dan 10% data validasi. Efektivitas setiap skenario dinilai berdasarkan akurasi dan identifikasi parameter optimal untuk masing-masing model.

Dalam skenario menggunakan kernel *polynomial*, model yang paling efektif tercapai pada pembagian data 80:10:10, dengan akurasi tertinggi mencapai 90.91%, menggunakan parameter $C = 0.1$, $\gamma = 0.001$, $\text{coef} = 1$, dan $\text{degree} = 3$. Sementara itu, pada kernel sigmoid, model dengan pembagian data 80:10:10 memberikan hasil terbaik, dengan akurasi 66.67%, menggunakan parameter $C = 0.1$, $\gamma = 0.1$, dan $\text{coef} = 0$. Dalam penggunaan kernel RBF, model mencapai hasil 93,94%, pada pembagian data 80:10:10 dengan menggunakan parameter $C = 0.1$, dan $\gamma = 0.031$. Untuk kernel linear, juga memiliki hasil akurasi yang sama dengan kernel RBF yaitu 93.94%, yang dicapai pada pembagian data 80:10:10 dengan parameter $C=10$. Analisis keseluruhan ini menunjukkan bahwa pemilihan skema pembagian data dan penyesuaian parameter dapat mempengaruhi kinerja

model dalam klasifikasi daun padi berpenyakit menggunakan SVM. Secara khusus, kernel RBF dan linear menunjukkan hasil yang signifikan.

DAFTAR PUSTAKA

- [1] Vandalisna, S. Mulyono, B. Putra and I. A. Wahab, *Pengelolaan Tanaman Terpadu Pada Usaha Tani Padi*, Jakarta Selatan: ASOSIASI PROFESI WIDYAISSWARA INDONESIA, 2023.
- [2] "Luas Panen dan Produksi Padi di Indonesia 2023 (Angka Sementara)," [Online]. Available: <https://www.bps.go.id/id/pressrelease/2023/10/16/2037/luas-panen-dan-produksi-padi-di-indonesia-2023--angka-sementara-.html>. [Accessed 27 04 2024].
- [3] M. Khoiruddin, A. Junaidi and W. A. Saputra, "Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network," *Journal of Dinda : Data Science, Information Technology, and Data Analysis*, 23 02 2022.
- [4] "Dinas Ketahanan Pangan dan Pertanian Kabupaten Ngawi," 12 01 2023. [Online]. Available: <https://pertanian.ngawikab.go.id/tag/hama/>.
- [5] A. Akram, K. Fayakun and H. Ramza, "Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network," *Building of Informatics, Technology and Science (BITS)*, 27 09 2023.
- [6] Y. F. Riti and S. S. Tandjung, "Klasifikasi Covid-19 Pada Citra CT Scans Paru-Paru Menggunakan Metode Convolution Neural Network," *Progresif: Jurnal Ilmiah Komputer*, 01 02 2022.
- [7] J. Jaafari, S. Douzi, K. Douzi and B. Hssina, "Towards more efficient CNN-based surgical tools classification using transfer learning," *Journal of Big Data*, 30 08 2021.
- [8] I. D. Id, *MACHINE LEARNING: Teori, Studi Kasus dan Implementasi Menggunakan Python*, Riau: Universitas Riau Press (UR Press), 2021.
- [9] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 2023.
- [10] A. R. I. Pratama, S. A. Latipah and B. N. Sari, "OPTIMASI KLASIFIKASI CURAH HUJAN MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM) DAN RECURSIVE FEATURE ELIMINATION (RFE)," *IPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 31 05 2022.
- [11] H. Kibriya, I. Abdullah and A. Nasrullah, "Plant Disease Identification and Classification Using Convolutional Neural Network and SVM," *IEEE Xplore*, 02 02 2022.
- [12] T. Sultana and M. Reza, "Identification of Potato Leaf Diseases Using Hybrid Convolution Neural Network with Support Vector Machine," *SpringerLink*, 12 07 2023.
- [13] E. F. A. Pratama, K. and J. Jumadi, "Implementasi Metode K-Means Clustering Pada Segmentasi Citra Digital," *JURNAL MEDIA INFOTAMA*, 27 10 2022.
- [14] P. Nabilla, M. F. Saputra and R. A. Saputra, "PERBANDINGAN RUANG WARNA RGB, HSV DAN YCBCR UNTUK SEGMENTASI CITRA IKAN KEMBUNG MENGGUNAKAN K-MEANS CLUSTERING," *JATI (Jurnal Mahasiswa Teknik Informatika)*, 07 08 2022.
- [15] G. F. C. Campos, S. M. Mastelini, G. J. Aguiar, R. G. Mantovani, L. F. Melo and S. B. Jr, "Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization," *EURASIP Journal on Image and Video Processing*, 06 05 2019.
- [16] N. D. Miranda, L. Novamizanti and S. Rizal, "CONVOLUTIONAL NEURAL NETWORK PADA KLASIFIKASI SIDIK JARI MENGGUNAKAN RESNET-50," *JUTIF (Jurnal Teknik Informatika)*, 08 12 2020.
- [17] A. V. Ikehukwu, S. Murali, R. Deepu and R. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Global Transitions Proceedings*, 11 2021.
- [18] C. Zhao, L. Dai and Y. Huang, "Fractional Order Sequential Minimal Optimization Classification Method," *Fractal and Fractional*, 21 08 2023.
- [19] M. Heydarian, T. E. Doyle and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, 2022.
- [20] Y. Ren, F. Hu and H. Miao, "The optimization of kernel function and its parameters for SVM in well-logging," *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, p. 06, 2016.