

# **IMPLEMENTASI SUPPORT VECTOR MACHINE PADA KLASIFIKASI PENYAKIT DAUN TOMAT DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX DAN HUE SATURATION VALUE**

## **TUGAS AKHIR**

Sebagai syarat untuk memperoleh gelar sarjana S-1 di Program Studi Informatika, Jurusan

Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran”

Yogyakarta



Disusun oleh :

Melanio Daris Ramadhan

123200136

**PROGRAM STUDI INFORMATIKA**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**

**YOGYAKARTA**

**2025**

## HALAMAN PENGESAHAN PEMBIMBING

### IMPLEMENTASI SUPPORT VECTOR MACHINE PADA KLASIFIKASI PENYAKIT DAUN TOMAT DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX DAN HUE SATURATION VALUE

Disusun oleh:

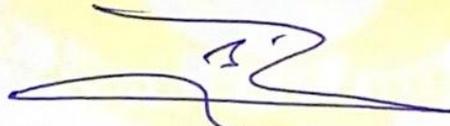
Melanio Daris Ramadhan

123200136

Telah diuji dan dinyatakan lulus oleh pembimbing

Pada tanggal : 22 April 2025

Menyetujui,  
Pembimbing



Bambang Yuwono, S.T., M.T.  
NIDN. 0512027401

Mengetahui,  
Koordinator Program Studi



Dessyanto Boedi Prasetyo, S.T., M.T.  
NIDN. 0505127501

# HALAMAN PENGESAHAN PENGUJI

## IMPLEMENTASI SUPPORT VECTOR MACHINE PADA KLASIFIKASI PENYAKIT DAUN TOMAT DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX DAN HUE SATURATION VALUE

Disusun oleh:

Melanio Daris Ramadhan

123200136

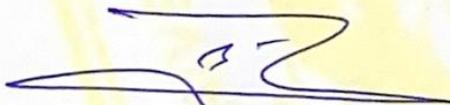
Telah diuji dan dinyatakan lulus oleh penguji

Pada tanggal : 22 April 2025

Menyetujui,

Penguji 1

Penguji 2



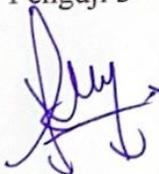
Bambang Yuwono, S.T., M.T.  
NIDN. 0512027401



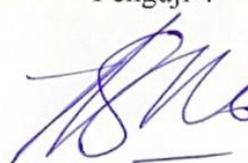
Rochmat Husaini, S.Kom., M.Kom.  
NIDN. 0026048804

Penguji 3

Penguji 4



Andrey Ferriyan, S.T., M.Cs., Ph.D.  
NUPTK. 1255761662130193



Agus Sasmito Aribowo., S.Kom., M.Cs Ph.D.  
NIP. 1975 04 12 2005 01 1001

## SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertandatangan di bawah ini, saya:

Nama : Melanio Daris Ramadhan

NIM : 123200136

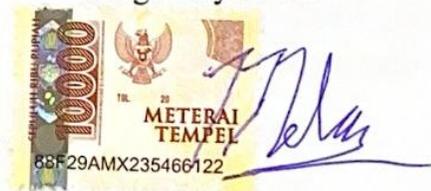
Menyatakan bahwa karya ilmiah saya yang berjudul:

***Implementasi Support Vector Machine Pada Klasifikasi Penyakit Daun Tomat Dengan Ekstraksi Fitur Gray Level Co-Occurrence Matrix dan Hue Saturation Value***

Merupakan karya asli dan belum pernah dipublikasikan dimanapun. Apabila dikemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apapun yang diberikan Program Studi Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta  
Pada tanggal : 22 April 2025  
Yang menyatakan



Melanio Daris Ramadhan  
NIM. 123200136

## PERNYATAAN BEBAS PLAGIASI

Saya yang bertandatangan di bawah ini:

Nama : Melanio Daris Ramadhan

NIM : 123200136

Fakultas/Prodi : Teknik Industri/Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir:

**Implementasi *Support Vector Machine* Pada Klasifikasi Penyakit Daun Tomat Dengan Ekstraksi Fitur *Gray Level Co-Occurrence Matrix* dan *Hue Saturation Value***

Adalah hasil kerja saya sendiri dan benar babas dari plagiasi kecuali cuplikan serta ringkasan yang dapat di dalamnya jelas sumbernya (Sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No. 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikina surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 22 April 2025  
Yang membuat pernyataan



Melanio Daris Ramadhan  
NIM.123200136

## ABSTRAK

Tomat (*Solanun Lycopersicum*) merupakan tanaman yang termasuk dalam tanaman hortikultura. Hortikultural merupakan tanaman yang biasanya ditanam oleh orang di kebun-kebun atau pekarangan rumah. Penelitian ini bertujuan untuk mengklasifikasi penyakit daun tomat dengan mengimplementasikan metode *Support Vector Machine* dengan fitur *Gray Level Co-occurrence Matrix* dan *Hue Saturation Value*. Penelitian ini menerapkan metode *Support Vector Machine* (SVM) untuk klasifikasi penyakit daun tomat dengan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) dan *Hue Saturation Value* (HSV). Hasil penelitian menunjukkan bahwa dengan menggunakan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) dan *Hue Saturation Value* (HSV) dengan *Support Vector Machine* (SVM) sebagai metode klasifikasi menghasilkan nilai akurasi sebesar 81%, nilai presisi sebesar 79%, dan nilai recall sebesar 81%. Penelitian ini menggabungkan ekstraksi fitur Hue Saturation Value (HSV) dan *Gray Level Co-occurrence Matrix* (GLCM) dengan *Support Vector Machine* (SVM) untuk mengklasifikasi penyakit daun tomat. Penelitian menggunakan GLCM dan HSV mampu mengenali fitur dari objek dengan lebih spesifik.

**Kata Kunci:** Klasifikasi Citra, SVM, HSV, GLCM, Daun Tomat.

## **ABSTRACT**

*Tomato (Solanum lycopersicum) is a plant that is included in horticultural plants. Horticultural is a plant that is usually planted by people in gardens or yards. This study aims to classify tomato leaf diseases by implementing the Support Vector Machine method with the Gray Level Co-occurrence Matrix and Hue Saturation Value features. This study applies the Support Vector Machine (SVM) method for tomato leaf disease classification by extracting Gray Level Co-occurrence Matrix (GLCM) and Hue Saturation Value (HSV) features. The results of the study show that using Gray Level Co-occurrence Matrix (GLCM) and Hue Saturation Value (HSV) feature extraction with Support Vector Machine (SVM) as a classification method produces an accuracy value of 81%, a precision value of 79%, and a recall value of 81%. This study combines Hue Saturation Value (HSV) and Gray Level Co-occurrence Matrix (GLCM) feature extraction with Support Vector Machine (SVM) to classify tomato leaf diseases. The study using GLCM and HSV was able to recognize features from objects more specifically.*

**Keywords:** *Image Classification, SVM, HSV, GLCM, Tomato Leaf.*

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “*Implementasi Support Vector Machine* pada Klasifikasi Penyakit Daun Tomat dengan Ekstraksi Fitur *Gray Level Co-occurrence Matrix* dan *Hue Saturation Value*” ini dengan baik.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional Yogyakarta. Penulisan tugas akhir ini tidak lepas dari batuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ini menyampaikan ucapan terimakasih yang sebesar besarnya kepada:

1. Allah SWT yang telah senantiasa memberikan petunjuk dan kemudahan selama penyelesaian Tugas Akhir ini.
2. Bapak Dr. Novrido Charibaldi, S.Kom, M.kom., selaku dosen wali.
3. Bambang Yuwono, S.T., M.T, selaku dosen pembimbing yang telah memberikan bimbingan, masukan, dan dorongan kepada penulis dalam menyelesaikan Tugas Akhir ini.
4. Seluruh dosen, karyawan, dan staf tata usaha prodi Informatika UPN “Veteran” Yogyakarta yang telah mendukung dan memberikan batuan kepada penulis selama menempuh pendidikan.
5. Bapak Sariman, Sri Surmarmi selaku bapak dan ibu, Anisa Daris Setyawati selaku kakak dari penulis yang telah memberikan doa, dukungan dan materiil, serta kasih sayang yang tiada henti.
6. Rusfida Endah Kurniawati selaku calon istri penulis yang selalu mendukung, menyemangati, memahami dan menemani hingga skripsi ini selesai dan Ibu calon mertua saya Ibu Sugiarti.
7. Teman-teman Informatika angkatan 2020 yang telah memberikan dukungan dan kerjasama selama masa studi.

Penulis berharap skripsi ini, khususnya memberikan kontribusi yang berarti bagi pengembangan ilmu pengetahuan, khususnya di bidang Informatika. Semoga hasil penulisan ini dapat bermanfaat tidak hanya penulis, tetapi juga para pembaca pihak-pihak lain yang berkepentingan.

Yogyakarta, 18 April 2025

Penulis

## DAFTAR ISI

HALAMAN PENGESAHAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	iv
PERNYATAAN BEBAS PLAGIASI.....	v
ABSTRAK.....	vi
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
DAFTAR MODUL PROGRAM .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metode Penelitian dan Metode Pengembangan Sistem.....	4
1.6.1 Metode Penelitian.....	5
1.6.2 Metode Pengembangan Sistem.....	5
1.7 Sistematika Penulisan.....	6
BAB II TINJAUAN LITERATUR.....	7
2.1 Tomat.....	7
2.2 <i>Computer Vision</i> .....	8
2.3 Pengolahan Citra Digital .....	9
2.4 Citra warna .....	9
2.5 Citra <i>Grayscale</i> .....	10
2.6 Citra Biner.....	10
2.7 <i>Hue Saturation Value (HSV)</i> .....	11
2.7.1 Mencari Nilai <i>Value</i> .....	12

2.7.2 Mencari Nilai <i>Saturation</i> .....	12
2.7.3 Mencari Nilai <i>Hue</i> .....	12
2.8 <i>Gray Level Co-Occurrence Matrix (GLCM)</i> .....	13
2.9 <i>Support Vector Machine (SVM)</i> .....	14
2.10 <i>Multiclass Support Vector Machine</i> .....	18
2.11 <i>Confusion Matrix</i> .....	19
2.12 Penelitian Terkait .....	20
3.1 Metodologi Penelitian .....	27
3.1.1 Analisis Masalah .....	28
3.1.2 Pengumpulan Data .....	28
3.1.3 <i>Pre-Processing Data</i> .....	29
3.1.4 Ekstraksi Fitur <i>Hue Saturation Value</i> .....	30
3.1.5 Ekstraksi Fitur <i>Grey Level Co-occurrence Matrix</i> .....	32
3.1.6 Penggabungan Nilai Fitur.....	37
3.1.7 Pelatihan Model.....	38
3.1.8 Evaluasi Sistem .....	41
3.2 Metodologi Pengembangan Sistem .....	42
3.2.1 Analisis Kebutuhan .....	42
3.2.2 Desain Sistem .....	44
3.2.3 Implementasi Sistem .....	47
3.2.4 Pengujian Sistem .....	48
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>49</b>
4.1 Hasil .....	49
4.1.1 Hasil Implementasi.....	49
4.1.2 <i>Pre-Processing</i> .....	51
4.1.3 Ekstraksi Fitur <i>Hue Saturation Value</i> .....	52
4.1.4 Ekstraksi Fitur <i>Gray Level Co-occurrence Matrix</i> .....	53
4.1.5 Klasifikasi SVM .....	55
4.2 Pengujian.....	56
4.2.1 Pengujian Model.....	56
4.2.2 Pengujian Sistem .....	63
4.3 Pembahasan.....	64

BAB V PENUTUP.....	66
5.1 Kesimpulan.....	66
5.2 Saran.....	66
DAFTAR PUSTAKA .....	67

## DAFTAR GAMBAR

Gambar 2. 1 Citra Warna.....	10
Gambar 2. 2 Citra <i>Grayscale</i> .....	10
Gambar 2. 3 Citra <i>Biner</i> .....	11
Gambar 2. 4 Visualisasi Model HSV .....	11
Gambar 2. 5 <i>Margin Hyperplane</i> Terbaik .....	15
Gambar 2. 6 <i>One-versus-One</i> .....	18
Gambar 2. 7 <i>One-versus-All</i> .....	19
Gambar 3. 1 Alur Penelitian .....	27
Gambar 3. 2 <i>Flowchart Pre-processing</i> .....	30
Gambar 3. 3 <i>Flowchart</i> Ekstraksi Fitur HSV .....	31
Gambar 3. 4 RGB Citra Daun Tomat.....	31
Gambar 3. 5 <i>Flowchart</i> Ekstraksi Fitur GLCM .....	33
Gambar 3. 6 <i>Flowchart</i> Klasifikasi SVM.....	38
Gambar 3. 7 <i>Waterfall</i> .....	42
Gambar 3. 8 Arsitektur Sistem .....	44
Gambar 3. 9 <i>Flowchart</i> Sistem.....	45
Gambar 3. 10 <i>Flowchart</i> Model .....	46
Gambar 3. 11 Rancangan <i>User Interface</i> .....	47
Gambar 4. 1 Tampilan Sistem klasifikasi .....	49
Gambar 4. 2 Tampilan Sistem Klasifikasi (Lanjutan).....	50
Gambar 4. 3 Tampilan Sistem Klasifikasi (Lanjutan).....	50
Gambar 4. 4 <i>Confusion Matrix</i> Data <i>Training</i> 80% Dan <i>Testing</i> 20% .....	57
Gambar 4. 5 <i>Confusion Matrix</i> Data <i>Training</i> 85% Dan <i>Testing</i> 15% .....	57
Gambar 4. 6 <i>Confusion Matrix</i> Data <i>Training</i> 90% Dan <i>Testing</i> 10% .....	58
Gambar 4. 7 <i>Confusion Matrix</i> dengan <i>Random State</i> 0.....	59
Gambar 4. 8 <i>Confusion Matrix</i> dengan <i>Random State</i> 21.....	60
Gambar 4. 9 <i>Confusion Matrix</i> dengan <i>Random State</i> 42.....	60
Gambar 4. 10 <i>Confusion Matrix</i> dengan <i>Random State</i> 63.....	61
Gambar 4. 11 <i>Confusion Matrix</i> dengan <i>Random State</i> 84.....	62
Gambar 4. 12 <i>K-Fold Cross Validation</i> dengan nilai $K=5$ .....	63

## DAFTAR TABEL

Tabel 2. 1 Ciri-Ciri Penyakit Daun Tomat .....	7
Tabel 2. 2 Ciri-Ciri Penyakit Daun Tomat (Lanjutan) .....	8
Tabel 2. 3 <i>Confusion Matrix</i> .....	19
Tabel 2. 4 <i>State Of Art</i> .....	21
Tabel 2. 5 <i>State Of Art</i> (Lanjutan) .....	22
Tabel 2. 6 <i>State Of Art</i> (Lanjutan) .....	23
Tabel 2. 7 <i>State Of Art</i> (Lanjutan) .....	24
Tabel 2. 8 <i>State Of Art</i> (Lanjutan) .....	25
Tabel 3. 1 Contoh Penyakit Daun Tomat .....	28
Tabel 3. 2 Contoh Penyakit Daun Tomat (Lanjutan) .....	29
Tabel 3. 3 <i>Matrix RGB</i> .....	32
Tabel 3. 4 <i>Matrix Grayscale</i> .....	33
Tabel 3. 5 Presentasi <i>Matrix GLCM</i> .....	33
Tabel 3. 6 Presentasi Perhitungan Jarak Sudut $0^\circ$ .....	34
Tabel 3. 7 Presentasi Hasil <i>Transpose</i> Sistem <i>Matrix GLCM</i> .....	34
Tabel 3. 8 Presentasi Normalisasi <i>Matrix GLCM</i> .....	35
Tabel 3. 9 Nilai Hasil Ekstraksi Fitur .....	37
Tabel 3. 10 Hasil Ekstraksi Fitur .....	38
Tabel 3. 11 Hasil Perhitungan Kernelisasi Linier .....	39
Tabel 3. 12 Hasil perhitungan <i>Matrix Hessian</i> .....	40
Tabel 3. 13 Hasil perhitungan Nilai <i>Error</i> .....	40
Tabel 3. 14 Hasil Perhitungan Nilai $\delta_i$ .....	40
Tabel 3. 15 Hasil Perhitungan $\alpha_i$ Baru .....	41
Tabel 3. 16 Evaluasi Sistem .....	42
Tabel 3. 17 Spesifikasi Perangkat Keras .....	43
Tabel 3. 18 Spesifikasi Perangkat Lunak .....	44
Tabel 3. 19 Spesifikasi Pengguna.....	44
Tabel 3. 20 Rancangan Pengujian <i>Black Box Testing</i> .....	48
Tabel 4. 1 Citra Sebelum dan Sesudah <i>Pre-processing</i> .....	52
Tabel 4. 2 Contoh Nilai Ekstraksi HSV .....	53
Tabel 4. 3 Contoh Nilai Ekstraksi Fitur GLCM Label Daun Hawar.....	54
Tabel 4. 4 Contoh Nilai Ekstraksi Fitur GLCM Label Infeksi Bakteri .....	55
Tabel 4. 5 Contoh Nilai Ekstraksi Fitur GLCM Nilai Kudis daun.....	55
Tabel 4. 6 Contoh Nilai Ekstraksi Fitur GLCM Label Daun Sehat .....	55
Tabel 4. 7 Hasil <i>Precision, Recall, F1-Score, Accuracy</i> Pengujian Pembagian Data.....	58
Tabel 4. 8 Hasil <i>Precision, Recall, F1-Score, Accuracy</i> Pengujian Random State.....	62
Tabel 4. 9 Hasil <i>Precision, Recall, F1-Score, Accuracy</i> Pengujian <i>K-Fold Cross Validation (K=5)</i> ...	63
Tabel 4. 10 Pengujian sistem <i>Black Box Testing</i> .....	64

## DAFTAR MODUL PROGRAM

Modul Program 4. 1 Fungsi Preprocessing <i>Remove Background</i> .....	51
Modul Program 4. 2 Fungsi Preprocessing <i>Resize</i> .....	51
Modul Program 4. 3 Ekstraksi Fitur HSV .....	52
Modul Program 4. 4 Ekstraksi Fitur GLCM .....	53
Modul Program 4. 5 Ekstraksi Fitur GLCM (Lanjutan).....	54
Modul Program 4. 6 Model SVM.....	55
Modul Program 4. 7 <i>Confusion Matrix</i> .....	56

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Indonesia sebagai negara agraris menghasilkan berbagai macam jenis produk industri pertanian dan perkebunan yang merupakan sektor penting sebagai penopang ekonomi masyarakat (Saputra et al., 2023). Tomat salah satu tanaman sayur yang banyak dikonsumsi oleh masyarakat Indonesia. Tomat (*Solanun Lycopersicum*) merupakan tanaman yang termasuk dalam tanaman hortikultura (Nainggolan et al., 2022). Hortikultural merupakan tanaman yang biasanya ditanam oleh orang di kebun-kebun atau pekarangan rumah (Nur'aini, 2019). Produksi tomat di Indonesia naik tiap tahunnya. Terdapat berbagai macam penyakit pada tanaman tomat, ada yang disebabkan oleh cendawan, bakteri, virus dan lain lain. Penyakit yang ada pada tanaman tomat sebagian besar terjadi pada daun tomat (Kotta et al., 2022). Untuk itu petani harus lebih teliti dalam mendeteksi penyakit pada daun tumbuhan tomat lebih awal untuk mengontrol penyebaran penyakit.

Banyak penelitian yang dilakukan untuk mengenai klasifikasi penyakit daun tomat. Menurut penelitian yang dilakukan oleh Tampinongkol et al., (2023) menggunakan algoritma SVM dan GLCM dengan dua jenis penyakit daun tomat yaitu penyakit bercak daun dan hawar menghasilkan akurasi sebesar 61,75%. Pada penelitian yang dilakukan oleh Astiningrum et al., (2020) menjelaskan bahwa implementasi sistem identifikasi penyakit pada daun tomat dapat dilakukan dengan menggunakan ekstraksi warna RGB, ekstraksi fitur GLCM dan klasifikasi KNN menghasilkan akurasi sebesar 92,89%. Penelitian Erwanto (2024) menjelaskan bahwa model YOLO versi 8 memiliki kinerja yang baik sebesar 96.6% dan *recall* sebesar 92.2%, menunjukkan kemampuan model dalam mengidentifikasi penyakit daun tomat dengan akurat. Pada penelitian yang dilakukan oleh Kurniawan (2023) bahwa pada masing-masing arsitektur CNN, arsitektur yang memiliki akurasi terbaik adalah arsitektur InceptionV3 dengan akurasi 99,67%. Kemudian peringkat selanjutnya adalah arsitektur ResNet50 dengan akurasi 97,36% dan terakhir arsitektur MobileNet dengan akurasi 85,81%. Hasil menunjukkan bahwa model CNN konvensional perlu digabungkan dengan arsitektur yang sudah diuji untuk dapat menghasilkan akurasi yang maksimal. Pengujian dengan metode CNN dalam proses pengolahan citra klasifikasi penyakit pada daun tomat yang dilakukan oleh Ningsih (2022) menghasilkan akurasi yang tinggi, proses *training* data menggunakan *learning rate* 0,0001 dan *batch size* 20. *Epoch* 1 menghasilkan *loss* 98%, akurasi 53%, *Recall* 46%.

Salah satu metode yang paling efektif dalam mengklasifikasi penyakit daun tomat yaitu *Support Vector Machine* (SVM). *Support Vector Machine* (SVM) adalah algoritma pembelajaran mesin yang dapat digunakan untuk klasifikasi dan regresi sekaligus (Putra et al. R. F., 2024). Dalam klasifikasi daun *Support Vector Machine* memiliki algoritma pembelajaran mesin yang efektif untuk memisahkan data dengan menggunakan *hyperplane* (batas keputusan) yang optimal di antara dua kelas yang berbeda (Mulyana & Muthmainnah, 2023). Algoritma

*Support Vector Machine* memiliki kelebihan dalam menghasilkan model klasifikasi yang baik walaupun dilatih menggunakan himpunan data yang sedikit (Amelia et al., 2023). Matriks GLCM merepresentasikan hubungan ketetanggaan antar piksel dalam citra pada berbagai arah orientasi dan jarak spasi (Andono et al., 2017).

Pada penelitian ini menggunakan kernel Linier untuk model SVM dari empat jenis penyakit daun yang terdapat pada tomat. Kernel linier digunakan ketika data dapat dipisahkan secara linier, yaitu dapat dipisahkan dengan menggunakan satu baris (Muhammad et al., 2023). Penelitian yang dilakukan oleh Mulyanan dan Wibowo (2023) menjelaskan bahwa identifikasi kematangan buah monk menggunakan ekstraksi GLCM dan SVM menghasilkan akurasi sebesar 89%. Dalam penelitian yang dilakukan oleh Sapura et al., (2022) menunjukkan perbandingan antar kernel RBF, Linier, Polynominal dan Sigmoid. Dalam penelitian tersebut mengidentifikasi kematangan buah melon menghasilkan pengujian terbaik oleh fungsi kernel linier dengan akurasi sebesar 78%, kernel presisi 78%, kernel *recall* sebesar 78% dan kernel *f1-score* sebesar 78%.

Penelitian terkait deteksi dini dan klasifikasi daun tanaman telah banyak dilakukan. Pada penelitian yang dilakukan oleh Suhendra (2022) menjelaskan bahwa metode SVM berhasil mengklasifikasi citra penyakit daun jagung dengan akurasi optimal sebesar 99,5%. Penelitian oleh Sooi et al., (2023) menjelaskan bahwa mendeteksi penyakit pada daun tumbuhan anggur menggunakan SVM memperoleh akurasi sebesar 97,6%. Puerwandono & Maulana (2023) menjelaskan bahwa penerapan SVM dalam mengklasifikasi jenis citra penyakit pada daun sirih mempunyai akurasi sebesar 91% untuk data training dan 80% untuk data test. Hal ini membuktikan bahwa pengklasifikasian menggunakan metode SVM merupakan metode yang cocok dan efektif dalam mengklasifikasi penyakit pada daun tumbuhan tomat.

*Gray Level Co-occurrence Matrix* didefinisikan sebagai suatu matriks frekuensi dari dua piksel yang dipisah oleh suatu *vector* yang ada dalam citra (Sinaga, 2019). Metode GLCM atau analisis pada tekstur merupakan suatu metode untuk melakukan klasifikasi pada suatu citra, GLCM pada dasarnya merupakan matriks persegi yang memiliki sifat-sifat tertentu dengan distribusi spesial (Alwy et al., 2023). Melalui fitur GLCM akan didapatkan tekstur berupa *energy*, *correlation*, *contrast*, dan *homogeneity* dari daun tomat. Algoritma GLCM merupakan sebuah metode yang efektif untuk mengekstraksi ciri atau menganalisis tekstur. Implementasi ekstraksi GLCM mampu mengoptimalkan algoritma SVM, sehingga kombinasi keduanya mampu mencapai akurasi yang tinggi.

Pada penelitian yang dilakukan oleh Adawiyah et al., (2022) yang menjelaskan bahwa metode klasifikasi SVM dan metode ekstraksi GLCM dapat digunakan untuk mengidentifikasi penyakit kulit dengan memperoleh hasil terbaik dan memberikan nilai akurasi sebesar 90%. Penelitian yang dilakukan oleh Sutisna & Sumantri (2022) menjelaskan bahwa ekstraksi ciri GLCM dan klasifikasi algoritma SVM dapat diterapkan untuk mengklasifikasi citra kematangan buah nanas dengan akurasi sebesar 86%. Pada penelitian Sriani et al., (2024) juga menjelaskan bahwa klasifikasi kelayakan mutu daun tembakau berdasarkan tekstur daun menggunakan metode GLCM dan SVM mendapatkan akurasi sebesar 73,3%. Keunggulan hasil akurasi tersebut membuktikan bahwa performa kombinasi GLCM dan SVM lebih baik dari pada

kombinasi GLCM dan M-SVM. Hal ini dibuktikan pada penelitian yang dilakukan oleh Nahak (2024) menjelaskan bahwa dalam klasifikasi penyakit pada tanaman apel melalui citra daun dengan penggabungan metode M-SVM dan GLCM akurasi tertinggi mencapai 90%.

Selain penerapan ekstraksi tekstur, dalam meningkatkan akurasi klasifikasi citra menggunakan juga harus memperhatikan ekstraksi warna. Dalam pemilihan ekstraksi warna yang tepat dapat meningkatkan kemampuan dalam mengenal objek secara signifikan. Penelitian ini menggunakan ekstraksi warna *Hue, Saturation, Value* (HSV). HSV atau akronim dari *Hue Saturation Value* merupakan sebuah ruang warna yang dianggap paling dekat dalam mendeskripsikan warna oleh mata manusia dari pada RGB (*red, green, blue*) (Zonyfar, Pengolahan Citra Digital Sebuah Pengantar, 2020). Menurut Hawa et al., (2023) model warna HSV, *Hue* merupakan atribut dari sensasi visual di mana suatu area tampak mirip dengan salah satu warna yang dirasakan, merah, kuning, hijau dan biru atau kombinasi dari keduanya. *Saturation* mengacu pada seberapa murni atau intens rona (*hue*) yang diberikan. Sedangkan *Value* mengacu pada terang atau gelapnya warna.

Penelitian yang dilakukan oleh Sari & Wihandika (2019) yang membandingkan ruang warna HSV dan LBP untuk mengklasifikasi citra makanan tunggal. Dalam penelitian tersebut menjelaskan bahwa klasifikasi fitur HSV mampu menghasilkan nilai akurasi sebesar 90,467%. Sedangkan klasifikasi fitur LBP hanya menghasilkan nilai akurasi sebesar 85,714%. Penelitian mengenai klasifikasi yang dilakukan oleh Sanusi et al., (2019) menjelaskan bahwa klasifikasi pada citra daun berdasarkan nilai HSV berhasil menampilkan nama daun di setiap proses implementasi dan memiliki akurasi sebesar 90,08%. Hal tersebut membuktikan bahwa penggunaan HSV sangat berpotensi dalam meningkatkan performa klasifikasi citra.

Berdasarkan tinjauan literatur penelitian sebelumnya, menjelaskan bahwa metode ekstraksi tekstur menggunakan GLCM dan ekstraksi fitur warna menggunakan HSV merupakan pendekatan yang efektif dalam mengklasifikasi citra penyakit yang ada dalam daun tomat. Penggunaan GLCM memungkinkan untuk pengambilan fitur penting dari daun tomat. Sementara HSV mengekstraksi informasi warna dari daun tomat. Penelitian ini menggunakan metode SVM sebagai algoritma klasifikasi karena SVM memiliki kelebihan dalam menghasilkan model klasifikasi. Pada penelitian ini, menggunakan dataset yang sedikit, sehingga penggunaan metode SVM lebih efektif digunakan. Dengan penggabungan ekstraksi tekstur dan warna serta klasifikasi metode SVM diharapkan menghasilkan klasifikasi yang lebih akurat dalam membedakan jenis penyakit pada daun tomat.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, maka dapat diambil rumusan masalah penelitian adalah:

1. Bagaimana implementasi SVM untuk klasifikasi jenis penyakit tanaman daun tomat dengan ekstraksi fitur GLCM dan HSV?

2. Bagaimana mengukur akurasi hasil klasifikasi penyakit daun tomat menggunakan metode *Support Vector Machine (SVM)* dengan kombinasi fitur *Gray Level Co-occurrence Matrix (GLCM)*, *Hue Saturation Value (HSV)*?

### 1.3 Batasan Masalah

Pembatasan permasalahan dari rumusan masalah dalam penelitian ini sebagai berikut:

1. Klasifikasi penyakit daun tomat menggunakan ekstraksi fitur tekstur dan ekstraksi fitur warna.
2. Citra penyakit daun tomat yang di gunakan terdiri dari 1000 dataset yang terbagi menjadi 4 klasifikasi yaitu 250 daun hawar, 250 terinfeksi bakteri, 250 kudis daun, 250 daun sehat.
3. Data set citra yang digunakan dalam penelitian berupa data set sekunder milik Rahmi21 yang diambil dari *kaggle* <https://www.kaggle.com/datasets/rahmi21/dataset-tomat>.
4. Pengklasifikasian yang digunakan tanpa mempertimbangkan jenis tomat.
5. Tiap citra untuk proses *training* dan *testing* memuat satu objek daun tomat dengan latar belakang seragam.
6. *Pre-processing* menggunakan *Remove Background* hanya bisa untuk gambar dengan latar belakang yang sewarna.

### 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah dapat disimpulkan tujuan dari penelitian antara lain:

1. Membangun model *Support Vector Machine (SVM)* untuk mengklasifikasi jenis penyakit daun tumbuhan tomat berdasarkan ekstraksi fitur GLCM dan HSV.
2. Mengetahui akurasi hasil klasifikasi penyakit daun tomat menggunakan metode *Support Vector Machine (SVM)* dengan kombinasi fitur *Gray Level Co-occurrence Matrix (GLCM)*, *Hue Saturation Value (HSV)*.

### 1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan mampu menjadi alternatif solusi dalam klasifikasi citra objek sekaligus mengurangi tingkat kesalahan dalam mengidentifikasi penyakit daun tomat.

### 1.6 Metode Penelitian dan Metode Pengembangan Sistem

Metodologi penelitian dan pengembangan sistem pada penelitian ini memiliki alur sebagai berikut.

### 1.6.1 Metode Penelitian

Tahapan-tahapan yang dilakukan dalam metode penelitian ini adalah sebagai berikut:

1. Pengumpulan data  
Penelitian ini menggunakan Dataset Rahmi21 yang didapatkan melalui *website* <https://www.kaggle.com/datasets/rahmi21/dataset-tomat>.
2. *Preprocessing* Data  
Sebelum data diolah ke tahap *training*, dilakukan beberapa proses pengolahan data yaitu *resize* untuk mengubah ukuran citra ke ukuran yang sama, *remove background* untuk memberikan fokus terhadap objek.
3. Ekstraksi Fitur dengan HSV  
Ekstraksi fitur dengan mencari nilai *Hue*, *Saturation*, dan *Value* dari gambar yang ada baik pada data *training* maupun data *testing* menggunakan formula yang ada.
4. Ekstraksi Fitur dengan GLCM  
Ekstraksi fitur dengan mencari nilai tekstur dari citra yang diperoleh dari nilai *Grey Level Co-occurrence Matrix* yaitu kontras (*contrast*), korelasi (*correlation*), energi (*energy*), homogenitas (*homogeneity*)
5. Pembuatan Model  
*Training* data dilakukan dengan menggunakan metode *Support Vector Machine* untuk klasifikasi penyakit daun tomat berdasarkan kelas data yang ada.
6. Evaluasi Model  
Evaluasi model ini dilakukan untuk mengukur seberapa baik model dalam mengklasifikasi kualitas penyakit daun tomat dengan akurat, di mana evaluasi ini dilakukan menggunakan metode *Confusion Matrix*.

### 1.6.2 Metode Pengembangan Sistem

Model pengembangan sistem pada penelitian ini menggunakan model *Waterfall*. Menurut Zein et al., (2022) model *Waterfall* adalah salah satu jenis model pengembangan perangkat lunak yang paling tua dan paling sederhana. Berikut penjelasan mengenai tahap-tahap dalam model *Waterfall*:

1. Analisis Kebutuhan  
Tahapan pertama adalah analisis kebutuhan. Pada tahap ini, pengembangan mengidentifikasi dan memahami kebutuhan pengguna dan sistem yang akan dibangun. Tahap ini juga merumuskan spesifikasi fungsi dan non-fungsional dari sistem.
2. Desain Sistem  
Setelah kebutuhan sistem diketahui, pada tahap ini peneliti mengembangkan merancang solusi teknis yang akan diimplementasikan. Pada tahap ini akan menentukan arsitektur sistem, membuat rancangan detail, dan menentukan teknologi yang digunakan.
3. Implementasi Sistem  
Tahap implementasi melibatkan pengembangan dan penulisan kode perangkat lunak sesuai dengan desain yang telah dibuat pada tahap sebelumnya.
4. Pengujian Sistem

Setelah tahap implementasi selesai pengembang melakukan pengujian perangkat lunak untuk memastikan bahwa perangkat lunak yang dihasilkan berfungsi sesuai kebutuhan.

## **1.7 Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam penyusunan laporan penelitian ini adalah sebagai berikut :

### **Bab I Pendahuluan**

Menjelaskan mengenai latar belakang masalah, rumusan masalah tujuan dan manfaat, serta sistematika penulisan. Pada bab ini bertujuan untuk memudahkan pembaca dalam memahami maksud dan tujuan dari penelitian ini.

### **Bab II Tinjauan Pustaka**

Menjelaskan mengenai uraian dan pembahasan terkait dasar teori dari semua aspek yang berkaitan dengan penerapan HSV dan GLCM pada algoritma SVM untuk permasalahan yang ada pada penelitian ini.

### **Bab III Metodologi Penelitian**

Menjelaskan mengenai metode yang digunakan pada penelitian dalam mengenai masalah yang diangkat dan telah di uraikan dalam latar belakang. Ekstraksi yang digunakan adalah HSV dan GLCM dan menggunakan metode SVM untuk mengklasifikasi penyakit daun tomat dengan hasil nilai akurasi dan kelas dari penyakit daun tomat.

### **Bab IV Hasil Dan Pembahasan**

Menjelaskan mengenai analisis dan pembahasan hasil implementasi dari perancangan ekstraksi HSV dan GLCM menggunakan metode SVM serta pengujian terhadap hasil penelitian.

### **Bab V Penutup**

Menjelaskan mengenai kesimpulan yang dilakukan dari penelitian serta saran yang digunakan untuk bekal pada penelitian selanjutnya.

## BAB II TINJAUAN LITERATUR

### 2.1 Tomat

Tomat (*Solanun Lycopersicum*) merupakan tanaman yang termasuk dalam tanaman hortikultura (Nainggolan et al., 2022). Hortikultura merupakan tanaman yang biasanya ditanam oleh orang di kebun-kebun atau pekarangan rumah (Nur'aini, 2019). Tanaman tomat terdiri dari akar, batang, daun, bunga dan biji. Sewaktu muda batangnya berbentuk bulat dan teksturnya lunak, tetapi setelah tua batangnya berubah menjadi bersudut dan bertekstur seperti berkayu. Daunnya berwarna hijau dan berbulu mempunyai panjang sekitar 20-30 cm dan lebar 1520 cm. Daun tomat tumbuh di ikat ujung dahan atau cabang. Sementara tangkainya berbentuk bulat memanjang sekitar 7-10 cm dan ketebalan 0,3-0,5 (Wiriyanta, 2002).

Adapun beberapa jenis penyakit yang tampak pada pada daun tomat yakni penyakit *yellow lead curl virus*, *potato virus y*, *tomato mozaik virus*, bercak daun (*Septoria leaf spot*), *bacterial spot*, *target spot* bercak, daun kering alternaria (*early blight*), busuk daun (*late blight*), daun berjamur (*leaf mold*), dan berasal dari gigitan serangga (*two-spot spider mite*). Penyakit daun tomat dapat diamati secara *macroscopic*, sehingga perubahannya dapat diamati menggunakan kamera digital. *Macroscopic* merupakan pengamatan dengan mata telanjang tanpa menggunakan alat bantu pembesar (Tampinongkol et a., 2023). Pada penelitian ini menggunakan sampel data jenis penyakit *yellow leaf curl virus*, daun kering alternaria (*early blight*), *bacterial spot* dan daun sehat untuk identifikasi. Contoh sampel data yang dapat dilihat pada Tabel 2.1.

**Tabel 2. 1 Ciri-Ciri Penyakit Daun Tomat**

No	Nama Penyakit	Sampel Gambar	Ciri-Ciri
1.	<i>Hawar Alternaria</i> (penyakit daun hawar)		Memiliki gejala awal daun seperti terbakar dari tepi daun dan layu. Memiliki bercak coklat kehitaman pada daun, kemudian menyebar pada ranting dan batang. Dengan adanya kelembapan yang tinggi, bercak pada daun akan menyebar/menular dengan cepat pada bagian daun sehat menyebabkan daun busuk dan akhirnya mengering (Wulandari et al. N. K., 2023).

Tabel 2. 2 Ciri-Ciri Penyakit Daun Tomat (Lanjutan)

No	Nama Penyakit	Sampel Gambar	Ciri-Ciri
2.	<i>Bacterial spot</i> (Infeksi bakteri daun)		Bercak kecil berwarna coklat tua hingga hitam dan bercak dikelilingi warna kuning dengan permukaan daun kasar dan kering, daun yang terinfeksi menguning dan rontok lebih banyak (Sahrani, 2021).
3.	<i>Early blight</i> (penyakit kudis daun)		Bercak-bercak berwarna kelabu dan akan berkembang menjadi bercak berbentuk bulat dengan diameter mencapai 1 cm dan terdapat bercak kecil berwarna coklat yang menyebabkan daun berlubang (Sahrani, 2021).
4.	Daun sehat		Memiliki daun berbentuk oval dan pada bagian tepi daun terdapat gerigi yang membentuk celah yang menyirip agak melengkung masuk. Daun berwarna hijau dan merupakan daun majemuk ganji I (Sahrani L. , 2021).

## 2.2 Computer Vision

*Computer vision* merupakan bidang interdisipliner yang memungkinkan sistem dan komputer memperoleh informasi berharga dari video, gambar digital, dan berbagai input visual lainnya (Putra et al., 2025). Kecerdasan buatan membantu komputer berpikir, sedangkan *computer vision* membantu komputer dalam melihat dan memahami lingkungan sekitarnya. Tujuan dari adanya *computer vision* adalah mengidentifikasi dan memproses data visual dalam jumlah yang besar dengan cepat dan akurat sehingga dapat membantu sebuah keputusan atau rekomendasi berdasarkan informasi tersebut (Arnita et al., 2022).

### 2.3 Pengolahan Citra Digital

Menurut Putra (2010) pengolahan citra digital merujuk pada pemrosesan gambar dua dimensi menggunakan komputer. Pengolahan citra mengacu pada pemrosesan setiap data dua dimensi. Dalam sebuah citra yang mengalami penurunan kualitas karena mengandung derau (*noise*), kabur (*blurring*), kurang tajam, kontras buruk, dan sebagainya, akan sulit diinterpretasikan baik oleh manusia maupun mesin. Sehingga pada citra tersebut perlu dilakukan sejumlah proses dan manipulasi agar menghasilkan citra dengan kualitas informasi yang lebih baik, sehingga mudah untuk diinterpretasi (Zonyfar, 2020).

Menurut Amran (2023) citra digital terbentuk dari serangkaian pixel (*pixel, picture* elemen, elemen gambar) dengan ukuran tertentu yang mempunyai nilai. Menurut (Putra, 2010) citra digital dapat ditulis dalam bentuk matriks sebagai berikut :

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & \dots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \dots\dots\dots(2.1)$$

Suatu citra digital mempunyai fungsi dua dimensi  $f(x,y)$  yang memiliki ukuran M baris dan N kolom di mana x dan y adalah koordinat pada bidang diantara dan  $f(x,y)$  adalah intensitas cahaya (*brightness*) atau amplitudo atau derajat keabuan (*grey level*). Menurut Al-Khowarizmi & Lubis (2023) citra digital merupakan suatu matriks di mana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya yang disebut sebagai elemen gambar atau piksel menyatakan nilai derajat keabuan pada titik tersebut. Citra digital berukuran N x M (tinggi = N, lebar = M) dinyatakan dengan matriks N x M.

### 2.4 Citra warna

Citra warna atau citra RGB merupakan citra yang dinyatakan setiap piksel dengan memuat tiga komponen warna yaitu, warna merah (R, *red*), hijau (G, *green*), dan biru (B,*blue*) (Wulandari et al. D. A., 2023). Setiap komponen menggunakan 8 bit yang nilainya berkisar 0 sampai 255. Berdasarkan komponen tersebut menghasilkan warna yang mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna.

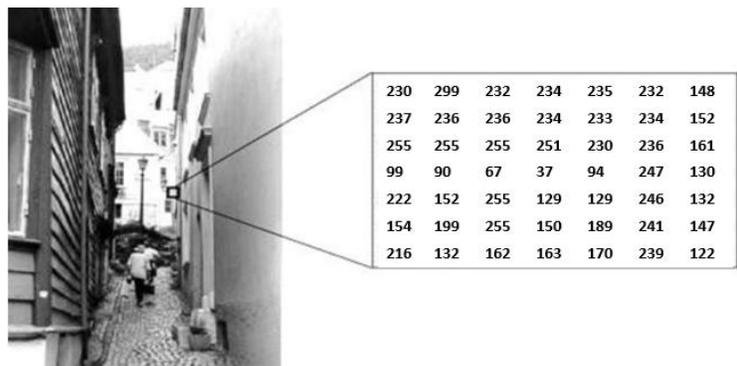


**Gambar 2. 1 Citra Warna**

## 2.5 Citra *Grayscale*

Citra *grayscale* berskala keabuan yaitu citra yang dinyatakan dengan gradasi warna hitam dan putih sehingga menghasilkan bayangan abu-abu (Wulandari et al. N. K., 2023). Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik *single band* (Hamzaidah, 2023). Citra *grayscale* disimpan dalam format 8 bit untuk setiap sampel piksel, yang memungkinkan sebanyak 256 intensitas. Untuk mengubah citra berwarna yang mempunyai nilai matriks masing-masing R, G, dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G, dan B sehingga dapat ditulis menjadi:

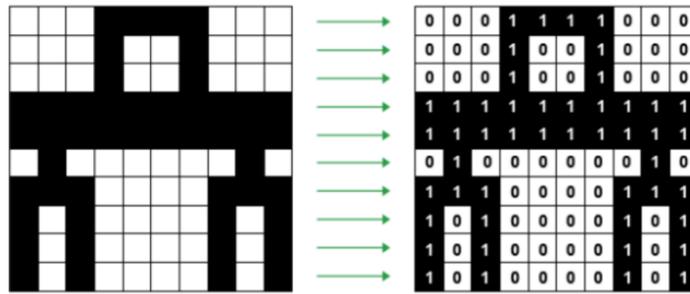
$$x = 0.299R + 0.587G + 0.114B \quad \dots\dots\dots(2.2)$$



**Gambar 2. 2 Citra *Grayscale***

## 2.6 Citra Biner

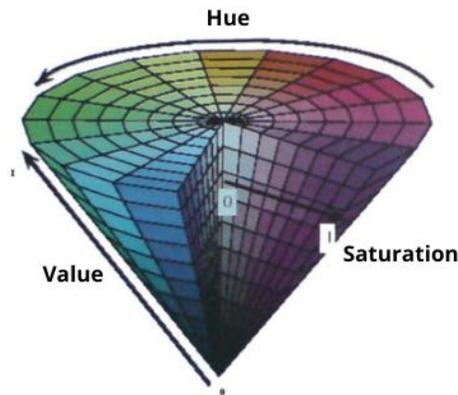
Citra biner berarti citra yang dinyatakan dengan biner. Di dalam biner hanya ada dua kemungkinan nilai yaitu nol dan satu. Nilai 0 digunakan untuk mewakili warna hitam dan nilai 1 digunakan untuk mewakili warna putih (Wulandari et al. N. K., 2023).



Gambar 2. 3 Citra Biner

## 2.7 Hue Saturation Value (HSV)

HSV atau akronim dari *Hue Saturation Value* merupakan sebuah ruang warna yang dianggap paling dekat dalam mendeskripsikan warna oleh mata manusia dari pada RGB (*red, green, blue*) (Zonyfar, 2020). Menurut Hawa et al., (2023) model warna HSV, *Hue* merupakan atribut dari sensasi visual di mana suatu area tampak mirip dengan salah satu warna yang dirasakan, merah, kuning, hijau dan biru atau kombinasi dari keduanya. *Saturation* mengacu pada seberapa murni atau intens rona (*hue*) yang diberikan. Sedangkan *Value* mengacu pada terang atau gelapnya warna.



Gambar 2. 4 Visualisasi Model HSV

Pada Gambar 2.4 menjelaskan bahwa model visualisasi lain dari model HSV adalah kerucut. *Hue* dilukiskan sebagai bentuk bundar dari roda warna (Putra, 2010). Saturasi (*Saturation*) diwakilkan dengan jarak dari pusat lingkaran melintasi bagian dari kerucut dan *alue* adalah jarak dari titik akhir kerucut. Menurut Widodo (2021) nilai *hue* antara 0 sampai 1, 0 berarti warna antara merah melewati kuning, hijau, cyan, biru dan magenta dan kembali menjadi merah. Nilai *saturation* antara 0 sampai 1, 0 berarti dari yang tidak tersaturasi (keabuan) sampai tersaturasi penuh (tidak putih). Nilai *value* atau *brightness* antara 0 sampai 1,0 berarti semakin cerah.

**2.7.1 Mencari Nilai Value**

$$r = \frac{R}{255} \dots\dots\dots(2.3)$$

$$g = \frac{G}{255} \dots\dots\dots(2.4)$$

$$B = \frac{B}{255} \dots\dots\dots(2.5)$$

$$V = \max(r, g, b) \dots\dots\dots(2.6)$$

Keterangan:  
 r = Nilai komponen merah ternormalisasi  
 g = Nilai komponen hijau ternormalisasi  
 b = Nilai komponen biru ternormalisasi  
 R = Intensitas warna merah dari sebuah piksel  
 G = Intensitas warna hijau dari sebuah piksel  
 B = Intensitas warna biru dari sebuah piksel  
 V = Nilai *value*

**2.7.2 Mencari Nilai Saturation**

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r, g, b)}{v}, & \text{jika } V > 0 \end{cases} \dots\dots\dots(2.7)$$

Keterangan:  
 H = Nilai *hue*  
 S = Nilai *saturation*  
 V = Nilai *value*  
 r = Nilai komponen merah ternormalisasi  
 g = Nilai komponen hijau ternormalisasi  
 b = Nilai komponen biru ternormalisasi

**2.7.3 Mencari Nilai Hue**

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ 60 * \left[ \frac{g - b}{V - \min(r, g, b)} \right] \text{ mod } 6, & \text{jika } V = r \\ 60 * \left[ 2 + \frac{b - r}{V - \min(r, g, b)} \right], & \text{jika } V = g \\ 60 * \left[ 4 + \frac{r - g}{V - \min(r, g, b)} \right], & \text{jika } V = b \end{cases} \dots\dots\dots(2.8)$$

Nilai hasil perhitungan *Hue*, *Saturation*, *Value* dilakukan penyederhanaan nilai dengan rentang 0-255 dengan rumus sebagai berikut.

$$H = \frac{H}{2} \dots\dots\dots(2.9)$$

$$S = S \times 255 \dots\dots\dots(2.10)$$

$$V = V \times 255 \dots\dots\dots(2.11)$$

**2.8 Gray Level Co-Occurrence Matrix (GLCM)**

Gray Level Co-occurrence Matrix didefinisikan sebagai suatu matrik frekuensi dari dua piksel yang dipisah oleh suatu *vector* yang ada dalam citra (Sinaga, 2019). Metode GLCM atau analisis pada tekstur merupakan suatu metode untuk melakukan klasifikasi pada suatu citra, GLCM pada dasarnya merupakan *matrix* persegi yang memiliki sifat sifat tertentu dengan distribusi spesial (Alwy et al., 2023).

**1. Kontras (Contrast)**

Representasi nilai terhadap variasi tingkat keabuan dalam *matrixs co-occurrence*.

$$Con = \sum_{i,j=0}^{level-1} P_{i,j} (i - j)^2 \dots\dots\dots(2.12)$$

Keterangan:

*Con* = Nilai *Contrast* dari elemen

*i* = Matriks baris *i*

*j* = Matriks baris *j*

*P<sub>i,j</sub>* = Nilai piksel pada koordinat *i,j* pada GLCM Matriks

*level* = Rentang *gray tone*

**2. Korelasi (Correlation)**

Nilai ukuran ketergantungan linier antara nilai atas keabuan dalam citra. Berikut persamaan untuk menghitung.

$$\mu = \sum_{i=0}^{level-1} \sum_{j=0}^{level-1} i \times P_{i,j} \sum_{i=0}^{255} \sum_{j=0}^{255} j \times P_{i,j} \dots\dots\dots(2.13)$$

$$\sigma = \sqrt{\sum_{i=0}^{level-1} \sum_{j=0}^{level-1} (i - \mu_i)^2 P_{i,j}} \dots\dots\dots(2.14)$$

$$= \sqrt{\sum_{i=0}^{level-1} \sum_{j=0}^{level-1} (j - \mu_j)^2 P_{i,j}}$$

$$Cor = \sum_{i=0}^{level-1} \sum_{j=0}^{level-1} P_{i,j} \frac{(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \dots\dots\dots(2.15)$$

Keterangan :

- $Cor$  = Nilai *Correlation* dari elemen
- $i$  = Matriks baris  $i$
- $j$  = Matriks baris  $j$
- $P_{i,j}$  = Nilai piksel pada koordinat  $i,j$  pada GLCM Matriks
- level = Rentang *gray tone*

### 3. Energi (*Energy*)

Untuk memeriksa tingkat keseragaman tekstur, semakin tinggi nilai energi. Semakin rendah variasi pada citra.

$$Eng = \sqrt{\sum_{i,j=0}^{level-1} P_{i,j}^2} \dots\dots\dots(2.16)$$

Keterangan:

- $Eng$  = Nilai *Energy*
- $i$  = Matriks baris  $i$
- $j$  = Matriks baris  $j$
- $P_{i,j}$  = Nilai piksel pada koordinat  $i,j$  pada GLCM Matriks
- level = Rentang *gray tone*

### 4. Homogenitas (*Homogeneity*)

Mengukur tingkat homogenitas atau kesamaan variasi dalam intensitas keabuan citra pada *matrix co-occurrence*.

$$Hom = \sum_{i=0}^{level-1} \sum_{j=0}^{level-1} \frac{P_{i,j}}{1 + |i - j|} \dots\dots\dots(2.17)$$

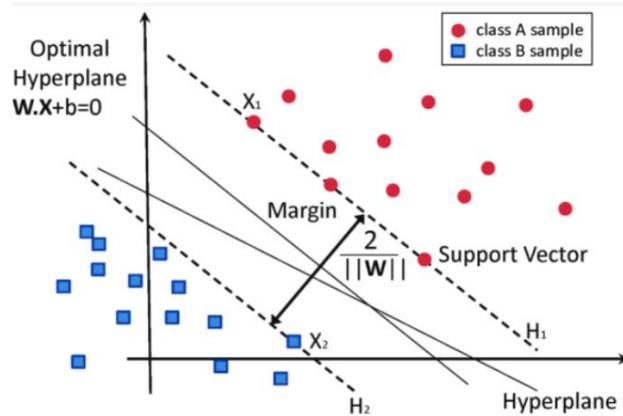
Keterangan:

- $Hom$  = Nilai *Homogeneity* dari elemen
- $i$  = Matriks baris  $i$
- $j$  = Matriks baris  $j$
- $P_{i,j}$  = Nilai piksel pada koordinat  $i,j$  pada GLCM Matriks
- level = Rentang *gray tone*

## 2.9 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) adalah algoritma pembelajaran mesin yang dapat digunakan untuk klasifikasi dan regresi sekaligus (Putra et al. R. F., 2024). Dalam klasifikasi daun *Support Vector Machine* memiliki algoritma pembelajaran mesin yang efektif untuk memisahkan data dengan menggunakan *hyperplane* (batas keputusan) yang optimal di antara

dua kelas yang berbeda (Mulyana & Muthmainnah, 2023). Algoritma *Support Vector Machine* memiliki kelebihan dalam menghasilkan model klasifikasi yang baik walaupun dilatih menggunakan himpunan data yang sedikit (Amelia et al., 2023).



**Gambar 2. 5 Margin Hyperplane Terbaik**

Proses pelatihan menggunakan metode SVM melibatkan beberapa langkah penting. Berikut langkah-langkah dalam melakukan *sequal training* akan ditunjukkan pada persamaan sebagai berikut:

1. Inisiasi parameter

Melakukan inisialisasi terhadap  $\alpha_i$  dan parameter lainnya, seperti nilai  $\varepsilon$ ,  $\gamma$ ,  $C$  dan integrasi maksimal

Keterangan:

- $\alpha_i$  = alfa ke-i, suntuk mencari *support vector*.
- $\gamma$  = konstanta gamma untuk mengontrol kecepatan.
- $C$  = Variabel *slack*.
- $\varepsilon$  = Epsilon, untuk menentukan nilai *error*.

2. Perhitungan kernel

a. Kernel Linier

$$K(x_i, x_j) = x_i^T \cdot x_j \dots\dots\dots(2.18)$$

Keterangan:

- $K(x_i, x_j)$  = Fungsi kernel
- $x_i$  = Nilai x padabaris ke-i dengan x *transpose*
- $x$  = Nilai x padabaris ke-j

b. Kernel Polynomial

$$K(x_i, x_j) = (y \cdot x_i^T + r)^d, y > 0 \dots\dots\dots(2.19)$$

Keterangan:

$K(x_i, x_j)$  = Fungsi kernel

$X$  = Nilai  $x$

$x_i^T$  = Nilai  $x$  pada baris ke- $i$  dengan  $x$  transpose

$y$  = *scaling* parameter dari jarak euclidan

$r$  = Nilai koefisien polynomial

$d$  = Nilai *degree*

c. Kernel *Radial Basic Function*

$$K(x_i, x_j) = e^{-y||x_i - x_j||^2}, y > 0 \dots\dots\dots(2.20)$$

Keterangan:

$K(x_i, x_j)$  = Fungsi kernel

$e$  = exponen

$y$  = Nilai gamma

$x_i$  = Nilai  $x$  pada baris ke- $i$

$x_j$  = Nilai  $x$  pada kolom ke- $j$

Nilai  $C$  adalah nilai cost.

d. Kernel Sigmoid

$$K(x_i, x_j) = \tanh(yx_i^T x_j + r) \dots\dots\dots(2.21)$$

Keterangan:

$K(x_i, x_j)$  = Fungsi kernel

$\tanh$  = Nilai tengah hiperbola

$y$  = Nilai gamma

$x_i^T$  = Nilai  $x$  pada baris ke- $i$  dengan  $x$  *transpose*

$x_j$  = Nilai  $x$  pada kolom ke- $j$

$r$  = Nilai konstanta

3. Perhitungan Matriks Hessian

$$D_{ij} = y_{ij}(K(x_i, x_j) + \lambda^2) \dots\dots\dots(2.22)$$

Keterangan:

$y_i$  = Kelas data ke- $i$

$y_j$  = Kelas data ke- $j$

$\lambda^2$  = Batas teoritis yang di turunkan

$K(x_i, x_j)$  = Fungsi kernel yang di gunakan

4. Perhitungan Nilai Error

$$E_i = \sum_{j=1}^i a_i D_{i,j} \dots\dots\dots(2.23)$$

Keterangan:

$E_i$  = Error rate

$a_i$  = Alfa ke-i

$D_{i,j}$  = Fungsi matriks hessian pada baris ke-i dan kolom ke-j

5. Perhitungan Nilai Delta

$$\delta a_i = \min\{\max\{[y(1 - E_i), \alpha], C - \alpha_i\} \dots\dots\dots(2.24)$$

Keterangan :

$\delta a_i$  = Delta alfa ke-i

$E_i$  = Error rate

$C$  = Konstanta C

$\alpha_i$  = Alfa ke-i

$y$  = Kontanta gamma

6. Perhitungan  $a_i$  baru

$$a_{in} = a_i + \delta a_i \dots\dots\dots(2.25)$$

Keterangan:

$a_{in}$  = Alfa baru ke-i

$a_i$  = Alfa ke-i

$\delta a_i$  = Delta alfa ke-i

7. Perhitungan Nilai Bobot (*weight*)

$$w = \sum_i^n a_i y_i x_i \dots\dots\dots(2.26)$$

Keterangan:

$w$  = Parameter bobot atau *weight vector*

$a_i$  = Alfa ke-i

$y_i$  = kelas data atau label

$x_i$  = Data latih

8. Perhitungan Nilai Bias

$$b = -\frac{1}{2}(w \cdot x^+) + w \cdot x^- \dots\dots\dots(2.27)$$

Keterangan:

$b$  = Nilai bias

$w$  = parameter bobot atau *weight vector*

- $x^+$  = Parameter fitur yang termasuk dalam kelas positif
- $x^-$  = Parameter fitur yang termasuk dalam kelas negatif

9. Penentuan Keputusan

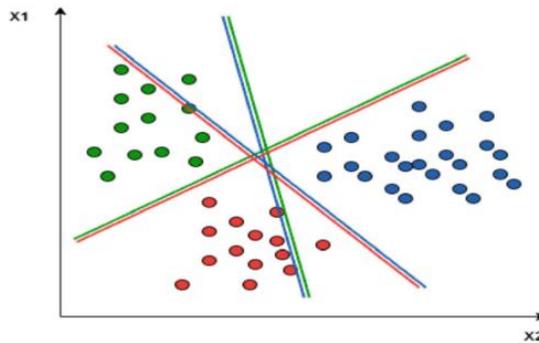
$$f(x) = \text{sign}(\sum_{i=1}^n a_i y_i K(x_i, x) + b) \dots\dots\dots(2.28)$$

Keterangan:

- $f(x)$  = Fungsi keputusan
- $a_i$  = Alfa ke-i
- $y_i$  = Kelas data atau label
- $K(x_i, x)$  = Fungsi kernel
- $b$  = Nilai bias

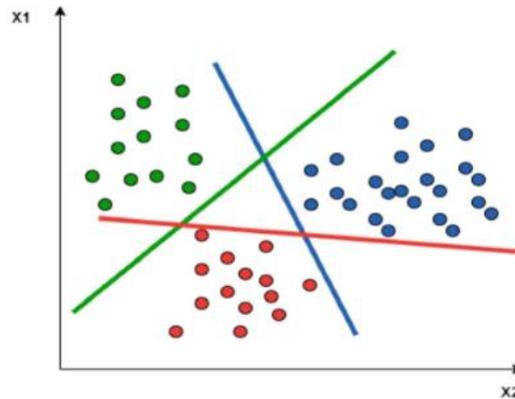
**2.10 Multiclass Support Vector Machine**

*Multiclass Support Machine* (M-SVM) adalah salah satu metode klasifikasi yang digunakan untuk menangani masalah lebih dari dua kelas. Metode ini dapat diperluas untuk mengatasi kasus *multiclass* melalui pendekatan *one-versus-One* (OVO) dan *one-versus-All* (OvA). *Multiclass SVM* berusaha untuk membangun *Hyperplane* yang memaksimalkan batas pemisah antara semua kelas yang terlibat (Sanjay Chakraborty, 2024).



**Gambar 2. 6 One-versus-One**

*One-versus-One* untuk setiap pasangan kelas, buat pengklasifikasi biner. Sekema pemungutan suara sering kali muncul digunakan untuk menentukan klasifikasi akhir. Untuk kelas  $k*(k-1)/2$  pada Gambar 2.6 satu satunya tujuan garis merah-biru adalah untuk memaksimalkan jarak antara titik biru dan merah. Titik hijau tidak terkait dengannya.



Gambar 2. 7 One-versus-All

Pada Gambar 2.7 menjelaskan *One-versus-All* dapat membangun pengklasifikasian biner yang membandingkan setiap kelas dengan setiap kelas lainnya. Ini menghasilkan k pengklasifikasi untuk k kelas, dengan menganggap satu kelas sebagai positif dan kelas lainnya sebagai negatif. Biasanya pengklasifikasi dengan skor keyakinan tertinggi menentukan kategori akhir pada gambar di atas hijau.

### 2.11 Confusion Matrix

Menurut Wulandari & Sifaunajah (2019) *confusion matrix* adalah metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining atau Sistem Pendukung Keputusan. Dalam tahap evaluasi kinerja algoritma pada klasifikasi didasarkan pada banyaknya (*count*) *test record* yang diprediksi secara benar dan secara tidak oleh model. *Count* ini ditabulasikan dalam sebuah tabel yang dikenal sebagai *confusion matrix*.

Tabel 2. 3 Confusion Matrix

		Kelas Hasil Prediksi	
		Kelas 1	Kelas 2
Kelas Sebenarnya	Kelas 1	$X_{11}$ True Positive (TP)	$X_{12}$ False Negative (FN)
	Kelas 2	$X_{21}$ False Positive (FP)	$X_{22}$ True Negative (TN)

Pada Tabel 2.3 menjelaskan bahwa, variabel  $X_{11}$  menerangkan jumlah TP, yaitu data pada kelas 1 yang terprediksi benar, variabel  $X_{12}$  menerangkan jumlah Fn, yaitu jumlah data kelas 1 yang diprediksi sebagai kelas 2. Variabel  $X_{21}$  menerangkan jumlah FP, yaitu jumlah data kelas 2 yang terprediksi sebagai kelas 1. Variabel  $X_{22}$  menerangkan jumlah TN, yaitu jumlah data kelas 2 yang terprediksi benar (Werdiningsih et al., 2022). Nilai-nilai tersebut akan dijadikan acuan untuk menghitung beberapa indikator sebagai hasil dari evaluasi sistem, diantaranya yaitu akurasi, sensitivitas, dan spesifisitas yang ditunjukkan pada persamaan berikut.

1. Akurasi

Akurasi adalah nilai untuk mengukur sejauh mana dapat mengklasifikasikan dengan benar. Perhitungan nilai akurasi didapat dari persamaan:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \dots\dots\dots(2.29)$$

2. Presisi

Presisi adalah nilai untuk mengukur sejauh mana model memberikan prediksi yang benar untuk kelas positif. Perhitungan nilai akurasi dapat dari persamaan:

$$Presisi\ positif = \frac{TP}{TP + FP} \dots\dots\dots(2.30)$$

$$Presisi\ negatif = \frac{TN}{TN + FN} \dots\dots\dots(2.31)$$

3. Recall

Recall adalah nilai untuk mengukur sejauh mana model dapat mendeteksi dengan benar kelas positif. Perhitungan nilai recall dapat dari persamaan:

$$Recall\ positif = \frac{TP}{TP + FN} \dots\dots\dots(2.32)$$

$$Recall\ negatif = \frac{TN}{TN + FP} \dots\dots\dots(2.33)$$

$$Recall = \frac{recall\ positif + recall\ negatif}{2} \times 100\% \dots\dots\dots(2.34)$$

4. F1-Score

F1-Score adalah nilai penggabungan antara nilai presisi dan nilai recall yang menunjukkan tingkat keseimbangan antara dua nilai tersebut. Perhitungan nilai F1-Score didapat dari persamaan:

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \dots\dots\dots(2.35)$$

### 2.12 Penelitian Terkait

Penelitian sebelumnya yang berkaitan dengan implementasi dari ekstrak fitur *Support Vector Machine* (SVM) dan ekstraksi tekstur *Grey Level Co-occurrence Matrix* (GLCM) serta klasifikasi menggunakan metode *Support Vector Machine* (SVM) yang akan dijadikan referensi dalam penelitian ini dapat dilihat pada Tabel 2.4 sampai Tabel 2.8.

Tabel 2. 4 *State Of Art*

No	Penulis	Judul	Metode	Hasil
1.	(Tampinongkol et a., 2023)	Identifikasi Penyakit Daun Tomat Menggunakan <i>Grey Level Co-occurrence Matrix</i> (GLCM) dan <i>Support Vector Machine</i> (SVM)	RGB, GLCM dan SVM	Total data test sebesar 400 data yang terbagi 2 data citra bercak dan citra hawar dipilih secara acak menggunakan bahasa pemrograman python. Berdasarkan dengan parameter tersebut diperoleh hasil model klasifikasi SVM dua jenis penyakit bercak dan hawar daun adalah sebesar 61.75% dengan kernel RBF (radial basis function) yang digunakan.
2.	(Astiningrum et al., 2020)	Identifikasi Penyakit Pada Daun Tomat Berdasarkan Fitur Warna dan Tekstur	RGB, GLCM dan KNN	Data set yang digunakan sebanyak 750 citra daun tomat. Hasil akurasi berdasarkan warna dan tekstur menggunakan akurasi <i>confusion matrix</i> dan uji coba nilai k=1 dengan tingkat akurasi sebesar 92,89%.
3.	(Kurniawan et al., 2023)	Pengaruh Arsitektur Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Tomat	CNN	Dataset yang digunakan adalah penyakit daun tomat sebanyak 4079 gambar, Setelah dilakukan pengujian pada masing-masing arsitektur CNN, arsitektur InceptionV3 yang memiliki akurasi sebesar 99,67%.
4.	(Ningsih et al., 2022)	Klasifikasi Penyakit Early Blight dan Late Blight Pada Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode CNN Berbasis Website	CNN, <i>Deploy Web</i> , dan <i>Machine Learning</i>	Dataset sebanyak 4.000 citra dan 2 jenis penyakit pada daun tomat. Hasil akurasi dan <i>recall</i> dari 50 sampel data uji pada tabel 4 adalah akurasi nya 80% .

Tabel 2. 5 *State Of Art* (Lanjutan)

No	Penulis	Judul	Metode	Hasil
5.	(Erwanto et al., 2024)	Pengembangan Sistem Deteksi Penyakit Tanaman Tomat Melalui Citra Daun dengan Metode <i>you Only Once</i> (YOLO) Berbasis Android	YOLO	Dataset yang digunakan sebanyak 14.799 label. Hasil evaluasi menunjukkan bahwa model YOLO versi 8 memiliki kinerja yang baik dengan sebesar 96.6% yang berarti kemampuan model dalam mengidentifikasi penyakit daun tomat dengan akurat.
6.	(Adawiyah et al., 2022)	Optimasi Deteksi Penyakit Kulit Menggunakan Metode <i>Support Vector Machine</i> (SVM) dan <i>Gray Level Co-occurrence Matrix</i> (GLCM)	SVM dan GLCM	Penelitian ini menggunakan 20 sampel data. Metode SVM dan metode ekstraksi GLCM dapat digunakan untuk mengidentifikasi penyakit kulit dengan memperoleh hasil terbaik dan memberikan nilai akurasi sebesar 90%.
7.	(Sooai et al., 2023)	Klasifikasi Citra Daun Anggur Menggunakan SVM Kernel Linier	SVM	Penelitian ini memiliki 4334 observasi untuk pelatihan. Terjadi proses peningkatan akurasi prediksi dari 97.6% menggunakan SVM kernel <i>cubic</i> sebelumnya, berhasil ditingkatkan menjadi 98.1% menggunakan SVM dengan kernel linier dan ekstraksi fitur berbantuan Squeezenet.

Tabel 2. 6 *State Of Art* (Lanjutan)

No	Penulis	Judul	Metode	Hasil
8.	(Suhendra et al., 2022)	Identifikasi dan Klasifikasi Penyakit Daun Jagung Menggunakan <i>Support Vector Machine</i>	SVM	Pengujian dilakukan terhadap 3600 data dengan kelas daun-normal dan daun-tidak-normal. Metode SVM berhasil mengklasifikasikan citra penyakit daun jagung dengan akurasi optimum sebesar 99,5% dengan fitur gabungan warna & tekstur
9.	(Puerwandono & Maulana, 2023)	Penerapan algoritma svm untuk klasifikasi citra daun sirih <i>application of the svm algorithm to image classification of betel leaf</i>	SVM dan RGB	Penelitian ini adalah data 2 jenis citra daun sirih dengan jumlah data latih (train) sebanyak 1400 kemudian data uji (test) sebanyak 600 citra daun sirih. <i>Support Vector Machine</i> (SVM) diperoleh nilai <i>accuracy</i> berdasarkan data <i>train</i> sebesar 91% dan data <i>test</i> sebesar 80 % nilai <i>accuracy</i>
10.	(Amelia et al., 2023)	Klasifikasi Citra Daun Mangga Menggunakan Algoritma <i>Support Vector Machine</i>	SVM	Penelitian ini menggunakan 34 data. Penggunaan metode SVM pada penelitian ini cukup handal, mengingat akurasi yang didapatkan juga cukup tinggi yaitu data pelatihan sebesar 85,29% dan untuk data uji sebesar 92,85%.

Tabel 2. 7 *State Of Art* (Lanjutan)

No	Penulis	Judul	Metode	Hasil
11.	(Nahak et al., 2024)	Klasifikasi Penyakit Pada Tanaman Apel Melalui Citra Daun Menggunakan Metode Multiclass <i>Support Vector Machine</i>	GLCM dan MSVM	Penelitian ini menggunakan data sebanyak 600 sampel citra daun apel yang terserang penyakit. Pada pengujian, M-SVM kernel = Polynomial, pada berbagai pecahan data ( <i>split ratio</i> ) seperti 90:10 mencapai akurasi sebesar 76%. Selanjutnya, hasil <i>score</i> akurasi yang terendah. Pada Pengujian, M-SVM kernel = Polynomial, <i>split ratio</i> 10:90 mencapai akurasi sebesar 72%.
12.	(Sriani et al., 2024)	Klasifikasi Kualitas Daun Tembakau Menggunakan GLCM ( <i>Grey Level Co-Occurrence Matrix</i> ) dan SVM ( <i>Support Vector Machine</i> )	GLCM dan SVM	Citra yang digunakan pada proses <i>training</i> sebanyak 60 citra dan proses pengujian citra sebanyak 30 sampel citra. Hasil akurasi untuk klasifikasi dengan 30 citra uji kelayakan mutu tembakau adalah sebesar 73.3 %.
13.	(Sari & Wihandika, 2019)	Klasifikasi Jenis Citra Makanan Tunggal Berdasarkan Fitur <i>Local Binary Patterns</i> dan <i>Hue Saturation Value</i> Menggunakan <i>Improved K-Nearest Neighbor</i>	HSV, LBP dan KNN	Penelitian ini terdapat 21 jenis citra makanan tunggal. Pada tiap jenis makanan terdapat sepuluh data latih dan lima data uji. Klasifikasi dengan fitur HSV saja menghasilkan nilai akurasi tertinggi sebesar 90,476%. Klasifikasi dengan fitur LBP saja menghasilkan nilai akurasi tertinggi sebesar 85,714%. Klasifikasi dengan kombinasi fitur HSV dan LBP menghasilkan nilai akurasi tertinggi sebesar 90,476%

Tabel 2. 8 State Of Art (Lanjutan)

No	Penulis	Judul	Metode	Hasil
14.	(Sanusi et al., 2019)	Pembuatan Aplikasi Klasifikasi Citra Daun Menggunakan Ruang Warna RGB dan HSV	RGB dan HSV	Penelitian ini menggunakan Citra daun yang digunakan dalam penelitian ini berjumlah 200 buah citra, yang didapat dari Flavia Dataset. Hasil klasifikasi citra daun memiliki rata-rata akurasi yang tinggi yaitu 90,08%.
15.	(Sutisna & Sumantri, 2022)	Klasifikasi Tingkat Kematangan Buah Nanas Berdasarkan Tekstur <i>Grey Level Co-Occurrence Matrix</i> Dengan Metode <i>Support Vector Machine</i>	GLCM dan SVM	Penelitian ini terdapat 969 dataset nanas dengan dua kelas label. Berdasarkan hasil penelitian, klasifikasi menggunakan metode <i>Support Vector Machine</i> (SVM) masih mendapatkan hasil akurasi yang kurang baik dalam melakukan klasifikasi citra tingkat kematangan buah Nanas, yaitu sebesar 86%.

Berdasarkan penelitian terdahulu, penelitian ini memiliki kesamaan dalam tema yang diambil yaitu klasifikasi. Pada penelitian ini, terdapat perbedaan dalam penggunaan metode dalam klasifikasi:

1. Pada penelitian (Sari & Wihandika, 2019) menggunakan ekstraksi fitur *Hue Saturation Value dan Local Binary Patterns*. Penelitian ini membandingkan kedua metode tersebut dan pada hasil akhir menunjukkan bahwa metode KNN dengan fitur *Hue Saturation Value* lebih unggul dan akurat dibandingkan dengan metode KNN dengan *Local Binary Patterns*.
2. Pada penelitian (Nahak et al., 2024) menggunakan ekstraksi fitur *Grey Level Co-Occurrence Matrix* (GLCM) dengan menggabungkan metode klasifikasi MSVM. Penelitian ini melakukan klasifikasi penyakit pada tanaman apel melalui citra daun. Ekstraksi fitur GLCM terbukti sangat membantu dalam memberikan input untuk proses klasifikasi dan ekstraksi nilai ciri tekstur dari citra daun apel. Hasil terbaik menggunakan ekstraksi fitur dengan kernel *polynomial* mencapai akurasi sebesar 76%.
3. Pada penelitian (Sooai et al., 2023) menggunakan metode klasifikasi *Support Vector Machine* (SVM) dengan penerapan ekstraksi fitur *Squeezenet*. Metode SVM pada penelitian ini digunakan untuk mengklasifikasi penyakit daun anggur. Hasil terbaik

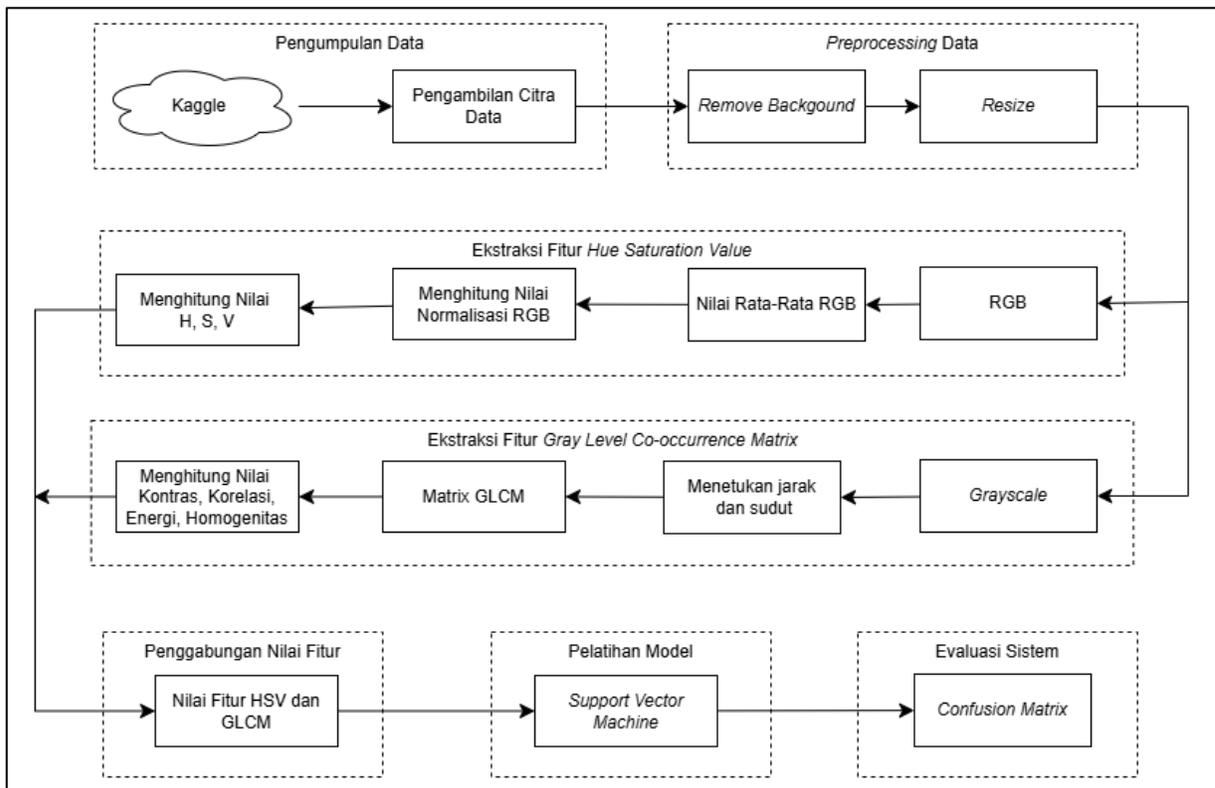
memiliki akurasi sebesar 98.1% dengan menggunakan metode SVM dengan kernel linier dan ekstraksi fitur squeezeNet. Dibandingkan dengan menggunakan metode SVM kernel *cubic* yang hanya mempunyai akurasi sebesar 97.6%.

Sehingga pada penelitian ini, peneliti menggabungkan algoritma klasifikasi *Support Vector Machine* (SVM), ekstraksi fitur *Grey Level Co-Occurrence Matrix* (GLCM) dan *Hue Saturation Value* (HSV) untuk membuat penelitian terbaru mengenai performa ketiga metode tersebut.

## BAB III METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Pada bagian ini akan membahas tentang metode penelitian yang akan digunakan pada penelitian ini menggunakan metode penelitian kuantitatif. Metode kuantitatif sendiri merupakan metode yang digunakan untuk menganalisis data *numerik*. Alur penelitian yang akan digunakan dapat dilihat pada Gambar 3.1.



**Gambar 3. 1 Alur Penelitian**

Penelitian ini menggunakan ekstraksi *Hue Saturation Value*, *Grey Level Co-occurrence Matrix* dan *Support Vector Machine*. Penelitian ini dimulai dari analisis masalah yaitu pengumpulan data terkait penyakit daun tomat yang telah dikelompokkan berdasarkan penyakit daun tomat. Tahapan selanjutnya dilakukan *pre-processing* data dengan melakukan *remove background* dan *resize* gambar supaya citra bisa fokus pada objek utama dan memiliki keseragaman ukuran. Kemudian dilakukan ekstraksi fitur warna *Hue Saturation Value* untuk mendapatkan nilai warna dari citra. Setelah mendapatkan nilai warna dari citra langkah selanjutnya adalah mengubah citra menjadi *grayscale* supaya citra bisa di proses ke tahap selanjutnya. Tahap selanjutnya yaitu ekstraksi tekstur, ekstraksi tekstur di sini menggunakan *Grey Level Co-occurrence Matrix* untuk mendapatkan nilai tekstur dari citra. Setelah

mendapatkan nilai hasil ekstraksi fitur HSV dan GLCM, kedua nilai ekstraksi fitur tersebut dilakukan penggabungan nilai fitur menggunakan *array*. Metode *Support Vector Machine* merupakan langkah selanjutnya untuk mengklasifikasi citra berdasarkan nilai-nilai ekstraksi fitur yang telah digabungkan dari proses sebelumnya. Hasil dari proses klasifikasi yang dilakukan menggunakan metode SVM selanjutnya akan dilakukan pengujian menggunakan *confusion matrix* untuk mengevaluasi kinerja dari model klasifikasi yang telah dilakukan saat mengetahui tingkat akurasi model dan keberhasilan model klasifikasi.

### 3.1.1 Analisis Masalah

Klasifikasi penyakit daun tomat memiliki beberapa tahapan. Tahapan pertama yang dilakukan agar citra siap di klasifikasi adalah *pre-processing* yang dilakukan adalah *resize* tujuannya untuk menyeragamkan seluruh citra karena mempengaruhi performa klasifikasi. Dalam klasifikasi citra ini, metode *Hue Saturation Value* digunakan untuk mengekstraksi fitur warna dari citra dan untuk ekstraksi fitur tekstur menggunakan *Grey Level Co-occurrence Matrix*. Selanjutnya metode *Support Vector Machine* digunakan untuk mengklasifikasi daun tomat berdasarkan 4 kelas penyakit daun tomat. Pada proses tersebut *data train* digunakan agar mesin bisa mempelajari untuk mengenali pola dan dataset akan menghasilkan klasifikasi berdasarkan pola yang telah dipelajari dari *data train*.

### 3.1.2 Pengumpulan Data

Pada penelitian ini proses pengambilan data menggunakan dataset sekunder yang bersumber dari *website* <https://www.kaggle.com/datasets/rahmi21/dataset-tomat>. Pada situs ini terdapat 1000 data citra penyakit daun tomat. Citra tersebut dibagi menjadi 4 kategori kualitas yaitu daun hawar 250 citra, infeksi bakteri 250 citra, kudis daun 250 citra dan sehat 250 citra. Pada penelitian ini semua data akan digunakan. Citra yang digunakan memiliki satu objek. Selanjutnya data tersebut dibagi menjadi perbandingan 90% untuk data training dan 10% untuk data testing yang mana pembagian data *training* dan *testing* dilakukan secara acak. Contoh pada data yang diambil dari situs tersebut dapat dilihat pada Tabel 3.1.

**Tabel 3. 1 Contoh Penyakit Daun Tomat**

No	Kualitas	Contoh
1	Penyakit daun hawar	

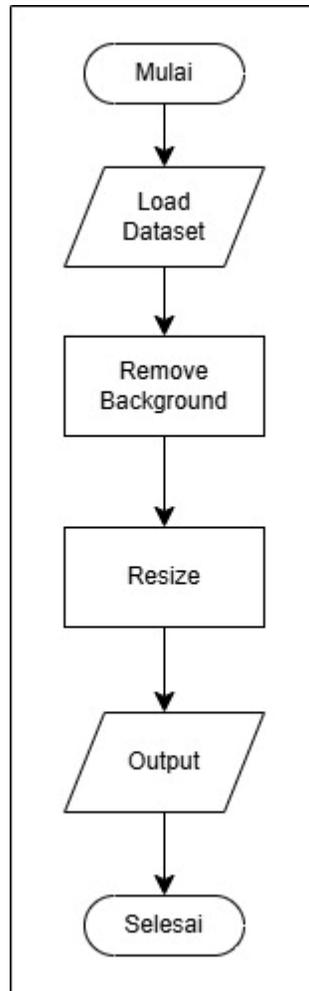
**Tabel 3. 2 Contoh Penyakit Daun Tomat (Lanjutan)**

No	Kualitas	Contoh
2	Penyakit kudis daun	
3	Infeksi bakteri daun	
4	Daun sehat	

### 3.1.3 *Pre-Processing Data*

Tahapan *pre-processing* adalah tahapan untuk menghilangkan atau mengubah informasi citra agar mendapat citra yang sesuai dan dapat di implementasikan ke dalam sistem. Langkah pertama yang dilakukan adalah memberi label pada setiap kelas data, yaitu label 1 untuk data citra penyakit daun tomat penyakit daun hawar, label 2 untuk data penyakit daun tomat penyakit kudis daun, label 3 penyakit daun tomat infeksi bakteri, Label 4 daun tomat sehat. Langkah selanjutnya yaitu menghapus *background* citra agar terfokus pada citra yang akan diolah. Setelah selesai langkah selanjutnya yaitu mengubah ukuran gambar agar data citra dapat disamakan ukurannya. Untuk ukuran yang dipakai tahap *resize* berukuran 320 x 320 piksel. ukuran ini di buat agar tidak terlalu jauh dengan ukuran gambar aslinya sehingga informasi dari gambar tetap terjaga. Hasil implementasi *pre-processing* tersebut yang nantinya akan digunakan

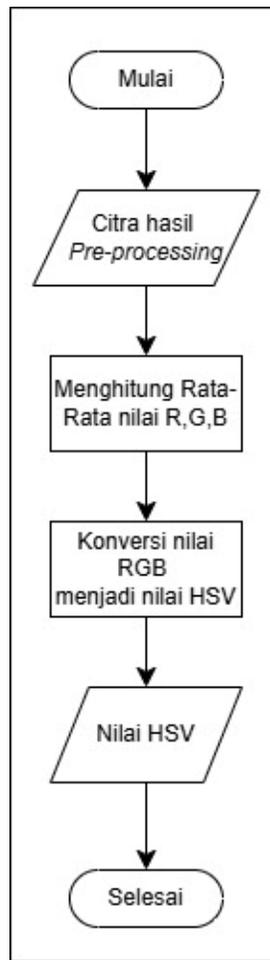
pada ekstraksi fitur *Hue Saturation Value* (HSV) dan *Grey Level Co-occurrence Matrix* (GLCM) Berikut *Flowchart Pre-processing* pada Gambar 3.2



**Gambar 3. 2** *Flowchart Pre-processing*

### **3.1.4 Ekstraksi Fitur *Hue Saturation Value***

*Hue Saturation Value* merupakan ekstraksi fitur warna yang berfungsi untuk konversi citra dari ruang warna RGB menjadi ruang warna HSV. Setelah terjadi konversi citra akan diperoleh rata-rata dari setiap elemen warna dalam citra yang dapat digunakan sebagai fitur analisis dalam proses klasifikasi. Berikut *flowchart* konversi RGB ke HSV pada Gambar 3.3.



**Gambar 3. 3 Flowchart Ekstraksi Fitur HSV**

Pada proses ini dilakukan proses input citra setelah itu akan mendapatkan nilai R, G, B dari citra. Berikut input citra dengan ukuran 3x5 piksel dapat dilihat pada Gambar 3.4.



(i,j)	20	21	22	23	24
20	[163,162,110]	[165,164,170]	[166,165,171]	[175,180,184]	[180,185,189]
21	[160,159,165]	[165,164,170]	[163,162,168]	[176,181,185]	[180,185,189]
22	[158,157,163]	[160,159,165]	[162,161,167]	[176,181,185]	[179, 184,188]

**Gambar 3. 4 RGB Citra Daun Tomat**

Berdasarkan matriks RGB pada Gambar 3.4 nilai RGB pada setiap piksel sebagai berikut:

**Tabel 3. 3 Matrix RGB**

R = 163 G =162 B =110	R = 165 G = 164 B = 170	R = 166 G = 165 B = 171	R = 176 G = 180 B =184	R = 180 G = 185 B =189
R = 160 G = 159 B =165	R =165 G = 164 B =170	R = 163 G = 162 B =168	R = 176 G = 181 B = 185	R = 180 G = 185 B = 189
R = 158 G = 157 B = 163	R = 160 G = 159 B = 165	R = 162 G = 161 B = 167	R = 176 G = 181 B = 185	R = 179 G = 184 B = 188

Pada tahap berikutnya setelah mendapat nilai RGB, nilai tersebut akan diproses untuk mendapatkan nilai HSV. Selanjutnya ditahap ekstraksi fitur HSV, citra dengan ruang warna RGB di rubah menjadi ruang warna HSV menggunakan persamaan persamaan 2.3 sampai persamaan 2.8. berikut merupakan perhitungan konversi ruang warna dari RGB menjadi ruang warna HSV berdasarkan Tabel 3.3 dengan hasil nilai rata-rata R = 168, dan rata-rata G = 169, dan rata rata B = 171.

$$r = \frac{168}{255} = 0.65, \quad g = \frac{169}{255} = 0.66, \quad b = \frac{171}{255} = 0.67$$

Perhitungan model HSV :

$$V = 0.67$$

$$S = 1 - \left( \frac{0.65}{0.67} \right) = 0.03$$

$$H = 60 \times \left( 4 + \left( \frac{0.65-0.66}{0.67-0.65} \right) \right) = 210$$

Setelah perhitungan nilai HSV, selanjutnya yaitu tahap penyederhanaan dengan persamaan 2.9 sampai persamaan 2.11 sehingga dapatkan matriks rumus penyederhaan dibawah ini:

$$H = \frac{210}{2} = 105$$

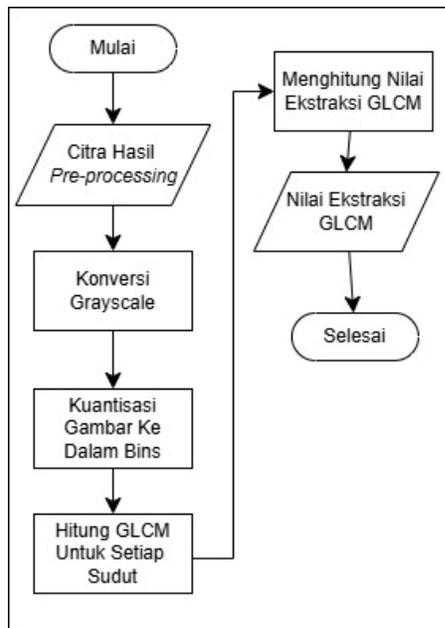
$$S = 0.03 \times 255 = 7$$

$$V = 0.67 \times 255 = 170$$

Berdasarkan matriks RGB pada Tabel 3.3 diperoleh nilai HSV =(105,7,170). Nilai H,S,V akan digunakan dalam proses klasifikasi.

### 3.1.5 Ekstraksi Fitur *Grey Level Co-occurrence Matrix*

Dari nilai RGB yang diperoleh sebuah citra akan digunakan dalam perhitungn GLCM untuk menentukan nilai parameter kontras, korelasi, energi, homogenitas. Langkah langkah dari proses ekstraksi ciri GLCM ditampilkan dalam flowchart sebagai berikut:



**Gambar 3. 5 Flowchart Ekstraksi Fitur GLCM**

Dari hasil perhitungan *grayscale* pada matriks 3x5 diatas maka diperoleh hasil citra *grayscale* yang dapat dilihat pada tabel berikut di mana nantinya data citra *grayscale* akan diproses dengan metode GLCM untuk mendapatkan informasi fitur teksturnya. Berikut merupakan matriks citra RGB ke *grayscale*:

**Tabel 3. 4 Matrix Grayscale**

157	164	165	179	183
159	164	162	179	183
157	159	161	179	182

Selanjutnya matriks *grayscale* dikuantisasi dengan menggunakan 16 nilai yang telah ditentukan, yaitu [0, 16,32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 240, 255], sehingga akan menghasilkan matriks yang menunjukkan nilai kuantisasi yang sesuai untuk setiap nilai piksel dalam citra. Sebagai contoh nilai piksel 12 dikuantisasi sehingga mendapatkan nilai hasil kuantisasi yaitu 0, nilai piksel 164 dan 165 dikuantisasi sehingga mendapatkan nilai hasil kuantisasi yaitu 10.

**Tabel 3. 5 Presentasi Matrix GLCM**

9	10	10	11	11
9	10	10	11	11
9	9	10	11	11

Selanjutnya melakukan perhitungan dengan jarak = 1 dan sudut =  $0^0$  terlebih dahulu lalu membuat *matrix* (I,j)

**Tabel 3. 6 Presentasi Perhitungan Jarak Sudut  $0^0$**

<b>I,j</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>0</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>2</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>3</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>4</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>6</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>7</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>8</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>9</b>	0	0	0	0	0	0	0	0	0	1	3	0
<b>10</b>	0	0	0	0	0	0	0	0	0	0	2	3
<b>11</b>	0	0	0	0	0	0	0	0	0	0	0	3

Selanjutnya melakukan penjumlahan *matrix* dengan nilai traspose *matrixny*:

**Tabel 3. 7 Presentasi Hasil Transpose Sistem Matrix GLCM**

<b>I,j</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>0</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>2</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>3</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>4</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>6</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>7</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>8</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>9</b>	0	0	0	0	0	0	0	0	0	2	3	0
<b>10</b>	0	0	0	0	0	0	0	0	0	3	4	3
<b>11</b>	0	0	0	0	0	0	0	0	0	0	3	6

Setelah mendapatkan nilai *matrix* simetris, setiap *matrix* dilanjutkan perhitungan nirmalisasi sehingga didapat *matrix* sebagai berikut:

Tabel 3. 8 Presentasi Normalisasi *Matrix* GLCM

I,j	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0.0833	0.125	0
10	0	0	0	0	0	0	0	0	0	0.125	0.1667	0.125
11	0	0	0	0	0	0	0	0	0	0	0.125	0.25

Setelah perhitungan normalisasi selesai lanjut ke langkah selanjutnya adalah perhitungan nilai ekstraksi GLCM:

a. Kontras (*Contrast*)

$$Con_{(9,9)} = 0.0833 \times (9 - 9)^2 = 0.0$$

$$Con_{(9,10)} = 0.125 \times (9 - 10)^2 = 0.125$$

$$Con_{(10,9)} = 0.125 (10 - 9)^2 = 0.125$$

$$Con_{(10,10)} = 0.1667(10 - 10)^2 = 0.0$$

$$Con_{(10,11)} = 0.125(10 - 11)^2 = 0.125$$

$$Con_{(11,10)} = 0.125(11 - 10)^2 = 0.125$$

$$Con_{(11,11)} = 0.25(11 - 11)^2 = 0.0$$

$$Con = 0.0 + 0.125 + 0.123 + 0.0 + 0.125 + 0.125 + 0.0 = 0.498$$

b. Korelasi (*Correlation*)

$$\mu_i = \frac{(9 \times 2) + (9 \times 3) + (10 \times 3) + (10 \times 4) + (10 \times 3) + (11 \times 3) + (11 \times 6)}{24}$$

$$\mu_i = 10.2$$

$$\mu_j = \frac{(9 \times 2) + (10 \times 3) + (9 \times 3) + (10 \times 4) + (11 \times 3) + (10 \times 3) + (11 \times 6)}{24}$$

$$\mu_j = 10.2$$

$$\sigma_{i(9,9)} = (9 - 10,2)^2 \times \frac{2}{24} = 0.12$$

$$\sigma_{i(9,10)} = (9 - 10,2)^2 \times \frac{3}{24} = 0.18$$

$$\sigma_{i_{10,9}} = (10 - 10,2)^2 \times \frac{3}{24} = 0.005$$

$$\sigma_{i_{10,10}} = (10 - 10.2)^2 \times \frac{4}{24} = 0.0067$$

$$\sigma_{i_{10,11}} = (10 - 10.2)^2 \times \frac{3}{24} = 0.005$$

$$\sigma_{i_{11,10}} = (11 - 10.2)^2 \times \frac{3}{24} = 0.08$$

$$\sigma_{i_{11,11}} = (11 - 10.2)^2 \times \frac{6}{24} = 0.16$$

$$\sigma_i^2 = 0.12 + 0.18 + 0.005 + 0.0067 + 0.005 + 0.08 + 0.16 = 0.5567$$

$$\sigma_i = \sqrt{0.5567}$$

$$\sigma_{j_{(9,9)}} = (9 - 10.2)^2 \times \frac{2}{24} = 0.12$$

$$\sigma_{j_{(9,10)}} = (10 - 10.2)^2 \times \frac{3}{24} = 0.005$$

$$\sigma_{j_{(10,9)}} = (9 - 10.2)^2 \times \frac{3}{24} = 0.18$$

$$\sigma_{j_{(10,10)}} = (10 - 10.2)^2 \times \frac{4}{24} = 0.0067$$

$$\sigma_{j_{(10,11)}} = (11 - 10.2)^2 \times \frac{3}{24} = 0.08$$

$$\sigma_{j_{(11,10)}} = (10 - 10.2)^2 \times \frac{3}{24} = 0.005$$

$$\sigma_{j_{(11,11)}} = (11 - 10.2)^2 \times \frac{6}{24} = 0.16$$

$$\sigma_j^2 = 0.12 + 0.005 + 0.18 + 0.0067 + 0.08 + 0.005 + 0.16 = 0.5567$$

$$\sigma_j = \sqrt{0.5567}$$

$$Cor_{(9,9)} = \frac{2}{24} \times \frac{|(9 - 10.2)(9 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.216$$

$$Cor_{(9,10)} = \frac{3}{24} \times \frac{|(9 - 10.2)(9 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.323$$

$$Cor_{(10,9)} = \frac{3}{24} \times \frac{|(10 - 10.2)(10 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.009$$

$$Cor_{(10,10)} = \frac{4}{24} \times \frac{|(10 - 10.2)(10 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.012$$

$$Cor_{(10,11)} = \frac{3}{24} \times \frac{|(10 - 10.2)(10 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.009$$

$$Cor_{(11,10)} = \frac{3}{24} \times \frac{|(11 - 10.2)(11 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.144$$

$$Cor_{(11,11)} = \frac{6}{24} \times \frac{|(11 - 10.2)(11 - 10.2)|}{\sqrt{0.5567} \times \sqrt{0.5567}} = 0.287$$

$$Cor = 0.216 + 0.323 + 0.009 + 0.012 + 0.009 + 0.144 + 0.287 = 1.0$$

c. Energi (*Energy*)

$$Eng = \sqrt{\left(\frac{2}{24}\right)^2 + \left(\frac{3}{24}\right)^2 + \left(\frac{3}{24}\right)^2 + \left(\frac{4}{24}\right)^2 + \left(\frac{3}{24}\right)^2 + \left(\frac{3}{24}\right)^2 + \left(\frac{6}{24}\right)^2} = 0.400$$

d. Homogenitas (*Homogeneity*)

$$Hom_{(9,9)} = \frac{\frac{2}{24}}{1 + |9 - 9|} = \frac{2}{24} = 0.0833$$

$$Hom_{(9,10)} = \frac{\frac{3}{24}}{1 + |9 - 10|} = \frac{3}{48} = 0.0625$$

$$Hom_{(10,9)} = \frac{\frac{3}{24}}{1 + |10 - 9|} = \frac{3}{48} = 0.0625$$

$$Hom_{(10,10)} = \frac{\frac{4}{24}}{1 + |10 - 10|} = \frac{4}{24} = 0.1667$$

$$Hom_{(10,11)} = \frac{\frac{3}{24}}{1 + |10 - 11|} = \frac{3}{48} = 0.0625$$

$$Hom_{(11,10)} = \frac{\frac{3}{24}}{1 + |11 - 10|} = \frac{3}{48} = 0.0625$$

$$Hom_{(11,11)} = \frac{\frac{6}{24}}{1 + |11 - 11|} = \frac{6}{24} = 0.25$$

$$Hom = 0.0833 + 0.0625 + 0.0625 + 0.1667 + 0.0625 + 0.0625 + 0.25 = 0.75$$

### 3.1.6 Penggabungan Nilai Fitur

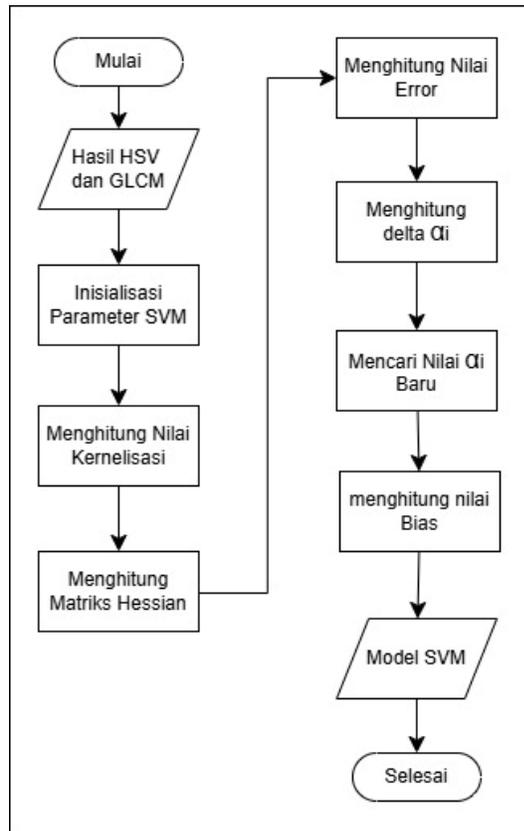
Langkah Selanjutnya adalah menggabungkan nilai hasil ekstraksi fitur. Nilai ekstraksi fitur yang akan digabungkan terdiri dari nilai *Hue*, *Saturation*, *Value*, dan nilai GLCM (kontras, Korelasi, Energi, Homogenitas). Data hasil penggabungan nilai ekstraksi fitur tersebut dapat dilihat pada Tabel 3.9.

**Tabel 3. 9 Nilai Hasil Ekstraksi Fitur**

HSV			GLCM			
<i>Hue</i>	<i>Saturation</i>	<i>Value</i>	Kontras	Korelasi	Energi	Homogenitas
105	7	170	0.498	1.0	0.400	0.75

### 3.1.7 Pelatihan Model

Setelah mendapatkan nilai-nilai fitur dari HSV dan GLCM dari proses sebelumnya, Langkah selanjutnya adalah membuat model klasifikasi dengan metode *Support Vector Machine* (SVM) untuk mendapatkan hasil akhir penyakit daun tomat. Alur perhitungan model klasifikasi dengan metode SVM dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Flowchart Klasifikasi SVM

#### 1. Data Hasil Ekstraksi Fitur

Sebelum melakukan perhitungan, dibutuhkan data hasil ekstraksi yang akan diolah untuk proses berikutnya. Dari data yang ditampilkan untuk kelas positif yaitu penyakit hawar daun tomat. Data hasil Ekstraksi fitur HSV dan GLCM dapat dilihat pada Tabel 3.10.

Tabel 3. 10 Hasil Ekstraksi Fitur

No	Label	H	S	V	Kontras	Korelasi	Energi	Homogenitas
1	1	105	7	170	0.498	1.0	0.400	0.75
2	-1	39	164	40	0.239	0.988	0.685	0.935
3	-1	42	122	38	0.347	0.978	0.628	0.893
4	-1	47	76	66	0.549	0.982	0.557	0.865

2. Inisialisasi Parameter

- a.  $\lambda = 0,5$
- b.  $\gamma = 0.001$
- c.  $C = 1$
- d.  $a = 0.001$

3. Menghitung nilai kernelisasi linier

Perhitungan nilai kernel menggunakan data yang sudah disiapkan sebelumnya sebagai berikut.

$$K(x_1, x_1) = [105,7,170,0.498,1.0,0.400,0.75]. [105,7,170,0.498,1.0,0.400,0.75]$$

$$K(x_1, x_1) = 39975.97$$

$$K(x_1, x_2)$$

$$= [105,7,170,0.498,1.0,0.400,0.75]. [39,164,40,0.239,0.988,0.685,0.935]$$

$$K(x_1, x_2) = 12045.08$$

$$K(x_1, x_3)$$

$$= [105,7,170,0.498,1.0,0.400,0.75]. [42,122,38,0.347,0.978,0.628,0.893]$$

$$K(x_1, x_3) = 11726.07$$

$$K(x_1, x_4) = [105,7,170,0.498,1.0,0.400,0.75]. [47,76,66,0.549,0.982,0.557,0.865]$$

$$K(x_1, x_4) = 16689.13$$

Setelah mendapatkan semua nilai  $K(x_i, x_j)$  dan disederhanakan maka diperoleh matriks D seperti pada Tabel 3.11.

**Tabel 3. 11 Hasil Perhitungan Kernelisasi Linier**

	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>
<b>D1</b>	3.99	1.20	1.17	1.66
<b>D2</b>	1.20	3.00	2.31	1.69
<b>D3</b>	1.17	2.31	1.80	1.37
<b>D4</b>	1.66	1.69	1.37	1.23

4. Menghitung matriks hessian

Berdasarkan persamaan 2.19 untuk mendapat nilai matriks hessian dengan nilai  $\lambda = 0.5$  didapatkan nilai sebagai berikut.

$$D_{1,1} = (1)(1)(3.99) + 0.5^2 = 4.24$$

Hasil perhitungan matriks hessian dapat dilihat pada Tabel 3.12

**Tabel 3. 12 Hasil perhitungan *Matrix Hessian***

	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>
<b>D1</b>	4.24	-0.95	-0.92	-1.41
<b>D2</b>	-0.95	3.25	2.56	1.94
<b>D3</b>	-0.92	2.56	2.05	1.62
<b>D4</b>	-1.41	1.94	1.62	1.48

5. Menghitung nilai *Error*

Berdasarkan persamaan 2.20 untuk mendapatkan nilai *Error* dengan nilai  $\alpha = 0.001$  didapatkan nilai sebagai berikut.

$$E_i = (0.001 \times 4.24) + (0.001 \times -0.95) + (0.001 \times -0.92) + (0.001 \times -1.41)$$

$$E_i = 0.00096$$

**Tabel 3. 13 Hasil perhitungan Nilai *Error***

<b>Data</b>	<b><math>E_i</math></b>
<b>D1</b>	0.00096
<b>D2</b>	0.00030
<b>D3</b>	0.00019
<b>D4</b>	0.00363

6. Menghitung delta  $a_i$

Berdasarkan persamaan 2.21 untuk mendapatkan nilai delta  $a_i$  dengan nilai  $\gamma = 0.001$  dan  $C = 1$  didapatkan nilai sebagai berikut.

$$\delta a_i = \min\{\max[0.001(1 - (0.00096)), 0], 1 - 0.001\} = 0.00099904$$

**Tabel 3. 14 Hasil Perhitungan Nilai  $\delta a_i$**

<b>Data</b>	<b><math>\delta a_i</math></b>
<b>D1</b>	0.00099904
<b>D2</b>	0.00099970
<b>D3</b>	0.00199981
<b>D4</b>	0.00099637

7. Menghitung nilai  $a_i$  baru

Berdasarkan persamaan 2.22 untuk mendapatkan nilai  $a_i$  baru dengan  $a_i$  lama = 0.001 mendapatkan nilai sebagai berikut.

$$a_{in} = 0.001 + 0.001005281 = 0.00199904$$

**Tabel 3. 15 Hasil Perhitungan  $\alpha$  Baru**

<b>Data</b>	<b><math>\alpha_{in}</math></b>
<b>D1</b>	0.00199904
<b>D2</b>	0.00199970
<b>D3</b>	0.00199520
<b>D4</b>	0.00199637

8. Menghitung nilai bias

Berdasarkan persamaan 2.24 untuk mendapatkan nilai bias yaitu sebagai berikut.

$$w \cdot x^+ = (0.002005281 \times 1 \times 2.07) + (0.001999421 \times -1 \times -1.96) \\ + (0.00199520 \times -1 \times -1.50) + (0.00199613 \times -1 \times -1.17)$$

$$w \cdot x^+ = 0.00192635$$

$$w \cdot x^- = (0.002005281 \times 1 \times -1.17) + (0.001999421 \times -1 \times 1.94) \\ + (0.00199520 \times -1 \times 1.62) + (0.00199613 \times -1 \times 1.48)$$

$$w \cdot x^- = -0.01738334$$

$$b = -\frac{1}{2}((0.00192635) + (-0.01738334)) = 0.00772850$$

9. Tes model SVM

Setelah memperoleh nilai bias selanjutnya adalah melakukan testing model SVM dengan data baru. Untuk data baru yang akan dipakai sebagai berikut.

$$x = [46, 146, 52, 0.229, 0.979, 0.453, 0.919]$$

Setelah melakukan proses perhitungan kernelisasi dan menggunakan persamaan 2.25 didapatkan hasil sebagai berikut.

$$f(x) = \text{sign}((0.00199904 \times 1 \times 1.46) + (0.00199970 \times -1 \times 2.78) \\ + (0.00199520 \times -1 \times 2.17) + (0.00199637 \times -1 \times 1.66) \\ + 0.00772850)$$

$$f(x) = \text{sign}(0.0238) = 1$$

Proses dari testing model akan mencari probabilitas pada semua kelas, berdasarkan  $f(x)$  yang dihasilkan adalah lebih dari 0 maka data testing tersebut berada di kelas positif atau bisa dikatakan termasuk dalam kelas hawar daun.

### 3.1.8 Evaluasi Sistem

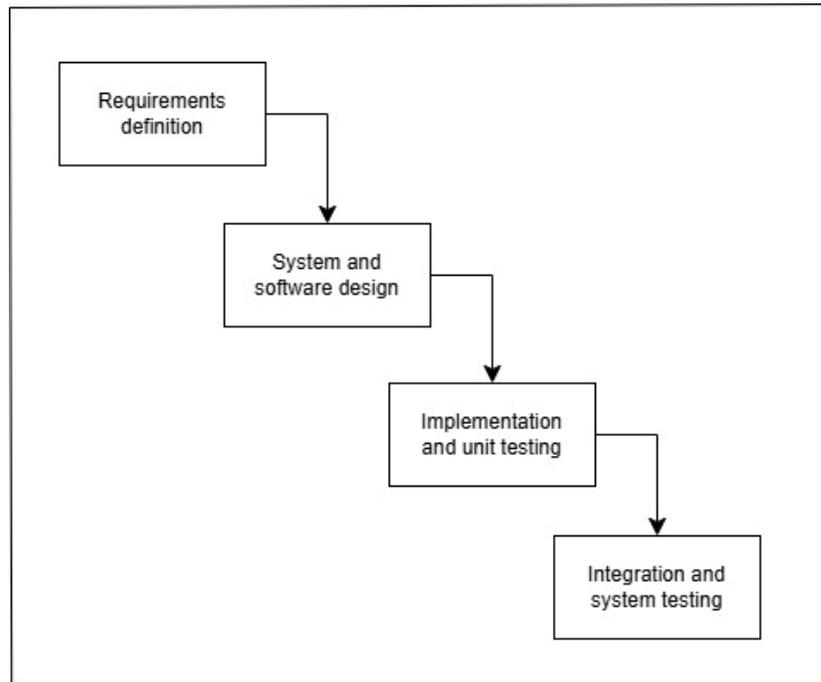
Pada tahap evaluasi sistem sistem akan di uji untuk mengevaluasi kemampuannya dalam mengklasifikasi penyakit daun tomat menggunakan metode *Confusion matrix* untuk menilai performa sistem. Nama nama lain ini akan digunakan untuk menghitung tingkat akurasi, presisi, dan *recall* dari sistem. Implementasi rancangan pengujian sistem menggunakan *confusion matrix* dapat dilihat sebagai Tabel 3.16.

Tabel 3. 16 Evaluasi Sistem

Data Aktual	Data Prediksi			
	Hawar daun	Infeksi bakteri	Kudis daun	Daun sehat
Hawar daun				
Infeksi bakteri				
Kudis daun				
Daun sehat				

### 3.2 Metodologi Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini menggunakan metode *Waterfall*. Model metode *waterfall* memiliki tahapan tahapan yang harus dilakukan bisa dilihat pada Gambar 3.7.



Gambar 3. 7 *Waterfall*

Pada Gambar 3.7 terdapat beberapa proses dari model *waterfall* dalam proses pengembangnya. Proses pengembangan dari model *waterfall* sebagai berikut.

#### 3.2.1 Analisis Kebutuhan

Analisis kebutuhan yang akan digunakan pada penelitian ini terdiri dari kebutuhan fungsional dan non fungsional. Kebutuhan fungsional dan non fungsional yaitu sebagai berikut:

1. Kebutuhan Fungsional

Kebutuhan fungsional yaitu kebutuhan proses yang mampu di kerjakan oleh sistem yang akan di kembangkan. Adapun kebutuhan fungsional pada sistem yang akan dikembangkan dalam penelitian ini sebagai berikut :

- a. Sistem dapat menerima data input citra yang di butuhkan
- b. Sistem menghasilkan output yang dibutuhkan
- c. Sistem dapat melakukan proses ekstraksi fitur *Hue Saturation Value, Grey Level Co-occurrence Matrix* dan *Support Vector Machine*
- d. Sistem dapat mengolah sumber data yang dibutuhkan

2. Kebutuhan Non Fungsional

Kebutuhan non- fungsional merupakan persyaratan untuk spesifikasi minimum yang diperlukan dalam perangkat keras, perangkat lunak, dan penggunaan untuk dapat mengimplementasikan sistem yang akan dikembangkan. Analisis kebutuhan dalam perancangan sistem pada penelitian sebagai berikut.

a. Analisis perangkat keras (*Hardware*)

Analisis perangkat keras dilakukan untuk merancang program dalam penelitian ini. Spesifikasi perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.17.

**Tabel 3. 17 Spesifikasi Perangkat Keras**

No	Perangkat Keras	Keterangan
1	Processor	10 <sup>th</sup> Gen Intel(R) Chre(TM) i5 Core 1.19GHz
2	RAM	20 GB DDR4
3	SSD	1 TB
4	Graphic	Nvidia Geforce MX330
5	Perangkat Input Output	Chatger/Adaptor
6	Koneksi	WiFi

b. Analisis perangkat lunak (*Software*)

Analisis perangkat lunak dilakukan untuk merancang model sistem yang digunakan dalam penelitian ini. Spesifikasi perangkat lunak yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.18.

**Tabel 3. 18 Spesifikasi Perangkat Lunak**

No	Perangkat Lunak	Keterangan
1	Windows 10	Sistem Operasi
2	Python 3.10	Bahas Pemrograman
3	Kaggle Editor	Web Open Source Editor
4	Draw.io	Desain Sistem dan Flowchart
5	Google Chrome	Web Browser

c. Analisis penggunaan (*User*)

Analisis pengguna dilakukan untuk penggunaan sistem *image processing* dalam mengklasifikasi penyakit daun Tomat. Spesifikasi penggunaan yang dibutuhkan pada penelitian ini dapat dilihat pada Tabel 3.19.

**Tabel 3. 19 Spesifikasi Pengguna**

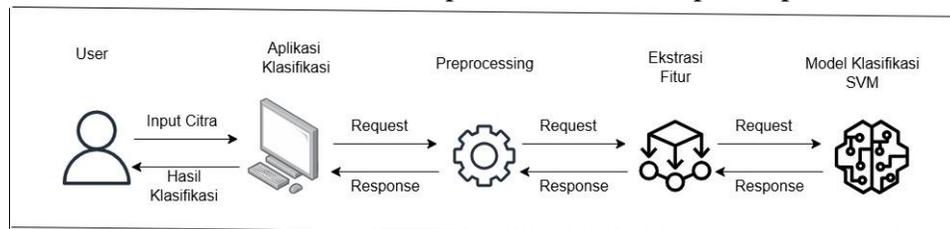
No	Keterangan
1	Memahami cara menggunakan komputer
2	Memahami cara menggunakan web browser
3	Memahami cara pengoperasian sistem <i>image processing</i> dalam mengklasifikasi penyakit daun tomat

### 3.2.2 Desain Sistem

Desain sistem yaitu merupakan spesifikasi dari tahap analisis kebutuhan yang diubah menjadi perancangan desain untuk implementasi sebagai program pada tahap berikutnya. Desain sistem ini berisikan perancangan arsitektur sistem, perancangan *flowchart* sistem dan perancangan antarmuka sistem.

a. Perancangan Arsitektur Sistem

Perancangan arsitektur sistem menggambarkan tahapan jalannya sistem secara keseluruhan, yang mencakup proses input data citra, pemrosesan dalam sistem dan output data yang dihasilkan oleh sistem. Arsitektur sistem dalam penelitian ini ditampilkan pada Gambar 3.8.



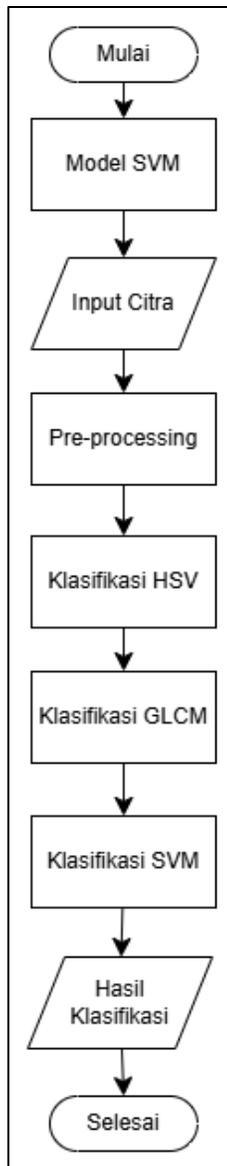
**Gambar 3. 8 Arsitektur Sistem**

Pada Gambar 3.8, *user* melakukan input citra daun tomat pada aplikasi klasifikasi. Selanjutnya sistem akan melakukan proses *Pre-processing* dari citra tersebut. Selanjutnya, citra

akan melakukan ekstraksi fitur warna dan ekstraksi fitur tekstur. Setelah semuanya selesai langkah terakhir adalah proses klasifikasi dengan model SVM. Hasil klasifikasi akan ditampilkan di aplikasi *user*.

b. Perancangan *Flowchart Sistem*

Perancangan *Flowchart* sistem yaitu gambaran bagaimana alur sistem yang di bangun pada penelitian ini. *Flowchart* sistem dalam penelitian ini ditampilkan pada Gambar 3.9.

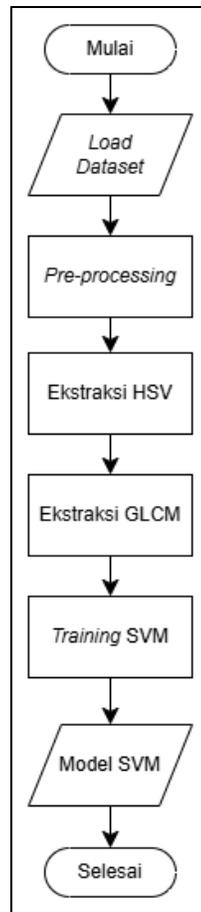


**Gambar 3. 9 Flowchart Sistem**

Gambar 3.9 menggambarkan proses-proses penelitian. Langkah awal adalah proses input data citra daun tomat yang akan digunakan untuk proses *pre-processing* data. Selanjutnya

melakukan perancangan arsitektur model SVM untuk memungkinkan sistem membagi setiap kelas yang tersedia dan mencapai akurasi tertinggi dalam penelitian ini. Setelah model berhasil di buat, model tersebut disimpan dan digunakan dalam proses klasifikasi.

Pada model SVM dirancang dengan tujuan untuk mendapatkan model dengan akurasi tertinggi yang nantinya akan digunakan dalam proses klasifikasi. Berikut Gambar alur kerja model SVM pada Gambar 3.10.

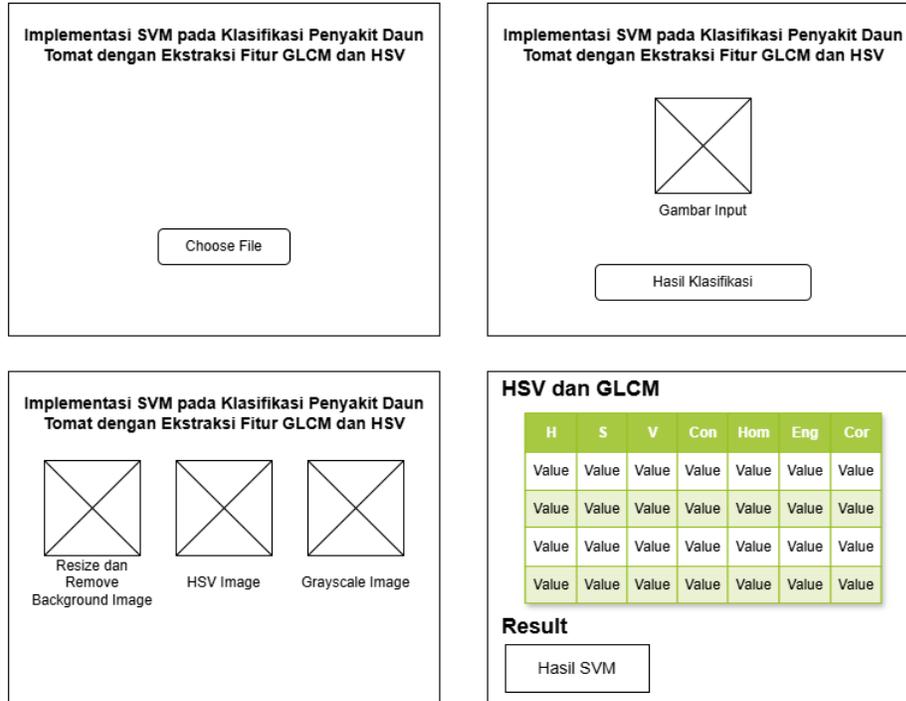


**Gambar 3. 10 Flowchart Model**

Gambar 3.10 langkah pertama dimulai dengan load dataset fungsinya untuk mengambil data dari dataset. Selanjutnya data akan dilakukan proses *pre-processing*. kemudian dilakukan ekstraksi HSV untuk mengambil nilai fitur warna dan ekstraksi GLCM untuk mengambil nilai fitur ekstraksi tekstur. lalu dataset tersebut digunakan sebagai data *input* dalam proses *training* sehingga didapatkan hasil berupa model SVM yang akan digunakan.

c. Perancangan Antarmuka

Perancangan *user interface* Menggambarkan desain antarmuka dari sistem yang akan di buat pada penelitian. Perancangan antarmuka dalam penelitian akan ditampilkan pada Gambar 3.11.



**Gambar 3. 11 Rancangan *User Interface***

Gambar 3.11 adalah rancangan *user interface* yang akan digunakan pada penelitian ini. Pada bagian kiri atas terdapat tombol untuk input file citra yang akan di klasifikasi. Setelah itu file input akan ditampilkan dan terdapat tombol untuk melakukan klasifikasi seperti pada gambar kanan atas. Apabila tombol klasifikasi di tekan maka akan muncul 3 gambar yang akan dihasilkan yaitu hasil *resize dan remove background*, *HSV image* dan *Grayscale image*. Selain adanya citra hasil *pre-processing*, terdapat nilai-nilai hasil ekstraksi fitur HSV dan GLCM yang ditampilkan dalam tabel serta hasil kelas dari data *input* seperti pada gambar di kanan bawah.

### 3.2.3 Implementasi Sistem

Implementasi sistem yaitu tahapan pengembangan perangkat lunak dengan mengimplementasikan seluruh desain sistem yang telah dipersiapkan sebelumnya sehingga menjadi sistem yang nyata dan berfungsi. Proses ini terdiri dari pengkodean berdasarkan perancangan arsitektur sistem, perancangan *flowchart* sistem dan perancangan antarmuka sistem yang sudah dipersiapkan sebelumnya.

### 3.2.4 Pengujian Sistem

Pengujian sistem merupakan proses evaluasi yang dilakukan untuk memastikan bahwa suatu sistem dapat berfungsi sesuai spesifikasi yang akan diterapkan di penelitian ini. Pengujian sistem yang di pakai pada penelitian adalah metode *black-box*, yang fokus pada pengujian fungsionalitas sistem. Dengan metode ini akan dilakukan serangkaian uji coba berdasarkan *input* dan *output* yang di harapkan, Sehingga dapat mengidentifikasi setiap kesalahan atau kekurangan dalam sistem.

**Tabel 3. 20 Rancangan Pengujian Black Box Testing**

<b>Skenario ke</b>	<b>Test Case</b>	<b>Hasil</b>
1	<i>User</i> dapat menginput file untuk dilakukan proses klasifikasi citra	
2	Sistem dapat menampilkan hasil <i>Resize</i> dan <i>Remove Background</i> , <i>HSV</i> , dan <i>Grayscale</i> citra	
3	Sistem dapat menampilkan hasil dari parameter <i>HSV</i> dan <i>GLCM</i>	
4	Sistem dapat menampilkan hasil klasifikasi metode <i>SVM</i>	

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Hasil

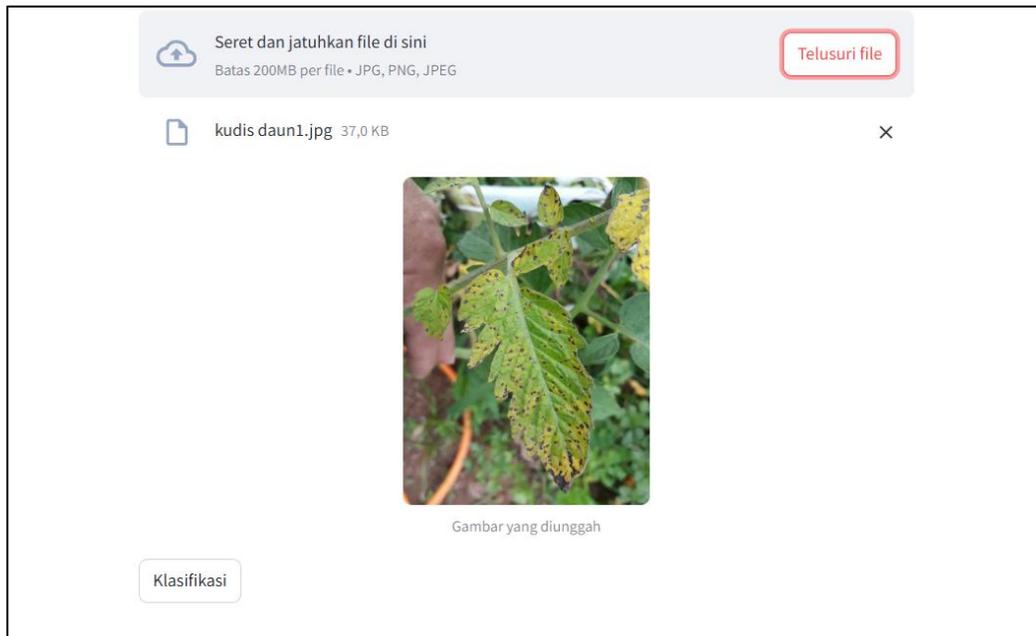
Bagian hasil berisi mengenai penjelasan tahapan implementasi dan hasil dari penelitian mengenai klasifikasi penyakit daun tomat berdasarkan citra penyakit daun tomat yang telah dikumpulkan sebelumnya. Tahapan ini mencakup implementasi dari perancangan *user interface*, *pre-processing*, ekstraksi fitur dengan *Hue Saturation Value*, ekstraksi fitur dengan *Grey Level Co-occurrence Matrix* dan klasifikasi penyakit daun tomat menggunakan *Support Vector Machine*.

#### 4.1.1 Hasil Implementasi

Pada hasil implementasi berisi peancangan *user interface* yang telah di buat sebelumnya. Pada sistem ini digunakan untuk menampilkan hasil dari klasifikasi penyakit daun tomat. Berikut merupakan hasil dari implematasi *user interface* klasifikasi tersebut.



Gambar 4. 1 Tampilan Sistem klasifikasi

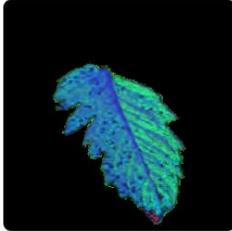


**Gambar 4. 2 Tampilan Sistem Klasifikasi (Lanjutan)**

**Klasifikasi**



Gambar Setelah Di Remove Background dan Resized



Gambar HSV



Gambar Grayscale

	Degree	Contrast	Homogeneity	Energy	Correlation
0	0.0000	0.3359	0.9338	0.7814	0.9858
1	45.0000	0.5891	0.9203	0.7795	0.9751
2	90.0000	0.3972	0.9272	0.7814	0.9832
3	135.0000	0.6066	0.9126	0.7795	0.9743

	Hue	Saturation	Value
0	37	127	37

**Daun tersebut termasuk dalam kategori : Kudis Daun**

**Gambar 4. 3 Tampilan Sistem Klasifikasi (Lanjutan)**

Tampilan sistem untuk klasifikasi penyakit daun tomat terdapat tampilan teks judul, berberapa tabel dan setiap tombol masing-masing mempunyai fungsi, pada Gambar 4.1 terdapat tombol telusuri file yang berfungsi untuk mengambil citra inputan daun tomat dari perangkat

lokal yang akan di klasifikasi. Setelah citra dipilih, citra akan di tampilkan beserta dengan nama berkas dari citra tersebut seperti pada Gambar 4.2. Tombol klasifikasi memiliki fungsi untuk memproses citra tersebut untuk mengetahui nilai fitur dan penyakit dari daun tomat tersebut. Pada Gambar 4.3 menunjukkan proses yang akan menghasilkan citra hasil *remove background*, *resize*, *citra HSV*, *grayscale*. Di bawah dari citra-citra tersebut terdapat nilai hasil ekstraksi fitur GLCM dan HSV. Di bawah nilai ekstraksi terdapat hasil dari klasifikasi menggunakan SVM berdasarkan nilai ekstraksi fitur yang didapatkan.

#### 4.1.2 Pre-Processing

*Pre-processing* yaitu tahapan yang dilakukan untuk mempersiapkan data sebelum diproses ke tahap ekstraksi fitur. Proses yang akan di kerjakan pada tahap *pre-processing* adalah *remove background* (menghapus latar belakang), *resize* (mengubah ukuran gambar) dan *grayscale*.

##### a. Remove Background

*Remove background* atau menghapus latar belakang menggunakan *library rembg*. Tahapan ini berfungsi untuk menghilangkan latar belakang citra dan hanya menyisakan objek daun tomat.

---

#### Modul Program 4. 1 Fungsi *Pre-processing Remove Background*

---

```
def remove_background_rembg(image):
    result = rembg.remove(image)
    return result
```

---

##### b. Resize

*Resize* atau mengubah ukuran gambar menggunakan fungsi *resize* yang ada pada *library cv2* dengan ukuran 320 x 320 piksel. Tahapan ini berfungsi supaya citra daun tomat mempunyai ukuran yang sama dan mengurangi kesalahan pada proses ekstraksi fitur.

---

#### Modul Program 4. 2 Fungsi *Pre-processing Resize*

---

```
def resize(image):
    result = cv2.resize(image, (320, 320),
        interpolation=cv2.INTER_LINEAR)
    return result
```

---

Berikut adalah contoh citra sebelum *pre-processing* dan setelah *preprocessing* terdapat pada Tabel 4.1

**Tabel 4. 1 Citra Sebelum dan Sesudah *Pre-processing***

<b>Citra Asli</b>	<b>Citra Setelah <i>Remove Background</i> dan <i>Resize</i></b>
	

### **4.1.3 Ekstraksi Fitur *Hue Saturation Value***

Tahapan setelah *pre-processing* adalah ekstraksi fitur HSV. Citra hasil *pre-processing* di konversi ke ruang warna HSV berdasarkan rata-rata nilai RGB dan selanjutnya dilakukan pengambilan fitur dari *channel* warna *Hue Saturation Value*.

---

#### **Modul Program 4. 3 Ekstraksi Fitur HSV**

---

```
def ekstraksi_fitur_hsv(image):  
  
    readImage=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)  
  
    r, g, b = np.mean(readImage, axis=(0, 1)) / 255.0  
    cmax = max(r, g, b)  
    cmin = min(r, g, b)  
    v = cmax  
  
    if cmax == 0:  
        s = 0  
    else:  
        s = 1 - (cmin/cmax)  
    if cmax == cmin:  
        h = 0  
    elif cmax == r:  
        h = 60 * ((g - b) / (cmax - cmin) % 6)  
    elif cmax == g:  
        h = 60 * (2 + (b - r) / (cmax - cmin))  
    else:  
        h = 60 * (4 + (r - g) / (cmax - cmin))  
    if h < 0:  
        h += 360  
    h = int(h / 2)  
    s = int(s * 255)  
    v = int(v * 255)  
    return (h, s, v)
```

---

Rentang nilai kelas daun hawar yaitu *Hue* dengan nilai antara 18 sampai 81, *Saturation* dengan nilai antara 30 sampai 179, dan *Value* dengan nilai antara 9 sampai 90. Sedangkan untuk kelas infeksi bakteri yaitu *Hue* dengan nilai antara 26 sampai 76, *Saturation* dengan nilai antara 54 sampai 219, dan *Value* dengan nilai antara 23 sampai 83. Selanjutnya untuk kelas kudis daun yaitu *Hue* dengan nilai antara 27 sampai 82, *Saturation* dengan nilai antara 41 sampai 146, dan *Value* dengan nilai antara 10 sampai 63. Sedangkan untuk kelas daun sehat yaitu *Hue* dengan nilai antara 34 sampai 60, *Saturation* dengan nilai antara 39 sampai 133, dan *Value* dengan nilai antara 23 sampai 72. Nilai warna *Hue*, *Saturation*, *Value* selanjutnya disimpan menjadi model untuk menjadi fitur warna seperti pada Tabel 4.2 berikut.

**Tabel 4. 2 Contoh Nilai Ekstraksi HSV**

No	H	S	V	Label
1	48	82	48	Hawar Daun
2	37	106	37	Hawar Daun
....	....	....	....	....
226	59	67	59	Infeksi Bakteri
227	47	118	47	Infeksi Bakteri
....	....	....	....	....
451	42	122	42	Kudis Daun
452	72	121	72	Kudis Daun
....	....	....	....	....
676	47	76	47	Daun Sehat
677	56	67	56	Daun Sehat

#### 4.1.4 Ekstraksi Fitur *Gray Level Co-occurrence Matrix*

Tahap selanjutnya yaitu ekstraksi fitur menggunakan metode GLCM. Dengan memanfaatkan berbagai parameter seperti *Energy*, *Contras*, *Homogenity*, *Correlation*. Parameter ini dihitung pada empat sudut yakni sudut  $0^{\circ}$ ,  $45^{\circ}$ ,  $90^{\circ}$ ,  $135^{\circ}$ . Proses pemrograman dilakukan dengan menggunakan *library Phyton* yaitu `from skimage.feature import graycomatrix, graycoprops`.

---

#### Modul Program 4. 4 Ekstraksi Fitur GLCM

---

```
def ekstraksi_fitur_glcm(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = img_as_ubyte(gray)
    bins = np.array([0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176,
192, 208, 224, 240, 255])
    inds = np.digitize(image, bins) - 1
    max value = inds.max() + 1
```

---

#### Modul Program 4. 5 Ekstraksi Fitur GLCM (Lanjutan)

```
glcm_features = {}

    for angle in [0, np.pi/4, np.pi/2, 3*np.pi/4]:matrix_cooccurrence =
graycomatrix(inds, [1], [angle], levels=max_value, normed=True,
symmetric=True)
        contrast = graycoprops(matrix_cooccurrence, 'contrast')[0][0]
        homogeneity = graycoprops(matrix_cooccurrence,
'homogeneity')[0][0]
        energy = graycoprops(matrix_cooccurrence, 'energy')[0][0]
        correlation = graycoprops(matrix_cooccurrence,
'correlation')[0][0]glcm_features[f'contrast_{int(np.degrees(angle))}']
= contrast
        glcm_features[f'homogeneity_{int(np.degrees(angle))}'] =
homogeneity
        glcm_features[f'energy_{int(np.degrees(angle))}'] = energy
        glcm_features[f'correlation_{int(np.degrees(angle))}'] =
correlation

    return glcm_features
```

Program tersebut menjelaskan mengenai fungsi ekstraksi tekstur GLCM diawali dengan konversi citra ke skala abu-abu menggunakan fungsi `cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)`. Selanjutnya dibuat *array numpy* yang berisi batasan nilai piksel untuk digitisasi, sehingga program dapat memprosesnya. Dalam program ini terdapat interval nilai piksel yang mewakili 16-bit. Kemudian fungsi `np.digitize` digunakan untuk mengonversi gambar menjadi bentuk digital sesuai interval yang ditentukan oleh variabel `bins`. Hasil konversi ini disimpan dalam variabel `inds`. Nilai maksimal dari `inds` kemudian dihitung dan ditambah dengan 1 untuk mendapatkan jumlah level dalam *matrix co-occurrence*.

Selanjutnya keterangan `[0, np.pi/4, np.pi/2, 3*np.pi/4]` merupakan sudut offset, fungsi `graycomatrix()` digunakan untuk menghitung *matrix co-occurrence*. Argumen indeks adalah citra yang telah digitalisasi, `[1]` adalah pixel offset, `levels` adalah jumlah level dalam *matrix co-occurrence*, `normed=True` menunjukkan bahwa adanya normalisasi dalam proses ekstraksi ini dan `symmetric=True` menunjukkan bahwa matriks tersebut bersifat simetris. Berikut merupakan contoh ekstraksi tekstur pada citra penyakit daun tomat.

Tabel 4. 3 Contoh Nilai Ekstraksi Fitur GLCM Label Daun Hawar

No	<i>Contrast</i>	<i>Homogeneity</i>	<i>Energy</i>	<i>Correlation</i>
0 <sup>0</sup>	0.3266	0.8949	0.5639	0.9882
45 <sup>0</sup>	0.5678	0.8672	0.5585	0.9795
90 <sup>0</sup>	0.2789	0.9074	0.5654	0.9899
135 <sup>0</sup>	0.4185	0.8783	0.5611	0.9845

**Tabel 4. 4 Contoh Nilai Ekstraksi Fitur GLCM Label Infeksi Bakteri**

No	<i>Contrast</i>	<i>Homogeneity</i>	<i>Energy</i>	<i>Correlation</i>
0 <sup>0</sup>	0.2353	0.9025	0.4811	0.9806
45 <sup>0</sup>	0.4094	0.8640	0.4707	0.9662
90 <sup>0</sup>	0.2894	0.8892	0.4781	0.9761
135 <sup>0</sup>	0.3907	0.8672	0.4715	0.9677

**Tabel 4. 5 Contoh Nilai Ekstraksi Fitur GLCM Nilai Kudis daun**

No	<i>Contras</i>	<i>Homogeneity</i>	<i>Energy</i>	<i>Correlation</i>
0 <sup>0</sup>	0.2258	0.9157	0.6309	0.9859
45 <sup>0</sup>	0.3700	0.8880	0.6272	0.9770
90 <sup>0</sup>	0.3117	0.8973	0.6295	0.9806
135 <sup>0</sup>	0.4772	0.8723	0.6260	0.9703

**Tabel 4. 6 Contoh Nilai Ekstraksi Fitur GLCM Label Daun Sehat**

No	<i>Contras</i>	<i>Homogeneity</i>	<i>Energy</i>	<i>Correlation</i>
0 <sup>0</sup>	0.4601	0.8775	0.5599	0.9853
45 <sup>0</sup>	0.6979	0.8504	0.5543	0.9777
90 <sup>0</sup>	0.4056	0.8798	0.5604	0.9870
135 <sup>0</sup>	0.6354	0.8544	0.5551	0.9797

Tabel 4.3 hingga Tabel 4.6 menyajikan contoh nilai ekstraksi fitur GLCM dari citra motif penyakit daun tomat dengan label daun hawar, infeksi bakteri, kudis daun, dan daun sehat. Ekstraksi fitur tekstur menghasilkan empat sudut yang berbeda yaitu 0<sup>0</sup>, 45<sup>0</sup>, 90<sup>0</sup>, 135<sup>0</sup>. Setiap fitur menghasilkan empat nilai berbeda arah sudut masing-masing. Dengan menggunakan empat parameter GLCM, total fitur yang dihasilkan sebanyak 16 fitur.

#### 4.1.5 Klasifikasi SVM

Tahap selanjutnya yaitu tahap mengklasifikasi dengan metode *Support Vector Machine*. Berdasarkan nilai-nilai input untuk mengklasifikasi dengan SVM.

---

##### **Modul Program 4. 6 Model SVM**

---

```
from sklearn.svm import SVC

svm_linear = SVC(kernel='linear', C=1.0)
svm_linear.fit(X_train, y_train)
```

---

Berdasarkan Modul Program 4.6 pelatihan klasifikasi SVM dilakukan dengan memanfaatkan modul bernama SVC() yang di impor dari *scikit-learn*. Setelah itu dilakukan inisialisasi parameter C=1 karena kernel yang digunakan adalah *linear*. Setelah itu dilakukan fit() untuk melakukan pelatihan model SVM.

## 4.2 Pengujian

Pada bagian ini akan diberi penjelasan terkait proses pengujian model pelatihan dan pengujian model dengan pengujian data *training* sebesar 80% dan data *testing* 20%, data *training* sebesar 85% dan data *testing* 15%, dan data *training* sebesar 90% dan data *testing* 10%. Selain itu juga mencakup pengujian terhadap sistem klasifikasi yang telah diimplementasikan sebelumnya.

### 4.2.1 Pengujian Model

Tahap pengujian model dilakukan pada proses pelatihan klasifikasi yang berfungsi untuk memastikan bahwa model yang dilatih dapat melakukan klasifikasi dengan baik. Pengujian model pada penelitian ini menggunakan metode *confusion matrix* untuk menguji model klasifikasi penyakit daun tomat.

---

#### Modul Program 4.7 *Confusion Matrix*

---

```
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, ConfusionMatrixDisplay

conf_matrix=confusion_matrix(y_test,y_pred)
display=ConfusionMatrixDisplay(confusion_matrix=conf_matrix,display_labels=svm_linear.classes_)
display.plot()
plt.title(f"confusion_matrix for linear kernel ")
plt.show()

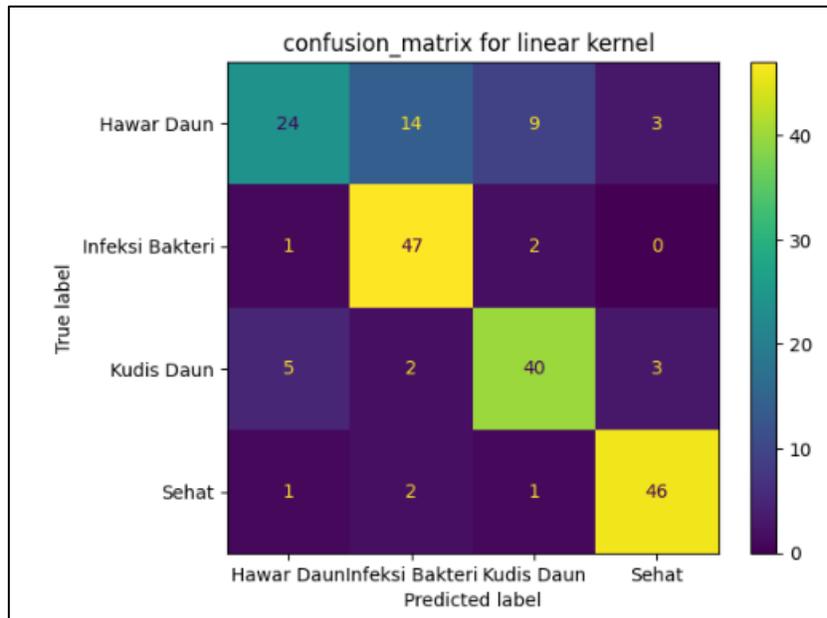
class_report = classification_report(y_test, y_pred,
target_names=svm_linear.classes_)
print("Classification Report:")
print(class_report)
```

---

Berdasarkan Modul Program 4.7 pengujian model dilakukan dengan memanfaatkan modul *confusion\_matrix*, *classification\_report*, dan *ConfusionMatrixDisplay* yang diimport dari *sklearn.metrics*. Fungsi *confusion\_matrix* menerima label dari data set dan label dari data yang diprediksi oleh model. Kemudian *confusion matrix* divisualisasikan menggunakan fungsi *ConfusionMatrixDisplay* beserta dengan label yang ada pada data latih. Kemudian untuk mendapatkan nilai akurasi, *precision*, dan *recall* digunakan fungsi *classification\_report*. Pengujian model tersebut digunakan untuk pengujian model dengan pembagian data yaitu data *training* sebesar 80% dan data *testing* 20%, data *training* sebesar 85% dan data *testing* 15%, dan data *training* sebesar 90% dan data *testing* 10%.

a. Data *Training* 80% dan Data *Testing* 20%

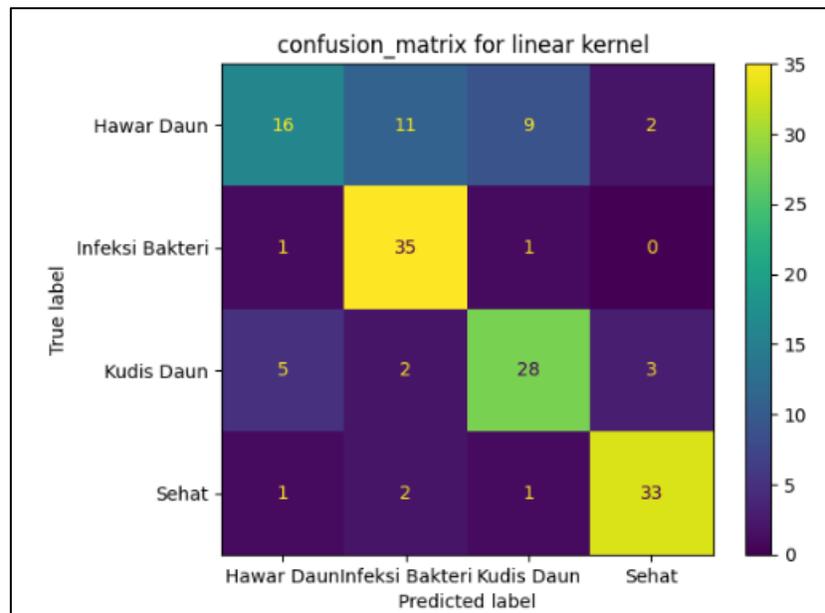
Pengujian model pertama yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 80% dan data *testing* 20%. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 80:20 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



Gambar 4. 4 *Confusion Matrix Data Training 80% Dan Testing 20%*

b. Data *Training* 85% dan Data *Testing* 15%

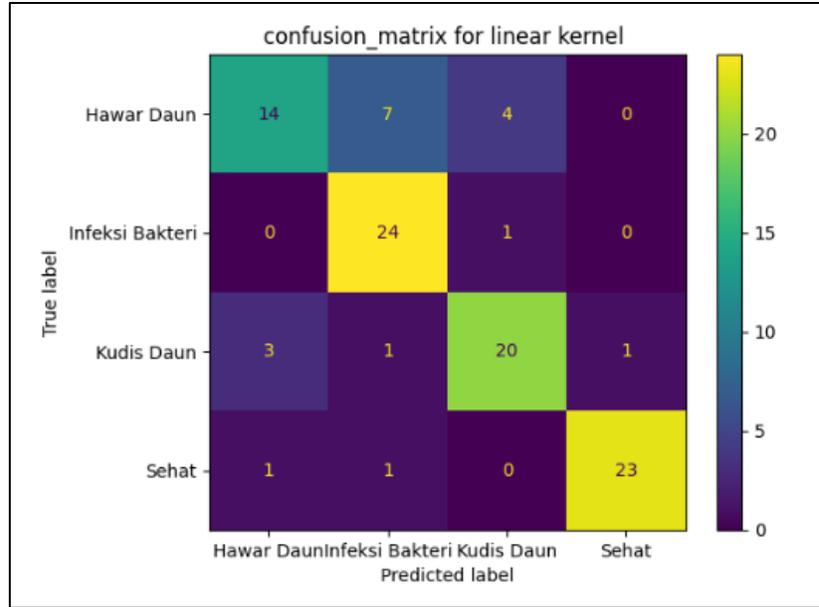
Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 85% dan data *testing* 15%. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 85:15 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



Gambar 4. 5 *Confusion Matrix Data Training 85% Dan Testing 15%*

c. Data *Training* 90% dan *Testing* 10%

Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10%. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confution matrix* sebagai berikut.



Gambar 4. 6 *Confusion Matrix* Data *Training* 90% Dan *Testing* 10%

Hasil pengujian beberapa sekenario yang telah dilakukan yang mencakup nilai *precision*, *recall*, *F1-score*, serta akurasi yang telah diperoleh menggunakan metode *confusion matrix* dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Hasil *Precision*, *Recall*, *F1-Score*, *Accuracy* Pengujian Pembagian Data

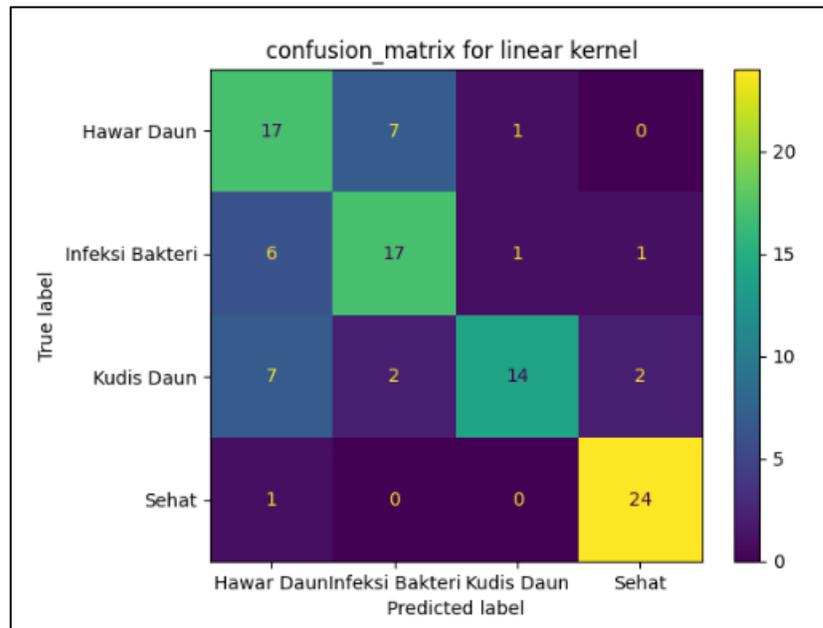
Pengujian	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
1 (80:20)	79%	78%	79%	79%
2 (85:15)	75%	75%	73%	75%
3 (90:10)	79%	81%	80%	81%

Berdasarkan hasil pengujian yang dapat dilihat pada Tabel 4.7 diperoleh nilai terendah pada *precision* sebesar 75% dan nilai tertinggi sebesar 79%. Nilai terendah pada *recall* sebesar 75% dan tertinggi sebesar 81%. Nilai terendah pada *f1-score* sebesar 73% dan tertinggi sebesar 80% pada *accuracy* nilai terendah adalah 75% sedangkan tertinggi adalah 81%. Pengujian selanjutnya yaitu dengan mengubah *random\_state* dengan rentang 0-84

menggunakan perbandingan data *training* 90% dan data *testing* 10% serta metode evaluasi *confusion matrix*.

a. *Confusion Matrix* dengan *Random State* 0

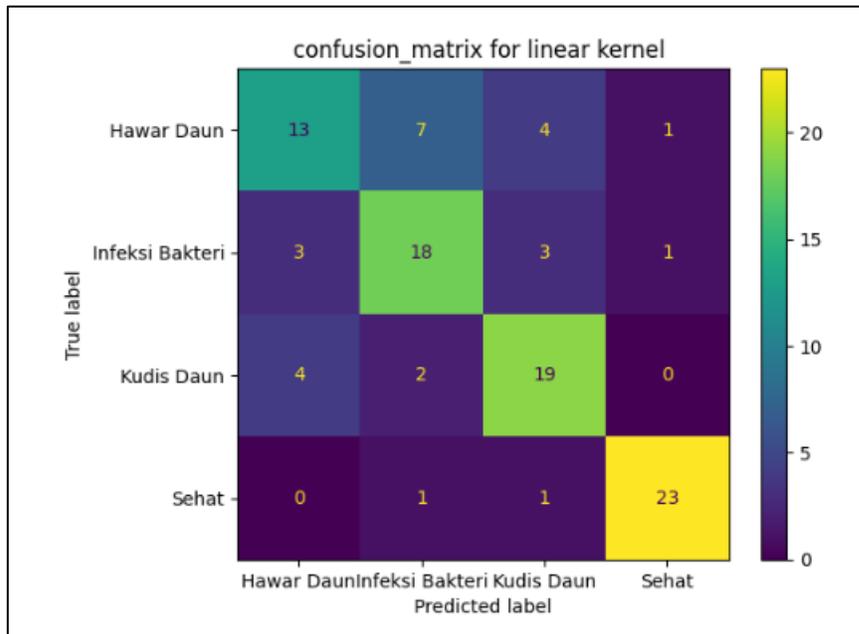
Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10% serta nilai *random\_state* 0. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



**Gambar 4.7** *Confusion Matrix* dengan *Random State* 0

b. *Confusion Matrix* dengan *Random State* 21

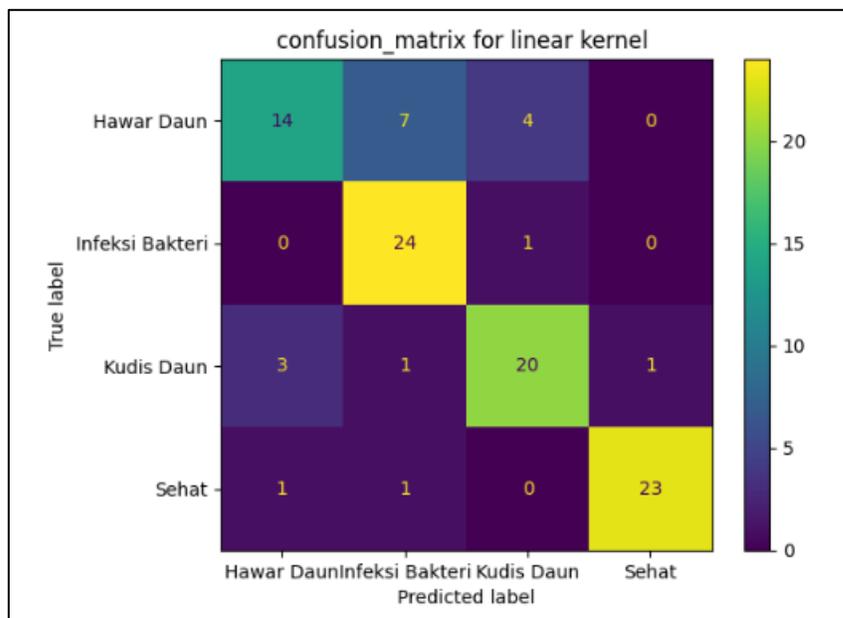
Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10% serta nilai *random\_state* 21. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



**Gambar 4. 8** *Confusion Matrix* dengan *Random State 21*

c. *Confusion Matrix* dengan *Random State 42*

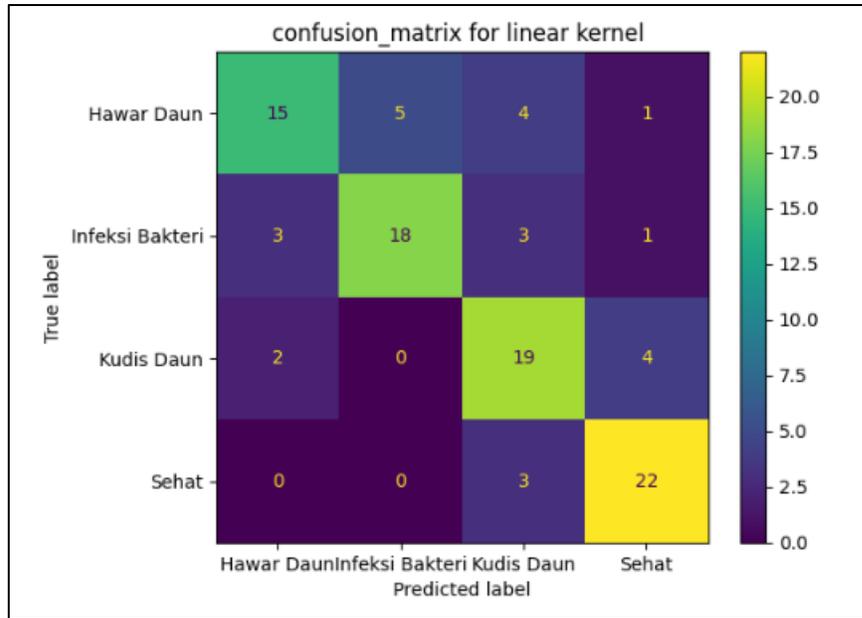
Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10% serta nilai *random\_state* 42. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



**Gambar 4. 9** *Confusion Matrix* dengan *Random State 42*

d. *Confusion Matrix dengan Random State 63*

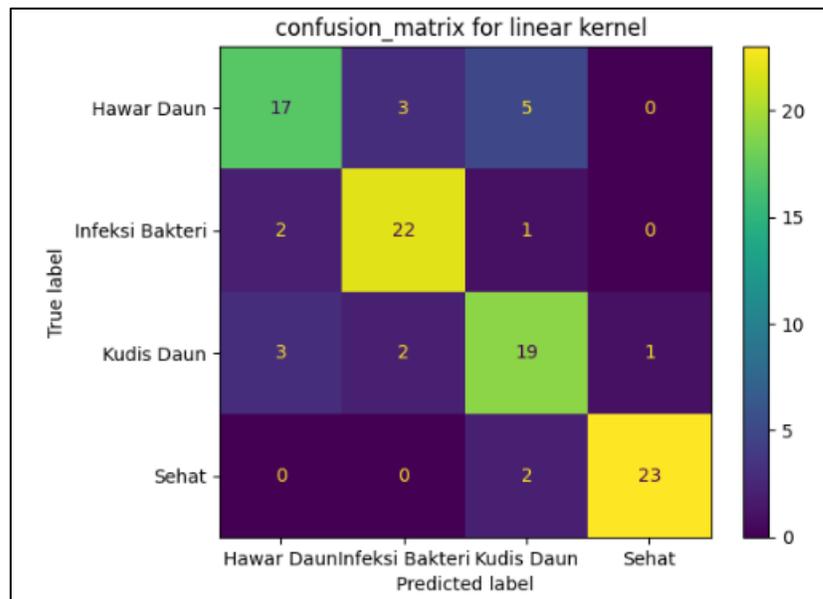
Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10% serta nilai *random\_state* 63. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



**Gambar 4. 10** *Confusion Matrix dengan Random State 63*

e. *Confusion Matrix dengan Random State 84*

Pengujian model kedua yaitu ekstraksi fitur HSV dan ekstraksi fitur GLCM dengan menggunakan SVM dengan data *training* sebesar 90% dan data *testing* 10% serta nilai *random\_state* 84. Setelah mendapatkan nilai ekstraksi fitur HSV dan GLCM kemudian nilai tersebut dilakukan pembagian data dengan rasio 90:10 kemudian di-*training* menggunakan SVM dan menghasilkan *confusion matrix* sebagai berikut.



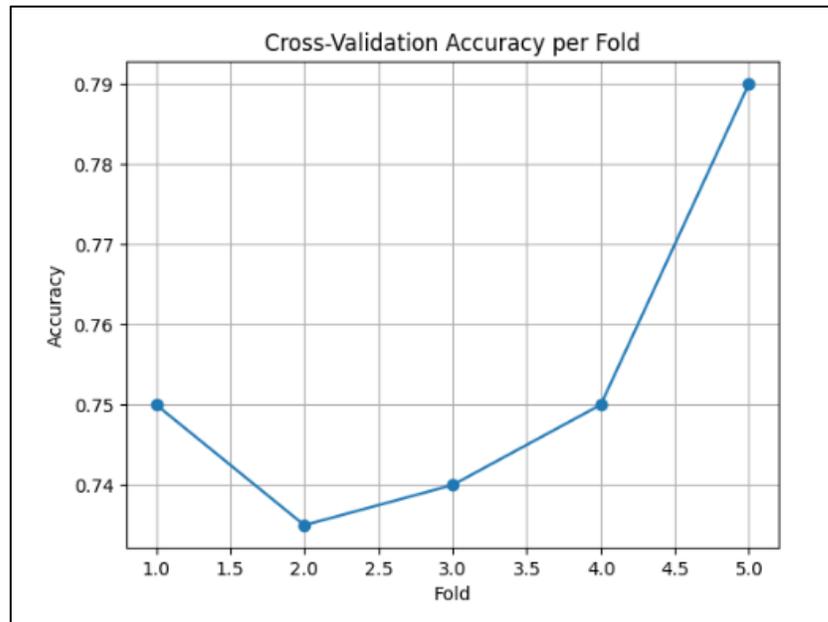
**Gambar 4. 11 Confusion Matrix dengan Random State 84**

Hasil pengujian beberapa skenario yang telah dilakukan yang mencakup nilai *precision*, *recall*, *F1-score*, serta akurasi yang telah diperoleh menggunakan metode *confusion matrix* dengan rentang nilai *Random State* 0-84 dapat dilihat pada Tabel 4.8.

**Tabel 4. 8 Hasil Precision, Recall, F1-Score, Accuracy Pengujian Random State**

Pengujian	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
1 (random_state= 0)	74%	72%	72%	72%
2 (random_state= 21)	73%	73%	73%	73%
3 (random_state= 42)	82%	81%	80%	81%
4 (random_state= 63)	74%	74%	74%	74%
5 (random_state= 84)	81%	81%	81%	81%

Berdasarkan hasil pengujian yang dapat dilihat pada Tabel 4.8 diperoleh nilai terendah pada *precision* sebesar 73% dan nilai tertinggi sebesar 82%. Nilai terendah pada *recall* sebesar 72% dan tertinggi sebesar 81%. Nilai terendah pada *f1-score* sebesar 72% dan tertinggi sebesar 81% pada *accuracy* nilai terendah adalah 72% sedangkan tertinggi adalah 81%. Pengujian yang dilakukan selanjutnya yaitu menggunakan metode evaluasi *K-Fold Cross Validation* dengan nilai K=5 serta nilai *Random State* sebesar 42.



**Gambar 4.12 K-Fold Cross Validation dengan nilai K=5**

Berdasarkan Gambar 4.12 didapatkan nilai akurasi pada *fold* pertama yaitu sebesar 75%, pada *fold* kedua yaitu sebesar 73%, pada *fold* ketiga yaitu sebesar 74%, pada *fold* keempat yaitu sebesar 75%, dan pada *fold* kelima yaitu sebesar 79%. Berdasarkan hasil lima *fold* tersebut didapatkan nilai rata-rata akurasi yaitu sebesar 75%. Pengujian yang dilakukan selanjutnya yaitu menggunakan metode evaluasi *K-Fold Cross Validation* dengan nilai  $K=5$  serta nilai *Random State* sebesar 42.

**Tabel 4.9 Hasil Precision, Recall, F1-Score, Accuracy Pengujian K-Fold Cross Validation (K=5)**

Pengujian	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
<i>K-Fold Cross Validation (K=5)</i>	75%	75%	75%	75%

Hasil pengujian yang dapat dilihat pada Tabel 4.13 diperoleh *precision* sebesar 75%. Nilai pada *recall* sebesar 75%. Nilai *f1-score* sebesar 75% dan nilai *accuracy* adalah 75%. Berdasarkan tiga pengujian yang telah dilakukan didapatkan akurasi terbaik dengan pembagian data *training* sebesar 90% dan data *testing* sebesar 10% serta nilai *Random State* sebesar 42 dan penggunaan metode evaluasi *Confusion Matrix* menghasilkan akurasi sebesar 81%.

#### 4.2.2 Pengujian Sistem

Tahap selanjutnya adalah pengujian sistem yang menggunakan metode pengujian *black box testing*. *Black box testing* yaitu sebuah metode pengujian yang berfokus pada pengujian fungsionalitas sistem dari perspektif pengguna akhir tanpa melihat ke dalam struktur internal atau kode sumber sistem tersebut. Proses pengujian ini mencakup berbagai skenario yang mencerminkan kondisi penggunaan nyata untuk mengidentifikasi potensi atau masalah

fungsional lainnya. Hasil dari pengujian ini, mencakup data tentang kinerja sistem yang disajikan dalam Tabel 4.10 berikut ini.

**Tabel 4. 10 Pengujian sistem *Black Box Testing***

Skenario ke	Test Case	Ekspektasi Hasil	Hasil		
			Bagas	Akmal	Vito
1	User dapat menginput file untuk dilakukan proses klasifikasi citra	Berhasil	Berhasil	Berhasil	Berhasil
2	Sistem dapat menampilkan hasil <i>Resize</i> dan <i>Remove Background</i> , HSV, dan <i>Grayscale</i> citra	Berhasil	Berhasil	Berhasil	Berhasil
3	Sistem dapat menampilkan hasil dari parameter HSV dan GLCM	Berhasil	Berhasil	Berhasil	Berhasil
4	Sistem dapat menampilkan hasil klasifikasi metode SVM	Berhasil	Berhasil	Berhasil	Berhasil

### 4.3 Pembahasan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan pada penelitian ini penerapan ekstraksi fitur *Hue Saturation Value* dan ekstraksi fitur tekstur *Gray Level Co-occurrence Matrix* dengan metode klasifikasi *Support Vector Machine* dapat melakukan klasifikasi penyakit daun tomat. Penelitian ini menggunakan data berjumlah 1000 data citra yang terbagi menjadi empat kategori yaitu daun hawar, infeksi bakteri, kudis daun, dan daun sehat dengan masing-masing kategori memiliki 250 data citra. Data kemudian dibagi menjadi data *training* dan data *testing*, dengan perbandingan 90:10. Pembagian dengan perbandingan tersebut menghasilkan 900 untuk data *training* dan 100 untuk data *testing*. Data *training* digunakan untuk melatih model klasifikasi sementara data *testing* digunakan untuk mengukur akurasi dari model klasifikasi.

Tahapan pertama pada penelitian ini dimulai dari analisis kebutuhan data yang diperoleh dari situs *kaggle*. Tahapan selanjutnya setelah data didapatkan, data tersebut melalui tahap *pre-processing* untuk mendapatkan kualitas data sesuai yang dibutuhkan untuk proses selanjutnya. Tahapan *pre-processing* yang dilakukan yaitu *remove background* dan *resize*. *Remove background* berfungsi untuk menghilangkan latar belakang pada citra sehingga hanya tersisa objek utamanya saja. *Resize* berfungsi untuk mengubah ukuran citra agar masing-masing citra memiliki ukuran yang sama.

Tahapan selanjutnya setelah data dilakukan *pre-processing* yaitu mendapatkan nilai ekstraksi fitur untuk menjadi nilai inputan klasifikasi. Ekstraksi fitur yang pertama yaitu ekstraksi fitur warna menggunakan *Hue Saturation Value*. Ekstraksi fitur kedua yaitu ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix*. Setelah mendapatkan nilai dari kedua ekstraksi fitur tersebut, nilai ekstraksi fitur akan digunakan model klasifikasi dengan metode *Support Vector Machine*. Selanjutnya model klasifikasi tersebut dilakukan proses evaluasi menggunakan metode *Confusion Matrix*.

Pengujian yang dilakukan pada penelitian ini menunjukkan nilai *accuracy* 79% rata-rata *recall* 79% dan rata-rata *precision* 79% untuk penerapan *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan metode klasifikasi *Support Vector Machine* dengan pembagian data *training* sebesar 80% dan data *testing* 20%. Kemudian untuk penerapan *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan metode klasifikasi *Support Vector Machine* dengan pembagian data *training* sebesar 85% dan data *testing* 15% menunjukkan nilai *accuracy* 75%, rata-rata *recall* 75% dan rata-rata *precision* 75%. Selanjutnya untuk penerapan *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan metode klasifikasi *Support Vector Machine* dengan pembagian data *training* 90% dan data *testing* 10% menunjukkan nilai *accuracy* 81%, rata-rata *recall* 81% dan rata-rata *precision* 81%. Pengujian selanjutnya yaitu menggunakan pembagian data *training* sebesar 90% dan data *testing* sebesar 10% dengan nilai *Random State* 0-84. Pengujian pertama dengan nilai *Random State* 0 menunjukkan nilai *accuracy* 72%, rata-rata *recall* 72% dan rata-rata *precision* 74%. Pengujian kedua dengan nilai *Random State* 21 menunjukkan nilai *accuracy* 73%, rata-rata *recall* 73% dan rata-rata *precision* 73%. Pengujian ketiga dengan nilai *Random State* 42 menunjukkan nilai *accuracy* 81%, rata-rata *recall* 81% dan rata-rata *precision* 82%. Pengujian keempat dengan nilai *Random State* 63 menunjukkan nilai *accuracy* 74%, rata-rata *recall* 74% dan rata-rata *precision* 74%. Pengujian kelima dengan nilai *Random State* 84 menunjukkan nilai *accuracy* 81%, rata-rata *recall* 81% dan rata-rata *precision* 81%. Pengujian selanjutnya yaitu menggunakan metode evaluasi *K-Fold Cross Validation* dengan nilai  $K=5$ . Pengujian tersebut menghasilkan nilai *accuracy* rata-rata sebesar 75%, nilai rata-rata *precision* 75%, rata-rata *recall* 75, dan rata-rata *f1-score* 75%. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa penerapan *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan metode klasifikasi *Support Vector Machine* dengan pembagian data *training* 90% dan data *testing* 10% serta nilai *Random State* sebesar 42 dan penggunaan metode evaluasi *Confusion Matrix* mendapatkan akurasi tertinggi.

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Berdasarkan hasil dari penelitian klasifikasi penyakit daun tomat dengan metode *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan *Support Vector Machine* didapatkan kesimpulan sebagai berikut:

1. Klasifikasi penyakit daun tomat dengan metode *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan *Support Vector Machine* dapat diimplementasikan.
2. Klasifikasi penyakit daun tomat dengan metode *Hue Saturation Value* dan *Gray Level Co-occurrence Matrix* menggunakan *Support Vector Machine* mendapatkan akurasi terbaik sebesar 81% dengan pembagian data *training* 90% dan data *testing* 10%.

### **5.2 Saran**

Adapun saran yang dapat dilakukan dalam mengembangkan penelitian ini adalah sebagai berikut.

1. Melakukan optimasi pada parameter yang digunakan pada *Support Vector Machine* seperti *Particle Swarm Optimization*.
2. Menggunakan dan menguji dengan metode klasifikasi lain untuk membandingkan tingkat akurasi dalam klasifikasi penyakit daun tomat.
3. Menggunakan dataset penyakit daun tomat yang memiliki kualitas gambar lebih baik agar mendapat hasil yang lebih baik.

## DAFTAR PUSTAKA

- Adawiyah et al., R. (2022). Optimasi Deteksi Penyakit Kulit Menggunakan Metode Support Vector Machine (SVM) dan Gray Level Co-occurrence Matrix (GLCM). *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, Vol 14 No.1, 18-33.
- Alwy et al., A. D. (2023). Klasifikasi Penyakit Pada Padi Dengan Ekstraksi Fitur LBP dan GLCM. *Jurnal of Deep Learning, Computer Vasion and Digital Image Prosessing*, 1-10.
- Amelia et al. (2023). Klasifikasi Citra Daun Mengga Menggunakan Algoritma Support Vector Machine. *Jurnal Ilmiah Manajemen Informatika dan Komputer Vol 07, No. 03*, 80--83.
- Amran, M. A. (2023). *Dasar Dasar Penginderaan Jauh Satelit*. Makassar: Penerbit Nas Media Pustaka.
- Andono et al. (2017). *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
- Arnita et al., F. M. (2022). *Computer Vision Dan Pengolahan Citra Digital*. Surabaya: Pustaka Aksara.
- Astiningrum et al., M. P. (2020). Identifikasi Penyakit Pada Daun Tomat Berdasarkan Fitur Warna dan Tekstur. *Jurnal Informatika Polinema, Vol. 6 Edisi 2*, 47-50.
- Astriratama, I. &. (2021). Klasifikasi Jenis Pantun dengan Metode Support Vector Machines (SVM). *Jurnal Rekayasa Sistem dan Teknologi Informasi, Vol 4, No 5*, 915-922.
- Erwanto et al., B. A. (2024). Pengembangan Sistem Deteksi Penyakit TanamanTomat Melalui Citra Daun dengan MetodeYou Only Look Once (YOLO) Berbasis Android. *Jurnal Teknologi Terapan, Vol.8, No. 3*, 1453-1463.
- Ghojogh et al., B. (2023). *Elements of Dimensionality Reduction and MANifold Learning*. Switzerland: Springer.
- Hamzaidah, P. &. (2023). *Pengolahan Citra dan Video* . Klaten : Penerbit Nasmedia.
- Kotta et al. (2022). Implementation of Convolutional Neural Network Method to Detect Diseases. *Jurnal Pekommas Vol. 7 No. 2*, 123-132.
- Kurniawan & Junaidi, S. D. (2022). Implementasi Algoritma K-Nearest Neighbor Dengan Metode Hue Saturation Value Untuk Pendeteksi Kematangan Buah Jambu. *Smart Comp, Vol. 1, No. 3*, 541-547.
- Kurniawan et al., R. A. (2023). Pengaruh Arsitektur Convolutional Neural Network Untuk Klasifikasi Penyakit Daun Tomat. *Teknologi Informasi dan Multimedia 4 (2)* , 126-131.

- Kusanti & Haris. (2018). Klasifikasi Penyakit Daun Padi Berdasarkan Hasil Ekstraksi Fitur GLCM Interval 4 Sudut. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, Vol.03, No.01, 1-6.
- La choviya hawa, Y. W. (2023). *Karakteristik Fisik Produk Pertanian*. Malang: Media Nusa Creative.
- Lubis, A.-K. &. (2023). *Artificial Intelligence*. Medan: UMSU Press.
- Muhammad et al., A. C. (2023). *Dasar Dasar Pembelajaran Mesin (Foundations of Machine Learning)*. Banten: Sada Kurnia Pustaka dan Penulis. Dipetik 3 2, 2025, dari [https://www.google.co.id/books/edition/Dasar\\_dasar\\_Pembelajaran\\_Mesin/8COzEAAAQBAJ?hl=ban&gbpv=1&dq=kernel+linier&pg=PA56&printsec=frontcover](https://www.google.co.id/books/edition/Dasar_dasar_Pembelajaran_Mesin/8COzEAAAQBAJ?hl=ban&gbpv=1&dq=kernel+linier&pg=PA56&printsec=frontcover)
- Mulyana & Muthmainnah. (2023). Analisis Perbandingan Tingkat Akurasi Algoritma CNN dan SVM Dalam Klasifikasi Pada Daun Gedi, Daun Pepaya dan Daun Ubi. *Jurnal Mahasiswa Teknik Informatika*, Vol. 8, No. 4, 6157-6162.
- Nahak et al., E. R. (2024). Klasifikasi Penyakit Pada Tanaman Apel Melalui Citra Daun Menggunakan Metode Multiclass Support Vector Machine . *jurnal Teknik Informatika dan Sistem Informasi*, Vol. 11, No. 3, 401-408 .
- Nainggolan et al. (2022). Identifikasi Penyakit Tanaman Tomat Berdasarkan Citra Penyakit Menggunakan Metode GLCM dan Naïve Bayes Classifier. *Methodika: Jurnal Ilmiah Teknik Informatika*, 22-28.
- Ningsih et al., N. P. (2022). Klasifikasi Penyakit Early Blight dan Late Blight Pada Tanaman Tomat Berdasarkan Citra Duan Menggunakan Metode CNN Berbasis Website. *Jurnal Kecerdasan Buatan dan Teknologi Informasi (JKBTI)* Vol. 1, No. 3, 27-35.
- Nur'aini, H. I. (2019). *Mengenal Tanaman Hortrikultural*. Bandung: Penerbit Duta.
- Puerwandono & Maulana. (2023). Penerapan Algoritma Svm Untuk Klasifikasi Citra Daun Sirih Application of The SVM Algorithm Image Classification of Betel Leaf. *Journal of Information Technology and Computer Science (INTECOMS)*, Vol. 6 ,No.2, 859-865.
- Putra et al. (2025). *Implementasi Deep Learning dan Computer Vision untuk Analisis Kerusakan Jalan: Teori dan Studi Pustaka*. Bojong: Penerbit NEM.
- Putra et al., R. F. (2024). *Algoritma Pembelajaran Mesin (Dasr, Teknik, dan Aplikasi)*. kota jambi: Sonpedia Publishing.
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.
- Qomaruddin et al., M. (2021). Segmentasi K-Means Citra Daun Tin Dengan Klasifikasi Ciri Grey Level Co-occurrence Matrix. *Jurnal Sistem dan Teknik Informasi*, Vol. 9, No.2, 223-233.

- Qur'ania et al., A. L. (2023). Identifikasi Defisiensi Unsur Pada Tanaman Cabai Menggunakan Support Vector Machine . *J-ICON, Vol. 11 No. 1*, 62-67.
- Sahrani. (2021). Klasifikasi Penyakit Daun Tomat Berdasarkan Ekstraksi Tekstur Daun Menggunakan Gabor Filter dan Algoritma Support Vector Machine. *Repository UINSU*, 1-100. Diambil kembali dari file:///C:/Users/ASUS/Downloads/Documents/SKRIPSI\_LELY\_perpu.pdf
- Sahrani, L. (2021). Klasifikasi Penyakit Daun Tomat Berdasarkan Ekstraksi Tekstur Daun Menggunakan Gabir Filter dan Algoritma SVM. *Repository UINSU*, 1-100.
- Sanusi et al., H. (2019). Pembuatan Aplikasi Klasifikasi Citra Daun Menggunakan Ruang Warna RGB dan HSV. *Jurnal Ilmiah Informatika Komputer Volume 24 No. 3*, 180-190.
- Saputra et al. (2023). Deteksi Penyakit Tomat Melalui Citra Daun Menggunakan Metode Convolutional Neural Network. *Aviation Electronics, Information Technology, Telecommunications, Electricals, Controls (AVITEC) Vol. 5, No. 1*, 43-51.
- Saputra et al., R. A. (2022). Deteksi Kematangan Buah Melon dengan Algoritma Support Vector Machine Berbasis Ekstraksi Fitur GLCM. *Jurnal Infortech, Volume 4 No. 2* , 200-206.
- Sari & Wihandika. (2019). Klasifikasi Jenis Citra Makanan Tunggal Berdasarkan Fitur Local Binary Patterns Dan Hue Saturation Value Menggunakan Improved K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 3, No. 3*, 2416-2424.
- Satrio, B. (2022). Perbandingan Multikelas Mesin Vektor Dukungan Kernel Dalam Analisis Deteksi PPKBM di Twitter. *Portaldata.org Vol. 2 No. 8*, 1-12.
- Sekaran & Bougie. (2017). *Metode Penelitian untuk Bisnis Pendekatan Pengembangan Keahlian* . Jakarta Selatan : Penerbit Salemba.
- Shinta & Atika, W. (2022). Klasifikasi Bunga Mawar Menggunakan KNN dan Ekstraksi Fitur GLCM dan HSV. *SKANIKA: Sistem Komputer dan Teknik Informatika*, 145-156.
- Sinaga, A. S. (2019). *Ekstraksi Ciri Komunikasi Non-Verbal Grey Level Co-occurrence Matrix dan Fuzzy C-Means*. Pasuruhan: Penerbit Qiara Media.
- Sooai et al., A. G. (2023). Klasifikasi Citra Daun Anggur Menggunakan SVM Kernel Linear. *Jurnal if Information Technology and Computer Science, Vol.8, No1*, 1926.
- Sriani et al., S. Y. (2024). Klasifikasi Kualitas Daun Tembaku Menggunakan GLCM (Grey Level Co-Occurrence Matrix) dan SVM (Support Vector Machine). *JITET (Jurnal Informatika dan Teknik Elektro Terapan), Vol. 12 No. 3*, 3342-3349.
- Suhendra et al., R. I. (2022). Identifikasi dan Klasifikasi Penyakit Daun Jagung Menggunakan Support Vector Machine. *JURNAL TEKNOLOGI INFORMASI - VOL. 1 NO. 1*, 2935.

- Sutisna & Sumantri. (2022). Klasifikasi Tingkat Kematangan Buah Nanas Berdasarkan Tekstur Gray Level Cooccurrence Matrix Dengan Metode Support Vector Machine. *JINTEKS (Jurnal Informatika Teknologi dan Sains) Vol. 04, No. 4*, 296-301.
- Tampinongkol et al., F. F. (2023). Identifikasi Penyakit Daun Tomat Menggunakan Grey Level Co-occurrence Matrix (GLCM) dan Support Vector Machine (SVM). *Techno Eplore Jurnal Ilmu Komputer dan Teknologi Informasi, Vol. 8, No.1*, 8-16.
- Werdingingsih et al., I. D. (2022). *Pengolahan Data Mining Dengan Pemrograman Matlab*. Jawa Timur: Airlangga University Press.
- Wibowo, M. &. (2023). Implementasi Tingkat Kematangan Buah Monk Dengan Menggunakan Ekstraksi Gray Level Co Occurrence dan Support Vector Machine. *JINTEKS (Jurnal Informatika Teknologi dan Sains), Vol. 5 No. 3*, 334 – 339.
- Widodo, S. (2021). *Ekstraksi Fitur Citra Biomedik: Dilengkapi dengan Source Code Matlab*. Magelang: Penerbit Pustaka Rumah C1nta. Dipetik 24, 2025, dari [https://www.google.co.id/books/edition/Ekstraksi\\_Fitur\\_Citra\\_Biomedik/ZYRaEAAAQBAJ?hl=ban&gbpv=1&dq=fitur+HSV&pg=PA38&printsec=frontcover](https://www.google.co.id/books/edition/Ekstraksi_Fitur_Citra_Biomedik/ZYRaEAAAQBAJ?hl=ban&gbpv=1&dq=fitur+HSV&pg=PA38&printsec=frontcover)
- Wiryanta, W. (2002). *Kiat mengatasi permasalahan praktis Bertanam Tomat*. Jakarta Selatan: Argo Media Pustaka.
- Wulandari & Sifaunajah. (2019). *Perbandingan Efektifitas Klasifikasi A-lgoritma C.4.5 dan Algoritma Naive Bayes Dengan Menggunakan Pihak Ke 3 (WEKA)*. Jombang: LPPM Universita KH. A. Wahab Hasbullah.
- Wulandari et al., D. A. (2023). *Perkembangan Tren Teknologi Deteksi Kemurnian Susu*. Jember : UM Jember Press.
- Wulandari et al., N. K. (2023). Efektifitas Formulasi Pupuk Organik Cair dan Pupuk Hayati-P60 Mengendalikan Penyakit Hawar Daun Bakteri pada Tanaman Tomat Ceri (*Solanum lycopersicum*) Sistem Hidroponik. *Agro Bali : Agricultural Journal, Vol. 6 No. 01*, 74-81.
- Zein et al., A. D. (2022). *Konsep Dasar Rekayasa Perangkat Lunak*. Batam: Penerbit Cendekia Mulia Mandiri.
- Zonyfar, C. (2020). *Pengolahan Citra Digital Sebuah Pengantar*. Banten: Penerbit Desanta Muliavisitama.