

**IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA  
SAWIT DENGAN PENDEKATAN NAÏVE BAYES  
MENGGUNAKAN OPTIMASI GENETIC ALGORITHM  
DAN PARTICLE SWARM OPTIMIZATION**

(STUDI KASUS: PT PERKEBUNAN NUSANTARA IV REGIONAL 7  
KEBUN BEKRI)

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di  
Program Studi Teknik Informatika, Fakultas Teknologi Industri,  
Institut Teknologi Sumatera

**Oleh:**

**Natasya Ate Malem Bangun**

**121140052**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN  
2025**

## **LEMBAR PENGESAHAN**

Tugas Akhir Sarjana dengan judul "IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA SAWIT DENGAN PENDEKATAN NAÏVE BAYES MENGGUNAKAN OPTIMASI GENETIC ALGORITHM DAN PARTICLE SWARM OPTIMIZATION (Studi Kasus: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri)" adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan, 21 Mei 2025  
Penulis,

Natasya Ate Malem Bangun  
NIM 121140052

Foto 2x3

Diperiksa dan disetujui oleh,

Pembimbing

Tanda Tangan

1. Meida Cahyo Untoro, S.Kom., M.Kom.  
NIP. 19890518 201903 1 011

.....

Pengaji

Tanda Tangan

1. Martin Clinton Tosima Manullang, Ph.D.  
NIP. 199301092019031017
2. Andika Setiawan, S.Kom., M.Cs.  
NIP. 199111272022031007

.....

.....

Disahkan oleh,  
Koordinator Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Institut Teknologi Sumatera

Andika Setiawan, S.Kom., M.Cs.  
NIP 19911127 2022 03 1 007

## **HALAMAN PERNYATAAN ORISINALITAS**

**Tugas Akhir dengan judul “IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA SAWIT DENGAN PENDEKATAN NAÏVE BAYES MENGGUNAKAN OPTIMASI GENETIC ALGORITHM DAN PARTICLE SWARM OPTIMIZATION**

(Studi Kasus: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri)”  
**adalah karya saya sendiri, dan semua sumber baik yang dikutip  
maupun dirujuk telah saya nyatakan benar.**

**Nama : Natasya Ate Malem Bangun**

**NIM : 121140052**

**Tanda Tangan : .....**

**Tanggal : .....**

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS Sebagai**

civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Natasya Ate Malem Bangun

NIM : 121140052

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif**

*(Non-exclusive Royalty Free Right)* atas karya ilmiah saya yang berjudul:

**IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA SAWIT  
DENGAN PENDEKATAN NAÏVE BAYES MENGGUNAKAN  
OPTIMASI GENETIC ALGORITHM DAN PARTICLE SWARM  
OPTIMIZATION**

(Studi Kasus: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri) beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Lampung Selatan

Pada tanggal : 21 Mei 2025

Yang menyatakan

Natasya Ate Malem Bangun

## KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yesus Kristus yang merupakan sumber hikmat dan pengharapan sejati, atas kasih karunia-Nya yang senantiasa menyertai setiap langkah penulis hingga akhirnya dapat menyelesaikan Tugas Akhir berjudul “Identifikasi Penyakit pada Daun Bibit Kelapa Sawit dengan Pendekatan *Naïve Bayes* Menggunakan Optimasi *Genetic Algorithm* dan *Particle Swarm Optimization*”. Tugas Akhir ini bukan sekadar hasil dari kerja keras dan proses berpikir ilmiah, tetapi juga buah dari dukungan, doa, dan kehadiran banyak pihak yang secara langsung maupun tidak langsung telah memberikan kontribusi luar biasa. Oleh karena itu, dengan penuh hormat dan rasa terima kasih yang mendalam, penulis ingin menyampaikan apresiasi kepada:

1. Tuhan Yesus Kristus yang telah memberikan penulis kekuatan, kesehatan, dan hikmat dalam proses penyelesaian Tugas Akhir ini, tanpa kasih karunia-Nya, penulis tidak akan mampu menyelesaiannya ini.
2. Ibu Sipta Purba dan Bapak Sempakata Bangun selaku Orang Tua tercinta yang menjadi sumber kekuatan penulis dengan memberikan cinta kasih, dukungan, doa, dan harapan kepada penulis dalam menyelesaikan Tugas Akhir ini. Sehingga Tugas Akhir ini dapat terselesaikan dengan baik dan penulis persembahkan sebagai gelar yang akan disandang penulis.
3. Pihak PT Perkebunan Nusantara IV Regional 7 Kebun Bekri, yang telah memberikan akses, izin, serta arahan selama pelaksanaan penelitian lapangan untuk Tugas Akhir ini dapat berjalan dengan

lancar.

4. Bapak Andika Setiawan, S.Kom., M.Cs., Koordinator Program Studi Teknik Informatika ITERA sekaligus Dosen Penguji 2, atas bimbingan, kritik yang membangun, dan kepercayaan yang diberikan hingga Tugas Akhir ini dapat terselesaikan. Penulis juga berterima kasih atas kesempatan yang diberikan untuk mengikuti Magang Prodi dan Laboran 2025 di Prodi Teknik Informatika.
5. Bapak Meida Cahyo Untoro, S.Kom. M.Kom. selaku Koordinator TA dan Dosen Pembimbing yang telah memberikan masukan, saran dan kelancaran kepada penulis dalam menyelesaikan Tugas akhir ini.
6. Bapak Martin Clinton Tosima Manullang, Ph.D. selaku Dosen Penguji 1, atas arahan, ilmu, serta kesempatan berharga yang diberikan dalam mempertajam kemampuan ilmiah, memperdalam diskusi, dan memperluas wawasan riset yang sangat memperkaya proses penyelesaian Tugas Akhir ini.
7. Bapak Andre Febrianto, S.Kom., M.Eng., selaku Dosen Pembimbing Akademik (Dosen Wali) yang mendampingi penulis sejak awal perkuliahan hingga akhir studi, atas arahan dan dukungan yang sangat membantu penyelesaian studi di Program Studi Teknik Informatika ITERA.
8. Rekan-rekan Capstone Project, yaitu Benedictus Budhi Dharmawan dan Tobyanto Putra Mandiri, atas kebersamaan, dukungan, dan pemikiran berharga yang senantiasa dibagikan sepanjang perjalanan ini, sehingga setiap tantangan terasa lebih ringan.

9. Teman-teman kecil saya, yaitu Emmia Sindilosa Ginting, Chetrine br Milala, dan Brigita May Putri Rosari Sidabutar, yang hadir dalam suka dan duka. Terima kasih telah menjadi tempat berbagi cerita, bertumbuh bersama, dan menguatkan penulis hingga akhir masa studi dan proses penyelesaian Tugas Akhir ini.
10. Rekan-rekan yang memberi warna, semangat, dan keceriaan selama perjalanan Tugas Akhir ini, yaitu Steganonegai Team (Alvin-Marchell-Attar), Event Pejuang TA (Alvin, Atha, Attar, Marchell, Nopri, Pandu, Tara, Qaessar, dan Vania), FamilyMart Team, Stoberi Salkomsel, Proyek Kelapa Sawit Team, Ruang TA Team, Permata GBKP Runggun Bandar Lampung, dan BOS Team. Terima kasih atas setiap tawa, dukungan, kebersamaan yang menjadi penguatan luar biasa, kehadiran kalian benar-benar berarti bagi penulis.
11. Bapak Radhinka Bagaskara, S.Si.Kom., M.Si., M.Sc., selaku Koordinator Laboratorium Teknik 3 dan pengelola Ruang TA, atas fasilitas dan suasana kerja yang kondusif yang diberikan kepada penulis dalam menyelesaikan riset ini, bahkan hingga larut malam. Ucapan terima kasih juga penulis sampaikan kepada Ibu Leslie Anggraini, S.Kom., M.Cs. dan Ibu Miranti Verdiana, M.Si., atas dukungan, semangat, serta perhatian tulus yang senantiasa diberikan selama masa studi penulis di Program Studi Teknik Informatika ITERA.
12. Seluruh Dosen, Tenaga Pendidik, dan Staff Program Studi Teknik Informatika ITERA yang telah memberikan ilmu, inspirasi dan pengalaman selama perjalanan akademik penulis.

13. Seluruh teman seperjuangan Teknik Informatika ITERA, khususnya Angkatan 2021 (Binary) atas kebersamaan dan dukungannya selama masa studi penulis.
14. Diri penulis sendiri, atas dedikasi yang telah ditunjukkan selama proses perkuliahan hingga penyusunan Tugas Akhir, dan kesabaran dalam menghadapi berbagai tantangan yang muncul.
15. Untuk seluruh pihak yang tidak dapat disebutkan satu per satu, terima kasih atas segala bentuk dukungan, baik langsung maupun tidak langsung yang telah menguatkan penulis selama menempuh studi dan menyelesaikan Tugas Akhir ini.

Akhir kata, penulis berharap semoga tugas akhir ini tidak hanya menjadi pencapaian pribadi, tetapi juga memberikan kontribusi nyata bagi pengembangan ilmu pengetahuan, khususnya dalam penerapan *Machine Learning* untuk sektor pertanian di Indonesia. Amin.

## RINGKASAN

### IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA SAWIT DENGAN PENDEKATAN NAÏVE BAYES MENGGUNAKAN OPTIMASI GENETIC ALGORITHM DAN PARTICLE SWARM OPTIMIZATION

(Studi Kasus: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri)

Natasya Ate Malem Bangun

Kelapa sawit termasuk komoditas penting di Indonesia karena memberikan kontribusi ekonomi yang signifikan. Namun, dalam proses budidaya, tanaman ini rentan terhadap serangan penyakit yang dapat menghambat pertumbuhan dan menurunkan produktivitas, terutama pada fase pembibitan. Terdapat lima kategori utama penyakit yang sering menyerang daun bibit kelapa sawit, yaitu bercak daun, daun berkerut, daun mengulung, daun menguning, dan daun berputar. Saat ini, penanganan penyakit tersebut masih mengandalkan pendekatan manual melalui identifikasi gejala visual pada daun. Namun, metode ini memiliki beberapa keterbatasan, di antaranya membutuhkan waktu dan sumber daya manusia yang besar serta berpotensi menimbulkan kesalahan dalam diagnosis.

Berdasarkan permasalahan tersebut, dilakukan penelitian untuk mengidentifikasi penyakit pada daun bibit kelapa sawit menggunakan pendekatan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) yang kemudian diklasifikasikan dengan algoritma *Machine Learning Naïve Bayes*. Penelitian ini bertujuan untuk membangun model klasifikasi yang efisien secara komputasi dan mampu mengenali lima kelas penyakit tersebut. Dataset yang digunakan berasal dari PT Perkebunan Nusantara

IV Regional 7 Bekri dan mencakup lima kelas penyakit daun bintang kelapa sawit. Untuk meningkatkan performa model, diterapkan dua metode optimasi, yaitu *Genetic Algorithm* dan *Particle Swarm Optimization*, guna mencari konfigurasi parameter terbaik.

Penelitian ini menggunakan total 340 citra dengan pembagian data latih sebesar 80% dan data uji sebesar 20%. Setelah fitur citra diekstraksi menggunakan GLCM, hasilnya digunakan sebagai input bagi algoritma *Naïve Bayes*. Optimasi dilakukan untuk menemukan kombinasi parameter terbaik dengan memanfaatkan *Genetic Algorithm* dan *Particle Swarm Optimization*. Hasil pengujian menunjukkan bahwa model *Naïve Bayes* yang dihasilkan memiliki akurasi validasi tertinggi sebesar 56%. Selain itu, evaluasi menggunakan *Confusion Matrix* menunjukkan bahwa nilai rata-rata *precision*, *recall*, dan *F1-score* berada pada kisaran 55% hingga 56% untuk seluruh kelas penyakit. Meskipun performa model ini masih tergolong rendah, pendekatan yang digunakan dapat menjadi dasar awal untuk pengembangan sistem klasifikasi penyakit daun kelapa sawit yang lebih canggih dan akurat di masa mendatang.

## ABSTRAK

# IDENTIFIKASI PENYAKIT PADA DAUN BIBIT KELAPA SAWIT DENGAN PENDEKATAN NAÏVE BAYES MENGGUNAKAN OPTIMASI GENETIC ALGORITHM DAN PARTICLE SWARM OPTIMIZATION

(Studi Kasus: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri)

Natasya Ate Malem Bangun

Tanaman kelapa sawit merupakan komoditas unggulan Indonesia yang memiliki peran penting dalam perekonomian nasional. Namun pada fase pertumbuhan, bibit kelapa sawit rentan terhadap serangan berbagai penyakit, terutama pada daun yang dapat menurunkan kualitas dan produktivitas bibit. Oleh karena itu, deteksi dini terhadap penyakit berperan penting dalam menjamin kualitas bibit. Penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit pada daun bibit kelapa sawit berbasis pengolahan citra digital dan machine learning. Ekstraksi fitur tekstur dan nilai RGB dari daun dilakukan dengan metode Gray Level Co-occurrence Matrix (GLCM). Fitur tersebut kemudian diklasifikasi dengan algoritma Naïve Bayes yang dioptimasi dengan dua metode, yakni Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi pelatihan dari model. Dataset yang digunakan mencakup 5 kelas penyakit. Hasil pengujian menunjukkan bahwa model terbaik menghasilkan akurasi sebesar 56% dengan rata-rata nilai presisi, recall, dan f1-score berkisar antara 55% hingga 56%. Pendekatan ini menunjukkan efisiensi komputasi yang baik dan dapat dijadikan baseline dalam pengembangan model klasifikasi yang lebih akurat di masa mendatang.

**Kata Kunci:** Identifikasi Penyakit, Kelapa Sawit, Naïve Bayes, Genetic Algorithm, Particle Swarm Optimization

## ABSTRACT

### IDENTIFICATION OF DISEASES IN OIL PALM SEEDLING LEAVES USING A NAÏVE BAYES CLASSIFIER OPTIMIZED BY GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

(Case Study: PT Perkebunan Nusantara IV Regional 7 Kebun Bekri)

Natasya Ate Malem Bangun

Palm oil is a leading commodity in Indonesia with a significant role in the national economy. However, during the growth phase, palm oil seedlings are vulnerable to various diseases, particularly on the leaves, which can reduce seedling quality and productivity. Therefore, early detection of leaf diseases is essential to ensure seedling quality. This study aims to develop a classification model for leaf diseases detection in palm oil seedlings based on digital image processing and machine learning. Texture and RGB features were extracted using the Gray Level Co-occurrence Matrix (GLCM) method. These features were then classified using the Naïve Bayes algorithm, optimized with 2 techniques: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), to improve the model's training accuracy. The dataset used includes 5 classes of diseases. Experimental results show that the best model achieved an accuracy of 56%, with average precision, recall, and f1-score values ranging from 55% to 56%. This approach demonstrates good computational efficiency and can serve as a baseline for the development of more accurate classification models in the future.

**Keywords:** Disease Identification, Palm oil, Naive Bayes, Genetic Algorithm, Particle Swarm Optimization

## DAFTAR ISI

<b>Lembar Pengesahan .....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>Halaman Pernyataan Orisinalitas .....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN ORISINALITAS .....</b>	<b>iii</b>
<b>HALAMAN PERSETUJUAN PUBLIKASI .....</b>	<b>iv</b>
<b>Kata Pengantar .....</b>	<b>v</b>
<b>Ringkasan .....</b>	<b>ix</b>
<b>Abstrak .....</b>	<b>xi</b>
<b>Abstract .....</b>	<b>xii</b>
<b>Daftar Isi .....</b>	<b>xiii</b>
<b>Daftar Tabel .....</b>	<b>xvii</b>
<b>Daftar Gambar .....</b>	<b>xviii</b>
<b>Daftar Rumus .....</b>	<b>xx</b>
<b>BAB I Pendahuluan .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan masalah .....	4
1.3 Tujuan .....	5
1.4 Batasan Masalah .....	5
1.5 Manfaat Penelitian .....	6

1.6 Sistematika Penulisan.....	7
1.6.1 Bab I.....	7
1.6.2 Bab II.....	7
1.6.3 Bab III.....	7
1.6.4 Bab IV.....	8
1.6.5 Bab V.....	8
<b>BAB II Tinjauan Pustaka.....</b>	<b>9</b>
2.1 Tinjauan Pustaka.....	9
2.2 Dasar Teori.....	17
2.2.1 Kelapa Sawit .....	18
2.2.2 <i>Naïve Bayes</i> .....	24
2.2.3 Genetic Algorithm .....	28
2.2.4 Particle Swarm Optimization .....	31
2.2.5 Citra Digital .....	33
2.2.6 Identifikasi .....	37
2.2.7 Penyakit .....	37
2.2.8 Ekstraksi Fitur .....	38
2.2.9 Machine Learning .....	43
2.2.10 Segmentasi Citra .....	44
2.2.11 Hyperparameter.....	45
2.2.12 Confusion Matrix .....	46
2.2.13 Roboflow .....	47
2.2.14 StreamLit .....	49
<b>BAB III Analisis dan Perancangan .....</b>	<b>50</b>
3.1 Alur Penelitian.....	50

3.2 Penjabaran Langkah Penelitian .....	51
3.2.1 Identifikasi Masalah .....	51
3.2.2 Studi Literatur .....	52
3.2.3 Akuisisi Dataset Citra .....	53
3.2.4 <i>Preprocessing</i> .....	54
3.2.5 Augmentasi Citra .....	55
3.2.6 Segmentasi Citra .....	57
3.2.7 Ekstraksi Fitur .....	58
3.2.8 Pembagian Dataset .....	59
3.2.9 Pembuatan Model .....	59
3.2.10 Hasil .....	60
3.2.11 Evaluasi .....	60
3.2.12 Deployment .....	61
3.3 Alat dan Bahan Tugas Akhir .....	61
3.3.1 Alat .....	61
3.3.2 Bahan .....	62
3.4 Metode Tugas Akhir .....	63
3.5 Ilustrasi Perhitungan Metode .....	65
3.5.1 Ekstraksi Fitur Warna .....	65
3.5.2 Ekstraksi Fitur Tekstur .....	68
<b>BAB IV Hasil dan Pembahasan .....</b>	<b>80</b>
4.1 Hasil Penelitian .....	80
4.1.1 Akuisisi Dataset .....	80
4.1.2 Persiapan Dataset .....	80
4.1.3 Augmentasi Dataset .....	81

4.1.4	Ekstraksi Fitur .....	81
4.1.5	Memproses Gambar Daun menjadi Kerangka Data .	84
4.1.6	Pembagian Dataset .....	85
4.1.7	Pembuatan Model .....	86
4.1.8	Original Naive Bayes .....	86
4.1.9	GA Optimized Naive Bayes .....	86
4.1.10	Evaluasi.....	90
<b>BAB V</b>	<b>Kesimpulan dan Saran .....</b>	<b>95</b>
5.1	Kesimpulan .....	95
5.2	Saran .....	96
<b>Daftar Pustaka .....</b>	<b>98</b>	
<b>Lampiran .....</b>	<b>110</b>	

## **DAFTAR TABEL**

Tabel 2.1	Tinjauan Pustaka .....	12
Tabel 3.2	Foto Dataset Daun Bibit Kelapa Sawit.....	63
Tabel 3.3	Tabel nilai RGB dalam format [R, G, B] .....	65
Tabel 3.4	Tabel nilai RGB dari citra 4x4 .....	68
Tabel 3.5	Tabel nilai grayscale dari citra RGB .....	70
Tabel 3.6	Kuantisasi Nilai Pixel Grayscale .....	70
Tabel 3.7	Matriks <i>Co-occurrence</i> .....	71
Tabel 3.8	Matriks Transpose.....	71
Tabel 3.9	Matriks Simetris .....	71
Tabel 3.10	Matriks Normalisasi.....	72
Tabel 3.11	Tabel Hasil Normalisasi.....	72

## **DAFTAR GAMBAR**

Gambar 2.1	<i>Elaeis Guineensis Jacq</i> atau Kelapa Sawit .....	18
Gambar 2.2	Bercak Daun .....	20
Gambar 2.3	Daun Berputar .....	21
Gambar 2.4	Daun Berkerut .....	22
Gambar 2.5	Daun Menguning .....	22
Gambar 2.6	Daun Menggulung .....	23
Gambar 2.7	RGB .....	34
Gambar 2.8	Grayscale .....	35
Gambar 2.9	<i>Gray Level Co-occurrence Matrix (GLCM)</i> ....	41
Gambar 2.10	Machine Learning .....	43
Gambar 2.11	Confusion Matrix .....	46
Gambar 3.12	Diagram Alur Penelitian .....	50
Gambar 3.13	Ilustrasi Pengambilan Dataset Daun Bibit Kelapa Sawit .....	54
Gambar 3.14	Ilustrasi Rotasi Citra .....	56
Gambar 3.15	Ilustrasi Flipping Citra .....	57
Gambar 3.16	Ilustrasi Perubahan Kecerahan Citra .....	57
Gambar 3.17	Ilustrasi Segmentasi Citra .....	58
Gambar 4.18	Contoh hasil ekstraksi fitur warna RGB .....	82
Gambar 4.19	Contoh hasil ekstraksi tekstur .....	83
Gambar 4.20	Contoh hasil pemrosesan gambar daun menjadi kerangka data .....	84

Gambar 4.21	Contoh hasil pembagian dataset menjadi data training dan testing.....	85
Gambar 4.22	Contoh hasil ekstraksi tekstur .....	86
Gambar 4.23	Contoh hasil training model dengan GA .....	88
Gambar 4.24	Contoh training model dengan PSO.....	89
Gambar 4.25	Contoh hasil evaluasi model .....	90
Gambar 4.26	Contoh hasil confusion matrix .....	91
Gambar 4.27	Contoh hasil konversi citra daun ke grayscale ..	91
Gambar 4.28	Contoh hasil pemrosesan gambar daun menjadi kerangka data.....	92
Gambar 4.29	Contoh hasil rata-rata RGB dari citra daun.....	92
Gambar 4.30	Contoh hasil rata rata ekstraksi fitur tekstur ....	93
Gambar 4.31	Contoh hasil classification metrics .....	94

## **DAFTAR RUMUS**

Rumus 2.1	Persamaan Klasifikasi <i>Naïve Bayes</i> .....	24
Rumus 2.2	Persamaan Klasifikasi Sederhana <i>Naïve Bayes</i> ....	24
Rumus 2.3	Persamaan Independen Fitur <i>Naïve Bayes</i> .....	25
Rumus 2.4	Persamaan Teorema Bayes .....	25
Rumus 2.5	Nilai Probabilitas Ditribusi Normal .....	26
Rumus 2.6	Nilai Rata-rata (mean) .....	27
Rumus 2.7	Nilai Standar Deviasi ( $\sigma$ ) .....	27
Rumus 2.8	Nilai Probabilitas Posterior.....	28
Rumus 2.9	Nilai Kelas Hasil Klasifikasi .....	28
Rumus 2.10	Nilai Probabilitas Seleksi .....	30
Rumus 2.11	Nilai Offspring Hasil Crossover.....	30
Rumus 2.12	Nilai Individu Setelah Mutasi.....	31
Rumus 2.13	Nilai Kecepatan Partikel .....	33
Rumus 2.14	Nilai Posisi Partikel .....	33
Rumus 2.15	Nilai Grayscale .....	36
Rumus 2.16	Rata-rata fitur warna C.....	39
Rumus 2.17	Standar deviasi fitur warna C .....	39
Rumus 2.18	Normalisasi komponen R .....	40
Rumus 2.19	Normalisasi komponen G .....	40
Rumus 2.20	Normalisasi komponen B .....	40
Rumus 2.21	Nilai GLCM .....	41
Rumus 2.22	Nilai Kontras .....	42
Rumus 2.23	Nilai Korelasi.....	42

Rumus 2.24 Nilai Energi .....	42
Rumus 2.25 Nilai Homogenitas .....	43
Rumus 2.26 Nilai Akurasi .....	47
Rumus 2.27 Nilai Presisi .....	47
Rumus 2.28 Nilai Recall.....	47

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Tanaman kelapa sawit (*Elaeis Guineensis Jacq*) merupakan salah satu pilar utama komoditas perkebunan Indonesia yang terkenal menghasilkan minyak nabati dalam jumlah besar secara efisien dan ekonomis, baik untuk pasar domestik maupun internasional [1]. Dengan kondisi iklim tropis Indonesia membuat lingkungan ideal bagi pertumbuhan kelapa sawit, membuat negara menjadi salah satu produsen minyak kelapa sawit di dunia dengan sebagian besar produksi ditujukan untuk ekspor dan sisanya memenuhi kebutuhan dalam negeri [2][3].

Namun, dibalik potensinya yang besar, proses pembibitan dan budidaya kelapa sawit tetap menghadapi berbagai tantangan, khususnya pada fase awal pertumbuhan. Pada fase awal pertumbuhan bibit kelapa sawit, terutama dalam rentang usia 1 minggu hingga 3 bulan sangat rentan terhadap serangan penyakit daun [4]. Beberapa gejala umum yang sering muncul adalah daun menguning akibat infeksi jamur *Ganoderma Boninense*, bercak-bercak akibat jamur *Curvularia*, daun yang menggulung, berputar dan berkerut akibat serangan hama atau faktor genetik [5][6][7][8].

Penyakit-penyakit ini dapat memperlambat pertumbuhan, menghambat fotosintesis bahkan berdampak negatif terhadap kualitas bibit yang akan ditanam di lahan produksi [8]. Oleh karena itu, deteksi dini terhadap penyakit pada bibit kelapa sawit sangat penting untuk

memastikan pertumbuhan optimal dan kualitas bibit yang baik. Namun, proses identifikasi penyakit ini sering kali memerlukan keahlian khusus dan waktu yang cukup lama, sehingga diperlukan pendekatan yang lebih efisien dan akurat untuk mendeteksi penyakit pada bibit daun kelapa sawit [9].

Salah satu lokasi yang mengalami tantangan tersebut adalah PT Perkebunan Nusantara (PTPN) IV Regional 7 Kebun Bekri di Provinsi Lampung menunjukkan bahwa serangan penyakit pada daun bibit kelapa sawit menjadi salah satu kendala signifikan yang mempengaruhi kualitas dan kuantitas bibit siap tanam. Keterlambatan penanganan penyakit dapat mengurangi persentase bibit layak tanam yang pada akhirnya mempengaruhi produktivitas lahan secara keseluruhan.

Untuk menjawab tantangan ini, penelitian bertujuan untuk mengidentifikasi penyakit pada bibit kelapa sawit menggunakan teknologi pengolahan citra dan pembelajaran mesin. Pengolahan citra digunakan untuk mengekstraksi fitur penting seperti warna (RGB), tekstur (*Gray Level Co-Occurrence Matrix* atau GLCM) dan bentuk [10].

Proses klasifikasi mengelompokkan objek berdasarkan fitur-fitur atau nilai atribut unik dari setiap objeknya. Pada proses ini, klasifikasi jenis penyakit pada daun kelapa sawit menerapkan pengolahan citra digital yang mengolah dan menganalisis sebuah citra dengan inputan data citra untuk menghasilkan informasi berupa citra. Penerapan pengolahan citra digital digunakan sebagai pembeda antara struktur bentuk, tekstur dan warna pada daun kelapa sawit yang sehat dan yang terserang penyakit, sekaligus menjadi dasar untuk parameter dalam penelitian [11]. Daun

kelapa sawit akan diteliti lebih lanjut dengan fitur tekstur *Gray Level Co-Occurrence Matrix* (GLCM) untuk menemukan perbedaan pada struktur daun, kemudian hasilnya akan diekstraksi dengan metode RGB untuk mengidentifikasi terkait kondisi daun melalui perbedaan warnanya [12].

Metode klasifikasi *Naïve Bayes* telah menunjukkan potensi signifikan dalam berbagai penelitian terkait identifikasi berbasis citra [13]. Misalnya, pada penelitian Septian sebelumnya menunjukkan bahwa metode klasifikasi seperti *Naïve Bayes* mampu bekerja cukup baik dalam mengenali pola berdasarkan fitur visual dengan teknik *10-Fold Cross Validation* yang berhasil mencapai akurasi hingga 86,06% dalam membedakan jamur beracun dan tidak beracun [14].

Lain halnya dengan penelitian oleh penelitian Ardi dkk yang menerapkan *Naïve Bayes* untuk mendeteksi penyakit pada daun tanaman tomat berdasarkan fitur warna dan bentuk dengan GLCM dan *Naïve Bayes* menunjukkan tingkat akurasi sebesar 80% dari 15 dataset yang digunakan [15]. Sementara itu, pada penelitian Felicia membandingkan efektivitas *Naïve Bayes* dengan KNN untuk mengidentifikasi jenis buah apel menggunakan fitur tekstur *Local Binary Pattern* (*LBP*) dan warna HSV. Penelitian tersebut menunjukkan bahwa *Naïve Bayes* memberikan akurasi lebih tinggi, berkisar 97% dibandingkan metode *K-Nearest Neighbor* (KKN) berkisar 82% [16].

Berdasarkan studi-studi tersebut [14][15][16], metode *Naïve Bayes* terbukti efektif untuk tugas klasifikasi yang melibatkan fitur warna dan tekstur dan mampu meminimalkan nilai *error* pada dataset yang cukup besar. Namun, akurasi metode ini masih dapat ditingkatkan

dengan pendekatan optimasi. Oleh karena itu, dalam penelitian ini, digunakan integrasi metode *Naïve Bayes* dengan teknik optimasi seperti *Genetic Algorithm (GA)* dan *Particle Swarm Optimization (PSO)* untuk meningkatkan performa klasifikasi.

*Genetic Algorithm (GA)* terinspirasi oleh proses evolusi biologis [17][18], sedangkan Teknik *Particle Swarm Optimization (PSO)* terinspirasi oleh perilaku sosial burung dan ikan. Keduanya merupakan metode optimasi yang telah terbukti efektif dalam meningkatkan akurasi klasifikasi, sehingga diharapkan dapat meningkatkan akurasi deteksi penyakit pada bibit kelapa sawit.

Optimasi ini akan membantu dalam menyempurnakan parameter, seperti *Prior Probability* dan *Likelihood Estimation*, sehingga model dapat beradaptasi lebih baik terhadap variasi data [19]. Dalam evaluasi kinerja model digunakan juga metode *Confusion Matrix* yang memungkinkan analisis lebih rinci terhadap *presisi*, *recall*, dan akurasi hasil klasifikasi [20].

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka rumusan masalah yang terkait, antara lain:

1. Bagaimana proses implementasi dan pengembangan model *Naïve Bayes* dengan *Particle Swarm Optimization (PSO)* dan *Genetic Algorithm (GA)* dalam mengidentifikasi penyakit pada daun bibit kelapa sawit?
2. Apakah model *Naïve Bayes* dengan *Particle Swarm Optimization (PSO)* dan *Genetic Algorithm (GA)* terbukti efektif dalam

- mengidentifikasi penyakit pada daun bibit kelapa sawit?
3. Faktor-faktor apa saja yang mempengaruhi kinerja model *Naïve Bayes* dengan *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA) dalam mengidentifikasi penyakit pada daun bibit kelapa sawit?

### 1.3 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini, antara lain:

1. Mengembangkan dan mengimplementasikan model klasifikasi penyakit pada daun bibit kelapa sawit menggunakan algoritma *Naïve Bayes* dengan *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA).
2. Mengevaluasi efektivitas model *Naïve Bayes* dalam mengidentifikasi penyakit pada daun bibit kelapa sawit.
3. Menganalisis pengaruh fitur-fitur penting yang memengaruhi kinerja model *Naïve Bayes* dengan *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA) dalam mengidentifikasi penyakit pada daun bibit kelapa sawit.

### 1.4 Batasan Masalah

Berdasarkan rumusan masalah diatas, diperlukan beberapa batasan masalah agar menghindari perluasan masalah, antara lain:

1. Penelitian ini hanya berfokus pada identifikasi penyakit pada daun bibit kelapa sawit menggunakan metode *Naïve Bayes* dengan optimasi *Particle Swarm Optimization* (PSO) dan *Genetic*

*Algorithm* (GA).

2. Penelitian ini hanya menggunakan dataset citra daun bibit kelapa sawit yang diambil dari PT Perkebunan Nusantara IV Regional 7 Kebun Bekri.
3. Penelitian ini hanya difokuskan pada identifikasi penyakit yang menyerang daun bibit kelapa sawit dalam rentang usia 1 minggu hingga 3 bulan.
4. Data yang digunakan berupa citra digital daun kelapa sawit, dengan fitur utama yang dianalisis adalah warna (RGB) dan tekstur (*Gray Level Co-Occurrence Matrix (GLCM)*).
5. Model klasifikasi yang digunakan adalah *Naïve Bayes* dengan menggunakan *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA), tanpa membandingkan dengan algoritma klasifikasi lainnya.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat, antara lain:

1. Memudahkan petani perkebunan kelapa sawit dan pihak terkait dalam mengidentifikasi penyakit pada daun bibit kelapa sawit secara cepat dan akurat.
2. Memberikan wawasan dan kontribusi dalam pengembangan teknologi identifikasi penyakit pada daun bibit kelapa sawit menggunakan metode *Naïve Bayes* dengan optimasi *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA).
3. Menyajikan evaluasi tentang kinerja metode *Naïve Bayes* dengan optimasi *Particle Swarm Optimization* (PSO) dan *Genetic*

*Algorithm* (GA) dalam klasifikasi penyakit pada daun bibit kelapa sawit.

4. Menjadi salah satu pembuktian akurasi model *Naïve Bayes* dengan optimasi *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA) pada penyakit daun bibit kelapa sawit.
5. Menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan identifikasi penyakit pada tanaman menggunakan teknologi pengolahan citra dan pembelajaran mesin.

## **1.6 Sistematika Penulisan**

Pada penulisan Tugas Akhir ini, penulis menyusun sistematika penulisan sebagai berikut:

### **1.6.1 Bab I**

Bab I Pendahuluan membahas mengenai latar belakang permasalahan, rumusan masalah, tujuan penelitian, batasan-batasan masalah penelitian, manfaat penelitian, dan sistematika penulisan.

### **1.6.2 Bab II**

Bab II Tinjauan Pustaka membahas mengenai kajian tinjauan pustaka yang menjelaskan tentang pengertian, teori, dan penelitian terdahulu yang berkaitan dengan penelitian ini.

### **1.6.3 Bab III**

Bab III Analisis dan Perancangan membahas mengenai metodologi penelitian yang menjelaskan tentang langkah-langkah yang dilakukan dalam penelitian ini, mulai dari alur penelitian, pengumpulan data,

pengolahan data, hingga analisis hasil.

#### **1.6.4 Bab IV**

Bab IV Hasil dan Pembahasan membahas menganai hasil dan pembahasan yang menjelaskan tentang hasil penelitian yang diperoleh dengan alur yang tertera.

#### **1.6.5 Bab V**

Bab V Kesimpulan dan Saran berisi tentang kesimpulan dan saran yang menjelaskan tentang kesimpulan dari penelitian yang dilakukan dan saran-saran untuk penelitian selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Penelitian pada bidang klasifikasi citra berbasis *Machine Learning* telah banyak dilakukan dalam berbagai objek. Namun, implementasi model klasifikasi penyakit secara spesifik pada daun bibit kelapa sawit masih tergolong terbatas, baik dari segi jumlah kajian maupun pendekatan optimasi yang digunakan. Meskipun demikian, terdapat beberapa studi sebelumnya yang menggunakan algoritma *Naïve Bayes* dan pendekatan optimasi seperti *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO) dengan objek dan fitur yang berbeda.

Salah satu penelitian yang relevan dilakukan oleh Septian dkk yang memanfaatkan metode *Naïve Bayes* untuk mengklasifikasikan jenis jamur berdasarkan dataset yang terdiri dari 8124 citra. Proses pembelajaran yang digunakan adalah teknik validasi silang *10-Fold Cross Validation* untuk membagi data latih dan uji. Penelitian ini menunjukkan bahwa metode *Naïve Bayes* mampu mencapai akurasi sebesar 86,06% dalam membedakan jamur yang layak konsumsi dan yang beracun. Hal ini menunjukkan bahwa algoritma ini memiliki fleksibilitas tinggi dalam menangani klasifikasi data visual pada objek biologis [14].

Penelitian lain yang dilakukan oleh Ardi dkk yang menerapkan *Naïve Bayes* untuk mendeteksi penyakit pada daun tanaman tomat berdasarkan fitur warna dan bentuk dengan menggunakan *Gray Level Co-occurrence Matrix (GLCM)*. Penelitian ini menunjukkan bahwa metode *Naïve*

*Bayes* dapat digunakan untuk mengidentifikasi penyakit pada tanaman dengan akurasi yang cukup baik, meskipun dataset yang digunakan lebih kecil dibandingkan dengan penelitian sebelumnya. Dalam penelitian ini, akurasi yang diperoleh mencapai 80% dari 15 dataset yang digunakan [15].

Penelitian oleh Felicia dkk yang membandingkan performa metode *Naïve Bayes* dengan *K-Nearest Neighbor (KNN)* dalam mengidentifikasi jenis buah apel. Pada proses ekstraksi menggabungkan fitur tekstur dari *Local Binary Pattern (LBP)* dan warna dari model HSV. Dataset terdiri dari 100 citra yang terbagi menjadi 80 data latih, 10 data uji, dan 10 data evaluasi. Hasilnya menunjukkan bahwa *Naïve Bayes* memperoleh akurasi sebesar 97%, lebih tinggi dibandingkan KNN yang mencapai 82%, sehingga dapat dikatakan *Naïve Bayes* tetap relevan dan kompetitif dalam domain klasifikasi citra berbasis fitur warna dan tekstur [16].

Penelitian lainnya dilakukan Sherin dkk yang membuktikan bahwa optimasi *Particle Swarm Optimization (PSO)* efektif digunakan dalam memprediksi hasil panen tanaman cabai rawit setan di Kota Pagar Alam dengan data yang digunakan mencakup curah hujan, jenis hama, jenis pupuk, dan luas lahan. Terbukti dengan mengintegrasikan metode *Naïve Bayes* dan *Particle Swarm Optimization (PSO)* yang menghasilkan akurasi *Naïve Bayes* meningkat dari 75% menjadi 92% setelah dioptimasi dengan PSO [21].

Penelitian lain dilakukan oleh Dadang dkk dengan menerapkan *Naïve Bayes* untuk mengklasifikasikan 5 jenis bunga kantong semar berdasarkan citra digital menggunakan fitur warna RGB. Penelitian ini

juganya memanfaatkan teknik data augmentasi dan metode *holdout* untuk validasi. Dari total 1750 data latih dan 350 data uji, metode *Naïve Bayes* berhasil mencapai akurasi sebesar 97,06%, menunjukkan bahwa algoritma ini efektif untuk klasifikasi citra tanaman langka sekalipun [22].

Penelitian lainnya oleh Evy yang bertujuan meningkatkan akurasi metode *Naïve Bayes* dalam klasifikasi jenis bakteri dengan menambahkan algoritma genetika. Data terdiri dari 336 atribut numerik, termasuk fitur-fitur seperti Mcg, gvh, Lip, dan alm. Hasil penelitian menunjukkan bahwa penggunaan algoritma genetika mampu memperbaiki konfigurasi dan pemodelan pada *Naïve Bayes*, sehingga akurasi meningkat dari 80,93% menjadi 81,19% pada klasifikasi 8 kelas protein bakteri [23].

Tabel 2.1 Tinjauan Pustaka

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
1	Implementasi Metode <i>Naïve Bayes Classifier</i> untuk Identifikasi Jenis Jamur (Septian Arie Prayoga, Ismasari Nawangsih, Tri Ngudi Wiyatno)	Menentukan akurasi metode <i>Naïve Bayes Classifier</i> dalam klasifikasi jamur konsumsi dan beracun serta perancangan aplikasinya	<i>Naïve Bayes</i> dengan <i>10-Fold Cross Validation</i>	Dataset 8124 citra jamur	Akurasi 86,06%	Penelitian ini berbeda dengan Prayoga dkk. pada objek kajian, metode fitur, dan optimasi. Prayoga dkk. mengklasifikasi jenis jamur menggunakan <i>Naïve Bayes</i> tanpa segmentasi atau fitur warna/tekstur, sedangkan penelitian ini fokus pada penyakit daun kelapa sawit dengan fitur warna RGB, tekstur GLCM, segmentasi K-Means, serta optimasi <i>GA</i> dan <i>PSO</i> . Hasilnya mencapai akurasi 59% dengan <i>precision, recall</i> , dan <i>F1-score</i> 58–59%.

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
2	Identifikasi Penyakit Tanaman Tomat Berdasarkan Citra Penyakit Menggunakan Metode GLCM dan <i>Naïve Bayes Classifier</i> (Ardi Nainggolan dkk.)	Meng-identifikasi penyakit tanaman tomat berdasarkan citra daun, buah, dan batang	GLCM (ekstraksi fitur) dan <i>Naïve Bayes Classifier</i> (klasifikasi penyakit)	15 citra uji dengan ukuran sampel 3x3 piksel (manual)	Akurasi 80%	Perbedaan penelitian ini dengan Ardi dkk. terletak pada objek, dataset, dan metode. Ardi dkk. fokus pada penyakit daun tomat (15 citra) dengan GLCM dan <i>Naïve Bayes</i> , akurasi 80%. Penelitian ini mendekripsi penyakit daun kelapa sawit (5 kelas) dengan GLCM dan <i>Naïve Bayes</i> yang dioptimasi <i>GA</i> dan <i>PSO</i> , akurasi 59% dan <i>precision</i> , <i>recall</i> , <i>F1-score</i> 58–59%, menekankan optimasi dan dataset lebih kompleks.

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
3	Perbandingan Klasifikasi <i>Naïve Bayes</i> dan KNN untuk Meng-identifikasi Jenis Buah Apel dengan Ekstraksi Ciri LBP dan HSV (Felicia Febriana dkk.)	Membandingkan efektivitas klasifikasi <i>Naïve Bayes</i> dan KNN, untuk Meng-identifikasi Jenis Buah Apel dengan Ekstraksi Ciri LBP dan HSV (Felicia Febriana dkk.)	Klasifikasi menggunakan <i>Naïve Bayes</i> dan KNN, dengan ekstraksi ciri HSV dan LBP	100 citra apel (80 data latih, 10 evaluasi, 10 uji)	Akurasi <i>Naïve Bayes</i> (HSV), sebesar 97%, KNN (HSV) sebesar 82%	Perbedaan penelitian ini dengan Felicia Febriana dkk. (2021) terletak pada objek, metode, fitur, dan hasil. Penelitian Felicia Febriana dkk. fokus pada identifikasi jenis buah apel dengan <i>Naïve Bayes</i> dan KNN menggunakan fitur LBP (tekstur) dan HSV (warna), mencapai akurasi 90–97%. Penelitian ini fokus pada penyakit daun bibit kelapa sawit menggunakan <i>Naïve Bayes</i> dengan fitur tekstur GLCM dan optimasi GA serta PSO, dengan akurasi tertinggi 59% dan <i>precision</i> , <i>recall</i> , <i>F1-score</i> rata-rata 58–59%.

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
4	Integrasi <i>Particle Swarm Optimization</i> dengan <i>Naïve Bayes</i> untuk Prediksi Tanaman Cabai (Sherin Junisthia dkk.)	Meningkatkan akurasi prediksi panen cabai rawit dengan <i>Naïve Bayes</i>	Kombinasi <i>Naïve Bayes</i> dan PSO, dalam kerangka CRISP-DM waktu dan parameter <i>Naïve Bayes</i>	Data pertanian Kota Pagar Alam dengan 8 variabel fitur	Akurasi tanpa PSO sebesar 75%, dengan PSO sebesar 92%	Perbedaan penelitian ini dengan penelitian cabai rawit terletak pada objek, fitur, dan hasil. Penelitian cabai rawit menggunakan data numerik untuk prediksi hasil panen dengan <i>Naïve Bayes</i> dioptimasi <i>PSO</i> , meningkatkan akurasi dari 75% menjadi 92%. Penelitian ini mendeteksi penyakit daun bibit kelapa sawit berbasis citra dengan fitur tekstur GLCM, diklasifikasikan <i>Naïve Bayes</i> yang dioptimasi <i>GA</i> dan <i>PSO</i> , dengan akurasi 59% dan <i>precision, recall, F1-score</i> 58–59%.

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
5	Optimasi Klasifikasi Bunga Kantong Semar Menggunakan <i>Naïve Bayes</i> , Data Augmentasi dan PSO (Dadang Iskandar Mulyana dkk.)	Klasifikasi lima jenis bunga kantong semar dengan pembagian data semar dengan mengakurasi tinggi melalui augmentasi data	<i>Naïve Bayes</i> dengan citra pembagian data menggunakan metode <i>Holdout</i> (80%-20%)	2100 citra (1750 latih, 350 uji)	Akurasi pelatihan pengujian sebesar 97,60%	Perbedaan penelitian ini dengan penelitian kantong semar terletak pada objek, fitur, optimasi, dan hasil. Penelitian kantong semar mengklasifikasi 5 jenis citra dengan fitur RGB dan <i>ImageNet</i> , menggunakan <i>Naïve Bayes</i> tanpa optimasi, mencapai akurasi 97,60%. Penelitian ini mengklasifikasi 5 kelas penyakit daun kelapa sawit dengan fitur tekstur GLCM, <i>Naïve Bayes</i> dioptimasi <i>GA</i> dan <i>PSO</i> , dengan akurasi 59% dan <i>precision</i> , <i>recall</i> , <i>F1-score</i> 58–59%.

No	Judul	Masalah	Metode	Data	Hasil	Perbandingan
6	Peningkatan Algoritma <i>Naïve Bayes</i> Menggunakan Algoritma Genetika pada Klasifikasi Bakteri (Evy Priyanti)	Mengatasi keterbatasan akurasi klasifikasi bakteri dengan optimasi parameter <i>Naïve Bayes</i>	<i>Naïve Bayes</i> dan <i>Genetic Algorithm</i> untuk tuning parameter	336 atribut numerik untuk 8 kelas protein bakteri	Akurasi awal sebesar 80,93%, meningkat menjadi 81,19%	Perbedaan penelitian ini dengan penelitian bakteri terletak pada objek, fitur, optimasi, dan hasil. Penelitian bakteri menggunakan data numerik dan <i>Naïve Bayes</i> yang dioptimasi <i>Genetic Algorithm</i> , meningkatkan akurasi dari 80,93% menjadi 81,19%. Penelitian ini mengklasifikasi penyakit daun kelapa sawit berbasis citra dengan fitur tekstur GLCM, <i>Naïve Bayes</i> dioptimasi <i>GA</i> dan <i>PSO</i> , dengan akurasi 59% serta <i>precision, recall, F1-score</i> 58–59%.

## 2.2 Dasar Teori

Penelitian ini didukung dengan beberapa dasar teori sebagai bahan pembelajaran guna membantuk penelitian ini. Dibawah ini adalah dasar

teori yang dipakai untuk penelitian Tugas Akhir ini.

### 2.2.1 Kelapa Sawit



Gambar 2.1 *Elaeis Guineensis Jacq* atau Kelapa Sawit

Seperti terlihat pada Gambar 2.1, *Elaeis Guineensis Jacq* nama latin dari Kelapa Sawit, merupakan salah satu komoditas perkebunan terpenting di Indonesia. Indonesia adalah produsen dan eksportir minyak kelapa sawit (*Crude Palm Oil*) terbesar di dunia, dengan kontribusi lebih dari 50% produksi global [24][25]. Perkebunan kelapa sawit tersebar luas di Sumatera, Kalimantan, Sulawesi, dan Papua, dengan luas lahan mencapai lebih dari 16 juta hektar pada tahun 2017 [26].

Industri kelapa sawit memberikan kontribusi signifikan terhadap perekonomian nasional, menyerap jutaan tenaga kerja, serta menjadi sumber devisa utama negara. Selain minyak goreng, produk turunan kelapa sawit juga digunakan dalam industri makanan, kosmetik, farmasi, hingga energi terbarukan (biodiesel) [27]. Namun, pengembangan kelapa sawit juga menghadapi tantangan seperti isu lingkungan, deforestasi, dan keberlanjutan yang terus menjadi perhatian nasional maupun internasional.

Di Provinsi Lampung, perkebunan kelapa sawit juga berkembang pesat dan menjadi salah satu komoditas unggulan daerah. Menurut data, luas areal perkebunan kelapa sawit di Lampung tersebar di beberapa kabupaten, seperti Kabupaten Mesuji, Tulang Bawang, Lampung Tengah [28]. Perkebunan ini dikelola oleh perusahaan besar swasta, perkebunan negara, serta petani rakyat. Selain berkontribusi pada perekonomian daerah, pengembangan kelapa sawit di Lampung juga menghadapi tantangan terkait produktivitas, akses pasar, dan isu lingkungan [29].

Selain tantangan terkait produktivitas dan lingkungan, penyakit pada daun bibit kelapa sawit juga menjadi permasalahan penting yang dapat menghambat pertumbuhan dan perkembangan tanaman, terutama pada fase awal pertumbuhan, sekitar 1 minggu hingga 3 bulan awal. Terdapat lima jenis penyakit yang umum menyerang daun bibit kelapa sawit, yaitu bercak daun, daun berputar, daun berkerut, daun menguning, dan daun menggulung.

#### **2.2.1.1 Bercak Daun**

Bercak daun adalah penyakit yang disebabkan oleh jamur patogen dari genus *Curvularia sp.* merupakan jamur patogen utama penyebab penyakit bercak daun pada bibit kelapa sawit [30][31]. Infeksi jamur ini ditandai dengan munculnya bercak kecil berwarna kuning hingga coklat pada permukaan daun, yang secara progresif meluas dan menyebabkan nekrosis jaringan. Bercak biasanya berbentuk oval atau tidak beraturan, sering kali dikelilingi warna kuning dan pada serangan berat dapat menyebabkan daun mengering serta gugur sebelum waktunya [32]. Penyakit ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 Bercak Daun

Penyakit ini dapat berdampak langsung pada penurunan kapasitas fotosintesis akibat rusaknya jaringan daun, sehingga pertumbuhan bibit menjadi terhambat dan produktivitas tanaman dewasa berkurang. Penyebaran *Curvularia sp.* dapat melalui angin, percikan air hujan dan alat pertanian yang terkontaminasi, terutama pada kondisi lingkungan lembap dan sirkulasi udara yang buruk [32].

### 2.2.1.2 Daun Berputar

Daun berputar adalah penyakit yang disebabkan oleh infeksi virus dan faktor genetik. Penyakit ini ditandai dengan daun yang tumbuh melingkar atau berputar, serta pertumbuhan yang terhambat. Gejala awal biasanya muncul pada daun muda, di mana daun tampak lebih kecil dari ukuran normal dan memiliki bentuk yang tidak simetris. Pada serangan berat, daun dapat mengalami deformasi parah dan menguning, serta pertumbuhan tanaman menjadi terhambat [33][31]. Penyakit ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Daun Berputar

Penyakit ini dapat menyebar melalui faktor serangga kutu daun dan melalui alat pertanian yang terkontaminasi. Lingkungan yang lembap dan suhu tinggi juga dapat meningkatkan risiko infeksi. Daun berputar dapat menyebabkan penurunan hasil panen pada tanaman dewasa, sehingga penting untuk melakukan pengendalian secara dini [34].

#### 2.2.1.3 Daun Berkerut

Daun berkerut pada bibit kelapa sawit merupakan kondisi fisiologis di mana permukaan daun tampak mengerut, tidak rata dan sering kali melipat secara tidak beraturan. Penyakit ini umumnya disebabkan oleh ketidakseimbangan suplai air dan unsur hara mikro, terutama pada masa pertumbuhan awal bibit [35][31]. Faktor lingkungan seperti kelembapan rendah, paparan sinar matahari yang minim atau terlalu berlebihan, serta media tanam yang kurang optimal dapat memperparah gejala ini [36]. Penyakit ini dapat dilihat pada Gambar 2.4.



Gambar 2.4 Daun Berkerut

Daun berkerut dapat mengganggu proses fotosintesis dan penyerapan cahaya, sehingga menghambat pertumbuhan bibit. Gejala awal biasanya muncul pada daun muda, di mana daun tampak lebih kecil dari ukuran normal dan memiliki bentuk yang tidak simetris. Pada serangan berat, daun dapat mengalami deformasi parah dan menguning, serta pertumbuhan tanaman menjadi terhambat.

#### **2.2.1.4 Daun Menguning**

Daun menguning adalah penyakit yang ditandai dengan perubahan warna daun dari hijau menjadi kuning, yang dapat disebabkan oleh berbagai faktor, termasuk kekurangan unsur hara, serangan hama, infeksi patogen, dan kondisi lingkungan yang tidak optimal [33][31]. Penyakit ini dapat dilihat pada Gambar 2.5.



Gambar 2.5 Daun Menguning

Penyakit juga dapat disebabkan oleh kekurangan unsur hara, seperti nitrogen, fosfor, dan kalium, yang berperan penting dalam proses fotosintesis dan pertumbuhan tanaman. Selain itu, serangan hama seperti kutu daun dan ulat juga dapat menyebabkan daun menguning [31]. Lingkungan yang tidak optimal, seperti suhu ekstrem atau kelembapan yang rendah, juga dapat memicu gejala ini.

#### **2.2.1.5 Daun Menggulung**

Daun menggulung adalah penyakit yang ditandai dengan perubahan bentuk daun yang melipat atau menggulung ke arah dalam, sehingga mengurangi luas permukaan daun yang dapat menyerap cahaya matahari. Penyakit ini dapat disebabkan oleh infeksi virus, serangan hama, atau faktor lingkungan yang tidak optimal [37][31]. Penyakit ini dapat dilihat pada Gambar 2.6.



Gambar 2.6 Daun Menggulung

Penyakit ini dapat disebabkan oleh infeksi virus, melalui hama serangga, seperti kutu daun dan ulat [31]. Lingkungan yang tidak optimal, seperti suhu ekstrem atau kelembapan yang rendah, juga dapat memicu gejala ini. Jika tidak segera ditangani, daun menggulung dapat mengganggu proses fotosintesis dan mengurangi hasil panen.

### 2.2.2 Naïve Bayes

*Naïve Bayes* merupakan salah satu metode klasifikasi berbasis probabilistik yang paling sederhana dan efisien, serta banyak digunakan dalam berbagai kasus klasifikasi seperti pengenalan citra, analisis teks, dan deteksi penyakit. Metode ini didasarkan pada Teorema Bayes dengan asumsi bahwa setiap fitur (atribut) pada data bersifat saling bebas (independen) [38].

Dalam konteks klasifikasi, *Naïve Bayes* bertujuan untuk menentukan kelas  $C_k$  yang paling mungkin diberikan fitur-fitur  $X = (x_1, x_2, \dots, x_n)$  dari suatu data. Proses klasifikasi dilakukan dengan menghitung probabilitas posterior menggunakan Teorema Bayes sebagai berikut:

$$P(C_k|X) = \frac{P(X|C_k) \cdot P(C_k)}{P(X)} \quad (2.1)$$

Dengan keterangan sebagai berikut:

$P(C_k|X)$  : probabilitas kelas  $C_k$  diberikan fitur  $X$  (posterior)

$P(X|C_k)$  : probabilitas fitur  $X$  terjadi jika kelas  $C_k$  benar (likelihood)

$P(C_k)$  : probabilitas awal kelas  $C_k$  (prior)

$P(X)$  : probabilitas total fitur  $X$  (evidence)

Dalam prakteknya, karena  $P(X)$  adalah konstan untuk semua kelas, maka proses klasifikasi dapat disederhanakan menjadi:

$$\hat{C} = \arg \max_{C_k} P(C_k) \cdot P(X|C_k) \quad (2.2)$$

Dimana  $\hat{C}$  adalah kelas yang diprediksi.

Jika fitur  $X = (x_1, x_2, \dots, x_n)$  dianggap independen, maka:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k) \quad (2.3)$$

Dengan keterangan sebagai berikut:

$P(X|C_k)$  : probabilitas fitur  $X$  terjadi jika kelas  $C_k$  benar  
(likelihood)

$P(x_i|C_k)$  : probabilitas fitur ke- $i$  terjadi jika kelas  $C_k$  benar  
(likelihood)

### 2.2.2.1 Teorema Bayes

Teorema Bayes adalah prinsip dasar yang digunakan dalam *Naïve Bayes* untuk menghitung probabilitas posterior dari suatu hipotesis berdasarkan data yang diamati [39]. Teorema ini menyatakan bahwa probabilitas suatu hipotesis  $H$  diberikan data  $X$  dapat dihitung dengan menggunakan probabilitas awal  $P(H)$ , probabilitas data  $X$  diberikan hipotesis  $H$  dan probabilitas total data  $X$  [40].

Persamaan umum dari Teorema Bayes dinyatakan sebagai berikut:

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)} \quad (2.4)$$

Dengan keterangan sebagai berikut:

$P(H|X)$  : probabilitas hipotesis  $H$  diberikan data  $X$  (posterior)

$P(X|H)$  : probabilitas data  $X$  terjadi jika  $H$  benar (likelihood)

$P(H)$  : probabilitas awal hipotesis  $H$  (prior)

$P(X)$  : probabilitas total data  $X$  (evidence)

### 2.2.2.2 Distribusi Gaussian Fitur Numerik

*Naïve Bayes* dapat digunakan untuk data numerik dengan asumsi bahwa fitur-fitur tersebut mengikuti distribusi Gaussian. Dalam hal ini, kita perlu menghitung probabilitas dari setiap fitur  $x_i$  pada kelas  $C_k$  menggunakan distribusi normal (Gaussian) [41]. Untuk fitur numerik yang diasumsikan mengikuti distribusi normal (Gaussian), nilai probabilitas dihitung dengan:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{k,i}^2}} \exp\left(-\frac{(x_i - \mu_{k,i})^2}{2\sigma_{k,i}^2}\right) \quad (2.5)$$

Dengan keterangan sebagai berikut:

$\mu_{k,i}$  : rata-rata fitur ke- $i$  pada kelas  $C_k$

$\sigma_{k,i}$  : standar deviasi fitur ke- $i$  pada kelas  $C_k$

Pada data bertipe numerik, proses perhitungannya memiliki beberapa tahapan khusus yang berbeda dari persamaan umum sebelumnya. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. **Menghitung nilai rata-rata (mean)** untuk setiap fitur:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.6)$$

Dengan keterangan sebagai berikut:

$\mu$  : rata-rata fitur

$x_i$  : nilai fitur ke- $i$

$n$  : jumlah data pada kelas tertentu

## 2. Menghitung standar deviasi ( $\sigma$ ):

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (2.7)$$

Dengan keterangan sebagai berikut:

$\sigma$  : standar deviasi fitur

$x_i$  : nilai fitur ke- $i$

$\mu$  : rata-rata fitur

$n$  : jumlah data pada kelas tertentu

## 3. Menghitung likelihood menggunakan distribusi Gaussian berdasarkan persamaan (2.5).

Likelihood adalah probabilitas kemunculan data fitur  $x_i$  pada kelas  $C_k$ , yaitu  $P(x_i|C_k)$ , yang menunjukkan seberapa besar kemungkinan data tersebut muncul jika diketahui kelasnya.

## 4. Menghitung probabilitas posterior untuk setiap kelas:

Probabilitas posterior adalah probabilitas suatu kelas  $C_k$  setelah mempertimbangkan data fitur  $X$  yang diamati. Nilai ini dihitung dengan mengalikan probabilitas awal (prior) kelas  $C_k$  dengan hasil perkalian probabilitas setiap fitur  $x_i$  pada kelas tersebut, sesuai

dengan rumus berikut:

$$P(C_k|X) \propto P(C_k) \cdot \prod_{i=1}^n P(x_i|C_k) \quad (2.8)$$

Dengan keterangan sebagai berikut:

$P(C_k|X)$  : probabilitas posterior kelas  $C_k$  terhadap data  $X$

$P(C_k)$  : probabilitas awal (prior) kelas  $C_k$

$P(x_i|C_k)$  : probabilitas fitur  $x_i$  pada kelas  $C_k$

$n$  : jumlah fitur pada data  $X$

##### 5. Menentukan kelas hasil klasifikasi:

$$\hat{C} = \arg \max_{C_k} P(C_k|X) \quad (2.9)$$

Dengan keterangan sebagai berikut:

$\hat{C}$  : kelas hasil prediksi (kelas dengan probabilitas tertinggi)

$P(C_k|X)$  : probabilitas posterior kelas  $C_k$  terhadap data  $X$

#### 2.2.3 Genetic Algorithm

*Genetic Algorithm* (GA) adalah metode optimasi yang terinspirasi oleh proses evolusi biologis . GA menggunakan prinsip seleksi alam, di mana individu yang lebih baik memiliki peluang lebih besar untuk bertahan dan berkembang biak [42].

Dalam konteks klasifikasi, GA dapat digunakan untuk memilih fitur yang paling relevan atau mengoptimalkan parameter model. Kelebihan GA adalah kemampuannya untuk menjelajahi ruang solusi yang besar dan kompleks, serta fleksibilitas dalam mengadaptasi berbagai jenis

masalah. Namun, kelemahannya adalah waktu komputasi yang lebih lama dibandingkan metode optimasi lainnya [43].

Secara umum, proses utama dalam *Genetic Algorithm* meliputi seleksi, crossover, dan mutasi [44]. Berikut adalah rumus-rumus yang digunakan dalam setiap tahap:

### 2.2.3.1 Seleksi (Selection)

Seleksi adalah tahap dalam Genetic Algorithm (GA) yang bertujuan memilih individu-individu terbaik dari populasi berdasarkan nilai fitnessnya untuk menjadi induk (parent) pada generasi berikutnya. Individu dengan nilai fitness lebih tinggi memiliki peluang lebih besar untuk terpilih, sehingga karakteristik yang baik dapat diwariskan ke generasi selanjutnya. Metode seleksi yang umum digunakan antara lain roulette wheel selection, tournament selection, dan rank selection [45].

Pada tahap ini, individu-individu dalam populasi dievaluasi berdasarkan fungsi fitness yang telah ditentukan. Fungsi fitness mengukur seberapa baik individu tersebut dalam menyelesaikan masalah yang dihadapi. Setelah itu, individu-individu dengan nilai fitness tertinggi akan dipilih untuk menjadi induk pada generasi berikutnya [45][46].

Sebagai contoh, jika kita memiliki populasi  $P = \{I_1, I_2, \dots, I_N\}$  dengan nilai fitness masing-masing  $f_i$  [46], misalnya menggunakan *Roulette Wheel Selection*, maka kita dapat menghitung probabilitas seleksi untuk setiap individu sebagai berikut:

$$P_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (2.10)$$

dengan  $P_i$  adalah probabilitas individu ke- $i$  terpilih,  $f_i$  adalah nilai fitness individu ke- $i$ , dan  $N$  adalah jumlah total individu dalam populasi.

### 2.2.3.2 Crossover

Crossover adalah proses pertukaran gen antara dua individu (parent) untuk menghasilkan individu baru (offspring). Proses ini bertujuan untuk menggabungkan informasi genetik dari kedua parent untuk menciptakan individu yang lebih baik. Crossover dapat dilakukan dengan berbagai cara, seperti *Single Point Crossover*, *Two Point Crossover*, atau *Uniform Crossover* [47].

Crossover dilakukan dengan memilih dua parent dari populasi yang telah diseleksi, kemudian memilih titik crossover secara acak pada kromosom [48]. Setelah itu, gen-gen di antara kedua parent akan ditukar untuk membentuk dua individu baru (offspring). Proses ini diulang hingga jumlah individu baru yang diinginkan tercapai [49].

Sebagai contoh, jika kita memiliki dua parent  $P_1 = (g_{1,1}, g_{1,2}, \dots, g_{1,n})$  dan  $P_2 = (g_{2,1}, g_{2,2}, \dots, g_{2,n})$ , maka setelah crossover, kita dapat menghasilkan dua offspring sebagai berikut:

$$\begin{aligned} \text{Offspring}_1 &= (g_{1,1}, g_{1,2}, \dots, g_{1,k}, g_{2,k+1}, \dots, g_{2,n}) \\ \text{Offspring}_2 &= (g_{2,1}, g_{2,2}, \dots, g_{2,k}, g_{1,k+1}, \dots, g_{1,n}) \end{aligned} \quad (2.11)$$

dengan  $g_{1,i}$  adalah gen ke- $i$  dari parent 1,  $g_{2,i}$  adalah gen ke- $i$  dari parent 2,  $k$  adalah titik crossover, dan  $n$  adalah panjang kromosom.

### 2.2.3.3 Mutasi

Mutasi adalah proses perubahan acak pada gen individu untuk menjaga keragaman genetik dalam populasi. Proses ini bertujuan untuk mencegah konvergensi prematur dan meningkatkan eksplorasi ruang solusi. Mutasi dapat dilakukan dengan mengganti nilai gen secara acak atau mengubah posisi gen dalam kromosom [50].

Sebagai contoh, jika kita memiliki individu  $I = (g_1, g_2, \dots, g_n)$ , maka setelah mutasi, individu tersebut dapat berubah menjadi  $I' = (g'_1, g'_2, \dots, g'_n)$  dengan rumus:

$$g'_i = \begin{cases} \text{random value,} & \text{jika terjadi mutasi} \\ g_i, & \text{jika tidak terjadi mutasi} \end{cases} \quad (2.12)$$

dengan keterangan sebagai berikut:

$g_i$  : gen sebelum mutasi

$g'_i$  : gen setelah mutasi

$i$  : indeks gen yang mengalami mutasi

$n$  : jumlah gen dalam individu

### 2.2.4 Particle Swarm Optimization

*Particle Swarm Optimization* (PSO) merupakan algoritma optimasi berbasis populasi yang dikembangkan oleh James Kennedy dan Russell Eberhart pada tahun 1995, terinspirasi dari perilaku kolektif kawanan burung atau ikan dalam mencari sumber makanan [51][52]. Dalam konteks komputasi, PSO digunakan untuk menyelesaikan masalah

optimasi, khususnya pada fungsi non-linear yang kompleks dan tidak memiliki solusi analitik.

PSO bekerja dengan menginisialisasi sejumlah partikel sebagai kandidat solusi yang bergerak dalam ruang pencarian. Setiap partikel menyesuaikan posisinya berdasarkan pengalaman terbaiknya sendiri (personal best/pbest) dan pengalaman terbaik seluruh populasi (global best/gbest). Gerakan partikel dikendalikan oleh dua komponen utama, yaitu kognitif (berbasis pengalaman diri) dan sosial (berbasis interaksi kawan) [53].

Keunggulan PSO terletak pada kesederhanaannya, efisiensi memori, dan kecepatan konvergensi. PSO tidak memerlukan turunan atau informasi gradien, sehingga cocok diterapkan pada fungsi objektif yang tidak kontinu atau tidak terdiferensialkan. Namun, kelemahan utama PSO adalah kecenderungannya untuk terjebak pada solusi lokal serta ketergantungannya pada parameter awal, seperti kecepatan awal, bobot inersia, dan koefisien pembelajaran [54].

Dalam pengembangan sistem klasifikasi, PSO sering digunakan untuk mengoptimalkan parameter model, memilih fitur yang relevan, atau meningkatkan akurasi algoritma klasifikasi seperti *Naïve Bayes*. Kombinasi ini terbukti meningkatkan kinerja model secara signifikan, terutama pada data berskala besar dan kompleks.

Secara matematis, pembaruan kecepatan dan posisi partikel pada PSO dinyatakan dengan rumus berikut:

$$v_i^{(t+1)} = w \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (pbest_i - x_i^{(t)}) + c_2 \cdot r_2 \cdot (gbest - x_i^{(t)}) \quad (2.13)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (2.14)$$

Dengan keterangan sebagai berikut:

$v_i^{(t)}$  : kecepatan partikel ke- $i$  pada iterasi ke- $t$

$x_i^{(t)}$  : posisi partikel ke- $i$  pada iterasi ke- $t$

$w$  : *inertia weight* (bobot inersia) yang mengontrol pengaruh kecepatan sebelumnya

$c_1$  : *cognitive coefficient* (koefisien kognitif, biasanya sekitar 2)

$c_2$  : *social coefficient* (koefisien sosial, biasanya sekitar 2)

$r_1, r_2$  : bilangan acak antara 0 dan 1 (di-generate setiap iterasi)

$pbest_i$  : posisi terbaik yang pernah dicapai partikel ke- $i$

$gbest$  : posisi terbaik yang pernah dicapai seluruh partikel (global best)

Proses iterasi dilakukan hingga kriteria penghentian tercapai, seperti jumlah iterasi maksimum atau solusi optimal ditemukan.

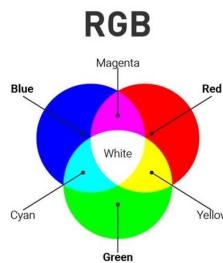
### 2.2.5 Citra Digital

Citra digital merupakan representasi visual dari suatu objek atau pemandangan dalam bentuk diskrit, yang dihasilkan melalui proses digitalisasi menggunakan perangkat seperti kamera digital, pemindai (scanner), atau sensor khusus. Fundamental dari citra digital adalah piksel (picture element), yaitu elemen terkecil yang menyusun citra

secara keseluruhan. Setiap piksel mengandung informasi mengenai intensitas warna pada posisi tertentu dalam citra [55].

Informasi warna ini direpresentasikan dalam berbagai model warna, seperti grayscale (skala keabuan yang merepresentasikan intensitas cahaya dari hitam hingga putih), RGB (Red, Green, Blue yang umum digunakan pada tampilan layar), CMYK (Cyan, Magenta, Yellow, Key/Black yang sering digunakan dalam percetakan), dan HSV (Hue, Saturation, Value yang lebih intuitif dalam merepresentasikan warna berdasarkan corak, kejemuhan, dan kecerahan). Pemilihan model warna dapat disesuaikan dengan kebutuhan analisis dan fitur yang akan diekstraksi [56].

#### 2.2.5.1 RGB



Gambar 2.7 RGB

Model warna RGB adalah salah satu model warna yang paling umum digunakan dalam citra digital. Model ini menggunakan tiga komponen warna dasar, yaitu merah (Red), hijau (Green), dan biru (Blue), untuk membentuk berbagai warna lainnya. Setiap komponen memiliki rentang nilai dari 0 hingga 255, sehingga kombinasi ketiga komponen ini dapat menghasilkan lebih dari 16 juta warna yang berbeda [57].

Model RGB bekerja berdasarkan prinsip aditif, di mana warna-warna dihasilkan dengan menjumlahkan intensitas dari ketiga komponen tersebut. Misalnya, jika semua komponen memiliki nilai maksimum (255), maka warna yang dihasilkan adalah putih. Sebaliknya, jika semua komponen memiliki nilai minimum (0), maka warna yang dihasilkan adalah hitam. Kombinasi berbagai nilai dari ketiga komponen ini memungkinkan representasi warna yang sangat kaya dan beragam [58].

Model RGB banyak digunakan dalam berbagai aplikasi, seperti fotografi digital, desain grafis, dan pemrosesan citra. Dalam konteks penelitian ini, model RGB digunakan untuk menganalisis citra daun bibit kelapa sawit guna mendeteksi penyakit yang mungkin terjadi. Dengan memanfaatkan informasi warna yang terkandung dalam citra, algoritma klasifikasi dapat mengidentifikasi pola-pola yang menunjukkan adanya penyakit pada daun [59].

#### 2.2.5.2 *Grayscale*



Gambar 2.8 Grayscale

Model warna grayscale adalah representasi citra dalam skala keabuan, di mana setiap piksel hanya memiliki satu nilai intensitas yang merepresentasikan kecerahan dari hitam (0) hingga putih (255).

Model ini menghilangkan informasi warna dan hanya mempertahankan informasi tentang intensitas cahaya [60].

Citra grayscale sering digunakan dalam pemrosesan citra karena lebih sederhana dan lebih mudah untuk dianalisis dibandingkan dengan citra berwarna. Dalam banyak aplikasi, seperti deteksi tepi, segmentasi, dan pengenalan pola, informasi warna tidak selalu diperlukan, sehingga konversi citra berwarna menjadi grayscale dapat mengurangi kompleksitas perhitungan [60].

Rumus konversi citra RGB ke grayscale yang umum digunakan adalah sebagai berikut:

$$I_{gray} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (2.15)$$

dengan keterangan sebagai berikut:

$I_{gray}$  : nilai intensitas piksel pada citra grayscale

$R$  : nilai intensitas komponen merah (Red) pada piksel

$G$  : nilai intensitas komponen hijau (Green) pada piksel

$B$  : nilai intensitas komponen biru (Blue) pada piksel

Rumus ini menggunakan pembobotan berdasarkan sensitivitas mata manusia terhadap warna, sehingga menghasilkan representasi keabuan yang lebih natural.

## 2.2.6 Identifikasi

Identifikasi adalah proses pengenalan dan penentuan karakteristik atau atribut tertentu dari objek atau fenomena yang diamati. Dalam konteks penelitian ini, identifikasi dilakukan untuk mendeteksi dan mengklasifikasikan penyakit pada daun bibit kelapa sawit berdasarkan citra digital yang diambil dari tanaman. Proses identifikasi melibatkan beberapa langkah, seperti pra-pemrosesan citra, ekstraksi fitur, dan klasifikasi menggunakan algoritma *Naïve Bayes* yang telah dioptimalkan dengan *Genetic Algorithm* dan *Particle Swarm Optimization* [61].

Proses identifikasi bertujuan untuk meningkatkan akurasi dan efisiensi dalam mendeteksi penyakit pada daun bibit kelapa sawit, sehingga dapat membantu petani dalam pengambilan keputusan yang lebih baik terkait perawatan tanaman.

## 2.2.7 Penyakit

Penyakit adalah kondisi abnormal yang terjadi pada organisme, baik itu manusia, hewan, maupun tumbuhan, yang disebabkan oleh berbagai faktor seperti infeksi patogen (virus, bakteri, jamur), gangguan genetik, atau faktor lingkungan. Penyakit dapat mempengaruhi fungsi normal organisme dan menyebabkan gejala klinis yang beragam [62].

Dalam konteks penelitian ini, penyakit yang diteliti adalah penyakit pada daun bibit kelapa sawit, yang dapat mengganggu pertumbuhan dan produktivitas tanaman. Penyakit ini dapat disebabkan oleh infeksi jamur, virus, atau faktor lingkungan yang tidak optimal. Deteksi dini dan akurat terhadap penyakit pada daun bibit kelapa sawit sangat penting untuk menjaga kesehatan tanaman dan meningkatkan hasil panen.

Penyakit pada daun bibit kelapa sawit dapat menyebabkan kerugian ekonomi yang signifikan bagi petani dan industri kelapa sawit secara keseluruhan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem identifikasi penyakit pada daun bibit kelapa sawit menggunakan algoritma *Naïve Bayes* yang dioptimalkan dengan *Genetic Algorithm* dan *Particle Swarm Optimization* untuk meningkatkan akurasi dan efisiensi dalam mendeteksi penyakit tersebut [62].

### 2.2.8 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengambilan informasi penting dari citra yang dapat digunakan untuk membedakan antara kelas-kelas yang berbeda. Dalam konteks penelitian ini, ekstraksi fitur dilakukan untuk mendapatkan karakteristik yang relevan dari citra daun bibit kelapa sawit yang terinfeksi penyakit. Fitur-fitur ini akan digunakan sebagai input untuk algoritma klasifikasi *Naïve Bayes* yang telah dioptimalkan dengan *Genetic Algorithm* dan *Particle Swarm Optimization* [63].

Ekstraksi fitur dapat dilakukan dengan berbagai metode, seperti analisis tekstur, warna, bentuk, dan pola. Dalam penelitian ini, fitur yang diekstraksi meliputi informasi warna (RGB), tekstur (menggunakan metode Gray Level Co-occurrence Matrix/GLCM), dan bentuk daun. Fitur-fitur ini diharapkan dapat memberikan informasi yang cukup untuk membedakan antara daun bibit kelapa sawit yang sehat dan yang terinfeksi penyakit.

### 2.2.8.1 Ekstraksi Fitur Warna

Ekstraksi fitur warna adalah proses pengambilan informasi warna dari citra yang dapat digunakan untuk membedakan antara kelas-kelas yang berbeda. Dalam penelitian ini, fitur warna diekstraksi dari citra daun bintangor yang terinfeksi penyakit menggunakan model warna RGB. Fitur warna yang diekstraksi meliputi nilai rata-rata, deviasi standar, dan histogram dari setiap komponen warna (merah, hijau, dan biru) [64].

Rumus ekstraksi fitur warna RGB secara matematis dapat dinyatakan sebagai berikut:

$$\bar{C} = \frac{1}{N} \sum_{i=1}^N C_i \quad (2.16)$$

$$\sigma_C = \sqrt{\frac{1}{N} \sum_{i=1}^N (C_i - \bar{C})^2} \quad (2.17)$$

dengan  $C$  mewakili salah satu komponen warna (R, G, atau B),  $\bar{C}$  adalah nilai rata-rata,  $\sigma_C$  adalah deviasi standar, dan  $N$  adalah jumlah piksel pada citra. Rumus ini digunakan untuk memperoleh karakteristik statistik warna yang dapat membedakan kondisi daun sehat dan terinfeksi penyakit.

Untuk normalisasi warna RGB, digunakan rumus:

$$R = \frac{r}{r + g + b} \quad (2.18)$$

$$G = \frac{g}{r + g + b} \quad (2.19)$$

$$B = \frac{b}{r + g + b} \quad (2.20)$$

dengan  $r$ ,  $g$ , dan  $b$  mewakili nilai intensitas komponen warna merah, hijau, dan biru pada piksel, sedangkan  $R$ ,  $G$ , dan  $B$  adalah nilai yang sudah dinormalisasi.

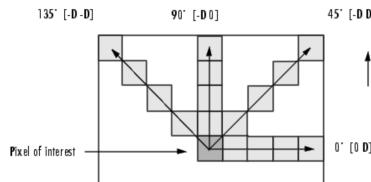
Histogram warna memberikan informasi tentang distribusi intensitas warna dalam citra, yang dapat digunakan untuk membedakan antara daun bibit kelapa sawit yang sehat dan yang terinfeksi penyakit. Histogram warna dihitung dengan menghitung jumlah piksel pada setiap rentang intensitas warna (0-255) untuk setiap komponen warna.

#### 2.2.8.2 Ekstraksi Fitur Tekstur

Ekstraksi fitur tekstur adalah proses pengambilan informasi tentang pola dan struktur permukaan dari citra. Dalam penelitian ini, fitur tekstur diekstraksi menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*. GLCM adalah matriks yang menggambarkan frekuensi kemunculan pasangan piksel dengan nilai intensitas tertentu pada jarak dan arah tertentu [65].

GLCM dapat digunakan untuk menghitung berbagai fitur tekstur, seperti kontras, energi, homogenitas, dan entropi. Fitur-fitur ini dapat

memberikan informasi yang berguna untuk membedakan antara daun bibit kelapa sawit yang sehat dan yang terinfeksi penyakit.



Gambar 2.9 *Gray Level Co-occurrence Matrix (GLCM)*

Rumus GLCM untuk menghitung frekuensi kemunculan pasangan piksel dapat dinyatakan sebagai berikut:

$$P(i, j) = \sum_{x,y} \delta(I(x, y), i) \cdot \delta(I(x + dx, y + dy), j) \quad (2.21)$$

dengan  $P(i, j)$  adalah nilai GLCM untuk pasangan piksel dengan intensitas  $i$  dan  $j$ ,  $I(x, y)$  adalah nilai intensitas piksel pada posisi  $(x, y)$ , dan  $\delta$  adalah fungsi delta Kronecker yang bernilai 1 jika argumennya sama dan 0 jika berbeda. Parameter  $dx$  dan  $dy$  menentukan jarak dan arah antara pasangan piksel yang dihitung.

Fitur tekstur yang umum dihitung dari GLCM, meliputi Contrast (Kontras), Correlation (Korelasi), Energy (Energi/Uniformity/ASM), dan Homogeneity (Homogenitas) [66].

### 1. Contrast (Kontras)

Kontras merupakan fitur GLCM yang digunakan untuk mengukur perbedaan lokal dalam citra. Nilai kontras yang tinggi menunjukkan

variasi intensitas yang besar antara piksel bersebelahan.

$$\text{Contrast} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - j)^2 P(i, j) \quad (2.22)$$

## 2. Correlation (Korelasi)

Korelasi adalah fitur GLCM yang digunakan untuk mengukur hubungan linear antara nilai intensitas piksel pada pasangan piksel dalam matriks. Nilai korelasi mencerminkan seberapa jauh hubungan antara baris dan kolom dalam GLCM, semakin tinggi nilai korelasi, semakin kuat hubungan antara intensitas piksel yang bersebelahan. Hubungan ini dapat dihitung menggunakan rumus berikut:

$$\text{Correlation} = \frac{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - \mu_i)(j - \mu_j) P(i, j)}{\sigma_i \sigma_j} \quad (2.23)$$

dengan  $\mu_i$ ,  $\mu_j$  adalah nilai rata-rata baris dan kolom, serta  $\sigma_i$ ,  $\sigma_j$  merupakan simpangan baku (standar deviasi) dari baris dan kolom.

## 3. Energy (Energi/Uniformity/ASM)

Energy mengukur tingkat keseragaman tekstur dalam citra. Semakin seragam sebuah citra, semakin tinggi nilai energinya.

$$\text{Energy} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i, j)^2 \quad (2.24)$$

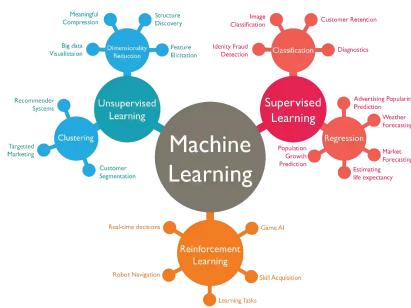
#### 4. Homogeneity (Homogenitas)

Homogenitas mengukur kedekatan nilai elemen dalam GLCM terhadap diagonal utama. Nilai homogenitas tinggi menunjukkan bahwa nilai elemen lebih terkonsentrasi di sekitar diagonal.

$$\text{Homogeneity} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{P(i, j)}{1 + |i - j|} \quad (2.25)$$

#### 2.2.9 Machine Learning

Machine Learning adalah cabang dari kecerdasan buatan (Artificial Intelligence) yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit. Machine Learning mengandalkan teknik statistik dan matematis untuk menganalisis pola dalam data, sehingga dapat digunakan untuk berbagai aplikasi, seperti klasifikasi, regresi, clustering, dan rekomendasi [67].



Gambar 2.10 Machine Learning

Dalam konteks penelitian ini, Machine Learning digunakan untuk mengembangkan sistem identifikasi penyakit pada daun bibit kelapa

sawit. Algoritma yang digunakan adalah *Naïve Bayes* yang dioptimalkan dengan *Genetic Algorithm* dan *Particle Swarm Optimization*. Dengan menggunakan teknik Machine Learning, sistem dapat belajar dari data citra daun bintangor sawit yang terinfeksi penyakit dan mengklasifikasikan kondisi daun berdasarkan fitur-fitur yang diekstraksi.

### 2.2.10 Segmentasi Citra

Segmentasi citra adalah proses pemisahan citra menjadi beberapa bagian atau objek yang lebih kecil untuk memudahkan analisis dan pengolahan. Dalam konteks penelitian ini, segmentasi citra digunakan untuk mengidentifikasi dan mengekstraksi area daun bintangor sawit yang terinfeksi penyakit dari latar belakang citra [68].

Segmentasi citra dapat dilakukan dengan berbagai metode, seperti thresholding, region growing, clustering, dan deteksi tepi. Metode yang dipilih tergantung pada karakteristik citra dan tujuan analisis. Dalam penelitian ini, metode segmentasi yang digunakan adalah thresholding untuk memisahkan area daun dari latar belakang berdasarkan nilai intensitas warna.

Selain thresholding, penelitian ini juga memanfaatkan *Phototoroom*, yaitu website untuk menghilangkan latar belakang pada citra. Dengan *Phototoroom*, proses segmentasi menjadi lebih efisien dan akurat, terutama untuk citra dengan latar belakang yang kompleks. Penggunaan *Phototoroom* memungkinkan area daun dapat diekstraksi secara otomatis tanpa perlu pengaturan threshold manual, sehingga mempercepat proses pra-pemrosesan citra [68].

Setelah proses segmentasi, area daun yang terinfeksi penyakit dapat

dianalisis lebih lanjut untuk ekstraksi fitur dan klasifikasi menggunakan algoritma *Naïve Bayes* yang telah dioptimalkan dengan *Genetic Algorithm* dan *Particle Swarm Optimization*.

### 2.2.11 Hyperparameter

Hyperparameter adalah parameter yang ditentukan sebelum proses pelatihan model dimulai dan tidak diperoleh dari data pelatihan. Hyperparameter berfungsi untuk mengontrol proses pembelajaran dan mempengaruhi kinerja model. Dalam konteks penelitian ini, hyperparameter digunakan untuk mengoptimalkan algoritma *Naïve Bayes* yang diterapkan pada citra daun bibit kelapa sawit [61].

Pemilihan hyperparameter yang tepat sangat penting untuk mencapai kinerja model yang optimal. Dalam penelitian ini, hyperparameter dioptimalkan menggunakan algoritma *Genetic Algorithm* dan *Particle Swarm Optimization* untuk meningkatkan akurasi klasifikasi penyakit pada daun bibit kelapa sawit [69].

Pada penelitian ini, hyperparameter yang digunakan adalah *Variable Smoothing*. *Variable Smoothing* adalah teknik yang digunakan untuk menghindari pembagian dengan nol pada perhitungan probabilitas dalam algoritma *Naïve Bayes*. Dengan menggunakan *Variable Smoothing*, setiap frekuensi kata dalam perhitungan probabilitas akan ditambahkan dengan nilai *Smoothing* yang ditentukan. Hal ini membantu meningkatkan stabilitas dan akurasi model klasifikasi.

### 2.2.12 Confusion Matrix

*Confusion Matrix* adalah alat evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan membandingkan hasil prediksi model dengan label sebenarnya. *Confusion Matrix* memberikan informasi tentang jumlah prediksi yang benar dan salah untuk setiap kelas dalam dataset [70].

*Confusion Matrix* dapat digunakan untuk menghitung berbagai metrik evaluasi, seperti akurasi, presisi, recall, dan F1-score. Metrik-metrik ini memberikan gambaran yang lebih jelas tentang kinerja model klasifikasi dalam mendeteksi penyakit pada daun bibit kelapa sawit.

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive)	FP (False Positive)
	FALSE	FN (False Negative)	TN (True Negative)

Gambar 2.11 Confusion Matrix

Dengan keterangan sebagai berikut:

*TP* : jumlah data yang sebenarnya positif dan diprediksi positif

*FP* : jumlah data yang sebenarnya negatif tetapi diprediksi positif

*FN* : jumlah data yang sebenarnya positif tetapi diprediksi negatif

*TN* : jumlah data yang sebenarnya negatif dan diprediksi negatif

Dari Confusion Matrix di atas, nilai-nilai tersebut digunakan untuk menghitung metrik evaluasi sebagai berikut.

1. Akurasi (Accuracy)

Akurasi (*Accuracy*) menunjukkan seberapa akurat prediksi yang benar

dari metode yang digunakan:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.26)$$

## 2. Presisi (Precision)

Presisi (*Precision*) adalah rasio antara jumlah prediksi positif yang benar (TP) dengan keseluruhan hasil prediksi positif (TP + FP), yang menggambarkan tingkat keakuratan prediksi positif dari model:

$$\text{precision} = \frac{TP}{TP + FP} \quad (2.27)$$

## 3. Recall

Recall adalah rasio antara jumlah prediksi positif yang benar (TP) dengan jumlah keseluruhan data yang sebenarnya positif (TP + FN), yang menunjukkan kemampuan model dalam menemukan data positif yang sebenarnya:

$$\text{recall} = \frac{TP}{TP + FN} \quad (2.28)$$

### 2.2.13 Roboflow

Roboflow adalah platform yang menyediakan alat dan layanan untuk membantu pengembang dan peneliti dalam membangun, melatih,

dan menerapkan model pembelajaran mesin untuk pengenalan citra. Roboflow menawarkan berbagai fitur, termasuk pengumpulan data, anotasi, augmentasi, dan pelatihan model [71].

Dengan Roboflow, pengguna dapat dengan mudah mengunggah citra, menandai objek dalam citra, dan melakukan augmentasi data untuk meningkatkan variasi dan jumlah data pelatihan. Platform ini juga menyediakan integrasi dengan berbagai framework pembelajaran mesin populer, seperti TensorFlow, PyTorch, dan Keras.

Roboflow memungkinkan pengguna untuk membangun model pembelajaran mesin tanpa perlu memiliki pengetahuan mendalam tentang pemrograman atau algoritma pembelajaran mesin. Dengan antarmuka yang intuitif dan alat yang mudah digunakan, Roboflow membantu mempercepat proses pengembangan model pengenalan citra.

Roboflow juga menyediakan API dan SDK yang memungkinkan pengguna untuk mengintegrasikan model yang telah dilatih ke dalam aplikasi mereka. Dengan menggunakan Roboflow, pengguna dapat dengan cepat membangun dan menerapkan model pembelajaran mesin untuk berbagai aplikasi, seperti deteksi objek, segmentasi citra, dan klasifikasi citra.

Roboflow juga menyediakan fitur untuk mengelola dan menyimpan dataset, serta melakukan kolaborasi dengan tim dalam proyek pembelajaran mesin. Dengan semua fitur ini, Roboflow menjadi alat yang sangat berguna bagi pengembang dan peneliti yang ingin membangun model pembelajaran mesin untuk pengenalan citra dengan cepat dan efisien.

### 2.2.14 StreamLit

Streamlit adalah framework open-source yang digunakan untuk membangun aplikasi web interaktif dengan cepat dan mudah, khususnya untuk aplikasi yang berkaitan dengan data sains dan pembelajaran mesin. Dengan Streamlit, pengguna dapat membuat antarmuka pengguna (UI) yang menarik dan responsif tanpa perlu memiliki pengetahuan mendalam tentang pengembangan web. Streamlit memungkinkan pengguna untuk menulis aplikasi web menggunakan bahasa pemrograman Python, sehingga memudahkan integrasi dengan berbagai pustaka dan alat analisis data yang sudah ada [72].

Dengan menggunakan Streamlit, pengguna dapat dengan cepat membangun aplikasi web yang menarik dan interaktif untuk memvisualisasikan data, menjalankan model pembelajaran mesin, dan berbagi hasil analisis dengan orang lain. Dalam konteks penelitian ini, Streamlit digunakan untuk membuat antarmuka pengguna yang memungkinkan pengguna untuk mengunggah citra daun bintik kelapa sawit dan mendapatkan hasil klasifikasi penyakit secara real-time.

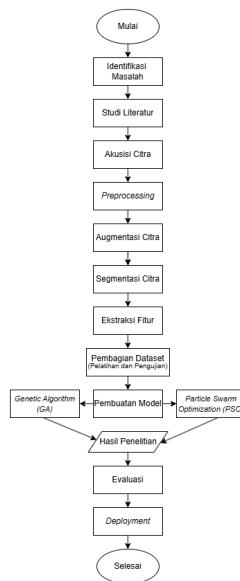
## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Alur Penelitian

Alur penelitian menggambarkan tahapan yang diikuti dalam proses pelaksanaan penelitian identifikasi penyakit pada bibit daun kelapa sawit dengan menggunakan ekstraksi fitur warna dan bentuk dengan metode *Naïve Bayes* untuk proses identifikasinya, optimasi menggunakan *Genetic Algorithm* dan *Particle Swarm Optimization*.

Berikut adalah gambaran visual dari alur penelitian yang akan dilakukan dalam studi ini, seperti yang terlihat pada Gambar 3.12.



Gambar 3.12 Diagram Alur Penelitian

### **3.2 Penjabaran Langkah Penelitian**

Penjabaran langkah-langkah penelitian yang dilakukan dalam penelitian ini. Penjelasan langkah-langkah penelitian ini akan menjelaskan secara rinci setiap langkah yang diambil dalam penelitian ini, sudah tergambar dalam flowchart di subbab 3.1.

#### **3.2.1 Identifikasi Masalah**

Tahap identifikasi masalah dalam penelitian ini diawali dengan pelaksanaan observasi lapangan secara sistematis dan wawancara mendalam di PT Perkebunan Nusantara IV Regional 7 Kebun Bekri, Lampung Tengah. Observasi dilakukan pada area pembibitan kelapa sawit (*Elaeis guineensis Jacq*) dengan rentang umur bibit 1 minggu hingga 3 bulan, guna mengidentifikasi secara langsung berbagai gejala patologis yang muncul pada daun bibit. Proses observasi difokuskan pada deteksi dini penyakit yang menyerang daun bibit, termasuk perubahan morfologi, perubahan warna, serta pola penyebaran gejala pada populasi bibit.

Selain observasi, dilakukan pula wawancara dengan tenaga ahli agronomi dan petugas lapangan yang memiliki pengalaman langsung dalam pengelolaan pembibitan kelapa sawit. Wawancara ini bertujuan untuk memperoleh data primer terkait karakteristik gejala penyakit, faktor-faktor penyebab, serta pola penyebaran penyakit pada bibit kelapa sawit. Informasi yang diperoleh dari wawancara digunakan untuk memperkuat temuan observasi lapangan dan memastikan validitas data yang dikumpulkan.

Hasil identifikasi menunjukkan bahwa terdapat lima kategori utama

penyakit yang sering menyerang daun bibit kelapa sawit di lokasi penelitian, yaitu: bercak daun, daun menguning, daun menggulung, daun berputar, dan daun berkerut. Setiap kategori penyakit memiliki karakteristik gejala yang berbeda, baik dari segi bentuk, warna, maupun pola penyebarannya.

### 3.2.2 Studi Literatur

Studi literatur dalam penelitian ini dilakukan secara sistematis melalui penelusuran sumber-sumber akademis terpilih, seperti jurnal ilmiah, buku referensi, artikel prosiding konferensi, dan publikasi terbaru yang relevan dengan topik penelitian. Referensi yang dikaji difokuskan pada tiga aspek utama, yaitu (1) metode identifikasi penyakit tanaman berbasis citra digital, khususnya pada bibit kelapa sawit, (2) teknik ekstraksi fitur warna dan bentuk pada citra daun, dan (3) penerapan dan optimasi algoritma *Naïve Bayes*, *Genetic Algorithm*, dan *Particle Swarm Optimization* dalam klasifikasi dan pengenalan pola penyakit tanaman.

Proses studi literatur diawali dengan penentuan kata kunci yang relevan, seperti "identifikasi penyakit bibit daun kelapa sawit", "ekstraksi fitur warna dan bentuk", "Naïve Bayes", "Genetic Algorithm", dan "Particle Swarm Optimization". Selanjutnya, dilakukan pencarian literatur pada basis data ilmiah seperti IEEE Xplore, ScienceDirect, SpringerLink, dan Google Scholar. Setiap literatur yang diperoleh dianalisis untuk mengidentifikasi perkembangan metode, kelebihan dan keterbatasan pendekatan yang telah ada, serta peluang inovasi yang dapat diadopsi dalam penelitian ini.

Hasil studi literatur menjadi dasar dalam merumuskan metodologi

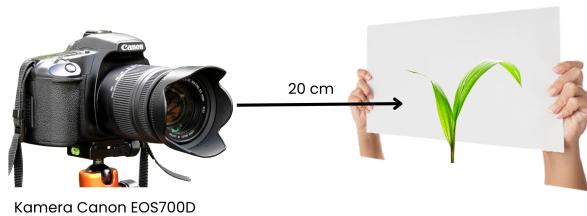
penelitian dan tinjauan pustaka, menentukan parameter dan teknik yang digunakan, serta membandingkan hasil penelitian dengan studi terdahulu untuk menilai kontribusi dan keunggulan pendekatan yang diusulkan.

### 3.2.3 Akuisisi Dataset Citra

Pengumpulan dataset dilakukan secara langsung di area pembibitan PT Perkebunan Nusantara IV Regional 7 Kebun Bekri, Lampung Tengah, sesuai prosedur perizinan dari pihak perusahaan. Data citra daun bibit kelapa sawit diambil dengan mengelompokkan sesuai lima kategori penyakit utama yang telah diidentifikasi, serta daun sehat sebagai banding.

Pengambilan citra dilakukan menggunakan kamera Canon EOS700D pada jarak sekitar 20 cm dari objek daun. Untuk menjaga konsistensi dan memudahkan proses segmentasi, digunakan kertas HVS putih sebagai latar belakang saat pengambilan gambar. Setiap citra diambil dengan memperhatikan variasi sudut dan pencahayaan agar dataset yang dihasilkan representatif.

Seluruh citra yang diperoleh diberi label sesuai kategori penyakit berdasarkan hasil observasi dan konfirmasi tenaga ahli agronomi. Dataset citra disimpan dalam format JPEG/PNG dan diorganisasi ke dalam folder sesuai kelas penyakit untuk memudahkan proses ekstraksi fitur dan pelatihan model klasifikasi. Ilustrasi teknik pengambilan dataset citra daun bibit kelapa sawit dapat dilihat pada Gambar 3.13.



Gambar 3.13 Ilustrasi Pengambilan Dataset Daun Bibit Kelapa Sawit

### 3.2.4 *Preprocessing*

Dataset yang telah diakuisisi selanjutnya melalui tahap *preprocessing* untuk memastikan kualitas data sebelum digunakan dalam pelatihan model. *Preprocessing* bertujuan meningkatkan konsistensi dan akurasi data citra, sehingga fitur yang diekstraksi menjadi lebih representatif.

Tahapan *preprocessing* yang dilakukan meliputi penghapusan latar belakang untuk memudahkan deteksi objek utama, yaitu daun kelapa sawit. Selanjutnya, dilakukan *cropping* untuk memfokuskan area citra pada objek daun, diikuti dengan proses resizing agar seluruh citra memiliki dimensi seragam. Pada penelitian ini, citra diubah ke ukuran 256 x 256 piksel dengan rasio 1:1, menyesuaikan kebutuhan model *Naive Bayes*. Setelah itu, normalisasi dilakukan menggunakan nilai rata-rata (mean) dan standar deviasi (*standard deviation*) agar distribusi intensitas piksel seragam dan tidak terpengaruh variasi pencahayaan.

Dengan tahapan *preprocessing* ini, diharapkan data citra yang digunakan pada proses ekstraksi fitur dan pelatihan model menjadi lebih optimal dan siap untuk tahap analisis selanjutnya.

### 3.2.5 Augmentasi Citra

Augmentasi citra dilakukan untuk memperbanyak jumlah data pelatihan dan meningkatkan keragaman dataset. Proses ini bertujuan untuk mengurangi risiko *overfitting* pada model klasifikasi yang dibangun. Probabilitas augmentasi citra ini disusun secara paralel pada platform Roboflow, yang menyediakan berbagai teknik augmentasi citra secara otomatis.

Dalam platform Roboflow, transformasi augmentasi diterapkan menggunakan pendekatan probabilistik (misalnya  $p=0.5$  untuk rotasi), menghasilkan variasi data lebih komprehensif dibandingkan metode deterministik konvensional. Sebagai contoh, jika rotasi  $90^\circ$  diterapkan dengan probabilitas 0.5 (50%), maka hanya sekitar setengah dari gambar dataset akan mengalami transformasi rotasi tersebut.

Pendekatan stokastik ini kritis untuk mencegah bias sistematis dan membangun model robust. Pada identifikasi patologi daun kelapa sawit, augmentasi probabilistik memungkinkan model mengakomodasi variabilitas morfologis penyakit, termasuk variasi sudut, orientasi, dan pencahayaan yang menyerupai kondisi lapangan. Konfigurasi multi-probabilitas menciptakan ruang fitur yang lebih kaya, meningkatkan kapasitas generalisasi terhadap kondisi tidak terkontrol. Transformasi independen pada setiap citra memperkaya distribusi data pelatihan tanpa menambah beban akuisisi data primer.

Penyusunan probabilitas paralel ini memungkinkan setiap citra dalam dataset mengalami beberapa transformasi secara acak, sehingga menghasilkan variasi citra yang berbeda dari citra asli. Dengan demikian,

model dapat belajar mengenali pola penyakit dari berbagai sudut pandang dan kondisi pencahayaan yang berbeda.

Augmentasi dilakukan dengan menerapkan beberapa teknik, antara lain rotasi, *flipping*, *shear*, dan perubahan kecerahan. Teknik rotasi dilakukan dengan memutar citra pada sudut tertentu, teknik *flipping* dilakukan dengan membalik citra secara horizontal atau vertikal, teknik *shear* adalah teknik yang mengubah perspektif citra dengan menggeser piksel pada sumbu x atau y, sehingga menghasilkan efek miring, sedangkan perubahan kecerahan dilakukan dengan menyesuaikan nilai intensitas piksel pada citra.

### 3.2.5.1 Rotasi

Rotasi citra dilakukan dengan memutar citra pada sudut tertentu, misalnya 90, 180, atau 270 derajat. Proses ini bertujuan untuk menghasilkan variasi citra yang berbeda dari citra asli, sehingga model dapat belajar mengenali pola penyakit dari berbagai sudut pandang.



Gambar 3.14 Ilustrasi Rotasi Citra

### 3.2.5.2 Flipping

*Flipping* citra dilakukan dengan membalik citra secara horizontal atau vertikal. Teknik ini bertujuan untuk menghasilkan variasi citra yang berbeda dari citra asli, sehingga model dapat belajar mengenali pola penyakit dari berbagai sudut pandang.



Gambar 3.15 Ilustrasi Flipping Citra

### 3.2.5.3 Shear

*Shear* adalah teknik yang mengubah perspektif citra dengan menggeser piksel pada sumbu x atau y, sehingga menghasilkan efek miring. Teknik ini bertujuan untuk menghasilkan variasi citra yang berbeda dari citra asli, sehingga model dapat belajar mengenali pola penyakit dari berbagai sudut pandang.

### 3.2.5.4 Perubahan Kecerahan

Perubahan kecerahan dilakukan dengan menyesuaikan nilai intensitas piksel pada citra. Teknik ini bertujuan untuk menghasilkan variasi citra yang berbeda dari citra asli, sehingga model dapat belajar mengenali pola penyakit dari berbagai sudut pandang.



Gambar 3.16 Ilustrasi Perubahan Kecerahan Citra

### 3.2.6 Segmentasi Citra

Segmentasi Citra merupakan proses pemisahan objek dari latar belakang dalam citra. Pada penelitian ini, segmentasi dilakukan untuk memisahkan bibit daun kelapa sawit dari latar belakang putih yang digunakan saat pengambilan gambar. Proses segmentasi yang dilakukan

pada penelitian ini adalah dengan cara manual dengan memakai Website Photoroom yang dapat menghapus latar belakang pada citra daun kelapa sawit. Dengan menggunakan metode ini, diharapkan hasil segmentasi dapat lebih akurat dan memudahkan proses ekstraksi fitur selanjutnya.



Gambar 3.17 Ilustrasi Segmentasi Citra

### 3.2.7 Ekstraksi Fitur

Ekstraksi fitur merupakan langkah dalam proses pengenalan pola, di mana fitur-fitur yang relevan diambil dari citra daun untuk digunakan dalam pembuatan model. Pada penelitian ini, dua jenis fitur diekstraksi, yaitu fitur warna dan fitur bentuk.

Ekstraksi fitur warna menggunakan klasifikasi hex code RGB dan ekstraksi fitur tekstur menggunakan metode GLCM. Pada ekstraksi warna, dilakukan perhitungan rata-rata nilai RGB dari setiap citra dan keseluruhan dataset, sementara untuk tekstur, citra dikonversi ke *grayscale* untuk menganalisis karakteristik seperti kontras, korelasi, energi, dan homogenitas menggunakan GLCM. Hasil ekstraksi fitur ini kemudian digunakan dalam proses klasifikasi menggunakan metode *Naïve Bayes*, di mana dilakukan perhitungan nilai rata-rata, standar deviasi, *prior probability*, dan *posterior probability* untuk setiap kelas.

Fitur-fitur yang diekstraksi kemudian digabungkan menjadi vektor fitur tunggal untuk setiap citra. Vektor fitur ini akan digunakan sebagai

input pada model yang dibangun.

### 3.2.8 Pembagian Dataset

Pembagian Dataset dilakukan untuk memisahkan data menjadi dua bagian, yaitu data pelatihan dan data pengujian. Data pelatihan digunakan untuk melatih model klasifikasi, sedangkan data pengujian digunakan untuk mengukur kinerja model setelah dilatih.

Pada penelitian ini, pembagian dataset dilakukan dengan proporsi 80:20, di mana 80% dari total dataset digunakan sebagai data pelatihan dan 20% sebagai data pengujian. Pembagian ini bertujuan untuk memastikan bahwa model yang dibangun dapat generalisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya.

### 3.2.9 Pembuatan Model

Pembuatan model dilakukan dengan menggunakan metode algoritma *Naïve Bayes*, optimasi *Genetic Algorithm* dan *Particle Swarm Optimization*. Model *Naïve Bayes* digunakan untuk identifikasi penyakit berdasarkan fitur yang diekstraksi dari citra daun. Algoritma ini bekerja dengan menghitung probabilitas setiap kelas penyakit berdasarkan fitur yang ada.

*Genetic Algorithm* digunakan untuk optimasi parameter model, dengan tujuan meningkatkan akurasi klasifikasi. Algoritma ini bekerja dengan cara mensimulasikan proses evolusi, di mana individu-individu dalam populasi dihasilkan melalui kombinasi dan mutasi parameter yang ada.

*Particle Swarm Optimization* digunakan untuk mencari solusi optimal

dalam ruang parameter model. Algoritma ini bekerja dengan cara mensimulasikan perilaku kawanan burung, di mana setiap partikel dalam kawanan bergerak menuju posisi terbaik yang ditemukan oleh individu dan kelompok.

### 3.2.10 Hasil

Hasil dari penelitian ini adalah model yang dapat mengidentifikasi penyakit pada bibit daun kelapa sawit dengan akurasi yang tinggi. Model yang dihasilkan diuji menggunakan data pengujian yang telah dipisahkan sebelumnya.

### 3.2.11 Evaluasi

Tahapan evaluasi merupakan ulasan terkait penelitian yang telah dilaksanakan untuk menilai kinerja model yang telah dibuat. Evaluasi dilakukan menggunakan dua pendekatan, yaitu menggunakan *confusion matrix* dan metrik evaluasi.

*Confusion matrix* digunakan untuk mengukur performa model klasifikasi dengan memberikan rincian jumlah prediksi yang benar dan salah pada masing-masing kelas. Nilai-nilai yang dihasilkan dari *confusion matrix* meliputi *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*. Berdasarkan nilai-nilai ini, dapat dihitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*.

Akurasi mengukur proporsi prediksi yang benar terhadap seluruh data, presisi mengukur ketepatan prediksi positif, *recall* mengukur kemampuan model dalam menemukan seluruh kasus positif, dan *F1-score* merupakan rata-rata harmonis dari presisi dan recall. Dengan

menggunakan *confusion matrix* dan metrik evaluasi tersebut, dapat diketahui seberapa baik model dalam mengidentifikasi penyakit pada bibit daun kelapa sawit. Hasil evaluasi ini menjadi dasar untuk menilai efektivitas dan keandalan model yang dihasilkan dalam penelitian.

### **3.2.12 Deployment**

*Deployment* adalah tahap akhir dari penelitian ini, di mana model yang telah dibangun dan diuji akan diterapkan dalam lingkungan nyata. Pada penelitian ini, deployment dilakukan dengan mengintegrasikan model ke dalam aplikasi berbasis web menggunakan Streamlit. Aplikasi web ini bertujuan untuk memudahkan pengguna dalam mengidentifikasi penyakit pada daun bibit kelapa sawit.

Pengguna dapat mengunggah citra daun bibit kelapa sawit ke dalam aplikasi, kemudian aplikasi akan memproses citra tersebut dan memberikan hasil klasifikasi penyakit secara otomatis. Dengan adanya aplikasi ini, proses identifikasi penyakit menjadi lebih praktis dan dapat diakses oleh pengguna secara luas.

## **3.3 Alat dan Bahan Tugas Akhir**

Alat dan bahan digunakan sebagai penunjang dalam penelitian ini guna mendapatkan hasil yang optimal. Spesifikasi alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

### **3.3.1 Alat**

Alat yang digunakan selama penelitian ini berlangsung, antara lain:

1. ASUS Vivobook Pro 15 OLED M3500QC-OLED956 Quiet Blue, AMD Ryzen 9 5900HX, 16GB DDR4, 512GB SSD M.2 PCIe

NVMe, NVIDIA GeForce RTX 3050 4GB DDR6, 15.6" Full HD (1920x1080) OLED, Wi-Fi, Bluetooth, Webcam, Backlight Keyboard, Fingerprint, 1.65kg, Windows 11 + Office Home Student 2021.

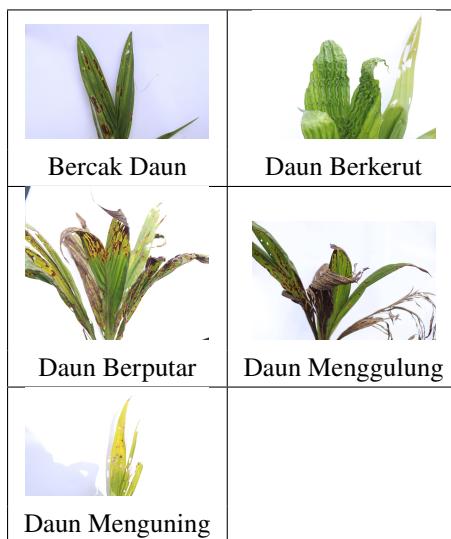
2. Notebook dengan spesifikasi minimum sistem operasi MacOS Sonoma 14.7, processor Apple M3, RAM 8GB, SSD 512 GB.
3. Notebook dengan spesifikasi minimum sistem operasi Windows 11, processor AMD Ryzen 5 5560u CPU @ 6 core 12 Threads / 2.3 GHz, RAM 16GB DDR4 grafis AMD Radeon RX Vega 6 512MB, SSD 256 GB.
4. Canon EOS 700D
5. Kertas HVS 1 lembar
6. Google Collab
7. Google Drive
8. Python 3.11.12
9. Code editor Microsoft Visual Studio Code
10. Github
11. Roboflow
12. Streamlit
13. Photoroom

### **3.3.2 Bahan**

Bahan yang digunakan/diperlukan untuk melakukan penelitian, dapat berupa:

1. Dataset citra daun bintik kelapa sawit yang diambil secara langsung di PT Perkebunan Nusantara IV Regional 7 Kebun Bekri, Lampung

Tengah pada tanggal 6 November 2024 dan 28 Oktober 2024. Pengambilan citra menggunakan kamera Canon EOS700D dengan jarak 20 cm antara daun dan kamera, serta menggunakan kertas HVS putih sebagai latar belakang untuk memudahkan proses segmentasi. Dataset citra yang digunakan untuk model *Naïve Bayes* terdiri dari 5 kelas, yaitu bercak daun sebanyak 75 citra, daun berkerut sebanyak 95 citra, daun berputar sebanyak 65 citra, daun menggulung sebanyak 40 citra, dan daun menguning sebanyak 65 citra.



Tabel 3.2 Foto Dataset Daun Bibit Kelapa Sawit

### 3.4 Metode Tugas Akhir

Metode yang digunakan dalam penelitian ini terdiri dari beberapa tahapan utama, yaitu augmentasi data, segmentasi citra, ekstraksi fitur, identifikasi menggunakan *Naïve Bayes Gaussian*, optimasi model dengan

*Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO), serta evaluasi menggunakan *confusion matrix*. Berikut penjelasan setiap tahapan:

1. Augmentasi Data: Proses augmentasi citra dilakukan menggunakan platform Roboflow untuk memperbanyak dan memperkaya variasi dataset. Teknik augmentasi yang digunakan meliputi rotasi, *flipping*, *shear*, dan perubahan kecerahan, sehingga model dapat belajar dari data yang lebih beragam dan mengurangi risiko *overfitting*.
2. Segmentasi Citra: Segmentasi citra dilakukan secara otomatis menggunakan *library Rembg* untuk menghapus latar belakang, kemudian dilakukan penyempurnaan secara manual melalui website Photoroom agar objek daun kelapa sawit terpisah sempurna dari latar belakang.
3. Ekstraksi Fitur: Fitur warna dan tekstur diekstraksi dari citra hasil segmentasi. Fitur warna diambil dari rata-rata nilai RGB, sedangkan fitur tekstur diperoleh menggunakan metode *Gray Level Co-occurrence Matrix (GLCM)*.
4. Klasifikasi *Naïve Bayes Gaussian*: Model klasifikasi utama yang digunakan adalah *Naïve Bayes Gaussian*, yang menghitung probabilitas setiap kelas penyakit berdasarkan distribusi normal dari fitur yang diekstraksi.
5. Optimasi Model: Untuk meningkatkan akurasi, parameter model *Naïve Bayes* dioptimasi menggunakan dua algoritma metaheuristik, yaitu *Genetic Algorithm* (GA) dan *Particle Swarm Optimization*

(PSO). Kedua metode ini digunakan untuk mencari parameter terbaik yang memaksimalkan performa klasifikasi.

6. Evaluasi Model: Evaluasi performa model dilakukan menggunakan *confusion matrix* yang menghasilkan metrik akurasi, presisi, recall, dan F1-score untuk menilai efektivitas model dalam mengidentifikasi penyakit pada daun bintek kelapa sawit.

Metode ini dirancang agar proses identifikasi penyakit pada daun bintek kelapa sawit dapat dilakukan secara otomatis, akurat, dan dapat diimplementasikan pada aplikasi berbasis web.

### 3.5 Ilustrasi Perhitungan Metode

Ilustrasi perhitungan metode yang digunakan dalam penelitian ini adalah sebagai berikut:

#### 3.5.1 Ekstraksi Fitur Warna

Ekstraksi fitur warna pada penelitian ini dilakukan dengan menghitung rata-rata nilai RGB dari citra daun bintek kelapa sawit. Sebagai ilustrasi, proses ekstraksi nilai RGB dapat dijelaskan melalui contoh citra berukuran 4x4 piksel seperti yang ditunjukkan pada Tabel 3.3 berikut.

[155, 48, 201]	[92, 210, 35]	[18, 130, 245]	[233, 79, 12]
[64, 188, 251]	[111, 22, 167]	[219, 143, 88]	[53, 199, 71]
[176, 31, 238]	[85, 250, 96]	[40, 165, 15]	[205, 58, 124]
[25, 140, 67]	[248, 103, 191]	[70, 14, 215]	[135, 228, 49]

Tabel 3.3 Tabel nilai RGB dalam format [R, G, B]

### 3.5.1.1 Nilai Rata-rata RGB

Selanjutnya, dilakukan perhitungan rata-rata untuk masing-masing komponen R, G, dan B dari seluruh piksel pada citra tersebut menggunakan metode averaging.

$$\begin{aligned}
 R &= \frac{155 + 92 + 18 + 233 + 64 + 111 + 219 + 53}{16} \\
 &+ \frac{176 + 85 + 40 + 205 + 25 + 248 + 70 + 135}{16} \\
 &= \frac{2134}{16} \\
 &= 133.375
 \end{aligned}$$

$$\begin{aligned}
 G &= \frac{48 + 210 + 130 + 79 + 188 + 22 + 143 + 199}{16} \\
 &+ \frac{31 + 250 + 165 + 58 + 140 + 103 + 14 + 228}{16} \\
 &= \frac{2008}{16} \\
 &= 125.5
 \end{aligned}$$

$$\begin{aligned}
 B &= \frac{201 + 35 + 245 + 12 + 251 + 167 + 88 + 71}{16} \\
 &+ \frac{238 + 96 + 15 + 124 + 67 + 191 + 215 + 49}{16} \\
 &= \frac{2065}{16} \\
 &= 129.0625
 \end{aligned}$$

### 3.5.1.2 Normalisasi Nilai Rata-rata RGB

Setelah diperoleh nilai rata-rata untuk setiap *channel* warna (R, G, dan B), langkah selanjutnya adalah melakukan normalisasi terhadap nilai rata-rata tersebut. Normalisasi bertujuan agar nilai rata-rata RGB berada pada rentang yang seragam (misal 0–1), sehingga fitur warna yang dihasilkan menjadi lebih representatif dan tidak terpengaruh oleh skala intensitas piksel.

$$\begin{aligned} R &= \frac{133.375}{133.375 + 125.5 + 129.0625} \\ &= \frac{133.375}{387.9375} \\ &= 0.343 \end{aligned}$$

$$\begin{aligned} G &= \frac{125.5}{133.375 + 125.5 + 129.0625} \\ &= \frac{125.5}{387.9375} \\ &= 0.323 \end{aligned}$$

$$\begin{aligned} B &= \frac{129.0625}{133.375 + 125.5 + 129.0625} \\ &= \frac{129.0625}{387.9375} \\ &= 0.332 \end{aligned}$$

### 3.5.2 Ekstraksi Fitur Tekstur

Ekstraksi fitur tekstur dilakukan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Setelah citra diubah ke *grayscale*, GLCM dihitung untuk memperoleh fitur tekstur utama seperti kontras, korelasi, energi, dan homogenitas. Nilai-nilai ini kemudian digunakan sebagai input pada model klasifikasi. Contoh nilai RGB yang digunakan dapat dilihat pada Tabel 3.3 di bawah.

[155, 48, 201]	[92, 210, 35]	[18, 130, 245]	[233, 79, 12]
[64, 188, 251]	[111, 22, 167]	[219, 143, 88]	[53, 199, 71]
[176, 31, 238]	[85, 250, 96]	[40, 165, 15]	[205, 58, 124]
[25, 140, 67]	[248, 103, 191]	[70, 14, 215]	[135, 228, 49]

Tabel 3.4 Tabel nilai RGB dari citra 4x4

#### 3.5.2.1 RGB menjadi Grayscale

Untuk mengekstrak nilai tekstur, pertama kita akan mengubah citra tersebut dari RGB menjadi *grayscale* dengan menggunakan rumus berikut:

$$\text{Grayscale} = (0.2989 \times \text{Red}) + (0.5870 \times \text{Green}) + (0.1140 \times \text{Blue})$$

Pemilihan rumus konversi RGB ke *grayscale* ini didasarkan pada analisis kritis terhadap beberapa alternatif yang tersedia. *Formula luminance* dari standar ITU-R BT.601 dipilih karena mempertimbangkan respons non-linear mata manusia terhadap berbagai panjang gelombang cahaya, yang sangat relevan untuk deteksi anomali warna pada daun. Dengan bobot yang proporsional terhadap sensitivitas retina manusia, yaitu 0.5870 untuk hijau, 0.2989 untuk merah, dan 0.1140 untuk

biru. Formula ini memungkinkan pemisahan fitur patologis yang lebih presisi dibandingkan metode sederhana seperti rata-rata aritmatik RGB  $(R+G+B)/3$  yang mengabaikan perbedaan persepsi warna.

Pendekatan ini secara signifikan meningkatkan kualitas ekstraksi fitur tekstur pada tahap selanjutnya, terutama untuk mengidentifikasi perbedaan subtil pada pola gejala penyakit daun kelapa sawit yang seringkali sulit dibedakan dengan metode konversi konvensional. Meskipun metode ini memerlukan komputasi lebih kompleks dibandingkan fungsi bawaan *library* seperti OpenCV, pengorbanan efisiensi tersebut terbukti sepadan dengan peningkatan akurasi dalam konteks penelitian ini. Evaluasi komparatif yang dilakukan menunjukkan bahwa metode ini memberikan nilai kontras dan homogenitas GLCM yang lebih representatif pada citra daun dengan berbagai tingkat keparahan penyakit.

Berikut adalah contoh perhitungan yang dilakukan pada salah satu piksel dengan nilai RGB (155, 48, 201) dari citra yang didapatkan di atas, maka nilai *grayscale* dihitung sebagai berikut:

$$\text{Grayscale} = (0.2989 \times 155) + (0.5870 \times 48) + (0.1140 \times 201)$$

Berikut adalah hasil konversi citra RGB menjadi grayscale, sebagaimana ditampilkan pada Tabel 3.5 di bawah ini:

97.42	154.76	109.62	117.39
158.11	65.14	159.41	140.75
97.94	183.10	110.52	109.46
97.29	156.36	53.65	179.77

Tabel 3.5 Tabel nilai grayscale dari citra RGB

### 3.5.2.2 Kuantisasi Nilai Pixel

Setelah matriks citra dikonversi ke *grayscale*, langkah selanjutnya adalah melakukan kuantisasi nilai setiap piksel menggunakan metode kuantisasi uniform. Proses kuantisasi ini bertujuan untuk menyederhanakan rentang nilai piksel menjadi beberapa tingkat (level) tertentu agar perhitungan fitur tekstur pada tahap berikutnya menjadi lebih mudah dan efisien. Hasil kuantisasi nilai piksel *grayscale* dapat dilihat pada Tabel 3.6 di bawah.

1	2	1	1
2	1	2	2
1	2	1	1
1	2	0	2

Tabel 3.6 Kuantisasi Nilai Pixel Grayscale

### 3.5.2.3 Matriks *Co-occurrence*

Selanjutnya, matriks (*co-occurrence matrix*) dibangun untuk merepresentasikan hubungan spasial antara piksel referensi dan piksel tetangga dengan sudut 0 derajat serta jarak spasial 1. Hasil konstruksi matriks ini dapat dilihat pada Tabel 3.7 di bawah.

i,j	0	1	2	3
0	0	0	1	0
1	0	2	4	0
2	1	3	1	0
3	0	0	0	0

Tabel 3.7 Matriks *Co-occurrence*

### 3.5.2.4 Matriks Transpose dari Matriks *Co-occurrence*

Selanjutnya, hasil dari matriks *co-occurrence* di transpose, yang artinya hasil dari pertukaran baris dan kolom pada matriks *co-occurrence*. Matriks ini digunakan untuk menghitung fitur tekstur yang lebih lanjut. Hasil matriks transpose dapat dilihat pada Tabel 3.8 di bawah.

0	0	1	0
0	2	3	0
1	4	1	0
0	0	0	0

Tabel 3.8 Matriks Transpose

### 3.5.2.5 Matriks Simetris

Selanjutnya, hasil matriks yang sebelumnya di transpose, kemudian dijumlahkan dengan matriks *co-occurrence* yang telah dibangun sebelumnya. Hasil dari penjumlahan ini adalah matriks simetris yang digunakan untuk menghitung fitur tekstur lebih lanjut. Hasil dari matriks simetris dapat dilihat pada Tabel 3.9 di bawah.

0	0	1	0
0	2	3	0
1	4	1	0
0	0	0	0

Tabel 3.9 Matriks Simetris

### 3.5.2.6 Normalisasi Matriks Simetris

Matriks simetris yang telah dibangun sebelumnya kemudian dinormalisasi dengan membagi setiap elemen pada matriks dengan jumlah total elemen pada matriks tersebut. Hasil dari normalisasi ini adalah matriks probabilitas yang digunakan untuk menghitung fitur tekstur lebih lanjut. Hasil dari normalisasi matriks simetris dapat dilihat pada Tabel 3.11 di bawah.

$$\begin{aligned} \text{Total} &= 0 + 0 + 2 + 0 + 0 + 4 + 7 + 0 + 2 + 7 + 2 + 0 + 0 + 0 + 0 + 0 \\ \text{Total} &= 24 \end{aligned}$$

$\frac{0}{24}$	$\frac{0}{24}$	$\frac{2}{24}$	$\frac{0}{24}$
$\frac{0}{24}$	$\frac{4}{24}$	$\frac{7}{24}$	$\frac{0}{24}$
$\frac{2}{24}$	$\frac{24}{24}$	$\frac{24}{24}$	$\frac{24}{24}$
$\frac{2}{24}$	$\frac{7}{24}$	$\frac{2}{24}$	$\frac{0}{24}$
$\frac{0}{24}$	$\frac{0}{24}$	$\frac{0}{24}$	$\frac{0}{24}$
$\frac{0}{24}$	$\frac{24}{24}$	$\frac{24}{24}$	$\frac{24}{24}$

Tabel 3.10 Matriks Normalisasi

Berikut adalah hasil dari normalisasi matriks simetris yang telah dibangun sebelumnya. Hasil dari matriks normalisasi dapat dilihat pada Tabel 3.11 di bawah.

0	0	0.083	0
0	0.166	0.291	0
0.083	0.291	0.083	0
0	0	0	0

Tabel 3.11 Tabel Hasil Normalisasi

### 3.5.2.7 Menghitung Nilai Contrast, Correlation, Energy, dan Homogenitas

#### 1. Contrast

$$\text{Contrast} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_{i,j} (i - j)^2$$

$$\begin{aligned} \text{Contrast} = & (0 \times (0 - 0)^2) + (0 \times (0 - 1)^2) + (0.083 \times (0 - 2)^2) \\ & + (0 \times (0 - 3)^2) + (0 \times (1 - 0)^2) + (0.166 \times (1 - 1)^2) \\ & + (0.291 \times (1 - 2)^2) + (0 \times (1 - 3)^2) \\ & + (0.083 \times (2 - 0)^2) + (0.291 \times (2 - 1)^2) \\ & + (0.083 \times (2 - 2)^2) + (0 \times (2 - 3)^2) \\ & + (0 \times (3 - 0)^2) + (0 \times (3 - 1)^2) \\ & + (0 \times (3 - 2)^2) + (0 \times (3 - 3)^2) \end{aligned}$$

$$\begin{aligned} \text{Contrast} = & 0 + 0 + 0.332 + 0 + 0 + 0 + 0.291 + 0 + 0.332 \\ & + 0.291 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \end{aligned}$$

$$\text{Contrast} = 1.246$$

## 2. Correlation

$$\mu_i = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i \cdot P_{i,j}$$

$$\begin{aligned}
\mu_i &= (0 \times 0) + (0 \times 0.083) + (0 \times 0) + (1 \times 0) \\
&\quad + (1 \times 0.166) + (1 \times 0.291) + (1 \times 0) \\
&\quad + (2 \times 0.083) + (2 \times 0.291) + (2 \times 0.083) \\
&\quad + (2 \times 0) + (3 \times 0) + (3 \times 0) \\
&\quad + (3 \times 0) + (3 \times 0) \\
&= 0 + 0 + 0 + 0 \\
&\quad + 0.166 + 0.291 + 0 \\
&\quad + 0.166 + 0.582 + 0.166 \\
&\quad + 0 + 0 + 0 + 0 + 0 \\
&= 1.371
\end{aligned}$$

$$\mu_j = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} j \cdot P_{i,j}$$

$$\begin{aligned}\mu_j &= (0 \times 0) + (1 \times 0) + (2 \times 0.083) + (3 \times 0) \\&\quad + (0 \times 0) + (1 \times 0.166) + (2 \times 0.291) + (3 \times 0) \\&\quad + (0 \times 0.083) + (1 \times 0.291) + (2 \times 0.083) + (3 \times 0) \\&\quad + (0 \times 0) + (1 \times 0) + (2 \times 0) + (3 \times 0) \\&= 0 + 0 + 0.166 + 0 \\&\quad + 0 + 0.166 + 0.582 + 0 \\&\quad + 0 + 0.291 + 0.166 + 0 \\&\quad + 0 + 0 + 0 + 0 \\&= 1.371\end{aligned}$$

$$\sigma_i^2 = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu_i)^2 \cdot P_{i,j}$$

$$\begin{aligned}
\sigma_i^2 &= ((0 - 1.371)^2 \times 0) + ((0 - 1.371)^2 \times 0) \\
&\quad + ((0 - 1.371)^2 \times 0.083) + ((0 - 1.371)^2 \times 0) \\
&\quad + ((1 - 1.371)^2 \times 0) + ((1 - 1.371)^2 \times 0.166) \\
&\quad + ((1 - 1.371)^2 \times 0.291) + ((1 - 1.371)^2 \times 0) \\
&\quad + ((2 - 1.371)^2 \times 0.083) + ((2 - 1.371)^2 \times 0.291) \\
&\quad + ((2 - 1.371)^2 \times 0.083) + ((2 - 1.371)^2 \times 0) \\
&= 0.156010003 + 0.022848406 + 0.040050831 + 0.032838003 \\
&\quad + 0.115141671 + 0.032838003 \\
&= 0.399726917
\end{aligned}$$

$$\sigma_j^2 = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (j - \mu_j)^2 \cdot P_{i,j}$$

$$\begin{aligned}
\sigma_j^2 &= ((0 - 1.371)^2 \times 0) + ((1 - 1.371)^2 \times 0) \\
&\quad + ((2 - 1.371)^2 \times 0.083) + ((3 - 1.371)^2 \times 0) \\
&\quad + ((0 - 1.371)^2 \times 0) + ((1 - 1.371)^2 \times 0.166) \\
&\quad + ((2 - 1.371)^2 \times 0.291) + ((3 - 1.371)^2 \times 0) \\
&\quad + ((0 - 1.371)^2 \times 0.083) + ((1 - 1.371)^2 \times 0.291) \\
&\quad + ((2 - 1.371)^2 \times 0.083) + ((3 - 1.371)^2 \times 0) \\
&= 0.032838003 + 0.022848406 + 0.115141671 + 0.156010003 \\
&\quad + 0.040050831 + 0.032838003 \\
&= 0.399726917
\end{aligned}$$

$$\text{Correlation} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_{i,j} \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}$$

$$\begin{aligned}
\text{Correlation} &= 0 \times \frac{(0 - 1.371)(0 - 1.371)}{\sqrt{0.399 \cdot 0.399}} + 0 \times \frac{(0 - 1.371)(1 - 1.371)}{\sqrt{0.399 \cdot 0.399}} \\
&\quad + 0.083 \times \frac{(0 - 1.371)(2 - 1.371)}{\sqrt{0.399 \cdot 0.399}} + \dots \\
&\quad + 0 \times \frac{(3 - 1.371)(3 - 1.371)}{\sqrt{0.399 \cdot 0.399}} \\
&= -0.559
\end{aligned}$$

### 3. Energy

$$Energy = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (P_{i,j})^2$$

$$\begin{aligned}
Energy &= (0)^2 + (0)^2 + (0.083)^2 + (0)^2 \\
&\quad + (0)^2 + (0.166)^2 + (0.291)^2 \\
&\quad + (0)^2 + (0)^2 + (0.083)^2 \\
&\quad + (0.291)^2 + (0.083)^2 \\
&\quad + (0)^2 + (0)^2 + (0)^2 + (0)^2 \\
&= 0.006889 + 0.027556 + 0.084681 + 0.006889 \\
&\quad + 0.084681 + 0.006889 \\
&= 0.217185
\end{aligned}$$

### 4. Homogeneity

$$Homogeneity = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{P_{i,j}}{1 + (i - j)^2}$$

$$\begin{aligned}
Homogeneity &= \frac{0}{1 + (0 - 0)^2} + \frac{0}{1 + (0 - 1)^2} \\
&+ \frac{0.083}{1 + (0 - 2)^2} + \frac{0}{1 + (0 - 3)^2} \\
&+ \frac{0}{1 + (1 - 0)^2} + \frac{0.166}{1 + (1 - 1)^2} \\
&+ \frac{0}{1 + (1 - 2)^2} + \frac{0}{1 + (1 - 3)^2} \\
&+ \frac{0.291}{1 + (2 - 1)^2} + \frac{0.083}{1 + (2 - 2)^2} \\
&+ \frac{0}{1 + (2 - 3)^2} + \frac{0}{1 + (2 - 0)^2} \\
&+ \frac{0.291}{1 + (3 - 2)^2} + \frac{0}{1 + (3 - 0)^2} \\
&+ \frac{0}{1 + (3 - 1)^2} + \frac{0}{1 + (3 - 3)^2}
\end{aligned}$$

$$\begin{aligned}
Homogeneity &= 0.0166 + 0.166 + 0.1455 + 0.0166 \\
&+ 0.1455 + 0.083
\end{aligned}$$

$$Homogeneity = 0.5732$$

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil Penelitian**

Dataset yang digunakan dalam penelitian ini merupakan dataset citra daun bintik kelapa sawit yang diambil secara langsung dari PT. Perkebunan Nusantara IV Regional 7 Bekri. Dataset terdiri dari 340 citra daun bintik kelapa sawit yang terbagi ke dalam 5 kelas penyakit, yaitu 75 citra bercak daun, 95 citra berkerut, 65 citra berputar, 40 citra menggulung, dan 65 citra daun menguning.

##### **4.1.1 Akuisisi Dataset**

Akuisisi dataset merupakan tahap awal dalam penelitian ini. Dataset citra daun bintik kelapa sawit diperoleh dengan cara pengambilan gambar menggunakan kamera Canon EOS 700D pada latar belakang putih. Setiap gambar diambil dari sudut pandang yang seragam dan pencahayaan yang cukup agar kualitas citra optimal.

##### **4.1.2 Persiapan Dataset**

Pada tahapan ini, dataset dilakukan pemrosesan awal sebelum digunakan pada proses selanjutnya. Preprocessing dilakukan agar citra memiliki format dan struktur yang konsisten. Adapun langkah-langkah preprocessing yang dilakukan meliputi:

- 1. Resizing:** Semua citra diubah ukurannya menjadi dimensi yang seragam agar memudahkan pemrosesan lebih lanjut.

2. **Cropping:** Bagian-bagian citra yang tidak relevan atau kosong dipangkas untuk memfokuskan pada objek daun.
3. **Segmentasi Citra:** Segmentasi dilakukan untuk memisahkan objek daun dari latar belakang. Proses segmentasi dilakukan secara manual menggunakan platform *Phototoroom*, yang secara otomatis menghapus latar belakang putih dari citra.

#### 4.1.3 Augmentasi Dataset

Augmentasi data dilakukan untuk meningkatkan jumlah dan variasi data citra dengan teknik transformasi seperti rotasi, flipping, zooming, dan pencahayaan. Proses augmentasi ini bertujuan untuk meningkatkan kemampuan generalisasi model terhadap variasi data di dunia nyata.

#### 4.1.4 Ekstraksi Fitur

Setelah citra daun diperbanyak melalui proses augmentasi, langkah selanjutnya adalah mengekstrak fitur warna dan tekstur dari citra. Ekstraksi fitur bertujuan untuk mengubah citra menjadi representasi numerik yang dapat diproses oleh model pembelajaran mesin.

Fitur warna diekstraksi dengan menghitung rata-rata nilai RGB dari citra, sementara fitur tekstur diperoleh menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Berikut adalah fungsi yang digunakan untuk ekstraksi fitur warna dalam Python:

```

● ● ● Extract RGB
1 def extract_rgb(image):
2     """
3         Extract normalized RGB values from image, ignoring transparent pixels.
4         Args:
5             image: numpy array of shape (height, width, 4) representing RGBA image
6         Returns:
7             tuple of normalized (r, g, b) values
8             """
9         # Convert to double precision
10        image = image.astype(np.float64)
11
12        # Create mask for non-transparent pixels (alpha > 0)
13        mask = image[:, :, 3] > 0
14
15        if image.shape[2] == 4: # RGBA image
16            # Create mask for non-transparent pixels (alpha > 0)
17            mask = image[:, :, 3] > 0
18
19            # Calculate mean of RGB values only where mask is True
20            r_mean = np.mean(image[:, :, 0][mask])
21            g_mean = np.mean(image[:, :, 1][mask])
22            b_mean = np.mean(image[:, :, 2][mask])
23
24        else: # RGB image
25            # Calculate mean of RGB values directly
26            r_mean = np.mean(image[:, :, 0])
27            g_mean = np.mean(image[:, :, 1])
28            b_mean = np.mean(image[:, :, 2])
29
30        # Calculate mean of RGB values only where mask is True
31        r_mean = np.mean(image[:, :, 0][mask])
32        g_mean = np.mean(image[:, :, 1][mask])
33        b_mean = np.mean(image[:, :, 2][mask])
34
35        total = r_mean + g_mean + b_mean
36        return (r_mean/total, g_mean/total, b_mean/total)

```

Gambar 4.18 Contoh hasil ekstraksi fitur warna RGB

Ekstraksi Fitur Warna RGB Fungsi ini mengekstrak rata-rata nilai RGB dari citra daun. Jika citra memiliki channel alpha (RGBA), fungsi hanya menghitung nilai RGB dari area yang tidak transparan — dengan kata lain, hanya dari bagian daun yang merupakan *Region of Interest* (ROI). Setelah rata-rata tiap channel warna didapatkan, dilakukan normalisasi dengan membagi masing-masing nilai dengan jumlah total ketiga channel tersebut. Hasil akhirnya adalah nilai RGB terstandarisasi yang merepresentasikan distribusi warna utama dari citra daun.

```

Extract Texture
1 def extract_texture(image):
2     """
3     Extract GLCM texture features from grayscale image.
4     Args:
5         image: numpy array of shape (height, width) representing grayscale image
6     Returns:
7         tuple of (contrast, correlation, energy, homogeneity)
8     """
9     # Convert image to uint8 and scale to 0-255
10    if image.dtype != np.uint8:
11        image = (image * 255).astype(np.uint8)
12
13    # Reduce to 8 gray levels as MATLAB does by default
14    image = (image / 32).astype(np.uint8)
15
16    # Calculate GLCM with same offsets as MATLAB
17    glcm = graycomatrix(image, distances=[1], angles=[0, 45, 90, 135],
18                         levels=8, symmetric=True, normed=True)
19
20    # Calculate properties and average over all directions
21    contrast = np.mean(graycoprops(glcm, 'contrast'))
22    correlation = np.mean(graycoprops(glcm, 'correlation'))
23    energy = np.mean(graycoprops(glcm, 'energy'))
24    homogeneity = np.mean(graycoprops(glcm, 'homogeneity'))
25
26    return contrast, correlation, energy, homogeneity

```

Gambar 4.19 Contoh hasil ekstraksi tekstur

Ekstraksi Fitur Tekstur (GLCM) Fungsi ini mengekstrak fitur tekstur dari citra grayscale dengan menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM). Citra terlebih dahulu dikonversi menjadi 8 tingkat keabuan (*gray level*) untuk menyederhanakan informasi tekstur. GLCM dihitung pada empat arah sudut ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ ) dengan jarak 1 piksel. Selanjutnya, empat properti tekstur utama dihitung dari GLCM, yaitu kontras, korelasi, energi, dan homogenitas. Nilai dari masing-masing properti kemudian dirata-ratakan dari seluruh arah sudut untuk menghasilkan representasi tekstur yang lebih stabil.

#### 4.1.5 Memproses Gambar Daun menjadi Kerangka Data

```
● ● ● Process Leaf
1 def process_leaf_images(image_dict):
2     """
3         Process dictionary of leaf disease images and create
4         feature table.
5         Args:
6             image_dict: dictionary where keys are disease labels
7                 and values are lists of RGB image arrays
8             Returns:
9                 pandas DataFrame with RGB and GLCM features plus
10                labels
11                """
12
13    # Lists to store features and labels
14    features = []
15    labels = []
16
17    # Process each label and its images
18    for label, images in image_dict.items():
19        for img in images:
20            # Extract RGB features
21            r, g, b = extract_rgb(img)
22
23            # Convert to grayscale using OpenCV
24            # OpenCV expects BGR format, so convert RGB to BGR
25            first
26            bgr_img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
27            gray_img = cv2.cvtColor(bgr_img,
28                                   cv2.COLOR_BGR2GRAY)
29
30            # Extract texture features
31            contrast, correlation, energy, homogeneity =
32            extract_texture(gray_img)
33
34            # Append features and label
35            features.append([r, g, b, contrast, correlation,
36                             energy, homogeneity])
37            labels.append(label)
38
39            # Create DataFrame
40            columns = ['R', 'G', 'B', 'Contrast', 'Correlation',
41                       'Energy', 'Homogeneity']
42            df = pd.DataFrame(features, columns=columns)
43            df['Label'] = labels
44
45    return df
```

Gambar 4.20 Contoh hasil pemrosesan gambar daun menjadi kerangka data

Fungsi ini mengelola keseluruhan proses ekstraksi fitur dari dataset citra daun. Fungsi menerima sebuah *dictionary* dengan label penyakit sebagai *key* dan daftar citra RGB sebagai *value*. Untuk setiap citra, fitur warna RGB dan fitur tekstur GLCM diekstrak, kemudian disimpan bersama labelnya dalam sebuah tabel (*DataFrame*).

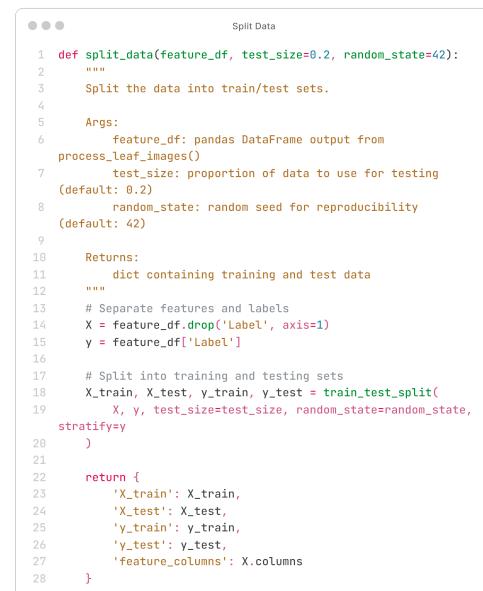
Tabel ini memiliki kolom sebagai berikut: **R, G, B, Contrast, Correlation, Energy, Homogeneity**, dan **Label**. Hasil akhirnya berupa

*DataFrame* yang siap digunakan sebagai input untuk pelatihan model.

#### 4.1.6 Pembagian Dataset

Setelah melalui proses ekstraksi, dataset yang diperoleh kemudian dibagi menjadi dua bagian: *data training* dan *data testing*. Rasio pembagian yang digunakan adalah 80:20, di mana 80% dari total data digunakan untuk melatih model, dan 20% sisanya digunakan untuk menguji performa model terhadap data yang belum pernah “dilihat” oleh model sebelumnya.

Berikut adalah kode yang digunakan untuk melakukan proses pembagian tersebut:



```

Split Data
1 def split_data(feature_df, test_size=0.2, random_state=42):
2     """
3         Split the data into train/test sets.
4
5         Args:
6             feature_df: pandas DataFrame output from
7                 process_leaf_images()
8             test_size: proportion of data to use for testing
9                 (default: 0.2)
10            random_state: random seed for reproducibility
11                 (default: 42)
12
13        Returns:
14            dict containing training and test data
15        """
16
17        # Separate features and labels
18        X = feature_df.drop('Label', axis=1)
19        y = feature_df['Label']
20
21        # Split into training and testing sets
22        X_train, X_test, y_train, y_test = train_test_split(
23            X, y, test_size=test_size, random_state=random_state,
24            stratify=y
25        )
26
27        return {
28            'X_train': X_train,
29            'X_test': X_test,
30            'y_train': y_train,
31            'y_test': y_test,
32            'feature_columns': X.columns
33        }

```

Gambar 4.21 Contoh hasil pembagian dataset menjadi data training dan testing

#### 4.1.7 Pembuatan Model

Pada tahap ini, dilakukan pembuatan tiga jenis model machine learning berbasis algoritma *Gaussian Naive Bayes*. Masing-masing model dibangun dengan pendekatan berbeda.

#### 4.1.8 Original Naive Bayes

Model ini menggunakan fitur yang diperoleh secara langsung tanpa melalui proses optimasi atau seleksi fitur lanjutan. Tujuan dari model ini adalah untuk membandingkan performa dasar algoritma *Naive Bayes* sebelum dilakukan peningkatan dengan algoritma optimasi.



```

● ● ● Train Model
1 def train_model(X_train, y_train):
2     """
3         Train a Gaussian Naive Bayes classifier.
4
5     Args:
6         X_train: training features
7         y_train: training labels
8
9     Returns:
10        trained GaussianNB model
11    """
12    model = GaussianNB()
13    model.fit(X_train, y_train)
14    return model

```

Gambar 4.22 Contoh hasil ekstraksi tekstur

Model ini menggunakan parameter default dari *GaussianNB* tanpa melakukan tuning. Cocok digunakan sebagai *baseline* untuk dibandingkan dengan model yang telah dioptimasi.

#### 4.1.9 GA Optimized Naive Bayes

Model ini menggunakan pendekatan algoritma *Genetic Algorithm* (*GA*) untuk melakukan seleksi fitur yang paling relevan terhadap

klasifikasi. Dengan mengoptimasi subset fitur, model diharapkan dapat meningkatkan akurasi klasifikasi dibandingkan model dasar. Algoritma GA bekerja berdasarkan prinsip evolusi biologis, seperti seleksi alam dan genetika, untuk menemukan kombinasi fitur terbaik.

```

  Train Mode GA

1 def train_model_ga(data_dict):
2     """
3         Optimize GaussianNB var_smoothing using Genetic Algorithm.
4
5         Args:
6             data_dict: dictionary from split_data() containing
7             training/test data
8
9         Returns:
10            optimized GaussianNB model and best var_smoothing
11            value
12            """
13
14     # Define the objective function for GA
15     def objective_function(var_smoothing):
16         # GA might give array, ensure float
17         var_smoothing = float(var_smoothing) # Convert to
18         float
19
20         # Create and train model
21         model = GaussianNB(var_smoothing=var_smoothing)
22         model.fit(data_dict['X_train'], data_dict['y_train'])
23
24         # Use evaluate_model function for consistency
25         eval_results = evaluate_model(model,
26                                       data_dict['X_test'], data_dict['y_test'])
27         accuracy = eval_results['accuracy']
28
29         # GA minimizes, so return negative accuracy
30         return -accuracy
31
32     # Define algorithm parameters
33     varbound = np.array([[1e-12, 1e-6]]) # bounds for
34     var_smoothing
35
36     algorithm_param = {
37         'max_num_iteration': 100,
38         'population_size': 10,
39         'mutation_probability': 0.1,
40         'elit_ratio': 0.01,
41         'crossover_probability': 0.5,
42         'parents_portion': 0.3,
43         'crossover_type': 'uniform',
44         'max_iteration_without_improv': 20
45     }
46
47     # Initialize GA
48     optimizer = ga(function=objective_function, dimension=1,
49                    variable_type='real',
50                    variable_boundaries=varbound,
51                    algorithm_parameters=algorithm_param)
52
53     # Perform optimization
54     optimizer.run()
55
56     # Train final model with best parameters
57     best_var_smoothing = float(optimizer.best_variable[0]) #
58     Convert to float
59     best_model = GaussianNB(var_smoothing=best_var_smoothing)
60     best_model.fit(data_dict['X_train'], data_dict['y_train'])
61
62     return best_model, best_var_smoothing

```

Gambar 4.23 Contoh hasil training model dengan GA

Model ini menggunakan algoritma *Genetic Algorithm (GA)* untuk menemukan nilai terbaik dari hyperparameter `var_smoothing`, yang dapat meningkatkan akurasi klasifikasi.

```

● ● ● Train Mode PSO
1 def train_model_pso(data_dict):
2     """
3         Optimize GaussianNB var_smoothing using Particle Swarm
4         Optimization.
5
6         Args:
7             data_dict: dictionary from split_data() containing
8                 training/test data
9
10            Returns:
11                optimized GaussianNB model and best var_smoothing
12                value
13                """
14
15        # Define the objective function for PSO
16        def objective_function(var_smoothing):
17            # PSO gives array of particles, take first dimension
18            var_smoothing = float(var_smoothing[0]) # Convert to
19            float
20
21            # Create and train model
22            model = GaussianNB(var_smoothing=var_smoothing)
23            model.fit(data_dict['X_train'], data_dict['y_train'])
24
25            # Use evaluate_model function for consistency
26            eval_results = evaluate_model(model,
27                data_dict['X_test'], data_dict['y_test'])
28            accuracy = eval_results['accuracy']
29
30            # PSO minimizes, so return negative accuracy
31            return -accuracy
32
33        # Set up PSO optimizer
34        options = {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
35        bounds = (np.array([1e-12]), np.array([1e-6])) # bounds
36        for var_smoothing
37            optimizer = ps.single.GlobalBestPSO(n_particles=10,
38                dimensions=1, options=options, bounds=bounds)
39
40            # Perform optimization
41            best_cost, best_pos =
42            optimizer.optimize(objective_function, iters=100)
43
44            # Train final model with best parameters
45            best_var_smoothing = float(best_pos[0]) # Convert to
46            float
47            best_model = GaussianNB(var_smoothing=best_var_smoothing)
48            best_model.fit(data_dict['X_train'], data_dict['y_train'])
49
50        return best_model, best_var_smoothing

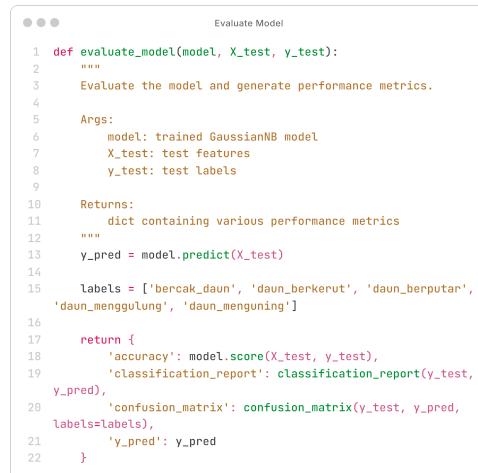
```

Gambar 4.24 Contoh training model dengan PSO

Model ini menggunakan algoritma *Particle Swarm Optimization* (*PSO*) yang bekerja dengan mensimulasikan pergerakan partikel dalam ruang solusi. Setiap partikel mewakili kemungkinan nilai `var_smoothing`. Model dengan akurasi terbaik dipilih sebagai solusi optimal.

#### 4.1.10 Evaluasi

Setelah model dilatih, langkah selanjutnya adalah mengevaluasi performanya menggunakan metrik seperti akurasi, laporan klasifikasi, dan *confusion matrix*.



```

  Evaluate Model

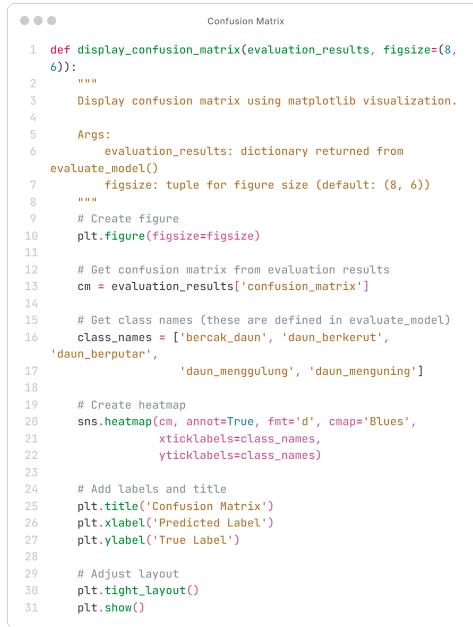
1  def evaluate_model(model, X_test, y_test):
2      """
3          Evaluate the model and generate performance metrics.
4
5      Args:
6          model: trained GaussianNB model
7          X_test: test features
8          y_test: test labels
9
10     Returns:
11         dict containing various performance metrics
12     """
13     y_pred = model.predict(X_test)
14
15     labels = ['bercak_daun', 'daun_berkerut', 'daun_berputar',
16               'daun_menggulung', 'daun_menguning']
17
18     return {
19         'accuracy': model.score(X_test, y_test),
20         'classification_report': classification_report(y_test,
21             y_pred),
22         'confusion_matrix': confusion_matrix(y_test, y_pred,
23             labels=labels),
24         'y_pred': y_pred
25     }

```

Gambar 4.25 Contoh hasil evaluasi model

Fungsi ini memberikan hasil evaluasi menyeluruh dari model, termasuk:

1. **Accuracy:** Persentase prediksi yang benar.
2. **Classification Report:** Menampilkan metrik *Precision*, *Recall*, dan *F1-score*.
3. **Confusion Matrix:** Matriks yang membandingkan klasifikasi aktual dengan prediksi model.



```

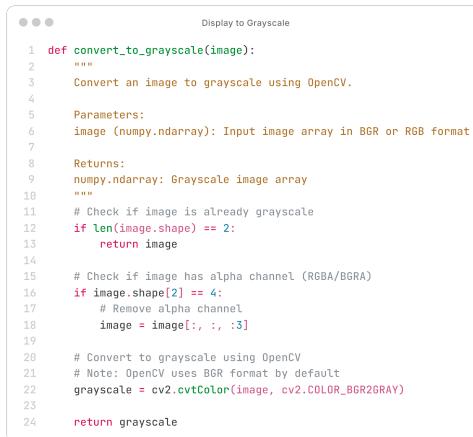
    Confusion Matrix

1 def display_confusion_matrix(evaluation_results, figsize=(8, 6)):
2     """
3         Display confusion matrix using matplotlib visualization.
4
5     Args:
6         evaluation_results: dictionary returned from
7             evaluate_model()
8             figsize: tuple for figure size (default: (8, 6))
9
10    # Create figure
11    plt.figure(figsize=figsize)
12
13    # Get confusion matrix from evaluation results
14    cm = evaluation_results['confusion_matrix']
15
16    # Get class names (these are defined in evaluate_model)
17    class_names = ['bercak_daun', 'daun_berkerut',
18                   'daun_berputar',
19                   'daun_menggulung', 'daun_menguning']
20
21    # Create heatmap
22    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
23                 xtickLabels=class_names,
24                 ytickLabels=class_names)
25
26    # Add labels and title
27    plt.title('Confusion Matrix')
28    plt.xlabel('Predicted Label')
29    plt.ylabel('True Label')
30
31    # Adjust layout
32    plt.tight_layout()
33    plt.show()

```

Gambar 4.26 Contoh hasil confusion matrix

Fungsi ini memvisualisasikan hasil prediksi dalam bentuk *heatmap*, sehingga mempermudah analisis kesalahan klasifikasi antar kelas.



```

    Display to Grayscale

1 def convert_to_grayscale(image):
2     """
3         Convert an image to grayscale using OpenCV.
4
5     Parameters:
6         image (numpy.ndarray): Input image array in BGR or RGB format
7
8     Returns:
9         numpy.ndarray: Grayscale image array
10
11    # Check if image is already grayscale
12    if len(image.shape) == 2:
13        return image
14
15    # Check if image has alpha channel (RGBA/BGRA)
16    if image.shape[2] == 4:
17        # Remove alpha channel
18        image = image[:, :, :3]
19
20    # Convert to grayscale using OpenCV
21    # Note: OpenCV uses BGR format by default
22    grayscale = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
23
24    return grayscale

```

Gambar 4.27 Contoh hasil konversi citra daun ke grayscale

```
● ● ● Saved Processed Image
1 def save_processed_images2(processed_images, output_path):
2     """
3         Save processed binary images to the specified output path.
4
5         Args:
6             > processed_images (dict): Dictionary of processed binary images as returned by
7                 batch_process_images.
8             > output_path (str): Path to save processed images.
9
10        """
11    for label, images in processed_images.items():
12        label_output_path = os.path.join(output_path, label)
13        os.makedirs(label_output_path, exist_ok=True)
14
15    for i, img in enumerate(images):
16        # Convert binary image (0/1) to proper image format (0/255)
17        img_to_save = (img * 255).astype(np.uint8)
18
19        # If the image is 2D (single channel), convert to 3 channels
20        if len(img_to_save.shape) == 2:
21            img_to_save = cv2.cvtColor(img_to_save, cv2.COLOR_GRAY2BGR)
22
23        img_name = f'{label}_{i}.png'
24        img_path = os.path.join(label_output_path, img_name)
25
26        # Save the image
27        success = cv2.imwrite(img_path, img_to_save)
28        if not success:
29            print(f"Warning: Failed to save image {img_name}")
30
print(f"Saved {len(images)} images for label: {label}")
```

Gambar 4.28 Contoh hasil pemrosesan gambar daun menjadi kerangka data

```
● ● ● Average RGB Values by Label
1 # Plotting
2 plt.figure(figsize=(10, 6))
3
4 # Set the positions for the bars
5 labels = grouped_features.index
6 x = range(len(labels))
7 width = 0.25 # Width of the bars
8
9 # Create bars for each color channel
10 plt.bar([i - width for i in x], grouped_features['R'], width, label='Red', colors='red', alpha=0.7)
11 plt.bar(x, grouped_features['G'], width, label='Green', colors='green', alpha=0.7)
12 plt.bar([i + width for i in x], grouped_features['B'], width, label='Blue', colors='blue', alpha=0.7)
13
14 # Customize the plot
15 plt.xlabel('Labels')
16 plt.ylabel('RGB Values')
17 plt.title('Average RGB Values by Label')
18 plt.xticks(x, labels)
19 plt.legend()
20
21 # Add value labels on top of each bar
22 for i in x:
23     plt.text(i - width, grouped_features['R'][i], f'{grouped_features["R"][i]:.1f}', ha='center', va='bottom')
24     plt.text(i, grouped_features['G'][i], f'{grouped_features["G"][i]:.1f}', ha='center', va='bottom')
25     plt.text(i + width, grouped_features['B'][i], f'{grouped_features["B"][i]:.1f}', ha='center', va='bottom')
26
27 # Adjust layout and display
28 plt.tight_layout()
29 plt.savefig('rgb_values_by_label.png')
30 plt.show()
```

Gambar 4.29 Contoh hasil rata-rata RGB dari citra daun

```

● ● ● Average Texture Feature by Label
1 # Plotting
2 plt.figure(figsize(12, 6)) # Made it slightly wider to accommodate 4 bars
3
4 # Set the positions for the bars
5 labels = grouped_features.index
6 x = range(len(labels))
7 width = 0.2 # Made bars slightly thinner since we have 4 features
8
9 # Create bars for each texture feature
10 plt.bar([i - 1.5*width for i in x], grouped_features['Contrast'], width,
11         label='Contrast', color='purple', alpha=0.7)
12 plt.bar([i - 0.5*width for i in x], grouped_features['Correlation'], width,
13         label='Correlation', color='blue', alpha=0.7)
14 plt.bar([i + 0.5*width for i in x], grouped_features['Energy'], width,
15         label='Energy', color='teal', alpha=0.7)
16 plt.bar([i + 1.5*width for i in x], grouped_features['Homogeneity'], width,
17         label='Homogeneity', color='brown', alpha=0.7)
18
19 # Customize the plot
20 plt.xlabel('Labels')
21 plt.ylabel('Texture Feature Values')
22 plt.title('Average Texture Features by Label')
23 plt.xticks(x, labels)
24 plt.legend()
25
26 # Add value labels on top of each bar
27 for i in x:
28     plt.text(i - 1.5*width, grouped_features['Contrast'][i],
29               f'{grouped_features["Contrast"][i]:.2f}', ha='center', va='bottom')
30     plt.text(i - 0.5*width, grouped_features['Correlation'][i],
31               f'{grouped_features["Correlation"][i]:.2f}', ha='center', va='bottom')
32     plt.text(i + 0.5*width, grouped_features['Energy'][i],
33               f'{grouped_features["Energy"][i]:.2f}', ha='center', va='bottom')
34     plt.text(i + 1.5*width, grouped_features['Homogeneity'][i],
35               f'{grouped_features["Homogeneity"][i]:.2f}', ha='center', va='bottom')
36
37 # Adjust layout and display
38 plt.tight_layout()
39 plt.savefig('texture_features_by_label.png')
40 plt.show()

```

Gambar 4.30 Contoh hasil rata rata ekstraksi fitur tekstur

```

Classification Metrics

1 def plot_classification_metrics(classification_dict):
2     """
3         Create a bar plot of classification metrics (precision, recall, f1-score) for each
4         class.
5
6         Args:
7             classification_dict: Dictionary containing classification metrics from
8                 evaluate_model
9
10        """
11    import matplotlib.pyplot as plt
12    import numpy as np
13
14    # Get the report text
15    report_text = classification_dict['classification_report']
16
17    # Split the report into lines
18    lines = report_text.split('\n')
19
20    # Initialize lists to store the data
21    labels = []
22    precision = []
23    recall = []
24    f1_score = []
25
26    # Parse each line of the report
27    for line in lines[2:-5]: # Skip header and footer lines
28        if line.strip(): # Skip empty lines
29            parts = line.strip().split()
30            # Handle cases where class names
31            # If len(parts) > 5, if class name contains spaces
32            if len(parts) > 5:
33                label = ' '.join(parts[:-4])
34                metrics = parts[-4:]
35            else:
36                label = parts[0]
37                metrics = parts[1:]
38
39            labels.append(label)
40            precision.append(float(metrics[0]))
41            recall.append(float(metrics[1]))
42            f1_score.append(float(metrics[2]))
43
44    # Plotting
45    plt.figure(figsize=(12, 6))
46
47    # Set the positions for the bars
48    x = range(len(labels))
49    width = 0.25 # Width of the bars
50
51    # Create bars
52    plt.bar([i - width for i in x], precision, width,
53            label='Precision', color='purple', alpha=0.7)
54    plt.bar([i + width, for i in x], recall, width,
55            label='Recall', color='orange', alpha=0.7)
56    plt.bar([i + width, for i in x], f1_score, width,
57            label='F1-score', color='teal', alpha=0.7)
58
59    # Customize the plot
60    plt.xlabel('Classes')
61    plt.ylabel('Scores')
62    plt.title('Classification Metrics by Class')
63    plt.xticks([i for i in x], labels, rotation=45, ha='right')
64    plt.legend()
65
66    # Add value labels on top of each bar
67    for i in x:
68        plt.text(i - width, precision[i],
69                    f'{precision[i]:.2f}', ha='center', va='bottom')
70        plt.text(i, recall[i],
71                    f'{recall[i]:.2f}', ha='center', va='bottom')
72        plt.text(i + width, f1_score[i],
73                    f'{f1_score[i]:.2f}', ha='center', va='bottom')
74
75    # Adjust layout and display
76    plt.tight_layout()
77    plt.ylim(0, 1.0, linestyle='-.', alpha=0.7)
78    plt.ylim(0, 1.0) # Set y-axis limit to 0-1.1 to accommodate labels
79    plt.savefig('classification_metrics.png')
80    plt.show()

```

Gambar 4.31 Contoh hasil classification metrics

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berisi kesimpulan dari hasil dan pembahasan terkait penelitian yang dilakukan, dapat juga berupa temuan yang Anda dapatkan setelah melakukan penelitian atau analisis terhadap tugas akhir Anda. Berhubungan dengan poin pada Subbab 1.2 dan 1.3.

1. Model klasifikasi ini dibangun menggunakan pendekatan ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) yang kemudian diklasifikasikan menggunakan algoritma *machine learning Naive Bayes*. Untuk memperoleh performa model yang optimal, penelitian dilakukan dengan menguji dua metode algoritma optimasi, yaitu *Genetic Algorithm* dan *Particle Swarm Optimization*. Dataset yang digunakan terdiri dari 5 kelas penyakit daun bibit kelapa sawit. Melalui proses optimasi tersebut, model mampu mengklasifikasikan penyakit pada daun bibit kelapa sawit berdasarkan fitur yang diperoleh model. Pendekatan ini menunjukkan bahwa kombinasi antara ekstraksi fitur dan algoritma *Naive Bayes* yang dioptimalkan dapat menjadi alternatif dalam mendeteksi penyakit pada daun kelapa sawit.
2. Berdasarkan hasil pengujian, model *Naive Bayes* yang dibangun dengan dua algoritma optimasi *Genetic Algorithm* dan *Particle Swarm Optimization* menghasilkan akurasi tertinggi sebesar 59%. Selain itu, nilai *precision*, *recall*, dan *F1-score* rata-rata berada pada

kisaran 58% hingga 59% untuk seluruh kelas penyakit daun bibit kelapa sawit. Meskipun performa tersebut masih tergolong rendah, pendekatan ini tetap menunjukkan efisiensi komputasi yang tinggi dan dapat dijadikan sebagai *baseline* awal untuk pengembangan metode klasifikasi yang lebih akurat di masa mendatang.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut adalah beberapa saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut:

1. Perlu dilakukan penambahan jumlah dataset citra pada setiap kelas agar distribusi data menjadi lebih seimbang dan model yang dikembangkan memiliki kemampuan generalisasi yang lebih baik dalam melakukan klasifikasi citra. Penambahan variasi data juga dapat membantu model mengenali karakteristik citra yang lebih beragam.
2. Disarankan untuk mengeksplorasi dan membandingkan algoritma optimasi lain selain Particle Swarm Optimization (PSO) dan Genetic Algorithm (GA), seperti Differential Evolution, Ant Colony Optimization, atau algoritma berbasis deep learning. Selain itu, penggunaan arsitektur model yang berbeda, misalnya Convolutional Neural Network (CNN) dengan konfigurasi yang lebih kompleks, dapat dipertimbangkan untuk meningkatkan akurasi dan performa klasifikasi.
3. Implementasi model klasifikasi citra yang telah dikembangkan pada platform aplikasi Android sangat direkomendasikan agar

hasil penelitian ini dapat dimanfaatkan secara lebih luas oleh masyarakat. Dengan adanya aplikasi mobile, pengguna dapat melakukan klasifikasi citra secara langsung dan praktis melalui perangkat smartphone, sehingga meningkatkan kemudahan dan aksesibilitas dalam penerapan teknologi ini di kehidupan sehari-hari.

## **DAFTAR PUSTAKA**

- [1] Ummu Habibah Gaffar et al. *Minyak kelapa dan minyak sawit: Dampak kesehatan, lingkungan, ekonomi dan sosial di balik produksi*. TOHAR MEDIA, 2024.
- [2] Hendra Maujana Saragih and Hanna Rahayu. “Pengaruh kebijakan Uni Eropa terhadap ekspor kelapa sawit Indonesia”. *JPPI (Jurnal Penelitian Pendidikan Indonesia)* 8.2 (2022), pp. 296–303.
- [3] Affan Hanif Imaduddin, Lucky Nugroho, and Dian Sugiarti. “Analisa Ekspor Minyak Kelapa Sawit Indonesia dan Isu Pembatasan Penggunaan Produk Minyak Kelapa Sawit oleh Uni Eropa”. *Trending: Jurnal Manajemen Dan Ekonomi* 1.3 (2023), pp. 163–181.
- [4] SYAFRIANDA SYAFRIANDA, IRA WAHYUNI, and NINGSIH ROZAINA. “ANALISIS KELAYAKAN FINANSIAL PERKEBUNAN KELAPA SAWIT SIKLUS KE-2 DAN INTEGRASINYA DENGAN USAHATANI JAGUNG DI DESA LAMPISI (Studi Kasus: KUD Sawit Kita)”. PhD thesis. AGRIBISNIS, 2024.
- [5] Lisnawita Lisnawita et al. “Peningkatan Kemampuan Tentang Penyakit Kelapa Sawit Akibat Serangan Patogen Ganoderma boninense Melalui Kegiatan Penyuluhan Lapangan”. *Journal Of Human And Education (JAHE)* 4.1 (2024), pp. 505–508.

- [6] Munawir Sajali Harahap et al. “Optimalisasi Strategi Pengendalian Bercak Daun Curvularia sp. pada Pembibitan Kelapa Sawit di Desa Binanga”. *Jurnal Masyarakat Indonesia (Jumas)* 3.02 (2024), pp. 73–79.
- [7] Agum Panogari Napitupulu. “TA: Pengendalian Penyakit Bercak Daun (Curvularia Sp.) Pada Main Nursery Kelapa Sawit (*Elaeis guineensis* Jacq.) Dengan Metode Sanitasi”. PhD thesis. Politeknik Negeri Lampung, 2022.
- [8] SHERLY MARCELIAN et al. “IDENTIFIKASI DAN PERSENTASE SERANGAN PATOGEN PENYAKIT PADA PEMBIBITAN UTAMA TANAMAN KELAPA SAWIT (*Elaeis guineensis* Jacq) DI DESA PEMATANG PAUH KECAMATAN TUNGKAL ULU KABUPATEN TANJUNG JABUNG BARAT”. PhD thesis. Universitas Batanghari Jambi, 2023.
- [9] DERIS PAKIDING et al. “IMPLEMENTASI COMPUTER VISION DALAM MENDETEKSI PENYAKIT PADA TANAMAN CABAI DAN TOMAT MENGGUNAKAN METODE CNN”. PhD thesis. Universitas Muhammadiyah Parepare, 2025.
- [10] Mohtar Yunianto et al. “Klasifikasi Kanker Paru Paru Menggunakan Naïve Bayes Dengan Variasi Filter Dan Ekstraksi Ciri Gray Level Co-Occurance Matrix (GLCM)”. *Indonesian Journal of Applied Physics* 11.2 (2021), pp. 256–268.

- [11] Dwi Elvira. “Klasifikasi Citra Daun Kelapa Sawit Yang Terkena Dampak Hama Menggunakan Metode K-Nearest Neighbor (KNN)”. PhD thesis. Universitas Islam Riau, 2021.
- [12] Husni Teja Sukmana, Dewi Khairani, et al. “Peningkatan performa klasifikasi jenis kopra menggunakan algoritma k-nearest neighbour berdasarkan eksstrasi fitur”. B.S. thesis. Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
- [13] M Afriansyah et al. “Algoritma Naive Bayes Yang Efisien Untuk Klasifikasi Buah Pisang Raja Berdasarkan Fitur Warna”. *Journal of Information Systems Management and Digital Business* 1.2 (2024), pp. 236–248.
- [14] Septian Arie Prayoga, Ismasari Nawangsih, and Tri Ngudi Wiyatno. “Implementasi metode naïve bayes classifier untuk identifikasi jenis jamur”. *Pelita Teknologi* 14.2 (2019), pp. 134–144.
- [15] Ardi Nainggolan et al. “Identifikasi Penyakit Tanaman Tomat Berdasarkan Citra Penyakit Menggunakan Metode GLCM dan Naïve Bayes Classifier”. *Methotika: Jurnal Ilmiah Teknik Informatika* 2.1 (2022), pp. 22–28.
- [16] Felicia Febriana et al. “Perbandingan Klasifikasi Naive-Bayes dan KNN untuk Mengidentifikasi Jenis Buah Apel dengan Ekstraksi Ciri LBP dan HSV”. *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*. Vol. 2. 2. 2021, pp. 191–201.

- [17] Made Agus Dwiputra, I Gede Pasek Suta Wijaya, and Ramaditia Dwiyansaputra. “Perancangan Mesin Klasifikasi Menggunakan Particle Swarm Optimization”. *Journal of Computer Science and Informatics Engineering (J-Cosine)* 8.2 (2024).
- [18] Hester Patmawati and Yohanes Anton Nugroho. “OPTIMALISASI RUTE DISTRIBUSI MATRAS SPRING BED DENGAN MENGGUNAKAN ALGORITMA GENETIKA PADA PENYELESAIAN CAPACITATED VEHICLE ROUTING PROBLEM” ().
- [19] Volkan Soner Özsoy, Mehmet Güray Ünsal, and H Hasan Örkçü. “Use of the heuristic optimization in the parameter estimation of generalized gamma distribution: comparison of GA, DE, PSO and SA methods”. *Computational Statistics* 35.4 (2020), pp. 1895–1925.
- [20] Romtika Manurung, Volvo Sihombing, and Mila Nirmala Sari Hasibuan. “Implementasi Deep Learning untuk Menentukan Harga Buah Sawit”. *INFORMATIKA* 12.3 (2024), pp. 427–436.
- [21] Sherin Junisthia, Yogi Isro’Mukti, and Debi Gusmaliza. “Integrasi Particle Swarm Optmization Dengan Naïve Bayes Untuk Memprediksi Tanaman Cabai”. *Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence)* 3.1 (2023), pp. 42–46.
- [22] Dadang Iskandar Mulyana, Herdiyansyah herdi Hartanto, and Mesra Betty Yel. “Optimasi Klasifikasi Bunga Kantong

- Semar Dengan Menggunakan Algoritma Naïve Bayes, Data Augmentasi Dan PSO”. *Swabumi* 10.2 (2022), pp. 167–174.
- [23] Evy Priyanti. “Peningkatan Algoritma Naïve Bayes Menggunakan Algoritma Genetika pada Klasifikasi Bakteri”. *Jurnal Khatulistiwa Informatika* 9.2 (Dec. 2021), pp. 85–88.
- [24] Sayyidatul Lutfiah Abdullah, Anisah Fadliyah Akbariyah, and Rinandita Wikansari. “Potensi Ekspor Crude Palm Oil (Cpo) Di Indonesia”. *Journal of Science and Social Research* 7.1 (2024), pp. 61–67.
- [25] Dina Sri Rejeki. “ANALISIS FAKTOR YANG MEMPENGARUHI VOLUME EKSPOR CRUDE PALM OIL (CPO) INDONESIA DI PASAR INTERNASIONAL”. PhD thesis. UNIVERSITAS JAMBI, 2023.
- [26] Dewa Ayu Puspitawati, Abd Azis Muthalib, and Irmawatty P Tamburaka. “ANALISIS PRODUKTIVITAS TENAGA KERJA KELAPA SAWIT PT. DAMAI JAYA LESTARI DI KECAMATAN WIWIRANO KABUPATEN KONAWE UTARA”. *JURNAL EKONOMI* 9.3 (2024), pp. 97–103.
- [27] Andy Christian and ST ST & Wasis Setiadi. *Industri Oleokimia Berbasis Kelapa Sawit*. Rasibook, 2019.
- [28] Ria Kurniasih, Raden Hanung Ismono, and Teguh Endaryanto. “Sistem replanting kelapa sawit, opportunity pendapatan kelapa sawit dan tingkat kesejahteraan petani pada masa replanting di perkebunan kelapa sawit rakyat kabupaten Lampung Tengah”. *Jurnal Ilmu-Ilmu Agribisnis* 9.2 (2021), pp. 309–316.

- [29] Andra Rizky Muflihani et al. “Analisis Sistem Agribisnis Pada Tanaman Kelapa Sawit Rakyat: Analysis Of Agribusiness Systems In Smallholder Palm Oil Plantations”. *Journal: Agricultural Review* 3.2 (2024), pp. 82–95.
- [30] Elizabeth Lalang, Helda Syahfari, and Noor Jannah. “Inventarisasi Penyakit Bercak Daun (*Curvularia sp.*) Di Pembibitan Kelapa Sawit PT Ketapang Hijau Lestari–2 Kampung Abit Kecamatan Mook Manaar Bulatn Kabupaten Kutai Barat. Penelitian ini bertujuan untuk mengetahui frekuensi dan intensitas serangan penya”. *Agrifor* 15.1 (2016), pp. 23–28.
- [31] Eni Suhesti, Ervayenri Ervayenri, et al. “Analisis Tingkat Kerusakan Serangan Hama dan Penyakit Dipersemaian BPDASHL Indragiri Rokan Pekanbaru”. *Wahana Forestra: Jurnal Kehutanan* 17.1 (2022), pp. 85–101.
- [32] Delia Agustina, Cahya Prihatna, and Antonius Suwanto. “Rapid inoculation technique and biological control of leaf spot disease in oil palm”. *International Journal of Oil Palm* 2.1 (2019), pp. 1–11.
- [33] Ferina Afriliya. “Keanekaragaman Jenis-Jenis Penyakit dan Cara Pengendaliannya di Pembibitan Kelapa Sawit (*Elaeis Guinensis* Jacq) PT. Perkebunan Nusantara I Langsa”. *Biologica Samudra* 1.1 (2019), pp. 34–40.
- [34] Mario Malado et al. *Pengendalian Hama dan Penyakit Tanaman Pertanian*. CV. Gita Lentera, 2024.

- [35] Yuni Sri Rahayu and Sari Kusuma Dewi. “Penyakit Tanaman Akibat Defisiensi Unsur Hara” (2023).
- [36] Helda Syahfari and A Syamad Ramayana. *Buku Ajar Dasar-Dasar Perlindungan Tanaman*. Penerbit NEM, 2024.
- [37] Nur Rokhimah Hanik, Rindi Diah Ayu Fitriani, and Desty Oktavianingtyas. “Identification of Pests and Diseases on Anthurium Jemani Ornamental Plants in the Ngledoksari Tawangmangu Tourism”. *Jurnal Biologi Tropis* 24.1 (2024), pp. 563–571.
- [38] TENGKU IRWAN AZIZ et al. “ANALISIS ALGORITMA MACHINE LEARNING NAIVE BAYES UNTUK KLASIFIKASI PENYAKIT PADA DATA PASIEN DI PUSKESMAS GUNUNG SELAMAT”. PhD thesis. Universitas Labuhanbatu, 2024.
- [39] Alvina Felicia Watratan, Dikwan Moeis, et al. “Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia”. *Journal of Applied Computer Science and Technology* 1.1 (2020), pp. 7–14.
- [40] Husin Muhamad et al. “Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris”. 2017.
- [41] Fahrur Riza, Safwandi Safwandi, and Said Fadlan. “Klasifikasi Varietas Kopi Berdasarkan Kondisi Tanah dan Suhu Menggunakan Algoritma Gaussian Naïve Bayes”. *Jurnal Algoritma* (2025).

- [42] Hani Zulfia Zahro', Santi Wahyuni, and Febriana. "OPTIMASI RUTE PENGANTARAN PAKET MENGGUNAKAN METODE GENETIC ALGORITHM (GA)". *Industri Inovatif: Jurnal Teknik Industri* 10.1 (2020), pp. 41–44.
- [43] Nadio Rodicky dan Sopian Soim dan Suzan Zefi. "SELEKSI FITUR DENGAN GENETIKA ALGORITHM UNTUK OPTIMASI RANDOM FOREST DALAM MENGKLASIFIKASI TENTANG JENIS PENGGUNAAN INTERNET". *Transmisi: Jurnal Ilmiah Teknik Elektro* 27.3 (2025).
- [44] Harim Adi Saputro, Wayan Firdaus Mahmudy, and Candra Dewi. "Implementasi algoritma genetika untuk optimasi penggunaan lahan pertanian". *Jurnal Mahasiswa PTIIK* 5.12 (2015), p. 12.
- [45] Rahmad Naufal and Muhammad Siddik Hasibuan. "Optimization of Distribution Routes Using the Genetic Algorithm in the Traveling Salesman Problem". *Journal of Applied Informatics and Computing* 9.1 (2025), pp. 211–220.
- [46] Radomil Matousek. "Genetic Algorithm and Advanced Tournament Selection Concept". *Nature Inspired Cooperative Strategies for Optimization*. 2008.
- [47] Joseph L. Pachuau, Arnab Roy, and Anish Kumar Saha. "An Overview of Crossover Techniques in Genetic Algorithm". *International Conference on Modeling, Simulations and Optimizations*. 2020.

- [48] Ismi Fadhilah, Yurika Permanasari, and Erwin Hotmansyah Harahap. “Representasi Matriks untuk Proses Crossover Pada Algoritma Genetika untuk Optimasi Travelling Salesman Problem”. 2016.
- [49] Agus Muliantara. “ANALISIS DAN IMPLEMENTASI PENJADWALAN DENGAN MENGGUNAKAN PENGEMBANGAN MODEL CROSSOVER DALAM ALGORITMA GENETIKA”. 2013.
- [50] Erianto Ongko. “Analisis Pengaruh Mutasi Terhadap Performance Algoritma Genetika”. *Jurnal Teknik Informatika Kaputama (JTIK)* 1.1 (2017), pp. 46–51. 2548-9704.
- [51] Mudita Juneja and Shyam Krishna Nagar. “Particle swarm optimization algorithm and its parameters: A review”. *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)* (2016), pp. 1–5.
- [52] Budi Santosa. “TUTORIAL PARTICLE SWARM OPTIMIZATION”. 2006.
- [53] Xingjuan Cai et al. “Individual Parameter Selection Strategy for Particle Swarm Optimization”. *Particle Swarm Optimization*. Ed. by Aleksandar Lazinica. Rijeka: IntechOpen, 2009. Chap. 5.
- [54] Wei-Chang Yeh. “A Simple Hybrid Particle Swarm Optimization”. *Advances in Evolutionary Algorithms* (2008), pp. 115–126.
- [55] Rohman Dijaya and Hamzah Setiawan. “Buku ajar pengolahan citra digital”. *Umsida Press* (2023), pp. 1–85.

- [56] Rizal Adha Saputra. “Perbandingan Ruang Warna RGB, HSV Dan Ycber Untuk Segmentasi Citra Ikan Kembung Menggunakan K-Means Clustering”. *JATI (Jurnal Mahasiswa Teknik Informatika)* ().
- [57] Elok Faiqotul Himmah, Maura Widyaningsih, and Maysaroh Maysaroh. “Identifikasi kematangan buah kelapa sawit berdasarkan warna RGB dan HSV menggunakan metode K-Means clustering”. *Jurnal Sains Dan Informatika* 6.2 (2020), pp. 193–202.
- [58] Hurriyatul Fitriyah and Randy Cahya Wihandika. *Dasar-Dasar Pengolahan Citra Digital*. Universitas Brawijaya Press, 2021.
- [59] Muhammad Ijlal Nurhadi et al. “SISTEM ESTIMASI MASSA TANDAN BUAH SEGAR KELAPA SAWIT MENGGUNAKAN CITRA UNMANNED AERIAL VEHICLE DENGAN MODEL RANDOM FOREST REGRESSION= Fresh Fruit Bunch Mass Estimation System for Oil Palm Using Unmanned Aerial Vehicle Imagery with Random Forest Regression Model”. PhD thesis. Universitas Hasanuddin, 2024.
- [60] Faridawaty Marpaung, Fitrahuda Aulia, Rinjani Cyra Nabila, et al. *Computer Vision Dan Pengolahan Citra Digital*. 2022.
- [61] Rahmi Rahmi Burhanuddin. “KLASIFIKASI penyakit padi melalui citra daun menggunakan metode naive bayes”. *Jurnal Informatika dan Teknik Elektro Terapan* 12.2 (2024).

- [62] Antoni Pribadi and Ade Kurniawan. “Deteksi Penyakit Sawit Menggunakan Metode Deep Learning”. *Jurnal Sains Dan Ilmu Terapan* 5.2 (2022), pp. 72–76.
- [63] Ananta Dwi Prayoga Alwy et al. “Klasifikasi penyakit pada padi dengan ekstraksi fitur LBP dan GLCM”. *Journal of Deep Learning, Computer Vision and Digital Image Processing* (2023), pp. 1–10.
- [64] Yohannes Yohannes et al. “Klasifikasi Motif Songket Palembang menggunakan Support Vector Machine berdasarkan Histogram of Oriented Gradients”. *Jurnal Teknologi Terpadu* 9.2 (2023), pp. 143–149.
- [65] Daurat Sinaga and Cahaya Jatmoko. “KLASIFIKASI CITRA BATIK SUMATERA MENGGUNAKAN NAÏVE BAYES BERBASIS FITUR EKSTRAKSI GLCM”. *Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK)*. Vol. 8. 01. 2024.
- [66] Mhd Hanafi, Hendra Setiawan, and Yessi Jusman. “Analisis Fitur Citra untuk Deteksi Kanker Prostat Menggunakan GLCM dan T-Test”. *Medika Teknika: Jurnal Teknik Elektromedik Indonesia* 6.2 (2025), pp. 62–81.
- [67] Rosita Diana, Heri Warni, and Tata Sutabri. “Penggunaan teknologi machine learning untuk pelayanan monitoring kegiatan belajar mengajar pada SMK Bina Sriwijaya Palembang”. *Jurnal Teknik Informatika (JUTEKIN)* 11.1 (2023).

- [68] FIRMANSYAH MUKTI WIJAYA. “IMPLEMENTASI ALGORITMA ADAPTIVE THRESHOLDING DAN K-MEANS CLUSTERING SEBAGAI METODE SEGMENTASI CITRA DAUN BAWANG MERAH HALAMAN JUDUL SKRIPSI” () .
- [69] Septiana Vratiwi. “Penerapan Metode Naïve Bayes Pada Sistem Penunjang Keputusan Bibit Unggul Kelapa Sawit”. *Jurnal Pustaka AI Vol 4.2* (2024), pp. 31–37.
- [70] Muhammad Rijal, Andi Muhammad Yani, and Abdul Rahman. “Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan Deep Learning dengan Model CNN”. *Jurnal Teknologi Terpadu* 10.1 (2024), pp. 56–62.
- [71] Saepudin Saepudin et al. “ANALISIS KINERJA YOLOV8 OPTIMALISASI ROBOFLOW UNTUK DETEKSI EKSPRESI WAJAH EMOSIONAL DENGAN MACHINE LEARNING”. *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika* 6.2 (2024), pp. 115–124.
- [72] Eka Dyar Wahyuni, Reisa Permata Sari, et al. “Transformasi Data Statistik Menjadi Visual Interaktif Menggunakan Streamlit: Studi Kasus BPS Kota Mojokerto”. *Jurnal Kridatama Sains dan Teknologi* 6.02 (2024), pp. 804–822.

## **LAMPIRAN**

