

**KLASIFIKASI TINGKAT KEMATANGAN LADA MENGGUNAKAN
ENSEMBLE LEARNING BERDASARKAN CITRA
WARNA KULIT**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana
Komputer (S.Kom) Program Studi Informatika



JIHAN IZZATHUL MUJIDAH

105841104021

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR**

2025



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK



PENGESAHAN

Skripsi atas nama Jihan Izzathul Mujidah dengan nomor induk Mahasiswa 105841104021, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 1 Agustus 2025.

Panitia Ujian :

Pengawas Umum

Makassar, 6 Rabi'ul Awwal 1447 H
30 Agustus 2025 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Muhammadiyah Makassar

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng

Penguji

a. Ketua : Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Fahim Irfan Rachman, S.Kom., M.T.

Anggota : 1. Chyquitha Danuputri, S.Kom., M.Kom.

2. Titin Wahyuni, S.Pd., M.T.

3. Ir. Muhammad Faisal, S.Si., M.T., Ph.D., IPM

Mengetahui :

Pembimbing I

Pembimbing II

Rizki Yusliana Bakti, S.T., MT.

Lukman, S.Kom., M.T.

Dekan



Ir. Muhy. Swifati S. Kuba, S.T., M.T.
NIM : 795 288





HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **KLASIFIKASI TINGKAT KEMATANGAN LADA MENGGUNAKAN ENSEMBLE LEARNING BERDASARKAN CITRA WARNA KULIT**

Nama : Jihan Izzathul Mujidah

Stambuk : 105 841104021

Makassar, 30 Agustus 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II


Rizki Yusliana Bakti, S.T., MT.


Lukman, S.Kom., M.T.

Mengetahui,
Ketua Prodi Informatika


Rizki Yusliana Bakti, S.T., M.T.
NBM : 1307 284



ABSTRAK

JIHAN IZZATHUL MUJIDAH, Klasifikasi Tingkat Kematangan Lada Menggunakan Ensemble Learning Berdasarkan Citra Warna Kulit. (dibimbing oleh Rizki Yusliana Bakti, S.T., M.T dan Lukman, S.Kom.,M.T)

Penelitian ini bertujuan untuk membangun model klasifikasi tingkat kematangan lada (*Piper nigrum* L.) berdasarkan citra warna kulit menggunakan metode ensemble learning. Lada merupakan komoditas penting di Sulawesi Selatan, dengan kualitas dan nilai ekonomi yang sangat dipengaruhi oleh tingkat kematangan buah. Data citra lada yang digunakan dalam penelitian ini terdiri dari empat kelas kematangan, yaitu belum matang, setengah matang, matang, dan lewat matang, yang dikumpulkan langsung dari perkebunan menggunakan kamera digital beresolusi tinggi. Dari setiap citra, diekstraksi fitur warna dalam ruang HSV (Hue, Saturation, Value) yang efektif merepresentasikan variasi warna kulit lada. Model ensemble yang menggabungkan Random Forest dan XGBoost melalui Soft Voting diterapkan untuk mengenali pola fitur warna dan menghasilkan klasifikasi tingkat kematangan. Hasil pengujian menunjukkan bahwa model ensemble mencapai akurasi 98,25%, presisi 98,27%, recall 98,35%, dan F1-score 98,26%, yang menunjukkan performa klasifikasi tinggi dan stabil antar kelas. Temuan ini menegaskan bahwa metode ensemble learning efektif dalam klasifikasi komoditas pertanian berbasis citra, serta membuka peluang pengembangan metode yang lebih adaptif dan akurat untuk mendukung pengolahan citra pada komoditas lain.

Kata Kunci : Klasifikasi, Kematangan Lada, *Ensemble Learning*, *Random Forest*, XGBoost.

ABSTRACT

JIHAN IZZATHUL MUJIDAH, *Classification of Pepper Ripeness Using Ensemble Learning Based on Fruit Skin Color Images (Supervised by Rizki Yuslima Bakti, S.T., M.T. and Lukman, S.Kom., M.T.)*

*This study aims to develop a classification model for the ripeness level of black pepper (*Piper nigrum* L.) based on skin color images using the ensemble learning method. Pepper is an important commodity in South Sulawesi, with its quality and economic value strongly influenced by the fruit's ripeness level. The pepper image dataset used in this study consists of four ripeness classes, namely unripe, half-ripe, ripe, and overripe, collected directly from plantations using a high-resolution digital camera. From each image, color features were extracted in the HSV (Hue, Saturation, Value) color space, which effectively represents the variation in pepper skin color. The ensemble model combining Random Forest and XGBoost through Soft Voting was applied to recognize color feature patterns and classify ripeness levels. The experimental results showed that the ensemble model achieved an accuracy of 98.25%, precision of 98.27%, recall of 98.35%, and F1-score of 98.26%, indicating high and stable classification performance across classes. These findings confirm that ensemble learning is effective for image-based classification of agricultural commodities and open opportunities for the development of more adaptive and accurate methods to support image processing in other commodities.*

Keywords: *classification, pepper ripeness, ensemble learning, Random Forest, XGBoost.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullah Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wata'ala atas segala rahmat dan karunia-Nya sehingga skripsi yang berjudul **“Klasifikasi Tingkat Kematangan Lada Menggunakan *Ensemble Learning* Berdasarkan Citra Warna Kulit”** dapat disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Makassar. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan guna penyempurnaan ke depannya.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, baik moril maupun materil, selama proses penyusunan skripsi ini. Ucapan terima kasih secara khusus penulis sampaikan kepada:

1. Bapak Dr. Ir. Hj. Abd. Rakhim Nanda, S.T., M.T., IPU, selaku Rektor Universitas Muhammadiyah Makassar.
2. Bapak Muh. Syafaat S Kuba, S.T., M.T, sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
3. Ibu Rizki Yusliana Bakti, S.T., M.T, selaku Ketua Program Studi Informatika, Dosen Pembimbing 1 Skripsi, serta Dosen Pembimbing Akademik, terima kasih atas bimbingan, koreksi, saran, dan motivasi Ibu yang sangat membantu penyusunan skripsi ini maupun dalam perjalanan akademik penulis.
4. Bapak Lukman, S.Kom., M.T, Selaku Dosen Pembimbing 2 Skripsi, terima kasih atas bimbingan, koreksi, saran, dan motivasi Bapak yang sangat membantu penyusunan skripsi ini.
5. Bapak dan Ibu dosen pengajar, yang telah memberikan ilmu, bimbingan, dan pengalaman berharga selama proses studi. Ilmu yang Bapak dan Ibu

berikan menjadi bekal penting bagi penulis untuk dapat menyelesaikan penelitian ini dengan baik.

6. Bapak/Ibu Staff Administrasi Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.
7. Kedua Orang Tua Saya, penulis ucapkan banyak terima kasih kepada ayahanda Nasrul dan Ibunda Aminah yang selalu menjadi sumber semangat, doa, dan cinta tanpa batas. Terima kasih atas doa, cinta, dan dukungan tanpa batas yang selalu menyertai setiap langkah penulis. Meski tidak meraih gelar sarjana, dengan ketulusan hati dan kerja keras telah mengantarkan penulis hingga meraih pendidikan tinggi. Kasih sayang dan pengorbanan beliau menjadi pijakan terkokoh sekaligus cahaya yang menerangi setiap langkah penulis.
8. Bapak Yusuf Yasir, S.IP, beserta keluarga, terimakasih telah mengizinkan dan mendukung penelitian ini di perkebunan ladanya, sehingga penelitian dapat terlaksana dengan baik dan lancar.
9. Teman-teman angkatan 21 kelas B, terkhusus penulis menyampaikan terima kasih kepada Sarina, Rizky Maulia, dan Nur Annisa Syarifuddin atas dukungan, motivasi, serta kebersamaan yang diberikan selama perkuliahan dan penyusunan skripsi ini.

Semoga Allah SWT membalas kebaikan semua pihak-pihak yang membantu penulis dalam menyusun skripsi ini. Penulis berharap agar skripsi ini dapat memberikan kontribusi yang bermanfaat bagi perkembangan ilmu pengetahuan terutama dalam bidang informatika aamiin.

“Billahi fii sabilil haq fastabiqul khairat”

Wassalamu’alaikum warahmatullahi wabarakatuh

Makassar, 30 Agustus 2025

Jihan Izzathul Mujidah

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK.....	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR.....	v
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN	xi
DAFTAR ISTILAH	xii
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah.....	3
C. Tujuan Penelitian.....	3
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian.....	4
F. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	6
A. Landasan Teori	6
B. Penelitian Tekait.....	20
C. Kerangka Berpikir.....	23
BAB III METODE PENELITIAN.....	24
A. Tempat dan Waktu Penelitian.....	24

B. Alat dan Bahan.....	24
C. Perancangan Sistem	24
D. Teknik Pengujian Sistem.....	29
E. Teknik Analisis Data	30
BAB IV HASIL DAN PEMBAHASAN.....	33
A. Sumber Data.....	33
B. Pelabelan Data	33
C. Preprocessing Data	34
D. Ekstraksi Fitur Color Moments HSV	36
E. Pembagian Data	39
F. Pengujian Model	40
G. Hasil Pengujian Model.....	45
H. Evaluasi kinerja.....	63
BAB V PENUTUP.....	65
A. Kesimpulan	65
B. Saran.....	65
DAFTAR PUSTAKA.....	67
Lampiran	74

DAFTAR GAMBAR

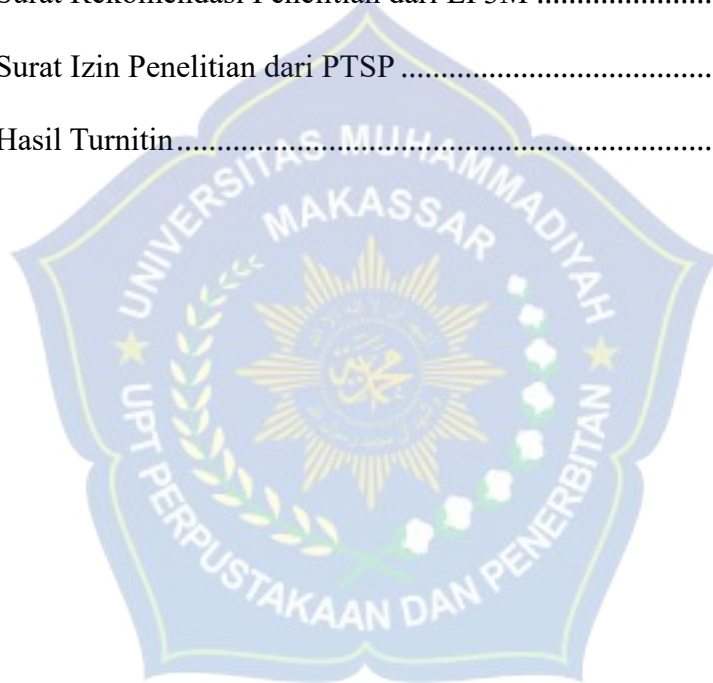
Gambar 1. Lada Belum Matang	8
Gambar 2. Lada Setengah Matang	8
Gambar 3. Lada Matang Sempurna	9
Gambar 4. Lada Lewat Matang	9
Gambar 5. Confusion Matrix	19
Gambar 6. Diagram Kerangka Berpikir	23
Gambar 7. Flowchart Perancangan Sistem	25
Gambar 8. Buah Lada Berdasarkan Tingkat Kematangan	33
Gambar 9. Pelabelan Data	33
Gambar 10. Hasil Resize dan Segmentasi ROI	35
Gambar 11. Hasil Augmentasi	35
Gambar 12. Kurva Random Forest	46
Gambar 13. Confusion Matrix Random Forest	46
Gambar 14. Kurva XGBoost	51
Gambar 15. Confusion Matriks XGBoost	52
Gambar 16. Confusion Matriks Ensemble Soft Voting	57
Gambar 17. Performa Masing-Masing Model	62

DAFTAR TABEL

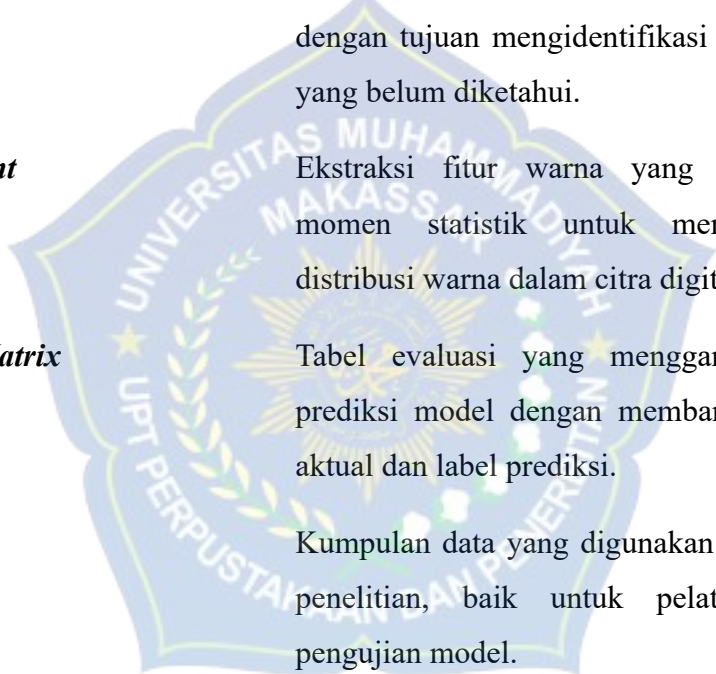
Tabel 1. Penelitian Terkait.....	21
Tabel 2. Jumlah Data.....	36
Tabel 3. Contoh Nilai Piksel RGB	37
Tabel 4. Hasil Ekstraksi Fitur HSV	38
Tabel 5. Library untuk Ekstraksi Fitur HSV	39
Tabel 6. Library Model Random Forest	40
Tabel 7. Output Nilai Probabilitas Random Forest	41
Tabel 8. Library Model XGBoost	42
Tabel 9. Output Nilai Probabilitas XGBoost.....	43
Tabel 10. Library Soft Voting Ensemble	44
Tabel 11. Output Nilai Probabilitas Ensemble Soft Voting.....	45
Tabel 12. Klasifikasi Report Random Forest	49
Tabel 13. Klasifikasi Report XGBoost.....	55
Tabel 14. Klasifikasi Report Ensemble Soft Voting.....	60
Tabel 15. Akurasi Per Kelas	62
Tabel 16. Perbandingan Masing-Masing Model	64

DAFTAR LAMPIRAN

Lampiran 1. Dataset Asli.....	74
Lampiran 2. Dataset Hasil Preprocessing	77
Lampiran 3. Dataset Hasil Augmentasi.....	79
Lampiran 4. Source Code.....	81
Lampiran 5. Permohonan Penelitian Kepada Ketua Program Studi Informatika	100
Lampiran 6. Surat Rekomendasi Penelitian dari LP3M	101
Lampiran 7. Surat Izin Penelitian dari PTSP	102
Lampiran 8. Hasil Turnitin.....	103



DAFTAR ISTILAH



<i>Accuracy</i>	Metrik evaluasi yang digunakan untuk mengukur tingkat ketepatan model klasifikasi, dihitung berdasarkan rasio antara jumlah prediksi benar terhadap total prediksi yang dilakukan.
<i>Classification</i>	Proses pengelompokan objek atau data ke dalam kelas tertentu berdasarkan kesamaan fitur, dengan tujuan mengidentifikasi label dari data yang belum diketahui.
<i>Color Moment</i>	Ekstraksi fitur warna yang memanfaatkan momen statistik untuk merepresentasikan distribusi warna dalam citra digital.
<i>Confusion Matrix</i>	Tabel evaluasi yang menggambarkan hasil prediksi model dengan membandingkan label aktual dan label prediksi.
<i>Dataset</i>	Kumpulan data yang digunakan sebagai bahan penelitian, baik untuk pelatihan maupun pengujian model.
<i>Ensemble Learning</i>	Metode dalam machine learning yang menggabungkan beberapa model untuk meningkatkan akurasi dan stabilitas hasil prediksi.
<i>Extreme Gradient Boosting (Xgboost)</i>	Algoritma berbasis pohon keputusan yang menggunakan teknik boosting untuk meningkatkan akurasi, dengan cara membangun

pohon secara berurutan untuk memperbaiki kesalahan prediksi sebelumnya.

F1-Score

Metrik evaluasi yang merepresentasikan keseimbangan antara precision dan recall melalui perhitungan rata-rata harmonik keduanya.

False Negative

Kondisi ketika data yang sebenarnya termasuk kelas positif salah diprediksi sebagai kelas negatif oleh model.

False Positive

Kondisi ketika data yang sebenarnya termasuk kelas negatif salah diprediksi sebagai kelas positif oleh model.

Hue

Komponen pada ruang warna HSV yang merepresentasikan rona atau jenis warna utama, seperti merah, hijau, dan biru.

Learning Curve

Grafik yang menggambarkan hubungan antara ukuran data pelatihan dengan kinerja model, digunakan untuk menganalisis fenomena overfitting maupun underfitting.

Machine Learning

Cabang kecerdasan buatan yang mempelajari algoritma untuk memungkinkan sistem komputer belajar dari data dan melakukan prediksi tanpa pemrograman eksplisit.

Matplotlib

Pustaka Python yang digunakan untuk menghasilkan visualisasi data berupa grafik, diagram, maupun kurva analisis performa model.

<i>Mean</i>	Ukuran tendensi sentral yang menunjukkan nilai rata-rata dari distribusi data piksel warna dalam citra.
<i>Numpy</i>	Pustaka Python yang menyediakan fungsi komputasi numerik tingkat tinggi, termasuk operasi matriks, array, dan perhitungan statistik.
<i>Overfitting</i>	Onidisi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga performanya menurun pada data uji.
<i>Pandas</i>	Pustaka Python untuk pengolahan data berbentuk tabel (dataframe), digunakan dalam membaca, memanipulasi, dan menganalisis data.
<i>Precision</i>	Metrik evaluasi yang mengukur ketepatan model dalam mengklasifikasikan data positif, dihitung berdasarkan rasio prediksi positif benar terhadap seluruh prediksi positif.
<i>Preprocessing</i>	Tahap awal pengolahan data, seperti resize, normalisasi, dan augmentasi, untuk mempersiapkan data sebelum masuk ke model.
<i>Random Forest</i>	Algoritma ensemble berbasis decision tree yang bekerja dengan membangun banyak pohon keputusan dan menggabungkan hasil prediksinya melalui voting.
<i>Recall</i>	Metrik evaluasi yang mengukur kemampuan model dalam mengenali semua data yang termasuk dalam kelas positif, dihitung dari rasio

prediksi positif benar terhadap seluruh data positif aktual.

Resize

Proses mengubah dimensi citra agar seragam, sehingga memudahkan proses ekstraksi fitur dan pelatihan model.

ROI (Region Of Interest)

Bagian spesifik dari citra yang dipilih untuk dianalisis, dengan tujuan memfokuskan proses pengolahan hanya pada objek utama.

Saturation

Komponen dalam model HSV yang menggambarkan tingkat kejenuhan atau intensitas warna.

Scikit-Learn

Pustaka Python yang menyediakan berbagai algoritma machine learning serta fungsi untuk evaluasi model.

Seaborn

Pustaka Python yang digunakan untuk visualisasi data dengan tampilan grafik yang lebih interaktif dan estetik dibandingkan Matplotlib.

Skewness

Ukuran statistik yang menunjukkan tingkat kemencengan distribusi warna pada citra.

Soft Voting

Metode ensemble yang menggabungkan probabilitas prediksi dari beberapa model dasar, kemudian memilih kelas dengan rata-rata probabilitas tertinggi.

Standard Deviation

Ukuran statistik yang menunjukkan seberapa jauh penyebaran atau variasi data terhadap rata-rata.

<i>Testing Data</i>	Bagian dari dataset yang digunakan untuk menguji performa model yang telah dilatih.
<i>Training Data</i>	Bagian dari dataset yang digunakan untuk melatih model agar mampu mengenali pola data.
<i>True Negative</i>	Kondisi ketika data yang sebenarnya negatif berhasil diprediksi sebagai negatif oleh model.
<i>True Positive</i>	Kondisi ketika data yang sebenarnya positif berhasil diprediksi sebagai positif oleh model.
<i>Tuning Hyperparameter</i>	Proses pencarian kombinasi parameter terbaik pada model machine learning untuk meningkatkan performa prediksi.
<i>Value</i>	Komponen dalam model HSV yang merepresentasikan tingkat kecerahan warna pada citra.
<i>Voting Classifier</i>	Metode ensemble learning yang menggabungkan hasil prediksi beberapa algoritma dasar untuk menghasilkan keputusan akhir klasifikasi.

BAB I

PENDAHULUAN

A. Latar Belakang

Lada (*Piper nigrum L.*) merupakan komoditas hasil usaha tani yang tersebar di berbagai wilayah tropis termasuk di Sulawesi Selatan yang memiliki lahan dan agroklimat yang mendukung. Tanaman ini memiliki nilai ekonomi yang besar karena pemanfaatannya yang luas dalam sektor makanan, farmasi, dan kosmetik. Sebagai negara agraris dengan iklim tropis yang mendukung, Indonesia memiliki potensi besar dalam pengembangan dan ekspor komoditas ini. Namun, kualitas dan nilai jual lada sangat bergantung pada tingkat kematangan buah saat panen karena memengaruhi mutu produk akhir seperti lada hitam maupun lada putih (Evizal, 2023).

Penentuan kematangan buah saat ini umumnya masih dilakukan secara manual dengan mengandalkan pengamatan visual terhadap warna kulit buah. Warna hijau menunjukkan belum matang, sedangkan warna merah menandakan buah telah matang. Meskipun metode ini telah lama digunakan, proses klasifikasi visual sangat dipengaruhi oleh subjektivitas individu dan pencahayaan. Akibatnya, proses sortasi sering kali tidak konsisten dan berisiko menurunkan mutu hasil panen secara keseluruhan (Wu et al., 2024).

Penerapan teknologi pengolahan citra digital dan machine learning menjadi salah satu pendekatan yang berkembang dalam mengatasi permasalahan klasifikasi berbasis visual. Warna kulit buah dapat dijadikan parameter utama dalam proses klasifikasi karena mengalami perubahan yang jelas selama proses pematangan. Melalui kamera digital, citra buah dapat ditangkap, kemudian diekstraksi fitur warnanya seperti nilai RGB atau HSV untuk dianalisis lebih lanjut (Benlachmi et al., 2022). Ruang warna HSV lebih selaras dengan persepsi warna manusia, sehingga menghasilkan analisis yang lebih mudah dipahami secara intuitif dan lebih tahan terhadap variasi pencahayaan (Fahmi et al., 2025). Penerapan color moments, yaitu perhitungan

mean, standard deviation, dan skewness pada fitur warna, juga mampu membedakan objek berdasarkan karakteristik warna dan meningkatkan akurasi klasifikasi (Kurniasari et al., 2022).

Penggunaan algoritma *machine learning*, seperti *Random Forest* dan *XGBoost* terbukti efektif dalam klasifikasi tingkat kematangan buah (Cahyani, 2024). Penggabungan beberapa algoritma melalui metode *ensemble learning* mampu meningkatkan performa klasifikasi secara signifikan karena kemampuan masing-masing model yang saling melengkapi dalam membedakan batas antar kelas sehingga meningkatkan akurasi keseluruhan (Sunarko et al., 2023). Pendekatan *soft voting* mengkombinasikan hasil prediksi beberapa model dengan menghitung rata-rata probabilitas, sehingga keputusan akhir ditentukan oleh kelas dengan nilai probabilitas tertinggi (Jones et al., 2024).

Penggunaan satu model klasifikasi saja sering kali belum cukup untuk memperoleh akurasi tinggi, terutama pada data kompleks. Kondisi ini mendorong munculnya pendekatan *ensemble learning*, yaitu teknik yang menggabungkan beberapa model pembelajaran untuk menghasilkan prediksi yang lebih akurat dan stabil dibandingkan penggunaan satu model tunggal (Rayadin et al., 2024).

Sebagian besar penelitian sebelumnya berfokus pada komoditas buah konsumsi seperti kelapa, mangga, dan tomat. Penelitian yang secara khusus mengkaji klasifikasi tingkat kematangan buah lada yang dikenal sebagai tanaman rempah dengan pendekatan pengolahan citra digital dan *ensemble learning* masih jarang dilakukan.

Berdasarkan permasalahan yang telah dijelaskan, penelitian ini bertujuan untuk membangun klasifikasi tingkat kematangan lada dengan menggunakan metode *ensemble learning* berdasarkan fitur *color moments* dari model warna HSV pada citra kulit buah. Metode *ensemble* yang digunakan dalam penelitian ini menggabungkan algoritma *machine learning*, yaitu *Random Forest* dan *XGBoost* melalui pendekatan *soft voting* yang menghitung rata-rata probabilitas dari masing-masing model sebagai teknik pengambilan

keputusan akhir. Melalui pendekatan model yang dirancang, dapat meningkatkan akurasi dan efisiensi proses klasifikasi, serta dapat diimplementasikan dalam kegiatan sortasi pascapanen guna mendukung peningkatan mutu hasil produksi lada.

B. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang yang telah dipaparkan, permasalahan yang diangkat pada penelitian ini adalah sebagai berikut:

1. Apakah metode *Random Forest* dan *XGBoost* dapat digunakan untuk mengklasifikasikan tingkat kematangan buah lada berdasarkan citra warna kulit?
2. Bagaimana tingkat akurasi yang dicapai metode *Ensemble Learning* dalam klasifikasi tingkat kematangan buah lada berbasis citra?

C. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan :

1. Untuk mengetahui apakah metode *Random Forest* dan *XGBoost* dapat digunakan dalam mengklasifikasikan tingkat kematangan buah lada berdasarkan citra warna kulit.
2. Untuk mengetahui tingkat akurasi dari metode *Ensemble Learning* dalam mengklasifikasikan tingkat kematangan buah lada berbasis citra

D. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Penulis

Penelitian ini memberikan kesempatan bagi penulis untuk memperdalam pemahaman dan keterampilan dalam bidang pengolahan citra digital dan penerapan algoritma *machine learning*, khususnya metode *ensemble learning* dalam klasifikasi citra buah lada berdasarkan tingkat kematangan.

2. Bagi Lembaga Pendidikan

Penelitian ini menambah khazanah ilmu pengetahuan di bidang pengolahan citra digital dan *machine learning*, khususnya pada klasifikasi buah rempah seperti lada. Selain itu, penelitian ini dapat menjadi referensi atau studi awal bagi mahasiswa dalam melakukan penelitian serupa, serta memperkuat peran institusi dalam mendukung pengembangan ilmu pengetahuan terapan.

3. Bagi Masyarakat

Penelitian ini memberikan informasi objektif mengenai tingkat kematangan buah lada berdasarkan warna kulitnya, sehingga dapat membantu menentukan waktu panen yang lebih tepat. Hal ini diharapkan berdampak pada peningkatan kualitas dan nilai jual hasil pertanian.

E. Ruang Lingkup Penelitian

Ruang lingkup penelitian adalah untuk mendefinisikan lingkup permasalahan secara lebih spesifik, dengan batasan masalah sebagai berikut :

1. Penelitian dilakukan terhadap buah lada sebagai objek utama, dengan fokus pada tingkat kematangan berdasarkan warna kulit buah.
2. Data yang digunakan berupa gambar (citra digital) buah lada yang diambil secara langsung menggunakan kamera, dengan pencahayaan dan latar belakang yang tidak seragam.
3. Citra dianalisis berdasarkan fitur warna menggunakan pendekatan *color moment* dalam model warna HSV.
4. Klasifikasi dibatasi pada empat kategori kematangan: belum matang, setengah matang, matang, dan lewat matang yang ditentukan berdasarkan perbedaan warna kulit buah.
5. Metode yang digunakan adalah *ensemble learning*, dengan model *Random Forest* dan *XGBoost*.
6. Penelitian ini berfokus pada proses klasifikasi tingkat kematangan lada menggunakan model *machine learning*. penelitian ini terbatas pada penggunaan dataset hasil dokumentasi langsung yang telah disiapkan peneliti. Penelitian tidak menggunakan dataset tambahan dari luar dan

tidak diarahkan untuk pengujian secara real time maupun pengembangan antarmuka pendukung.

F. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini menguraikan latar belakang yang menjadi dasar pelaksanaan penelitian, serta memuat rumusan masalah, tujuan, manfaat, ruang lingkup, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menyajikan teori-teori yang relevan dan algoritma yang digunakan serta studi literatur sebelumnya sebagai acuan. Tujuan bab ini adalah memberikan dasar teoretis untuk mendukung pelaksanaan penelitian.

BAB III METODE PENELITIAN

Bab ini menjelaskan secara rinci metode yang digunakan dalam penelitian, meliputi proses pengumpulan data, ekstraksi fitur, pelatihan model, serta analisis data. Bab ini menjadi pedoman teknis dalam pelaksanaan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari proses penelitian yang telah dilakukan berdasarkan data yang telah dikumpulkan dan diolah. Hasil ditampilkan untuk menunjukkan kinerja metode yang digunakan dalam mencapai tujuan penelitian.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan sebagai jawaban terhadap rumusan masalah. Selain itu, disampaikan pula saran yang dapat menjadi masukan untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Lada

Lada merupakan tanaman rempah dari famili *Piperaceae* dan banyak digunakan sebagai bumbu masakan. Selain perannya dalam kuliner, lada juga digunakan dalam berbagai formulasi herbal sebagai bagian dari pengobatan tradisional untuk mengatasi berbagai jenis penyakit (Alves et al., 2023).

Lada merupakan salah satu tanaman rempah unggulan dengan potensi nilai jual tinggi dan tumbuh subur di kawasan tropis. Lada digunakan secara luas dalam industri makanan, minuman, kosmetik, dan farmasi karena kandungan senyawa aktif seperti piperin (Evizal, 2023).

Di Indonesia, budidaya lada tersebar di berbagai daerah, dengan sentra produksi utama berada di Bangka Belitung, Lampung, Kalimantan Barat, dan Sulawesi Selatan. Berdasarkan data dari Kementerian Pertanian, Bangka Belitung menjadi penyumbang terbesar produksi lada nasional, diikuti oleh daerah lain yang juga memiliki potensi besar untuk kembangkan tanaman ini (Pusat Data dan Informasi Pertanian, 2022).

2. Tingkat Kematangan Buah Lada

Tingkat kematangan buah lada menjadi penentu kualitas hasil panen, baik untuk produksi lada hitam maupun lada putih. Kematangan buah berkaitan langsung dengan karakteristik fisik dan kimia buah, seperti warna kulit, kadar air, kandungan senyawa aktif, aroma, serta daya simpan. Buah lada yang dipanen pada tingkat kematangan yang sesuai akan menciptakan produk dengan kualitas yang prima dan nilai jual yang lebih baik (Evizal, 2023).

Menurut Hafizah dan Rizal (2024) menentukan tingkat kematangan dilakukan dengan mengamati perubahan warna pada kulit buahnya. Warna menjadi indikator utama dalam membedakan tahap kematangan lada. Meskipun manusia mampu mengenali tingkat kematangan melalui warna, akurasi pengamatan visual oleh mata manusia seringkali kurang konsisten.

Perubahan warna pada kulit buah lada menjadi indikator visual utama dalam menilai tingkat kematangan. Buah yang masih dalam kondisi belum matang biasanya berwarna hijau pekat atau hijau tua pada seluruh permukaannya. Buah pada tahap ini memiliki kadar air tinggi, kandungan piperin yang belum optimal, dan belum siap untuk proses pengolahan. Seiring proses pematangan, warna kulit buah mulai menunjukkan semburat kuning, yang menandakan bahwa buah berada pada fase setengah matang. Pada tahap ini, kandungan biokimia di dalam buah mulai berkembang, namun belum mencapai kualitas maksimum (Anjani et al., 2023).

Buah yang mencapai tingkat kematangan sempurna ditandai dengan dominasi warna merah di permukaan kulit, meskipun bagian lainnya dapat memiliki warna hijau atau kekuningan. Buah lada pada tahap ini sangat ideal untuk dipanen karena telah mencapai kandungan senyawa aktif yang optimal serta memiliki karakteristik warna dan aroma yang baik (Reinaldi, 2021).

Menurut Pusat Data dan Informasi Pertanian (2022) tingkat kematangan buah lada dapat dikenali berdasarkan ciri-ciri warna kulit sebagai berikut :

a. Lada Belum Matang

Memiliki ciri utama berupa warna kulit yang hijau cerah, menandakan bahwa buah belum memasuki tahap pematangan. Permukaannya tampak mengkilap dan terasa keras saat disentuh, mencerminkan struktur sel yang masih padat dan kadar air yang tinggi (Din et al., 2022). Sebagaimana ditunjukkan pada gambar 1.



Gambar 1. Lada Belum Matang (Aeni, 2022)

b. Lada Setengah Matang

Ditandai dengan perubahan warna kulit dari hijau menuju kekuningan atau orange pucat, meskipun masih terdapat bagian yang berwarna hijau dalam jumlah cukup besar. Perubahan warna ini menunjukkan bahwa proses pematangan telah dimulai, namun belum berlangsung secara merata di seluruh permukaan buah (Usman et al., 2024). Hal ini dapat dilihat pada gambar 2.



Gambar 2. Lada Setengah Matang (Pak Tani, 2021)

c. Lada Matang Sempurna

Memiliki ciri khas berupa warna kulit yang didominasi oleh merah cerah, meskipun masih mungkin terdapat sedikit bagian yang berwarna hijau atau kekuningan. Warna merah ini menandakan bahwa proses pematangan telah berlangsung hampir sepenuhnya dan kadar piperin serta senyawa aromatik berada pada tingkat optimal. Permukaan kulit mulai kehilangan kilapnya dan teksturnya menjadi lebih lunak dibandingkan tahap sebelumnya, namun masih

cukup kokoh untuk dipanen (Rahman dan Koodoh, 2022). Representasi visualnya dapat dilihat pada gambar 3.



Gambar 3. Lada Matang Sempurna (Candra, 2023)

d. Lada Lewat Matang

Ditandai dengan perubahan warna kulit menjadi merah tua atau kecoklatan, menunjukkan bahwa buah telah melewati fase kematangan optimal. Tekstur kulit dan daging buah menjadi lebih lunak, bahkan cenderung lembek, sehingga lebih rentan terhadap kerusakan mekanis maupun serangan mikroorganisme. Permukaan kulit kehilangan kilapnya dan sering kali menunjukkan tanda-tanda keriput (Atika, 2023). Berikut ilustrasi pada gambar 4.



Gambar 4. Lada Lewat Matang (Agri Edu, 2023)

Pada gambar di atas tingkat kematangan buah lada yang memperlihatkan variasi warna kulit sebagai indikator utama dalam menentukan kematangan buah. Analisis terhadap warna kulit ini sangat penting karena kematangan buah lada berpengaruh langsung terhadap kualitas serta

karakteristik fisik dan kimia lada yang dihasilkan. Menurut Anjani et al. (2023) buah yang dipanen pada waktu yang tepat akan mengandung senyawa kimia optimal seperti piperin dan minyak atsiri, yang menentukan cita rasa dan aroma khas lada. Panen yang dilakukan terlalu awal atau terlalu lewat dapat menyebabkan penurunan kualitas, baik dari segi rasa, aroma, maupun daya simpan, yang pada akhirnya berdampak negatif terhadap nilai jual dan kepuasan konsumen.

3. Pengolahan Citra Digital

Citra atau gambar adalah tampilan visual yang direkam pada permukaan dua dimensi dalam bentuk digital, perlu diubah menjadi data numerik melalui proses digitalisasi. Dengan proses ini, komputer mampu secara otomatis merekam, memproses, dan menganalisis data visual secara efektif (Tri Laksono et al., 2022).

Pengolahan citra digital adalah teknik untuk mengolah dan menganalisis gambar guna mengenali atau mengukur ciri-ciri objek di dalamnya. Proses ini dimulai dari pengambilan citra, dilanjutkan dengan pra-pemrosesan seperti resize dan peningkatan kualitas, agar gambar siap dianalisis lebih lanjut. Setelah data citra terkumpul, dilakukan proses pengolahan dengan penyesuaian pada data digital tersebut, salah satunya melalui ekstraksi ciri. Pada tahap ini, citra akan dipotong (*cropping*) untuk meminimalkan latar belakang yang tidak relevan sehingga objek menjadi lebih jelas dan fokus dalam citra yang dianalisis (Bangun et al., 2021).

Menurut Angriana (2023) terdapat beberapa tahapan dalam pengolahan citra digital, di antaranya:

- a. Akuisisi citra, yaitu tahap awal pengambilan citra menggunakan perangkat seperti kamera atau pemindai (*scanner*) untuk memperoleh gambar yang akan diproses.
- b. Processing citra, bertujuan untuk mempersiapkan citra agar siap menjalani proses lanjutan, baik untuk ekstraksi fitur maupun klasifikasi. Langkah ini biasanya mencakup teknik seperti pemotongan (*cropping*) dan pengubahan ukuran gambar.

- c. Segmentasi, yaitu proses membagi citra menjadi bagian-bagian yang lebih kecil untuk memisahkan objek utama dari latar atau objek lain.
- d. Augmentasi data, yaitu proses menghasilkan variasi baru dari citra asli dengan teknik transformasi seperti rotasi, flipping, perubahan pencahayaan, atau pergeseran posisi. Augmentasi bertujuan untuk memperbesar jumlah data latih dan meningkatkan kemampuan generalisasi model.
- e. Representasi dan deskripsi, di mana representasi mengubah data piksel menjadi format yang dapat diproses komputer, sementara deskripsi mengekstrak fitur atau atribut dari citra yang berguna untuk membedakan antar kelas.
- f. Pengenalan, yaitu proses pemberian label pada objek berdasarkan fitur-fitur yang telah diperoleh, sehingga objek dapat diidentifikasi sesuai kategorinya.

4. Ekstraksi Fitur HSV

Ekstraksi fitur HSV adalah model warna yang dirancang untuk mempermudah pemrosesan citra digital, terutama dalam aplikasi pengenalan warna. Model ini membagi warna menjadi tiga komponen utama yaitu *Hue* (H), yang menggambarkan jenis warna, seperti merah, hijau, atau biru; *Saturation* (S), yang menunjukkan kejenuhan warna atau seberapa murni warna tersebut; dan *Value* (V), yang merepresentasikan kecerahan atau intensitas warna (Ali Buchari et al., 2022).

Menurut Maulana (2021), model warna HSV (*Hue, Saturation, Value*) terdiri dari tiga bagian utama, yaitu:

- a. *Hue*, yaitu jenis warna utama dalam model warna yang menunjukkan rona atau nuansa warna tertentu.
- b. *Saturation*, yaitu tingkat kejenuhan warna yang menunjukkan seberapa murni atau pekat suatu warna.
- c. *Value*, yaitu tingkat kecerahan warna. Nilai 0% merepresentasikan warna hitam, sedangkan nilai yang semakin mendekati 100% menunjukkan warna yang semakin terang.

Berdasarkan penjelasan mengenai HSV di atas, menurut Nurulrachman et al. (2023) nilai HSV dapat dihitung dari RGB dengan cara sederhana yang dirumuskan sebagai berikut:

$$V = \max(R, G, B) \quad (1)$$

$$S = \begin{cases} 0 & \text{jika } v = 0 \\ \frac{v - \min(R, G, B)}{v} & \text{jika } v > 0 \end{cases} \quad (2)$$

$$H = \begin{cases} 0 & \text{jika } \Delta = 0 \\ 60^\circ \times \frac{G-B}{\Delta} & \text{jika } V = R \\ 60^\circ \times \left(2 + \frac{B-R}{\Delta}\right) & \text{jika } V = G \\ 60^\circ \times \left(4 + \frac{R-G}{\Delta}\right) & \text{jika } V = B \end{cases} \quad (3)$$

Dimana :

R : intensitas warna Merah (Red) pada piksel (skala 0–255)

G : intensitas warna Hijau (Green) pada piksel (skala 0–255)

B : Intensitas warna Biru (Blue) pada piksel (skala 0–255)

V : Tingkat kecerahan warna, diambil dari nilai maksimum (value)

S : Tingkat kejenuhan warna.

H : Nilai rona warna dalam derajat.

Δ : Selisih antara nilai maksimum dan minimum dari R , G , dan B .

$\max(R, G, B)$: Nilai tertinggi dari tiga komponen warna.

$\min(R, G, B)$: Nilai terendah dari tiga komponen warna.

5. Color Moment

Color Moment merupakan metode ekstraksi ciri warna yang digunakan untuk mengukur kemiripan warna suatu citra. Konsep ini menganggap distribusi warna pada citra sebagai distribusi probabilitas, sehingga ciri-ciri statistik seperti *Mean* (rata-rata), Standar Deviasi, dan *Skewness* (kemencengan) dapat digunakan sebagai representasi citra (Nurulrachman et al., 2023). Ada 3 parameter yang dihasilkan pada Momen Warna yaitu *mean*, *standard deviation*, *skewness* (Maulana Alfaruq et al., 2023).

a) *Mean*

Mean adalah nilai rata-rata dari semua warna dalam gambar. Untuk itu, digunakan rumus berikut:

$$\mu = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N x_{ij} \quad (4)$$

Dimana :

μ : nilai rata-rata (mean)

x_{ij} : elemen index baris i , index kolom j

M: nilai baris matriks.

N : nilai kolom matrix

b) *Standar Deviasi*

Standar deviasi adalah ukuran seberapa beragam warna di dalam gambar. Rumus deviasi dapat dituliskan sebagai berikut:

$$\sigma = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \mu)^2} \quad (5)$$

Dimana:

σ : nilai standar deviasi.

μ : nilai rata-rata (mean).

x_{ij} : elemen matriks indeks baris I dan indeks j.

M : nilai baris matriks

N : nilai kolom matriks

c) *Skewness*

Skewness adalah ukuran apakah warna di gambar lebih cenderung ke warna gelap atau terang. *Skewness* secara matematis dirumuskan sebagai berikut:

$$Skewness = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \left(\frac{x_{ij} - \mu}{\sigma} \right)^3 \quad (6)$$

Dimana :

σ : nilai standar deviasi

μ : nilai rata-rata (mean)

x_{ij} : elemen matriks indeks baris I dan indeks j.

M : nilai baris matriks

N : nilai kolom matriks

6. Random Forest

Metode *Random Forest* muncul pada tahun 2001 diperkenalkan oleh Breiman (Amini et al., 2022). Menurut Istiani (2024), algoritma ini memiliki sejumlah keunggulan, di antaranya mampu memberikan performa yang baik dalam tugas klasifikasi, menghasilkan tingkat kesalahan (*error rate*) yang relatif rendah, efisien dalam menangani data berukuran besar, serta efektif dalam mengatasi data yang memiliki nilai hilang (*missing values*). *Random Forest* termasuk dalam pendekatan *ensemble* yang populer dan banyak digunakan untuk tugas klasifikasi maupun regresi.

Menurut Argadianata et al. (2025), algoritma *Random Forest* membangun banyak pohon keputusan keputusan secara paralel, di mana tiap pohon dilatih menggunakan sample data yang berbeda, kemudian menggabungkan hasil klasifikasi dari seluruh pohon untuk menghasilkan keputusan yang lebih andal dan presisi. Pendekatan tersebut menjadikan *Random Forest* sangat handal dalam menangani data kompleks tanpa memerlukan proses pra-pemrosesan yang rumit.

Dalam konteks klasifikasi kualitas buah, Argadianata et al. (2025) menambahkan bahwa *Random Forest* mampu mengklasifikasikan apel berdasarkan ciri warna, ukuran, dan tekstur ke dalam kategori kualitas, sehingga mendukung penilaian mutu buah secara otomatis dan objektif. *Random Forest* cocok untuk pengolahan citra karena mampu menangani data berdimensi tinggi, fitur kompleks, serta distribusi data yang variatif dan non-linear.

Menurut Istiani (2024), terdapat tiga aspek utama dalam *metode Random Forest*, yaitu:

- a. Membangun sejumlah pohon keputusan melalui teknik *bootstrap sampling*, yaitu pengambilan sampel acak dengan pengembalian dari dataset pelatihan.
- b. Setiap pohon keputusan melakukan prediksi secara independen dan acak berdasarkan subset fitur tertentu.

- c. Random Forest menghasilkan prediksi akhir dengan metode agregasi (*voting*) menggunakan mayoritas suara dalam kasus klasifikasi, dan menggunakan nilai rata-rata dalam kasus regresi.

Menurut Amrozi et al. (2024) dalam algoritma *Random Forest*, hasil prediksi $\hat{y}$ dihitung sebagai rata-rata dari prediksi semua pohon keputusan yang ada dalam model. Misalkan $\hat{y}_i$ adalah prediksi dari pohon keputusan ke- i dan terdapat total prediksi T pohon keputusan dalam ensemble. Maka, hasil prediksi akhir $\hat{y}$ dapat dihitung menggunakan rumus berikut :

$$\hat{y} = \frac{1}{T} \sum_{i=1}^T \hat{y}_i \quad (7)$$

Dimana :

$\hat{y}_i$ = prediksi dari pohon keputusan ke - i

T = Jumlah total pohon keputusan dalam ensemble

$\hat{y}$ = Hasil prediksi akhir setelah agregasi (rata-rata) dari semua pohon keputusan

7. XGBoost

Algoritma Extreme Gradient Boosting (XGBoost) adalah model pohon keputusan yang memiliki aturan pembagian yang serupa dengan *decision tree*. *XGBoost* dapat digunakan untuk tugas regresi dan klasifikasi, dan termasuk dalam varian *Gradient Boosting Machine (GBM)* yang efisien serta terukur, yang telah banyak diterapkan dalam berbagai bidang seperti computer vision, data mining, dan lainnya. Pada algoritma *XGBoost*, setiap pohon bertugas untuk memperkirakan variabel target (Cahyani, 2024). Pohon pertama digunakan untuk memprediksi nilai target, pohon kedua memperkirakan selisih antara prediksi dari pohon pertama dan nilai target yang sebenarnya, dan seterusnya, hingga residu antara prediksi dan target diminimalkan.

Berdasarkan penjelasan dari Elzandy et al. (2025), XGBoost dirancang untuk mengatasi berbagai jenis permasalahan seperti klasifikasi, regresi, maupun pemeringkatan (*ranking*). Algoritma ini mulai mendapat perhatian luas setelah

sukses digunakan secara dominan oleh peserta dalam kompetisi *Higgs Boson*, yang menjadikannya sebagai salah satu metode paling populer dan efektif.

XGBoost digolongkan ke dalam algoritma *tree ensemble*, yang terdiri dari beberapa pohon keputusan klasifikasi dan regresi (*Classification and Regression Trees* atau CART). Keunggulan utama XGBoost terletak pada kemampuannya dalam melakukan proses pelatihan model dengan sangat cepat, yang bahkan dapat mencapai sepuluh kali lebih efisien dibandingkan dengan versi klasik dari GBM.

Menurut Cahyani et al. (2024), Nilai prediksi pada algoritma *XGBoost* ditentukan oleh pohon K seperti dalam persamaan berikut :

$$\hat{y}_i = \sum_{k=1}^k f_k(x_i), f_k \in Y \quad (8)$$

Dimana :

$\hat{y}_i$ = nilai prediksi untuk sampel data ke- i

$\sum_{k=1}^k$ = penjumlahan dari kontribusi semua pohon keputusan dimulai dari pohon pertama $k = 1$ hingga pohon k

$f_k(x_i)$ = Fungsi pohon ke- k yang memetakan fitur input x_i (data sampel ke- i) ke nilai prediksi.

$f_k \in Y$ = Setiap fungsi f_k berada dalam ruang fungsi Y yang berisi semua kemungkinan pohon keputusan.

8. Ensemble Learning

Ensemble learning adalah metode dalam *machine learning* yang menggabungkan beberapa model untuk meningkatkan akurasi dan konsistensi hasil klasifikasi. Strategi ini bertujuan untuk mengurangi kesalahan prediksi yang mungkin dapat terjadi jika hanya menggunakan satu model tunggal (Kavzoglu dan Teke, 2022).

Menurut Mienye dan Sun (2022) langkah utama dalam model *ensemble* adalah memilih metode yang digunakan, di mana jenis metode *ensemble learning* biasanya menentukan mekanisme penggabungan seperti *bagging*, *boosting*, *stacking* atau *voting*. Natras et al., (2022) menerapkan ensemble learning dengan

menggabungkan prediksi dari beberapa algoritma dasar, yaitu *Random Forest*, *AdaBoost*, dan *XGBoost*. Hasil prediksi dari model-model tersebut digabungkan menggunakan metode *voting*, yang bertujuan meningkatkan akurasi dan kinerja model.

Voting classifier merupakan salah satu metode dalam *ensemble learning* yang menggabungkan hasil prediksi dari beberapa model dasar untuk menentukan keputusan akhir. Dalam penerapannya, terdapat dua jenis *voting* yang umum digunakan, yaitu *hard voting* dan *soft voting* (Kibria et al., 2022).

Soft voting merupakan salah satu metode penggabungan prediksi dalam *ensemble learning*, di mana setiap model dasar menghasilkan nilai probabilitas untuk setiap kelas. Nilai-nilai probabilitas tersebut kemudian digabungkan dengan cara mengambil rata-ratanya, untuk menentukan hasil akhir. Kelas dengan rata-rata nilai probabilitas tertinggi dipilih sebagai hasil klasifikasi (Josés et al., 2024). Menurut Bey Lirna et al. (2024), perumusan matematis model *soft voting* ditunjukkan pada rumus berikut :

$$S = \frac{1}{N} \sum_{i=1}^N P_i \quad (9)$$

Dimana :

S = nilai gabungan atau prediksi akhir klasifikasi.

N = jumlah model yang digunakan dalam *ensemble*

P_i = nilai probabilitas prediksi kelas dari model ke -i

9. Klasifikasi

Klasifikasi adalah langkah untuk membangun sebuah sistem yang dapat membedakan berbagai kelas data, dengan tujuan mengidentifikasi kelas dari objek yang belum dilabeli. Pada proses ini, data dikumpulkan berdasarkan kesamaan yang sudah ditetapkan oleh analis sebelum analisis dimulai. Metode ini melakukan pemeriksaan terhadap data yang telah dikelompokkan dan dikategorikan sesuai dengan keanggotaan masing-masing (Melisa et al., 2022).

Klasifikasi dilakukan untuk memisahkan data ke dalam kategori tertentu dan menerapkan model tersebut untuk menentukan kelas dari data yang tidak

diketahui kelasnya. Tahapan klasifikasi mencakup pembangunan model menggunakan data training dan pengujian model pada data testing untuk memastikan kinerjanya (Raysyah et al., 2021).

Menurut penelitian Melisa et al. (2022) proses klasifikasi terdiri dari tiga tahap utama:

- a. *Learning* (Pembelajaran): Algoritma mencari record dari training set untuk membangun model klasifikasi yang menggambarkan hubungan antara fitur dan label kelas.
- b. Pengujian Model: Model yang telah dibuat diuji dengan data uji yang sudah diklasifikasikan, berbeda dari data training, untuk mengevaluasi akurasi.
- c. *Classification* (Klasifikasi): Model digunakan untuk mengklasifikasikan data yang belum dilabeli dalam basis data.

10. Metrik Evaluasi

Dalam pemodelan klasifikasi, proses utamanya adalah memprediksi label kelas yang telah ditetapkan sebelumnya. Dalam evaluasi model yang paling lazim digunakan adalah akurasi klasifikasi, di mana model dianggap memiliki performa baik jika mencapai nilai di atas 80%. Proses pengukuran akurasi dilakukan dengan menerapkan model klasifikasi pada data uji untuk menghasilkan prediksi, kemudian membandingkan hasil prediksi tersebut dengan label aktual yang sudah diketahui (Hernando et al., 2024).

Akurasi berfungsi sebagai parameter evaluasi performa model, khususnya dalam mengukur ketepatan hasil prediksi dibandingkan dengan seluruh prediksi yang dihasilkan. Metrik ini mencerminkan kemampuan model untuk mengidentifikasi kelas target secara akurat, baik itu kelas positif maupun negatif (Cahyani, 2024).

Salah satu metode yang digunakan untuk mengevaluasi akurasi model klasifikasi adalah dengan memanfaatkan *confusion matrix*. *Confusion matrix* menyajikan ringkasan hasil prediksi model dalam bentuk tabel yang disusun berdasarkan kategori kelas. Dalam tabel tersebut, baris merepresentasikan label

sebenarnya dari data, sementara kolom menunjukkan label yang diprediksi oleh model (Heydarian et al., 2022) .

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Gambar 5. Confusion Matrix (Rachmi, 2020)

Dalam evaluasi kinerja menggunakan *confusion matrix*, terdapat empat komponen utama untuk menilai hasil prediksi model (Prasetyo et al., 2021) :

- TP (*True Positive*): data yang benar-benar merupakan objek dan berhasil dikenali sebagai objek oleh model.
- TN (*True Negative*): data yang bukan objek dan berhasil dikenali sebagai bukan objek oleh model.
- FP (*False Positive*): data yang bukan objek tetapi keliru dikenali sebagai objek oleh model.
- FN (*False Negative*): data yang merupakan objek namun tidak dikenali oleh model sebagai objek

Menurut Helmud et al. (2024) Dalam klasifikasi, terdapat sejumlah metrik evaluasi yang umum digunakan. Nilai-nilai tersebut dapat diperoleh dari *confusion matrix*. Perhitungan tersebut menghasilkan rumus sebagai berikut:

1) Accuracy

Akurasi klasifikasi merupakan metrik evaluasi yang paling umum digunakan untuk menilai seberapa tepat suatu model klasifikasi prediktif dalam menghasilkan prediksi yang benar dari seluruh prediksi yang dilakukan.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

2) *Precision*

Precision adalah metrik yang mengukur seberapa banyak prediksi positif yang benar, sehingga digunakan untuk menilai ketepatan model dalam mengklasifikasikan kelas minoritas.

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

3) *Recall*

Recall adalah metrik yang digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data yang termasuk dalam kelas positif, termasuk memperhitungkan prediksi positif yang terlewat.

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

4) *F1-Score*

F1-Score adalah metrik evaluasi yang menggambarkan keseimbangan antara precision dan recall.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (13)$$

B. Penelitian Tekait

Kajian penelitian terkait dilakukan untuk memahami metode, pendekatan, dan hasil yang telah dicapai pada penelitian sebelumnya. Informasi ini penting sebagai dasar perbandingan agar penelitian yang dilakukan lebih terarah. Kajian juga membantu menemukan celah penelitian yang belum banyak dibahas. Dengan adanya kajian, penelitian ini dapat disusun lebih kuat dan memiliki kebaruan. Ringkasan penelitian terkait ditampilkan pada Tabel 1 berikut.

Tabel 1. Penelitian Terkait

No.	Judul dan Penulis	Tahun	Hasil
1	Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes (Maretva Cendani dan Wibowo, 2022).	2022	Penelitian ini membandingkan efektivitas metode ensemble untuk klasifikasi penyakit diabetes menggunakan data medis. Metode yang digunakan (<i>Random Forest</i> , <i>Extra Trees</i>), <i>Boosting</i> (<i>XGBoost</i> , <i>LightGBM</i> , <i>CatBoost</i>), dan <i>Stacking</i> . Hasil terbaik diperoleh dari <i>Boosting</i> .
2	Optimalisasi Model Ensemble Learning dengan Augmentasi dan SMOTE pada Sistem Pendeteksi Kualitas Buah (Hidayat et al., 2024).	2024	Objek penelitian adalah buah mangga dan srikaya. Menggunakan DCS-OLA dan <i>Stacking</i> dengan basis Logistic Regression, Decision Tree, Gaussian Naive Bayes, dan MLP. Augmentasi jittering dan SMOTE digunakan untuk meningkatkan performa.
3	Klasifikasi Kualitas Kelapa Menggunakan Random Forest, XGBoost, dan Decision Tree Dengan Ekstraksi Fitur Warna, Tekstur dan Bentuk (Cahyani, 2024).	2024	Objek penelitian berupa citra buah kelapa lima kelas dengan fitur warna (Color Moment), tekstur (GLCM), dan bentuk (HOG), kemudian dibandingkan menggunakan model Random Forest, XGBoost, dan Decision Tree.
4	Perbandingan Random Forest, CNN, dan SVM Untuk Klasifikasi Jenis Mangga (Mardianto et al., 2024).	2024	Penelitian ini menggunakan citra digital mangga dari berbagai varietas. Metode yang digunakan antara lain Random Forest, SVM, dan CNN. Random Forest menunjukkan performa terbaik.
5	Klasifikasi Kualitas Buah Apel Menggunakan Metode Random Forest (Argadianata et al., 2025).	2025	Objek berupa 4000 sampel buah apel dari Kaggle, dengan fitur fisik seperti ukuran, rasa, dan keasaman. Random Forest digunakan dan menghasilkan akurasi 88,5% serta F1-score 88,5%. Analisis confusion matrix menunjukkan kesalahan klasifikasi yang dapat diperbaiki.

6	Klasifikasi Varietas Anggur Berdasarkan Citra Daun Menggunakan VGG16-XGBoost (Elzandy et al., 2025).	2025	Objek berupa citra daun anggur (560 data latih, 60 uji). VGG16 digunakan untuk ekstraksi fitur, lalu XGBoost untuk klasifikasi. Model mencapai akurasi 89,58% dan F1-score 89,55% dengan waktu pelatihan sekitar 832 detik.
---	--	------	---

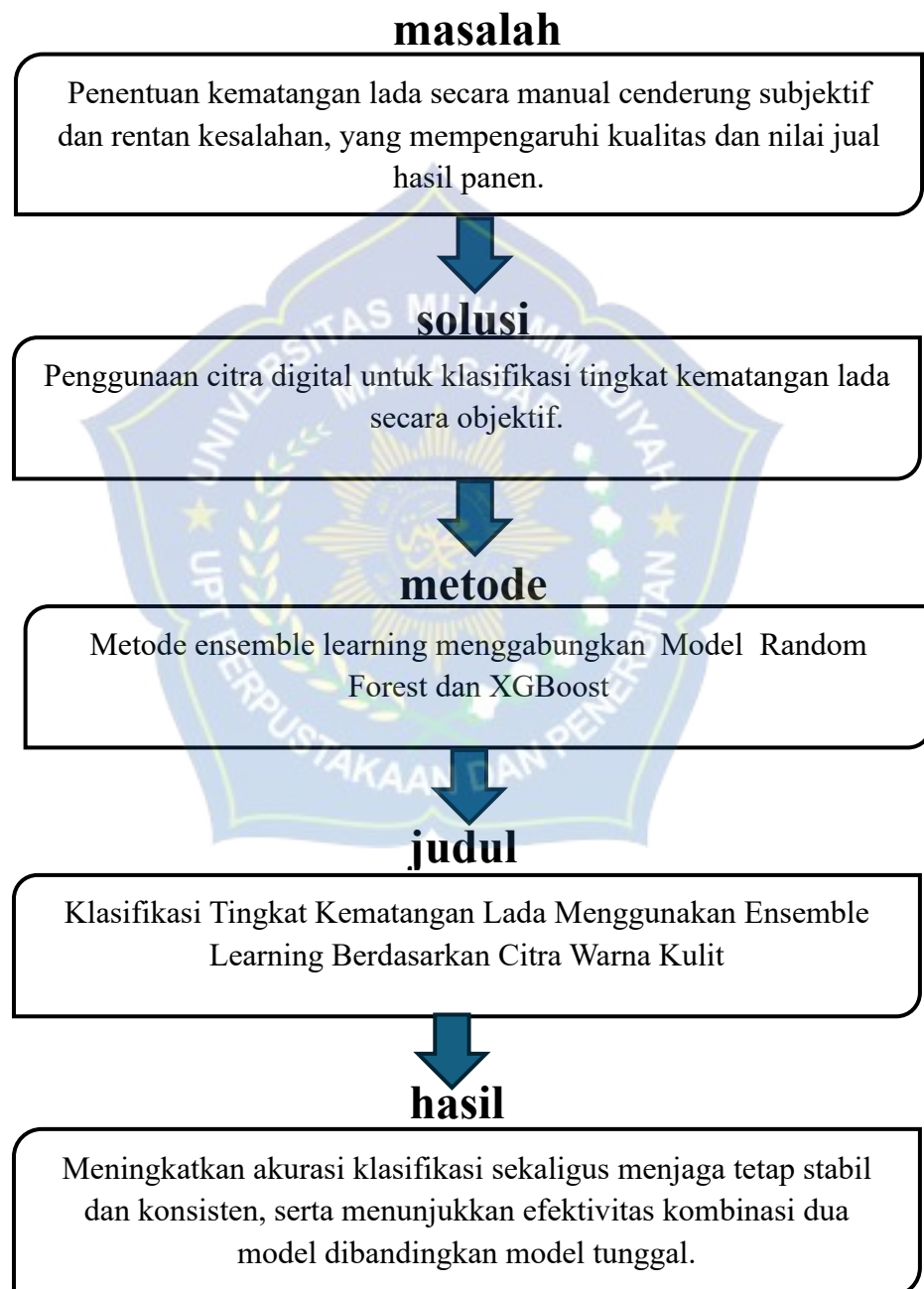
Penelitian Yang Penulis Lakukan

7	Klasifikasi Tingkat Kematangan Lada Menggunakan Ensemble Learning Berdasarkan Citra Warna Kulit (Jihan, 2025).	2025	Objek penelitian ini adalah buah lada dengan empat tingkat kematangan. Fitur yang digunakan berasal dari HSV Color Moment pada citra buah. Metode yang diterapkan adalah soft voting ensemble yang menggabungkan Random Forest dan XGBoost, dan hasilnya menunjukkan performa yang unggul, stabil, serta konsisten dalam klasifikasi tingkat kematangan lada.
---	--	------	---

Penelitian ini memiliki kebaruan dalam objek dan pendekatan yang digunakan, yakni klasifikasi tingkat kematangan buah lada berdasarkan citra warna kulit, yang belum banyak dikaji dalam penelitian sebelumnya. Berbeda dengan studi terdahulu yang lebih banyak menggunakan buah lain seperti apel, mangga, atau kelapa, penelitian ini berfokus pada buah lada dengan empat tingkat kematangan. Selain itu, pendekatannya hanya mengandalkan fitur warna HSV tanpa melibatkan fitur tambahan seperti tekstur, bentuk, atau teknik deep learning. Model yang digunakan berupa kombinasi ensemble dari *Random Forest* dan *XGBoost* yang digabungkan menggunakan teknik *soft voting*, namun tetap mampu menghasilkan akurasi tinggi dan stabilitas antar kelas. Pendekatan ini ringan secara komputasi, mengisi celah dalam studi sebelumnya yang cenderung rumit.

C. Kerangka Berpikir

Kerangka berpikir merupakan susunan logis yang digunakan untuk menggambarkan keterkaitan antara konsep atau variabel yang terlibat dalam suatu penelitian atau proyek. Kerangka berpikir yang digunakan dalam penelitian ini disajikan pada bagian berikut :



Gambar 6. Diagram Kerangka Berpikir

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Lokasi Penelitian ini di Desa Tokalimbo, Kecamatan Towuti, Kabupaten Luwu Timur, Sulawesi Selatan, yang dipilih karena dikenal sebagai salah satu daerah penghasil lada. Pengambilan data dilakukan secara langsung dengan cara mendokumentasikan gambar buah lada di lapangan. Pelaksanaan penelitian dimulai pada bulan Mei 2025 dan berlangsung hingga Juli 2025.

B. Alat dan Bahan

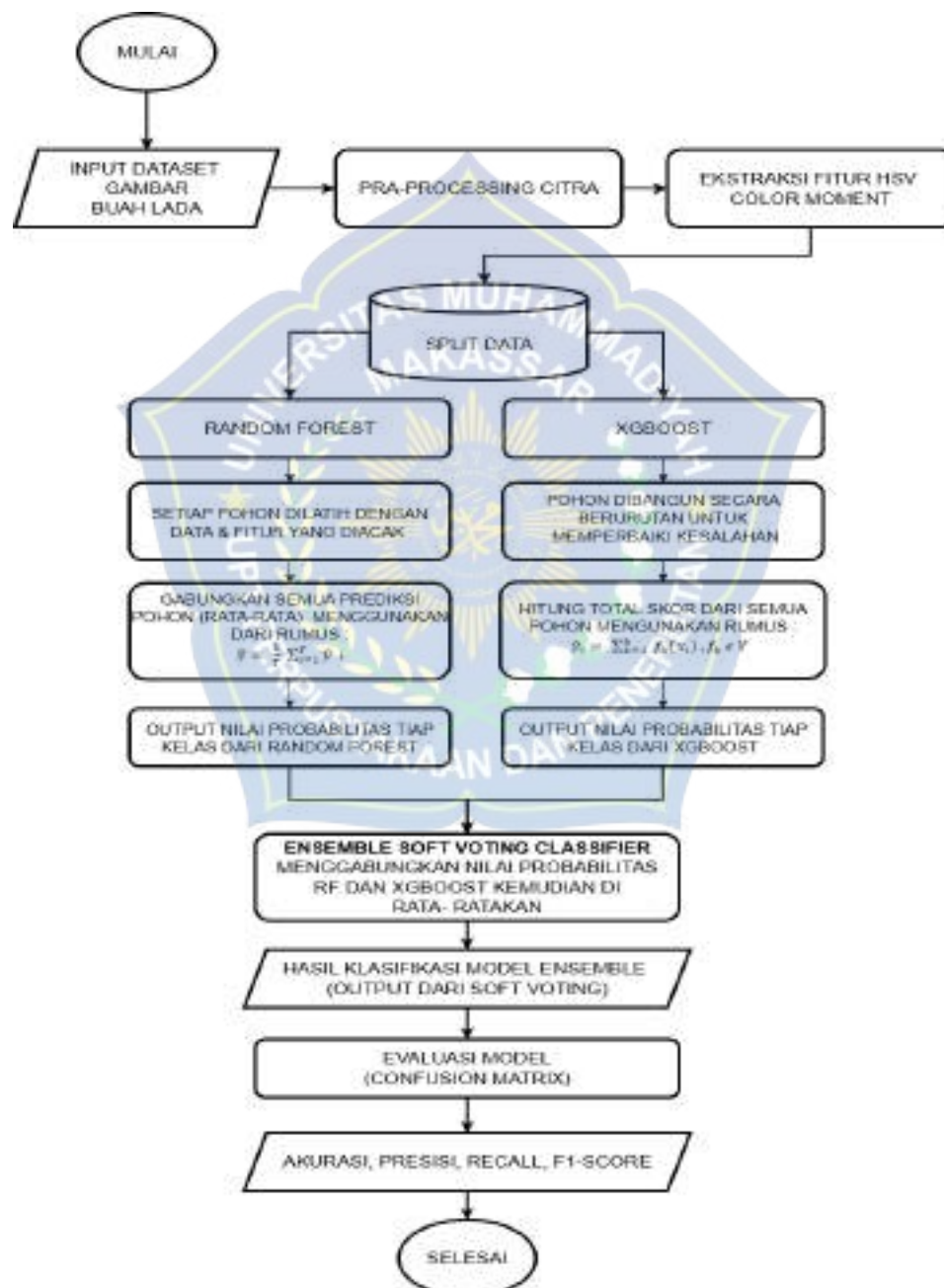
Peralatan dan bahan yang digunakan dalam penelitian ini meliputi:

1. Hardware:
 - a. Smartphone Vivo V40 Lite 5G
 - 1) RAM 8 GB
 - 2) ROM 256 GB
 - 3) Kamera Belakang dengan resolusi 50MP
 - b. Laptop Acer Aspire A514-54G.
 - 1) RAM 8 GB
 - 2) SSD 512 GB
 - 3) Processor Intel Core i3 gen 11
2. Software:
 - a. Python versi 3.12.5
 - b. Visual Studio Code
 - c. OS Windows 11 Home
3. Bahan:
 - a. Dataset citra lada.

C. Perancangan Sistem

Perancangan sistem merupakan proses penyusunan langkah-langkah terstruktur untuk membangun model klasifikasi yang akurat. Dalam penelitian

ini, tahapan meliputi akuisisi citra buah lada, konversi warna dari RGB ke HSV, serta pengembangan model menggunakan metode Ensemble Learning dengan pendekatan Soft Voting Classifier, yang menggabungkan prediksi dari Random Forest dan XGBoost. Seluruh implementasi dilakukan menggunakan bahasa pemrograman Python pada Visual Studio Code. Rancangan sistem secara keseluruhan ditampilkan pada Gambar 7.



Gambar 7. Flowchart Perancangan Sistem

Berikut adalah penjelasan mengenai flowchart perancangan sistem

1. Mulai

Proses dimulai dengan penyusunan alur kerja klasifikasi berbasis citra buah lada.

2. Dataset Citra Buah Lada

Dataset yang digunakan terdiri dari citra buah lada yang dikumpulkan secara langsung melalui dokumentasi lapangan. Setiap citra dikategorikan berdasarkan tingkat kematangan, yaitu belum matang, setengah matang, matang, dan lewat matang. Dataset ini menjadi sumber utama dalam proses pengolahan dan klasifikasi.

3. Pra-Processing Citra

Setiap citra mengalami tahapan pra-pemrosesan, yang terdiri dari:

- a. Segmentasi ROI: memfokuskan citra pada area objek utama, yaitu buah lada, dengan cara memotong bagian citra yang mengandung objek dan menghilangkan latar belakang atau area lain yang tidak berkaitan. Langkah ini bertujuan agar proses ekstraksi fitur dan klasifikasi hanya dilakukan pada bagian citra yang benar-benar mewakili objek penelitian.
- b. Resize Citra: mengubah ukuran seluruh citra menjadi 224×224 piksel untuk memastikan setiap gambar memiliki dimensi yang seragam. Standarisasi ini bertujuan mempermudah proses ekstraksi fitur warna dan memastikan konsistensi dalam pemrosesan oleh model klasifikasi.
- c. Augmentasi Data: dilakukan untuk memperbanyak dan menyeimbangkan jumlah data antar kelas. Data ditambah dengan teknik seperti rotasi, flipping, perubahan pencahayaan, dan pergeseran posisi citra, agar distribusi data lebih seimbang dan model tidak bias terhadap kelas tertentu.
- d. Normalisasi: menyesuaikan skala nilai piksel citra ke dalam rentang 0 hingga 1 dengan cara membagi setiap nilai piksel dengan nilai maksimum (255). Langkah ini dilakukan untuk menyamakan skala data, sehingga proses pembelajaran mesin menjadi lebih stabil dan menghasilkan kinerja model yang optimal.

4. Ekstraksi Fitur HSV dan Color Moments

Citra diekstraksi fiturnya menggunakan kombinasi HSV dan *Color Moment*. Nilai rata-rata (*mean*) dari *Hue*, *Saturation*, dan *Value* diambil sebagai fitur HSV. Sementara itu, *Color Moment* dihitung dari setiap channel HSV menggunakan tiga statistik, yaitu *mean*, *standard deviation*, dan *skewness*. Gabungan kedua fitur ini digunakan sebagai input model untuk menghasilkan representasi warna yang lebih informatif.

5. Split Data

Data dibagi menjadi dua kelompok utama, yaitu data latih dan data uji. Data latih digunakan dalam proses pelatihan model agar model dapat mempelajari pola-pola dari data yang tersedia. Sementara itu, data uji dimanfaatkan untuk menguji dan mengevaluasi performa model yang telah dilatih. Pembagian ini biasanya dilakukan secara proporsional, 80% untuk data latih dan 20% untuk data uji, agar proses pelatihan dan pengujian berlangsung optimal serta hasil evaluasi model dapat lebih representatif.

6. Machine Learning Models: Random Forest dan XGBoost

Dua algoritma machine learning digunakan dalam penelitian ini:

- a. *Random Forest* merupakan algoritma *ensemble* berbasis *decision tree* yang menghasilkan keputusan berdasarkan voting dari sejumlah pohon keputusan. Setiap pohon dalam *Random Forest* memberikan hasil prediksi, dan algoritma ini mengeluarkan *output* berupa probabilitas dari setiap kelas, yang dihitung berdasarkan proporsi pohon yang memilih kelas tersebut.
- b. *XGBoost* adalah algoritma *gradient boosting* yang bekerja dengan meningkatkan akurasi melalui proses pembelajaran dari kesalahan model sebelumnya secara bertahap. Setelah dilatih, *XGBoost* menghasilkan *output* berupa probabilitas kelas berdasarkan kombinasi bobot dari seluruh pohon yang dibentuk selama proses boosting.

7. Setiap Pohon Dilatih Dengan Data dan Fitur Yang Diacak

Dalam Random Forest, model terdiri dari sekitar 100 hingga 400 pohon keputusan, tergantung hasil tuning. Setiap pohon dilatih menggunakan subset

acak dari data dan fitur, membuatnya unik dan membantu mengurangi overfitting serta meningkatkan generalisasi.

8. Pohon Dibangun Secara Berurutan Untuk Memperbaiki Kesalahan

Pada XGBoost, model juga terdiri dari sekitar 100 hingga 400 pohon boosting. Namun, tidak seperti Random Forest, pohon-pohon ini dibangun secara berurutan, di mana masing-masing pohon bertugas memperbaiki kesalahan prediksi dari pohon sebelumnya, menjadikan prosesnya lebih adaptif terhadap kesalahan.

9. Gabungkan Semua Prediksi Pohon (Rata-Rata)

Random Forest menghasilkan output akhir dengan menggabungkan seluruh hasil prediksi dari setiap pohon secara rata-rata, sehingga menghasilkan nilai probabilitas untuk tiap kelas berdasarkan konsensus seluruh pohon.

10. Hitung Total Skor Dari Semua Pohon

XGBoost menghitung output akhir berdasarkan akumulasi skor dari semua pohon yang telah dibangun, di mana setiap skor kontribusi dari masing-masing pohon akan mempengaruhi hasil klasifikasi akhir dalam bentuk nilai probabilitas tiap kelas.

11. Prediksi dan *Output* nilai Probabilitas dari Model

- a. Random Forest menghasilkan *output* berupa nilai probabilitas untuk setiap kelas pada setiap data uji.
- b. XGBoost juga menghasilkan *output* berupa nilai probabilitas untuk setiap kelas pada setiap data uji.

12. *Soft Voting Classifier*

Soft voting digunakan sebagai metode *ensemble* dengan cara menggabungkan hasil prediksi probabilitas dari *Random Forest* dan *XGBoost*. Output probabilitas dari masing-masing model dikombinasikan, dan kelas dengan probabilitas tertinggi dipilih sebagai hasil klasifikasi akhir. Metode ini memastikan hasil klasifikasi lebih stabil dan representatif karena mempertimbangkan kekuatan kedua model.

13. Hasil Klasifikasi Model *Ensemble* (Output dari *soft voting*)

Hasil klasifikasi akhir diperoleh dari *output soft voting*, yang merupakan keputusan bersama berdasarkan probabilitas prediksi dari kedua model. Tahap ini menghasilkan label kelas yang menjadi hasil klasifikasi .

14. Evaluasi Model Confusin matriks

Confusion matrix digunakan untuk mengevaluasi performa klasifikasi dengan melihat jumlah prediksi yang benar dan salah dari masing-masing kelas.

15. Akurasi, Presisi, recall dan F1-Score

Dari confusion matrix, diperoleh metrik evaluasi seperti akurasi, presisi, recall, dan F1-score yang memberikan gambaran detail tentang ketepatan dan keseimbangan model dalam melakukan klasifikasi.

16. Selesai

Setelah evaluasi performa selesai, proses perancangan dan pengujian model dinyatakan selesai.

D. Teknik Pengujian Sistem

Pengujian dalam penelitian ini dilakukan untuk mengevaluasi kinerja model klasifikasi yang telah dibangun menggunakan metode pembelajaran mesin. Data uji yang digunakan merupakan data yang belum pernah dilibatkan dalam proses pelatihan, sehingga hasil pengujian mampu memberikan gambaran objektif tentang kemampuan model dalam menghadapi data baru. Pengujian ini tidak hanya bertujuan untuk memastikan bahwa model bekerja optimal pada data pelatihan, tetapi juga untuk mengukur kemampuannya melakukan generalisasi terhadap kondisi nyata. Dengan demikian, proses ini menjadi tolok ukur penting untuk mengetahui apakah model sudah mampu menghasilkan prediksi yang akurat, bebas dari overfitting, dan dapat diandalkan. Seluruh proses pengujian ini dapat direalisasikan secara langsung dalam bentuk kode program menggunakan bahasa *Python* di *Visual Studio Code*. Implementasi berbasis *Python* memungkinkan setiap tahapan, mulai dari pra-pemrosesan citra, pelatihan model, hingga evaluasi kinerja, dijalankan secara terstruktur dan terukur dalam satu alur sistematis.

Flowchart disusun untuk memberikan visualisasi terstruktur mengenai setiap tahapan yang dilalui dalam sistem klasifikasi ini. Proses dimulai dengan pengumpulan dataset citra buah lada yang kemudian masuk ke tahap pra-pemrosesan, meliputi segmentasi ROI, penyesuaian ukuran (*resize*), augmentasi data untuk penyeimbangan jumlah antar kelas, dan normalisasi piksel. Selanjutnya, dilakukan ekstraksi fitur warna menggunakan konversi citra dari RGB ke HSV, di mana nilai *Hue*, *Saturation*, dan *Value* digunakan sebagai fitur utama. Selain itu, diterapkan juga ekstraksi momen warna (*color moment*) yang menghitung nilai statistik seperti rata-rata (*mean*), standar deviasi, dan *skewness* dari masing-masing kanal warna, sehingga menghasilkan fitur warna yang lebih representatif. Setelah fitur diperoleh, dataset dibagi menjadi data latih dan data uji. Tahap pelatihan dilakukan menggunakan dua model, yaitu *Random Forest* dan *XGBoost*, yang masing-masing menghasilkan output berupa nilai probabilitas untuk setiap kelas pada data uji yang diberikan.

Setelah kedua model menghasilkan prediksi, tahap selanjutnya adalah proses penggabungan hasil melalui metode *soft voting classifier*. Pada metode ini, nilai probabilitas yang dihasilkan oleh *Random Forest* dan *XGBoost* dirata-ratakan, kemudian kelas dengan nilai probabilitas tertinggi ditetapkan sebagai hasil klasifikasi akhir. Hasil *ensemble* inilah yang menjadi keputusan akhir dalam mengkategorikan tingkat kematangan buah lada. Langkah terakhir adalah evaluasi kinerja model yang dilakukan terhadap data uji, menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-Score*.

E. Teknik Analisis Data

Teknik analisis data dalam penelitian ini meliputi metode untuk mengolah dan menganalisis data citra yang dikumpulkan. Tujuannya adalah memaksimalkan penggunaan data citra dalam model klasifikasi guna menghasilkan hasil yang akurat dan menjawab masalah penelitian.

1. Pengumpulan data

Pengumpulan data merupakan tahap awal dalam penelitian yang dilakukan dengan cara mendokumentasikan langsung citra buah lada di

lapangan menggunakan kamera digital. Proses ini mencakup pengambilan gambar buah lada dalam berbagai tingkat kematangan, yaitu belum matang, setengah matang, matang, dan lewat matang. Setiap citra diambil dengan memperhatikan kualitas gambar, seperti ketajaman, pencahayaan, serta gangguan visual seperti blur atau noise, agar data yang diperoleh layak digunakan untuk proses ekstraksi fitur dan pelatihan model klasifikasi.

2. Pemrosesan Citra

Pemrosesan citra adalah teknik untuk mengolah gambar digital agar informasi penting bisa diekstraksi dan dianalisis dengan efektif. Dalam penelitian ini, pemrosesan citra digunakan untuk menyiapkan data citra lada agar konsisten dari segi ukuran dan format warna, seperti konversi dari RGB ke HSV, normalisasi serta ekstraksi momen warna. Proses ini penting untuk menghasilkan fitur warna yang representatif, sehingga meningkatkan akurasi dan kestabilan model klasifikasi tingkat kematangan lada.

3. Evaluasi Model dan Penyajian Data

Evaluasi model adalah proses mengukur dan menilai performa model yang digunakan dalam penelitian, misalnya model klasifikasi tingkat kematangan lada. Teknik evaluasi menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score* untuk mengetahui sejauh mana model mampu memberikan hasil yang benar dan konsisten.

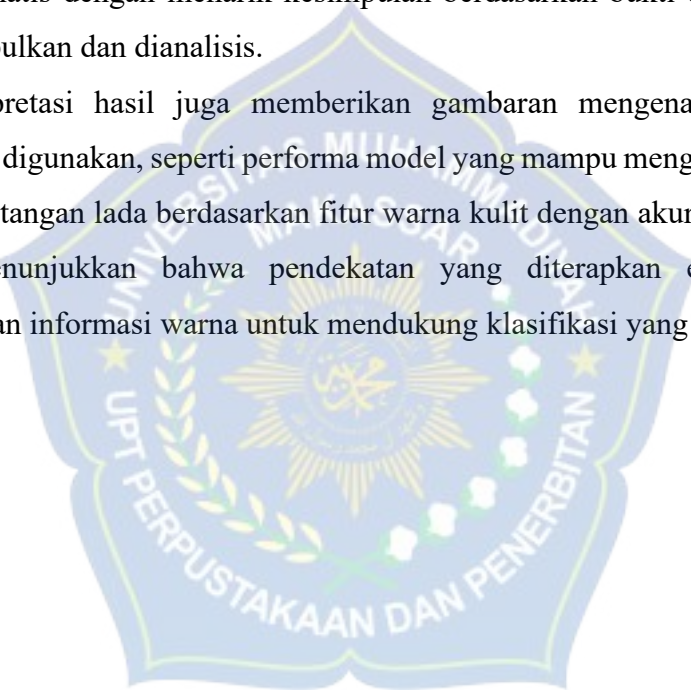
Evaluasi performa juga didasarkan pada *confusion matrix* yang memberikan gambaran detail mengenai kemampuan model dalam mengklasifikasikan masing-masing kelas secara tepat, serta membantu mengidentifikasi jenis kesalahan klasifikasi yang terjadi. Selain itu, evaluasi juga dilakukan terhadap hasil klasifikasi yang diperoleh menggunakan teknik *soft voting*, sebagai metode penggabungan dari dua model yang digunakan.

Penyajian data hasil evaluasi dilakukan melalui tabel dan diagram agar informasi mengenai performa model menjadi lebih jelas dan mudah dianalisis, baik untuk membandingkan antar model maupun untuk menilai efektivitas metode ensemble yang digunakan.

4. Interpretasi Hasil

Interpretasi hasil merupakan tahap penting dalam penelitian yang berfungsi untuk menganalisis dan memahami makna dari temuan yang diperoleh selama proses pengumpulan data hingga evaluasi model. Proses ini tidak sekadar menyajikan angka atau metrik evaluasi, tetapi lebih menekankan pada pemahaman mendalam mengenai arti hasil tersebut dalam konteks permasalahan yang diteliti. Interpretasi hasil membantu menjawab rumusan masalah penelitian secara sistematis dengan menarik kesimpulan berdasarkan bukti dan data yang telah dikumpulkan dan dianalisis.

Interpretasi hasil juga memberikan gambaran mengenai keunggulan metode yang digunakan, seperti performa model yang mampu mengklasifikasikan tingkat kematangan lada berdasarkan fitur warna kulit dengan akurasi yang baik. Hal ini menunjukkan bahwa pendekatan yang diterapkan efektif dalam memanfaatkan informasi warna untuk mendukung klasifikasi yang diinginkan.



BAB IV

HASIL DAN PEMBAHASAN

A. Sumber Data

Dataset berupa gambar buah lada yang diperoleh dengan cara mengambil gambar langsung di perkebunan lada yaitu di desa Tokalimbo, Kecamatan Towuti, Kabupaten Luwu Timur, Sulawesi Selatan. Dataset gambar di ambil dengan menggunakan kamera Vivo V40 Lite. Gambar dataset yang dikumpulkan adalah gambar buah lada belum matang, setengah matang, matang dan lewat matang. Berikut adalah hasil pengambilan gambar buah lada belum matang, setengah matang, matang dan lewat matang.



Gambar 8. Buah Lada Berdasarkan Tingkat Kematangan

B. Pelabelan Data

Pada penelitian ini, pelabelan data dilakukan secara manual berdasarkan tingkat kematangan buah lada. Tujuan pelabelan adalah untuk membandingkan hasil prediksi model dengan label asli, sehingga dapat diketahui apakah prediksi model benar atau tidak. Proses ini penting dalam mengevaluasi akurasi dan kinerja model klasifikasi.



Gambar 9. Pelabelan Data

C. Preprocessing Data

Data preprocessing dilakukan untuk mempersiapkan gambar sebelum digunakan sebagai dataset dalam pemodelan. Penulis melakukan data preprocessing dalam penelitian ini berupa segmentasi roi, hilangkan background, resize, augmentasi dan normalisasi.

1. Penghapusan Latar Belakang

Penghapusan latar belakang bertujuan untuk memisahkan objek utama (buah lada) dari area yang tidak relevan pada gambar, seperti daun, tanah, atau elemen lain. Proses ini dilakukan menggunakan *library rembg*, yang merupakan pustaka Python untuk menghapus *background* secara otomatis. Hasilnya adalah gambar dengan objek utama lada yang tetap terlihat dan latar belakangnya dihilangkan. Berikut adalah perintah *rembg* menggunakan *Python*:

```
from rembg import remove
def hapus_background(image_path):
    with open(image_path, 'rb') as f:
        input_data = f.read()
        output_data = remove(input_data)
```

2. Segmentasi ROI

Setelah proses penghapusan latar belakang, dilakukan segmentasi *Region of Interest (ROI)* untuk mengekstraksi area citra yang secara spesifik memuat objek utama, yaitu buah lada. Tahapan ini bertujuan untuk menghilangkan bagian-bagian citra yang tidak relevan, seperti batang, daun, atau elemen latar lainnya yang dapat mengganggu proses ekstraksi fitur. Dengan menggunakan *library OpenCV (cv2)*, dilakukan identifikasi bounding box yang mengelilingi buah lada, kemudian objek dipotong (*cropping*) agar hanya area penting yang dianalisis lebih lanjut. Adapun fungsi *python* yang digunakan adalah sebagai berikut:

```
def segmentasi_roi(img):
    if img.shape[2] == 4:
        alpha = img[:, :, 3]
```

```
_,mask=cv2.threshold(alpha,0,255, cv2.THRESH_BINARY)
kernel = np.ones((5, 5), np.uint8)
closed_mask= cv2.morphologyEx(mask, cv2.MORPH_CLOSE,
kernel, iterations=2)
contours, _=cv2.findContours(closed_mask,cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

3. Resize

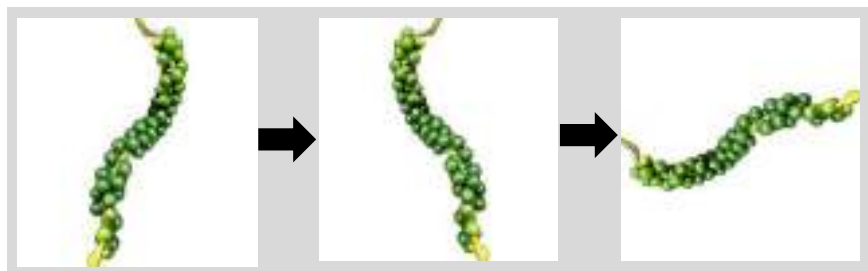
Resize adalah menyesuaikan ukuran gambar hasil ROI menjadi seragam, yaitu 224×224 piksel. Proses *resize* ini menggunakan metode *letterbox* atau *padding* agar aspek rasio objek tetap terjaga tanpa distorsi. *Library cv2* digunakan untuk melakukan *resize*, Langkah ini memastikan bahwa seluruh dataset memiliki dimensi yang konsisten saat digunakan dalam pelatihan model. Berikut adalah contoh gambarnya:



Gambar 10. Hasil Resize dan Segmentasi ROI

4. Augmentasi

Augmentasi citra digunakan untuk meningkatkan keragaman data pelatihan dengan menerapkan transformasi seperti flip horizontal, rotasi 90 dan 180 derajat terhadap hasil flip, menggunakan *library OpenCV*. Setiap gambar asli menghasilkan tiga variasi baru, sehingga distribusi data antar kelas menjadi lebih seimbang dan proses pelatihan menjadi lebih efektif. Tampilan citra setelah augmentasi ditunjukkan pada gambar berikut:



Gambar 11. Hasil Augmentasi

Berikut adalah Tabel 2 jumlah data yang berhasil dibuat untuk memperkaya dataset asli lada:

Tabel 2. Jumlah Data

Kelas	Jumlah dataset
Belum Matang	404 gambar
Setengah Matang	400 gambar
Matang	408 gambar
Lewat Matang	384 gambar
Objek Bukan Lada	400 gambar

Tabel di atas menunjukkan hasil jumlah data per kelas setelah proses augmentasi selesai. Data inilah yang digunakan sebagai dasar untuk proses pelatihan dan pengujian model klasifikasi tingkat kematangan lada.

D. Ekstraksi Fitur Color Moments HSV

Bagian ini membahas proses ekstraksi fitur pada dataset gambar buah lada. Ekstraksi fitur bertujuan untuk mengambil informasi penting dari citra yang akan digunakan sebagai input dalam model klasifikasi. Pada penelitian ini, citra buah lada yang telah dikonversi ke ruang warna HSV kemudian diekstraksi fitur statistik warnanya menggunakan metode *Color Moment*. Fitur-fitur ini diharapkan mampu merepresentasikan karakteristik warna pada setiap tingkat kematangan buah lada.

1. Konversi Warna RGB ke HSV

Konversi warna dari RGB ke HSV dilakukan menggunakan *library OpenCV* dengan fungsi *cv2.cvtColor()*. Model RGB memiliki tiga kanal (*Red, Green, Blue*) dengan rentang nilai 0 hingga 255. Sedangkan model HSV terdiri dari *Hue* (0–179), *Saturation* (0–255), dan *Value* (0–255). Perbedaan skala ini mengikuti standar *OpenCV* untuk efisiensi penyimpanan. Konversi ini bertujuan memisahkan informasi warna agar lebih sesuai digunakan dalam proses ekstraksi fitur. Berikut adalah tabel contoh nilai dalam setiap piksel di ruang warna RGB:

Tabel 3. Contoh Nilai Piksel RGB

Label	R	G	B
Lada Belum Matang	239	242	236
Lada Lewat Matang	238	237	237
Lada Matang	250	227	224
Lada Setengah Matang	236	238	220
Objek Bukan Lada	218	200	191

Perbedaan nilai piksel RGB antar kelas mencerminkan perubahan warna kulit buah lada. Konversi ke HSV dilakukan untuk memperoleh representasi warna yang lebih stabil dan terpisah berdasarkan hue, saturation, dan value.

2. Normalisasi Data

Normalisasi merupakan salah satu tahapan penting dalam proses preprocessing data citra, yang bertujuan untuk menyamakan skala antar fitur dan meningkatkan konsistensi data. Dalam penelitian ini, penulis melakukan normalisasi setelah terlebih dahulu mengubah ruang warna citra dari RGB ke HSV, karena HSV lebih mampu merepresentasikan informasi warna secara terpisah antara rona (Hue), kejenuhan (Saturation), dan kecerahan (Value). Penulis melakukan pembagian nilai kanal Hue dengan 179, serta kanal Saturation dan Value masing-masing dengan 255, sesuai dengan batas maksimum nilai HSV pada format *OpenCV*. Hasil dari proses ini membuat seluruh nilai HSV berada dalam rentang 0 hingga 1, sehingga skala antar fitur menjadi seragam. Penulis melakukan langkah ini untuk mempersiapkan data agar lebih terstruktur dan mudah diolah pada tahap selanjutnya, sekaligus mengurangi potensi bias yang mungkin muncul akibat perbedaan skala antar fitur warna.

3. Ekstraksi Fitur Color Moment HSV

Pada Penelitian ini, setelah gambar dinormalisasi, dilakukan ekstraksi fitur untuk mendapatkan informasi statistik dari citra HSV. Fitur yang diambil adalah *Color Moment*, berupa *mean*, standar deviasi, dan *skewness* dari tiap kanal HSV. Proses ini menggunakan *library NumPy* untuk menghitung mean dan standar deviasi, serta *SciPy* untuk menghitung *skewness*. Hasil ekstraksi disimpan dalam

bentuk CSV sebagai data input model klasifikasi. Berikut adalah Tabel 4 hasil ekstraksi fitur HSV:

Tabel 4. Hasil Ekstraksi Fitur HSV

NO	Nama File	Mean H	Mean S	Mean V	Std H	Std S	Std V	Skew H	Skew S	Skew V	Label
1	1BM.png	0.032	0.079	0.946	0.092	0.245	0.165	29.10	30.61	-33.03	lada belum matang
2	2BM.png	0.041	0.078	0.917	0.0857	0.204	0.177	17.41	27.67	-21.58	lada belum matang
3	3LM.png	0.065	0.0148	0.920	0.196	0.048	0.210	30.01	44.70	-24.95	lada lewat matang
4	4LM.png	0.054	0.0205	0.957	0.196	0.086	0.154	3.578	57.53	-39.12	lada lewat matang
5	5M.png	0.062	0.150	0.889	0.212	0.309	0.247	37.57	18.61	-23.22	lada matang
6	6M.png	0.017	0.093	0.934	0.053	0.273	0.193	41.65	27.47	-2.976	lada matang
7	7SM.png	0.015	0.075	0.965	0.047	0.248	0.122	29.81	31.63	-39.54	lada setengah matang
8	8SM.png	0.0388	0.2033	0.898	0.072	0.369	0.204	30.23	14.17	-20.66	lada setengah matang
9	9.png	0.264	0.112	0.765	0.339	0.202	0.325	0.869	21.39	-11.35	objek bukan lada
10	10.png	0.132	0.202	0.826	0.272	0.289	0.255	22.67	11.21	-11.36	objek bukan lada
11	11BM.png	0.023	0.097	0.946	0.067	0.286	0.162	29.21	27.07	-33.50	lada belum matang
12	12SM.png	0.019	0.082	0.958	0.053	0.233	0.120	3.874	2.728	-32.32	lada setengah matang
13	13M.png	0.026	0.102	0.942	0.113	0.282	0.1650	70.57	2.555	-30.32	lada matang
14	14LM.png	0.087	0.035	0.930	0.245	0.112	0.196	26.10	4.287	-2.872	lada lewat matang
15	15.png	0.252	0.389	0.793	0.294	0.402	0.253	0.668	0.328	-0.904	objek bukan lada
...	...	...	...	...	...	...	...	...	...	...	...
1996	100.png	0.354	0.139	0.829	0.432	0.432	0.219	0.567	0.960	-1.017	objek bukan lada

Tabel 4 menampilkan hasil ekstraksi fitur momen warna dari gambar buah lada. Ekstraksi dilakukan pada ruang warna HSV dengan menghitung nilai *mean*, standar deviasi, dan *skewness* pada masing-masing kanal. Nilai-nilai ini digunakan sebagai fitur numerik yang mewakili karakteristik warna untuk proses klasifikasi tingkat kematangan. Berikut Tabel 5 penjelasan *library python* apa saja yang dipakai oleh penulis untuk melakukan pemodelan ekstraksi fiturnya.

Tabel 5. Library untuk Ekstraksi Fitur HSV

Library	Fungsi
NumPy	Menghitung mean dan standar deviasi
SciPy	Menghitung skewness
OS	Mengakses dan membaca file dari folder
CSV	Menyimpan hasil ekstraksi dalam file CSV

Tabel 5 menjelaskan fungsi dari masing-masing library Python yang digunakan dalam proses ekstraksi fitur citra pada penelitian ini. Setiap library memiliki peran spesifik, mulai dari perhitungan statistik (*NumPy* dan *SciPy*), pengelolaan file (OS), hingga penyimpanan hasil (CSV).

E. Pembagian Data

Penelitian ini menggunakan rasio pembagian data sebesar 80% untuk data latih dan 20% untuk data uji. Data latih digunakan untuk membangun dan melatih model klasifikasi tingkat kematangan lada, sedangkan data uji digunakan untuk mengukur akurasi dan ketepatan prediksi model yang telah dilatih. Pembagian dilakukan secara *stratified*, yaitu mempertahankan proporsi kelas yang seimbang, agar distribusi label pada data latih dan uji tetap konsisten.

Secara keseluruhan, dataset yang digunakan dalam penelitian ini terdiri dari 1.996 data citra. Setelah dilakukan pembagian, sebanyak 1.596 data dialokasikan untuk proses pelatihan model, sedangkan 400 data sisanya digunakan untuk pengujian.

F. Pengujian Model

Penelitian ini menguji model klasifikasi menggunakan Random Forest dan XGBoost untuk mengidentifikasi tingkat kematangan lada berdasarkan fitur warna HSV. Kedua model dilatih secara terpisah, lalu digabungkan menggunakan metode *Soft Voting Ensemble*, yang menggabungkan prediksi probabilitas dari masing-masing model dan memilih kelas dengan probabilitas tertinggi. Berikut adalah metode yang digunakan dalam pengujian:

1. Random Forest

Dalam penelitian ini, algoritma *Random Forest* digunakan untuk mengklasifikasikan tingkat kematangan buah lada berdasarkan fitur warna HSV. *Random Forest* merupakan metode *ensemble* berbasis pohon keputusan yang bekerja dengan membentuk banyak pohon dari subset data latih yang dipilih secara acak, baik dari sisi data maupun fitur. Setiap pohon menghasilkan prediksi masing-masing, kemudian hasil akhir ditentukan melalui voting mayoritas untuk klasifikasi. Proses implementasi dilakukan menggunakan library Python seperti *pandas*, *numpy*, *matplotlib*, *seaborn*, dan *scikit-learn*, yang mendukung tahapan pemrosesan data, pelatihan model, *tuning hyperparameter*, evaluasi, hingga visualisasi. Tabel 6 merangkum fungsi dari masing-masing library yang digunakan dalam implementasi algoritma ini.

Tabel 6. Library Model Random Forest

Library	Fungsi
pandas	Membaca data dalam format tabel (CSV).
numpy	Operasi numerik, termasuk perhitungan array dan statistik dasar.
matplotlib.pyplot	Membuat grafik visual confusion matrix.
seaborn	Visualisasi statistik (heatmap confusion matrix).
sklearn.ensemble. RandomForestClassifier	Membuat model klasifikasi menggunakan algoritma Random Forest.

sklearn.metrics	Menghitung metrik evaluasi: akurasi, presisi, recall, F1-score, dan confusion matrix.
-----------------	---

Untuk menguji kemampuan model *Random Forest* dalam mengklasifikasikan tingkat kematangan buah lada, dilakukan pengujian terhadap data uji yang telah dipisahkan sebelumnya. Hasil prediksi model tidak hanya menghasilkan label kelas akhir, tetapi juga memberikan informasi probabilitas untuk setiap kelas yang tersedia. Hal ini dapat di analisis lebih mendalam terhadap keyakinan model dalam membuat keputusan klasifikasi. Label prediksi ditentukan berdasarkan nilai probabilitas tertinggi dari kelima kelas yang ada, sehingga kelas dengan kemungkinan terbesar menurut model akan dipilih sebagai hasil klasifikasi akhir. Tabel 7 menyajikan beberapa contoh hasil prediksi model *Random Forest* :

Tabel 7. Output Nilai Probabilitas Random Forest

No	Nama File	Label Asli	Prob belum matang	Prob lewat matang	Prob matang	Prob setengah matang	Prob objek bukan lada	Label prediksi RF
1	3LM.png	Lada Lewat Matang	0	1	0	0	0	Lada Lewat Matang
2	26.png	Objek Bukan Lada	0	0	0	0.005	0.995	Objek Bukan Lada
3	84M.png	Lada Matang	0	0	0.995	0.005	0	Lada Matang
4	29SM.png	Lada Setengah Matang	0	0	0.055	0.905	0	Lada Setengah Matang
5.	12SM.png	Lada Setengah Matang	0.015	0.005	0.025	0.95	0.0005	Lada Setengah Matang
...	...	...	...	...	...	...	...	...
400	68BM.png	Lada Belum Matang	0.99	0	0.01	0	0	Lada Belum Matang

2. XGBoost

Pada penelitian ini, algoritma XGBoost (*Extreme Gradient Boosting*) digunakan sebagai metode klasifikasi untuk mengidentifikasi tingkat kematangan buah lada berdasarkan fitur warna HSV. XGBoost bekerja dengan membangun pohon keputusan secara bertahap, di mana setiap pohon baru difokuskan untuk memperbaiki kesalahan dari model sebelumnya menggunakan pendekatan *gradient boosting*. Dalam implementasinya, digunakan kombinasi library Python seperti *xgboost*, *pandas*, *numpy*, *matplotlib*, *seaborn*, serta *scikit-learn* untuk proses preprocessing, pelatihan model, *tuning hyperparameter*, evaluasi, dan visualisasi.

Masing-masing *library* memiliki fungsi yang mendukung proses *end-to-end* mulai dari input data hingga evaluasi hasil klasifikasi. Tabel 8 adalah *library Python* yang digunakan dalam implementasi algoritma XGBoost pada penelitian ini.

Tabel 8. Library Model XGBoost

Library	Fungsi
pandas	Membaca data dalam format tabel (CSV).
numpy	Operasi numerik, termasuk perhitungan array dan statistik dasar.
matplotlib.pyplot	Membuat grafik visual confusion matrix.
seaborn	Visualisasi statistik (heatmap confusion matrix).
sklearn.preprocessing. LabelEncoder	Mengubah label kategori menjadi numerik.
xgboost.XGBClassifier	Membangun model klasifikasi dengan algoritma XGBoost.
sklearn.metrics	Menghitung metrik evaluasi: akurasi, presisi, recall, F1-score, dan confusion matrix.

Setelah proses pelatihan menggunakan algoritma XGBoost, dilakukan pengujian terhadap data uji untuk memperoleh hasil klasifikasi tingkat kematangan lada. Tabel berikut menyajikan hasil prediksi model XGBoost, yang mencakup nama file citra, label asli, probabilitas untuk setiap kelas, serta label akhir yang dipilih berdasarkan nilai probabilitas tertinggi. Hasil ini digunakan untuk mengevaluasi performa model dalam mengidentifikasi kelas yang paling sesuai berdasarkan fitur warna HSV dari setiap citra. Hasil prediksi tersebut dapat dilihat pada Tabel 9.

Tabel 9. Output Nilai Probabilitas XGBoost

No	Nama File	Label Asli	Prob belum matang	Prob lewat matang	Prob matang	Prob setengah matang	Prob objek bukan lada	Label prediksi XGBoost
1	3LM.png	Lada Lewat Matang	0.000023	0.99	0.00004	0.00001	0.00004	Lada Lewat Matang
2	26.png	Objek Bukan Lada	0.000002	0.00006	0.00001	0.00001	0.99	Objek Bukan Lada
3	84M.png	Lada Matang	0.000001	0.00004	0.99	0.000003	0.000007	Lada Matang
4	29SM.png	Lada Setengah Matang	0.00016	0.0002	0.0009	0.99	0.00012	Lada Setengah Matang
5.	12SM.png	Lada Setengah Matang	0.0002	0.00002	0.002	0.99	0.00008	Lada Setengah Matang
...	...	...	...	...	...	...	...	...
400	68BM.png	Lada Belum Matang	0.99	0.00044	0.00011	0.0002	0.0002	Lada Belum Matang

3. Soft Voting Ensemble

Pada tahapan ini, dilakukan proses Soft Voting Ensemble dengan menggabungkan hasil prediksi probabilitas dari dua model yaitu Random Forest dan XGBoost. Proses ini menghitung rata-rata dari probabilitas setiap kelas yang dihasilkan oleh kedua model, kemudian kelas dengan probabilitas tertinggi dipilih

sebagai hasil akhir prediksi. Metode ini bertujuan untuk meningkatkan performa klasifikasi dengan memanfaatkan kekuatan dari dua model yang berbeda.

Berikut adalah library-library Python berikut digunakan dalam proses pengujian model ensemble Soft Voting, yang berperan penting dalam penggabungan prediksi model, evaluasi performa klasifikasi, serta penyajian visualisasi hasil secara komprehensif. Tabel 10 adalah *library Python* yang digunakan dalam implementasi Soft Voting Ensemble pada penelitian ini.

Tabel 10. Library Soft Voting Ensemble

Library	Fungsi
pandas	Membaca data dalam format tabel (CSV).
numpy	Operasi numerik, termasuk perhitungan array dan statistik dasar.
matplotlib.pyplot	Membuat grafik visual seperti kurva dan confusion matrix.
seaborn	Visualisasi statistik (heatmap confusion matrix).
sklearn.metrics	Menghitung metrik evaluasi: akurasi, presisi, recall, F1-score, dan confusion matrix.

Setelah model ensemble diterapkan dengan menggabungkan prediksi dari beberapa algoritma, dilakukan pengujian terhadap data uji untuk memperoleh hasil klasifikasi tingkat kematangan buah lada. Model ensemble ini menghitung rata-rata probabilitas dari masing-masing model dasar, kemudian menentukan label akhir berdasarkan nilai probabilitas tertinggi. Tabel berikut menyajikan hasil prediksi model ensemble, yang memuat nama file citra, label asli, probabilitas rata-rata untuk setiap kelas, serta label akhir yang dihasilkan. Informasi ini digunakan untuk menilai efektivitas pendekatan ensemble dalam mengklasifikasikan citra berdasarkan karakteristik warna HSV. Hasil prediksi tersebut dapat dilihat pada Tabel 11.

Tabel 11. Output Nilai Probabilitas Ensemble Soft Voting

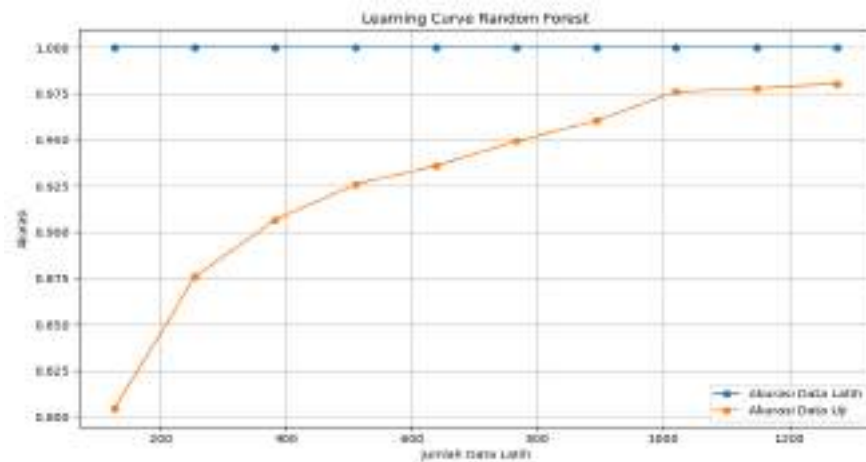
No	Nama File	Label Asli	Prob belum matang	Prob lewat matang	Prob matang	Prob setengah matang	Prob objek bukan lada	Label prediksi Ensemble Soft Voting
1	3LM.png	Lada Lewat Matang	0.00001	0.99	0.00002	0.00008	0.00002	Lada Lewat Matang
2	26.png	Objek Bukan Lada	0.000001	0.00003	0.00007	0.0025	0.99	Objek Bukan Lada
3	84M.png	Lada Matang	0.000009	0.00002	0.99	0.0025	0.000003	Lada Matang
4	29SM.png	Lada Setengah Matang	0.020	0.00012	0.027	0.95	0.000006	Lada Setengah Matang
5.	12SM.png	Lada Setengah Matang	0.00011	0.00001	0.016	0.98	0.00004	Lada Setengah Matang
...	...	...	...	...	...	...	...	...
400	68BM.png	Lada Belum Matang	0.99	0.0002	0.0050	0.0014	0.0001	Lada Belum Matang

G. Hasil Pengujian Model

Hasil pengujian model merupakan tahap evaluasi untuk menilai sejauh mana model klasifikasi yang telah dibangun mampu mengenali dan mengelompokkan data uji dengan benar. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, *F1-score*, serta visualisasi melalui *confusion matrix* dan *learning curve*. Adapun model yang di uji meliputi:

1. Random forest

Model klasifikasi tingkat kematangan lada telah diuji menggunakan data testing. Pengujian dilakukan untuk mengetahui sejauh mana performa model dalam mengenali kelas dengan tepat. Berikut adalah gambaran performa model secara visual yang ditampilkan dalam bentuk kurva dan *confusion matrix*:



Gambar 12. Kurva Random Forest

Berdasarkan grafik *learning curve* Random Forest, terlihat bahwa akurasi pada data uji meningkat secara signifikan seiring dengan bertambahnya jumlah data. Hal ini menunjukkan bahwa model mampu belajar lebih baik ketika jumlah data meningkat, sehingga generalisasi terhadap data uji menjadi lebih akurat. Sementara itu, akurasi data latih tetap tinggi, yang menunjukkan bahwa model mampu menangkap pola dari data latih secara konsisten. Meningkatnya akurasi data uji ini juga mengindikasikan bahwa penambahan data pelatihan berdampak positif terhadap performa model hingga mencapai titik konvergensi. Kemudian untuk visualisasi confusion matrix model random forest dapat dilihat pada gambar 13.



Gambar 13. Confusion Matrix Random Forest

Evaluasi kinerja model Random Forest ditunjukkan melalui *confusion matrix* yang terdiri dari *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN).

Pada kelas lada belum matang, model Random Forest mampu mengenali sebanyak 80 data lada belum matang secara benar sehingga diperoleh nilai *True Positive* (TP) = 80. Namun, terdapat kesalahan prediksi yaitu 1 data dari kelas lada matang dan 3 data dari kelas lada setengah matang yang salah diprediksi sebagai lada belum matang, sehingga diperoleh *False Positive* (FP) = 1 + 3 = 4. Selain itu, terdapat 1 data lada belum matang yang salah diprediksi sebagai kelas lada setengah matang sehingga diperoleh *False Negative* (FN) = 1. Dengan demikian, nilai *True Negative* (TN) = 400 – (80 + 4 + 1) = 315, yang berarti seluruh data dari kelas lada matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada belum matang.

Untuk kelas lada lewat matang, model berhasil mengenali seluruh 77 data dengan benar sehingga diperoleh *True Positive* (TP) = 77. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai lada lewat matang, sehingga nilai *False Positive* (FP) = 0, dan semua data lada lewat matang berhasil diprediksi dengan tepat sehingga *False Negative* (FN) = 0. Dengan kondisi ini, maka nilai *True Negative* (TN) = 400 – (77 + 0 + 0) = 323, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan bukan lada yang tidak salah diprediksi sebagai lada lewat matang.

Pada kelas lada matang, model berhasil mengenali 80 data secara benar sehingga diperoleh *True Positive* (TP) = 80. Namun, terdapat kesalahan prediksi yaitu 1 data dari kelas lada belum matang dan 1 data dari kelas lada setengah matang yang salah diprediksi sebagai lada matang, sehingga diperoleh *False Positive* (FP) = 1 + 1 = 2. Selain itu, terdapat 1 data lada matang yang salah diprediksi sebagai lada belum matang dan 1 data lagi salah diprediksi sebagai lada setengah matang, sehingga total *False Negative* (FN) = 1 + 1 = 2. Dengan demikian, nilai *True Negative* (TN) = 400 – (80 + 2 + 2) = 316, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada matang.

Untuk kelas lada setengah matang, model mampu mengenali 75 data secara benar sehingga diperoleh *True Positive* (TP) = 75. Namun, terdapat kesalahan prediksi yaitu 1 data dari kelas lada belum matang dan 1 data dari kelas lada matang yang salah diprediksi sebagai lada setengah matang, sehingga diperoleh *False Positive* (FP) = 1 + 1 = 2. Selain itu, terdapat 3 data lada setengah matang yang salah diprediksi sebagai lada belum matang dan 2 data lagi salah diprediksi sebagai lada matang, sehingga total *False Negative* (FN) = 3 + 2 = 5. Dengan demikian, nilai *True Negative* (TN) = 400 – (75 + 2 + 5) = 318, yang berarti seluruh data dari kelas lada belum matang, lada matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada setengah matang.

Terakhir, pada kelas objek bukan lada, model berhasil mengenali seluruh 80 data dengan benar sehingga diperoleh *True Positive* (TP) = 80. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai objek bukan lada, sehingga nilai *False Positive* (FP) = 0, dan seluruh data objek bukan lada juga tidak ada yang salah diprediksi ke kelas lain, sehingga diperoleh *False Negative* (FN) = 0. Dengan demikian, nilai *True Negative* (TN) = 400 – (80 + 0 + 0) = 320, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan lada lewat matang yang tidak salah diprediksi sebagai bukan lada.

Berikut adalah *classification report* dari model *Random Forest* yang menunjukkan performa klasifikasi terhadap masing-masing kelas berdasarkan nilai akurasi, *precision*, *recall*, dan *f1-score*.

Evaluasi Random Forest:

Akurasi : 0.9800
 Presisi : 0.9804
 Recall : 0.9802
 F1-Score : 0.9801

Laporan Klasifikasi:

	precision	recall	f1-score	support
lada belum matang	0.95	0.99	0.97	81
lada lewat matang	1.00	1.00	1.00	77

lada matang	0.98	0.98	0.98	82
lada setengah matang	0.97	0.94	0.96	80
objek bukan lada	1.00	1.00	1.00	80
accuracy			0.98	400
macro avg	0.98	0.98	0.98	400
weighted avg	0.98	0.98	0.98	400

Evaluasi Random Forest menunjukkan akurasi 98,00% dengan presisi 98,04%, recall 98,02%, dan F1-score 98,01%. Model mampu mengklasifikasikan data dengan sangat baik. Laporan klasifikasi memperlihatkan kelas “lada lewat matang” dan “objek bukan lada” mencapai nilai sempurna. Kelas “lada belum matang” memperoleh presisi 0,95 dan recall 0,99, sedangkan “lada setengah matang” sedikit lebih rendah dengan presisi 0,97 dan recall 0,94. Kolom *support* menunjukkan jumlah data uji tiap kelas dengan total 400 data. Nilai *macro average* merupakan rata-rata sederhana dari tiap kelas, sedangkan *weighted average* mempertimbangkan distribusi data sehingga lebih mewakili kinerja keseluruhan model.

Dari confusion matrix tersebut didapatkan informasi untuk masing-masing kelas, seperti pada Tabel 12.

Tabel 12. Klasifikasi Report Random Forest

Kelas	TP	FP	FN	TN
Lada belum matang	80	4	1	315
Lada lewat matang	77	0	0	323
Lada matang	80	2	2	316
Lada setengah matang	75	2	5	318
Objek bukan lada	80	0	0	320

Kemudian dari tabel ini ditentukanlah jumlah akurasi, presisi, recall dan f1-score:

a) Akurasi

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{80 + 77 + 80 + 75 + 80}{400} = \frac{392}{400} = 0.98$$

Jadi akurasi model random forest adalah 98%

b) Presisi

- 1) $Precision$ (belum matang) = $\frac{TP}{TP+FP} = \frac{80}{80+4} = 0.9524$
- 2) $Precision$ (lewat matang) = $\frac{TP}{TP+FP} = \frac{77}{77+0} = 1.000$
- 3) $Precision$ (matang) = $\frac{TP}{TP+FP} = \frac{80}{80+2} = 0.9756$
- 4) $Precision$ (setengah matang) = $\frac{TP}{TP+FP} = \frac{75}{75+2} = 0.9740$
- 5) $Precision$ (Objek bukan lada) = $\frac{TP}{TP+FP} = \frac{80}{80+0} = 1.000$

Rata rata presisi:

$$\frac{0.9524+1.000+0.9756+0.9740+1.000}{5} = 0.9804$$

Jadi, di dapatkan presisi random forest 98,04%

c) Recall

- 1) $Recall$ (belum matang) = $\frac{TP}{TP+FN} = \frac{80}{80+1} = 0.9877$
- 2) $Recall$ (lewat matang) = $\frac{TP}{TP+FN} = \frac{77}{77+0} = 1.000$
- 3) $Recall$ (matang) = $\frac{TP}{TP+FN} = \frac{80}{80+1} = 0.9756$
- 4) $Recall$ (setengah matang) = $\frac{TP}{TP+FN} = \frac{75}{75+5} = 0.9375$
- 5) $Recall$ (objek bukan lada) = $\frac{TP}{TP+FN} = \frac{80}{80+0} = 1.000$

Rata rata recall:

$$\frac{0.9877+1.000+0.9756+0.9375+1.0000}{5} = 0.9802$$

Jadi di dapatkan recall sebesar 98,02%

d) F1-score

- 1) $F1 - Score$ (belum matang) = $2 \times \frac{(0.9524) \times (0.9877)}{(0.9524 + 0.9877)} = 0.9697$
- 2) $F1 - Score$ (lewat matang) = $2 \times \frac{(1.0000 \times 1.0000)}{(1.0000 + 1.0000)} = 1.000$
- 3) $F1 - Score$ (matang) = $2 \times \frac{(0.9756 \times 0.9756)}{(0.9756 + 0.9756)} = 0.9756$
- 4) $F1 - Score$ (setengah matang) = $2 \times \frac{(0.9740 \times 0.9375)}{(0.9740 + 0.9375)} = 0.9554$
- 5) $F1 - Score$ (objek bukan lada) = $2 \times \frac{(1.0000 \times 1.0000)}{(1.0000 + 1.0000)} = 1.0000$

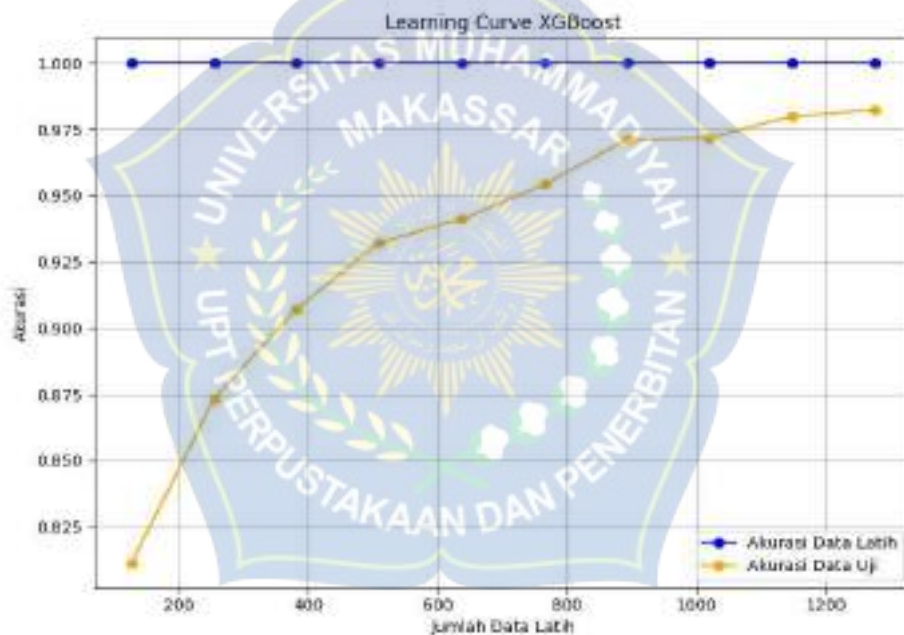
Rata rata F1-Score:

$$\frac{0.9697+1.000+0.9756+0.9554+1.0000}{5} = 0.9801$$

Jadi di dapatkan F1-score sebesar 98,01%

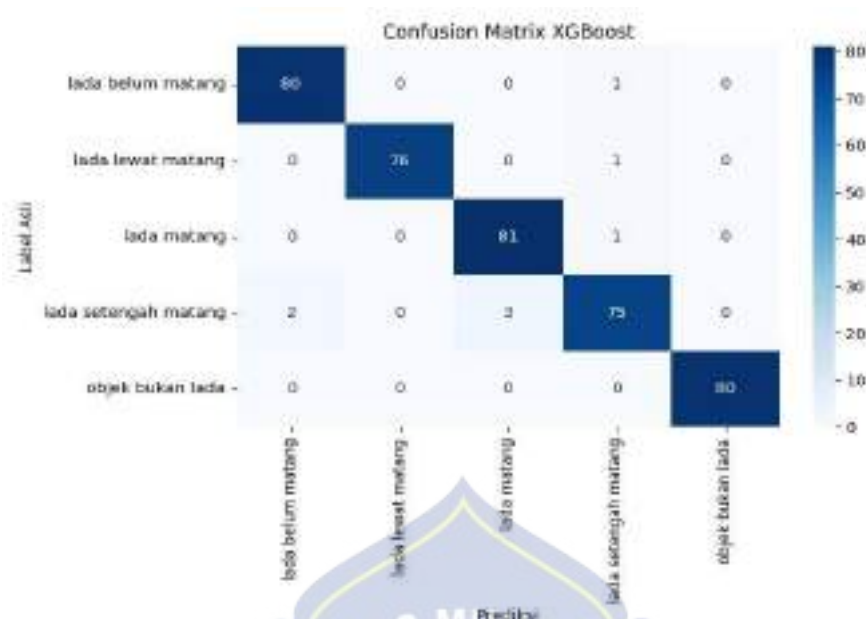
2. XGBoost

Pada penelitian ini, XGBoost digunakan untuk mengklasifikasikan tingkat kematangan buah lada berdasarkan fitur warna HSV. Hasil akhirnya berupa prediksi probabilistik yang digunakan untuk menentukan kelas, menghasilkan model yang akurat dan andal dalam klasifikasi. Berikut adalah gambaran performa model secara visual yang ditampilkan dalam bentuk kurva dan confusion matrix yang dapat dilihat pada Gambar 14.



Gambar 14. Kurva XGBoost

Berdasarkan grafik *learning curve* XGBoost, akurasi data uji meningkat seiring bertambahnya data latih. Hal ini menunjukkan bahwa semakin banyak data yang digunakan, semakin baik model dalam mengenali pola dan melakukan generalisasi. Akurasi data latih tetap tinggi dan stabil, menandakan model tidak overfitting. Peningkatan akurasi uji ini membuktikan bahwa penambahan data pelatihan berdampak positif terhadap performa prediksi. Berikut tampilan confusion matrix pada Gambar 15.



Gambar 15. Confusion Matriks XGBoost

Evaluasi kinerja model XGBoost ditunjukkan melalui *confusion matrix* dengan empat komponen, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN).

Pada kelas lada belum matang, model XGBoost mampu mengenali sebanyak 80 data dengan benar sehingga diperoleh *True Positive* (TP) = 80. Terdapat 2 data dari kelas lada setengah matang yang salah diprediksi sebagai lada belum matang sehingga *False Positive* (FP) = 2. Terdapat 1 data lada belum matang yang salah diprediksi sebagai lada setengah matang sehingga *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari total data uji sebanyak 400, yaitu $400 - (80 + 2 + 1) = 317$, yang berarti seluruh data dari kelas lada matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada belum matang.

Pada kelas lada lewat matang, model mampu mengenali sebanyak 76 data dengan benar sehingga diperoleh *True Positive* (TP) = 76. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai lada lewat matang sehingga *False Positive* (FP) = 0. Terdapat 1 data lada lewat matang yang salah diprediksi sebagai lada setengah matang sehingga *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari jumlah data uji 400, yaitu $400 - (76 + 0 + 1) = 323$, yang berarti

seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan bukan lada yang tidak salah diprediksi sebagai lada lewat matang.

Pada kelas lada matang, model berhasil mengenali sebanyak 81 data dengan benar sehingga diperoleh *True Positive* (TP) = 81. Terdapat 3 data dari kelas lada setengah matang yang salah diprediksi sebagai lada matang sehingga *False Positive* (FP) = 3. Terdapat 1 data lada matang yang salah diprediksi sebagai lada setengah matang sehingga *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari jumlah data uji 400, yaitu $400 - (81 + 3 + 1) = 315$, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada matang.

Pada kelas lada setengah matang, model mampu mengenali sebanyak 75 data dengan benar sehingga diperoleh *True Positive* (TP) = 75. Terdapat 1 data dari kelas lada belum matang, 1 data dari kelas lada lewat matang, dan 1 data dari kelas lada matang yang salah diprediksi sebagai lada setengah matang sehingga *False Positive* (FP) = $1 + 1 + 1 = 3$. Terdapat 2 data lada setengah matang yang salah diprediksi sebagai lada belum matang dan 3 data lagi salah diprediksi sebagai lada matang sehingga *False Negative* (FN) = $2 + 3 = 5$. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (75 + 3 + 5) = 317$, yang berarti seluruh data dari kelas lada belum matang, lada matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada setengah matang.

Pada kelas objek bukan lada, model mampu mengenali seluruh 80 data dengan benar sehingga diperoleh *True Positive* (TP) = 80. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai objek bukan lada sehingga *False Positive* (FP) = 0. Seluruh data objek bukan lada juga tidak ada yang salah diprediksi ke kelas lain sehingga *False Negative* (FN) = 0. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (80 + 0 + 0) = 320$, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan lada lewat matang yang tidak salah diprediksi sebagai bukan lada.

Dari *confusion matrix* tersebut diperoleh informasi mengenai jumlah prediksi benar dan salah pada masing-masing kelas. Nilai-nilai ini kemudian digunakan untuk menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan

F1-score. Rincian hasil klasifikasi report dan perhitungan manual untuk model XGBoost disajikan berikut:

Evaluasi XGBoost:

Akurasi : 0.9800

Presisi : 0.9803

Recall : 0.9800

F1-Score: 0.9801

Laporan Klasifikasi:

	precision	recall	f1-score	support
lada belum matang	0.98	0.99	0.98	81
lada lewat matang	1.00	0.99	0.99	77
lada matang	0.96	0.99	0.98	82
lada setengah matang	0.96	0.94	0.95	80
objek bukan lada	1.00	1.00	1.00	80
accuracy			0.98	400
macro avg	0.98	0.98	0.98	400
weighted avg	0.98	0.98	0.98	400

Evaluasi XGBoost menunjukkan akurasi 98,00% dengan presisi 98,03%, *recall* 98,00%, dan *F1-score* 98,01%. Hasil ini membuktikan model bekerja sangat baik dalam mengklasifikasikan tingkat kematangan lada. Laporan klasifikasi memperlihatkan kelas “objek bukan lada” memiliki nilai sempurna, sementara “lada setengah matang” sedikit lebih rendah dengan presisi 0,96 dan recall 0,94. Kolom *support* menunjukkan jumlah data uji pada tiap kelas, dengan total 400 data uji. Nilai *macro average* adalah rata-rata kinerja tiap kelas tanpa memperhitungkan jumlah data, sedangkan *weighted average* mempertimbangkan distribusi data tiap kelas sehingga lebih merepresentasikan kinerja keseluruhan model.

Dari *confusion matrix* tersebut didapatkan informasi untuk masing-masing kelas, seperti pada Tabel 13.

Tabel 13. Klasifikasi Report XGBoost

Kelas	TP	FP	FN	TN
Lada Belum Matang	80	2	1	317
Lada Lewat Matang	76	0	1	323
Lada Matang	81	3	1	315
Lada Setengah Matang	75	3	5	317
Objek Bukan Lada	80	0	0	320

Kemudian dari tabel ini ditentukanlah jumlah akurasi, presisi, recall dan f1-score, berikut adalah perhitungan manualnya:

a. Akurasi

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{80+76+81+75+80}{400} = \frac{392}{400} = 0.9800$$

Jadi akurasi model XGBoost adalah 98%

b. Presisi

$$1) \text{ Precision (belum matang) } = \frac{TP}{TP+FP} = \frac{80}{80+2} = 0.9756$$

$$2) \text{ Precision (lewat matang) } = \frac{TP}{TP+FP} = \frac{76}{76+0} = 1.000$$

$$3) \text{ Precision (matang) } = \frac{TP}{TP+FP} = \frac{81}{81+3} = 0.9643$$

$$4) \text{ Precision (setengah matang) } = \frac{TP}{TP+FP} = \frac{75}{75+3} = 0.9615$$

$$5) \text{ Precision (Objek bukan lada) } = \frac{TP}{TP+FP} = \frac{80}{80+0} = 1.000$$

Rata rata presisi:

$$\frac{0.9756+1.000+0.9643+1.000}{5} = 0.9803$$

Jadi, di dapatkan presisi 98,03%

c. Recall

$$1) \text{ Recall (belum matang) } = \frac{TP}{TP+FN} = \frac{80}{80+1} = 0.9877$$

$$2) \text{ Recall (lewat matang) } = \frac{TP}{TP+FN} = \frac{76}{76+1} = 0.9870$$

$$3) \text{ Recall (matang) } = \frac{TP}{TP+FN} = \frac{81}{81+1} = 0.9878$$

$$4) \text{ Recall (setengah matang) } = \frac{TP}{TP+FN} = \frac{75}{75+5} = 0.9375$$

$$5) \text{ Recall (objek bukan lada) } = \frac{TP}{TP+FN} = \frac{80}{80+0} = 1.000$$

Rata rata recall:

$$\frac{0.9877+0.9625+0.9877+0.9375+1.0000}{5} = 0.9802$$

Jadi di dapatkan recall sebesar 98,02%

e) F1-score

$$1) \text{ F1 - Score (belum matang) } = 2 \times \frac{0.9756 \times 0.9877}{(0.9756 + 0.9877)} = 0.9816$$

$$2) \text{ F1 - Score (lewat matang) } = 2 \times \frac{(1.0000 \times 0.9870)}{(1.0000 + 0.9870)} = 0.9935$$

$$3) \text{ F1 - Score (matang) } = 2 \times \frac{(0.9643 \times 0.9878)}{(0.9643 + 0.9878)} = 0.9760$$

$$4) \text{ F1 - Score (setengah matang) } = 2 \times \frac{(0.9615 \times 0.9375)}{(0.9615 + 0.9375)} = 0.9493$$

$$5) \text{ F1 - Score (objek bukan lada) } = 2 \times \frac{(1.0000 \times 1.0000)}{(1.0000 + 1.0000)} = 1.0000$$

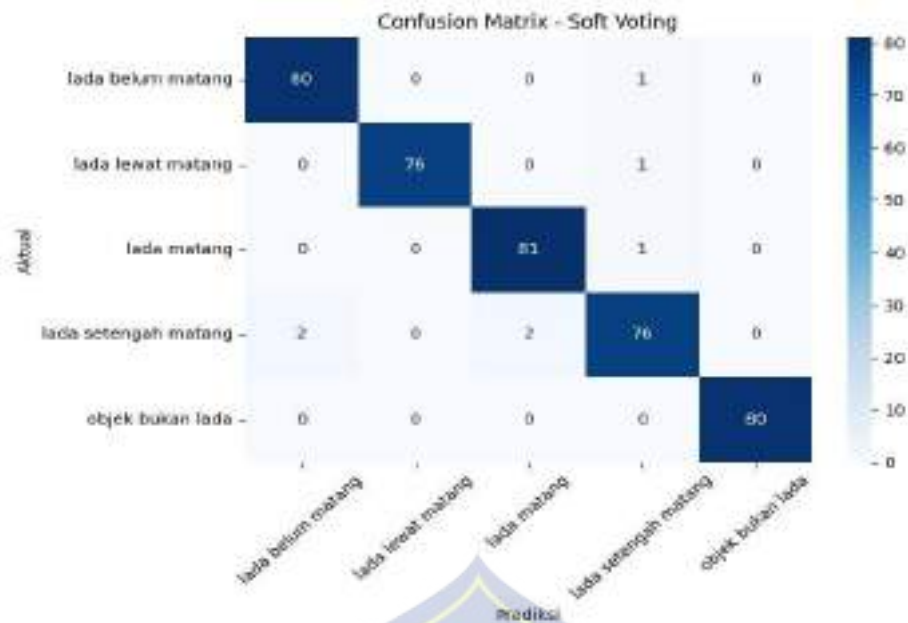
Rata rata f1-score:

$$\frac{0.9816+0.9933+0.9760+0.9493+1.0000}{5} = 0.9801$$

Jadi di dapatkan F1-score sebesar 98,01%

3. Ensemble Soft Voting

Model *Soft Voting* untuk klasifikasi tingkat kematangan lada telah diterapkan dan dievaluasi menggunakan data uji. Evaluasi ini dilakukan untuk mengukur kemampuan model dalam mengidentifikasi masing-masing kelas secara akurat. Kinerja model ditampilkan secara visual melalui *confusion matrix* dan dilengkapi dengan metrik evaluasi, seperti akurasi, presisi, *recall*, dan *F1-score*, guna memberikan gambaran menyeluruh terhadap efektivitas klasifikasi yang dilakukan. Hasil evaluasi performa model dapat diamati secara langsung melalui *confusion matrix* yang ditampilkan pada Gambar 16.



Gambar 16. Confusion Matriks Ensemble Soft Voting

Evaluasi kinerja model pada *Ensemble Soft Voting* ditunjukkan melalui *confusion matrix* dengan empat komponen, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN).

Berdasarkan hasil pengujian pada kelas lada belum matang, model *Soft Voting Ensemble* mampu mengenali sebanyak 80 data lada belum matang secara benar sehingga diperoleh nilai *True Positive* (TP) = 80. Terdapat 2 data dari kelas lada setengah matang yang salah diprediksi sebagai lada belum matang sehingga menghasilkan *False Positive* (FP) = 2. Terdapat 1 data lada belum matang yang salah diprediksi sebagai lada setengah matang sehingga diperoleh *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (80 + 2 + 1) = 317$, yang berarti seluruh data dari kelas lada matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada belum matang.

Pada kelas lada lewat matang, model mampu mengenali sebanyak 76 data secara benar sehingga diperoleh *True Positive* (TP) = 76. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai lada lewat matang sehingga nilai *False Positive* (FP) = 0. Terdapat 1 data lada lewat matang yang salah diprediksi sebagai lada setengah matang sehingga diperoleh *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (76 + 0 + 1) = 323$,

yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan bukan lada yang tidak salah diprediksi sebagai lada lewat matang.

Pada kelas lada matang, model berhasil mengenali sebanyak 81 data dengan benar sehingga diperoleh *True Positive* (TP) = 81. Terdapat 2 data dari kelas lada setengah matang yang salah diprediksi sebagai lada matang sehingga diperoleh *False Positive* (FP) = 2. Terdapat 1 data lada matang yang salah diprediksi sebagai lada setengah matang sehingga *False Negative* (FN) = 1. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (81 + 2 + 1) = 316$, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada matang.

Pada kelas lada setengah matang, model mampu mengenali sebanyak 76 data dengan benar, sehingga diperoleh *True Positive* (TP) = 76. Terdapat 1 data dari kelas lada belum matang, 1 data dari kelas lada lewat matang, dan 1 data dari kelas lada matang yang salah diprediksi sebagai lada setengah matang, sehingga *False Positive* (FP) = $1 + 1 + 1 = 3$. Terdapat 2 data lada setengah matang yang salah diprediksi sebagai lada belum matang dan 2 data lainnya salah diprediksi sebagai lada matang sehingga total *False Negative* (FN) = $2 + 2 = 4$. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (76 + 3 + 4) = 317$, yang berarti seluruh data dari kelas lada belum matang, lada matang, lada lewat matang, dan bukan lada yang tidak salah diprediksi sebagai lada setengah matang.

Pada kelas objek bukan lada, model mampu mengenali seluruh 80 data dengan benar sehingga diperoleh *True Positive* (TP) = 80. Tidak terdapat data dari kelas lain yang salah diprediksi sebagai objek bukan lada sehingga *False Positive* (FP) = 0. Seluruh data objek bukan lada juga tidak ada yang salah diprediksi ke kelas lain sehingga *False Negative* (FN) = 0. Nilai *True Negative* (TN) dihitung dari total data uji 400, yaitu $400 - (80 + 0 + 0) = 320$, yang berarti seluruh data dari kelas lada belum matang, lada setengah matang, lada matang, dan lada lewat matang yang tidak salah diprediksi sebagai bukan lada.

Melalui *confusion matrix*, dapat diketahui jumlah klasifikasi yang dilakukan dengan benar maupun yang mengalami kesalahan pada setiap kategori. Data ini

menjadi acuan dalam menghitung berbagai metrik evaluasi, seperti akurasi, presisi, *recall*, dan *F1-score*.

Dari hasil visualisasi confusion matrix untuk data test menggunakan model *Ensemble Soft Voting* kita dapat melihat perolehan akurasi pada dataset. Berikut adalah klasifikasi report dan perhitungan manual nilai akurasi presisi, *recall* dan *f1-score*.

Evaluasi Soft Voting:

Akurasi : 0.9825
 Presisi : 0.9827
 Recall : 0.9825
 F1 Score : 0.9826

Laporan Klasifikasi:

	precision	recall	f1-score	support
lada belum matang	0.98	0.99	0.98	81
lada lewat matang	1.00	0.99	0.99	77
lada matang	0.98	0.99	0.98	82
lada setengah matang	0.96	0.95	0.96	80
objek bukan lada	1.00	1.00	1.00	80
accuracy			0.98	400
macro avg	0.98	0.98	0.98	400
weighted avg	0.98	0.98	0.98	400

Evaluasi menggunakan metode *Soft Voting* menunjukkan performa klasifikasi yang sangat baik, dengan akurasi sebesar 98,25 persen, presisi 98,27 persen, recall 98,25 persen, dan F1-score 98,26 persen.

Akurasi merupakan ukuran keseluruhan dari kemampuan model dalam memprediksi kelas yang benar dari seluruh data. Presisi menunjukkan ketepatan model dalam memprediksi suatu kelas, yaitu seberapa banyak prediksi positif yang benar dibandingkan seluruh prediksi positif, sedangkan *recall* menunjukkan kemampuan model dalam menangkap semua data aktual dari kelas tertentu. *F1-score* adalah rata-rata harmonis dari presisi dan recall yang menilai keseimbangan antara ketepatan dan kemampuan deteksi model. Hasil ini menunjukkan bahwa

model mampu mengklasifikasikan data dengan tingkat ketepatan yang tinggi sehingga prediksi terhadap masing-masing kelas relatif konsisten dan dapat diandalkan. Kolom *support* menunjukkan jumlah data uji pada tiap kelas, dengan total 400 sampel. Selain itu, *macro average* dihitung sebagai rata-rata sederhana dari metrik tiap kelas tanpa mempertimbangkan jumlah data, sedangkan *weighted average* mempertimbangkan proporsi data tiap kelas sehingga lebih merepresentasikan kinerja keseluruhan model.

Dari *confusion matrix* yang dapat dilihat pada Gambar 16 didapatkan informasi untuk masing-masing kelas, seperti pada Tabel 14 berikut.

Tabel 14. Klasifikasi Report Ensemble Soft Voting

Kelas	TP	FP	FN	TN
Lada Belum Matang	80	2	1	317
Lada Lewat Matang	76	0	1	323
Lada Matang	81	2	1	316
Lada Setengah Matang	76	3	4	317
Objek Bukan Lada	80	0	0	320

Kemudian dari tabel ini ditentukanlah akurasi, presisi, recall dan f1-score, berikut adalah hitungan manualnya:

a. Akurasi

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{80+76+81+76+80}{400} = \frac{393}{400} = 0.9825$$

Jadi akurasi model soft voting ensemble adalah 98.25%

b. Presisi

$$1) \text{ Precision (belum matang) } = \frac{TP}{TP+FP} = \frac{80}{80+2} = 0.9756$$

$$2) \text{ Precision (lewat matang) } = \frac{TP}{TP+FP} = \frac{76}{76+0} = 1.000$$

$$3) \text{ Precision (matang) } = \frac{TP}{TP+FP} = \frac{81}{81+2} = 0.9759$$

$$4) \text{ Precision (setengah matang) } = \frac{TP}{TP+FP} = \frac{76}{76+3} = 0.9620$$

$$5) \text{ Precision (Objek bukan lada) } = \frac{TP}{TP+FP} = \frac{80}{80+0} = 1.000$$

Rata rata presisi:

$$\frac{0.9756+1.000+0.9759+0.9620+1.000}{5} = 0.9827$$

Jadi, di dapatkan presisi 98,27%

c. Recall

$$1) \text{ Recall (belum matang)} = \frac{TP}{TP+FN} = \frac{80}{80+1} = 0.9877$$

$$2) \text{ Recall (lewat matang)} = \frac{TP}{TP+FN} = \frac{76}{76+1} = 0.9870$$

$$3) \text{ Recall (matang)} = \frac{TP}{TP+FN} = \frac{81}{81+1} = 0.9878$$

$$4) \text{ Recall (setengah matang)} = \frac{TP}{TP+FN} = \frac{76}{76+4} = 0.9500$$

$$5) \text{ Recall (objek bukan lada)} = \frac{TP}{TP+FN} = \frac{80}{80+0} = 1.000$$

Rata rata recall:

$$\frac{0.9877+0.9870+0.9878+0.9500+1.0000}{5} = 0.9825$$

Jadi di dapatkan recall sebesar 98,25%

d. F1-score

$$1) \text{ F1 - Score (belum matang)} = 2 \times \frac{0.9756 \times 0.9877}{(0.9756+0.9877)} = 0.9816$$

$$2) \text{ F1 - Score (lewat matang)} = 2 \times \frac{(1.0000 \times 0.9870)}{(1.0000 + 0.9870)} = 0.9935$$

$$3) \text{ F1 - Score (matang)} = 2 \times \frac{(0.9759 \times 0.9878)}{(0.9759 + 0.9878)} = 0.9818$$

$$4) \text{ F1 - Score (setengah matang)} = 2 \times \frac{(0.9620 \times 0.9500)}{(0.9620 + 0.9500)} = 0.9560$$

$$5) \text{ F1 - Score (objek bukan lada)} = 2 \times \frac{(1.0000 \times 1.0000)}{(1.0000 + 1.0000)} = 1.0000$$

Rata rata f1-score:

$$\frac{0.9816+0.9935+0.9818+0.9560+1.0000}{5} = 0.9826$$

Jadi di dapatkan F1-score sebesar 98,26%

Untuk mengevaluasi performa model secara lebih mendetail, dilakukan analisis akurasi per kelas terhadap hasil prediksi model *Soft Voting*. Evaluasi ini bertujuan untuk mengidentifikasi sejauh mana model mampu mengenali masing-masing kategori kematangan lada dan membedakan antara objek lada. Tabel

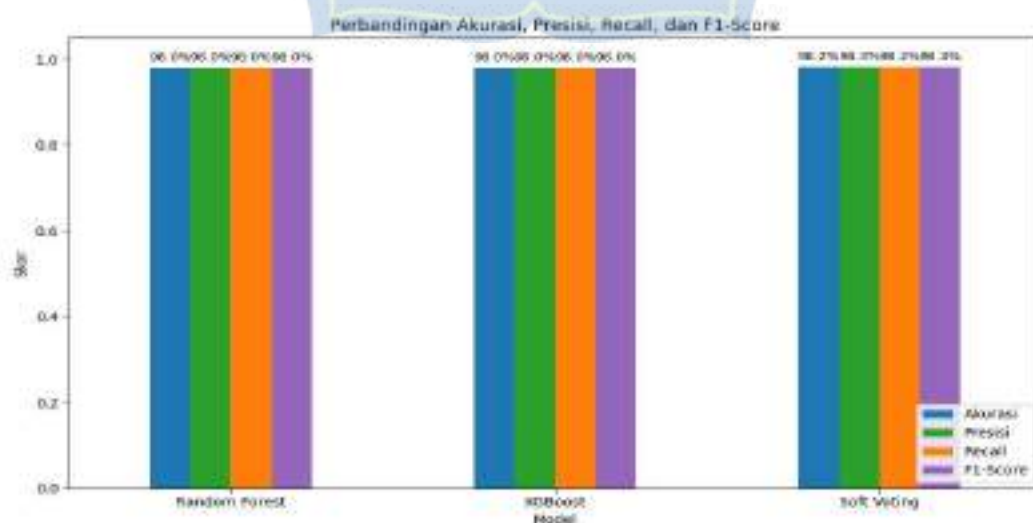
berikut menyajikan jumlah data uji, jumlah prediksi yang benar, serta persentase akurasi untuk setiap kelas. Hasil akurasi tiap kelas dapat dilihat pada tabel 15.

Tabel 15. Akurasi Per Kelas

Kelas	Data Uji	Prediksi Benar	Akurasi (%)
Lada Belum Matang	81	80	98.76
Lada Lewat Matang	77	76	98.70
Lada Matang	82	81	98.78
Lada Setengah Matang	80	76	95.00
Objek Bukan Lada	80	80	100.00

Berdasarkan hasil pada Tabel 15 dapat dilihat bahwa model mampu mengklasifikasikan setiap kelas dengan tingkat akurasi yang sangat tinggi, yaitu di atas 95%. Akurasi tertinggi dicapai pada kelas Objek Bukan Lada dengan nilai 100%, sedangkan akurasi terendah terdapat pada kelas Lada Setengah Matang sebesar 95%. Hal ini menunjukkan bahwa model memiliki performa yang konsisten dan andal dalam mengenali berbagai tingkat kematangan lada, meskipun terdapat sedikit kesalahan pada kelas yang memiliki kemiripan visual.

Selanjutnya dapat dilihat pada diagram hasil evaluasi pada data test masing-masing model seperti akurasi, presisi, recall dan f1-score yang ditunjukkan pada Gambar 17.



Gambar 17. Performa Masing-Masing Model

Gambar 17 menampilkan perbandingan hasil pengujian tiga model, yaitu *Random Forest*, *XGBoost*, dan *Soft Voting*. Perbandingan ini ditunjukkan berdasarkan empat metrik evaluasi, yaitu akurasi yang ditandai dengan warna biru, presisi dengan warna hijau, *recall* dengan warna oranye, dan *F1-score* dengan warna ungu. Model *Random Forest* dan *XGBoost* menghasilkan nilai yang sama, yaitu 98 persen pada semua metrik evaluasi.

Sementara itu, model *Soft Voting Ensemble* menunjukkan performa yang sedikit lebih tinggi, dengan akurasi 98,25 persen, presisi 98,3 persen, *recall* 98,2 persen, dan *F1-score* 98,3 persen. Meskipun perbedaan nilai numerik tampak kecil, *Soft Voting* lebih unggul karena memanfaatkan kombinasi prediksi dari kedua model dasar. Dengan cara ini, hasil prediksi menjadi lebih stabil, konsisten, dan lebih adaptif terhadap data baru dibandingkan penggunaan satu model tunggal.

Pendekatan *ensemble Soft Voting* tidak hanya meningkatkan akurasi, tetapi juga meningkatkan keandalan sistem klasifikasi tingkat kematangan lada. Dengan mengintegrasikan probabilitas prediksi dari *Random Forest* dan *XGBoost*, *Soft Voting* mampu mereduksi bias dan varians yang mungkin muncul pada model tunggal. Oleh karena itu, meskipun peningkatan akurasi terlihat minimal, keunggulan utama pendekatan ensemble terletak pada kekuatan kolaboratif antar model yang membuat hasil prediksi lebih tahan terhadap variasi data dan noise.

H. Evaluasi kinerja

Evaluasi kinerja dilakukan untuk mengukur seberapa baik model dalam mengklasifikasikan tingkat kematangan lada berdasarkan fitur warna HSV. Tiga model yang diuji dalam penelitian ini adalah *Random Forest*, *XGBoost*, dan *Ensemble Soft Voting*. Evaluasi dilakukan berdasarkan beberapa metrik utama, yaitu akurasi, presisi, *recall*, dan *F1-score*, dengan hasil perhitungan ditampilkan pada Tabel 16.

Tabel 16. Perbandingan Masing-Masing Model

Model	Akurasi	Presisi	Recall	F1-Score
Random Forest	98%	98,04%	98,02%	98,01%
XGBoost	98%	98,03%	98%	98,01%
Ensemble Soft Voting	98,25%	98,27%	98,35%	98,26%

Dari Tabel 16 *Ensemble Soft Voting* mencatat skor tertinggi pada seluruh metrik evaluasi, termasuk akurasi, presisi, *recall*, dan *F1-score*. Kombinasi probabilistik dari dua algoritma dasar menghasilkan prediksi yang lebih stabil, akurat, dan tahan terhadap variasi data dibandingkan model individual. Evaluasi per kelas menunjukkan hasil yang konsisten, dengan seluruh kelas memiliki akurasi di atas 98%, dan kelas non-lada terklasifikasi dengan akurasi sempurna 100%. Hal ini menunjukkan bahwa pemilihan fitur yang tepat dan metode klasifikasi yang sesuai mampu menghasilkan model yang ringan namun tetap andal untuk tugas pengenalan citra. Pemilihan fitur yang tepat dan metode klasifikasi yang sesuai mampu menghasilkan model yang ringan namun tetap andal untuk tugas pengenalan citra. Pendekatan *ensemble*, khususnya metode *Soft Voting*, memberikan keunggulan tambahan dengan menggabungkan kekuatan beberapa model dasar dalam hal ini *Random Forest* dan *XGBoost* untuk menghasilkan prediksi yang lebih stabil dan akurat. Dengan mengintegrasikan prediksi melalui rata-rata probabilitas, model ensemble mampu mereduksi kesalahan individual dari masing-masing algoritma dan meningkatkan generalisasi terhadap data baru. Hasil penelitian ini menunjukkan bahwa teknik *ensemble* bukan hanya adaptif, tetapi juga unggul dalam konteks klasifikasi citra berbasis warna seperti pada buah lada.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah penulis lakukan dan hasil penelitian yang telah dijabarkan pada bab sebelumnya, berikut merupakan kesimpulan yang diperoleh dari penelitian ini:

1. Berdasarkan hasil penelitian, model Random Forest dan XGBoost berhasil digunakan untuk mengklasifikasikan tingkat kematangan buah lada berdasarkan fitur warna kulit. Citra buah lada didokumentasikan menggunakan kamera handphone Vivo V40 dengan resolusi 50MP. Ekstraksi fitur dilakukan pada ruang warna HSV. Setelah melalui tahapan pra-pemrosesan, kedua model mampu mengenali pola visual dengan baik sehingga menunjukkan efektivitas pendekatan berbasis warna.
2. Penulis melakukan penerapan metode Ensemble Learning (Soft Voting). Metode ini diterapkan dengan menggabungkan dua model dasar melalui pendekatan Soft Voting Ensemble. Hasil evaluasi menunjukkan bahwa model ensemble ini memberikan performansi terbaik daripada model tunggal, dengan akurasi sebesar 98,25%, presisi 98,27%, recall 98,35%, dan f1-score 98,26%. Meskipun peningkatan akurasi dibandingkan model tunggal relatif kecil, metode ensemble terbukti lebih stabil dan konsisten di seluruh kelas. Hal ini membuktikan bahwa ensemble merupakan pendekatan yang efektif untuk klasifikasi citra berbasis warna yang kompleks seperti kulit buah lada.

B. Saran

Berdasarkan hasil penelitian yang telah dilakukan, penulis menyadari masih terdapat beberapa aspek yang dapat ditingkatkan untuk penelitian selanjutnya agar hasil klasifikasi menjadi lebih optimal. Adapun saran-saran yang dapat dipertimbangkan adalah sebagai berikut:

1. Mempertimbangkan penggunaan arsitektur deep learning, seperti Convolutional Neural Network (CNN), untuk melakukan ekstraksi fitur secara otomatis dari citra mentah tanpa ketergantungan pada fitur warna yang eksplisit.
2. Mengembangkan sistem aplikasi real-time, sehingga pengguna dapat langsung memindai citra lada dan memperoleh hasil klasifikasi secara cepat dan praktis dalam berbagai kondisi di lapangan.



DAFTAR PUSTAKA

- Aeni, S. N. (2022). *Budidaya Lada Agar Panen Melimpah*. Kompas.Com.
<https://agri.kompas.com/read/2022/12/13/095931484/5-tahapan-budidaya-lada-agar-panennya-melimpah?page=all>
- Agri Edu. (2023). *Tanaman Lada, Rempah Lokal Yang Mendunia*. Gokomodo.
<https://gokomodo.com/blog/tanaman-lada-rempah-lokal-yang-mendunia>
- Ali Buchari, M., Haritami, K., & Passarella, R. (2022). *Implementasi Metode Hue Saturation Value Pada Identifikasi Citra Warna Kulit Studi Kasus: Pembuatan Surat Keterangan Catatan Kepolisian*.
- Alves, F. S., Cruz, J. N., de Farias Ramos, I. N., do Nascimento Brandão, D. L., Queiroz, R. N., da Silva, G. V., da Silva, G. V., Dolabela, M. F., da Costa, M. L., Khayat, A. S., de Arimatéia Rodrigues do Rego, J., & do Socorro Barros Brasil, D. (2023). Evaluation of Antimicrobial Activity and Cytotoxicity Effects of Extracts of *Piper nigrum* L. and Piperine. *Separations*, 10(1), 1–18.
<https://doi.org/10.3390/separations10010021>
- Amini, S., Saber, M., Rabiei-Dastjerdi, H., & Homayouni, S. (2022). Urban Land Use and Land Cover Change Analysis Using Random Forest Classification of Landsat Time Series. *Remote Sensing*, 14(11), 1–23.
<https://doi.org/10.3390/rs14112654>
- Amrozi, M. A., Figo SW, D., & Wahyusari, R. (2024). Perbandingan Segmentasi Ruang Warna HSV dan YCbCr untuk Deteksi Objek. *Infomatek*, 26(2), 217–222. <https://doi.org/10.23969/infomatek.v26i2.19025>
- Angriani, P. (2023). *KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI. (Doctoral Dissertation, Universitas Hasanuddin)*.
- Anjani, N. C., Kusmiadi, R., & Inonu, I. (2023). Hubungan Penerapan GAP dan GHP dengan Mutu Lada Putih (*Piper nigrum* L.) di Kabupaten Bangka. *Agrikultura*, 34(3), 463. <https://doi.org/10.24198/agrikultura.v34i3.43151>

- Argadianata, A. P., Fatah, D. A., Sukri, H., Informasi, S., Madura, U. T., Kamal, K., Timur, J., & Smith, G. (2025). *Klasifikasi Kualitas Buah Apel Menggunakan Metode Random Forest*. 9(2), 2016–2022.
- Atika, N. (2023). *PENGARUH CARA PENGOLAHAN MERICA TERHADAP PENDAPATAN PETANI DI DESA ERA BARU KECAMATAN TELLULIMPOE KABUPATEN SINJAI*. VIII(I), 1–19.
- Bangun, P., Nurhayati, & Sihombing, M. (2021). PENGOLAHAN CITRA UNTUK IDENTIFIKASI KEMATANGAN BUAH JERUK DENGAN MENGGUNAKAN METODE BACKPROPAGATION BERDASARKAN NILAI HSV. *Jurnal Teknik Informatika Kaputama (JTik)*, 5(1).
- Benlachmi, Y., El Airej, A., & Hasnaoui, M. L. (2022). Fruits disease classification using machine learning techniques. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, 10(4), 917–929.
- Bey Lirna, C. C., Trimono, T., & Damaliana, A. T. (2024). Employee Voluntary Attrition Prediction At Pt.Xyz: Ensemble Machine Learning Approach With Soft Voting Classifier. *Jurnal Teknik Informatika (Jutif)*, 5(5), 1231–1239. <https://doi.org/10.52436/1.jutif.2024.5.5.2007>
- Cahyani, A. T. (2024). *KLASIFIKASI KUALITAS KELAPA MENGGUNAKAN ALGORITMA RANDOM FOREST, XGBOOST DAN DECISION TREE DENGAN EKSTRAKSI FITUR WARNA, TEKSTUR DAN BENTUK*.
- Candra, A. (2023). *Buah Lada – piper-nigrum-l – Agrokompleks Kita*. <https://agrokomplekskita.com/lada/buah-lada-piper-nigrum-l/>
- Din, M. S. U., Gohar, U. F., Hameed, U., Mukhtar, H., Morar, A., Herman, V., & Imre, K. (2022). Piper nigrum fruit extract as an antibiotic resistance reversal agent in MDR bacteria. *Applied Sciences*, 12(24), 12542.
- Elzandy, I., Puspaningrum, E. Y., Nurlaili, A. L., Raya, J., Madya, R., Anyar, G., & Timur, J. (2025). *KLASIFIKASI VARIETAS ANGGUR BERDASARKAN CITRA DAUN MENGGUNAKAN VGG16-XGBOOST*. 9(3), 5483–5491.

- Evizal, R. (2023). *BUKU PENGELOLAAN PERKEBUNAN LADA*.
- Fahmi, A., Hafidz, A., Puspaningrum, E. Y., & Nurlaili, A. L. (2025). *RUANG WARNA HSV UNTUK KLASIFIKASI REMPAH RIMPANG*. 9(3), 5492–5498.
- Hafizah, N., & Rizal, A. P. (2024). *Klasifikasi Kematangan Buah Pepaya Berdasarkan Fitur Warna Menggunakan Metode SVM*.
- Helmud, E., Helmud, E., Fitriyani, F., & Romadiana, P. (2024). Classification Comparison Performance of Supervised Machine Learning Random Forest and Decision Tree Algorithms Using Confusion Matrix. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 13(1), 92–97. <https://doi.org/10.32736/sisfokom.v13i1.1985>
- Hernando, E., Chandra Saputra, A., & Parhusip, J. (2024). *KLASIFIKASI TINGKAT KEMATANGAN BUAH NAGA BERDASARKAN WARNA KULIT MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS* (Vol. 2, Issue 2).
- Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, 10, 19083–19095. <https://doi.org/10.1109/ACCESS.2022.3151048>
- Hidayat, S., Achmadi, T. A., Ardhiansyah, H., Hidayat, H., Febriyanto, R., Abdulloh, A., & Ermawati, I. (2024). Optimalisasi Model Ensemble Learning dengan Augmentasi dan SMOTE pada Sistem Pendeteksi Kualitas Buah. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 6(1), 27–36. <https://doi.org/10.35746/jtim.v6i1.406>
- Istiani, A. (2024). *IMPLEMENTASI METODE RANDOM FOREST UNTUK KLASIFIKASI KATEGORI BERITA ONLINE*.
- Joses, S., Yulvida, D., & Rochimah, S. (2024). Pendekatan Metode Ensemble Learning untuk Prakiraan Cuaca menggunakan Soft Voting Classifier. *Journal of Applied Computer Science and Technology*, 5(1), 72–80. <https://doi.org/10.52158/jacost.v5i1.741>

- Kavzoglu, T., & Teke, A. (2022). Predictive Performances of Ensemble Machine Learning Algorithms in Landslide Susceptibility Mapping Using Random Forest, Extreme Gradient Boosting (XGBoost) and Natural Gradient Boosting (NGBoost). *Arabian Journal for Science and Engineering*, 47(6), 7367–7385. <https://doi.org/10.1007/s13369-022-06560-8>
- Kibria, H. B., Nahiduzzaman, M., Goni, M. O. F., Ahsan, M., & Haider, J. (2022). An Ensemble Approach for the Prediction of Diabetes Mellitus Using a Soft Voting Classifier with an Explainable AI. *Sensors*, 22(19). <https://doi.org/10.3390/s22197268>
- Kurniasari, A., Erwanto, D., & Rahayu, P. N. (2022). Ekstraksi Fitur Tekstur dan Warna pada Kulit Katak Menggunakan GLCM dan Momen Warna. *Jurnal ELTIKOM*, 6(1), 1–12. <https://doi.org/10.31961/eltikom.v6i1.287>
- Mardianto, R., Stefanie Quinevera, & Rochimah, S. (2024). Perbandingan Metode Random Forest, Convolutional Neural Network, dan Support Vector Machine Untuk Klasifikasi Jenis Mangga. *Journal of Applied Computer Science and Technology*, 5(1), 63–71. <https://doi.org/10.52158/jacost.v5i1.742>
- Maretva Cendani, L., & Wibowo, A. (2022). *Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes* (Vol. 13, Issue 1).
- Maulana Alfaruq, B., Erwanto, D., & Yanuartanti, I. (2023). Klasifikasi Kematangan Buah Tomat Dengan Metode Support Vector Machine. *Generation Journal*, 7(3), 64–72. <https://doi.org/10.29407/gj.v7i3.21092>
- Maulana, M. M. (2021). IMPLEMENTASI CITRA WARNA HSV UNTUK IDENTIFIKASI RHESUS GOLONGAN DARAH. *Pharmacognosy Magazine*, 75(17), 399–405.
- Melisa, S., Fauzi, A., & Yusfrizal. (2022). Penerapan Machine Learning Untuk Klasifikasi Tingkat Kematangan Buah Anggur (Vitis) Dengan Metode K-Nearest Neighbor. In *Media Online* (Vol. 1, Issue 5).
- Mienye, I. D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts,

- Algorithms, Applications, and Prospects. *IEEE Access*, 10(August), 99129–99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
- Natras, R., Soja, B., & Schmidt, M. (2022). Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting. *Remote Sensing*, 14(15), 1–34. <https://doi.org/10.3390/rs14153547>
- Nurulrachman, A. I., Wihandika, R. C., & ... (2023). Ekstraksi Ciri pada Klasifikasi Citra Batik menggunakan Metode Gray Level Co-Occurrence Matrix, Local Binary Pattern, dan HSV Color Moment. ... *Teknologi Informasi Dan ...*, 7(1), 374–383. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12197%0Ahttp://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/12197/5557>
- Pak Tani. (2021). *Lada Tanaman Tertua di Dunia - Lim Corporation*. <https://www.tokopertanian99.com/2021/06/mengenal-lada-hitam.html>
- Prasetyo, S. E., Prastyo, P. H., & Arti, S. (2021). A Cardiotocographic Classification using Feature Selection: A comparative Study. *JITCE (Journal of Information Technology and Computer Engineering)*, 5(01), 25–32. <https://doi.org/10.25077/jitce.5.01.25-32.2021>
- Pusat Data dan Informasi Pertanian. (2022). *OUTLOOK KOMODITAS LADA 2022 Pusat Data dan Sistem Informasi Pertanian i OUTLOOK LADA Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal-Kementerian Pertanian 2022*.
- Rachmi, A. N. (2020). *Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn*. 1–101.
- Rahman, J., & Koodoh, E. E. (2022). BERTANI LADA PADA MASYARAKAT DESA LAROMERUI KECAMATAN ROUTA KABUPATEN KONAWE PEPPER FARMING IN THE COMMUNITY LAROMERUI VILLAGE, ROUTA DISTRICT, KONAWE REGENCY. In *Jurnal Kerabat Antropologi*

(Vol. 6, Issue 1). <http://journal.fib.uho.ac.id/index.php/kabanti%7C39>

- Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, 5(2), 111–119. <https://doi.org/10.37148/bios.v5i2.128>
- Raysyah, S., Arinal, V., & Mulyana, D. I. (2021). KLASIFIKASI TINGKAT KEMATANGAN BUAH KOPI BERDASARKAN DETEKSI WARNA MENGGUNAKAN METODE KNN DAN PCA. *Sistem Informasi* |, 8(2), 88–95.
- Reinaldi, Ri. Y. (2021). Analisis Ekspor Lada Hitam Indonesia ke Negara Tujuan Utama. *Pharmacognosy Magazine*, 75(17), 399–405.
- Sunarko, B., Hasanah, U., Hidayat, S., Muhammad, N., Irfan Ardiansyah, M., Khikam Hakiki, M., Putra Ananda, B. D., & Baroroh, L. T. (2023). Penerapan Stacking Ensemble Learning untuk Klasifikasi Efek Kesehatan Akibat Pencemaran Udara. *Edu Komputika*, 10(1). <http://journal.unnes.ac.id/sju/index.php/edukom>
- Tri Laksono, A., Endryansyah, Wanarti Rusmamto, P., & Syariffuddien Zuhrie, M. (2022). Pengolahan Citra Digital Buah Murbei Dengan Algoritma LDA (Linear Discriminant Analysis). In *Indonesian Journal of Engineering and Technology* (Vol. 4, Issue 2). <https://journal.unesa.ac.id/index.php/inajet>
- Usman, Anis, M., Ibrahim, H., Riyani, M., & Rahman, A. (2024). MENELUSURI JEJAK SEJARAH DAN WARISAN KULTURAL KENEGERIAN LADA : PERGULATAN PERDAGANGAN REMPAH DI PESISIR TIMUR ACEH. *Jurnal Penelitian Dan Pengembangan Arkeologi*, 42(2), 137–152. <https://doi.org/10.55981/amt.2024.5515>
- Wu, X., Wu, X., Huang, H., Zhang, F., & Wen, Y. (2024). Characterization of Pepper Ripeness in the Field Using Hyperspectral Imaging (HSI) with Back

Propagation (BP) Neural Network and Kernel Based Extreme Learning Machine (KELM) Models. *Analytical Letters*, 57(3), 409–424.



Lampiran

Lampiran 1. Dataset Asli

a. Lada Belum Matang



b. Lada Setengah Matang





c. Lada Matang



d. Lada Lewat Matang



e. Objek Bukan Lada



Lampiran 2. Dataset Hasil Preprocessing

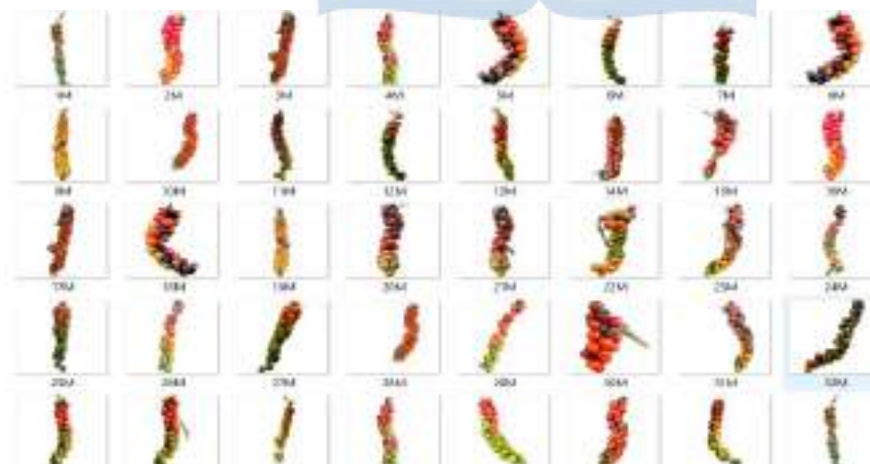
a. Lada Belum Matang



B. Lada Setengah Matang



c. Lada Matang



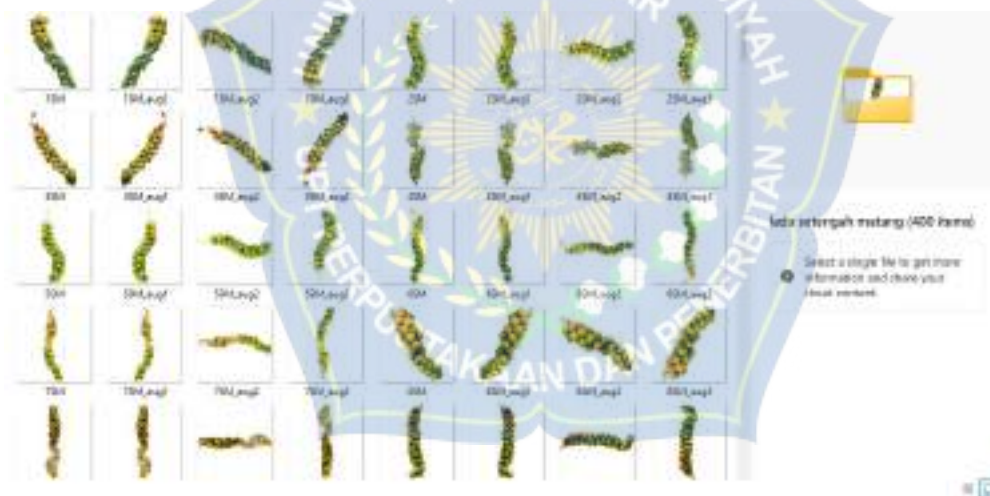
[illegible]

Lampiran 3. Dataset Hasil Augmentasi

a. lada Belum Matang



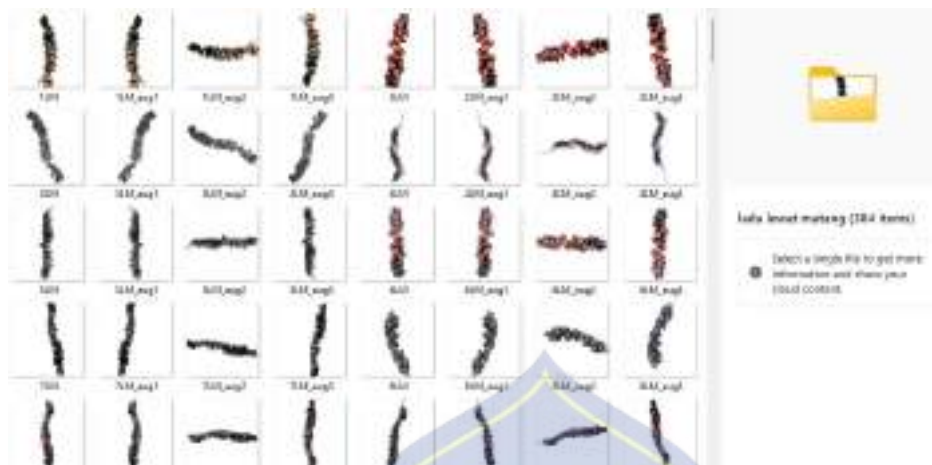
b. Lada Setengah Matang



c. Lada Matang



d. Lada Lewat Matang



e. Objek Bukan Lada



Lampiran 4. Source Code

a) Segmentasi_ROI.py

```
import os
import cv2
import numpy as np
from rembg import remove

def hapus_background(image_path):
    with open(image_path, 'rb') as f:
        input_data = f.read()
        output_data = remove(input_data)
        nparr = np.frombuffer(output_data, np.uint8)
        img = cv2.imdecode(nparr, cv2.IMREAD_UNCHANGED)
        if img is None:
            raise ValueError("Gagal memuat gambar")
        return img

def segmentasi_roi(img):
    if img.shape[2] == 4:
        alpha = img[:, :, 3]
        _, mask = cv2.threshold(alpha, 0, 255, cv2.THRESH_BINARY)
        kernel = np.ones((5, 5), np.uint8)
        closed_mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel,
            iterations=2)
        contours, _ = cv2.findContours(closed_mask, cv2.RETR_EXTERNAL,
            cv2.CHAIN_APPROX_SIMPLE)
        if len(contours) > 0:
            x_min, y_min, x_max, y_max = np.inf, np.inf, 0, 0
            for cnt in contours:
                x, y, w, h = cv2.boundingRect(cnt)
                x_min = min(x_min, x)
                y_min = min(y_min, y)
                x_max = max(x_max, x + w)
                y_max = max(y_max, y + h)
            img_cropped = img[int(y_min):int(y_max), int(x_min):int(x_max)]
        return img_cropped
    return img

def hilangkan_alpha_dan_background_putih(img_cropped):
    if img_cropped.shape[2] == 4:
        alpha_channel = img_cropped[:, :, 3]
        rgb_channels = img_cropped[:, :, :3]
        white_bg = np.ones_like(rgb_channels, dtype=np.uint8) * 255
        alpha_factor = alpha_channel[:, :, np.newaxis] / 255.0
        img_no_alpha = rgb_channels * alpha_factor + white_bg * (1 -
            alpha_factor)
        img_final = img_no_alpha.astype(np.uint8)
        return img_final
    return img_cropped

def resize_letterbox(img, output_size=(224, 224)):
    h0, w0 = img.shape[:2]
    scale = min(output_size[0]/h0, output_size[1]/w0)
    new_w, new_h = int(w0 * scale), int(h0 * scale)
```

```

img_resized = cv2.resize(img, (new_w, new_h))

result = np.ones((output_size[1], output_size[0], 3),
dtype=np.uint8) * 255
x_offset = (output_size[0] - new_w) // 2
y_offset = (output_size[1] - new_h) // 2
result[y_offset:y_offset+new_h, x_offset:x_offset+new_w] =
img_resized
return result

def proses_file(image_path, output_size=(224, 224)):
img_no_bg = hapus_background(image_path)
img_roi = segmentasi_roi(img_no_bg)
img_clean = hilangkan_alpha_dan_background_putih(img_roi)
img_final = resize_letterbox(img_clean, output_size)
return img_final

def proses_folder(input_folder, output_folder):
if not os.path.exists(output_folder):
os.makedirs(output_folder)

for kelas in os.listdir(input_folder):
kelas_path = os.path.join(input_folder, kelas)
if os.path.isdir(kelas_path):
save_kelas_path = os.path.join(output_folder, kelas)
os.makedirs(save_kelas_path, exist_ok=True)

for nama_file in os.listdir(kelas_path):
file_path = os.path.join(kelas_path, nama_file)
try:
hasil = proses_file(file_path)
save_path = os.path.join(save_kelas_path, nama_file)
cv2.imwrite(save_path, hasil)
print(f"☑ {file_path}")
except Exception as e:
print(f"✗ {file_path} - {e}")

if __name__ == '__main__':
proses_folder('dataset', 'hasil_ROI')

```

b) Augmentasi.py

```

import cv2
import numpy as np
import os

def rotasi_gambar(gambar, sudut):
tinggi, lebar = gambar.shape[:2]
matriks_rotasi = cv2.getRotationMatrix2D((lebar // 2, tinggi //
2), sudut, 1)
return cv2.warpAffine(gambar, matriks_rotasi, (lebar, tinggi),
borderValue=(255, 255, 255))

```

```

def augmentasi(gambar):
    hasil_augmentasi = []
    flip = cv2.flip(gambar, 1)
    hasil_augmentasi.append(flip)
    flip_rot90 = rotasi_gambar(flip, 90)
    hasil_augmentasi.append(flip_rot90)
    flip_rot180 = rotasi_gambar(flip, 180)
    hasil_augmentasi.append(flip_rot180)
    return hasil_augmentasi

def proses_augmentasi_folder(folder_input, folder_output):
    if not os.path.exists(folder_output):
        os.makedirs(folder_output)

    for nama_kelas in os.listdir(folder_input):
        path_kelas = os.path.join(folder_input, nama_kelas)
        if os.path.isdir(path_kelas):
            path_simpan_kelas = os.path.join(folder_output, nama_kelas)
            os.makedirs(path_simpan_kelas, exist_ok=True)

            for nama_file in os.listdir(path_kelas):
                path_file = os.path.join(path_kelas, nama_file)
                gambar = cv2.imread(path_file)
                if gambar is None:
                    print(f"⚠️ Gagal baca gambar: {path_file}")
                    continue
                # Simpan gambar asli
                cv2.imwrite(os.path.join(path_simpan_kelas, nama_file), gambar)

                # Simpan hasil augmentasi
                hasil_aug = augmentasi(gambar)
                for idx, gambar_aug in enumerate(hasil_aug):
                    nama_aug = f"{os.path.splitext(nama_file)[0]}_aug{idx+1}.jpg"
                    cv2.imwrite(os.path.join(path_simpan_kelas, nama_aug), gambar_aug)
                print(f"✅ {path_file} - Augmentasi selesai")

    if __name__ == '__main__':
        proses_augmentasi_folder('hasil_ROI', 'hasilAugmentasi')

```

c) KonversidanNormalisasi.py

```

import cv2
import numpy as np
import os
import pandas as pd

def konversi_rgb_ke_hsv(gambar):
    hsv = cv2.cvtColor(gambar, cv2.COLOR_BGR2HSV).astype('float32')
    hsv[:, :, 0] = hsv[:, :, 0] / 179.0 # Normalisasi H (0-1)
    hsv[:, :, 1] = hsv[:, :, 1] / 255.0 # Normalisasi S (0-1)
    hsv[:, :, 2] = hsv[:, :, 2] / 255.0 # Normalisasi V (0-1)
    return hsv

```

```

def proses_konversi_dan_simpan(folder_input, folder_output,
output_csv):
    if not os.path.exists(folder_output):
        os.makedirs(folder_output)

    data_list = []

    for nama_kelas in os.listdir(folder_input):
        path_kelas = os.path.join(folder_input, nama_kelas)
        if os.path.isdir(path_kelas):
            path_simpan_kelas = os.path.join(folder_output, nama_kelas)
            os.makedirs(path_simpan_kelas, exist_ok=True)

            for nama_file in os.listdir(path_kelas):
                path_file = os.path.join(path_kelas, nama_file)
                gambar = cv2.imread(path_file)
                if gambar is None:
                    print(f"✗ {path_file} - Gambar tidak dapat dibaca")
                    continue
                hsv_normal = konversi_rgb_ke_hsv(gambar)
                # Simpan ke .npy
                nama_simpan = os.path.splitext(nama_file)[0] + '.npy'
                path_simpan_file = os.path.join(path_simpan_kelas, nama_simpan)
                np.save(path_simpan_file, hsv_normal)
                # Hitung rata-rata per kanal (ambil rata-rata HSV normalisasi *
                100 dan bulatkan)
                mean_h = round(np.mean(hsv_normal[:, :, 0]) * 100)
                mean_s = round(np.mean(hsv_normal[:, :, 1]) * 100)
                mean_v = round(np.mean(hsv_normal[:, :, 2]) * 100)
                data_list.append([nama_file, nama_kelas, mean_h, mean_s, mean_v])
                print(f"☑ {path_file} - Konversi & Simpan selesai")

            # Simpan ke CSV
            if data_list:
                df = pd.DataFrame(data_list, columns=['filename', 'label', 'H',
                'S', 'V'])
                df.to_csv(output_csv, index=False)
                print(f" CSV selesai dibuat: {output_csv}")
            else:
                print("⚠ Tidak ada data yang diproses.")

    if __name__ == '__main__':
        proses_konversi_dan_simpan('hasilAugmentasi', 'hasilKonversiHSV',
        'output_hsv_normalisasi.csv')

```

d) ekstraksiHSV.py

```

import numpy as np
import os
import csv
from scipy.stats import skew

def ekstrak_color_moment(hsv_array):
    # Hitung Mean, Std Dev, dan Skewness untuk H, S, V

```

```

h_mean = np.mean(hsv_array[:, :, 0])
s_mean = np.mean(hsv_array[:, :, 1])
v_mean = np.mean(hsv_array[:, :, 2])

h_std = np.std(hsv_array[:, :, 0])
s_std = np.std(hsv_array[:, :, 1])
v_std = np.std(hsv_array[:, :, 2])

h_skew = skew(hsv_array[:, :, 0].flatten())
s_skew = skew(hsv_array[:, :, 1].flatten())
v_skew = skew(hsv_array[:, :, 2].flatten())

return h_mean, s_mean, v_mean, h_std, s_std, v_std, h_skew,
s_skew, v_skew

def proses_ekstraksi_folder(folder_input, file_output_csv):
    header = [
        'nama_file',
        'mean_h', 'mean_s', 'mean_v',
        'std_h', 'std_s', 'std_v',
        'skew_h', 'skew_s', 'skew_v',
        'label'
    ]
    data_fitur = []

    for nama_kelas in os.listdir(folder_input):
        path_kelas = os.path.join(folder_input, nama_kelas)
        if os.path.isdir(path_kelas):
            for nama_file in os.listdir(path_kelas):
                if not nama_file.endswith('.npy'):
                    continue

            path_file = os.path.join(path_kelas, nama_file)
            try:
                hsv_array = np.load(path_file)
                fitur = ekstrak_color_moment(hsv_array)
                data_fitur.append([nama_file, *fitur, nama_kelas])
            except Exception as e:
                print(f"❌ {path_file} - Error: {e}")

    # Simpan ke CSV
    with open(file_output_csv, mode='w', newline='') as file_csv:
        writer = csv.writer(file_csv)
        writer.writerow(header)
        writer.writerows(data_fitur)

    print(f"\n✅ Semua fitur Color Moment disimpan di {file_output_csv}")

if __name__ == '__main__':
    proses_ekstraksi_folder('hasilKonversiHSV',
        'dataset_HSVDanColor_moment.csv')

```

e) splitdata.py

```
import pandas as pd
from sklearn.model_selection import train_test_split

data = pd.read_csv('dataset_HSVDanColor_moment.csv')

data_latih, data_uji = train_test_split(data, test_size=0.2,
random_state=42, stratify=data['label'])

# 4 Simpan Data Latih & Uji
data_latih.to_csv('data_latih.csv', index=False)
data_uji.to_csv('data_uji.csv', index=False)
data.to_csv('data_keseluruhan.csv', index=False)

# 5 Tampilkan Jumlah Data
print(f"✅ Split Data Selesai")
print(f"Jumlah Data Keseluruhan: {len(data)}")
print(f"Jumlah Data Latih : {len(data_latih)}")
print(f"Jumlah Data Uji : {len(data_uji)}")
```

f) RFModel.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import RandomizedSearchCV,
StratifiedKFold, learning_curve
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
accuracy_score, precision_score, recall_score, f1_score,
classification_report, confusion_matrix
)

# Baca Data
data_latih = pd.read_csv('data_latih.csv')
data_uji = pd.read_csv('data_uji.csv')

# Pisahkan Fitur & Label
fitur = ['mean_h', 'mean_s', 'mean_v', 'std_h', 'std_s', 'std_v',
'skew_h', 'skew_s', 'skew_v']
X_latih = data_latih[fitur]
y_latih = data_latih['label']
X_uji = data_uji[fitur]
y_uji = data_uji['label']

# Standardisasi
scaler = StandardScaler()
X_latih_scaled = scaler.fit_transform(X_latih)
X_uji_scaled = scaler.transform(X_uji)
```

```

# Hyperparameter Tuning
param_dist = {
    'n_estimators': [100, 200, 300, 400],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']
}
rf = RandomForestClassifier(random_state=42, n_jobs=-1)
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

random_search = RandomizedSearchCV(
    rf, param_distributions=param_dist, n_iter=30,
    scoring='accuracy', n_jobs=-1, cv=cv, verbose=1, random_state=42
)
random_search.fit(X_latih_scaled, y_latih)
rf_terbaik = random_search.best_estimator_

# Prediksi & Probabilitas
y_pred = rf_terbaik.predict(X_uji_scaled)
proba = rf_terbaik.predict_proba(X_uji_scaled)

# Evaluasi
akurasi = accuracy_score(y_uji, y_pred)
presisi = precision_score(y_uji, y_pred, average='macro')
recall = recall_score(y_uji, y_pred, average='macro')
f1 = f1_score(y_uji, y_pred, average='macro')

print(" Parameter terbaik:", random_search.best_params_)
print(" Evaluasi Random Forest:")
print(f" Akurasi : {akurasi:.4f}")
print(f" Presisi : {presisi:.4f}")
print(f" Recall : {recall:.4f}")
print(f" F1-Score : {f1:.4f}")
print("\nLaporan Klasifikasi:\n", classification_report(y_uji,
y_pred))

# Simpan Output CSV
output = data_uji[['nama_file']].copy()
output['Label_Aslis'] = y_uji.reset_index(drop=True)
output['Label_Prediksi'] = y_pred
for idx, kelas in enumerate(rf_terbaik.classes_):
    output[f'Prob_{kelas}'] = proba[:, idx]
output.to_csv('rf_output_probabilitas.csv', index=False)

# Confusion Matrix
cm = confusion_matrix(y_uji, y_pred, labels=rf_terbaik.classes_)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=rf_terbaik.classes_, yticklabels=rf_terbaik.classes_)
plt.title('Confusion Matrix Random Forest')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.tight_layout()
plt.savefig('rf_confusion_matrix1.png')
plt.close()

```

```

# Learning Curve
train_sizes, train_scores, val_scores = learning_curve(
    rf_terbaik, X_latih_scaled, y_latih,
    cv=5, scoring='accuracy',
    train_sizes=np.linspace(0.1, 1.0, 10),
    n_jobs=-1
)
train_mean = train_scores.mean(axis=1)
val_mean = val_scores.mean(axis=1)

plt.figure(figsize=(10, 6))
plt.plot(train_sizes, train_mean, 'o-', label='Akurasi Data
Latih')
plt.plot(train_sizes, val_mean, 'o-', label='Akurasi Data Uji')
plt.title('Learning Curve Random Forest')
plt.xlabel('Jumlah Data Latih')
plt.ylabel('Akurasi')
plt.grid(True)
plt.legend(loc='best')
plt.tight_layout()
plt.savefig('rf_learning_curve.png')
plt.close()

```

g) XGBModel.py

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import RandomizedSearchCV,
StratifiedKFold, learning_curve
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    classification_report, confusion_matrix
)
from xgboost import XGBClassifier, plot_importance

# Baca Dataset
data_latih = pd.read_csv('data_latih.csv')
data_uji = pd.read_csv('data_uji.csv')

# Pisahkan Fitur & Label
fitur = ['mean_h', 'mean_s', 'mean_v', 'std_h', 'std_s', 'std_v',
'skew_h', 'skew_s', 'skew_v']
X_latih = data_latih[fitur]
y_latih = data_latih['label']
X_uji = data_uji[fitur]
y_uji = data_uji['label']

# Standarisasi Fitur
scaler = StandardScaler()
X_latih_scaled = scaler.fit_transform(X_latih)

```

```

X_uji_scaled = scaler.transform(X_uji)

# Encode Label
le = LabelEncoder()
y_latih_encoded = le.fit_transform(y_latih)
y_uji_encoded = le.transform(y_uji)

# Parameter & Model XGBoost
param_dist = {
    'n_estimators': [100, 200, 300, 400],
    'max_depth': [3, 5, 7, 10],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0]
}

xgb = XGBClassifier(random_state=42, n_jobs=-1,
                    use_label_encoder=False, eval_metric='mlogloss')
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

random_search = RandomizedSearchCV(
    xgb,
    param_distributions=param_dist,
    n_iter=30,
    scoring='accuracy',
    n_jobs=-1,
    cv=cv,
    verbose=1,
    random_state=42
)

# Training
random_search.fit(X_latih_scaled, y_latih_encoded)

# Prediksi & Decode
y_pred_encoded =
random_search.best_estimator_.predict(X_uji_scaled)
y_pred = le.inverse_transform(y_pred_encoded)

# Evaluasi
akurasi = accuracy_score(y_uji, y_pred)
presisi = precision_score(y_uji, y_pred, average='macro')
recall = recall_score(y_uji, y_pred, average='macro')
f1 = f1_score(y_uji, y_pred, average='macro')

print(f" Parameter Terbaik: {random_search.best_params_}")
print(" Evaluasi XGBoost:")
print(f" Akurasi : {akurasi:.4f}")
print(f" Presisi : {presisi:.4f}")
print(f" Recall : {recall:.4f}")
print(f" F1-Score : {f1:.4f}")
print("\nLaporan Klasifikasi:\n", classification_report(y_uji,
y_pred))

# Output Probabilitas & Prediksi ke CSV
probabilitas =
random_search.best_estimator_.predict_proba(X_uji_scaled)

```

```

kelas = le.classes_
df_prob = pd.DataFrame(probabilitas, columns=[f'Prob_{label}' for
label in kelas])

# kolom 'nama_file' dan 'Label_Aslis'
df_prob.insert(0, 'Label_Aslis', y_uji.values)
if 'nama_file' in data_uji.columns:
df_prob.insert(0, 'nama_file', data_uji['nama_file'])
else:
df_prob.insert(0, 'nama_file', [f"file_{i}" for i in
range(len(data_uji))])
df_prob['Label_Prediksi'] = y_pred

# Simpan ke CSV
df_prob.to_csv('xgb_output_probabilitas.csv', index=False)
print(" Output probabilitas dan prediksi disimpan sebagai
'xgb_output_probabilitas.csv'")

# Confusion Matrix Visualisasi
cm = confusion_matrix(y_uji, y_pred, labels=kelas)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=kelas, yticklabels=kelas)
plt.title('Confusion Matrix XGBoost')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.tight_layout()
plt.savefig('xgb_confusion_matrix.jpg')
plt.close()
print(" Confusion Matrix disimpan sebagai
'xgb_confusion_matrix.jpg'")

# Learning Curve
train_sizes, train_scores, valid_scores = learning_curve(
estimator=random_search.best_estimator_,
X=X_latih_scaled,
y=y_latih_encoded,
cv=cv,
scoring='accuracy',
n_jobs=-1,
train_sizes=np.linspace(0.1, 1.0, 10),
shuffle=True,
random_state=42
)

train_mean = np.mean(train_scores, axis=1)
valid_mean = np.mean(valid_scores, axis=1)

plt.figure(figsize=(8, 6))
plt.plot(train_sizes, train_mean, 'o-', color='blue',
label='Akurasi Data Latih')
plt.plot(train_sizes, valid_mean, 'o-', color='orange',
label='Akurasi Data Uji')
plt.title('Learning Curve XGBoost')
plt.xlabel('Jumlah Data Latih')
plt.ylabel('Akurasi')

```

```
plt.legend(loc='best')
plt.grid(True)
plt.tight_layout()
plt.savefig('xgb_learning_curve.jpg')
plt.close()
print(" Learning Curve disimpan sebagai 'xgb_learning_curve.jpg'")
```

h) Softvoting.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    classification_report, confusion_matrix
)
# Baca Output RF dan XGB
rf = pd.read_csv('rf_output_probabilitas.csv')
xgb = pd.read_csv('xgb_output_probabilitas.csv')

# Validasi File Sama
assert all(rf['nama_file'] == xgb['nama_file']), "✗ Nama file
tidak konsisten!"

# Ambil Label Asli dan Nama Kelas
output = rf[['nama_file', 'Label_Aslis']].copy()
kelas = [col.replace('Prob_', '') for col in rf.columns if
col.startswith('Prob_')]
y_true = rf['Label_Aslis']

# Prediksi Random Forest
rf_pred = rf[[f'Prob_{kls}' for kls in
kelas]].idxmax(axis=1).str.replace('Prob_', '')
acc_rf = accuracy_score(y_true, rf_pred)
pres_rf = precision_score(y_true, rf_pred, average='macro')
rec_rf = recall_score(y_true, rf_pred, average='macro')
f1_rf = f1_score(y_true, rf_pred, average='macro')

# Prediksi XGBoost
xgb_pred = xgb[[f'Prob_{kls}' for kls in
kelas]].idxmax(axis=1).str.replace('Prob_', '')
acc_xgb = accuracy_score(y_true, xgb_pred)
pres_xgb = precision_score(y_true, xgb_pred, average='macro')
rec_xgb = recall_score(y_true, xgb_pred, average='macro')
f1_xgb = f1_score(y_true, xgb_pred, average='macro')

# Ensemble Soft Voting: Rata-rata Probabilitas
for kls in kelas:
    output[f'Prob_{kls}'] = (rf[f'Prob_{kls}'] + xgb[f'Prob_{kls}']) /
    2
soft_pred = output[[f'Prob_{kls}' for kls in
kelas]].idxmax(axis=1).str.replace('Prob_', '')
output['Prediksi_SoftVoting'] = soft_pred
output.to_csv('soft_voting_output.csv', index=False)
```

```

# Evaluasi Soft Voting
acc_soft = accuracy_score(y_true, soft_pred)
pres_soft = precision_score(y_true, soft_pred, average='macro')
rec_soft = recall_score(y_true, soft_pred, average='macro')
f1_soft = f1_score(y_true, soft_pred, average='macro')
report = classification_report(y_true, soft_pred,
target_names=kelas)

print("\n Hasil Soft Voting disimpan di
'soft_voting_output.csv'\n")
print(" Evaluasi Soft Voting:")
print(f" Akurasi      : {acc_soft:.4f}")
print(f" Presisi      : {pres_soft:.4f}")
print(f" Recall       : {rec_soft:.4f}")
print(f" F1 Score      : {f1_soft:.4f}\n")
print("Laporan Klasifikasi:")
print(report)

# Confusion Matrix Soft Voting
plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_true, soft_pred, labels=kelas)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=kelas, yticklabels=kelas)
plt.title('Confusion Matrix - Soft Voting')
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('softvoting_conf_matrix.jpg')
plt.close()

# Perbandingan Akurasi, Presisi, Recall, F1
df_skor = pd.DataFrame({
'Model': ['Random Forest', 'XGBoost', 'Soft Voting'],
'Akurasi': [acc_rf, acc_xgb, acc_soft],
'Presisi': [pres_rf, pres_xgb, pres_soft],
'Recall': [rec_rf, rec_xgb, rec_soft],
'F1-Score': [f1_rf, f1_xgb, f1_soft]
})

df_skor_plot = df_skor.set_index('Model')
colors = ['#1f77b4', '#2ca02c', '#ff7f0e', '#9467bd']

ax = df_skor_plot.plot(kind='bar', figsize=(10, 6), color=colors)
plt.title('Perbandingan Akurasi, Presisi, Recall, dan F1-Score')
plt.ylabel('Skor')
plt.ylim(0, 1.05)
plt.xticks(rotation=0)
plt.legend(loc='lower right')

# Tambahkan label nilai persentase di atas bar
for p in ax.patches:
height = p.get_height()
if not np.isnan(height):

```

```

ax.annotate(f'{height*100:.1f}%',
(p.get_x() + p.get_width() / 2., height + 0.01),
ha='center', va='bottom', fontsize=9, color='black')
plt.tight_layout()
plt.savefig('perbandingan_model_rf_xgb_soft.jpg')
plt.close()

```

i) trainmodel.py

```

import os
import glob
import cv2
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier,
VotingClassifier
import xgboost as xgb
from sklearn.preprocessing import LabelEncoder
import joblib
# --- Konfigurasi ---
DATASET_PATH = 'hasilAugmentasi'
MODEL_FILENAME = 'models/model_ensemble.pkl'
LABEL_ENCODER_FILENAME = 'models/label_encoder.pkl'

def extract_hsv_features(image):
    """Mengekstrak fitur color moment (mean, std, skew) dari channel
    HSV."""
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    h, s, v = cv2.split(hsv_image)

    features = []
    for channel in [h, s, v]:
        mean = np.mean(channel)
        std = np.std(channel)
        skew = np.mean(((channel - mean) / std) ** 3) if std != 0 else 0
        features.extend([mean, std, skew])

    return features

def train_and_save_model():
    """Mengekstrak fitur, melatih model ensemble, dan menyimpannya."""
    print("Memulai proses ekstraksi fitur...")

    # Cari semua gambar dan ekstrak labelnya dari path
    image_paths = glob.glob(os.path.join(DATASET_PATH, '**', '*.png'),
recursive=True)
    image_paths.extend(glob.glob(os.path.join(DATASET_PATH, '**',
 '*.jpg'), recursive=True))
    image_paths.extend(glob.glob(os.path.join(DATASET_PATH, '**',
 '*.jpeg'), recursive=True))

    if not image_paths:
        print(f"Error: Tidak ada gambar yang ditemukan di '{DATASET_PATH}'
atau '{raw_dataset_path}'.")
    return

```

```

data = []
for img_path in image_paths:
    # Ekstrak label dari nama folder
    label = os.path.basename(os.path.dirname(img_path))

    # Baca gambar dan ekstrak fitur
    image = cv2.imread(img_path)
    if image is not None:
        features = extract_hsv_features(image)
        data.append([*features, label])
    else:
        print(f"Warning: Gagal membaca gambar {img_path}")

# Buat DataFrame
df = pd.DataFrame(data, columns=['mean_h', 'std_h', 'skew_h',
                                'mean_s', 'std_s', 'skew_s', 'mean_v', 'std_v', 'skew_v',
                                'label'])

print(f"Ekstraksi fitur selesai. Ditemukan {len(df)} gambar.")

# --- Pelatihan Model ---
print("Mempersiapkan data untuk pelatihan...")
X = df.drop('label', axis=1)
y = df['label']

# Encode label kelas
le = LabelEncoder()
y_encoded = le.fit_transform(y)

print(f"Kelas yang akan dilatih: {le.classes_}")

# Inisialisasi model
rf = RandomForestClassifier(n_estimators=100, random_state=42)
xgb_clf = xgb.XGBClassifier(use_label_encoder=False,
                             eval_metric='mlogloss', random_state=42)

# Buat dan latih model ensemble
print("Melatih model ensemble Soft Voting... Ini mungkin butuh beberapa saat.")
ensemble_model = VotingClassifier(estimators=[('rf', rf), ('xgb', xgb_clf)], voting='soft')
ensemble_model.fit(X, y_encoded)

# --- Simpan Model dan Encoder ---
print(f"Menyimpan model ke {MODEL_FILENAME}...")
joblib.dump(ensemble_model, MODEL_FILENAME)

print(f"Menyimpan label encoder ke {LABEL_ENCODER_FILENAME}...")
joblib.dump(le, LABEL_ENCODER_FILENAME)

print("\nPelatihan selesai. Model dan label encoder berhasil disimpan.")

if __name__ == '__main__':
    train_and_save_model()

```

j) app.py

```
import streamlit as st
import cv2
import numpy as np
import joblib
from PIL import Image
import pandas as pd
from rembg import remove

# --- Konfigurasi Halaman & Gaya ---
try:
    icon = Image.open("assets/pippernigrum.png")
except FileNotFoundError:
    icon = "🌿"

st.set_page_config(
    page_title="Analisis Kepercayaan Model untuk Klasifikasi Tingkat  
Kematangan Lada",
    page_icon=icon,
    layout="centered"
)

def set_custom_style():
    st.markdown('''
<style>
body, .stApp {
background-color: #D9EAD3 !important;}
.st-emotion-cache-1y4p8pa {
padding-top: 2rem;
}
.st-emotion-cache-1v0mbdj, .st-emotion-cache-1r4qj8v {
background-color: #FFFFFF;
border-radius: 10px;
padding: 25px;
border: 1px solid #E0E0E0;
box-shadow: 0 2px 4px rgba(0,0,0,0.05);
}
h1 {
color: #004D40; /* Warna hijau tua */
font-weight: 600;
}
h2, h3 {
color: #333333;
font-weight: 600;
}
hr {
background-color: #E0E0E0;
}
.footer {
text-align: center;
padding: 1rem;
color: #888888;
font-size: 0.9rem;
}
</style>''')
```

```

'''', unsafe_allow_html=True)
set_custom_style()
def segmentasi_roi(img):
    if img.shape[2] == 4:
        alpha = img[:, :, 3]
        _, mask = cv2.threshold(alpha, 0, 255, cv2.THRESH_BINARY)
        kernel = np.ones((5, 5), np.uint8)
        closed_mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel,
            iterations=2)
        contours, _ = cv2.findContours(closed_mask, cv2.RETR_EXTERNAL,
            cv2.CHAIN_APPROX_SIMPLE)
        if len(contours) > 0:
            x_min, y_min, x_max, y_max = np.inf, np.inf, 0, 0
            for cnt in contours:
                x, y, w, h = cv2.boundingRect(cnt)
                x_min = min(x_min, x)
                y_min = min(y_min, y)
                x_max = max(x_max, x + w)
                y_max = max(y_max, y + h)
            img_cropped = img[int(y_min):int(y_max), int(x_min):int(x_max)]
            return img_cropped
        return img

def hilangkan_alpha_dan_background_putih(img_cropped):
    if img_cropped.shape[2] == 4:
        alpha_channel = img_cropped[:, :, 3]
        rgb_channels = img_cropped[:, :, :3]
        white_bg = np.ones_like(rgb_channels, dtype=np.uint8) * 255
        alpha_factor = alpha_channel[:, :, np.newaxis] / 255.0
        img_no_alpha = rgb_channels * alpha_factor + white_bg * (1 -
            alpha_factor)
        img_final = img_no_alpha.astype(np.uint8)
        return img_final
    return img_cropped

def resize_letterbox(img, output_size=(224, 224)):
    h0, w0 = img.shape[:2]
    scale = min(output_size[0]/h0, output_size[1]/w0)
    new_w, new_h = int(w0 * scale), int(h0 * scale)
    img_resized = cv2.resize(img, (new_w, new_h))
    result = np.ones((output_size[1], output_size[0], 3),
        dtype=np.uint8) * 255
    x_offset = (output_size[0] - new_w) // 2
    y_offset = (output_size[1] - new_h) // 2
    result[y_offset:y_offset+new_h, x_offset:x_offset+new_w] =
        img_resized
    return result

def preprocess_image_from_bytes(image_bytes):
    try:
        output_data = remove(image_bytes)
        nparr = np.frombuffer(output_data, np.uint8)
        img_no_bg = cv2.imdecode(nparr, cv2.IMREAD_UNCHANGED)
        if img_no_bg is None: return None
        img_roi = segmentasi_roi(img_no_bg)
        img_clean = hilangkan_alpha_dan_background_putih(img_roi)

```

```

img_final_bgr = resize_letterbox(img_clean)
return img_final_bgr
except Exception as e:
st.error(f"Error saat pre-processing: {e}")
return None

def extract_hsv_features(image_array_rgb):
image_bgr = cv2.cvtColor(image_array_rgb, cv2.COLOR_RGB2BGR)
hsv_image = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image)
features = []
for channel in [h, s, v]:
mean = np.mean(channel)
std = np.std(channel)
skew = np.mean(((channel - mean) / std) ** 3) if std != 0 else 0
features.extend([mean, std, skew])
return np.array([features])

@st.cache_resource
def load_model():
try: return joblib.load('models/model_ensemble.pkl')
except FileNotFoundError: return None

@st.cache_resource
def load_label_encoder():
try: return joblib.load('models/label_encoder.pkl')
except FileNotFoundError: return None

model = load_model()
le = load_label_encoder()
# --- END: Fungsi-fungsi backend ---

# --- Sidebar ---
with st.sidebar:
if isinstance(icon, Image.Image):
st.image(icon, use_container_width=True)
st.title("Tentang")
st.info(
"Sistem ini menggunakan metode ensemble Soft-Voting yang
menggabungkan dua model machine learning: "
"Random Forest dan XGBoost, untuk menganalisis tingkat kematangan
buah lada secara akurat."
)
st.warning("Pastikan gambar yang diunggah memiliki kualitas baik
dan fokus pada objek lada.")

# --- Header ---
with st.container():
st.title("Analisis Kepercayaan Model untuk Klasifikasi Tingkat
Kematangan Lada")
st.write("Unggah gambar buah lada untuk diproses oleh sistem.
Sistem akan memprediksi tingkat kematangan buah lada serta
menampilkan nilai kepercayaan model terhadap hasil prediksi.")

# --- Konten Utama ---
main_container = st.container()

```

```

with main_container:
    uploaded_file = st.file_uploader("Pilih sebuah gambar...",
    type=["jpg", "jpeg", "png"])

    if model is None or le is None:
        st.error("***Model atau Label Encoder tidak ditemukan!** Pastikan
        file `model_ensemble.pkl` dan `label_encoder.pkl` ada di folder
        `models/`.")
    elif uploaded_file is not None:
        image_bytes = uploaded_file.getvalue()
        original_image = Image.open(uploaded_file)

    with st.spinner('Sistem sedang menganalisis gambar...'):
        processed_bgr_image = preprocess_image_from_bytes(image_bytes)

    if processed_bgr_image is None:
        st.error("Gagal memproses gambar. Coba lagi dengan gambar lain.")
    else:
        processed_rgb_image = cv2.cvtColor(processed_bgr_image,
        cv2.COLOR_BGR2RGB)
        features = extract_hsv_features(processed_rgb_image)
        ensemble_proba = model.predict_proba(features)[0]
        rf_model = model.estimators_[0]
        xgb_model = model.estimators_[1]
        rf_proba = rf_model.predict_proba(features)[0]
        xgb_proba = xgb_model.predict_proba(features)[0]
        top_prediction_idx = np.argmax(ensemble_proba)
        predicted_label = le.classes_[top_prediction_idx]
        confidence = ensemble_proba[top_prediction_idx] * 100

    st.header("Hasil Analisis")

    # 1. Ringkasan
    st.subheader(f"Prediksi Sistem: **{predicted_label}**")
    col1, col2 = st.columns(2)
    with col1:
        st.metric(label="Tingkat Keyakinan", value=f"{confidence:.2f}%")
        st.progress(int(confidence))
    with col2:
        st.image(original_image, caption='Gambar Asli', width=200)

    st.divider()

    # 2. Perbandingan Gambar
    st.subheader("Perbandingan Gambar")
    col1, col2 = st.columns(2)
    with col1:
        st.image(original_image, caption='Gambar Asli yang Diunggah',
        width=200)
    with col2:
        st.image(processed_rgb_image, caption='Gambar Setelah Pre-
        processing', width=200)

    st.divider()

    # 3. Distribusi Keyakinan

```

```

st.subheader("Distribusi Keyakinan Ensemble")
confidence_df = pd.DataFrame({'Kelas': le.classes_, 'Keyakinan (%)': ensemble_proba * 100})
st.bar_chart(confidence_df.set_index('Kelas'))

st.divider()

# 4. Rincian Model
st.subheader("Perbandingan Keyakinan Setiap Model")
detail_df = pd.DataFrame({
    'Kelas': le.classes_,
    'Random Forest (%)': rf_proba * 100,
    'XGBoost (%)': xgb_proba * 100,
    'Ensemble (Soft Voting) (%)': ensemble_proba * 100
})
st.dataframe(detail_df.set_index('Kelas').style.format("{:.2f}"))

# --- Footer ---
st.markdown('<div class="footer">Dibuat oleh Jihan Izzathul Mujidah © 2025</div>', unsafe_allow_html=True)

```



Lampiran 5. Permohonan Penelitian Kepada Ketua Program Studi Informatika



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 046/05/IF/C.4-VI/VII/47/2025
Lamp. : -
Hal : Permohonan Data Penelitian

Makassar, 12 Muharram 1447 H
7 Juli 2025 M

Kepada yang Terhormat,
Ketua LP3M Unismuh Makassar
Di -
Tempat

Assalamu 'Alaikum Warahmatullahi Wabarakatuh

Dengan Rahmat Allah SWT, Semoga aktivitas kita bermilai ibadah di Sisi-Nya. Dalam rangka penyelesaian Tugas Akhir pada Program Studi Informatika dengan judul "Klasifikasi Tingkat Kemungkinan Lada Menggigitkan Ensemble Learning Berdasarkan Ciri Warna Kulit", Bersama ini kami sampaikan mahasiswa:

Stambuk	Nama
105 84 1194021	Jihan Izzatul Mujidah

Sehubungan dengan hal tersebut, maka kami memohon dibuatkan surat pengantar pada instansi di bawah ini:

Nama Instansi : Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu
Alamat : Jl. Hougenville No.5, Masale, Kec. Panakkukang, Kota Makassar

Demikian surat kami atas perhatian dan kerja samanya kami haturkan banyak terima kasih.

Jazakumullah Khaerun Kairan

Wassalamu 'Alaikum warahmatullahi Wabarakatuh

Ketua Program Studi
Informatika,

Mahmudin A.M Hayat, S.Kom., M.T.
NPM 0604577

Tembusan:
1. Dekan Fakultas Teknik
2. Arsip

Lampiran 6. Surat Rekomendasi Penelitian dari LP3M



UNIVERSITAS MUHAMMADIYAH MAKASSAR
LEMBAGA PENELITIAN PENGEMBANGAN DAN PENGABDIAN KEPADA MASYARAKAT
Jl. Sultan Alauddin No. 259 Telp. 0411 866072 Fax. (0411) 865588 Makassar 90221 e-mail: lp3m@unismuh.ac.id



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 70/LP3M/05/C.4-VIII/VII/1447/2025
Lampiran : 1 (satu) rangkap proposal
Hal : Permohonan Izin Pelaksanaan Penelitian

Kepada Yth:
Bapak Gubernur Provinsi Sulawesi Selatan
Cq. Kepala Dinas Penanaman Modal & PTSP Provinsi Sulawesi Selatan
di-
Makassar

Assalamu Alaikum Wr. Wb
Berdasarkan surat Dekan Fakultas Program Study Informatika, nomor: 046 tanggal: 08 Juli 2025, menerangkan bahwa mahasiswa dengan data sebagai berikut:

Nama	JIHAN IZZATHUL MUJIDAH
Nim	: 105841104021
Fakultas	: Teknik
Prodi	: Informatika

Bermaksud melaksanakan penelitian/pengumpulan data dalam rangka penulisan laporan tugas akhir Skripsi dengan judul:
"Klasifikasi Tingkat Kematangan Lada Menggunakan Ensemble Learning Berdasarkan Citra Warna Kulit"
Yang akan dilaksanakan dari tanggal 14 Juli 2025 s/d 14 September 2025.
Sehubungan dengan maksud di atas, kiranya Mahasiswa tersebut diberikan izin untuk melakukan penelitian sesuai ketentuan yang berlaku.

Demikian, atas perhatian dan kerjasamanya diucapkan jazakumullahu khaeran katziraa.
Billahi Fii Sabilil Haq. Fatahiqul Khaerat.
Wassalamu Alaikum Wr. Wb.

Makassar 13 Muharram 1447
09 Juli 2025

Ketua LP3M Unismuh Makassar,



Dr. Muh. Arief Muhsin, M.Pd.
NBM. 112 7761



Jl. Sultan Alauddin No. 259 Telp. (0411) 866072 Fax (0411) 865588 Makassar 90221
E-mail: lp3m@unismuh.ac.id Official Web: <https://lp3m.unismuh.ac.id>

Lampiran 7. Surat Izin Penelitian dari PTSP


PEMERINTAH PROVINSI SULAWESI SELATAN
DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU
Jl. Bougainville No.5 Telp. (0411) 441077 Fax. (0411) 440606
Website : <http://pimap-nem.sulselprov.go.id> Email : ptsp@sulselprov.go.id
Makassar 90231

Nomor	: 15414/S.01/PTSP/2025	Kepada Yth.
Lampiran	: -	Bupati Luwu Timur
Perihal	: <u>Izin penelitian</u>	

di-
Tempat

Berdasarkan surat Ketua LP3M Unismuh Makassar Nomor : 70/LP3M/05/C.4-VIII/VIII/1447/2025 tanggal 09 Juli 2025 perihal tersebut diatas, mahasiswa/peneliti dibawah ini:

Nama	: JIHAN IZZATHUL MUJIDAH
Nomor Pokok	: 105841104021
Program Studi	: Teknik Informatika
Pekerjaan/Lembaga	: Mahasiswa (S1)
Alamat	: Jl. Sri Alauddin No. 259, Makassar

PROVINSI SULAWESI SELATAN

Bermaksud untuk melakukan penelitian di daerah/kantor saudara dalam rangka menyusun SKRIPSI, dengan judul :

*** KLASIFIKASI TINGKAT KEMATANGAN LADA MENGGUNAKAN ENSEMBLE LEARNING BERDASARKAN CITRA WARNA KULIT ***

Yang akan dilaksanakan dari : Tgl. 14 Juli s/d 14 September 2025

Sehubungan dengan hal tersebut diatas, pada prinsipnya kami *menyetujui* kegiatan dimaksud dengan ketentuan yang tertera di belakang surat izin penelitian.

Demikian Surat Keterangan ini diberikan agar dipergunakan sebagaimana mestinya.

Diterbitkan di Makassar
Pada Tanggal 14 Juli 2025

**KEPALA DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU
SATU PINTU PROVINSI SULAWESI SELATAN**



ASRUL SANI, S.H., M.Si.
Pangkat : PEMBINA UTAMA MUDA (IV/c)
Nip : 19750321 200312 1 008

Tembusan Yth:

1. Ketua LP3M Unismuh Makassar di Makassar,
2. Peninggal.

Lampiran 8. Hasil Turnitin

 **MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH**
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat Kantor: Jl. Sultan Alauddin No. 259 Makassar 90222 Telp (0411) 865072, 881593, Fax (0411) 865348

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Jihan Izzathul Muhidah
Nim : 105841104021
Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	8%	10 %
2	Bab 2	7%	25 %
3	Bab 3	10%	10 %
4	Bab 4	2%	10 %
5	Bab 5	5%	5 %

Dinyatakan telah lulus cek plagiat yang dilakukan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 25 Agustus 2025
Mengetahui,
Kepala UPT- Perpustakaan dan Penerbitan,


Nurhidayah S. H. M.P.
NIDN. 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)865072,881 593,fax (0411)865 348
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Bab I Jihan Izzathul Mujidah

105841104021

by Tahap Tutup



Submission date: 24-Aug-2025 06:20PM (UTC+0700)
Submission ID: 2736220144
File name: BAB_1-jihan.docx (336.95K)
Word count: 1112
Character count: 7950

Bab I Jihan Izzathul Mujidah 105841104021

ORIGINALITY REPORT

8%	6%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Itera Student Paper	3%
2	text-id.123dok.com Internet Source	2%
3	etd.umy.ac.id Internet Source	2%
4	repository.lainkudus.ac.id Internet Source	2%

Exclude quotes

Off

Exclude bibliography

Off

Exclude matches

Off



Bab II Jihan Izzathul Mujidah

105841104021

by Tahap Tutup



Submission date: 24-Aug-2025 06:21PM (UTC+0700)

Submission ID: 2734220683

File name: BAB_2-jihan.docx (330,77K)

Word count: 3555

Character count: 22200

Bab II Jihan Izzathul Mujidah 105841104021

ORIGINALITY REPORT

7%	5%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	jutif.if.unsoed.ac.id Internet Source	1%
2	Muhammad Wisnu Nugroho. "Analisis Performa Algoritma Random Forest dalam Mengatasi Overfitting pada Model Prediksi", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2025 Publication	1%
3	www.researchgate.net Internet Source	<1%
4	Submitted to Universitas Dian Nuswantoro Student Paper	<1%
5	Submitted to Universitas Gunadarma Student Paper	<1%
6	Submitted to Universitas Muslim Indonesia Student Paper	<1%
7	Submitted to Universitas Pancasila Student Paper	<1%
8	journal.sekawan-org.id Internet Source	<1%
9	ojs.unpkediri.ac.id Internet Source	<1%
10	repository.nurulfikri.ac.id Internet Source	<1%

11	Submitted to Fakultas Teknik Student Paper	<1 %
12	journal.ipb.ac.id Internet Source	<1 %
13	ejournal.undip.ac.id Internet Source	<1 %
14	docobook.com Internet Source	<1 %
15	ejournal.upnvj.ac.id Internet Source	<1 %
16	www.scribd.com Internet Source	<1 %
17	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	<1 %
18	doku.pub Internet Source	<1 %
19	id.123dok.com Internet Source	<1 %
20	www.slideshare.net Internet Source	<1 %

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

Bab III Jihan Izzathul Mujidah

105841104021

by Tahap Tutup



Submission date: 24-Aug-2025 16:24PM (UTC+0700)

Submission ID: 2394221992

File name: BAB_3-jihan.docx (181K)

Word count: 1812

Character count: 11801

Bab III Jihan Izzathul Mujidah 105841104021

ORIGINALITY REPORT

10% SIMILARITY INDEX	10% INTERNET SOURCES	5% PUBLICATIONS	4% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	-----------------------------

PRIMARY SOURCES

1	media.neliti.com Internet Source	1%
2	repository.its.ac.id Internet Source	1%
3	repository.uin-suska.ac.id Internet Source	1%
4	doaj.org Internet Source	1%
5	Submitted to UIN Sunan Ampel Surabaya Student Paper	1%
6	docplayer.info Internet Source	1%
7	Jefrin Waruwu, Abdi Dharma. "Perbandingan Algoritma Klasifikasi Pada Pasien Penyakit Jantung", INTECOMS: Journal of Information Technology and Computer Science, 2024 Publication	1%
8	Submitted to Universitas Muhammadiyah Makassar Student Paper	1%
9	journal.ikopin.ac.id Internet Source	1%
10	repository.upi.edu Internet Source	1%

Bab IV Jihan Izzathul Mujidah

105841104021

by Tahap Tutup

Submission date: 24-Aug-2025 06:29PM (UTC+0750)
Submission ID: 2734223397
File name: BAB_4-jihan.docx (2.05M)
Word count: 6392
Character count: 38384



Bab IV Jihan Izzathul Mujidah 105841104021

ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

2%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES



e-jurnal.lppmunsera.org
Internet Source

2%

Exclude quotes Off
Exclude bibliography Off

Exclude matches < 2%



Bab V Jihan Izzathul Mujidah

105841104021

by Tahap Tutup



Submission date: 24-Aug-2025 06:31 PM (UTC+0700)

Submission ID: 2734224281

File name: BAB_5-jihan.docx (42.62K)

Word count: 263

Character count: 1793

Bab V Jihan Izzathul Mujidah 105841104021

ORIGINALITY REPORT

5%	5%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

 repository.uinjkt.ac.id	5%
Internet Source	

Exclude quotes Off

Exclude bibliography Off

Exclude matches < 2%

