

**IMPLEMENTASI ALGORITMA RANDOM FOREST UNTUK
IDENTIFIKASI CITRA KAYU MAKROSKOPIK DENGAN
EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE
MATRIX (GLCM)**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat dalam menyelesaikan studi jenjang pendidikan strata 1 (S1) program studi Teknik Informatika pada Universitas Sangga Buana YPKP Bandung

Disusun Oleh:

Muhammad Ghiffaari Ilham Ramadhan

2113201060



**FAKULTAS TEKNIK
PROGRAM STUDI S1 TEKNIK INFORMATIKA
UNIVERSITAS SANGGA BUANA YPKP
BANDUNG
2024**

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Ghiffaari Ilham Ramadhan
NPM : 2113201060
Program Studi : Teknik Informatika

Dengan ini menyatakan bahwa penelitian yang saya buat dengan judul **“IMPLEMENTASI ALGORITMA RANDOM FOREST UNTUK IDENTIFIKASI CITRA KAYU MAKROSKOPIK DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)”** adalah asli atau tidak menjiplak (plagiat) dan belum pernah di publikasikan di mana pun dan dalam bentuk apa pun.

Demikian surat pernyataan ini saya buat dengan sebenarnya tanpa ada paksaan dan tekanan dari pihak mana pun dan apabila dikemudian hari ternyata ada pihak lain yang mengklaim judul dan isi penelitian ini atau saya memberi keterangan palsu maka saya bersedia kelulusan saya dari program studi Teknik Informatika dibatalkan.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Jika dikemudian hari terbukti bahwa laporan skripsi saya merupakan hasil jiplakan, maka saya bersedia menerima sanksi apa pun yang diberikan.

Bandung, 12 Agustus 2024


METERAI TEMPEL
SEPULUH RIBU RUPIAH
TR
DDALX224499085
Muhammad Ghiffaari Ilham Ramadhan

LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Muhammad Ghiffaari Ilham Ramadhan

NPM : 2113201060

Program Studi : Teknik Informatika

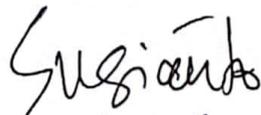
Judul : IMPLEMENTASI ALGORITMA RANDOM FOREST UNTUK IDENTIFIKASI CITRA KAYU MAKROSKOPIK DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)

Untuk dipertahankan pada sidang skripsi Semester Genap Tahun 2023/2024 di hadapan para penguji dan diterima sebagai bagian dari persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik (S.T) pada Fakultas Teknik Program Studi S1 Teknik Informatika Universitas Sangga Buana YPKP Bandung.

Bandung, 12 Agustus 2024

Menyetujui,

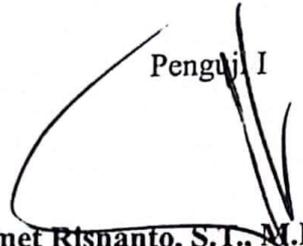
Pembimbing



Bambang Sugiarto, S.T., M.T.

NIDN. 8861060017

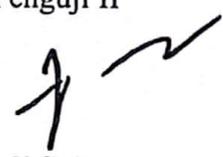
Penguji I



Slamet Rishanto, S.T., M.Kom., Ph.D.

NIDN. 0424047307

Penguji II



Dr. Teguh Nurhadi Suharsono, S.T., M.T.

NIDN. 0021077101

Mengetahui:

Ketua Program Studi S1 Teknik Informatika



Gunawan, S.T., M.Kom., MOS., MTA., MCE.

NIDN. 0404027604

KATA PENGANTAR

Segala puji dan syukur atas kehadiran Allah SWT atas berkah dan karunia-Nya yang senantiasa dilimpahkan kepada penulis, sehingga penulis dapat menyelesaikan skripsi ini dengan judul “*Implementasi Algoritma Random Forest untuk Identifikasi Citra Kayu Makroskopik dengan ekstraksi fitur Gray Level Co-occurrence Matrix (GLCM)*” sebagai syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Universitas Sangga Buana YPKP.

Dalam penyusunan laporan skripsi ini, penulis memiliki beberapa hambatan dan rintangan yang dihadapi, namun pada akhirnya dapat dilalui berkat adanya bantuan dari berbagai pihak baik secara moral maupun spiritual. Sehingga penulis akan menyampaikan banyak terima kasih kepada:

1. Gunawan, S.T., M.Kom., MOS., MTA., MCE. Selaku Ketua Program Studi dan Jurusan S1 Teknik Informatika.
2. Bambang Sugiarto, S.T., M.T. sebagai dosen pembimbing baik dalam penulisan skripsi ini, terima kasih atas ilmu yang diberikan serta kesabarannya.
3. Gunawansyah S.T., M.Kom., MOM., MCP., MTA. Selaku dosen wali yang selalu membantu kapan pun kepada penulis selama perkuliahan.
4. Keluarga AZFAMILY, khususnya Bapak Azwin, dan Ibu Siti Nurdjanah yang telah memberikan motivasi kepada penulis dan memberikan doa serta dukungannya dalam bentuk moril maupun materiil.
5. Teman-teman seperjuangan angkatan 2020 Teknik Informatika yang saling membantu satu sama lain.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, memiliki beragam kekurangan, dan keterbatasan. Penulis berharap kritikan dan saran yang membangun dapat diberikan dan akan dijadikan sebagai acuan untuk menjadi lebih baik lagi kedepannya.

ABSTRAK

Perkembangan teknologi yang pesat telah mendorong berbagai inovasi dalam bidang Machine Learning, Computer Vision, dan Pengolahan Citra, termasuk dalam proses identifikasi citra kayu yang menjadi kunci dalam industri kayu. Studi ini bertujuan untuk mengembangkan sistem identifikasi jenis kayu menggunakan algoritma Random Forest dan metode ekstraksi fitur Gray Level Co-occurrence Matrix (GLCM) untuk meningkatkan akurasi dan efisiensi dibandingkan dengan metode identifikasi manual yang rentan terhadap kesalahan. Terdapat 5 jenis kayu sebagai bagian dari dataset kayu yang digunakan, dengan jumlah 750 citra, di mana terdapat 150 citra mewakili masing-masing jenis citra kayu. Berdasarkan analisis hasil pelatihan dan pengujian model, model C dengan proporsi data pelatihan 90:10 menunjukkan performa terbaik dengan akurasi 81%, dibandingkan dengan Model A (70:30) dengan akurasi 74% dan model B (80:20) dengan akurasi 79%. Bagian pengujian dengan citra baru berjumlah 5 citra acak menunjukkan perbedaan kinerja yang signifikan terkait dengan probabilitas prediksi dan citra yang dikenali, di mana hasil tertinggi didapatkan oleh model C (90:10) sangat akurat secara keseluruhan dengan probabilitas tertinggi untuk sebagian besar jenis citra dengan skor probabilitas tertinggi 99% dan terendah 75%. Hasil penelitian ini menunjukkan bahwa model Random Forest dapat menjadi alat yang efektif untuk identifikasi citra jenis kayu.

Kata Kunci: kayu, *machine learning*, *computer vision*, sistem identifikasi, *random forest*, *glcm*

ABSTRACT

The rapid advancement in technology has driven numerous innovations in Machine Learning, Computer Vision, and Image Processing, particularly in the identification of wood species, which is crucial for the timber industry. This study aims to develop a wood species identification system using Random Forest algorithms and Gray Level Co-occurrence Matrix (GLCM) feature extraction methods to enhance accuracy and efficiency compared to manual identification methods, which are prone to errors. The dataset includes 750 images of 5 different wood species, with 150 images representing each species. Based on the training and testing results, Model C with a 90:10 training-to-testing ratio demonstrated the best performance with an accuracy of 81%, compared to Model A (70:30) with 74% accuracy and Model B (80:20) with 79% accuracy. Testing on 5 randomly chosen new images yielded the highest results, indicating that the Random Forest model can be an effective tool for wood species identification.

Keywords: *wood, machine learning, computer vision, identification system, random forest, glcm*

DAFTAR ISI

SURAT PERNYATAAN.....	i
LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI	Error! Bookmark not defined.
KATA PENGANTAR	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	7
2.1 Penelitian Terdahulu.....	7
2.2 Computer Vision	9
2.3 Gray Level Co-occurrence Matrix (GLCM)	10
2.4 Machine Learning.....	11
2.4.1 Supervised Learning.....	12
2.5 Pengolahan Citra Digital	13
2.6 Python.....	14
2.7 Sci-kit Learn	15
2.8 Tkinter	16
2.9 CustomTkinter.....	17
2.10 Jupyter Notebook.....	17
2.11 Visual Studio Code	18
2.12 Confusion Matrix.....	18

2.13	Random Forest.....	19
2.14	Unified Modeling Language (UML)	22
2.14.1	Use Case Diagram	23
2.14.2	Activity Diagram	26
2.14.3	Sequence Diagram.....	29
BAB III METODE PENELITIAN.....		33
3.1	Metodologi Penelitian	33
3.2	Penelitian Pendahuluan	34
3.3	Pengembangan Sistem.....	37
3.3.1	Analisis Sistem.....	37
3.3.2	Tahapan Pengembangan Sistem.....	41
3.3.2.1	Pengumpulan Data	42
3.3.2.2	Pembagian Data	44
3.3.2.3	Image Preprocessing	44
3.3.2.4	Ekstraksi Fitur	46
3.3.2.5	Implementasi Algoritma Random Forest.....	47
3.3.3	Perancangan UML	51
3.3.3.1	Use Case Diagram.....	51
3.3.3.2	Activity Diagram.....	53
3.3.3.3	Sequence Diagram	55
3.3.4	Perancangan Mockup Aplikasi	57
BAB IV PEMBAHASAN & HASIL.....		59
4.1	Hasil Pengembangan Sistem	59
4.1.1	Validasi Sistem	59
4.1.1.1	Performance Matrix	59
4.1.1.2	Confusion Matrix	64
4.1.2	Pengujian Sistem.....	67
4.1.3	Evaluasi Hasil.....	71
BAB V PENUTUP.....		73
5.1	Kesimpulan.....	73
5.2	Saran.....	73

DAFTAR PUSTAKA 74



DAFTAR GAMBAR

Gambar 2.1 Cara Kerja Metode Random Forest.....	22
Gambar 2.2 Contoh UML: Use Case Diagram	26
Gambar 2.3 Contoh UML: Activity Diagram	29
Gambar 2.4 Contoh UML: Sequence Diagram.....	32
Gambar 3.1 Metodologi Penelitian	34
Gambar 3.2 Diagram Alur Sistem Berjalan	39
Gambar 3.3 Diagram Alur Sistem yang Diusulkan	41
Gambar 3.4 Tahapan Pengembangan Sistem.....	42
Gambar 3.5 Image Preprocessing Result	45
Gambar 3.6 Alur Pohon Keputusan	48
Gambar 3.7 Pohon Keputusan Fitur GLCM	48
Gambar 3.8 Use Case Diagram.....	51
Gambar 3.9 Activity Diagram: Upload Model	54
Gambar 3.10 Activity Diagram: Upload Image.....	54
Gambar 3.11 Activity Diagram: Classify Image.....	55
Gambar 3.12 Sequence Diagram: Upload Model	56
Gambar 3.13 Sequence Diagram: Upload Image.....	56
Gambar 3.14 Sequence Diagram: Classify Image	57
Gambar 3.15 Mockup Aplikasi Desktop.....	57
Gambar 4.1 Confusion Matrix: Model A (70/30).....	64
Gambar 4.2 Confusion Matrix: Model B (80/20)	65
Gambar 4.3 Confusion Matrix: Model C (90/10)	66
Gambar 4.4 Tangkapan Layar Aplikasi: Halaman Utama	68
Gambar 4.5 Model Loaded	69
Gambar 4.6 Image Loaded.....	70
Gambar 4.7 Identification Result	71

DAFTAR TABEL

Tabel 2.2 Notasi UML: Use Case Diagram	23
Tabel 2.3 Notasi UML: Activity Diagram	26
Tabel 2.4 Notasi UML: Sequence Diagram	29
Tabel 3.1 Pemetaan Penelitian Pendahuluan	34
Tabel 3.2 Pemetaan Penelitian Terdahulu.....	36
Tabel 3.3 Sampel Citra Kayu	43
Tabel 3.4 Skenario Split Dataset.....	44
Tabel 3.5 Distribusi Kelas.....	49
Tabel 3.6 Skenario Use Case: Upload Model	52
Tabel 3.7 Skenario Use Case: Upload Image.....	52
Tabel 3.8 Skenario Use Case: Classify Image	53
Tabel 4.1 Confusion Matrix dalam Tabel	59
Tabel 4.2 Performance Matrix Result	62
Tabel 4.3 Skor Akurasi Model	67
Tabel 4.4 Hasil Pengujian Citra Baru	71

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi telah mengubah berbagai aspek kehidupan manusia (Danuri, 2019). Di era digital ini berbagai sektor teknologi menjadi pendorong utama pada perubahan dan menghadirkan inovasi baru untuk kehidupan sehari-hari. Termasuk di dalamnya bidang *Machine Learning* dan *Computer Vision*. Perkembangan teknologi juga membuka peluang baru untuk meningkatkan efisiensi dan akurasi dalam berbagai hal termasuk proses identifikasi pada kayu.

Machine Learning atau Pembelajaran Mesin adalah salah satu cabang dari Kecerdasan Buatan yang memungkinkan komputer belajar dari data dan pengalaman tanpa pemrograman secara rinci, mengenali pola dalam data, dan membuat suatu ketetapan tanpa adanya panduan dari manusia lagi dengan spesifik (Faiz, 2023). Dalam bidang pemrosesan citra dan kecerdasan buatan juga telah memungkinkan pengembangan solusi yang inovatif. Teknologi-teknologi ini nantinya dapat memberikan peluang besar untuk mengembangkan sistem identifikasi citra kayu menggunakan metode-metode baru yang efektif dan efisien.

Industri kayu merupakan salah satu sektor primer pada perekonomian Indonesia, dukungan dari pemerintah untuk perkembangan industri kayu di Indonesia diperlukan, baik berupa pendanaan, teknologi, hingga melibatkan berbagai pihak terkait (Mutaqin et al., 2022). Kayu dapat digunakan dalam beragam hal seperti konstruksi, pabrikasi meubel furnitur, dan bahan baku industri lainnya (Adianto, 2020). Identifikasi citra kayu merupakan aspek penting dalam industri kayu. Dengan menggunakan teknologi *machine learning*, *computer vision*, dan pengolahan citra, kita dapat mengoptimalkan proses identifikasi citra kayu secara cepat dan tepat.

Kayu memiliki ciri khas tekstur yang dapat dijadikan landasan untuk pengembangan model identifikasi. Studi ini berfokus pada penerapan teknologi

machine learning, computer vision dan pengolahan citra untuk mengatasi tantangan dalam industri kayu yang terkait dengan identifikasi jenis kayu.

Banyaknya industri kayu di Indonesia yang masih memiliki keterbatasan peralatan identifikasi kayu yang modern dan canggih menyebabkan proses identifikasi kayu menjadi lambat dan kurang akurat, sehingga menyebabkan rendahnya efisiensi dan kualitas produk yang dihasilkan (Susanti, 2023). Pada industri kayu, terdapat juga ahli anatomi kayu yang memiliki keahlian dalam identifikasi spesies kayu, tetapi ketersediaan identifikasi kayu terbatas karena jumlah ahli anatomi kayu yang sedikit (Prakasa et al., 2023).

Saat ini, identifikasi jenis kayu umumnya masih dilakukan secara manual oleh manusia berdasarkan ciri-ciri tertentu, umumnya dapat dilakukan secara langsung oleh panca indera, baik berdasarkan bau, tekstur, warna hingga corak, ataupun berdasarkan ciri anatomi yang hanya dapat diamati secara jelas dengan bantuan alat seperti kaca pembesar atau mikroskop (Rahayu et al., 2021). Proses ini tentunya memerlukan pengulangan dan waktu pelatihan yang cukup lama sehingga menjadi tidak efektif (Utami Putri & Redi Susanto, 2020). Keterbatasan dalam keterampilan visual untuk mengidentifikasi jenis kayu dapat mempengaruhi akurasi hasil yang diinginkan dalam dunia industri, sehingga diperlukan solusi untuk mengatasi masalah ini, yaitu sistem untuk mengidentifikasi citra kayu secara otomatis menggunakan algoritma berbasis *Machine Learning*.

Melalui identifikasi citra kayu makroskopik dengan memanfaatkan teknologi *Machine Learning* ini, diharapkan dapat memberikan hasil yang akurat dan efektif dibandingkan dengan proses identifikasi secara manual. Diharapkan juga dapat mempercepat proses identifikasi jenis kayu sehingga dapat memberikan kontribusi positif terhadap perkembangan industri kayu dan pengelolaan sumber daya kayu secara berkelanjutan.

1.2 Identifikasi Masalah

Berdasarkan isian latar belakang yang telah dicantumkan di atas, maka dapat diuraikan sebagai berikut.

1. Keterbatasan dalam penerapan teknologi untuk melakukan proses identifikasi kayu secara manual tentu rentan terhadap keakuratan yang dapat mempengaruhi proses identifikasi kayu.
2. Diperlukan proses validasi sistem untuk memastikan bahwa sistem identifikasi kayu yang dikembangkan dapat digunakan secara efektif.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, mengenai rumusan masalah dalam penelitian ini diuraikan sebagai berikut.

1. Bagaimana cara mengimplementasikan metode *Machine Learning* dan *Computer Vision* untuk mengatasi ketidakakuratan dalam identifikasi kayu?
2. Bagaimana cara melakukan validasi yang efektif untuk memastikan tingkat kinerja akurasi yang tinggi dalam mengidentifikasi jenis kayu?

1.4 Tujuan Penelitian

Bersumber pada identifikasi dan rumusan masalah di atas. Tujuan penelitian yang sudah ditentukan, diuraikan sebagai berikut.

1. Mengembangkan sistem identifikasi kayu dengan mengimplementasikan metode klasifikasi *Random Forest*, dan metode ekstraksi fitur menggunakan *Gray Level Co-occurrence Matrix (GLCM)* untuk mengatasi keakuratan dalam proses identifikasi.
2. Menguji dan mengevaluasi metode validasi sistem untuk mempercepat identifikasi jenis kayu.

1.5 Manfaat Penelitian

Dari penelitian ini diharapkan dapat memberikan kontribusi yang bermanfaat bagi masyarakat, yang dapat dijelaskan sebagai berikut.

1. Keterlibatan positif terhadap pengembangan teknologi *Machine Learning* dan *Computer Vision* khususnya identifikasi jenis kayu secara otomatis.
2. Solusi yang efektif untuk meningkatkan efisiensi dalam proses identifikasi jenis kayu dalam industri kayu.
3. Mengurangi kesalahan dan ketidakpastian dalam identifikasi jenis kayu.

4. Mengurangi ketergantungan pada metode identifikasi manual yang memerlukan waktu dan keterampilan tinggi, sehingga mempercepat proses identifikasi dan mengurangi biaya operasional.
5. Mendorong penerapan teknologi canggih dalam industri kayu di Indonesia, yang dapat membuka peluang untuk inovasi dan pengembangan teknologi di sektor lain.
6. Berkontribusi pada pertumbuhan ekonomi lokal melalui peningkatan produktivitas dan kualitas produk kayu yang lebih baik.
7. Hasil penelitian dapat digunakan sebagai bahan referensi untuk pendidikan dan penelitian lebih lanjut dalam bidang teknologi *machine learning* dan *computer vision*.

1.6 Sistematika Penulisan

Dalam penyusunan skripsi pada penelitian ini, terdapat sistematika penulisannya yang dibagi menjadi beberapa bab. Penjelasan sistematika penulisan dapat dijelaskan sebagai berikut.

BAB I PENDAHULUAN

Bab Pendahuluan berisi latar belakang penelitian yang menguraikan konteks penelitian permasalahan yang dihadapi dalam industri kayu, kebutuhan serta implementasi akan teknologi baru dan urgensi topik penelitian, serta identifikasi dan rumusan masalah yang menjadi fokus penelitian yang ingin dijawab. Tujuan penelitian diidentifikasi untuk menentukan hasil yang ingin dicapai, dan manfaat penelitian diuraikan untuk menunjukkan kontribusinya terhadap ilmu pengetahuan serta praktik di lapangan. Bab ini juga menjelaskan ruang lingkup penelitian untuk menetapkan batasan-batasan yang relevan dan sistematika penulisan yang memberikan gambaran struktur laporan. Elemen-elemen ini secara keseluruhan membangun fondasi yang jelas bagi penelitian yang dilakukan.

BAB II LANDASAN TEORI

Bab ini mencakup pembahasan-pembahasan mengenai teori yang berkaitan dengan penelitian yang sedang dilakukan, teori yang lengkap tentang industri kayu seperti industri kayu di Indonesia yang masih memiliki keterbatasan peralatan

identifikasi kayu yang modern, kayu-kayu yang digunakan meliputi Angsana, Kenari, Lengkeng, Meranti, dan Sonokeling serta bagaimana Kayu memiliki ciri khas tekstur yang dapat dijadikan landasan untuk pengembangan sistem identifikasi. Teori konsep dasar pembelajaran mesin termasuk pengertian metode algoritma random forest dan bagaimana algoritma ini diterapkan dalam pengolahan citra, penjelasan mengenai teknik dan metode dari computer vision berupa image preprocessing yang digunakan untuk analisis citra kayu, metode Gray Level Co-occurrence Matrix (GLCM) dan bagaimana ekstraksi fitur ini digunakan untuk meningkatkan akurasi identifikasi citra kayu, hingga deskripsi alat dan perangkat lunak yang digunakan dalam penelitian, seperti perangkat keras untuk pengambilan citra, serta perangkat lunak untuk pengolahan data dan pengembangan sistem, termasuk Unified Modeling Language (UML), Bahasa Pemrograman Python, Text Editor seperti Jupyter Notebook untuk pengembangan program pelatihan model dan Visual Studio Code untuk pengembangan aplikasi pengujian, yang dijadikan sebagai rujukan dalam penelitian ini. Referensi teori diambil dari artikel jurnal, buku, hingga sumber bereputasi tinggi lainnya yang relevan dengan topik penelitian sebagai acuan penelitian.

BAB III METODE PENELITIAN

Bab Metode Penelitian berisi penjelasan dan langkah-langkah yang dilakukan pada penelitian ini, berupa gambaran desain penelitian, dimulai dari penelitian pendahuluan sebagai tahap awal yang mencakup identifikasi masalah dengan melakukan observasi pada permasalahan yang ada di masyarakat dan industri kayu seperti keterbatasan alat bantu manual, termasuk jenis data yang dikumpulkan yaitu citra kayu, metode pengumpulan data, dan tahapan atau prosedur yang dilakukan dalam melakukan penelitian ini dengan sistematis. Tahapan pengembangan sistem identifikasi citra kayu, termasuk analisis sistem berupa analisis sistem yang berjalan serta perbandingan dengan metode sistem yang diusulkan dan implementasi model machine learning menggunakan algoritma Random Forest dan metode ekstraksi fitur GLCM. Tahapan pengujian aplikasi yang dikembangkan, termasuk pengujian fungsionalitas dan evaluasi hasil yang diperoleh. Dengan metode penelitian ini,

peneliti dapat menemukan jawaban atas masalah yang diajukan, dan dapat memberikan pemahaman terhadap proses penelitian yang dilakukan.

BAB IV HASIL DAN PEMBAHASAN

Bab Hasil dan Pembahasan berisi tangkapan layar dan penjelasan mengenai antarmuka aplikasi yang dibangun serta hasil identifikasi citra kayu, pembahasan mengenai performa model machine learning yang dikembangkan dengan aplikasi Jupyter Notebook yang menghasilkan Performance Matrix yang berisi akurasi presisi, recall, dan F1-Score untuk melihat kinerja model yang sudah dilatih sebagai validasi sistem, dan juga menghasilkan visualisasi dalam bentuk Confusion Matrix, Interpretasi hasil penelitian termasuk penjelasan mengenai kesimpulan dari hasil pengujian dengan data citra acak dan bagaimana hasil tersebut menjawab tujuan dan rumusan masalah pada penelitian pendahuluan.

BAB V PENUTUP

Bab ini merangkum kesimpulan dari hasil penelitian, dengan ringkasan pencapaian penelitian yang telah dicapai dan bagaimana penelitian ini menjawab rumusan masalah yang diajukan. Memberikan saran-saran untuk pengembangan selanjutnya termasuk potensi perbaikan sistem, pengembangan metode baru, serta kontribusi penelitian terhadap industri kayu, masyarakat dan teknologi machine learning untuk identifikasi citra. Diskusi mengenai implikasi praktis dari hasil penelitian untuk industri kayu, serta potensi manfaat bagi stakeholders dan masyarakat.

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada subbab ini akan mengkaji beberapa penelitian terdahulu yang relevan dengan topik penelitian ini. Dapat dijelaskan sebagai berikut.

Menurut (Muna et al., 2020), pada penelitiannya yang menerapkan algoritma *Random Forest* untuk melakukan identifikasi dehidrasi berdasarkan citra urine, mengemukakan bahwa tingkat dehidrasi dapat diamati secara visual dari warna urine, yang seringkali sulit dibedakan oleh panca indera manusia karena kemiripan warna. Penelitian ini menunjukkan bahwa penggunaan teknologi kamera dengan sistem cerdas, seperti algoritma *Random Forest*, dapat mengatasi keterbatasan ini dengan memberikan akurasi identifikasi sebesar 90% pada 30 dataset, serta *precision* 90.2%, *recall* 90%, *Mean Absolute Error* 0.2473, dan *Root Mean Squared Error* 0.3208, yang mendemonstrasikan efektivitas model dalam mengklasifikasikan tingkat dehidrasi berdasarkan warna urine.

Menurut (Suaib et al., 2022), dalam penelitiannya yang membahas identifikasi Covid-19 menggunakan *X-Ray* paru-paru, mengemukakan bahwa untuk mendiagnosis Covid-19, analisis citra *X-Ray* paru-paru dapat dilakukan dengan metode ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) dan metode klasifikasi menggunakan *Random Forest*. Penelitian ini menggunakan dataset yang terdiri dari 1.200 gambar *X-Ray* paru-paru, yang dibagi menjadi 900 data pelatihan dan 300 data uji, dengan tiga kategori klasifikasi yaitu paru-paru normal, positif Covid-19, dan Pneumonia. Hasil uji menunjukkan bahwa sistem mampu mengidentifikasi Covid-19 dan mengklasifikasikan gambar *X-Ray* paru-paru ke dalam tiga kelas dengan akurasi tertinggi sebesar 85.67%, yang diperoleh dengan menggunakan variasi *resizing* gambar 200x200 piksel, radius LBP 8, dan jumlah pohon dalam *Random Forest* sebanyak 200.

Menurut (Wahyuningtyas et al., 2022), dalam penelitian mengenai identifikasi penyakit pada daun kopi, dikemukakan bahwa sistem pengolahan citra

digital yang dirancang menggunakan metode ekstraksi ciri *Local Binary Pattern* (LBP) dan metode klasifikasi *Random Forest* dapat mengidentifikasi tiga jenis penyakit pada daun kopi, yaitu *Leaf Blight*, *Leaf Miner*, dan *Leaf Rust*. Dengan menggunakan dataset sebanyak 240 citra, yang terbagi menjadi 192 citra data latih dan 48 citra data uji, sistem ini mencapai akurasi terbaik sebesar 95,83% dengan parameter ukuran citra 128x128 piksel, radius LBP 1, dan jumlah *estimator Random Forest* sebanyak 100. Penelitian ini menunjukkan bahwa sistem yang dikembangkan dapat secara efektif mengidentifikasi penyakit pada daun kopi, membantu mengatasi masalah identifikasi yang sulit dalam skala perkebunan yang luas.

Menurut (Nadiyah Hidayati & Maulidah, 2023), dalam penelitiannya mengenai pengklasifikasian jenis kupu-kupu, dikemukakan bahwa untuk mengidentifikasi berbagai spesies kupu-kupu, dilakukan penggabungan dari tiga metode ekstraksi fitur yaitu *color histogram*, *Haralick*, dan *Hu-moments*, yang kemudian diklasifikasikan menggunakan metode *Random Forest* (RF). Penelitian ini melibatkan 2.400 citra kupu-kupu yang dibagi menjadi dua kelas. Hasil pengujian menunjukkan bahwa akurasi sistem mencapai 75%, dengan *precision* 78% dan *recall* 69%. Penelitian ini menyimpulkan bahwa algoritma *Random Forest* memberikan akurasi tertinggi dibandingkan dengan algoritma klasifikasi lainnya, dan *color histogram* merupakan fitur ekstraksi terbaik dalam eksperimen ini.

Menurut (Rosyani et al., 2021), dalam penelitian tentang klasifikasi citra bunga, dilakukan segmentasi untuk menghilangkan *noise* dari latar belakang serta untuk memperoleh fitur gambar yang relevan. Fitur yang digunakan termasuk *eccentricity*, *perimeter*, *metric*, dan *area*. Penelitian ini memanfaatkan 120 gambar dari dataset 17 bunga, yang terdiri dari dua jenis bunga berwarna putih dan kuning. Data dibagi menjadi data pelatihan dan data uji, dan diproses menggunakan metode klasifikasi *Random Forest* dan SMO dengan skenario *10-fold cross-validation* dan 66% split. Metode klasifikasi menggunakan *Random Forest* mendapatkan hasil yang lebih baik, yaitu akurasi 100% untuk kedua skenario $k=10$ dan split 66%. Hal ini disebabkan oleh proses segmentasi yang efektif dalam membersihkan *noise*,

sehingga hanya area terpenting yang dihitung oleh sistem. Sebaliknya, SMO tidak jauh berbeda dari *Random Forest* pada skenario split 66%, dengan akurasi sebesar 92.68%.

2.2 Computer Vision

Computer Vision (CV), dapat diartikan sebagai yang berfokus pada pengembangan metode agar komputer dapat mengenali dan menginterpretasi visual dalam citra digital, seperti foto dan video. *Computer Vision* memungkinkan komputer untuk memproses informasi dalam bentuk piksel dengan berbagai warna, membandingkan objek citra, meningkatkan transparansi objek, serta membantu meringankan pekerjaan manusia. *Vision* dalam konteks ini adalah sistem evaluasi informasi yang bersumber dari citra dan umumnya dari kamera, dengan ekstraksi penting dan menggunakan algoritma tertentu (Kristiawan et al., 2020).

Tujuan utama dari *Computer Vision* yaitu untuk mengetahui konten citra digital. Melibatkan pengembangan metode yang mencoba mereplikasi kapabilitas penglihatan manusia. Melibatkan ekstraksi informasi dari gambar, seperti objek, deskripsi teks, dan model tiga dimensi. Pandangan lain tentang *Computer Vision* menyatakan bahwa itu adalah ekstraksi otomatis informasi dari gambar, yang bisa mencakup model tiga dimensi, posisi kamera, deteksi dan identifikasi objek, hingga penelusuran konten citra.

Computer Vision juga mencakup *Image Processing*, di mana gambar dan video diolah. *Image Processing* melibatkan pembuatan gambar baru dari gambar yang ada, seringkali dengan menyederhanakan atau meningkatkan kontennya. Termasuk normalisasi sifat fotometrik gambar, seperti kecerahan dan warna, pemotongan batas gambar untuk memusatkan objek, dan penghilangan noise digital. Beberapa contoh pengimplementasian bidang *computer vision* seperti klasifikasi objek, identifikasi objek, deteksi objek, segmentasi objek, hingga rekognisi objek. (Ariyani & Arif, 2022).

2.3 Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix adalah salah satu metode ekstraksi fitur tekstur yang bekerja berdasarkan perhitungan statistik yang merujuk kepada nilai piksel asli dan kaitan tetangga antara dua piksel dengan jarak spesifik. Langkah awal untuk menghitung fitur Matriks Kemunculan Tingkat Abu-abu adalah dengan membuat matriks kemunculan yang diteruskan dengan menentukan kaitan spasial antara piksel tetangga dan frekuensi piksel berdasarkan sudut 0 derajat. GLCM dihitung berdasarkan seberapa sering *Gray pixel Level i* tampak secara horizontal, vertikal, atau diagonal dan berpasangan dengan Gray piksel Level *j*.

Fitur yang diekstraksi oleh GLCM meliputi korelasi, homogenitas, kontras, dan energi. Homogenitas adalah distribusi komponen dalam GLCM, korelasi digunakan untuk mengukur probabilitas bahwa pasangan piksel muncul secara serentak, kontras digunakan untuk variasi lokal dalam GLCM, dan energi digunakan untuk menaksir setiap elemen pangkat dua (Lamasigi & Bode, 2021).

GLCM juga merupakan salah satu metode ekstraksi fitur dalam tekstur yang termasuk dalam second-order statistics sebagai teknik untuk mendapatkan nilai dengan menghitung probabilitas hubungan tetangga dua piksel pada jarak (*d*) dan sudut (θ) tertentu (Mulyono et al., 2020).

Rumus untuk menghitung nilai GLCM pada posisi (*i,j*) yang memiliki hubungan tetangga pada jarak dan sudut spesifik pada citra. Rumus tersebut dapat dilihat pada Persamaan (2.1).

$$GLCM(i, j, d, \theta) = \sum_{x=1}^N \sum_{y=1}^M \begin{cases} 1, & \text{jika } I(x, y) = i \text{ dan } I(x + d \cos(\theta), y + d \sin(\theta)) = j \\ 0, & \text{jika lainnya} \end{cases} \quad (2.1)$$

Dimana:

1. $I(x,y)$ adalah intensitas piksel pada posisi (x,y) pada citra.
2. d adalah jarak antara dua piksel.
3. θ adalah sudut antar dua piksel.
4. N dan M adalah lebar dan tinggi citra, secara berturut-turut.

Cara Kerja Rumus:

1. Iterasi Semua Piksel: Untuk setiap piksel (x,y) pada citra:
2. Hitung posisi piksel tetangga dengan jarak d dan sudut θ : $(x + d\cos(\theta), y + d\sin(\theta))$.
3. Perbandingan Intensitas: Bandingkan intensitas piksel saat ini $(I(x,y))$ dengan intensitas piksel tetangganya.
4. Jika intensitas keduanya sama dengan i dan j masing-masing, maka tambahkan 1 pada elemen $GLCM(i,j)$.
5. Jika tidak sama, maka tidak ada perubahan pada elemen $GLCM(i,j)$.
6. Ulangi langkah 1 dan langkah 2 untuk semua piksel pada citra.

2.4 Machine Learning

Machine Learning merupakan komponen dari kecerdasan buatan yang terfokus pada pembelajaran dari data, di mana perhatiannya tertuju pada pengembangan sistem yang dapat belajar secara otomatis tanpa perlu diprogram secara berulang oleh manusia (Cholissodin et al., 2020).

Machine learning menerapkan proses belajar seperti yang dilakukan oleh manusia pada mesin, menciptakan model matematis yang dapat mencerminkan pola-pola dalam data, lalu mengumpulkan data, kemudian mengolahnya menjadi informasi atau pengetahuan baru (Putra, 2020).

Machine learning memungkinkan suatu perangkat untuk menunjang tugas pengguna atau menghasilkan informasi sesuai keinginan pengguna dengan cara melatih perangkat tersebut menggunakan data yang tersedia. Menurut penelitian Jaspers, *machine learning* merupakan suatu proses yang secara otomatis berkembang atau belajar dari pengalaman atau penelitian, lalu bertindak tanpa perlu diprogram terlebih dahulu karena telah dilatih dengan menggunakan data riset atau data sebelumnya.

Penerapan *machine learning* dalam komputasi semakin mengoptimalkan proses pengolahan data, terutama data kompleks. *Machine learning* telah merevolusi komputasi dengan kemampuannya untuk beradaptasi. Algoritma dari

machine learning yang terus belajar dari data masa lalu memungkinkan sistem untuk membuat keputusan yang lebih baik saat menghadapi data baru yang kompleks. Algoritma *machine learning* telah menjadi kunci dalam meningkatkan kemampuan komputer untuk memprediksi dan mengambil keputusan. Dengan mempelajari pola dalam data yang ada, algoritma ini memungkinkan sistem untuk mengolah data baru dengan lebih cepat dan akurat.

Dengan perkembangan algoritma *machine learning*, perangkat ini dapat menyelesaikan berbagai jenis permasalahan dan diterapkan dalam beragam bidang aplikasi untuk mendukung dalam prediksi dan pendukung keputusan. Algoritma *machine learning* memiliki fungsi vital dalam membantu sistem menemukan pola berpikir dengan mempelajari data sebelumnya. Beragam algoritma *machine learning* yang umumnya digunakan dan dikelaskan berdasarkan teknik *machine learning*, yang akan dijelaskan lebih lanjut pada bagian selanjutnya, membantu pemahaman implementasi dan pemilihan algoritma yang sebanding dengan masalah yang dihadapi (Hardini, 2019).

2.4.1 Supervised Learning

Supervised Learning adalah model *Machine Learning* yang mengasah kemampuannya dengan mempelajari data yang telah diberi label ataupun target. Dalam evaluasi, kinerja model ini dinilai berdasarkan ketepatan prediksi terhadap target yang telah ditentukan (Wijaya, 2021).

Untuk melatihnya, *Supervised Learning* memanfaatkan data training yang terdiri dari input data dan target yang diinginkan. Model ini kemudian mengenali pola dalam data *training* untuk membuat prediksi saat dihadapkan pada data uji. Dalam proses evaluasi, model *Supervised Learning* diuji pada *data test*, dan performanya dievaluasi berdasarkan perbandingan hasil prediksi dengan target pada *data test*.

Contoh algoritma yang umum digunakan dalam *Supervised Learning* meliputi klasifikasi untuk memprediksi kategori fitur (seperti Ya/Tidak, Mau/Tidak Mau) dan Regresi untuk memprediksi nilai fitur yang kontinu

(misalnya, harga rumah atau harga saham). Beberapa contoh model yang sering diaplikasikan dalam *Supervised Learning* mencakup *Linear Regression*, *Logistic Regression*, *Random Forest*, *XGBoost*, *K-NN*, dan *SVM*.

2.5 Pengolahan Citra Digital

Pengolahan Citra Digital atau *Digital Image Processing* adalah bagian dari ilmu yang mengkaji cara pembentukan, manipulasi, dan analisis citra untuk mewujudkan informasi yang dapat dipahami oleh manusia. Dalam konteks ini, teknik-teknik dipelajari untuk memanipulasi citra digital guna memberikan pemahaman yang lebih baik tentang informasi yang terkandung di dalamnya. *Digital Image Processing* mengacu pada proses pengolahan citra secara digital menggunakan komputer (Ratna, 2020).

Secara matematis, citra dianggap sebagai fungsi berkelanjutan dengan intensitas cahaya pada sudut pandangan dua dimensi yang kemudian diubah menjadi representasi numerik diskrit untuk dapat diproses oleh komputer. Digitalisasi citra merupakan langkah penting dalam proses ini.

Tujuan utama dari pengolahan citra digital ini yaitu guna meningkatkan kualitas atau karakteristik citra agar dapat dengan mudah diinterpretasi oleh manusia atau komputer. Digitasi adalah tahap penting dalam mengonversi citra menjadi format digital, di mana gambar, teks, atau suara dari objek yang dapat dilihat dikonversi menjadi data elektronik yang dapat disimpan dan diproses untuk berbagai tujuan.

Beberapa teknik pengolahan citra yang umum digunakan adalah *resizing*, *mean subtraction*, dan *scaling* (Dompeipen et al., 2021).

1. *Resizing* mengubah ukuran citra digital, baik untuk memperkecil atau memperbesar dari ukuran asli.
2. *Mean subtraction* digunakan untuk menangani masalah iluminasi pada gambar dengan mengurangi nilai piksel rata-rata dari setiap gambar selama pelatihan.

3. Scaling melibatkan penyesuaian terhadap standar deviasi oleh set data pelatihan, yang memungkinkan pengolahan oleh jaringan *Deep Learning* dan koordinasi skala *bounding box*.

Tahapan-tahapan ini membantu dalam melakukan normalisasi citra masukan dan koordinasi skala yang diperlukan dalam proses penyimpanan serta tampilan citra.

2.6 Python

Diciptakan oleh *Guido Van Rossum* pada tahun 1991, *python* adalah bahasa pemrograman kelas atas yang saat ini sangat diminati. Keunggulan *Python* tidak hanya terletak pada popularitasnya, tetapi juga pada kemampuannya yang serbaguna. Misalnya, *Python* dapat digunakan untuk *Machine Learning* dan *Deep Learning* (Riziq sirfatullah Alfarizi et al., 2023).

Pemilihan *Python* sebagai fokus penelitian ini didasarkan pada beberapa faktor, termasuk sintaksis yang mudah dipahami, ketersediaan *library* yang lengkap, dan dukungan komunitas yang aktif karena sifat open source dari *Python*. Untuk menulis kode *Python*, tersedia berbagai *Integrated Development Environment* (IDE) seperti *VS Code*, *Sublime Text*, *PyCharm*, atau bahkan melalui platform online seperti *Jupyter Notebook* dan *Google Colab*.

Perkembangan *Python* terinspirasi oleh kemajuan bahasa pemrograman ABC pada masa tersebut. Salah satu perbedaan kunci antara *Python* dan bahasa pemrograman lainnya adalah pengembangannya yang melibatkan jutaan programmer, peneliti, dan pengguna dari berbagai latar belakang, bukan hanya dari lingkungan IT, karena *Python* diimplementasikan sebagai proyek *open source*.

Keunggulan *Python* terletak pada interpreter-nya yang memungkinkan eksekusi kode secara cepat dan portabilitasnya yang tinggi karena mendukung multi-platform seperti *Windows*, *Linux*, dan sebagainya. Dengan menggabungkan berbagai paradigma pemrograman prosedural, objek, dan fungsional, *Python* menawarkan kemudahan dalam mengembangkan aplikasi yang kompleks dengan memberikan fleksibilitas tinggi dalam pengembangan perangkat lunak.

Beberapa alasan mengapa *Python* menjadi pilihan utama meliputi kemampuannya untuk beroperasi di berbagai *platform*, sintaksis yang sederhana dan mirip dengan bahasa Inggris, penulisan kode yang lebih ringkas, pelaksanaan program yang cepat melalui *interpreter*, serta dukungan terhadap paradigma pemrograman prosedural, berorientasi objek, dan fungsional (Rahman et al., 2023).

2.7 Sci-kit Learn

Scikit-Learn adalah sebuah library open source untuk *Machine Learning* yang menggunakan *Python* dan dapat diintegrasikan dengan *Data Science*. Salah satu keunggulan utama *Scikit-Learn* adalah kemudahan penggunaannya melalui API yang sederhana dan kinerjanya yang cepat saat melakukan berbagai metrik pada dataset. Selain itu, *Scikit-Learn* dapat berkolaborasi dengan *NumPy* dan *SciPy*, memungkinkan pengguna untuk berinteraksi dengan berbagai *library Python* dengan mudah (Reyvan, 2020).

Fitur-fitur yang disediakan oleh *Scikit-Learn* untuk keperluan *Data Science* mencakup berbagai algoritma seperti Regresi, pengelompokan, *Naive Bayes*, *Decision Tree*, parameter tuning, alat preprocessing data, ekspor/impor model, *pipeline Machine Learning*, serta berbagai algoritma klasifikasi seperti gradien, K-means, mesin dukungan vektor, DBSCAN, dan kemampuan berintegrasi dengan *SciPy* dan *NumPy*.

Scikit Learn didesain berdasarkan beberapa data umum dan pustaka matematika *Python*, sehingga dapat dengan mudah untuk diintegrasikan dengan berbagai *library* lainnya. Salah satu impelentasinya seperti dapat meneruskan array pada *NumPy* dan dataframe pada *Pandas* secara langsung ke algoritma *Machine Learning (ML)*. *Scikit Learn* memiliki dan dapat menggunakan beberapa perpustakaan berikut.

1. *NumPy*: Untuk tugas yang melibatkan matriks, khususnya operasi matematika.
2. *SciPy*: Digunakan untuk komputasi ilmiah dan teknis.
3. *Matplotlib*: Untuk melakukan visualisasi data.

4. Ipython: Console interaktif untuk bahasa *Python*.
5. Sympy: Digunakan untuk matematika *symbolic mathematics* sebagai penyelesaian permasalahan matematika.
6. Pandas: Untuk pengurusan data, manipulasi data, dan analisis data.

Selain itu, *Scikit Learn* juga dapat diintegrasikan dengan *OpenCV (cv2)* untuk pemrosesan citra (*Computer Vision*), serta algoritma klasifikasi seperti *Naive Bayes*, *Random Forest*, dan *K-Nearest Neighbors (KNN)* dari *scikit-learn*. Selain itu, *library* lain seperti *sklearn.model_selection* untuk *train_test_split* (mengubah data latih dan data uji), *sklearn.metrics* untuk *accuracy score*, *clasification_report*, dan *confusion matrix* (untuk data report hasil), *statistics*, *os*, dan *re* juga dapat digunakan bersama dengan *Scikit Learn* (Gifari, 2020).

2.8 Tkinter

Tkinter adalah perpustakaan antarmuka pengguna grafis (GUI) standar *Python*. Dengan *Tkinter*, dapat memudahkan untuk melakukan desain antarmuka pengguna grafis. *Tkinter* memberikan toolkit GUI Tk antarmuka berorientasi objek yang luas. *Tkinter* juga mendukung kontrol dalam program GUI, seperti tombol, label, dan *textbox* (Vayadande et al., 2022).

Tkinter adalah perpustakaan antarmuka pengguna grafis (GUI) standar *Python*. Dengan *Tkinter*, dapat memudahkan untuk melakukan desain antarmuka pengguna grafis. *Tkinter* memberikan *toolkit* GUI Tk antarmuka berorientasi objek yang luas. *Tkinter* memiliki kumpulan *widget* yang kuat yang dapat digunakan untuk dengan cepat membangun antarmuka pengguna yang kompleks. Selain itu, *Tkinter* memungkinkan pemrogram untuk membuat aplikasi berbasis peristiwa, sehingga memungkinkan interaksi dengan pengguna. Berbagai jenis *widget*, seperti tombol, label, kotak teks, dan menu, dapat digunakan dalam *Tkinter* untuk mengembangkan berbagai jenis antarmuka pengguna, termasuk formulir, kotak dialog, dan menu.

Modul *Tkinter* juga mencakup beberapa fungsi untuk membuat antarmuka pengguna grafis (GUI). Fungsi *mainloop()* digunakan untuk memulai perulangan

utama aplikasi dan memproses peristiwa. Setelah semua *widget* telah dibuat, fungsi *mainloop()* dipanggil. Fungsi *geometry()* digunakan untuk menentukan ukuran dan posisi jendela. Fungsi *title()* digunakan untuk mengatur judul jendela. Fungsi *update()* digunakan untuk menyegarkan jendela setelah perubahan dilakukan (Seetha et al., 2023).

2.9 CustomTkinter

CustomTkinter adalah sebuah perpustakaan Python untuk membuat kustomisasi antarmuka pengguna atau disebut dengan *Graphical User Interface (GUI)*. Perpustakaan *CustomTkinter* merupakan ekstensi dari perpustakaan Tkinter yang menambahkan fitur-fitur dan *widget* baru. Tujuannya adalah untuk memudahkan pembangunan antarmuka pengguna (GUI) *Python* yang kompleks dan kuat. Perpustakaan ini mencakup berbagai *widget* dan fungsi yang dapat digunakan untuk membangun antarmuka pengguna secara mudah dan dapat dikustomisasi (Seetha et al., 2023).

CustomTkinter juga menyediakan berbagai alat untuk mengembangkan GUI, seperti *custom widget*, *dialog*, panel, hingga menu kustom. Selain itu, *CustomTkinter* juga memiliki fitur-fitur lanjutan seperti penanganan acara, animasi, dan grafis.

Aplikasi yang menggunakan *CustomTkinter* dapat merespons masukan pengguna seperti klik mouse dan tekanan tombol dengan menggunakan penanganan acara. Aplikasi juga dapat menggunakan animasi untuk menciptakan efek visual dinamis seperti memudar masuk dan keluar teks atau gambar. Grafis memungkinkan aplikasi menghasilkan gambar yang dipersonalisasi seperti grafik dan diagram.

2.10 Jupyter Notebook

Jupyter Notebook adalah aplikasi web gratis yang merangkum tiga bahasa pemrograman, yaitu *Julia (Ju)*, *Python (Py)*, dan *R*. Aplikasi ini menjadi pilihan utama bagi para data scientist, memungkinkan mereka membuat dan berbagi dokumen yang mencakup kode, hasil perhitungan, visualisasi, dan teks. Kehadiran

ketiga bahasa pemrograman ini menjadi elemen kunci bagi seorang data scientist yang membutuhkan fleksibilitas (Driscoll, 2022).

Secara umum, Jupyter Notebook berperan sebagai alat bantu bagi data scientist dalam mengembangkan narasi komputasi. Narasi komputasi tersebut bertugas menjelaskan makna dari data yang diproses dan memberikan wawasan yang mendalam terkait data tersebut.

Kelebihan lain dari Jupyter Notebook terletak pada kemudahan dalam menulis dan berbagi teks serta kode di dalamnya, menjadikannya sangat cocok untuk kegiatan kolaborasi. Aplikasi ini membantu menyederhanakan proses kerja sama antara insinyur dan *data scientist*, serta memfasilitasi kolaborasi di antara para *data scientist*, *data researcher*, dan *data engineer*.

2.11 Visual Studio Code

Menurut (Setiawan, 2022), Visual Studio Code adalah sebuah editor kode sumber yang sangat populer dan dapat disesuaikan secara luas. Dikembangkan oleh Microsoft, alat ini tersedia secara gratis dan mendukung berbagai sistem operasi. VS Code menawarkan beragam fitur canggih seperti debugging interaktif, integrasi langsung dengan sistem kontrol versi Git dan GitHub, serta kemampuan untuk menyoroti sintaksis berbagai bahasa pemrograman. Selain itu, fitur penyelesaian kode secara cerdas, cuplikan kode, dan pembaruan kode secara otomatis sangat membantu meningkatkan produktivitas pengembang.

Fleksibilitas VS Code semakin ditingkatkan dengan kemampuan untuk menyesuaikan tampilan, pintasan keyboard, dan preferensi lainnya. Pengguna juga dapat memperluas fungsionalitas VS Code dengan instal berbagai ekstensi yang tersedia di marketplace.

2.12 Confusion Matrix

Hasil pengujian dari algoritma yang diimplementasikan, dievaluasi dengan menghitung parameter kinerja dengan deksripsi klasifikasi dihitung merujuk pada Confusion Matrix .

Confusion Matrix adalah metode yang digunakan untuk mengevaluasi performa suatu sistem klasifikasi. Secara mendasar, *Confusion Matrix* memberikan informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil yang seharusnya (Karsito & Susanti, 2019).

Dalam pengukuran performa menggunakan *Confusion Matrix*, terdapat empat terminologi yang merepresentasikan hasil proses klasifikasi. Keempat terminologi tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai *True Negative* (TN) mencerminkan jumlah data negatif yang berhasil terdeteksi dengan tepat, sementara *False Positive* (FP) merujuk pada data negatif yang keliru terdeteksi sebagai positif (Prakasa et al., 2023).

2.13 Random Forest

Diperkenalkan pertama kalinya oleh Leo Breiman pada tahun 2001, *Random Forest* (RF) merupakan sebuah teknik yang telah terbukti meningkatkan tingkat akurasi dengan cara menghasilkan atribut secara acak untuk setiap nodenya. Metode ini terdiri dari sejumlah pohon keputusan, di mana setiap pohon bertugas untuk mengklasifikasikan data ke dalam kelas tertentu. Meskipun proses pembuatan pohon keputusan pada RF mirip dengan *Classification and Regression Tree* (CART), namun perlu diperhatikan bahwa RF tidak melibatkan tahap pemangkasan (*pruning*) (Suci Amaliah et al., 2022).

Dalam konstruksi pohon keputusan pada *Random Forest* (RF), prosesnya serupa dengan yang dilakukan pada CART, namun tanpa adanya tahap pemangkasan (*pruning*). Indeks Gini digunakan untuk memilih fitur di setiap simpul internal dari pohon keputusan. Perhitungan nilai Indeks Gini dapat diamati pada Persamaan (2.2).

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \quad (2.2)$$

Dengan P_i mewakili frekuensi relatif kelas atau proporsi data dari kelas C_i dalam kumpulan data, di mana C_i adalah jumlah total kelas untuk $i = 1, \dots, c-1$, dan c adalah jumlah kelas yang telah ditentukan. Ketika melakukan evaluasi pembagian pada fitur k menjadi subset S_i , langkahnya melibatkan perhitungan jumlah sampel yang termasuk dalam kelas C_i .

Evaluasi pembagian ini diukur dengan menghitung jumlah perubahan Indeks Gini dari subset yang dihasilkan. Ketika melakukan evaluasi pembagian pada fitur k menjadi subset S_i , kualitas pembagian diukur dengan menghitung *Gini impurity* untuk subset yang dihasilkan.. Rumus untuk menghitung ini dapat dilihat pada Persamaan (2.3).

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n} \right) Gini(S_i) \quad (2.3)$$

Di mana n_i adalah jumlah sampel yang terdapat dalam subset S_i setelah dilakukan pembagian, dan n merupakan total jumlah sampel di node yang sedang dievaluasi.

Dalam Random Forest, kesalahan generalisasi bergantung pada kekuatan individu pohon keputusan dan ketergantungan antar pohon. Sehingga terdapat *Error Generalization*. Kumpulan pohon keputusan, dilambangkan sebagai $\{h(x, \Theta_k), k= 1, \dots\}$, di mana $\{\Theta_k\}$ adalah vektor acak yang independen dan identik didistribusikan (iid), dan setiap pohon memilih kelas mayoritas suara. Dalam *Random Forest* (RF), karena terdapat batas atas untuk kesalahan generalisasi, akan bergantung pada dua parameter. Parameter ini mengukur seberapa kuat pengklasifikasian secara individual dan seberapa ketergantungannya satu sama lain. Fungsi margin untuk RF dapat dilihat pada Persamaan (2.4).

$$mr(X, Y) = P_{\theta}(h(X, \theta) = Y) - \max_{j \neq Y} P_{\theta}(h(X, \theta) = j) \quad (2.4)$$

Dimana $h(X, \theta)$ adalah prediksi oleh pohon keputusan, Y adalah label kelas yang benar, dan j adalah kelas lain selain Y . Dan rumus himpunan pada $\{h(X, \theta)\}$, dapat dilihat pada Persamaan (2.5).

$$s = Ex, mr(X, Y) \quad (2.5)$$

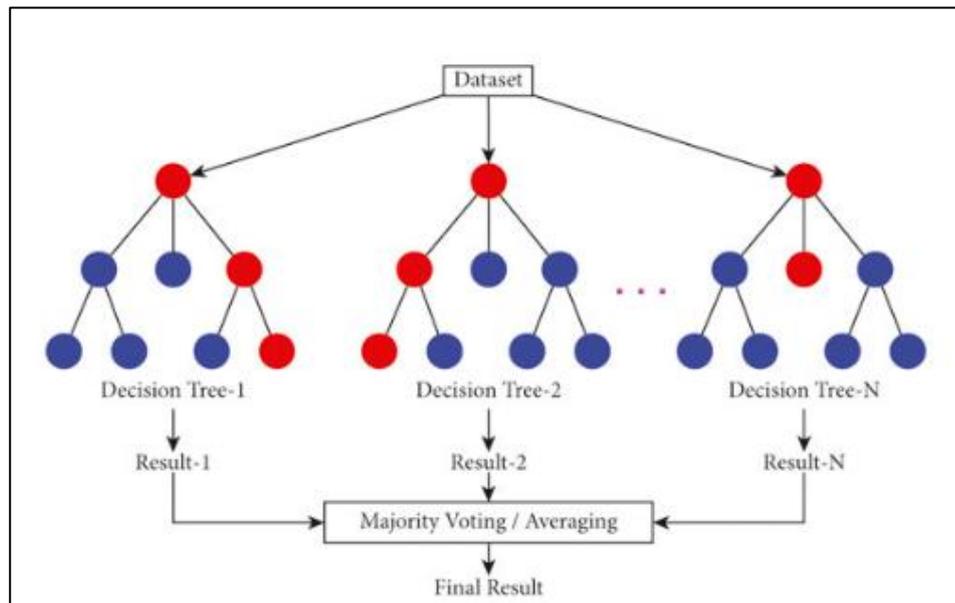
Dengan asumsi $s \geq 0$, dan menggunakan ketidaksetaraan *Chebyshev* serta turunan varians dari fungsi margin pada metode RF. Rumus untuk menghitung batas atas generalisasi dapat dilihat pada Persamaan (2.6).

$$PE \leq \frac{\rho(1 - s^2)}{s^2} \quad (2.6)$$

Di mana $\bar{\rho}$ merujuk pada nilai korelasi rata-rata antara pohon-pohon dalam hutan, dan s adalah fungsi margin.

Kesimpulannya, *Gini impurity* digunakan untuk memilih split terbaik di pohon keputusan dalam *Random Forest*, Evaluasi kualitas split melibatkan perhitungan *Gini impurity* untuk subset yang dihasilkan, serta bagaimana *Random Forest* dapat mengurangi kesalahan generalisasi melalui kekuatan kolektif pohon-pohon keputusan dengan korelasi yang rendah antar pohon yang membantu meningkatkan generalisasi.

Konsep mengenai cara kerja metode *Random Forest* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Cara Kerja Metode Random Forest

Dimana cara kerja metode *Random Forest* dapat dijabarkan dalam langkah-langkah berikut (Trivusi, 2022).

1. Algoritma memilah sampel secara acak dari dataset yang tersedia.
2. Selanjutnya, untuk per-sampel yang dipilih, algoritma membuat bentuk pohon keputusan. Hasil prediksi akan didapatkan dari setiap pohon keputusan yang dibuat.
3. Kemudian, dilakukan proses pemungutan suara untuk hasil prediksi dari setiap pohon. Jika masalahnya adalah klasifikasi, pemungutan suara dilakukan dengan menggunakan modus (nilai yang paling sering tampil), sementara untuk masalah regresi, digunakan nilai mean (rata-rata).
4. Prediksi akhir dari algoritma dipilih berdasarkan hasil pemungutan suara yang paling banyak (vote terbanyak).

2.14 Unified Modeling Language (UML)

Unified Modeling Language (UML) yaitu bahasa visual yang dipakai untuk membuat model, merancang alur diagram, dan mendokumentasikan sistem perangkat lunak. Pertama kali diperkenalkan pada tahun 1997 oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh, UML telah menjadi standar de facto dalam

industri perangkat lunak. UML memiliki kapabilitas yang luas, memungkinkan penciptaan model untuk berbagai jenis aplikasi perangkat lunak yang dapat beroperasi di berbagai sistem operasi, jaringan, dan ditulis dalam berbagai bahasa pemrograman.

Bahasa pemodelan visual ini mencakup notasi grafis yang merepresentasikan berbagai aspek sistem, termasuk struktur, perilaku, interaksi antar objek, dan lingkungan operasional. UML terdiri dari berbagai diagram, seperti use case diagram, class diagram, activity diagram, dan sequence diagram, yang masing-masing menggambarkan aspek tertentu dari sistem secara visual (Faulina, 2023).

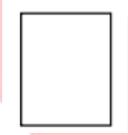
Beberapa pakar berpendapat bahwa *Unified Modeling Language* (UML) merupakan alat pemrograman perangkat lunak yang dipergunakan untuk konstruksi sistem informasi. Salah satu diagram UML disertakan dalam penjelasan berikut.

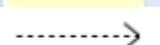
2.14.1 Use Case Diagram

Diagram Usecase memvisualisasikan data tentang Kasus Penggunaan (Use Case) dan Pihak yang Terlibat (Aktor). Pihak yang terlibat dapat berupa individu, perangkat, atau sistem lain yang berinteraksi dengan sistem yang telah dibangun. Dari perspektif pengguna, dapat dilihat pada Tabel 2.2 menunjukkan notasi dari fungsi atau kebutuhan yang harus dipenuhi oleh sistem.

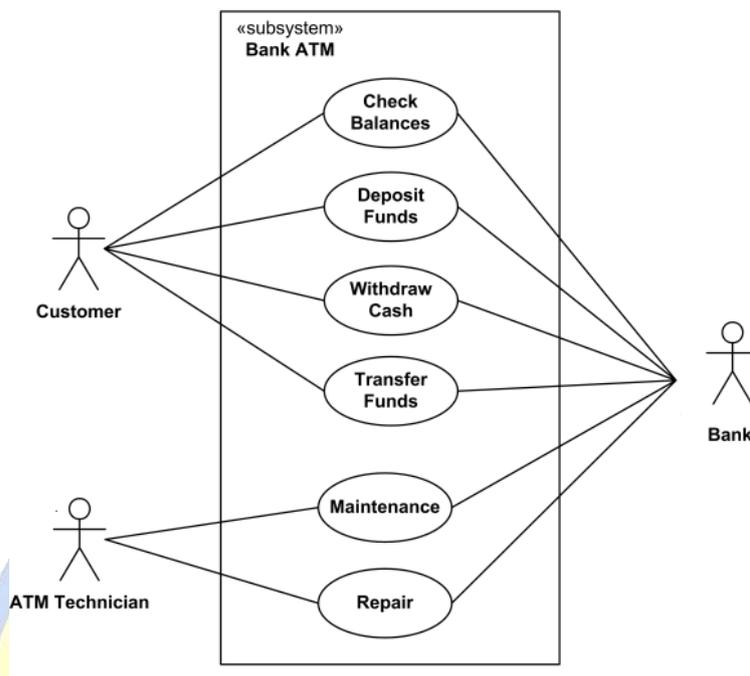
Tabel 2.1 Notasi UML: Use Case Diagram

No	SIMBOL	NAMA	KETERANGAN
1		<i>Actor</i>	Menentukan peran khusus yang dimainkan oleh pengguna saat terlibat dalam use case.

2		<i>Association</i>	Menunjukkan peran atau fungsi objek dalam konteks hubungan asosiasi (antara objek satu dengan objek lainnya).
3		<i>Use Case</i>	Objek untuk penempatan berupa penjelasan mengenai urutan langkah-langkah sistem yang menampilkan hasil terukur untuk seorang aktor.
4		<i>Collaboration</i>	Objek yang mengandung kumpulan aturan dan elemen lain yang bekerja bersama untuk memberikan perilaku yang lebih besar daripada jumlah individu dan elemennya agar menciptakan kombinasi optimal.
5		<i>System</i>	Menjelaskan isi yang memperlihatkan sistem dalam ruang lingkup yang terbatas.
6		<i>Note</i>	Objek yang memuat elemen fisik yang hadir ketika aplikasi berjalan dan mencerminkan sumber daya komputasi.

7		<i>Include</i>	Hubungan yang menunjukkan bahwa satu use case menggabungkan fungsionalitas dari use case lain secara jelas.
8		<i>Extend</i>	Menunjukkan bahwa dalam suatu use case yang melakukan “extend,” terdapat kemungkinan untuk menyertakan langkah-langkah tambahan dalam rangkaian kerjanya apabila kondisi tertentu dipenuhi.
9		<i>Dependency</i>	Keterkaitan di mana modifikasi yang berlaku pada elemen independen akan berpengaruh pada elemen dependen yang tergantung padanya.
10		<i>Generalization</i>	Keterkaitan di mana objek turunan berbagi tingkah laku dan struktur data dari objek induk yang berada di atasnya.

Berikut adalah contoh gambaran dari implementasi Diagram *Use Case*, dapat dilihat pada Gambar 2.2 (Ahaddin, 2015).



Gambar 2.2 Contoh UML: Use Case Diagram

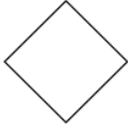
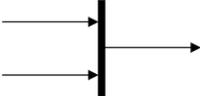
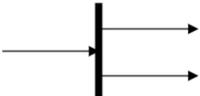
2.14.2 Activity Diagram

Diagram aktivitas mengilustrasikan berbagai aliran kegiatan dalam sistem yang sedang direncanakan, menunjukkan bagaimana setiap aliran dimulai, keputusan yang mungkin diambil, dan cara aliran tersebut berakhir. Sebuah kegiatan dapat diimplementasikan oleh satu use case atau lebih. Kegiatan mencerminkan proses yang sedang berlangsung, sedangkan use case memperlihatkan cara aktor memanfaatkan sistem untuk menjalankan kegiatan (Muhammad Rizky, 2019). Notasi dalam diagram ini dapat dilihat pada Tabel 2.3.

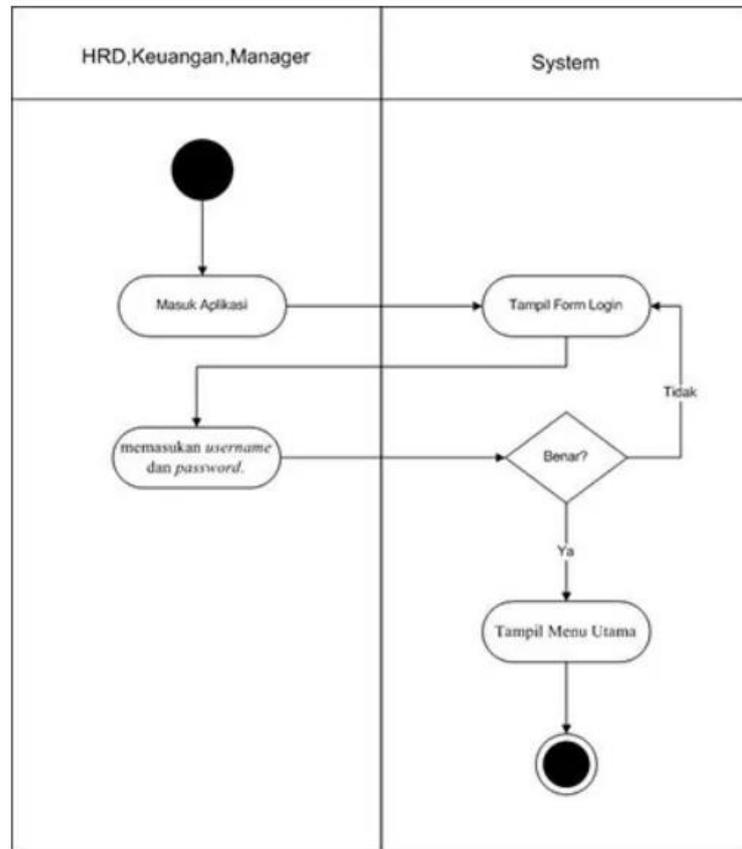
Tabel 2.2 Notasi UML: Activity Diagram

No	SIMBOL	NAMA	KETERANGAN
1		<i>Initial State Node</i>	Menentukan peran khusus yang dimainkan oleh pengguna saat terlibat dalam use case.

2		<p><i>Final State Node</i></p>	<p>Keadaan akhir yang dicapai oleh sistem, sebuah diagram aktivitas yang memiliki suatu kondisi akhir dan dapat memiliki lebih dari satu keadaan akhir.</p>
3		<p><i>Activity State Node</i></p>	<p>Kegiatan yang dijalankan oleh sistem untuk menunjukkan interaksi antar kelas yang terlibat.</p>
4		<p><i>Action State Node</i></p>	<p>Keadaan dari suatu sistem yang mencerminkan pelaksanaan suatu tindakan.</p>
5		<p><i>Association/Line Connector</i></p>	<p>Digunakan untuk mengaitkan satu simbol dengan simbol-simbol lainnya dan menunjukkan aktivitas yang akan berlangsung setelah suatu aktivitas tertentu.</p>
5		<p><i>Swimlane</i></p>	<p>Diagram yang menggambarkan alur proses, mencerminkan interaksi antara berbagai bagian yang berbeda.</p>

5		<i>Decision Node</i>	Menjelaskan isi yang memperlihatkan sistem dalam ruang lingkup yang terbatas.
6		<i>Join Node</i>	Digunakan untuk menggambarkan beberapa kegiatan yang disatukan.
7		<i>Fork Node</i>	Dipakai untuk menunjukkan kegiatan yang berjalan secara bersamaan (bercabang).

Berikut adalah contoh gambaran implementasi penggunaan Diagram Aktivitas, yang dapat dilihat pada Gambar 2.3.



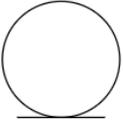
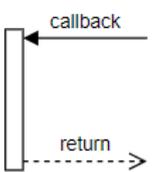
Gambar 2.3 Contoh UML: Activity Diagram

2.14.3 Sequence Diagram

Diagram sekuen mengilustrasikan hubungan antara objek-objek dalam dan sekitar sistem, termasuk pengguna, dengan menunjukkan pesan-pesan yang disusun berdasarkan urutan waktu. Struktur diagram sekuen melibatkan dimensi vertikal yang mewakili waktu dan dimensi horizontal yang menampilkan beberapa objek yang terkait. Diagram ini sering digunakan untuk merinci skema atau langkah-langkah yang dijalankan menjadi bagian dari suatu kejadian, menghasilkan keluaran tertentu. Notasi pada diagram ini tercantum dalam Tabel 2.4.

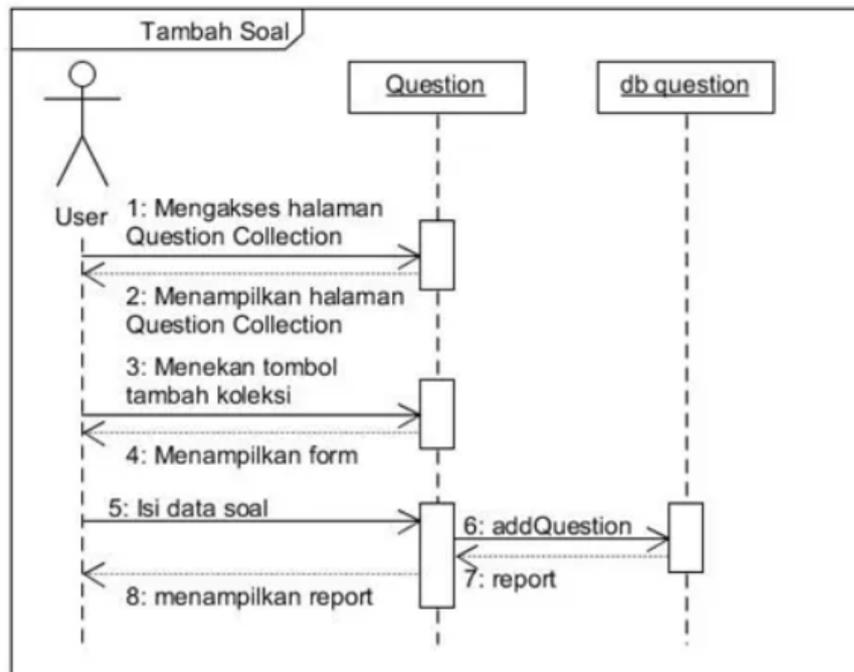
Tabel 2.3 Notasi UML: Sequence Diagram

No	SIMBOL	NAMA	KETERANGAN

1		<i>Actor</i>	mengilustrasikan individu yang sedang terlibat dalam interaksi dengan sistem.
2		<i>Entity Class</i>	Mengilustrasikan koneksi tindakan yang akan dilaksanakan.
3		<i>Boundary Class</i>	Menangani interaksi antara lingkungan sistem.
4		<i>Control Class</i>	Menggambarkan koneksi antara batas dengan tabel dan bertanggung jawab pada objek yang mengandung logika.
5		<i>Callback</i>	Menerima kembali pesan.

6		<i>Self Call</i>	Mengirim pesan kepada diri sendiri.
7		<i>Life Line</i>	Menggambarkan titik awal dan akhir suatu instruksi.
8		<i>Message</i>	Menggambarkan transmisi suatu instruksi.
9		<i>Frame</i>	Digunakan untuk mengorganisir kegiatan atau pesan-pesan yang terjadi dalam suatu lingkup tertentu atau sebagai tanggapan terhadap suatu peristiwa.

Berikut adalah contoh dari implementasi Diagram Sekuen, yang dapat dilihat pada Gambar 2.4.



Gambar 2.4 Contoh UML: Sequence Diagram

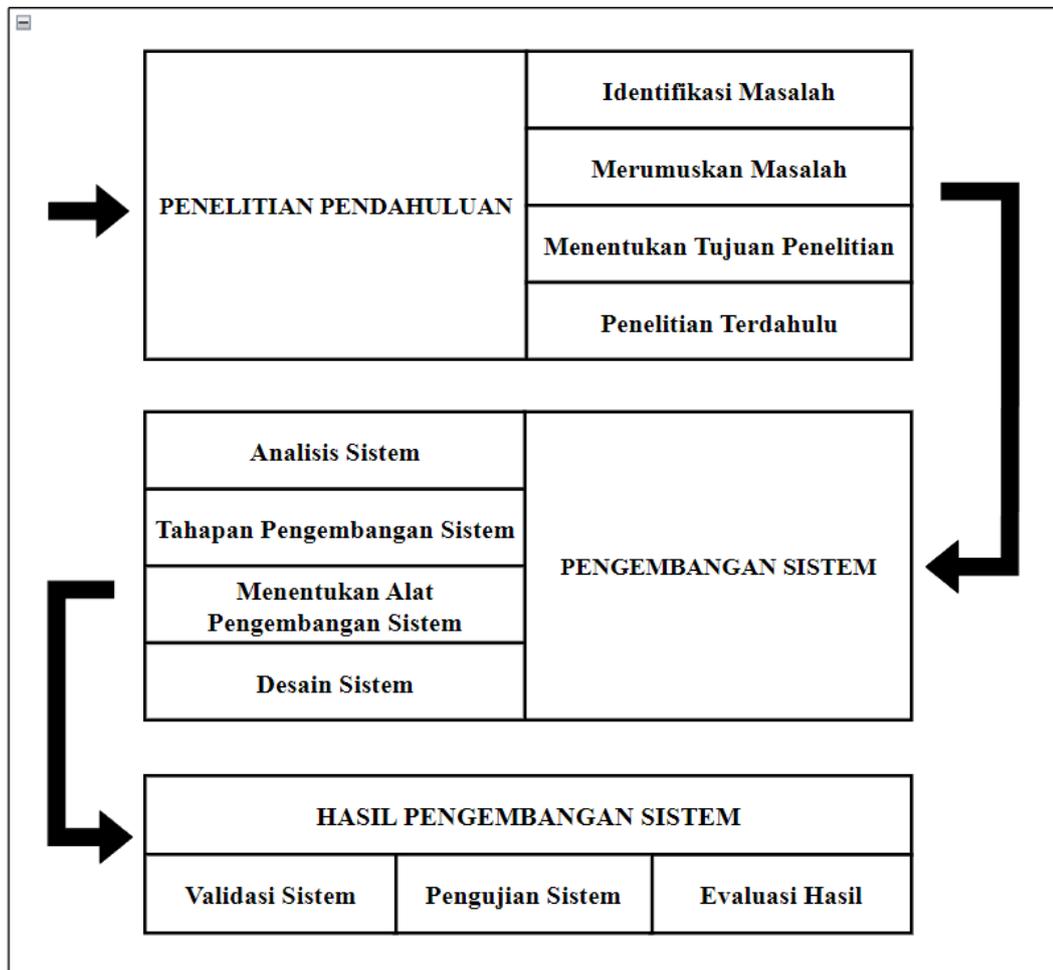
Confusion Matrix ini memungkinkan untuk menghitung metrik-metrik penting seperti akurasi, presisi, dan *recall*. Akurasi memberikan gambaran umum tentang kinerja model, namun tidak selalu cukup untuk kasus-kasus di mana kelas data tidak seimbang. Presisi lebih fokus pada kualitas prediksi positif, yaitu seberapa sering prediksi positif kita benar. Di sisi lain, *recall* mengukur kemampuan model dalam mengidentifikasi semua contoh positif yang sebenarnya ada.

BAB III

METODE PENELITIAN

3.1 Metodologi Penelitian

Pada penelitian ini terdapat serangkaian tahapan yang terstruktur untuk mengimplementasikan metode algoritma *Random Forest* dan metode ekstraksi fitur *Gray Level Co-occurrence Matrix (GLCM)* dalam identifikasi jenis kayu. Dimulai dari tahap pendahuluan, dimana dilakukan pencarian beberapa masalah yang ada pada studi litelatur, sehingga berdasarkan studi litelatur atau penelitian terdahulu yang sudah dipelajari, maka dapat ditentukan identifikasi masalah, rumusan masalah, hingga tujuan penelitian. Tahap kedua yaitu tahap pengembangan sistem, dimana pada tahap ini melakukan analisis sistem baik masalah yang sedang berjalan, analisis sistem yang akan dibangun, hingga menentukan alat pengembangan sistem untuk membantu dalam pembangunan sistem aplikasi ini, tahapan terakhir pada tahap pengembangan sistem yaitu mendesain sistem berupa UML, dan melakukan tahap validasi model untuk melakukan validasi model yang telah dilatih dan dibangun, dimana validasinya dapat dilihat berdasarkan *performance matrix* dan visualisasi *confusion matrix* yang dihasilkan pada tahap ini. Tahap terakhir pada metode penelitian yaitu pengujian sistem, dimana pengguna akan melakukan pengujian terhadap sistem yang dibangun, dalam hal ini model yang telah divalidasi akan dipanggil pada pengujian dengan aplikasi, lalu proses pengujian juga akan menggunakan lima citra acak, dimana hasil keluaran pada saat tahap pengujian ini berupa jenis yang diprediksi oleh model dan probabilitas per jenis. Langkah-langkah metodologi penelitian yang digunakan dalam penelitian ini, dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian

3.2 Penelitian Pendahuluan

Penelitian pendahuluan adalah tahap perencanaan yang melibatkan identifikasi masalah penelitian berdasarkan tinjauan penelitian sebelumnya terkait proses identifikasi citra kayu. Berikut adalah pemetaan untuk tahap penelitian pendahuluan.

3.2.1 Pemetaan Identifikasi, Rumusan Masalah, dan Tujuan

Berikut adalah pemetaan yang mencakup identifikasi masalah, rumusan masalah, dan tujuan penelitian dari studi ini, yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Pemetaan Penelitian Pendahuluan

Bab Pendahuluan	Keterangan	Detail
-----------------	------------	--------

Identifikasi Masalah	Keterbatasan Teknologi	Keterbatasan dalam penerapan teknologi untuk identifikasi kayu rentan terhadap kesalahan dan memerlukan waktu lama, menghambat produktivitas industri kayu.
	Proses Validasi	Diperlukan proses validasi komprehensif untuk memastikan efektivitas model identifikasi kayu dalam aplikasi nyata.
Rumusan Masalah	Teknologi Identifikasi	Bagaimana mengatasi keterbatasan alat bantu manual dengan mengadopsi teknologi Machine Learning, Computer Vision, dan metode Ekstraksi Fitur untuk identifikasi citra kayu?
	Validasi Model	Bagaimana melakukan validasi model yang efektif untuk memastikan akurasi tinggi dan keandalan

		dalam mengidentifikasi jenis kayu?
Tujuan Penelitian	Pengembangan Sistem	Mengembangkan sistem identifikasi citra kayu menggunakan Machine Learning, khususnya algoritma Random Forest, dan metode ekstraksi fitur GLCM untuk meningkatkan akurasi dan efisiensi.
	Evaluasi Kinerja	Menguji dan mengevaluasi kinerja model yang dikembangkan untuk mempercepat proses identifikasi jenis kayu.

3.2.2 Pemetaan Penelitian Terdahulu

Pemetaan penelitian sebelumnya dilakukan dengan menelaah jurnal berkualitas dan relevan dengan topik penelitian ini. Berikut adalah pemetaan dari penelitian terdahulu yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Pemetaan Penelitian Terdahulu

No	Nama Peneliti	Topik Penelitian	Hasil Penelitian
1	(Prakasa et al., 2023)	Identifikasi Kayu dengan CNN pada Citra <i>Cross-Section</i>	Model CNN yang dikembangkan menunjukkan akurasi yang baik, dengan F1 score mencapai 0.90 untuk lima spesies kayu,

2	(Anggiratih & Putra, 2019)	Identifikasi Kapal pada Citra Satelit dengan <i>Random Forest</i> .	Penggunaan metode RF didapatkan hasil akurasi sebesar 99,0%.
3	(Purwanto et al., 2023)	Implementasi <i>Random Forest</i> untuk Pemetaan Hutan Mangrove.	Hasil pengolahan data dan uji akurasi menunjukkan bahwa hasil klasifikasi dari algoritma RF adalah 99,12%.
4	(Alda Putri Utami et al., 2021)	Menerapkan Metode <i>Random Forest</i> untuk Mengenali Logo Kendaraan	Sistem yang dikembangkan menggunakan <i>Random Forest</i> ini dapat mengenali logo kendaraan, dimana hasil pengujian menunjukkan akurasi tertinggi mencapai 88,89%.
5	(Nuraini, 2022)	Klasifikasi Citra Jenis Kapasitor Menggunakan Algoritma <i>K-Nearest Neighbor</i> (KNN)	Berdasarkan hasil pengujian, model KNN yang dikembangkan mampu menghasilkan nilai akurasi rata-rata sebesar 82,50%.

3.3 Pengembangan Sistem

Pada tahap pengembangan sistem ini, dilakukan analisis menyeluruh, mulai dari sistem yang ada hingga sistem yang diusulkan. Proses ini mencakup tahapan pengembangan sistem dengan metode Random Forest dan GLCM, pemilihan alat pengembangan sistem, serta desain sistem yang meliputi perancangan UML dan Mockup. Penjelasan rinci dapat diuraikan sebagai berikut.

3.3.1 Analisis Sistem

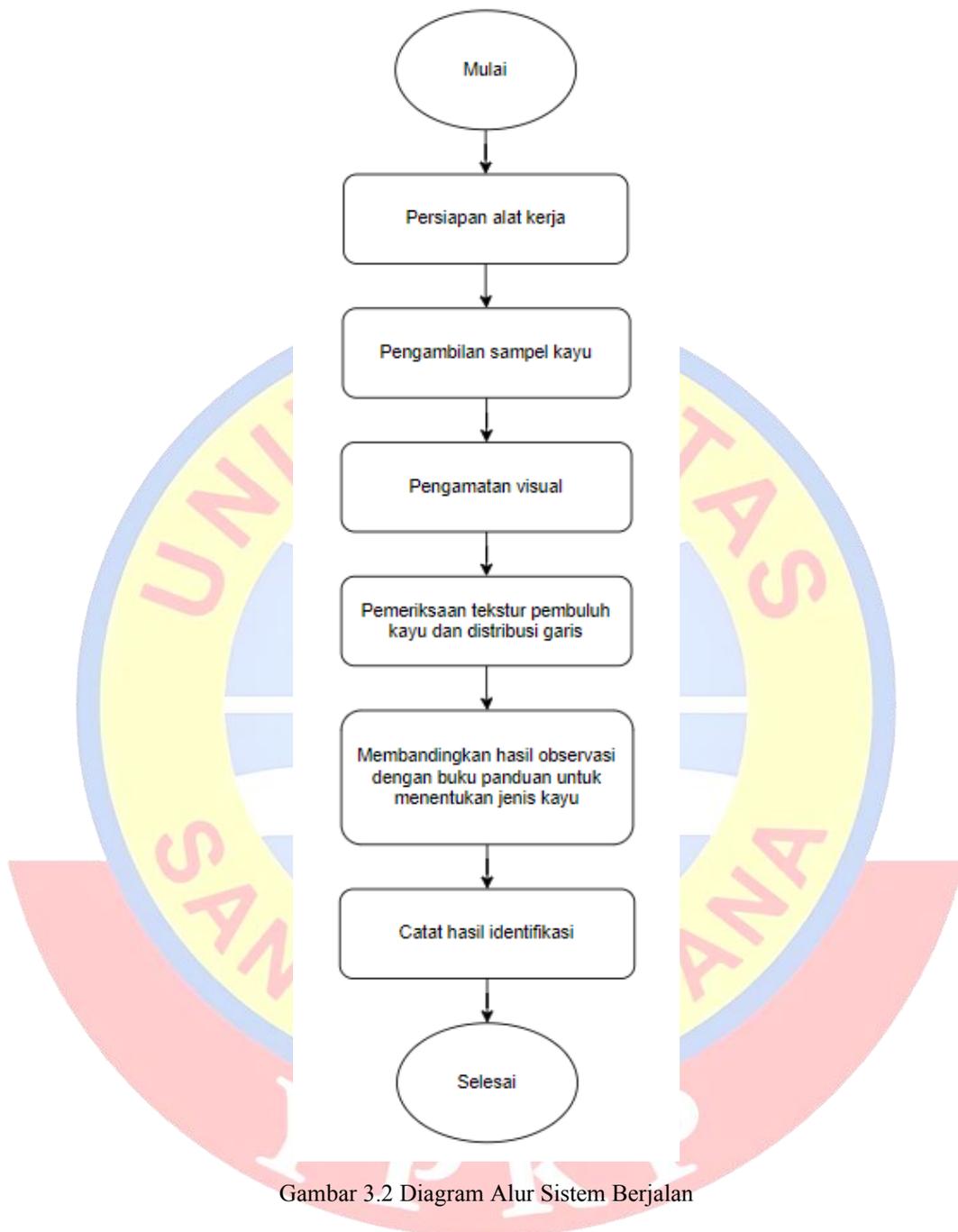
Analisis sistem merupakan proses pemaparan suatu informasi yang lengkap dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan

hingga kebutuhan yang ada. Melalui analisis ini, diharapkan dapat diusulkan berbagai perbaikan dari peluang yang ada dan yang diperlukan untuk meningkatkan kinerja dan efektivitas sistem tersebut.

3.3.1.1 Analisis Masalah dan Sistem yang Berjalan

Menentukan jenis kayu seperti pada pohon di hutan dapat dilakukan secara langsung dengan mudah melalui pemeriksaan daun, batang, ataupun buahnya. Ketika pohon ditebang, mengidentifikasi jenis kayunya menjadi sulit karena harus menggunakan beberapa cara, salah satunya ditentukan berdasarkan cirinya. Ciri pada kayu yang dapat dilakukan pengamatan berdasarkan tekstur meliputi susunan distribusi garis vena hingga bentuk pada pembuluh kayunya yang hanya dapat terlihat jelas dengan bantuan alat seperti mikroskop. Memiliki kemampuan untuk mengidentifikasi hanya dapat diperoleh melalui latihan yang panjang dan berulang.

Proses identifikasi kayu secara manual dimulai dengan persiapan alat yang memadai seperti penggaris, gergaji, dan alat ukur. Langkah pertama adalah pengambilan sampel kayu dari pohon yang telah ditebang, dengan ukuran potongan yang memadai untuk pemeriksaan. Selanjutnya, petugas melakukan pengamatan visual terhadap warna permukaan kayu dan tekstur, untuk mencatat ciri-ciri umum seperti kekasaran atau kehalusan permukaan. Kemudian, ahli anatomi kayu memeriksa pola serat kayu di sepanjang potongan, serta mengamati distribusi garis dan pembuluh vena yang ada pada permukaan kayu. Petugas kemudian membandingkan panduan manual untuk membandingkan ciri-ciri yang telah diamati dengan informasi tentang jenis-jenis kayu. Berdasarkan hasil observasi dan perbandingan tersebut, petugas menentukan jenis kayu yang sesuai lalu mencatat hasil identifikasinya. Jika petugas yang melakukan identifikasinya belum terampil, prosesnya akan memakan waktu lebih lama dan tentunya dapat meningkatkan biaya. Sehingga dibutuhkan suatu sistem yang dapat mengatasi hal tersebut. Proses identifikasi secara manual dapat dijelaskan dalam diagram *flowchart* yang dapat dilihat pada Gambar 3.2.

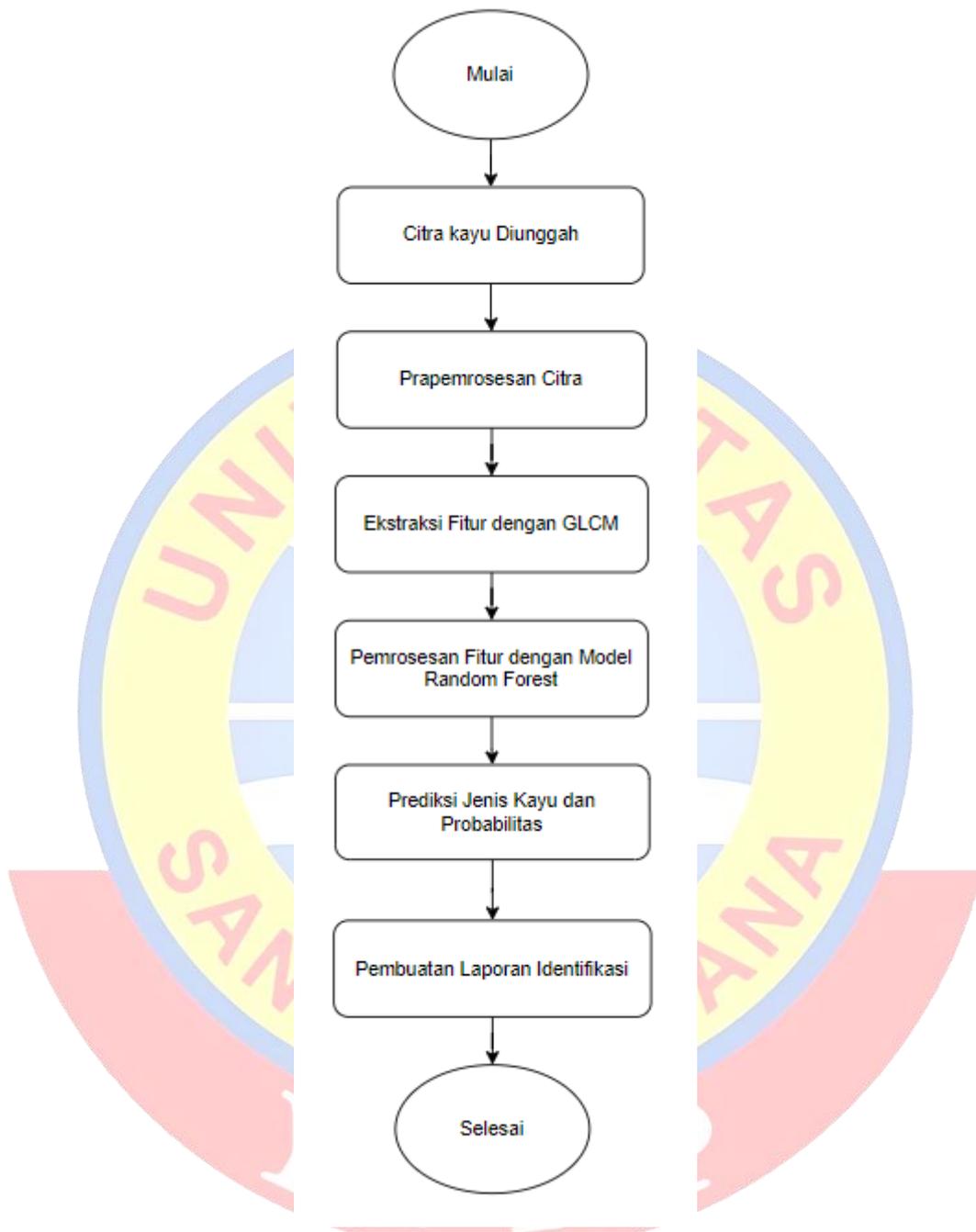


3.3.1.2 Sistem yang Diusulkan

Sistem yang diusulkan agar dapat digunakan oleh masyarakat khususnya petugas di industri kayu yaitu sistem yang bisa mengidentifikasi kayu secara cepat dan efektif dengan meningkatkan akurasi dan efisiensi dalam menentukan jenis kayu dari citra kayu yang diberikan. Sistem ini dimulai ketika citra kayu diunggah ke dalam sistem. Setelah citra kayu diunggah,

sistem akan memproses citra terlebih dahulu dengan preprocessing data, lalu dilanjutkan dengan proses menggunakan teknik *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur tekstur dari citra tersebut. Jika fitur tekstur yang diekstraksi sudah siap, sistem akan menggunakan model Random Forest yang telah dilatih sebelumnya dengan dataset citra dari 5 jenis kayu untuk memprediksi jenis kayu berdasarkan fitur yang dihasilkan. Sistem akan memberikan hasil prediksi jenis kayu yang paling sesuai beserta probabilitas kepercayaannya. Jika model tidak dapat memberikan prediksi dengan tingkat kepercayaan yang tinggi atau jika citra kayu yang diunggah tidak jelas, sistem akan meminta pengguna untuk mengunggah citra tambahan atau memberikan data lebih lanjut untuk meningkatkan akurasi prediksi. Setelah jenis kayu diprediksi, sistem akan menghasilkan laporan yang merangkum hasil identifikasi, termasuk jenis kayu yang diprediksi dan probabilitasnya. Laporan ini kemudian dapat digunakan untuk dokumentasi atau keperluan lainnya.

Sistem ini dirancang untuk meningkatkan efisiensi dan akurasi proses identifikasi kayu, memastikan bahwa jenis kayu dapat diidentifikasi dengan tepat dari citra, serta mengurangi ketergantungan pada keterampilan manual petugas. Proses sistem yang diusulkan dapat dilihat pada Gambar 3.3.

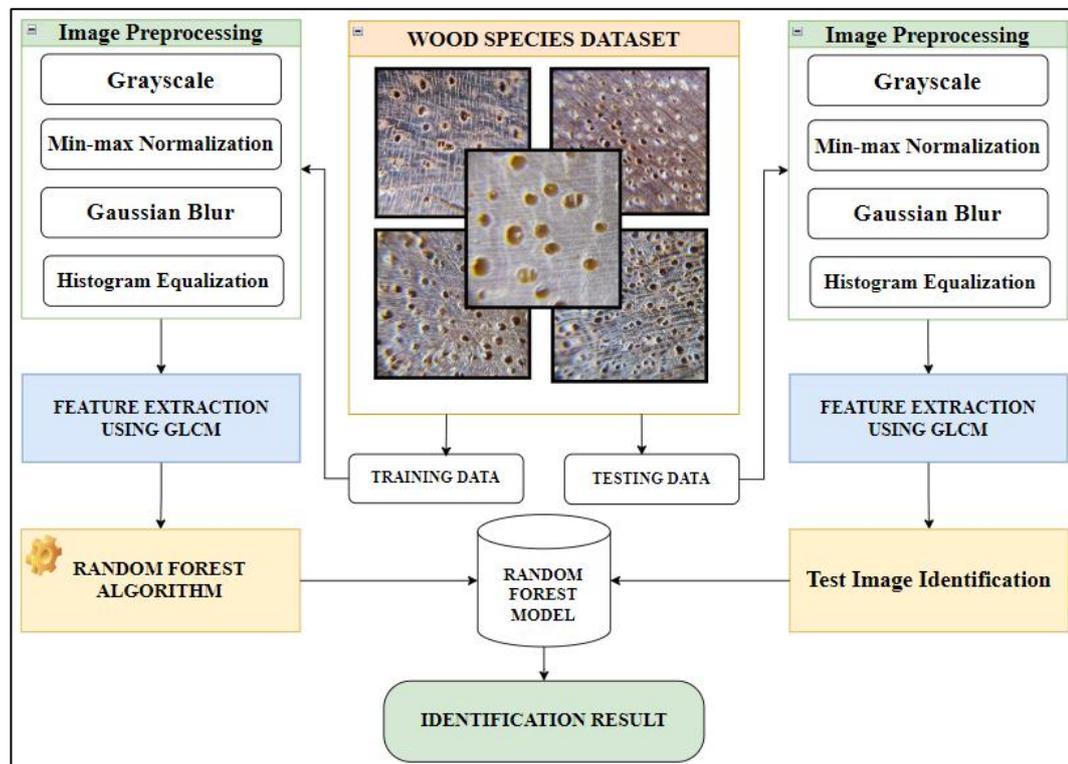


Gambar 3.3 Diagram Alur Sistem yang Diusulkan

3.3.2 Tahapan Pengembangan Sistem

Sesuai dengan pembahasan sistem yang akan diusulkan, maka sistem yang diusulkan dalam identifikasi jenis kayu adalah sistem yang diimplementasikan menggunakan metode algoritma *Random Forest* agar dapat mengoptimalkan hasil

prediksi yang dilakukan dengan akurasi yang tinggi. Tahapan berupa sistem yang diusulkan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tahapan Pengembangan Sistem

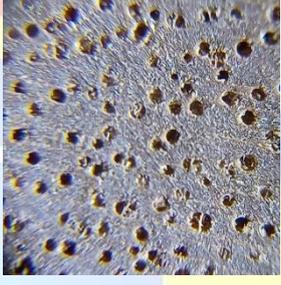
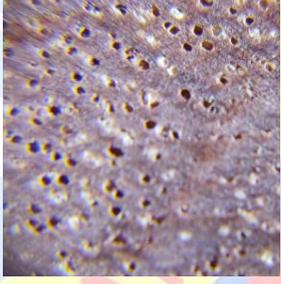
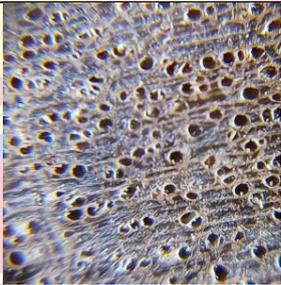
Berikut adalah penjelasan yang lebih lengkap dan terstruktur.

3.3.2.1 Pengumpulan Data

Proses pengumpulan data dimana data yang digunakan dalam penelitian ini berupa citra-citra kayu makroskopik yang telah diperoleh di tempat koleksi kayu yaitu Xylarium Indonesia, di Jl. Cibinong, Bogor, Jawa Barat., dan tempat penelitian BRIN KST Samaun Samadikun, di Jl. Sangkuriang, Bandung, Jawa Barat. Pengumpulan data melibatkan observasi langsung terhadap kayu dengan melakukan pemotretan menggunakan alat bantu makroskopik dan kamera pada *smartphone*, serta dokumentasi koleksi-koleksi berdasarkan data citra yang telah ada sebelumnya, terdapat 5 jenis citra kayu yang akan digunakan pada penelitian ini. Setiap sampel kayu direpresentasikan dalam bentuk citra digital.

Berikut adalah contoh sampel citra kayu, dapat dilihat pada Tabel 3.3.

Tabel 3.3 Sampel Citra Kayu

Jenis	Citra
<p>Angsana <i>(Pterocarpus indicus wild)</i></p>	
<p>Kenari <i>(Canarium Commune L)</i></p>	
<p>Lengkeng <i>(Litchi chinensis Sonn)</i></p>	
<p>Meranti <i>(Shorea balangeran (Korth) Buc)</i></p>	
<p>Sonokeling <i>(Dalbergia Latifolia Ro xb.)</i></p>	

3.3.2.2 Pembagian Data

Dataset citra kayu yang telah didapat kemudian dibagi menjadi dua kelompok yaitu dataset untuk pelatihan dan dataset untuk pengujian, dimana total dataset yang digunakan yaitu 750 citra dengan ketentuan masing-masing jenis citra kayu yaitu 150 citra. Skenario pembagian dataset yang selanjutnya akan dijadikan model dapat dilihat pada Tabel 3.4.

Tabel 3.4 Skenario Split Dataset

Skenario Pembagian Dataset		
Model	Data Latih	Data Uji
A	70%	30%
B	80%	20%
C	90%	10%

3.3.2.3 Image Preprocessing

Image preprocessing adalah tahap awal yang penting dalam pemrosesan citra sebelum melakukan ekstraksi fitur dan identifikasi. Proses ini memastikan citra berada dalam format yang konsisten dan optimal untuk analisis lebih lanjut. Berikut adalah tahapan proses dari masing-masing metode *preprocessing* yang digunakan.

1. Konversi ke Grayscale

Citra berwarna diubah menjadi citra dalam bentuk *grayscale* untuk menyederhanakan data dan mengurangi kompleksitas komputasi, dan hanya informasi intensitas yang dipertahankan. Bertujuan untuk mengurangi dimensi data citra dengan menghilangkan informasi warna yang tidak diperlukan untuk kebutuhan analisis tekstur atau fitur lainnya.

2. Normalisasi Citra

Normalisasi dilakukan untuk mengubah rentang nilai piksel citra agar berada dalam skala yang konsisten antara 0 hingga 255 menggunakan min-max, untuk memastikan bahwa intensitas piksel tersebar merata dalam

rentang yang sesuai dan dapat mengurangi efek dari pencahayaan yang berbeda.

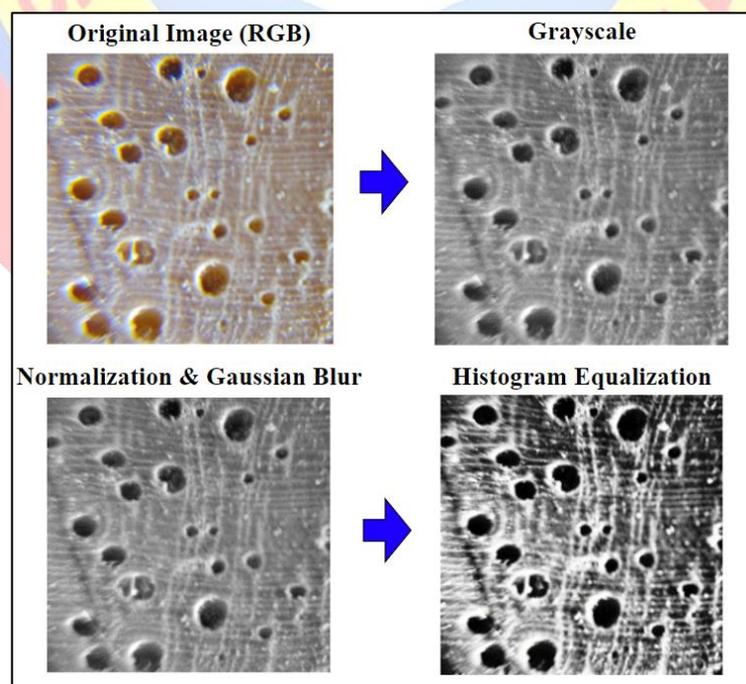
3. Penghilangan Noise

Gaussian blur diterapkan untuk menghilangkan derau (noise) dari citra. Penerapan filter Gaussian yang halus untuk mengaburkan citra dan mengurangi detail halus yang menyebabkan noise, sehingga dapat membantu dalam meningkatkan kualitas citra dan mempermudah ekstraksi fitur.

4. Equalization Histogram

Histogram equalization digunakan untuk meningkatkan kontras citra dengan mendistribusikan nilai intensitas piksel secara lebih merata di seluruh rentang intensitas, dengan meratakan distribusi histogram intensitas citra agar kontras citra meningkat sehingga fitur pentingnya dapat terlihat, juga meningkatkan kualitas visual citranya.

Sampel hasil *preprocessing* citra untuk proses dari citra awal (RGB) ke *Grayscale*, dan normalisasi dan *noise reduction* dengan *Gaussian Blur* ke *Histogram Equalization* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Image Preprocessing Result

3.3.2.4 Ekstraksi Fitur

Pada tahap ini, fitur citra diekstraksi menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM). Teknik ini menganalisis tekstur citra dengan melihat frekuensi kemunculan pasangan piksel dengan intensitas tertentu pada jarak dan orientasi yang spesifik. Dengan GLCM, dapat diperoleh berbagai fitur tekstur seperti kontras, homogenitas, entropi, energi, dan korelasi. Fitur-fitur ini kemudian digunakan untuk proses identifikasi. Proses ini memungkinkan sistem untuk membedakan antara jenis-jenis tekstur yang berbeda pada citra. Memungkinkan sistem untuk membedakan tekstur yang berbeda pada citra dengan lebih akurat serta meningkatkan akurasi dalam identifikasi dengan baik.

Berikut adalah satu sampel perhitungan fitur yang terdapat pada matriks GLCM 3x3 dengan sudut 0° dan jarak 1.

$$\begin{array}{|c|c|c|} \hline 0.2 & 0.3 & 0 \\ \hline 0.3 & 0.2 & 0 \\ \hline 0 & 0 & 0.2 \\ \hline \end{array}$$

Berdasarkan matriks di atas, maka dapat ditentukan perhitungannya sebagai berikut.

1. Contrast (mengukur intensitas perubahan pada citra)

Menghitung kontribusi setiap elemen, lalu dijumlahkan.

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 \cdot p_{(i,j)}$$

$$\text{Contrast} = 0 + 0.3 + 0 + 0.3 = 0.6$$

$$= (2-3)^2 * 0 + (3-1)^2 * 0 + (3-2)^2 * 0 + (3-3)^2 * 0.2 = 0 + 0.4 + 0 + 0 = 0.4$$

$$= 0.6 + 0.4 = 1.0$$

2. Dissimilarity (mengukur seberapa berbeda nilai piksel satu dengan lainnya)

Menghitung selisih absolut antara nilai piksel, lalu dikalikan dengan dengan probabilitasnya dan terakhir dijumlahkan.

$$\text{Dissimilarity} = \sum_{[i,j]} |i-j| * P(i,j)$$

$$\begin{aligned} \text{Dissimilarity} &= 0 + 0.3 + 0 + 0.3 = 0.6 \\ &= |2-3| * 0 + |3-1| * 0 + |3-2| * 0 + |3-3| * 0.2 = 0 + 0.2 + 0 + 0 = 0.2 \\ &= 0.6 + 0.2 = 0.8 \end{aligned}$$

3. Homogeneity (mengukur keseragaman nilai piksel)

Menghitung invers dari kuadrat selisih dari nilai piksel, lalu dikalikan dengan probabilitasnya dan terakhir dijumlahkan.

$$\begin{aligned} \text{Homogeneity} &= \sum_i \sum_j \frac{p(i,j)}{1+|i-j|^2} \\ \text{Homogeneity} &= 0.2 + 0.15 + 0.15 = 0.5 \\ &= (1 / (1 + (2-1)^2)) * 0.3 + (1 / (1 + (3-1)^2)) * 0 + (1 / (1 + (3-2)^2)) * 0 \\ &+ (1 / (1 + (3-3)^2)) * 0.2 = 0.15 + 0 + 0 + 0.2 = 0.35 \\ &= 0.5 + 0.35 = 0.85 \end{aligned}$$

4. Energy (mengukur energi dari citra)

Menghitung setiap elemen matriks dengan dikuadratkan, lalu dijumlahkan.

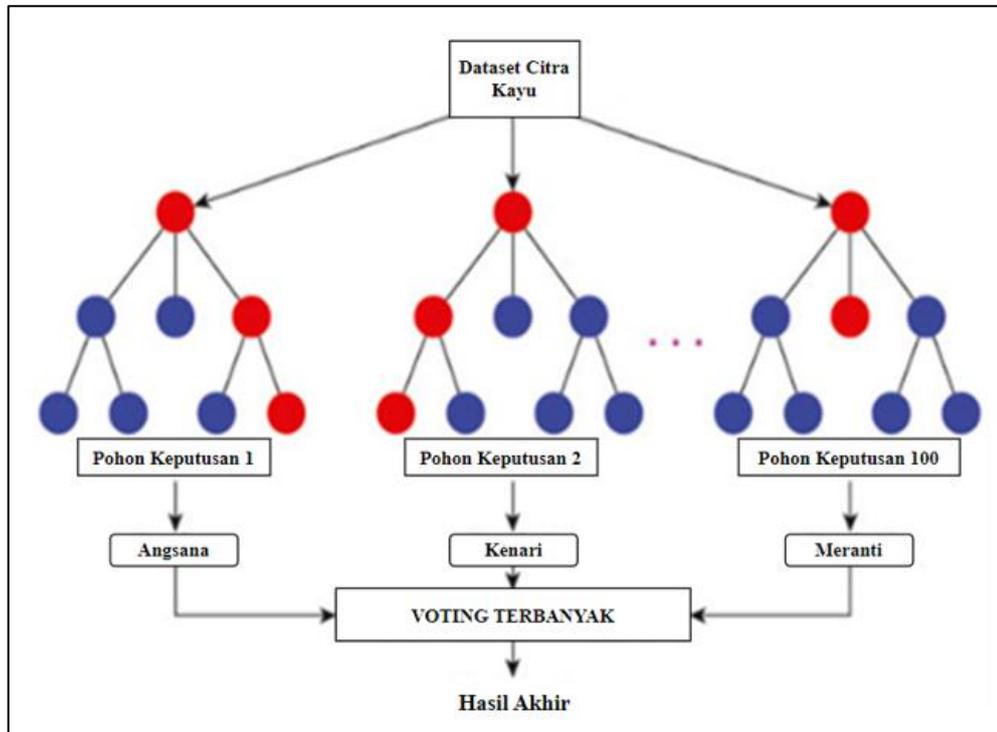
$$\begin{aligned} \text{Energy} &= \sum_i \sum_j p(i,j)^2 \\ \text{Energy} &= 0.04 + 0.09 + 0 + 0.09 = 0.22 \\ &= (0)^2 + (0)^2 + (0.2)^2 = 0.04 \\ &= 0.22 + 0.04 = 0.26 \end{aligned}$$

Berdasarkan perhitungan keempat fitur di atas, hasil yang didapatkan yaitu *contrast* = 1.0, *dissimilarity* = 0.8, *homogeneity* = 0.85, dan *energy* = 0.26.

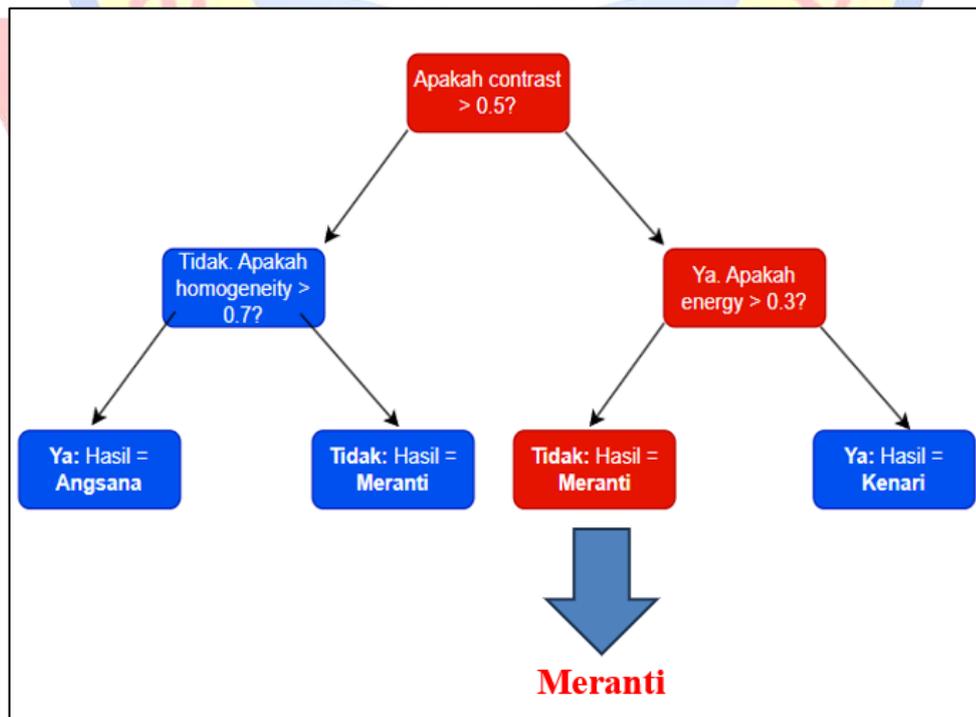
3.3.2.5 Implementasi Algoritma Random Forest

Metode *machine learning* digunakan untuk melatih model identifikasi citra kayu berdasarkan fitur yang diekstraksi yaitu metode algoritma *Random Forest*. Metode tersebut dipilih sebagai model *machine learning* utama karena mampu bekerja dengan baik dengan berbagai jenis fitur yang dapat diekstraksi dari citra kayu, termasuk fitur tekstur. Setelah mengimplementasikan algoritma *Random Forest*, berdasarkan data-data yang sudah dilatih dan sudah divalidasi, model akan dilatih dan diuji menggunakan *Jupyter Notebook* dan dibuat dengan format *pickle* (.pkl) yang selanjutnya akan dipanggil untuk dilakukan pengujian identifikasi dengan citra baru.

Berikut adalah contoh pohon keputusan, yang dapat dilihat pada Gambar 3.6 dan Gambar 3.7.



Gambar 3.6 Alur Pohon Keputusan



Gambar 3.7 Pohon Keputusan Fitur GLCM

Langkah-langkah implementasi metode dapat dijelaskan sebagai berikut.

1. Pembentukan ensambel pohon keputusan menggunakan algoritma *Random Forest*.
2. Pelatihan model menggunakan data latih yang telah disiapkan sebelumnya.
3. Pengujian model menggunakan data uji untuk mengevaluasi kinerja model.

Pada algoritma *Random Forest*, terdapat beberapa perhitungan seperti menghitung *Gini Impurity* untuk mengukur ketidakhomogenan (*heterogen*) node dalam pohon keputusan, mengukur kualitas pembagian node untuk menghitung *Gini Impurity* untuk subset yang dihasilkan, dan mengukur seberapa baik model dapat membedakan antara kelas yang benar dan kelas lainnya menggunakan *Error Generalization*. Untuk contoh sampel perhitungan pada *Random Forest* dapat dijelaskan sebagai berikut.

Sampel data berjumlah 60 data di subset 1 dan sampel data berjumlah 40 data di subset 2, dengan distribusi kelas dapat dijelaskan pada Tabel 3.5.

Tabel 3.5 Distribusi Kelas

Subset 1	
Kelas	Data
Angsana	15
Kenari	25
Lengkeng	5
Meranti	10
Sonokeling	5
Subset 2	
Kelas	Data
Angsana	5
Kenari	5
Lengkeng	5

Meranti	15
Sonokeling	10

1. Menghitung Gini Impurity Subset 1

Subset 1 dengan menghitung proporsi tiap kelas:

1. Angsana: $15/60 = 0.25$
2. Kenari: $25/60 = 0.41$
3. Lengkeng: $5/60 = 0.08$
4. Meranti: $10/60 = 0.16$
5. Sonokeling: $5/60 = 0.08$

Maka,

$$\begin{aligned} \text{Gini}_{\text{Subset 1}} &= 1 - ((0.25)^2 + (0.4167)^2 + (0.0833)^2 + (0.1667)^2 + \\ & (0.0833)^2) = 1 - (0.0625 + 0.1736 + 0.0069 + 0.0278 + 0.0069) = 1 - \\ & 0.2777 = 0.7223 \end{aligned}$$

2. Menghitung Gini Impurity Subset 2

Subset 2 dengan menghitung proporsi tiap kelas:

1. Angsana: $5/40 = 0.12$
2. Kenari: $5/40 = 0.12$
3. Lengkeng: $5/40 = 0.12$
4. Meranti: $15/40 = 0.37$
5. Sonokeling: $10/40 = 0.25$

Maka,

$$\begin{aligned} \text{Gini}_{\text{Subset 2}} &= 1 - ((0.125)^2 + (0.125)^2 + (0.125)^2 + (0.375)^2 + (0.25)^2) \\ &= 1 - (0.0156 + 0.156 + 0.0156 + 0.1406 + 0.0625) = 1 - 0.2498 = \\ & 0.7502 \end{aligned}$$

3. Menghitung Gini Split

$$\begin{aligned} \text{Gini}_{\text{split}} &= (60/100 \times 0.7223) + (40/100 \times 0.7502) = (0.60 \times 0.7223) + \\ & (0.40 \times 0.7502) = 0.43336 + 0.30008 = 0.73346 \end{aligned}$$

Hasil *Gini* yang didapat setelah pembagian yaitu 0.73346, menjelaskan bahwa seberapa baik pembagian fitur telah dilakukan dalam mengurangi ketidakmurnian data di node keputusan.

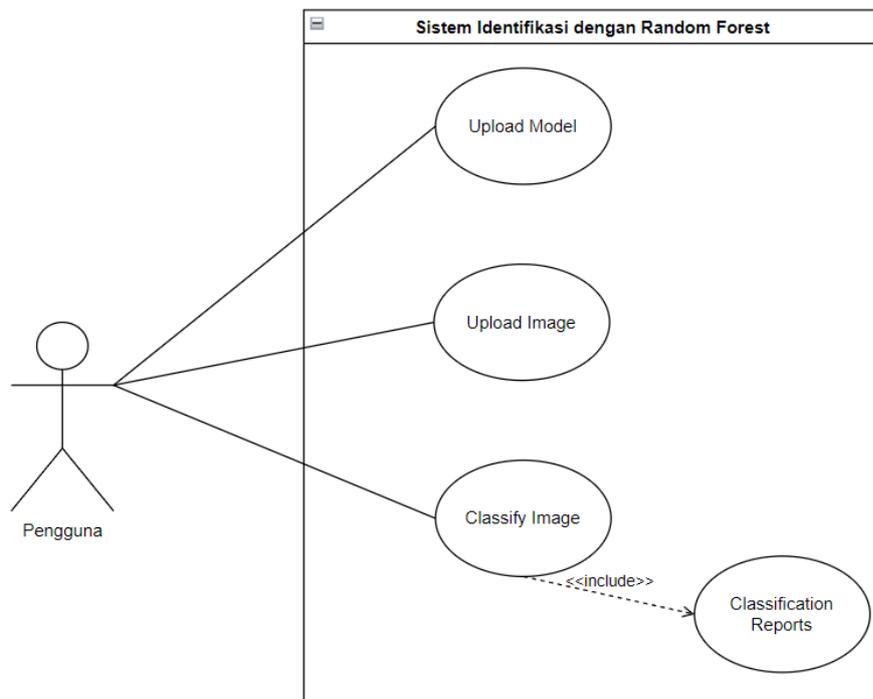
Penggunaan metrik seperti *Gini impurity* digunakan untuk membangun dan menyempurnakan pohon keputusan di dalam *Random Forest*. Hasil dari evaluasi dan pengoptimalan menggunakan metrik ini. Model akan terlatih dengan baik. Setelah data dilatih menggunakan *Random Forest*, model akan dioptimalkan dengan parameter lain yang sesuai yaitu *n_estimators* atau jumlah pohon.

3.3.3 Perancangan UML

Perancangan sistem ini menggunakan UML, yang terdiri dari *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

3.3.3.1 Use Case Diagram

Pada *Use Case Diagram* terdapat satu aktor dalam Sistem identifikasi Jenis Citra Kayu, yang dapat dilihat pada Gambar 3.8.



Gambar 3.8 Use Case Diagram

1. Skenario Use Case: Upload Model

Penjelasan mengenai skenario *Use Case Diagram* berikut, dapat dilihat pada Tabel 3.6.

Tabel 3.6 Skenario Use Case: Upload Model

Nama	Upload Model	
Aktor	Pengguna	
Deskripsi	Ini adalah fungsi dalam program yang digunakan untuk memuat model yang telah disimpan untuk identifikasi lebih lanjut.	
Skenario		
	Aksi Aktor	Reaksi Sistem
Input Dataset		Menampilkan kolom input model
		Membaca file model yang telah diunggah

2. Skenario Use Case: Upload Image

Penjelasan mengenai skenario *Use Case Diagram* berikut, dapat dilihat pada Tabel 3.7.

Tabel 3.7 Skenario Use Case: Upload Image

Nama	Upload Image	
Aktor	Pengguna	
Deskripsi	Ini adalah fungsi dalam program yang digunakan untuk memuat citra tunggal untuk pengujian.	
Skenario		
	Aksi Aktor	Reaksi Sistem
Menekan tombol "Upload Image"		Menampilkan kolom input citra
		Membaca citra
		Menampilkan citra

3. Skenario Use Case: Classify Image

Penjelasan mengenai skenario *Use Case Diagram* berikut, dapat dilihat pada Tabel 3.8.

Tabel 3.8 Skenario Use Case: Classify Image

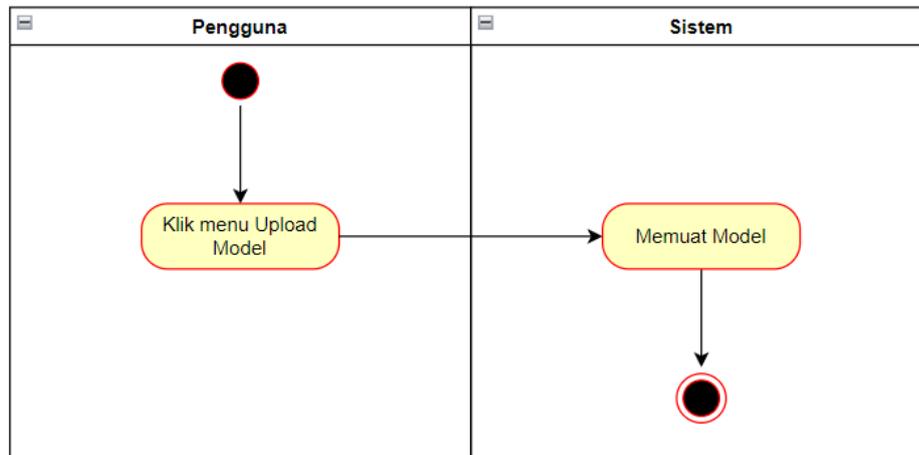
Nama	Classify Image	
Aktor	Pengguna	
Deskripsi	Ini adalah fungsi dalam program untuk memproses pengujian citra-citra dari folder data test, diawali proses <i>preprocessing</i> dan ekstraksi fitur dengan GLCM, dan proses identifikasi.	
Skenario		
	Aksi Aktor	Reaksi Sistem
	Menekan tombol “Classify Image”	Membaca citra yang diunggah
		Melakukan Preprocessing Citra (<i>Grayscale, normalization, gaussian blur, histogram equalization</i>)
		Menghitung fitur GLCM (<i>contrast, dissimilarity, homogeneity, energy</i>)
		Mengidentifikasi citra berdasarkan model
		Menampilkan hasil pengujian (prediksi jenis dan probabilitas per kelas)

3.3.3.2 Activity Diagram

Activity Diagram adalah gambaran alur aktivitas sistem. Berdasarkan usecase diagram diatas, terdapat tiga *activity diagram* yang dapat dijelaskan sebagai berikut.

1. Activity Diagram Upload Model

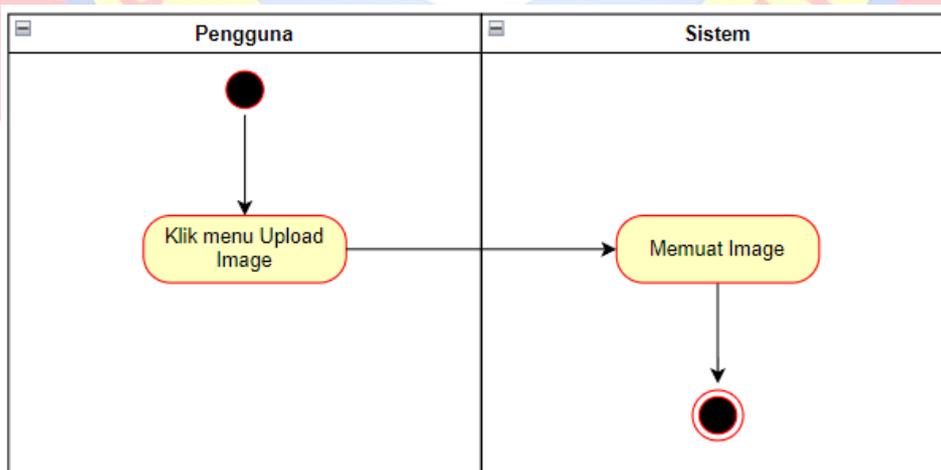
Pada *Activity Diagram* ini terdapat menu untuk memuat file model *pickle* (.pkl) dalam Sistem Identifikasi Citra Kayu, yang dapat dilihat pada Gambar 3.9.



Gambar 3.9 Activity Diagram: Upload Model

2. Activity Diagram Upload Image

Pada *Activity Diagram* ini terdapat menu untuk memuat file citra yang akan dilakukan identifikasi dalam Sistem Identifikasi Jenis Citra Kayu, yang dapat dilihat pada Gambar 3.10.

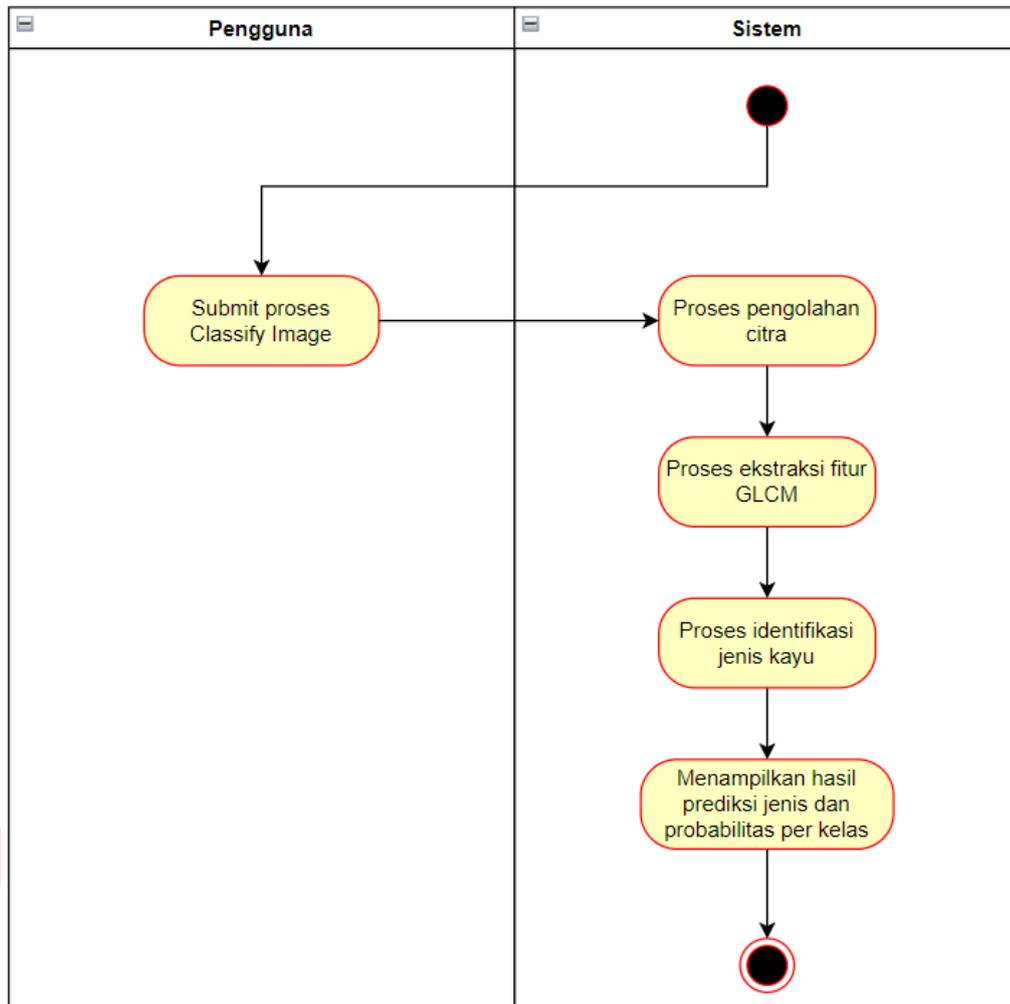


Gambar 3.10 Activity Diagram: Upload Image

3. Activity Diagram Klasifikasi Citra

Pada *Activity Diagram* ini terdapat menu untuk melakukan proses identifikasi, proses keseluruhannya dimulai dengan *preprocessing* citra,

melakukan ekstraksi fitur dengan GLCM, dan mengidentifikasi citra dari data uji berdasarkan model. Dapat dilihat pada Gambar 3.11.



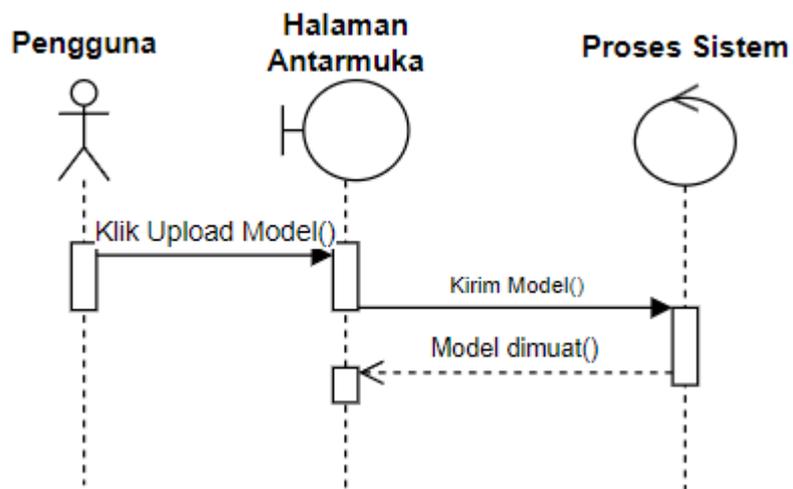
Gambar 3.11 Activity Diagram: Classify Image

3.3.3.3 Sequence Diagram

Pada *Sequence* Diagram terdapat proses yang dijalankan dari awal hingga menghasilkan hasil prediksi dalam sistem identifikasi citra kayu. Terdapat tiga diagram sekuens yang dapat dijelaskan sebagai berikut.

1. Sequence Diagram Upload Model

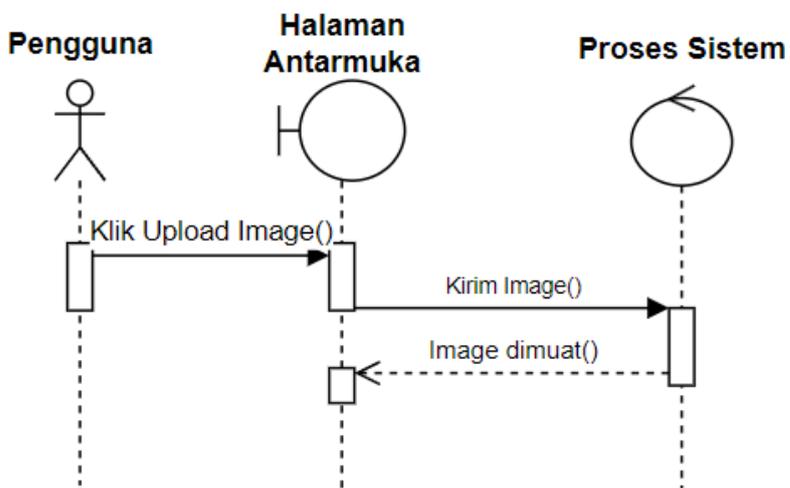
Pada diagram sekuens untuk memuat file model, dapat dilihat pada Gambar 3.12.



Gambar 3.12 Sequence Diagram: Upload Model

2. Sequence Diagram Upload Image

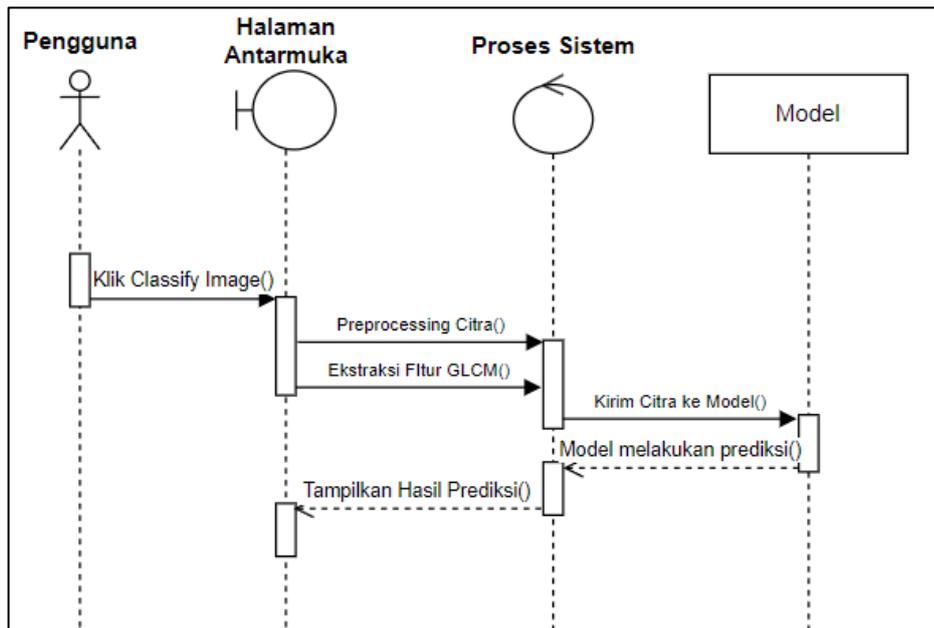
Pada diagram sekuens untuk memuat file citra, dapat dilihat pada Gambar 3.13.



Gambar 3.13 Sequence Diagram: Upload Image

3. Sequence Diagram Classify Image

Pada diagram sekuens untuk melakukan proses identifikasi, dapat dilihat pada Gambar 3.14.



Gambar 3.14 Sequence Diagram: Classify Image

3.3.4 Perancangan Mockup Aplikasi

Pada perancangan *interface* ini memperlihatkan *mockup* tampilan antarmuka aplikasi berbasis *desktop* yang memiliki beberapa menu tertentu sesuai dengan yang telah ditentukan. Tampilan *mockup* antarmuka aplikasi dapat dilihat pada Gambar 3.15.



Gambar 3.15 Mockup Aplikasi Desktop

Tampilan dari antarmuka dengan fitur-fitur yang ada, dapat dijelaskan sebagai berikut.

1. Tombol “Upload Model”

Setelah tombol di-klik, akan menampilkan *popup* untuk menginput model yang sebelumnya telah dibuat dalam format model *pickel* (.pkl).

2. Tombol “Upload Image”

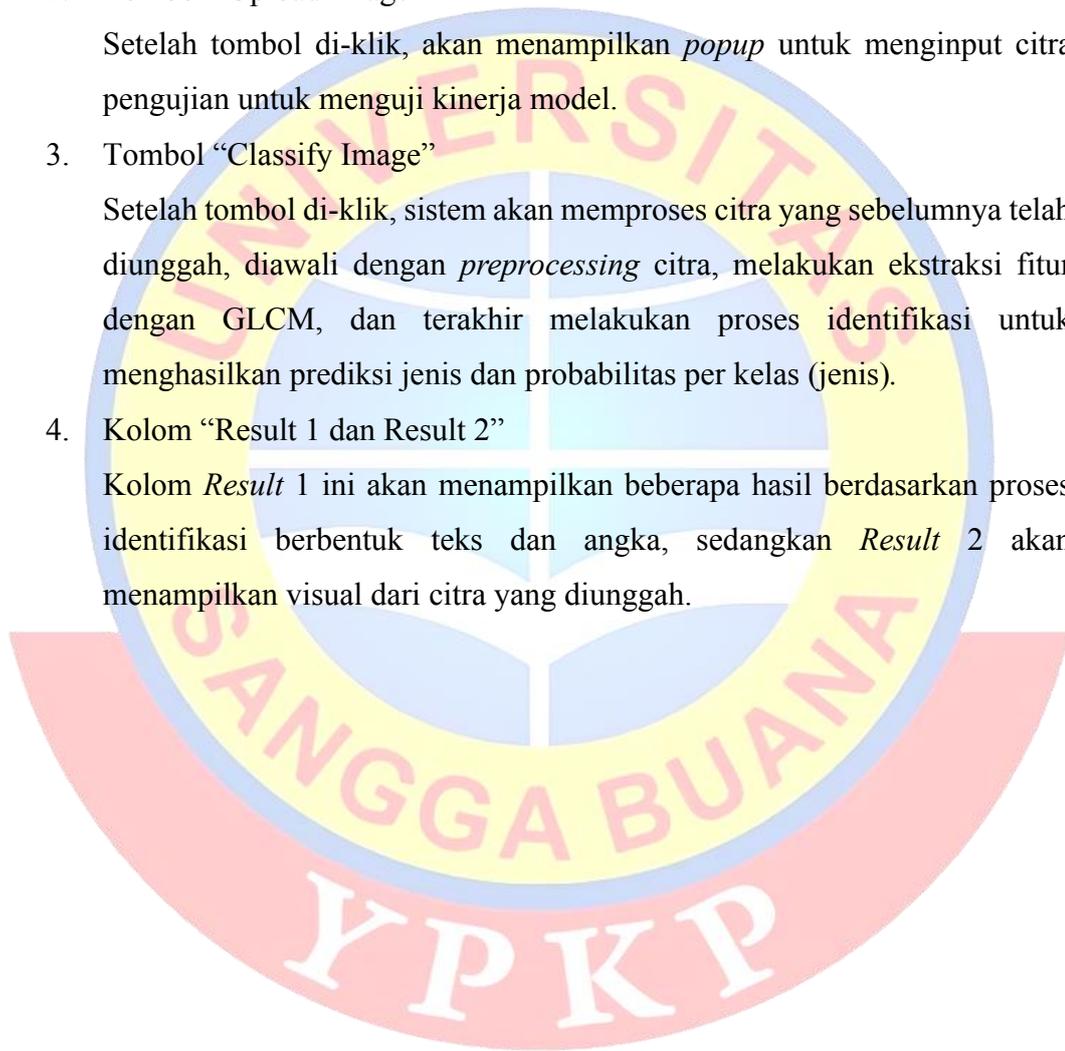
Setelah tombol di-klik, akan menampilkan *popup* untuk menginput citra pengujian untuk menguji kinerja model.

3. Tombol “Classify Image”

Setelah tombol di-klik, sistem akan memproses citra yang sebelumnya telah diunggah, diawali dengan *preprocessing* citra, melakukan ekstraksi fitur dengan GLCM, dan terakhir melakukan proses identifikasi untuk menghasilkan prediksi jenis dan probabilitas per kelas (jenis).

4. Kolom “Result 1 dan Result 2”

Kolom *Result 1* ini akan menampilkan beberapa hasil berdasarkan proses identifikasi berbentuk teks dan angka, sedangkan *Result 2* akan menampilkan visual dari citra yang diunggah.



BAB IV

PEMBAHASAN & HASIL

4.1 Hasil Pengembangan Sistem

Hasil pengembangan sistem berupa sistem yang berfungsi penuh dengan kemampuan untuk memproses dan mengidentifikasi citra baru menggunakan model *machine learning* yang telah dilatih. Sistem ini akan mencakup antarmuka pengguna yang memungkinkan upload model *pickle* dan citra, melakukan *preprocessing* dan ekstraksi fitur, serta menyajikan hasil identifikasi.

4.1.1 Validasi Sistem

Validasi Sistem dilakukan untuk menilai performa model klasifikasi berupa *performance matrix* yang terdiri dari *Precision*, *Accuracy Score*, *F1 Score*, *Recall*, dan Visualisasi *Confusion Matrix* untuk *Identification Reports* yang lebih detail secara menyeluruh.

4.1.1.1 Performance Matrix

Pada tahap ini, validasi sistem digunakan untuk memastikan pencapaian tujuan perancangan aplikasi atau sistem yang telah dibangun, dengan tujuan untuk menemukan kesalahan atau kekurangan yang terdapat pada aplikasi yang dibuat.

Berikut adalah salah satu penjelasan disertai dengan perhitungan yang lengkap dalam bentuk *Performance Matrix* berupa *precision*, *recall*, dan *f1-score* untuk model A (70:30).

Berikut adalah tabel *confusion matrix*, yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Confusion Matrix dalam Tabel

Class	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
Angsana	36	176	4	9

Kenari	40	174	6	5
Lengkeng	37	171	9	8
Meranti	29	168	12	16
Sonokeling	26	154	26	19

Selanjutnya, untuk perhitungannya dapat dijelaskan sebagai berikut.

1. Jenis Angsana

Berikut adalah perhitungan untuk *recall*, *precision*, dan *F1-Score* untuk jenis Angsana.

a. Precision

$$\text{Precision} = \frac{36}{36+4} = \frac{36}{40} = 0.9000$$

b. Recall

$$\text{Recall} = \frac{36}{36+9} = \frac{36}{45} = 0.8000$$

c. F1-Score

$$\text{F1-Score} = 2 \times \frac{0.9000 \times 0.8000}{0.9000+0.8000} = 2 \times \frac{0.7200}{1.7000} = \frac{1.4400}{1.7000} = 0.8471$$

2. Jenis Kenari

Berikut adalah perhitungan untuk *recall*, *precision*, dan *F1-Score* untuk jenis Kenari.

a. Precision

$$\text{Precision} = \frac{40}{40+6} = \frac{40}{46} = 0.8696$$

b. Recall

$$\text{Recall} = \frac{40}{40+5} = \frac{40}{45} = 0.8889$$

c. F1-Score

$$\text{F1-Score} = 2 \times \frac{0.8696 \times 0.8889}{0.8696+0.8889} = 2 \times \frac{0.7727}{1.7585} = \frac{1.5455}{1.7585} = 0.8785$$

3. Jenis Lengkeng

Berikut adalah perhitungan untuk *recall*, *precision*, dan *F1-Score* untuk jenis Lengkeng.

a. Precision

$$\text{Precision} = \frac{37}{37+9} = \frac{37}{46} = 0.8043$$

b. Recall

$$\text{Recall} = \frac{37}{37+8} = \frac{37}{45} = 0.8222$$

c. F1-Score

$$\text{F1-Score} = 2 \times \frac{0.8043 \times 0.8222}{0.8043+0.8222} = 2 \times \frac{0.6618}{1.6265} = \frac{1.3236}{1.6265} = 0.8125$$

4. Jenis Meranti

Berikut adalah perhitungan untuk *recall*, *precision*, dan *F1-Score* untuk jenis Meranti.

a. Precision

$$\text{Precision} = \frac{29}{29+12} = \frac{29}{41} = 0.7073$$

b. Recall

$$\text{Recall} = \frac{29}{29+16} = \frac{29}{45} = 0.6444$$

c. F1-Score

$$\text{F1-Score} = 2 \times \frac{0.7073 \times 0.6444}{0.7073+0.6444} = 2 \times \frac{0.4553}{1.3517} = \frac{0.9106}{1.3517} = 0.6740$$

5. Jenis Sonokeling

Berikut adalah perhitungan untuk *recall*, *precision*, dan *F1-Score* untuk jenis Sonokeling.

a. Precision

$$\text{Precision} = \frac{26}{26+26} = \frac{26}{52} = 0.5000$$

b. Recall

$$\text{Recall} = \frac{26}{26+19} = \frac{26}{45} = 0.5778$$

c. F1-Score

$$\text{F1-Score} = 2 \times \frac{0.5000 \times 0.5778}{0.5000+0.5778} = 2 \times \frac{0.2894}{1.0778} = \frac{0.5788}{1.0778} = 0.5375$$

Sehingga untuk keseluruhan hasil perhitungan, dapat dijelaskan pada Tabel 4.2.

Tabel 4.2 Performance Matrix Result

Jenis Angsana			
Perf. Matrix	Model A (70/30)	Model B (80/20)	Model C (90/10)
Precision	90%	92%	92%
Recall	80%	77%	80%
F1-Score	85%	84%	86%
Jenis Kenari			
Perf. Matrix	Model A (70/30)	Model B (80/20)	Model C (90/10)
Precision	87%	87%	88%
Recall	89%	77%	100%
F1-Score	88%	84%	94%
Jenis Lengkeng			
Perf. Matrix	Model A (70/30)	Model B (80/20)	Model C (90/10)
Precision	80%	93%	93%
Recall	82%	87%	87%
F1-Score	81%	90%	90%
Jenis Meranti			
Perf. Matrix	Model A (70/30)	Model B (80/20)	Model C (90/10)
Precision	71%	76%	71%
Recall	64%	73%	80%
F1-Score	67%	75%	75%
Jenis Sonokeling			
Perf. Matrix	Model A (70/30)	Model B (80/20)	Model C (90/10)
Precision	50%	57%	64%
Recall	58%	70%	60%
F1-Score	54%	63%	62%

Berdasarkan hasil dari *performance matrix* untuk jenis kayu dengan model yang berbeda, dapat dianalisis sebagai berikut.

1. Jenis Angsana

Model B dan Model C sama-sama menunjukkan performa yang terbaik dengan *precision* mencapai 92% sedangkan model C sedikit lebih rendah di 90%, walaupun *recall* pada model B yang didapat sedikit lebih rendah yaitu 77% dibandingkan dengan model lainnya.

2. Jenis Kenari

Keseluruhan model mendapatkan *precision* yang cukup baik yaitu 87%, dimana model C lebih tinggi dengan perbedaan satu persen saja yaitu 88%, *recall* sebesar 100%, dan *F1-score* sebesar 94%. Menunjukkan bahwa ketiga model sangat baik dalam mengidentifikasi jenis kayu Meranti.

3. Jenis Lengkeng

Kedua Model B dan C memiliki skor yang sama, dan menunjukkan performa yang sangat baik dengan *precision* mencapai 93%, *recall* mencapai 87%, dan *F1-score* mencapai 90%, dimana model A menunjukkan performa yang sedikit lebih rendah dari seluruh skor metrik.

4. Jenis Meranti

Model B sedikit lebih baik dengan skor *precision* mencapai 76%, tetapi untuk *recall* dan *F1-score* lebih unggul model C dengan skor 80% dan 75%.

5. Jenis Sonokeling

Ketiga model hanya memiliki skor rendah, dimana skor tertinggi yang didapat untuk *precision* yaitu 64% untuk model C, dan *recall* yaitu 70% untuk model B, sedangkan *F1-score* paling tinggi mencapai 63% untuk model B. Ini menunjukkan bahwa ketiga model kurang baik dalam mengidentifikasi jenis kayu Sonokeling.

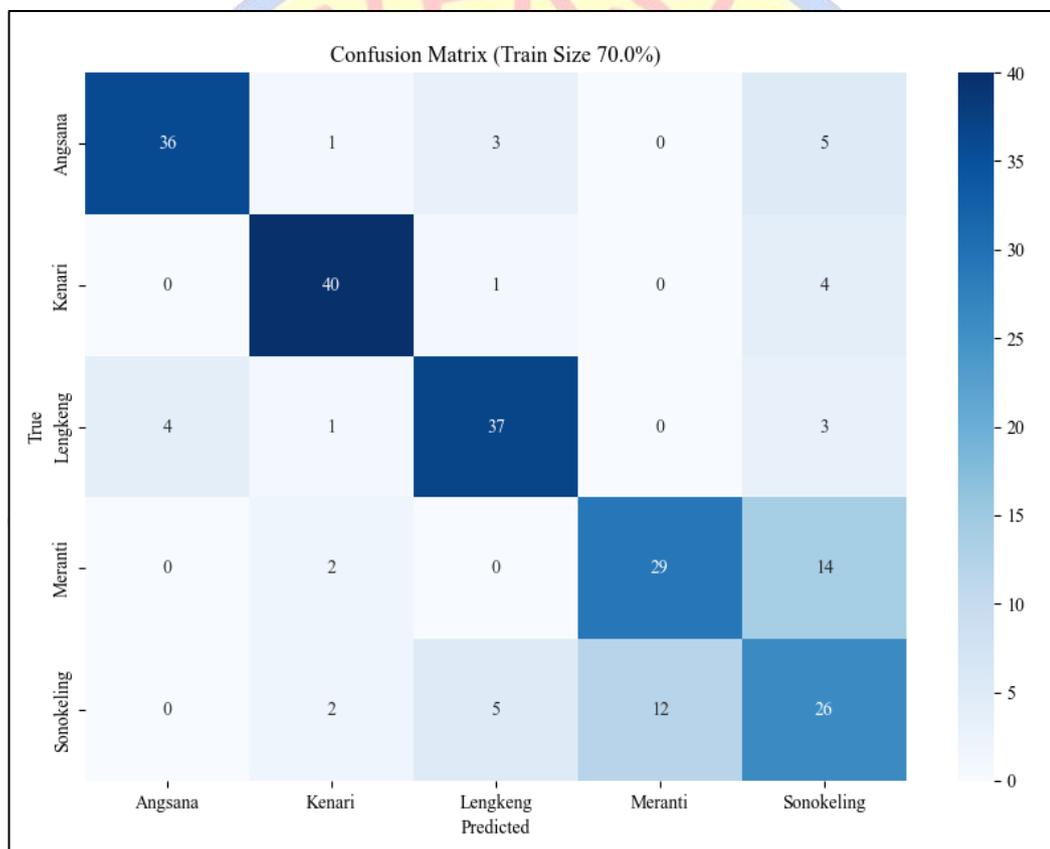
Berdasarkan hasil analisis *performance matrix*, model C (90/10) menunjukkan performa yang sangat baik terhadap hampir seluruh jenis kayu, terutama dalam matrik *precision* dan *F1-score*, sedangkan Model B (80/20) dan model A (70/30) memiliki performa yang sedikit baik untuk beberapa jenis kayu.

4.1.1.2 Confusion Matrix

Visualisasi dalam *confusion matrix* untuk validasi sistem terhadap kinerja sistem yang dibuat berdasarkan skenario pembagian data disertai dengan perhitungan untuk menentukan akurasi model.

1. Model A (70/30)

Berikut adalah visualisasi *confusion matrix* model A untuk skenario pembagian data 70/30, yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Confusion Matrix: Model A (70/30)

Perhitungan akurasi untuk model A dengan pembagian data 70/30, dapat dijelaskan sebagai berikut.

a. Hitung Total Prediksi

$$36 + 1 + 3 + 0 + 5 + 0 + 40 + 1 + 0 + 4 + 4 + 1 + 37 + 0 + 3 + 0 + 2 + 0 + 29 + 14 + 0 + 2 + 5 + 12 + 26 = 225$$

b. Hitung Jumlah Prediksi Benar

$$36 + 40 + 37 + 29 + 26 = 168$$

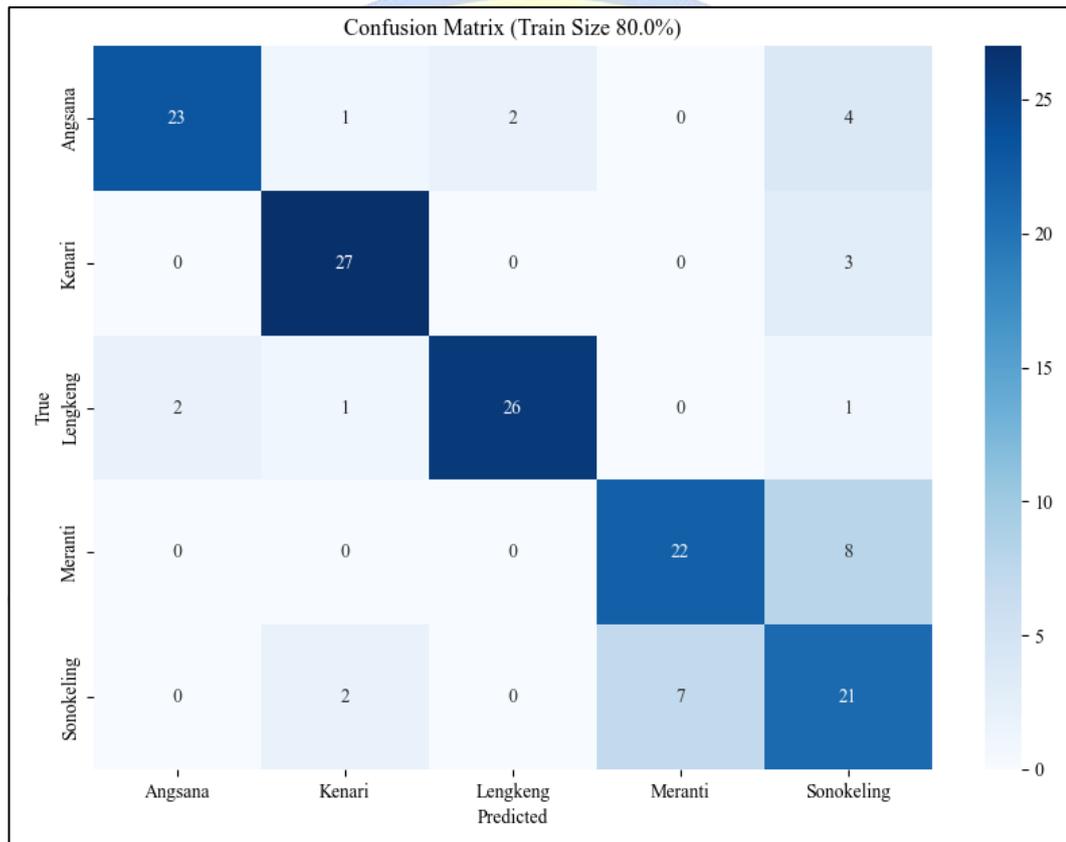
c. Hitung Akurasi

$$\text{Akurasi} = (\text{Jumlah Prediksi Benar}) / (\text{Total Prediksi}) = 168 / 225 = 0.7467$$

$$= 75\%$$

2. Model B (80/20)

Untuk skenario pembagian data model B, dapat dilihat pada Gambar 4.2.



Gambar 4.2 Confusion Matrix: Model B (80/20)

Perhitungan akurasi untuk model B dengan pembagian data 80/20, dapat dijelaskan sebagai berikut.

a. Hitung Total Prediksi

$$23 + 1 + 2 + 0 + 4 + 0 + 27 + 0 + 0 + 3 + 2 + 1 + 26 + 0 + 1 + 0 + 0 + 0 + 0 + 22 + 8 + 0 + 2 + 0 + 7 + 21 = 150$$

b. Hitung Jumlah Prediksi Benar

$$23 + 27 + 26 + 22 + 21 = 119$$

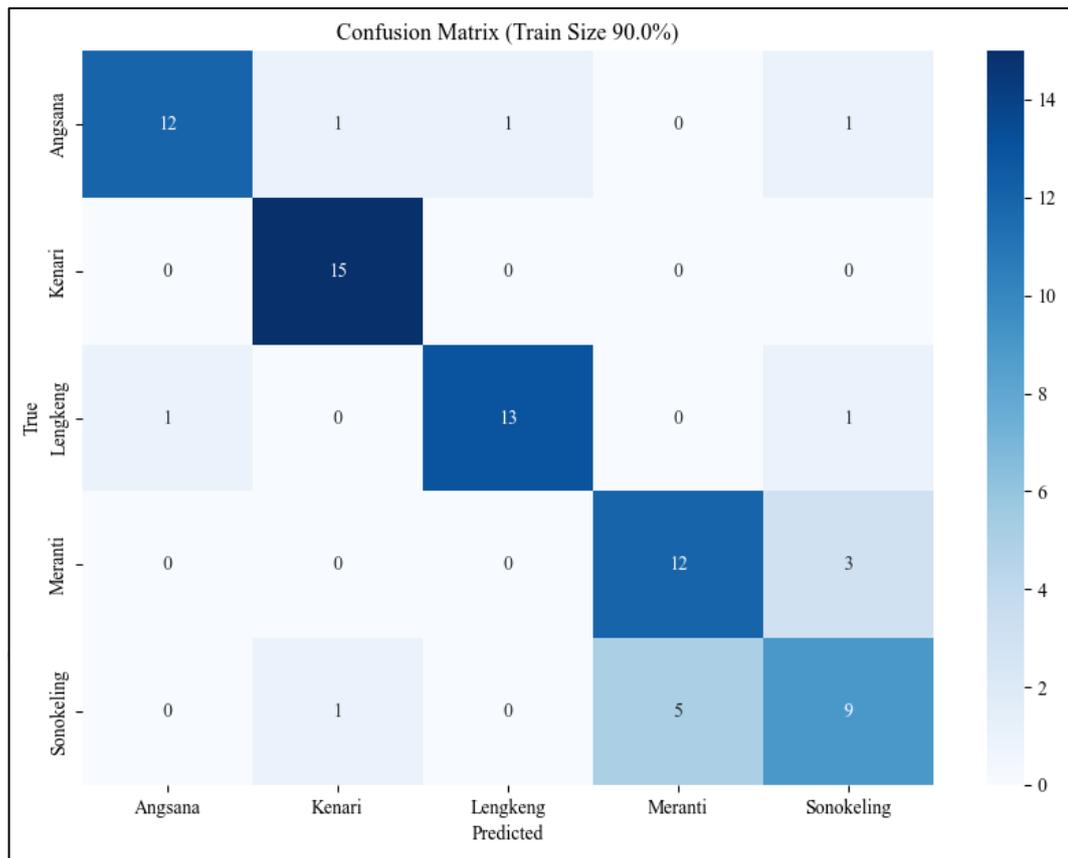
c. Hitung Akurasi

$$\text{Akurasi} = (\text{Jumlah Prediksi Benar}) / (\text{Total Prediksi}) = 119 / 150 = 0.7933$$

$$= 80\%$$

3. Model C (90/10)

Visualisasi *confusion matrix* untuk model C, dapat dilihat pada Gambar 4.3.



Gambar 4.3 Confusion Matrix: Model C (90/10)

Perhitungan akurasi untuk model C dengan pembagian data 90/10, dapat dijelaskan sebagai berikut.

a. Hitung Total Prediksi

$$12 + 1 + 1 + 0 + 1 + 0 + 15 + 0 + 0 + 0 + 1 + 0 + 13 + 0 + 1 + 0 + 0 + 0 + 12 + 3 + 0 + 1 + 0 + 5 + 9 = 75$$

b. Hitung Jumlah Prediksi Benar

$$12 + 15 + 13 + 12 + 9 = 61$$

c. Hitung Akurasi

$$\text{Akurasi} = (\text{Jumlah Prediksi Benar}) / (\text{Total Prediksi}) = 61 / 75 = 0.8133 = 81\%$$

Berdasarkan perhitungan akurasi, akurasi model yang dibuat dapat dilihat pada Tabel 4.3.

Tabel 4.3 Skor Akurasi Model

Model	Skenario Split Data	Skor Akurasi
A	70:30	74%
B	80:20	79%
C	90:10	81%

Berdasarkan hasil akurasi yang didapat dari ketiga model tersebut, Model C menunjukkan skor akurasi tertinggi dengan nilai 81%.

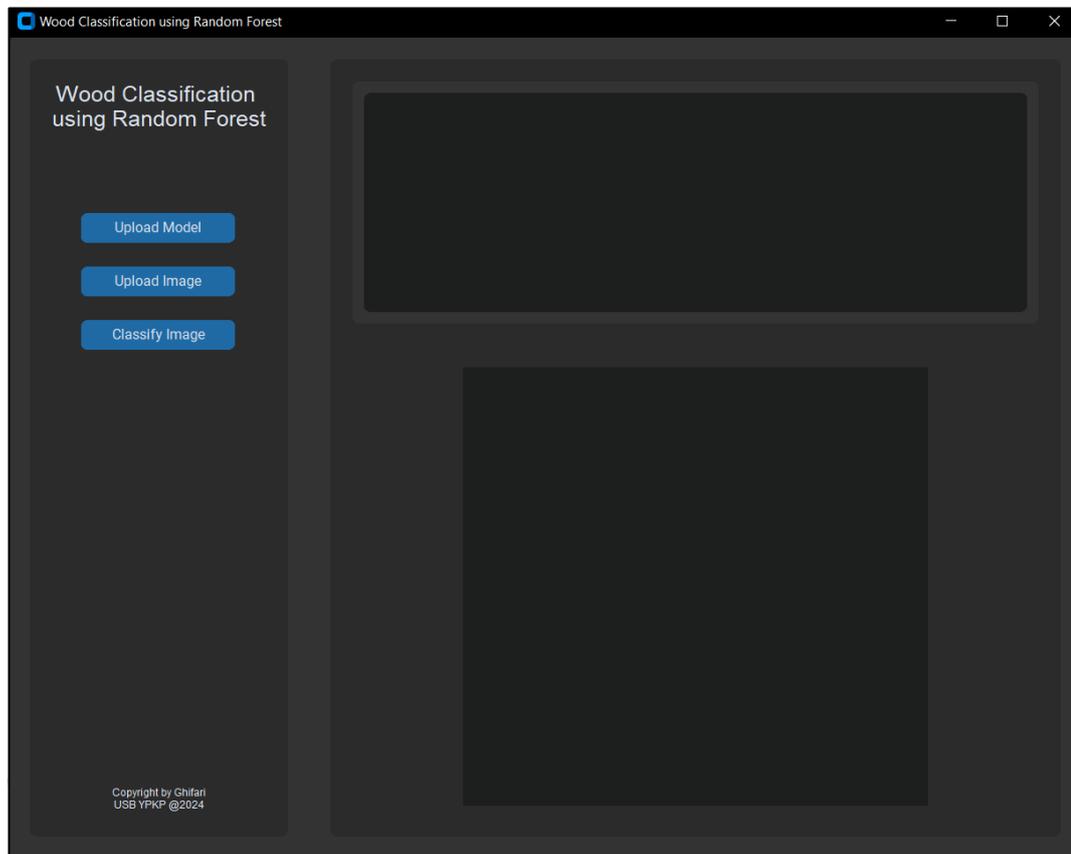
4.1.2 Pengujian Sistem

Pengujian Sistem dilakukan untuk mengevaluasi kinerja model dengan data yang baru, dimana data uji ini terpisah dari data latih dan data validasi pada saat proses pelatihan. Pengujian ini digunakan untuk mengukur seberapa baik model dapat menggeneralisasi pengetahuan yang diperoleh pada saat pelatihan, untuk menghasilkan prediksi identifikasi berdasarkan pelatihan yang telah dilakukan, dan untuk mengetahui apakah model yang dibuat telah berjalan dengan sesuai atau tidak. Hasil yang didapat berupa jenis data yang diprediksi dan probabilitas tiap kelas.

Pada tahap pengujian ini, akan menggunakan aplikasi pengujian yang dibuat dengan *Visual Studio Code* menggunakan ekstensi *customtkinter* dan *tkinter* sebagai konstruksi untuk membangun *Graphical User Interface* (GUI) aplikasi berbasis *desktop*. Tangkapan layar berupa tampilan antarmuka aplikasi *desktop* ini adalah sebagai berikut.

1. Tampilan Halaman Utama

Berikut ini representasi visual berupa tangkapan layar dari antarmuka halaman utama aplikasi yang telah diimplementasikan. Dapat dilihat pada Gambar 4.4.



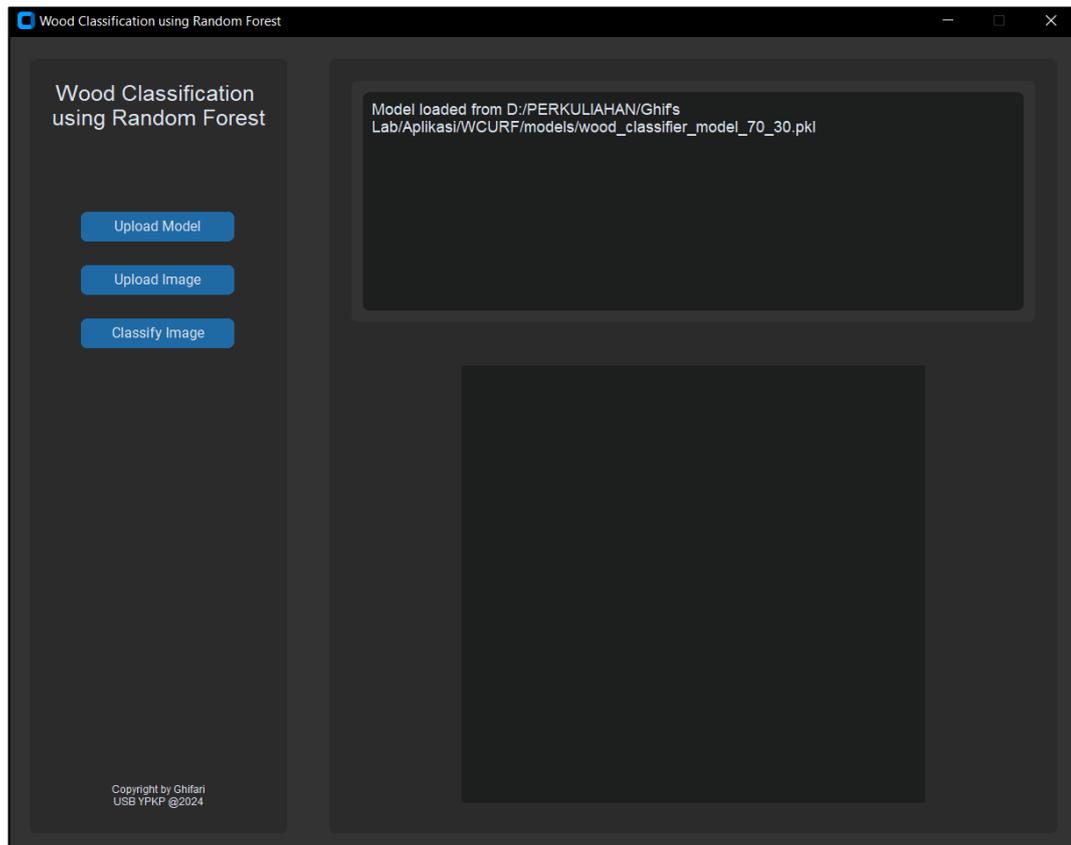
Gambar 4.4 Tangkapan Layar Aplikasi: Halaman Utama

Berikut adalah penjelasan untuk cara kerja aplikasi yang telah dibangun.

a. Tombol “Upload Model”

Proses pertama yaitu pengguna memilih tombol upload model, lalu mengupload model yang sebelumnya telah dibuat dalam format *pickle* (.pkl).

Setelah model diunggah, pada aplikasi terdapat *output* pada panel kanan yang menampilkan teks jika model telah berhasil dimuat dalam aplikasi. Dapat dilihat pada Gambar 4.5.

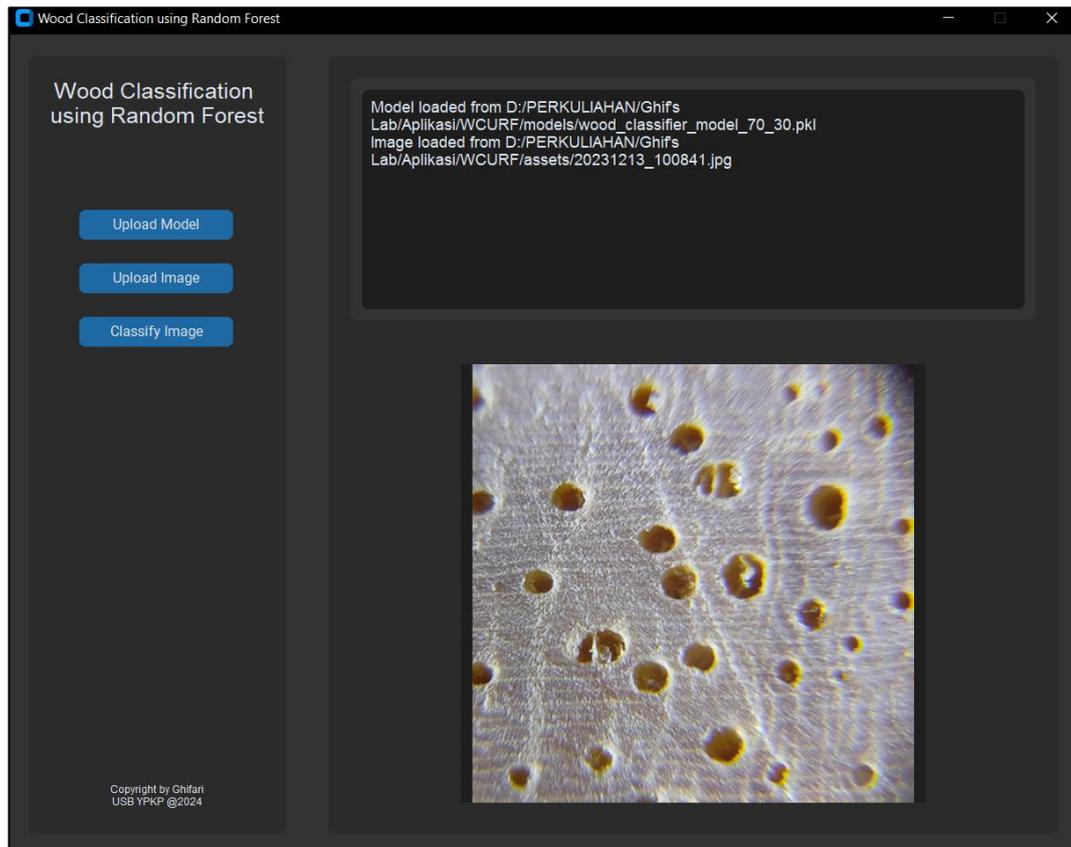


Gambar 4.5 Model Loaded

b. Tombol “Upload Image”

Proses kedua yaitu pengguna memilih tombol *upload image* dengan format *jpg*, *jpeg*, *png*, dan *bmp* untuk mengunggah citra tunggal baru yang selanjutnya akan diproses ke tahap prediksi.

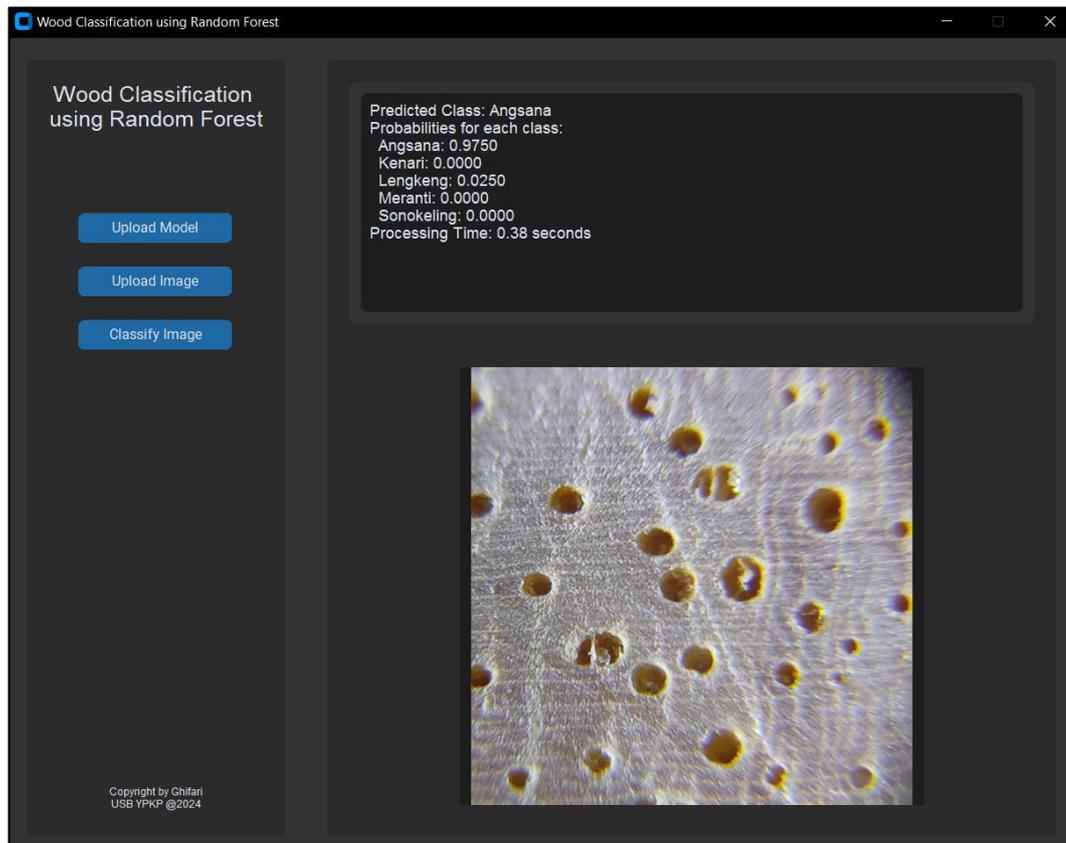
Setelah memilih citra, pada aplikasi terdapat *output* pada panel kanan yang menampilkan teks jika citra telah berhasil dimuat dalam aplikasi dan menampilkan pratinjau citra yang diunggah. Dapat dilihat pada Gambar 4.6.



Gambar 4.6 Image Loaded

c. Tombol “Classify Image”

Proses terakhir ketika pengguna menekan tombol ini, maka sistem akan menjalankan proses untuk citra yang sebelumnya telah diunggah meliputi *preprocessing image*, ekstraksi fitur, dan terakhir akan melakukan identifikasi prediksi jenis berdasarkan model, serta menampilkan jenis prediksi dan probabilitas per jenis. Hasil proses pengujian dengan aplikasi dapat dilihat pada Gambar 4.7.



Gambar 4.7 Identification Result

Berdasarkan proses yang telah dilakukan pada aplikasi. Didapat hasilnya bahwa citra acak yang diproses telah terdeteksi dan terprediksi oleh sistem dengan jenis Angsana, dan memiliki probabilitas tertinggi dari seluruh kelas sebesar 97%.

4.1.3 Evaluasi Hasil

Evaluasi hasil berupa penilaian kinerja model berdasarkan hasil pengujian pada aplikasi dengan sejumlah citra acak untuk menentukan efektivitas model dalam mengidentifikasi atau mengklasifikasikan citra baru. Hasil evaluasi memberikan gambaran tentang seberapa baik model bekerja pada data yang tidak dilatih sebelumnya dan dapat mencakup analisis terperinci berdasarkan metrik performa dan prediksi model, dimana hasil yang didapat pada saat pengujian dengan jumlah lima citra acak tiap model, dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Citra Baru

Model A (70/30)

Nama Citra	Terprediksi	Probabilitas Tertinggi
20231213_100841.jpg	Angsana	97%
20231213_100843.jpg	Angsana	98%
20231213_220915.jpg	Sonokeling	49%
20231214_142501.jpg	Sonokeling	45%
20231213_145502.jpg	Kenari	40%
Model B (80/20)		
Nama Citra	Terprediksi	Probabilitas Tertinggi
20231213_145600.jpg	Lengkeng	65%
20231213_100844.jpg	Angsana	63%
20231213_145605.jpg	Lengkeng	90%
20231214_142439.jpg	Sonokeling	73%
20231214_142447.jpg	Sonokeling	92%
Model C (90/10)		
Nama Citra	Terprediksi	Probabilitas Tertinggi
20231213_220930.jpg	Kenari	99%
20231213_100925.jpg	Angsana	94%
20231214_152840.jpg	Meranti	93%
20231213_220933.jpg	Kenari	98%
20231214_153325.jpg	Meranti	75%

Berdasarkan hasil prediksi, ketiga model menunjukkan perbedaan kinerja yang signifikan. Model A (70/30) sangat baik dalam mengenali Angsana dengan probabilitas tinggi, tetapi kinerjanya rendah untuk jenis lainnya seperti Sonokeling dan Kenari, dengan probabilitas terendah 41%. Model B (80/20) lebih baik dalam mengidentifikasi Lengkeng dan Sonokeling, dengan probabilitas tinggi yaitu 92% dan 90%. Model C (90/10) menunjukkan kinerja terbaik secara keseluruhan, dengan probabilitas tinggi untuk Kenari, Angsana, dan Meranti. Model C adalah yang paling konsisten dan akurat, dengan skor probabilitas tertinggi 99% dan terendah 75%.

BAB V

PENUTUP

5.1 Kesimpulan

Berikut adalah kesimpulan berupa poin-poin yang menjawab tujuan penelitiannya, dapat dijelaskan sebagai berikut.

1. Sistem identifikasi jenis kayu yang mengaplikasikan algoritma Random Forest dan metode ekstraksi fitur Gray Level Co-occurrence Matrix (GLCM) telah berhasil diimplementasikan, menunjukkan efektivitas yang signifikan dengan akurasi tertinggi mencapai 81%. Temuan ini mengindikasikan bahwa penggunaan metode Random Forest untuk identifikasi dan metode GLCM untuk ekstraksi fitur tekstur sangat efektif dalam mengatasi keterbatasan dalam identifikasi kayu dibandingkan dengan metode identifikasi secara manual.
2. Berdasarkan validasi sistem, model yang dikembangkan telah melalui proses validasi yang baik dalam identifikasi jenis kayu. Melalui tahap pengujian, dimana model C dengan proporsi pembagian data 90:10 menunjukkan hasil yang optimal, dimana citra kayu jenis Kenari yang terprediksi memiliki skor probabilitas mencapai 99%.

5.2 Saran

Sebagai masukan untuk penelitian berikutnya, dijabarkan sebagai berikut.

1. Penerapan dengan teknik *Deep Learning* seperti *Convolutional Neural Networks (CNN)*, untuk meningkatkan akurasi identifikasi dengan memanfaatkan fitur yang lebih kompleks.
2. Penerapan dengan teknologi lain seperti penggunaan *tensorflow* untuk pengujian secara langsung pada kayu.

DAFTAR PUSTAKA

- Adianto, A. (2020, September 24). *Kayu sebagai Bahan dan Struktur Konstruksi Bangunan*. <https://www.constructionplusasia.com/id/material-kayu/>
- Ahaddin, F. (2015, June 28). *Mengenal Use Case Diagram*. <https://www.dumetschool.com/blog/Mengenal-Use-Case-Diagram>
- Alda Putri Utami, Febryanti Sthevanie, & Kurniawan Nur Ramadhani. (2021). Pengenalan Logo Kendaraan Menggunakan Metode Local Binary Pattern dan Random Forest. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(4), 639–646.
- Anggiratih, E., & Putra, A. E. (2019). Ship Identification on Satellite Image Using Convolutional Neural Network and Random Forest. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(2), 117.
- Ariyani, R., & Arif, Z. (2022). Tinjauan Pustaka Sistematis: Penerapan Metode Position Averaging Point Pada Computer Vision. In *JACIS : Journal Automation Computer Information System* (Vol. 2, Issue 02).
- Cholissodin, I., Sutrisno, Soebroto, A. A., Hasanah, U., & Febiola, I. Y. (2020). *AI, MACHINE LEARNING & DEEP LEARNING (Teori & Implementasi)*.
- Danuri, M. (2019). *PERKEMBANGAN DAN TRANSFORMASI TEKNOLOGI DIGITAL*.
- Dompeipen, T. A., Sompie, S. R. U. A., & Najoran, M. E. I. (2021). *Computer Vision Implementation for Detection and Counting the Number of Humans*.
- Driscoll, M. (2022, March 15). *Jupyter Notebook: An Introduction*. <https://realpython.com/jupyter-notebook-introduction/#getting-up-and-running-with-jupyter-notebook>
- Faiz. (2023, December 6). *Mengenal Apa Itu Machine Learning: Pemahaman Dasar dan Penerapannya*. <https://dte.telkomuniversity.ac.id/mengenal-apa-itu-machine-learning-pemahaman-dasar-dan-penerapannya/>
- Faulina, R. A. (2023, March 23). *Apa itu UML? Ini Pengertian, Fungsi, dan Contohnya*. <https://www.sekawanmedia.co.id/blog/apa-itu-uml/>
- Gifari, J. (2020, December 1). *Belajar Machine Learning Dengan Library Python : Scikit-Learn*. <https://dqqlab.id/belajar-machine-learning-dengan-library-python-scikit-learn>

Hardini, I. R. (2019). A Survey on Machine learning and IoT. *ITEJ*, 4(2), 99–113.

Karsito, & Susanti, S. (2019). *KLASIFIKASI KELAYAKAN PESERTA PENGAJUAN KREDIT RUMAH DENGAN ALGORITMA NAÏVE BAYES DI PERUMAHAN AZZURA RESIDENCIA*.

Kristiawan, K., Somali, D. D., Linggan jaya, T. A., & Widjaja, A. (2020). Deteksi Buah Menggunakan Supervised Learning dan Ekstraksi Fitur untuk Pemeriksa Harga. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(3).

Lamasigi, Z. Y., & Bode, A. (2021). Influence of gray level co-occurrence matrix for texture feature extraction on identification of batik motifs using k-nearest neighbor. *ILKOM Jurnal Ilmiah*, 13(3), 322–333.

Muhammad Rizky. (2019, November 22). *UML Diagram : Activity Diagram*. <https://socs.binus.ac.id/2019/11/22/uml-diagram-activity-diagram/>

Mulyono, I. U. W., Lukita, T. C., Sari, C. A., Setiadi, D. R. I. M., Rachmawanto, E. H., Susanto, A., Putra, M. D. M., & Santoso, D. A. (2020). Parijoto Fruits Classification using K-Nearest Neighbor Based on Gray Level Co-Occurrence Matrix Texture Extraction. *Journal of Physics: Conference Series*, 1501(1).

Muna, N., Lutfi Afriansyah, F., & Bagus Suprayogy, A. (2020). *PENERAPAN ALGORITMA RANDOM FOREST UNTUK IDENTIFIKASI DEHIDRASI BERBASIS CITRA URINE*.

Mutaqin, D. J., Nurhayani, F. O., & Rahayu, N. H. (2022). Performa Industri Hutan Kayu dan Strategi Pemulihan Pascapandemi Covid-19. *Bappenas Working Papers*, 5(1), 48–62.

Nadiyah Hidayati, & Maulidah, M. (2023). EKSTRAKSI FITUR DENGAN COLOR HISTOGRAM DAN CLASSIFIER RANDOM FOREST PADA CITRA KUPU-KUPU. *JAMI: Jurnal Ahli Muda Indonesia*, 4(2), 148–157.

Nuraini, R. (2022). Klasifikasi Citra Jenis Kapasitor Menggunakan Kombinasi Algoritma K-Nearest Neighbor dan Principal Component Analysis. *Journal of Computer System and Informatics (JoSYC)*, 3(3), 133–140.

Prakasa, E., Sugiarto, B., Rianto Prajitno, D., Muhammad Erwin, I., Wardoyo, R., & Damayanti, R. (2023). Wood Classification using Convolutional Neural Networks and Vessel Features on Cross Section Images. *International Journal of Computing and Digital Systems*, 13(1).

Purwanto, A. D., Wikantika, K., Deliar, A., & Darmawan, S. (2023). Decision Tree and Random Forest Classification Algorithms for Mangrove Forest Mapping in Sembilang National Park, Indonesia. *Remote Sensing*, 15(1).

Putra, J. W. G. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*.

Rahayu, N. M. Y. D., Antara Kesiman, M. W., & Gunadi, I. G. A. (2021). Identifikasi Jenis Kayu Berdasarkan Fitur Tekstur Local Binary Pattern Menggunakan Metode Learning Vector Quantization. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 10(3), 157.

Rahman, S., Sembiring, A., Siregar, D., Khair, H., Prahmana, G., Puspadini, R., & Zen, M. (2023). *PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK*.

Ratna, S. (2020). PENGOLAHAN CITRA DIGITAL DAN HISTOGRAM DENGAN PHYTON DAN TEXT EDITOR PHYCHARM. In *Technologia* (Vol. 11, Issue 3).

Reyvan. (2020, November 17). *3 Library yang Esensial dalam Belajar Machine Learning dengan Python*. <https://dqlab.id/belajar-machine-learning-kenali-2-librarynya-pada-python>

Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. In *Karimah Tauhid* (Vol. 2, Issue 1).

Rosyani, P., Saprudin, S., & Amalia, R. (2021). Klasifikasi Citra Menggunakan Metode Random Forest dan Sequential Minimal Optimization (SMO). *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 9(2), 132.

Seetha, H., Tiwari, V., Anugu, K. R., Makka, D. S., & Karnati, D. R. (2023). A GUI Based Application for PDF Processing Tools Using Python & CustomTkinter. *International Journal for Research in Applied Science and Engineering Technology*, 11(1), 1613–1618.

Setiawan, D. (2022). *Mengenal Editor Visual Studio Code*. <https://teknik-komputer-d3.stekom.ac.id/informasi/baca/Mengenal-Editor-Visual-Studio-Code/a812f5ddfdd241c29f9645ea09e14dedbdae2f1b>

Suaib, A. A., Tritosmoro, I. I., & Ibrahim, N. (2022). IDENTIFIKASI COVID-19 BERDASARKAN CITRA X-RAY PARU-PARU MENGGUNAKAN METODE LOCAL BINARY PATTERN DAN RANDOM FOREST. *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 5(2), 419.

Suci Amaliah, Nusrang, M., & Aswi, A. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa

Bantaeng. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 4(3), 121–127.

Susanti, A. R. (2023, February 23). *Masa Depan Industri Kayu*. <https://www.forestdigest.com/detail/2156/industri-kayu>

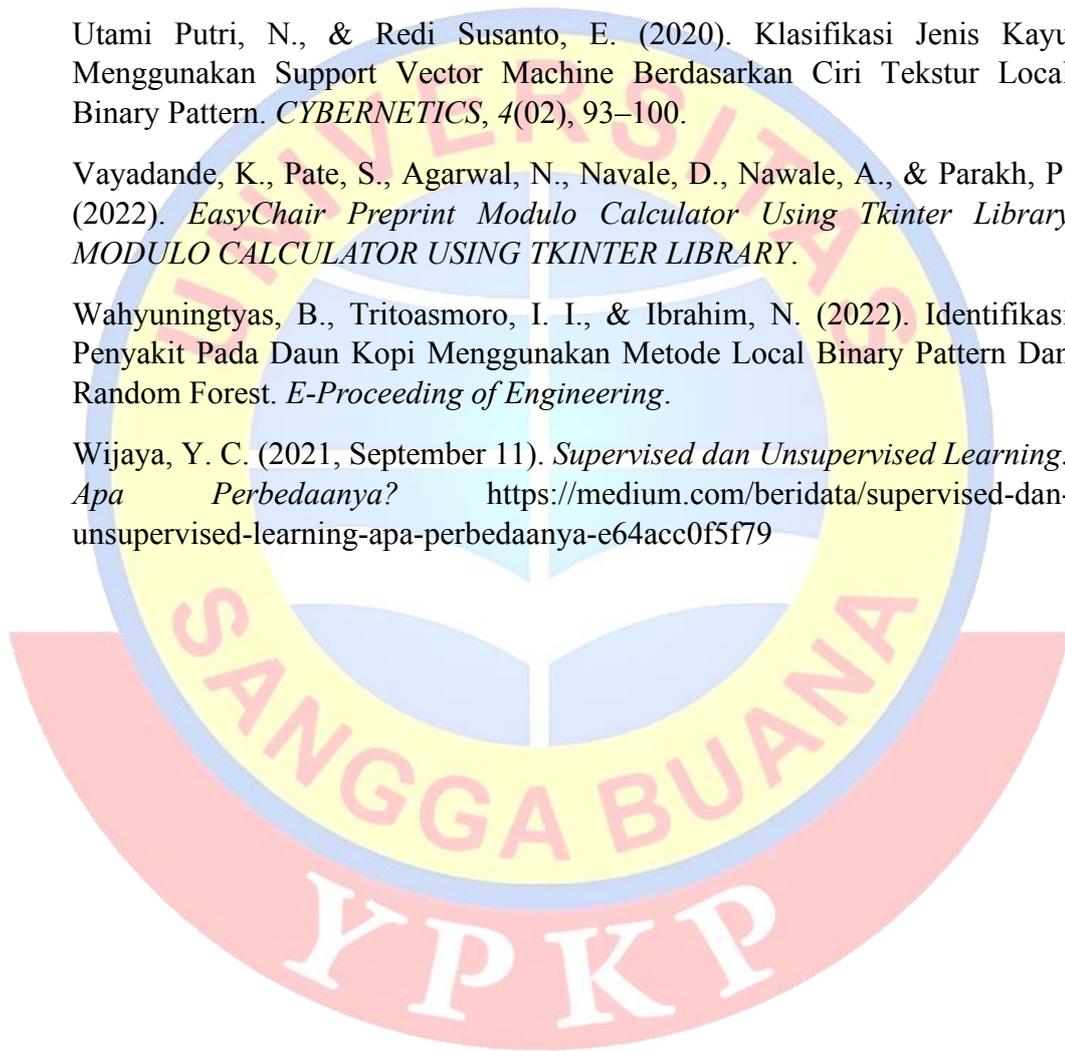
Trivusi. (2022, September 17). *Algoritma Random Forest: Pengertian dan Kegunaannya*. https://www.trivusi.web.id/2022/08/algoritma-random-forest.html#cara_kerja_algoritma_random_forest

Utami Putri, N., & Redi Susanto, E. (2020). Klasifikasi Jenis Kayu Menggunakan Support Vector Machine Berdasarkan Ciri Tekstur Local Binary Pattern. *CYBERNETICS*, 4(02), 93–100.

Vayadande, K., Pate, S., Agarwal, N., Navale, D., Nawale, A., & Parakh, P. (2022). *EasyChair Preprint Modulo Calculator Using Tkinter Library MODULO CALCULATOR USING TKINTER LIBRARY*.

Wahyuningtyas, B., Tritoasmoro, I. I., & Ibrahim, N. (2022). Identifikasi Penyakit Pada Daun Kopi Menggunakan Metode Local Binary Pattern Dan Random Forest. *E-Proceeding of Engineering*.

Wijaya, Y. C. (2021, September 11). *Supervised dan Unsupervised Learning: Apa Perbedaannya?* <https://medium.com/beridata/supervised-dan-unsupervised-learning-apa-perbedaannya-e64acc0f5f79>



LAMPIRAN 1 KARTU BIMBINGAN



KARTU BIMBINGAN SKRIPSI
S1 – TEKNIK INFORMATIKA
 UNIVERSITAS SANGGA BUANA - YPKP

TAHUN AJAR	Genap 2023/2024
NPM	2113201060
NAMA	Muhammad Ghiffaari Ilham Ramadhan
PEMBIMBING	Bambang Sugiarto, ST., M.T.
JUDUL	IMPLEMENTASI ALGORITMA RANDOM FOREST UNTUK IDENTIFIKASI CITRA KAYU MAKROSKOPIK DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)



NO	TANGGAL	POKOK BAHASAN	PARAF PEMBIMBING
1	02/05/24	Pembahasan Struktur Laporan	[Signature]
2	05/05/24	Pembuatan Bab 1	[Signature]
3	22/05/24	Revisi bab 1	[Signature]
4	12/06/24	Pembuatan bab 2 dan 3	[Signature]
5	18/06/24	Revisi bab 3	[Signature]
6	24/06/24	Pembuatan aplikasi	[Signature]
7	10/07/24	Pembuatan bab 4	[Signature]
8	17/07/24	Revisi aplikasi	[Signature]
9	23/07/24	Revisi bab 4	[Signature]
10	26/07/24	Revisi bab 4 dan Pembuatan bab 5	[Signature]
11	01/08/24	Revisi aplikasi	[Signature]
12	02/08/24	Pembuatan PPT	[Signature]
13	06/08/24	Revisi PPT	[Signature]

Cat :

1. Minimal bimbingan sebanyak 8x.
2. Kartu ini dikumpulkan sebagai syarat sidang beserta berkas yang lainnya.

Bandung, 7 Agustus 2024

Pembimbing

Sugiarto

(Bambang Sugiarto, ST., M.T.)

LAMPIRAN 2 SURAT CEK PLAGIARISME



UPT PERPUSTAKAAN
UNIVERSITAS SANGGA BUANA YPKP
Jl. PHH Mustofa No. 68 Bandung Gedung E Lantai 5
Email: library@usbykp.ac.id Website: perpustakaan.usbykp.ac.id

Surat Keterangan Cek Plagiarisme **Nomor: 434/VIII/SKCP/USB-YPKP/2024**

Sehubungan dengan kewajiban Cek Plagiarisme dengan *similarity check maximal 25%* sebagai salah satu kelengkapan persyaratan administrasi bagi mahasiswa tingkat akhir, dengan ini UPT Perpustakaan Universitas Sangga Buana menerangkan bahwa:

Nama : Muhammad Ghiffaari Ilham Ramadhan
NPM : 2113201060
Program Studi : S1 Teknik Informatika
Judul Karya Tulis Ilmiah : "IMPLEMENTASI ALGORITMA RANDOM FOREST UNTUK IDENTIFIKASI CITRA KAYU MAKROSKOPIK DENGAN EKSTRAKSI FITUR GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)"
Tanggal Cek Turnitin : 12 Agustus 2024
Status : Lulus dengan 19% *Similarity Check*

Adalah benar telah dilakukan *similarity check* sebagaimana data tersebut diatas, dan surat ini dibuat berdasarkan keadaan yang sebenar benarnya, untuk bisa dipergunakan sebagaimana mestinya.

Bandung, 12 Agustus 2024

Kepala UPT Perpustakaan

Widayapuri Prasastiningtyas, S.Sos., M.I.kom.

NIP. 432.200.173

LAMPIRAN 3 SURAT PENERIMAAN TA DARI INSTANSI

M. Ghiffaari Ilham .R <ghifari.ir12@gmail.com>

Pemberitahuan Permohonan Layanan Nomor #104077
1 pesan

Notifikasi dari E-Layanan Sains (ELSA) - BRIN <no-reply@brin.go.id> 12 Juli 2023 pukul 18.41
Balas Ke: "no-reply@brin.go.id" <no-reply@brin.go.id>
Kepada: ghifari.ir12@gmail.com

Permohonan layanan nomor #104077 telah disetujui.

Selamat, permohonan layanan nomor #104077 yang Anda ajukan telah disetujui.

Anda dapat mengakses menu Profil > Transaksi lalu tekan tombol Detail Transaksi untuk mengisi logbook, pengajuan lembur, dan mengunduh berkas hasil.

Hormat kami,
Tim E-Layanan Sains - BRIN (ELSA).

DETAIL TRANSAKSI Beranda / Kegiatan dan Bimbingan Ilmiah / Pembimbingan Tugas Akhir Mahasiswa



Pembimbingan Tugas Akhir Mahasiswa (S1/S2/S3) di Bidang Analisis Data Citra dan Video (Image and Video Data Analysis)
DESKRIPSI LAYANAN | Pembimbingan Tugas Akhir terkait topik Analisis Data Citra dan Video (Image and Video Data Analysis)

- Tanggal Transaksi: 12-07-2023
- Tanggal Pelaksanaan Layanan: 17-07-2023 - 17-09-2023
- Pembimbing di BRIN: Bambang Sugianto M.I.
- Total Tagihan: Gratis

Status Proses Unduh

No	Detail	Catatan	Aktor
1	Diterima 2023-07-12 18:41:09	Diterima untuk melakukan KP dan TA.	Bambang Sugianto
2	Memunggu Verifikasi Supervisor Opsi Pertama 2023-07-12 18:03:40		Muhammad Ghiffaari Ilham Ramadhan

LAMPIRAN 4 BALOK KAYU



LAMPIRAN 5 MICROSCOPE MAGNIFIER 60X ZOOM TOOL



LAMPIRAN 6 PROSES PENGAMBILAN CITRA KAYU

