

Aplikasi Penentuan Tanaman Yang Sesuai Pada Dataran Tinggi Menggunakan Metode Random Forest

M. Abil Aqram Alfaridzy^{1*}, Jufri², Nurul Aini²

¹Rekayasa Perangkat Lunak, Universitas Dipa Makassar, Makassar, Indonesia.

²Universitas Dipa Makassar, Makassar, Indonesia.

*¹abilaqram47@gmail.com, ²jufri.ldap@dipanegara.ac.id, ³nurulaini@undipa.ac.id

Abstrak

Pemilihan tanaman yang sesuai untuk lahan pertanian di dataran tinggi sering menjadi tantangan bagi petani, terutama di Desa Garing, Kecamatan Tompobulu, Kabupaten Gowa, yang masih mengandalkan pengalaman pribadi dalam menentukan jenis tanaman. Penelitian ini bertujuan untuk mengembangkan aplikasi penentuan tanaman berbasis Android menggunakan metode Random Forest guna membantu petani dalam memilih tanaman berdasarkan kondisi lingkungan seperti pH tanah, suhu, kelembaban, intensitas cahaya, curah hujan, dan cuaca. Dengan pendekatan kuantitatif, data dikumpulkan melalui observasi lapangan, wawancara dengan petani, serta sumber sekunder untuk melatih model prediksi. Aplikasi ini memungkinkan petani memasukkan data kondisi lahan secara langsung dan mendapatkan rekomendasi tanaman dengan tingkat akurasi tertentu. Hasil pengujian menunjukkan bahwa model Random Forest yang diterapkan memiliki akurasi rata-rata sebesar 68%, dengan performa terbaik pada tanaman kopi dan jagung yang memiliki presisi dan recall tinggi. Dengan adanya aplikasi ini, diharapkan petani dapat mengambil keputusan yang lebih tepat dalam memilih tanaman yang sesuai, sehingga meningkatkan produktivitas lahan dan efisiensi pertanian di Desa Garing.

Kata kunci: Random Forest, Prediksi Tanaman, Dataran Tinggi, Android, Desa Garing.

I. PENDAHULUAN

Kabupaten Gowa, salah satu daerah yang terletak di Sulawesi Selatan, Indonesia memiliki peran penting dalam konteks sejarah, budaya, dan ekonomi regional. Dikenal sebagai pusat kebudayaan Bugis-Makassar, Gowa memiliki warisan sejarah yang kaya, termasuk peninggalan kerajaan yang berpengaruh di masa lalu. Dengan luas wilayah yang signifikan, kabupaten ini memiliki beragam sumber daya alam, termasuk pertanian. Salah satu desa yang terdapat di Kabupaten Gowa adalah Desa Garing Kecamatan Tompo'bulu 41,33% masyarakatnya berprofesi sebagai petani [1].

Berdasarkan hal tersebut, beberapa penelitian sebelumnya telah mengaplikasikan teknologi untuk mendukung sektor pertanian. [2] dalam penelitiannya yang berjudul "Implementasi Algoritma Random Forest Regression untuk Memprediksi Hasil Panen Padi di Desa Minanga" menghasilkan sistem yang mampu memprediksi hasil panen padi berdasarkan tujuh variabel [2]. Selain itu, [3] dalam penelitian "Komparasi Multiple Linear Regression dan Random Forest Regression Dalam Memprediksi Anggaran Biaya Manajemen Proyek Sistem Informasi" menyimpulkan bahwa model Random Forest Regression memberikan nilai akurasi yang lebih tinggi, yaitu 81,5%, dibandingkan dengan model multiple linear regression yang hanya 62,9% [3].

Sebagai solusi untuk meningkatkan produktivitas lahan pertanian, penelitian ini mengedepankan integrasi teknologi dan analisis data. Pendekatan ini menggunakan aplikasi prediksi berbasis algoritma, seperti Random Forest, untuk memperkirakan sesuatu yang akan terjadi di masa depan secara sistematis berdasarkan data masa lalu dan masa kini [2].

Random Forest adalah salah satu teknik yang populer dalam machine learning. Teknik ini dapat diterapkan pada masalah klasifikasi maupun regresi. Konsep dasarnya adalah pembelajaran ensemble, yaitu teknik yang menggabungkan beberapa pengklasifikasi untuk memecahkan masalah yang kompleks [4].

II. METODOLOGI PENELITIAN

Penelitian ini merupakan jenis penelitian pengembangan (Research and Development) dengan pendekatan deskriptif dan kuantitatif. Penelitian ini dilaksanakan di Desa Garing, Kecamatan Tompobulu, Kabupaten Gowa. Pendekatan deskriptif digunakan untuk memahami kondisi lahan di dataran tinggi serta faktor-faktor lingkungan yang memengaruhi kesuburan tanah. Sementara itu, pendekatan kuantitatif digunakan untuk menganalisis hasil prediksi tanaman menggunakan metode Random Forest.

A. Aplikasi

Aplikasi adalah program siap pakai yang dibuat untuk membantu pengguna menyelesaikan tugas atau masalah spesifik. Dalam konteks penelitian, aplikasi berfungsi sebagai alat komputasi untuk mengolah data dan menjalankan perintah sesuai kebutuhan, memastikan hasil yang akurat. Secara umum, aplikasi bisa diartikan sebagai perangkat lunak yang dirancang untuk tujuan tertentu, seperti yang dijelaskan oleh berbagai ahli seperti Jogiyanto (1999:12) yang mendefinisikannya sebagai sekumpulan instruksi yang memproses input menjadi output, atau Kamus Besar Bahasa Indonesia (1998:52) yang menganggapnya sebagai penerapan sistem untuk mengolah data menggunakan bahasa pemrograman. Intinya, aplikasi merupakan perangkat lunak yang memanfaatkan kemampuan komputer secara langsung untuk memenuhi kebutuhan pengguna [5].

B. Python

Python adalah bahasa pemrograman interpretatif dan multiguna yang dikembangkan oleh Guido van Rossum pada tahun 1990. Filosofi utamanya adalah keterbacaan kode dan kesederhanaan, menjadikannya mudah dipahami. Python mendukung berbagai paradigma pemrograman, termasuk berorientasi objek, imperatif, dan fungsional. Sebagai bahasa yang dinamis, ia memiliki manajemen memori otomatis dan pustaka standar yang komprehensif. Keunggulan Python terletak pada sifatnya yang gratis dan open source, serta fleksibel karena dapat berjalan di berbagai sistem operasi seperti Linux, Windows, dan macOS. Selain itu, Python dikenal serbaguna (versatile), cocok untuk pengembangan web, aplikasi desktop, bahkan aplikasi seluler, dan didukung oleh komunitas besar serta perpustakaan (library) yang luas. Kemudahan instalasi pustaka baru melalui perintah pip install juga menjadi salah satu kelebihan utamanya [6].

C. Dart

Dart adalah bahasa pemrograman berorientasi objek yang dirancang dan dikembangkan oleh Google sejak 2007. Diciptakan oleh Lars Bak dan Kasper Lund, Dart didesain untuk pengembangan sisi klien yang optimal, mendukung berbagai jenis aplikasi, termasuk web, seluler (Android dan iOS), desktop, dan server.

Salah satu keunggulan utama Dart adalah sintaksisnya yang mirip dengan C++, Java, dan JavaScript, sehingga memudahkan pengembang yang sudah familiar dengan bahasa-bahasa tersebut untuk mempelajarinya. Dart juga memiliki fitur-fitur modern yang meningkatkan produktivitas, seperti Hot Reload yang memungkinkan pengembang melihat perubahan kode secara real-time saat mengedit. Untuk kinerja, Dart menggunakan teknologi kompilasi Just-in-Time (JIT) untuk mempercepat proses pengembangan, dan Ahead-of-Time (AOT) untuk menghasilkan kode yang sudah dikompilasi sebelumnya, memastikan performa dan waktu loading aplikasi yang cepat [7]. Dukungan dari komunitas pengembang yang kuat juga menjadi nilai tambah, menyediakan banyak library dan sumber daya yang mudah diakses.

D. Flutter

Flutter adalah Software Development Kit (SDK) sumber terbuka dari Google yang dirancang untuk membangun aplikasi mobile dengan kinerja tinggi untuk iOS dan Android dari satu basis kode tunggal. Tujuan utamanya adalah memungkinkan pengembang menciptakan aplikasi yang terasa alami di berbagai platform berbeda. Flutter menjalankan aplikasi melalui dua mode utama:

- 1) **Debug:** Menggunakan kompilasi Just-in-Time (JIT), yang memungkinkan fitur hot reload dan hot restart. Hot reload sangat berguna untuk mempercepat pengembangan karena memungkinkan perubahan kode diterapkan secara instan tanpa kehilangan status aplikasi, sementara hot restart mereset status aplikasi.
- 2) **Release:** Menggunakan kompilasi Ahead-of-Time (AOT), yang menghasilkan aplikasi yang dioptimalkan untuk kecepatan dan ukuran, dengan semua informasi debug dimatikan.

Flutter menggunakan konsep widget untuk membangun antarmuka pengguna (UI), di mana setiap elemen UI, dari tombol hingga teks, adalah sebuah widget. Widget terbagi menjadi dua tipe: Stateful widget yang dapat diperbarui atau digambar ulang saat data berubah, dan Stateless widget yang tidak bisa digambar ulang. Flutter juga menyediakan dua set widget yang berbeda: Material Design (gaya Android) dan Cupertino (gaya iOS), yang keduanya dapat digunakan secara cross-platform [8].

E. Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) adalah protokol tingkat aplikasi yang mengatur komunikasi dan pertukaran data, seperti teks, gambar, video, dan audio, di World Wide Web (WWW). Meskipun port 80 adalah standar, port lain juga dapat digunakan. Sejak versi awalnya HTTP/0.9 pada tahun 1991, protokol ini telah berkembang, dengan versi terbaru, HTTP/3, yang dirilis pada September 2019. Versi ini menggunakan Protokol Datagram Pengguna (UDP) sebagai transportasi dasar alih-alih Protokol Kontrol Transmisi (TCP), yang meningkatkan efisiensi. HTTP memungkinkan komunikasi antara berbagai host dan klien, dan digunakan secara fundamental saat pengguna berinteraksi dengan browser web mereka [9].

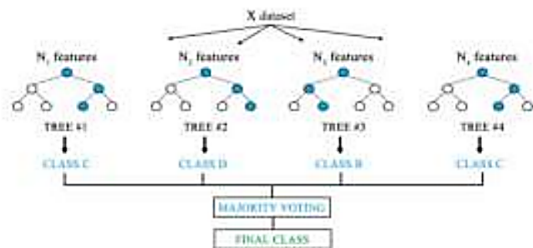
F. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang sangat populer di kalangan pengembang aplikasi web. Berbeda dengan sistem non-relasional, MySQL mengorganisasi informasi ke dalam tabel-tabel yang saling terhubung, memudahkan akses dan pengelolaan data. Untuk mempercepat pencarian, MySQL menggunakan indeks, termasuk primary key atau pengenal unik. MySQL menggunakan bahasa SQL (Structured Query Language) untuk mengelola databasenya dan tersedia dalam lisensi FOSS (Free and Open Source Software) serta versi komersial. MySQL dapat diinstal pada berbagai platform seperti Windows dan Linux, dan administrasinya dapat dipermudah dengan

menggunakan perangkat lunak seperti phpMyAdmin dan MySQL Yog [10].

G. Metode Random Forest

Random Forest adalah salah satu teknik supervised learning yang populer dalam machine learning, digunakan untuk masalah klasifikasi maupun regresi. Metode ini termasuk dalam kategori pembelajaran ensemble, sebuah teknik yang menggabungkan beberapa pengklasifikasi lemah untuk memecahkan masalah kompleks dan meningkatkan kinerja model. Konsep utamanya adalah menciptakan "hutan" dari banyak pohon keputusan (decision trees) yang dilatih secara independen. Setiap pohon dibangun dengan menggunakan subset data pelatihan dan pemilihan fitur yang acak, yang menghasilkan kumpulan pohon dengan bentuk yang beragam. Prediksi akhir dari model Random Forest diperoleh dengan menggabungkan hasil prediksi dari setiap pohon melalui metode mayoritas suara (voting). Korelasi yang kecil antar pohon diharapkan dapat menurunkan tingkat kesalahan prediksi [4].



Gambar 1. Metode Random Forest

Rumus dan Perhitungan

Dalam pembentukan pohon keputusan, terdapat beberapa rumus penting yang digunakan untuk mengukur kemurnian data dan kualitas pemisahan, di antaranya:

1. Rumus Entropy (H)

Mengukur tingkat ketidakteraturan atau ketidakpastian dalam data.

$$H(S) = \sum_{i=1}^n p_i \log_2 p_i \quad (1)$$

Di mana:

$H(S)$ = Entropy dari dataset S

p_i = Probabilitas kemunculan kelas ke- i

n = Jumlah kelas dalam dataset

Entropy bernilai 0 jika semua sampel dalam satu kategori (pasti), dan 1 jika kategori terdistribusi secara merata (tidak pasti).

2. Rumus Information Gain (IG)

Mengukur seberapa besar pengurangan entropy setelah memisahkan dataset berdasarkan suatu fitur.

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2)$$

Di mana:

$IG(S, A)$ = Information Gain setelah membagi dataset S berdasarkan fitur A

$H(S)$ = Entropy sebelum pemisahan S_v = Subset dari S untuk nilai tertentu v dalam fitur A

$\frac{|S_v|}{|S|}$ = Proporsi data dalam subset tersebut

3. Rumus Gini Index

Alternatif untuk entropy, digunakan untuk mengukur kemurnian suatu node. Semakin kecil nilainya, semakin murni node tersebut (Sevani et al., 2009).

$$Gini(S) = 1 - \sum_{i=1}^n P_i^2 \quad (3)$$

Di mana:

$Gini(S)$ = Gini Index dari dataset S

p_i = Proporsi sampel kelas ke- i

Semakin kecil Gini Index, semakin murni node tersebut (lebih baik untuk klasifikasi).

H. Penentuan

Bagian ini membahas pendekatan yang digunakan dalam penelitian untuk menentukan rekomendasi atau kesesuaian, terutama dalam konteks pertanian. Penentuan pilihan rekomendasi dievaluasi menggunakan pendekatan multi nutrient response [11]. Pendekatan ini memanfaatkan model kuadratik dari percobaan pemupukan untuk menentukan dosis pemupukan maksimum.

Rumus untuk menentukan dosis maksimum ini adalah:

$$R = a + bX - cX^2 \quad (4)$$

Di mana:

R = nilai relatif tanaman

X = konsentrasi pupuk

a, b, c = konstanta

Dosis pupuk maksimum (X) dapat diketahui dengan menggunakan rumus turunan dari persamaan regresi, yaitu:

$$X = b/2c \quad (5)$$

Selain itu, teks juga menyoroti pentingnya penentuan waktu tanam yang tepat untuk mencegah hilangnya nutrisi, terutama saat transisi musim. Informasi curah hujan harian menjadi parameter yang sangat penting untuk penentuan waktu tanam dan panen yang akurat [12].

Penetapan curah hujan sebagai parameter penentu kesesuaian lahan untuk tanaman pangan dilakukan setelah mengamati pola data dari 14 jenis tanaman pangan. Hasil pengamatan menunjukkan bahwa pola curah hujan memiliki perbedaan yang paling signifikan dibandingkan parameter lainnya, menjadikannya faktor kunci dalam menentukan kesesuaian lahan.

I. Data

Menurut Webster New World Dictionary, data didefinisikan sebagai "sesuatu yang diketahui atau diasumsikan." Data adalah sekumpulan informasi atau nilai yang diperoleh dari pengamatan suatu objek, yang bisa berupa angka, lambang, atau sifat. Beberapa jenis data mencakup data populasi, data sampel, data primer, dan data sekunder.

Pada dasarnya, data yang telah diolah dan dianalisis berfungsi sebagai dasar yang objektif untuk pengambilan keputusan. Oleh karena itu, data yang baik harus memiliki tiga kualitas utama: dapat dipercaya kebenarannya (reliable), tepat waktu, dan relevan, yang artinya dapat memberikan gambaran masalah secara menyeluruh. Proses riset menghasilkan data dalam tiga tingkatan: data mentah (hasil pengumpulan), data hasil pengolahan (berupa jumlah atau persentase), dan data

hasil analisis (berupa kesimpulan), yang merupakan tingkatan tertinggi karena dapat langsung digunakan sebagai dasar pembuatan keputusan [13].

J. Pemodelan Sistem

Desain sistem aplikasi ini menggunakan Unified Modeling Language (UML), sebuah bahasa visual yang digunakan untuk merencanakan dan membuat program berorientasi objek. UML berfungsi sebagai standar untuk mengumpulkan rencana dan gambaran umum, menggabungkan siklus bisnis, dan menyusun kelas dalam dialek pemrograman tertentu [14].

1. Diagram Usecase

Use Case menggambarkan hubungan antara setidaknya satu penghibur dan kerangka data yang akan dibuat [15].

Tabel 1. Diagram Usecase

Simbol	Nama	Deskripsi
	Aktor	Individu, proses, atau kerangka kerja berbeda yang bekerja sama dengan kerangkadata akan dijadikan di luar kerangka data.
	Use Case	Fungsional yang disediakan oleh sistem berperan sebagai unit-unit yang saling berkomunikasi dengan bertukar pesan antara satu unit dan aktor lainnya.
	Asosiasi	Komunikasi antar aktor dan use case yang berpartisipasi.
	<<include>>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya.
	<<extend>>	Hubungan kasus penggunaan tambahan dengan situasi pemanfaatan di mana kasus penggunaan tambahan dapat tetap berdiri sendiri bahkan tanpa kasus penggunaan tambahan.
	Generalisasi	Hubungan spekulasi dan spesialisasi antara dua situasi penggunaan dimana kemampuan salah satu lebih luas dibandingkan yang lain.

Sumber: [15]

2. Diagram Class

Diagram kelas mengilustrasikan struktur sistem dengan cara mendefinisikan kelas-kelas yang diperlukan untuk membangun sistem. Setiap kelas memiliki atribut dan metode, juga dikenal sebagai operasi [15].

Tabel 2. Diagram Class

Simbol	Nama	Deskripsi
	Kelas	Kelas pada struktur sistem.
	Antarmuka	Sama dengan konsep antarmuka dalam pemrograman berorientasi objek.
	Asosiasi	Erlasi antar kelas dengan makna

Simbol	Nama	Deskripsi
	Asosiasi berarah	Hubungan antar kelas dengan kepentingan suatu kelas dimanfaatkan oleh kelas yanglain, afiliasi biasanya juga disertai dengan variasi.
	Generalisasi	Hubungan antar kelas dengan signifikansi spekulasi spesialisasi (umum khusus).
	Kebergantungan	Relasi antar kelas dengan makna kebergantungan antarkelas.
	Agregasi	Relasi antar kelas dengan makna semua-bagian

Sumber: [15]

3. Diagram Activity

Diagram aktivitas menunjukkan bagaimana sistem perangkat lunak, proses bisnis, atau menu bekerja bersama.

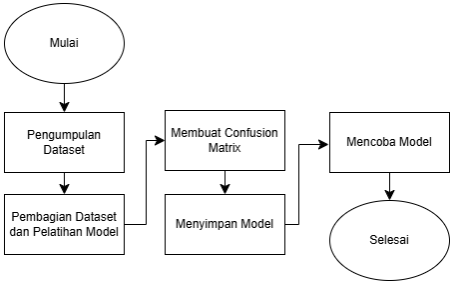
Tabel 3. Diagram Activity

Simbol	Nama	Deskripsi
	Status awal	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status Akhir	Kondisi terakhir dari kerangka kerja, garis besar tindakan memiliki keadaan akhir.

Sumber: [15].

III. HASIL DAN PEMBAHASAN

Berikut perancangan solusi yang dibuat penulis untuk penyelesaian masalah dalam penelitian ini.



Gambar 2. Perancangan Solusi

Pelatihan model Random Forest untuk mengidentifikasi jenis tanaman di dataran tinggi adalah proses yang terstruktur. Tahap awal melibatkan pengumpulan data dari observasi langsung di lapangan dan sumber publik seperti Kaggle. Setelah data terkumpul, data tersebut dibagi menjadi dua set: data pelatihan untuk mengajar model dan data pengujian untuk mengevaluasi kinerjanya. Kemudian, model dilatih menggunakan data pelatihan yang sudah disiapkan. Setelah pelatihan selesai, model diuji secara menyeluruh. Pengujian ini tidak hanya mengukur seberapa akurat model, tetapi juga menganalisis confusion matrix untuk mengetahui jenis kesalahan yang mungkin terjadi, seperti salah mengidentifikasi satu jenis tanaman dengan yang lain. Terakhir, model yang sudah teruji dan optimal akan disimpan dan siap digunakan untuk memprediksi jenis tanaman yang cocok di suatu area.

Koleksi data yang diambil terbagi dua, 40% dari data pribadi dan 60% dari kaggle. Data primer yang diperoleh dari penelitian lapangan mencakup nama tanaman beserta parameter yang diukur secara langsung pada lokasi penelitian. Sementara itu, data sekunder yang diperoleh dari Kaggle memuat parameter yang lebih lengkap sehingga dapat menunjang peningkatan akurasi hasil prediksi. Total dataset yang dikumpulkan 148 yang terdiri atas 8 label tanaman.

Tabel 4. Dataset

No	Label	Jumlah
1	Pisang	21
2	Kelapa	20
3	Mangga	20
4	Melon	20
5	Jagung	12
6	Padi	15
7	Kopi	20
8	Anggur	20
Total		148

Dilanjutkan bagian ini, data yang telah dipersiapkan dipisahkan menjadi fitur (X) dan label (y). Fitur yang digunakan terdiri dari berbagai variabel yang relevan untuk memprediksi jenis tanaman yang cocok, sementara label berisi kategori tanaman yang sesuai. Data kemudian dibagi menjadi dua set, yaitu data pelatihan (X_train, y_train) yang digunakan untuk melatih model, dan data pengujian (X_test, y_test) yang digunakan untuk mengevaluasi performa model, dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian. Model pembelajaran mesin yang digunakan adalah Random Forest, yang dilatih menggunakan data pelatihan. Proses pelatihan dilakukan menggunakan metode.fit(), dan hasilnya menunjukkan bahwa pelatihan telah selesai dengan pesan "Training completed". Dengan pembagian data yang tepat dan penggunaan model Random Forest, langkah ini bertujuan untuk mengoptimalkan prediksi jenis tanaman yang cocok berdasarkan kondisi yang ada.

```
X = df.drop('label', axis=1)
y = df['label']

class_names = np.unique(y)

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print("X_train shape: (X_train.shape)")
print("X_test shape: (X_test.shape)")

# Classifier
random_forest = RandomForestClassifier(n_estimators=100, random_state=42)

# Train model
random_forest.fit(X_train, y_train)

print("Training completed.")

X_train shape: (799, 5)
X_test shape: (200, 5)

Training completed.
```

Gambar 3. Pelatihan Model Random Forest

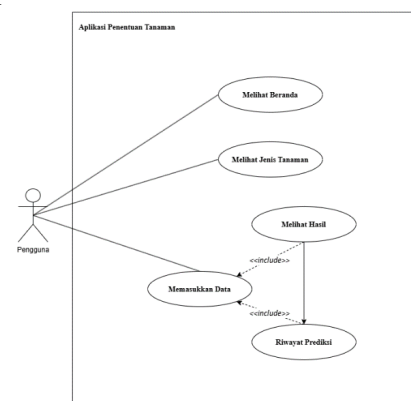
Kode di atas memproses data dengan memisahkan fitur (X) dan label (y), di mana fitur berisi variabel yang digunakan untuk prediksi, sedangkan label menunjukkan jenis tanaman yang sesuai. Selanjutnya, data dibagi menjadi 80% untuk pelatihan (X_train, y_train) sebanyak 799 sampel dan 20% untuk pengujian (X_test, y_test) sebanyak 200 sampel, dengan masing-masing sampel memiliki 5 fitur. Model Random Forest dengan 100 pohon keputusan kemudian dilatih menggunakan data pelatihan. Setelah proses pelatihan selesai, ditampilkan pesan "Training completed.", yang menandakan bahwa model telah berhasil belajar dari data pelatihan dan siap untuk digunakan dalam evaluasi serta prediksi jenis tanaman yang direkomendasikan.

Berikut adalah implementasi Random Forest dengan melakukan perhitungan manual.

Tabel 5. Hasil Perhitungan Manual

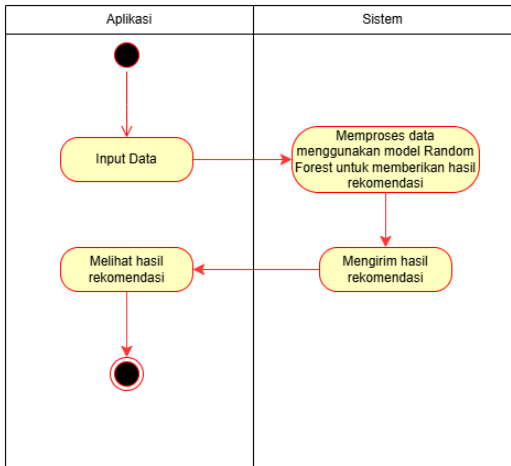
No	pH	Suhu	Cuaca	Kelembapan	Tanaman
1	pH7	30°C	980	5%	Kopi
2	pH7	31°C	500	10%	Padi
3	pH9	32°C	200	1%	Jagung
4	pH6	30°C	180	9%	Pisang
5	pH6	30°C	850	8%	Kelapa

Setelah dilakukan percobaan secara manual dengan menggunakan dataset, aplikasi terbukti mampu menentukan jenis tanaman yang tepat berdasarkan inputan parameter yang terdapat pada tabel dataset. Hasil uji coba menunjukkan bahwa setiap parameter yang dimasukkan, seperti suhu, kelembapan, pH tanah, maupun cuaca, dapat diproses dengan baik sehingga aplikasi dapat memberikan rekomendasi tanaman yang sesuai.



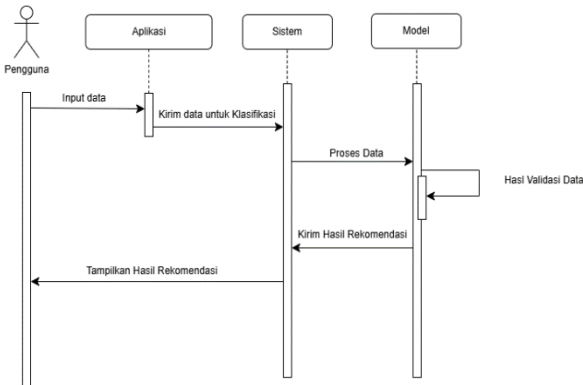
Gambar 4. Diagram Usecase

Diagram use case ini menggambarkan interaksi pengguna dengan aplikasi prediksi tanaman. Pengguna, sebagai aktor utama, dapat berinteraksi dengan aplikasi untuk melihat beranda, memasukkan data prediksi, melihat hasil prediksi, dan melihat riwayat prediksi. Untuk melakukan prediksi, pengguna harus memasukkan data terlebih dahulu, yang kemudian akan menghasilkan hasil prediksi dan mencatat riwayatnya. Selain itu, pengguna juga memiliki opsi untuk melihat informasi tentang aplikasi.



Gambar 5. Activity Diagram

Activity diagram ini menggambarkan alur kerja sistem rekomendasi berbasis model Random Forest, di mana aplikasi menerima input data dari pengguna, mengirimkannya ke sistem untuk diproses, sistem menggunakan model Random Forest untuk menghasilkan rekomendasi, mengirimkan hasil rekomendasi kembali ke aplikasi, dan aplikasi menampilkan hasil tersebut kepada pengguna.

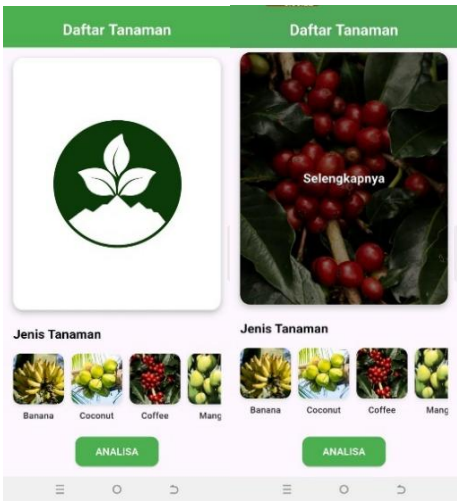


Gambar 6. Sequence Diagram

Sequence Diagram ini menggambarkan interaksi antara pengguna, aplikasi, sistem, dan model dalam sistem rekomendasi, dimulai dari pengguna memasukkan data, aplikasi mengirimkan data ke sistem untuk klasifikasi, sistem meneruskan data ke model untuk diproses dan divalidasi, model mengembalikan hasil validasi ke sistem, sistem mengirimkan hasil rekomendasi ke aplikasi, dan aplikasi menampilkan hasil tersebut kepada pengguna.

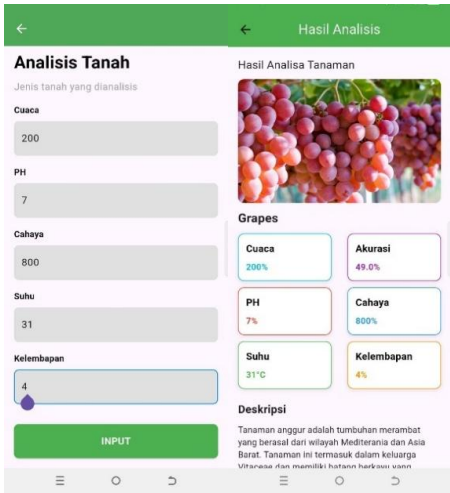
Masuk ke tahap pengujian, akan dilakukan perancangan uji coba pada sistem yang telah dibangun. Untuk menguji apakah aplikasi berjalan sesuai dengan yang diharapkan. Berikut merupakan hasil pengujian yang telah penulis lakukan dapat dilihat pada tabel berikut ini:

Tabel 6. Tahap Pengujian		
Teks Faktor	Keterangan	Hasil
Menekan tombol jenis tanaman	Menampilkan jenis tanaman yang telah dipilih dari daftar, kemudian menampilkan deskripsi dan gambar tanaman tersebut pada bagian logo aplikasi.	Berhasil
Menekan tombol analisa	Berpindah kehalaman berikutnya, yaitu penginputan data	Berhasil

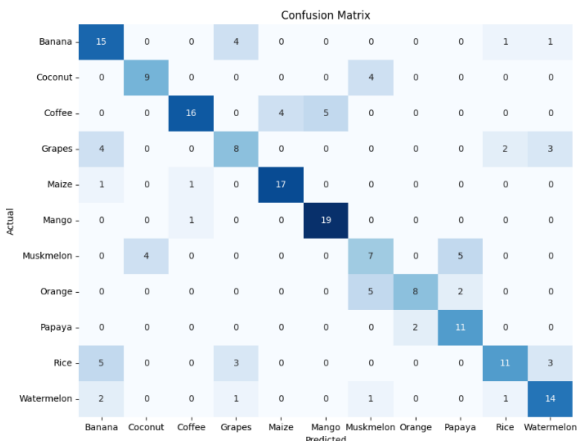


Tabel 7. Tahap Pengujian

Teks Faktor	Keterangan	Hasil
Menekan tombol input	Melakukan penginputan dan menampilkan hasil analisa pada halaman hasil analisis	Berhasil
Mengosongkan kolom inputan	Proses penginputan dinyatakan gagal apabila tidak mengisi data pada kolom yang tersedia	Gagal



Sehingga ketika di uji menggunakan confusion matrix seperti pada gambar confusion matrix di bawah, dapat dihitung nilai akurasi, presisi, dan recall untuk masing masing kelas menggunakan rumus berikut: Sebagai contoh, pada kelas Banana perhitungannya sebagai berikut:



Gambar 7. Pengujian Confussion Matrix

Akurasi = $b/n \times 100\%$
Akurasi = $135/200 \times 100\% = 68\%$
Presisi = $TP/(TP+FP) \times 100\%$
Presisi = $15/(15+0+0+4+1+0+0+0+0+5+2) \times 100\% = 56\%$
Recall = $TP/(TP+FN) \times 100\%$
Recall = $15/(15+0+0+4+0+0+0+0+0+0+1+1) \times 100\% = 71\%$

Laporan klasifikasi berikut ini menunjukkan bahwa model memiliki akurasi keseluruhan sebesar 69%. Performa model sangat baik dalam mengidentifikasi kelas "Banana" dan "Maize", ditunjukkan oleh nilai presisi dan recall yang tinggi. Ini berarti model mampu mengklasifikasikan kedua tanaman tersebut dengan efektif. Namun, model menunjukkan kinerja yang kurang baik pada kelas "Muskmelon", dengan presisi dan recall yang lebih rendah. Hal ini menandakan bahwa model masih kurang efektif dalam mengidentifikasi tanaman ini. Secara umum, meskipun model memiliki kinerja yang cukup baik, masih diperlukan perbaikan pada beberapa kelas agar akurasi dan konsistensinya dapat meningkat.

	precision	recall	f1-score	support
Banana	0.56	0.71	0.62	21
Coconut	0.69	0.69	0.69	13
Coffee	0.89	0.64	0.74	25
Grapes	0.50	0.47	0.48	17
Maize	0.81	0.89	0.85	19
Mango	0.79	0.95	0.86	20
Muskmelon	0.41	0.44	0.42	16
Orange	0.80	0.53	0.64	15
Papaya	0.61	0.85	0.71	13
Rice	0.73	0.50	0.59	22
Watermelon	0.67	0.74	0.70	19
accuracy			0.68	200
macro avg	0.68	0.67	0.67	200
weighted avg	0.69	0.68	0.67	200

Accuracy: 0.6750
Precision: 0.6897
Recall: 0.6750
F1-Score: 0.6719

Gambar 8. Pengujian Hasil Random Forest

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa model Random Forest menunjukkan kinerja yang cukup baik dalam mengklasifikasikan jenis-jenis

tanaman dengan akurasi keseluruhan sebesar 69%. Model ini sangat efektif dalam mengidentifikasi tanaman "Banana" dan "Maize", yang ditunjukkan oleh nilai presisi dan recall yang tinggi. Hal ini menandakan bahwa model memiliki kemampuan yang andal dalam mengklasifikasi kedua jenis tanaman tersebut. Namun, kinerja model menurun pada kelas "Muskmelon", di mana nilai presisi dan recall lebih rendah. Ini menunjukkan adanya kelemahan dalam kemampuan model untuk mengidentifikasi tanaman ini secara akurat.

IV. KESIMPULAN

Setelah melakukan proses penginputan data secara manual pada dataset, dapat disimpulkan bahwa aplikasi berjalan dengan baik dan mampu memberikan rekomendasi tanaman yang sesuai berdasarkan data yang dimasukkan. Parameter yang digunakan dalam penginputan meliputi pH tanah, suhu, kondisi cuaca, dan kelembapan. Hasil uji coba menunjukkan bahwa aplikasi dapat mengolah data tersebut dengan tepat sehingga memberikan keluaran berupa tanaman yang sesuai dengan kondisi lahan yang diinputkan.

Hasil pengujian klasifikasi menggunakan confusion matrix menunjukkan bahwa model memiliki akurasi keseluruhan sebesar 69%. Pada kelas "Banana", model menunjukkan performa yang sangat baik dengan nilai presisi dan recall yang tinggi. Hal ini membuktikan bahwa model mampu mengenali dan mengklasifikasikan tanaman pisang dengan cukup baik dibandingkan dengan kelas tanaman lainnya.

Dengan ini, sistem telah berjalan dengan benar dan implementasi masih dilakukan. Dengan menggunakan algoritma Sequential Search, proses pencarian buku di sistem menjadi lebih mudah. Hal ini memungkinkan siswa dan guru untuk menemukan informasi dengan efisien, sehingga meningkatkan pengalaman pengguna dalam mengakses koleksi perpustakaan.

V. SARAN

Saran yang dapat diberikan oleh penulis untuk mengembangkan Aplikasi Penentuan Tanaman Yang Sesuai Pada Dataran Tinggi Di Desa Garing yaitu:

- 1) Menentukan Rekomendasi tanaman lain, pengembangan model ini dapat ditingkatkan untuk merekomendasikan lebih banyak jenis tanaman, tidak terbatas pada beberapa pilihan saja. Hal ini akan sangat bermanfaat dalam mendukung pertanian yang beragam dan membantu petani di Desa Garing dalam menentukan tanaman yang paling sesuai untuk ditanam pada dataran tinggi.
- 2) Penelitian selanjutnya diharapkan dapat memperluas cakupan data dengan menambahkan lebih banyak jenis tanaman serta variabel pendukung seperti jenis tanah, kandungan nutrisi, topografi, musim tanam, ketersediaan air, potensi hama, dan faktor sosial ekonomi. Dengan data yang lebih beragam dan komprehensif, aplikasi ini diharapkan mampu memberikan rekomendasi tanaman yang lebih akurat, relevan, dan bermanfaat bagi petani dalam

meningkatkan produktivitas lahan serta efektivitas pengambilan keputusan.

REFERENSI

- [1] 2019 Elpi Sukaesi, "PERUBAHAN SOSIAL MASYARAKAT PASCA KONFLIK PERALIHAN," 2019.
- [2] N. Nur, F. Wajidi, S. Sulfayanti, and 2023 Wildayani, "Implementasi Algoritma Random Forest Regression untuk Memprediksi Hasil Panen Padi di Desa Minanga," *J. Komput. Terap.*, vol. 9, no. 1, pp. 58–64, 2023.
- [3] S. E. K. Sihombing, "Komparasi Multiple Linear Regression dan Random Forest Regression Dalam Memprediksi Anggaran Biaya Manajemen Proyek Sistem Informasi," *J. Comput. Digit. Bus.*, vol. 3, no. 2, pp. 86–97, 2024.
- [4] S. Saifullah, M. Zarlis, Z. Zakaria, and R. W. Sembiring, "Analisa Terhadap Perbandingan Algoritma Decision Tree Dengan Algoritma Random Tree Untuk Pre-Processing Data," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 1, no. 2, pp. 180–185, 2017.
- [5] H. Abdurahman, A. R. Riswaya, and A. Id, "Aplikasi pinjaman pembayaran secara kredit pada bank yudha bhakti," *J. Comput. Bisnis*, vol. 8, no. 2, pp. 61–69, 2014.
- [6] A. N. Syahrudin and T. Kurniawan, "Input dan output pada bahasa pemrograman python," *J. Dasar Pemograman Python STMIK*, vol. 20, pp. 1–7, 2018.
- [7] K. Tampubolon, "Pelatihan Pengenalan Dasar Bahasa Dart Dalam Pemograman OOP," *ORAHUA J. Pengabd. Kpd. Masy.*, vol. 1, no. 01, pp. 24–29, 2023.
- [8] S. Tjandra and G. S. Chandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *J. Inf. Syst. Graph. Hosp. Technol.*, vol. 2, no. 02, pp. 76–81, 2020.
- [9] M. K. C. E. H. C. Rometdo Muzawi, *Pengantar Konsep Dasar Http*. Serasi Media Teknologi, 2024.
- [10] A. Lutfi, "Sistem Informasi Akademik Madrasah Aliyah Salafiyah Syafi'iyah Menggunakan Php Dan Mysql," *J. AiTech*, vol. 3, no. 2, pp. 104–112, 2017.
- [11] S. A. D. A. D. Lestari, M. Melati, and H. Purnamawati, "Penentuan dosis optimum pemupukan n, p, dan k pada tanaman kacang Bogor [Vigna subterranea (L.) Verdcourt]," *Indones. J. Agron.*, vol. 43, no. 3, pp. 193–200, 2015.
- [12] K. S. Hariyanti, T. June, Y. Koesmaryono, R. Hidayat, and A. Pramudia, "Penentuan waktu tanam dan kebutuhan air tanaman padi, jagung, kedelai dan bawang merah di Provinsi Jawa Barat dan Nusa Tenggara Timur," *J. Tanah dan Iklim*, vol. 43, no. 1, pp. 83–92, 2019.
- [13] M. A. Fitriani Dwi Ramadhani, *SISTEM PREDIKSI PENJUALAN DENGAN METODE SINGLE EXPONENTIAL SMOOTHING DAN TREND PARABOLIK*. Pascal Books, 2022.
- [14] M. T. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik

- Berbasis Web,” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [15] W. Aprianti, U. Maliha, J. T. Informatika, and P. Negeri, “Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati Kabupaten Tanah Laut,” *J. Sains dan Inform. ISSN*, vol. 2460, p. 173X, 2016.